Experimental and Analytical Investigation of Early Frost Growth on Surfaces with Varying Wettability under Typical Heat Pump Operating Conditions

by

Ellyn N. Harges

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
May 1, 2021

Keywords: condensation frosting; droplet growth and coalescence; crystal growth; frost layer full growth

Approved by

Dr. Lorenzo Cremaschi, Chair, Associate Professor of Mechanical Engineering
Dr. Sushil Bhavnani, Henry M. Burt, Jr. Professor of Mechanical Engineering
Dr. Daniel Mackowski, Professor Emeritus of Mechanical Engineering
Dr. Jianjun Dong, Professor of Physics
Dr. Lauren Beckingham, Assistant Professor of Civil Engineering

Abstract

Frost formation on the fins of the outdoor evaporators of heat pump systems during winter months is a naturally occurring phenomenon with adverse effects on heat pump operation. The frost impedes heat transfer by adding an extra layer of thermal resistance to the fins, and it increases the air-side pressure drop by blocking the airflow through the fins. Over the past few decades, researchers have investigated using surface coatings with different wettability to inhibit frost growth passively. Some of these studies showed promising results, yet other studies' results were inconclusive or even contradictory.

At environmental conditions typical of heat pump operation, frost formation generally occurs in three distinct steps: liquid droplet growth and freezing, crystal growth, and frost layer growth. Surface wettability, often expressed by the contact angle of water on the surface, primarily affects the first step in the process: droplet growth and freezing. This research aims to enhance the understanding of surface wettability's effects on frost growth, emphasizing the first two stages of the frosting process.

Experiments were performed to investigate early frost growth behavior on a bare aluminum surface, a hydrophobic surface, and a hydrophilic surface. Droplet freezing time, frozen droplet diameter, droplet surface area coverage and distribution, droplet coalescence rate, frost layer thickness, and frost layer density were compared for several sets of test conditions on all three surfaces. It was observed that both surface contact angle and test conditions had significant effects on droplet geometry at the time of freezing. It was also observed that environmental parameters governed droplet freezing time at typical heat pump operating conditions, while the contact angle effects were inconclusive.

A semi-empirical model was developed and implemented in a simulation tool to describe frost growth behavior on various surface wettability types. The model is a multi-stage model and includes all three steps of the droplet growth process. The model’s droplet growth and crystal growth stages were developed as part of this research, then were coupled to a previously-existing frost layer growth model. It was initially developed to predict average frost properties on flat plates in convective airflow. However, it was also extended to predict the frost layer's changing behavior in the direction of airflow. It also accounted for multiple wettability types on the same surface and was extended for use in parallel plate channels, which mimic folded-flat fins of heat exchangers. This model's principles can be extended to many other surface geometries. They could be useful in investigating early frost growth for heat exchanger fin configurations where visualization and measurement of frost properties are difficult.

Acknowledgments

I want to thank Dr. Lorenzo Cremaschi for guiding me through this Ph.D. Under his guidance, I learned how to conduct state-of-the-art research and communicate my findings to the scientific community. He helped me to grow professionally and become confident in my abilities. He also offered insight and encouragement as I learned to succeed simultaneously in my studies and new parenthood. I believe that I have grown and improved as a person because of my interactions with him over the last several years.

I want to thank my committee members and university reader, Dr. Bhavnani, Dr. Mackowski, Dr. Dong, and Dr. Beckingham. They have offered valuable insight and support to this project.

I want to thank other students in Dr. Cremaschi's lab. This study would not have been possible without Burak Adanur, and Amy Strong and Fuqiang Ren provided invaluable assistance in data collection and analysis. Dr. Andrea Bigi played a large part in helping me learn to turn modeling principles into a useful simulation tool. Pratik Deokar, Josh Rothe, Ford Gibbes, Stefano Morcelli, Maliha Yel Mahi, Gerard Muteba, and Jerin Ebanesar, along with the others already mentioned, have not only been colleagues, but friends. Thank you for your friendship.

I want to thank those who supported this work materially and financially. The National Science Foundation Chemical, Bioengineering, Environmental, and Transport Systems Division supported the present work through Award No. 1604084. The Alabama State EPSCoR Graduate Research Scholars Program supported my work on this project as well. Dr. Amy Betz and her research group at Kansas State University provided the coatings for the test plates used in this study.

This work would also not have been possible without others in my life who supported me in a different way. My husband, Will, has been patient, understanding, and encouraging through everything, and challenged me to keep going when I felt discouraged. My son, William, has been patient with me as well and has brought so much joy to the journey. My parents taught me a love of learning, which led to a desire to pursue this opportunity. And finally, God has walked with me through it all, continually reminding me of what truly matters and makes life worth living.

I would not be here without all of you, and I can never thank you enough.

Table of Contents

List of Tables

List of Figures

List of Abbreviations and Nomenclature

English Symbols

A	area [m^2]
AC	area coverage [-]
c	droplet growth coefficient [m/s]
c	crystal aspect ratio [-]
c_p	specific heat capacity [J/kg-K]
D_A	surface-area-averaged droplet diameter [m]
D_{AB}	mass diffusivity [m^2/s]
D_h	hydraulic diameter [m]: $D_h = 4(area)/(perimeter)$
D_V	volumetrically averaged droplet diameter [m]
f	friction factor [-]
Fo	Fourier number [-]
h	heat transfer coefficient [W/m^2-K]
H	channel height [m]
H_{fg}	latent heat of evaporation [J/kg]
h_m	mass transfer coefficient [m/s]
H_{sg}	latent heat of sublimation [J/kg]
Ja	Jakob number [-]
k	thermal conductivity [W/m-K]
k_B	Boltzmann constant = 1.38 x 10^{-23} J/K

k_{eff}	effective thermal conductivity during crystal growth [W/m-K]
K_T	isothermal compressibility [m^2/N]
L	plate length in the direction of airflow [m]
L_c	crystal height [m]
Le	Lewis number [-]
m	mass [g]
$\dot{m}$	mass flow rate [kg/s]
$\dot{m}''$	mass flux [kg/m^2-s]
$\dot{m}''_\delta$	mass flux to increase frost thickness [kg/m^2-s]
$\dot{m}''_\rho$	mass flux to increase frost density [kg/m^2-s]
n	current segment number [-]
N''	droplet number density [1/m^2]
Nu	Nusselt number [-]
Nu_∞	fully developed Nusselt number [-]
P	air pressure [Pa]
P_v	vapor pressure [Pa]
Pr	Prandtl number [-]
q''	heat flux [W/m^2]
$\dot{Q}$	heat transfer rate [W]
r	radius [m]
R	specific gas constant [J/kg-K]
R_v	specific gas constant for water vapor [J/kg-K]

Ra	Rayleigh number [-]: $Ra = g\beta\left(T_\infty - T_f\right)L^3 Pr/v^2$
Re_D	Reynolds number for internal flow [-]: $Re_D = \rho V D_h/\mu$
Re_L	Reynolds number for external flow over a flat plater [-]: $Re_L = \rho V L/\mu$
Re_x	local Reynolds number for external flow over a flat plater [-]: $Re_x = \rho V x/\mu$
RH	relative humidity [-]
SSD	supersaturation degree [-]
T	temperature [K]
t	time [s]
t_f	freezing time [s]
v	molecular volume [m^3/mol]
V	air velocity [m/s]
Vol	volume [m^3]
W	plate or channel width [m]
x	distance along the surface in the direction of airflow [m]
y_f	frost seed height [m]

Greek Symbols

α	thermal diffusivity [m^2/s]
α_k	weighting factor for effective thermal conductivity [-]
β	weighting factor for air void temperature calculations [-]
Δ	difference [-]
δ_0	frost thickness at the end of droplet growth [m]

δ_f	frost thickness [m]
ε_c	crystal volume ratio (crystal growth) [-]; porosity (frost layer growth) [-]
θ	contact angle [radians]
μ	coalescence exponent [-]
ρ	density [kg/m^3]
ρ_v	vapor density [kg/m^3]
σ	surface tension [N/m]
ω	absolute humidity ratio [kg/kg]

Subscripts

avg	average
c	crystal
$cond$	conduction
$conv$	convection
$curv$	curvature
d	droplet
dew	dew point
f	frost
fs	frost surface
i	droplet/air interface
$proj$	projected
s	surface
sat	saturation

$sens$	sensible
tp	triple point
w	liquid water
x	local values
∞	bulk air stream

1. Introduction

1.1 Motivations for the Research

The formation of frost on outdoor surfaces is a natural and common occurrence during winter months, but it negatively impacts many different systems' operations. Understanding frost formation and being able to create robust frost-mitigating surfaces can reduce some of these undesirable consequences. For example, diminishing frost growth can help save energy by increasing the efficiency of air-source heat pumps and refrigeration systems (Iragorry et al., 2004; Xia et al., 2006). It can also improve power transmission and wind turbine performance (Jung et al., 2011) and improve aircraft safety and reliability by preventing ice formation on planes' wings (Lambregts et al., 2008).

Several researchers have experimentally investigated frost growth on surfaces of varying wettability in an attempt to inhibit frost growth passively, and these investigations show promising results. In these studies, data was usually provided for thickness and density of an overall porous frost layer, as in Shin et al. (2003). For the range of environmental conditions that are common for air-source heat pumps, frost generally grows in three distinct stages: condensation, growth, and freezing of supercooled liquid water droplets, growth of vertical crystals from the tops of those droplets, and growth of a homogeneous frost layer with average thermodynamic properties (Hoke et al., 2007). Furthermore, Nath et al. (2017) divided the droplet growth and freezing stage into five processes: supercooled condensation, onset of freezing, frost halos, inter-droplet ice bridging and dry zones, and percolation clusters and frost densification. The droplet growth and freezing stage is affected by the surface energy, i.e., the surface's contact angle. The effect of surface wettability, summarized by the measured surface contact angle in this research, should be isolated from the effect due to the surface's initial conditions (such as whether the surface had been cleaned) to quantify frost nucleation's actual dependence on surface energy. The crystal growth stage is only indirectly affected by the contact angle as the crystals grow directly from the frozen droplets. The growth of the homogeneous frost layer is not affected.

Many researchers have developed frosting models that predict frost characteristics on bare metal surfaces. However, few researchers developed models that predicted frosting behavior on different surface types, where the surface's wettability was quantified using the contact angle of water on the surface. Most of these models, such as those presented by Shin et al. (2003) and Hermes et al. (2019), used the contact angle in semi-empirical models to predict an overall frost layer's characteristics. The contact angle is the primary parameter related to surface wettability which affects frost growth. As stated earlier, under environmental conditions typical of heat pump operation, frost usually grows in three stages. Since the contact angle directly affects the droplet growth and freezing stage and affects the frost layer growth stage the least, analytical frost models need to include all three of the stages of frost growth when modeling different surface types.

In addition to mitigating frost growth by coating an entire surface with a material of specific wettability, there is also the potential to use biphilic surfaces, which combine hydrophobic and hydrophilic regions to prevent or control frost formation. Water transport can be controlled before freezing by using surfaces composed of resilient and optimally positioned interlaced patterns of water-attracting and water-repelling regions. A surface that can selectively attract and repel water vapor could change the frost growth pattern and limit the frost layer's thickness. Experimental studies exist which examined frost growth on biphilic surfaces, but analytical investigations into these surfaces are lacking.

1.2 Objectives of the Research

In order to fill some of the gaps in the existing experimental and analytical investigations of frost growth characteristics, the following research objectives are proposed:

1) Present new data of early frost growth on flat plates of varying wettability. These data are unique because they provide the average droplet size and distribution up until freezing and the subsequent frost layer's characteristics, all for a single experimental run. Most frost studies in the literature reported frost behavior for either very early frost growth or frost layer full growth and did not follow the entire process. Furthermore, an investigation will be made into differences between frost growth behaviors on surfaces with and without cleaning before the tests.
2) Present a new, semi-empirical model which predicts droplet growth, coalescence, and freezing characteristics on surfaces with different wettability types.
3) Present a new, semi-empirical model to predict the frost characteristics during the crystal growth frosting stage, which serves as a bridge between the droplet growth and frost layer growth modeling stages.
4) Present model results when the overall three-stage frost model is segmented both in the airflow direction and by wettability type.
5) Extend the model to a folded-flat fin configuration, in which air flows between two parallel plates and frost grows on both.

1.3 Methods Used to Achieve Research Objectives

For this research, frost growth under forced convection conditions and typical heat pump operating conditions was investigated both experimentally and analytically. Specific methods used to achieve the objectives listed above are as follows:

1) Data for droplet growth and subsequent frost growth were gathered and analyzed for bare aluminum, hydrophobic, and hydrophilic surfaces while environmental conditions were varied in a parametric fashion. The analysis's primary results were projected droplet size, droplet surface area coverage, droplet distribution, droplet freezing time, and thickness and density of the frost layer, which subsequently grew on top of the frozen droplets. The data presented in this dissertation were primarily for surfaces that were cleaned before each frost test. These data were also compared with data for uncleaned surfaces described in Burak Adanur's master's thesis (Adanur, 2019).
2) Classical dropwise condensation theory was combined with statistical and empirical methods to predict average droplet size under coalescence-driven environments. New correlations were developed to predict area coverage, coalescence effects, and freezing time of droplets on surfaces with different wettabilities. The model was developed and validated using data from several literature sources and data collected during this study.
3) The crystal growth model was developed by assuming that, immediately following droplet freezing, frost crystals only grew from the tops of the frozen droplets in a vertical direction. The model was based on a thermal resistance network through individual crystals, with empirical correlations describing the change in crystal aspect ratio with time. Expressions were developed to describe frost density and effective thermal conductivity during crystal growth which accounted for the frozen droplet contribution to

the overall frost layer. The crystal growth model ended when overall frost density reached a minimum value, at which point there was a transition to a model for homogeneous frost layer growth. The latter model's primary equations came from previous work by Padhmanabhan (2011), with a few alterations to suit this study's needs.

4) To segment the model in the direction of air flow, air conditions at each segment's outlet were calculated and used as inputs for the subsequent segment. Additionally, a local, rather than an average, expression for the Nusselt number was used when calculating the heat transfer coefficient. For transverse segmentation by wettability type, the pressure drop through each segment was calculated. If the pressure drop across all segments was not equal, air velocity through each segment was adjusted in such a way as to preserve the total air mass flow rate until the pressure drop values were the same.
5) The model was extended to a parallel plate configuration by treating the air flow as external flow over flat plates until the point at which the thermal boundary layers met. After this point, the flow was modeled as internal flow between parallel plates. This critical distance for transition changed as the frost layers grew since a smaller channel spacing would cause the boundary layers to meet at a shorter distance into the channel.

2. Literature Review

For the range of environmental conditions that are common for air-source heat pumps, frost generally grows in three distinct stages: condensation, growth, and freezing of supercooled liquid water droplets, growth of vertical crystals from the tops of those droplets, and growth of a homogeneous frost layer with average thermodynamic properties (Hoke et al., 2007). The last stage, frost layer growth, has been studied extensively by many researchers using experimental and analytical methods. Most of these studies explored frost growth on cold surfaces at temperatures less than -10 °C, outside the typical heat pump operation range. Additionally, studies did not often report frost characteristics within the first 10 minutes of frost growth. Some of these studies analyzed frost growth behavior on surfaces of different wettability. However, these studies generally did not explore wettability effects on all three stages of frost growth. Instead, they usually focused on how the overall homogeneous frost layer's properties changed with surface type or on droplet growth and freezing without continuing to how this stage affected more mature frost growth. The following review focuses primarily on the experimental and analytical analyses of the first two stages of frost growth, droplet growth and crystal growth, since they are most directly affected by surface wettability and have been the least studied. The discussion of frost layer growth studies is kept to those which investigated and compared frost characteristics on different wettability types.

2.1 Experimental Investigations into Droplet Growth and Freezing

Seki et al. (1985) were some of the first researchers to investigate, both experimentally and analytically, droplet growth and freezing under air-forced convection conditions on surfaces with different wettabilities. They primarily presented results for the frost layer growth stage of frosting, but they did present a few droplet size data with time at cold surface temperatures between -10 and -5 °C and contact angles between 43° and 110°. These data indicated that freezing happened more quickly at a lower surface temperature and smaller contact angle and that the projected droplet diameter was larger at a smaller contact angle.

Bryant (1995) investigated droplet growth and freezing at cold surface temperatures of -10 °C. He employed a 4-factor experimental design, varying humid air temperature between 1.2 and 7 °C, relative humidity between about 50 and 80%, air velocity between 0.3 and 0.8 m/s, and surface wettability between uncoated copper and hydrophobic types. He observed that droplet freezing time increased as air temperature and relative humidity decreased and as contact angle increased. Freezing time also increased as velocity decreased, though the velocity effects were minor. Droplet distributions were presented for different experimental test runs. These data indicated that droplet size increased as humidity increased. However, it was unclear what effects the other experimental parameters had on droplet size since few distributions were presented, and there were sometimes multiple parameters varied at a time between them.

Kim et al. (2016) studied droplet growth and freezing under similar experimental conditions as Bryant. They varied humid air temperature between 4 and 8 °C, relative humidity between 60 and 90%, air velocity between 2 and 4 m/s, and cold refrigerant temperature (similar to the cold surface temperature) between -14 and -8 °C. They also investigated many different surface types with contact angles ranging from about 75 to 160°. Their data indicated that freezing time increased as cold refrigerant and air temperature increased. In contrast, the effects of changing air velocity and humidity were much smaller and not as conclusive. They also observed that increasing contact angle led to delayed freezing. These effects were most significant within a specific range of Fourier number; outside this range, contact angle had little to no effect on freezing time. Additionally, they reported that droplet radius generally decreased while droplet height and number generally increased with increasing contact angle.

Sheng et al. (2020) reported droplet size and area coverage data with time and freezing time for frost nucleation tests where they varied cold surface temperature between -7 and 2 °C and surface contact angle between 15 and 155°. Humid air temperature, relative humidity, and air velocity were held constant at 13 °C, 58%, and 0.4 m/s, respectively. They reported larger droplet sizes and area coverage ratios for smaller contact angles and lower surface temperatures for a given time into a test. However, this was not necessarily the case if these values were compared at their respective freezing points because they also observed that freezing time increased significantly as contact angle and cold surface temperature increased, leading to longer growth periods and larger droplets and coverage ratios. Additionally, they presented droplet images which showed that hydrophobic and superhydrophobic surfaces had round droplets, while droplets on the baseline and hydrophilic surfaces had more spread out and irregular shapes.

Other researchers focused on lower cold surface temperatures in their forced convection frosting tests. Tao et al. (1993) used surface temperatures between -8 and -26 °C to investigate droplet growth and freezing on a bare aluminum surface. In addition to testing at many different surface temperatures, they also tested three humidity levels and two air velocities. Their data indicated that droplet freezing time increased as surface temperature increased and relative humidity decreased. Additionally, droplet size at freezing increased as both surface temperature and relative humidity increased. Hoke tested different surface types with contact angles ranging from 10 to 106° at surface temperatures between -5 and -27 °C. They observed stratification in droplet freezing times at low surface temperatures: freezing time increased as contact angle increased for a given absolute humidity difference. However, at surface temperatures above -16 °C, freezing was more stochastic, and freezing time trends were much vaguer.

More research groups focused on droplet growth and freezing under natural convection conditions. For example, Wu et al. (2007a), Li et al. (2010), and Chen et al. (2015) investigated droplet growth and freezing under air natural convection conditions on bare metal surfaces ranging in temperature from -6 to -20 °C. He et al. (2011) investigated droplet formation on superhydrophobic surfaces (θ = 150° - 171°) at temperatures from -5 to -10 °C. Petit and Bonaccurso (2014) measured droplets on five surfaces with different stiffness ranging in contact angles from 72° to 124°. They studied ultra-low surface temperatures, which ranged from -10 to -40 °C. Haque et al. (2020) measured droplet growth on hydrophobic and hydrophilic nanopillared surfaces at surface temperatures of -8 °C, and Wang and Cheng (2019) investigated droplet growth and freezing on hydrophobic and uncoated brass surfaces at surface temperatures ranging from -7.4 °C to -21.3 °C. Additionally, Chu et al. (2017) investigated droplet growth characteristics for surfaces with contact angles ranging from 85° to 160° at surface temperatures between 2 and 8 °C, slightly above those required for frosting. Several studies (such as Shen et al. (2019), Chu et al. (2017), Boreyko and Collier (2013), and Hao et al. (2014)) also investigated droplet growth and freezing on superhydrophobic surfaces specifically. Many of these papers' primary focus was on droplet motion behaviors due to surface characteristics, such as jumping, sweeping, and rotating, and how they affected droplet growth or defrosting.

This review of experimental investigations into droplet growth and freezing indicated that published data for high subzero surface temperatures and low surface contact angles under air forced convection frosting conditions were few. The research performed by Sheng et al.

(2020) included high surface temperatures and low contact angles, yet droplet growth data for subzero temperatures were primarily presented for the superhydrophobic surface. There also seemed to be apparent randomness of freezing times reported in the literature for similar surfaces and comparable test conditions in well-controlled laboratory settings. For example, data from papers such as Kim et al. (2016) and Tao et al. (1993) indicated that freezing time generally decreased as temperature difference and supersaturation degree between the humid air flow and the test surface increased. These trends did not necessarily hold when data at similar test conditions from various studies were compared and did not even hold for all data points from a specific study. Additionally, Hoke et al. (2000) reported an increase in the stochastic nature of droplet freezing on different surface types at higher surface temperatures, and Kim et al. (2016) observed that freezing delay due to increased contact angle was only significant for a specific range of Fourier number. While droplet average diameters were often reported in the literature, droplet distributions were seldom presented. Droplet growth processes, specifically regarding coalescence effects, were likewise rarely reported for temperatures below the water's freezing point.

2.2 Experimental Investigations into Crystal Growth

One of the first studies investigating the crystal growth stage of frosting was presented in Hayashi et al. (1977). They divided frost growth into three stages: crystal growth, frost layer growth, and frost layer full growth. They observed that frost density decreased during the crystal growth stage when frost thickness grew rapidly and increased during the remainder of the frost growth. They stated that the transition point between crystal growth and frost layer growth occurred when density reached a minimum value. For their experiments, the dip in density was higher at lower surface temperatures and disappeared when the surface temperature reached -5 °C. They also observed longer crystal growth stages at higher surface temperatures. Additionally, they identified four primary crystal types and mapped them on a $\Delta\omega$-T_s plot. As $\Delta\omega$ decreased and as T_s increased, crystals moved from needles to plates, and the frost layer density increased.

Tokura et al. (1983) observed that frost growth began with the freezing of supercooled liquid droplets, which become the initial nuclei for the frost layer, rather than beginning with crystal deposition. They split the subsequent frost growth into two stages: the "one-dimensional growth period" and the "three-dimensional growth period". During the "one-dimensional growth period", sublimation primarily occurred on the frozen droplets, creating crystals that grew in a direction normal to the surface (analogous to the crystal growth period described in Hayashi et al.). During the "three-dimensional growth period", the frost layer grew more slowly and in directions both normal and parallel to the cold surface. They observed very high frost density at the beginning of frost growth due to the frost nuclei consisting of liquid droplets which froze to solid ice. Frost density decreased rapidly during early frost growth but reached a minimum and increased as the frost layer matured. Higher initial densities and more rapid decreases were observed for high mass transfer cases, in which the liquid droplets grew larger before freezing into ice nuclei.

Wu et al. (2007a, b) investigated early frost growth on both bare copper and hydrophobic coated surfaces. They described four crystal types and mapped them on a $\Delta\omega$-T_s plot as Hayashi et al. (1977) did. They found that transitions between crystal types were primarily dependent on surface temperature and only secondarily dependent on humidity difference. At -10 °C, they observed differences in both droplet growth and crystal growth on surfaces with different wettability. On the hydrophobic surface (θ = 100°), droplets grew for a long time and became large before they froze. After freezing, a small number of crystals that had many branches grew from the droplets. On the bare surface (θ = 56°), droplets froze very quickly and were thus very small when they froze. Interestingly, crystal shape and number were different on this surface. Before the droplets on the hydrophobic surface had frozen, the bare surface was covered with many tall, slender crystals with few branches.

Lee et al. (2004) mapped crystal type for bare aluminum (θ = 88°) and hydrophilic (θ = 23°) surfaces. They observed the same three crystal types on both surfaces, but that the transitions between crystal types occurred at higher humidity levels for the hydrophilic surface. Chen et al. (2015) indicated that changing atmospheric pressure changed crystal structure, and Wang et al. (2004) observed that frost crystals were sparser and more fragile under the presence of an electric field. From the observations described above, there was agreement that frost density decreased during the crystal growth stage of frosting and then increased during subsequent frost layer growth. It was also clear that many factors affected crystal shape and structure, but the contact angle effects were not conclusive.

2.3 Impacts of Water Purity on Droplet Freezing

Many researchers studied homogeneous freezing of pure water and observed that droplets could be cooled to temperatures around -40 °C before freezing initiated. Since then, many other studies were performed to investigate how impurities in water affected its freezing characteristics. Hoffer (1961) observed that adding soluble salts such as those naturally existing in the atmosphere to pure water decreased the temperature required to initiate homogeneous ice nucleation and that the decrease was more drastic at higher solute concentrations. On the contrary, adding insoluble particles to the water increased its freezing temperature. Pruppacher (1963) observed results in agreement with Hoffer's. He tested many more solutions and observed decreases in freezing temperature, which increased with increasing concentration, for almost every case. Wilson and Haymet (2009) obtained similar results for cases of heterogenous nucleation. They observed freezing of sodium chloride and glucose solutions in a glass tube, introducing a grain of fine sand into the solutions as the catalyst for heterogeneous nucleation. The freezing temperature decreased with increasing solute concentration, just as for the homogeneous nucleation studies. Singha et al. (2009) performed a similar investigation for sessile droplets on a cold surface. They observed increasing freezing delays and decreasing freezing front velocities with increasing salinity. Additionally, the pointed tip at the tops of the droplets, often observed during freezing of sessile water droplets, decreased in height and nearly vanished as the solute concentration increased. This change in droplet shape was also observed by Boulogne and Salonen (2020). They noticed that this tip's angle was more sensitive to solute concentration at very low concentrations than were conductivity or surface tension. They proposed using measurements of tip angle as a quick way to test water purity.

These studies indicate that water purity can significantly affect water freezing characteristics, especially at high concentrations. The water vapor that condensed on the surfaces in the current study's experiments came from the surrounding atmospheric air primarily, and purification was not practical. However, Type I deionized water was used to humidify the air and a MERV 9 air filter was installed before the test section to avoid increasing the solute and particulate concentrations in the water vapor.

2.4 Droplet Growth Modeling

The investigation in Seki et al. (1985) was one of the few studies that developed a multi-stage frost model that began with the droplet condensation stage; the model then built a homogeneous frost layer on top of the frozen droplets. The droplet model presented in their paper accounted for the contact angle of the surface. However, because the authors reduced the frozen droplets into equivalent layers of ice and porous frost, it was not clear to what extent the effects from the surface contact angle on frost structure were accounted for by the model. For example, the effects of contact angle on the frost growth's crystal growth were not discussed. The Seki et al. model was validated against their study's data, with a maximum droplet diameter of about 0.6 mm.

2.4.1 Classical Nucleation Theory

The classical dropwise condensation theory can explain the physics of droplet growth. This theory is well-established, but researchers are still refining and modifying it today to apply it to a broader range of applications. Graham (1969) pointed out that heat transfer through a single hemispherical droplet could be calculated by adding all temperature differences due to the thermal resistances between the vapor and the cold surface. For dropwise condensation out of pure vapor, Graham identified the significant resistances to be due to droplet curvature, interfacial mass transfer, and conduction through the droplet. Droplet growth was then calculated by assuming all the heat transfer through the droplet was latent heat, leading to increased droplet mass and volume. Other studies (Glicksman and Hunt Jr., 1972; Wen and Jer, 1976; Burnside and Hadi, 1999) also utilized this temperature difference method with the same three resistances to calculate the heat transfer through hemispherical droplets. Abu-Orabi (1998) followed a similar method but included a resistance due to a promoter layer's thickness and conductivity on the condensing surface. Vemuri and Kim (2006) included only the conduction and interfacial resistances.

More recently, other researchers adapted some of these resistances for droplets that are not hemispherical but have contact angles other than 90°. For example, Kim and Kim (2011) developed an expression dependent on contact angle for the conduction resistance through a droplet. They did this by integrating over isothermal surfaces within the droplet from the cold test surface to the droplet/vapor interface. They also made minor adjustments to the interfacial and promoter layer resistance expressions but made no changes to the curvature resistance. Several other studies (Sikarwar et al., 2011; Miljkovic et al., 2012; Rykaczewski, 2012) also performed analyses while using the same expressions for resistances as Kim and Kim (2011). The models utilizing this classical dropwise condensation theory typically tracked individual droplets' growth, then included coalescence by merging two droplets when they grew large enough to touch.

2.4.2 Power Law Formulation

Researchers in the field of statistical physics have approached developing expressions for droplet growth in another way. Beysens and Knobler (1986) investigated the growth of breath figures from a flow of nitrogen gas nearly saturated with water vapor on a surface with a contact angle of 90°. They observed that droplet growth developed with three regimes: the growth of individual droplets with no inter-droplet interactions, growth of original droplets by direct condensation and coalescence, and continued growth of original droplets but with secondary nucleation in the free spaces between droplets. They noticed that the growth of individual, isolated droplets could be described by the power-law $R \propto t^{\mu}$ where $\mu \approx 0.23$. They applied this power-law form to the growth of average-sized droplets in a coalescence-driven environment as well, but for this case, μ increased to 0.75. Fritter et al. (1988) used simulations of droplet condensation to confirm that μ for coalescing droplets was three times greater than μ for individual droplet growth, and Viovy et al. (1988) showed analytically that μ = 1/3 and 1 for individual and average droplet growth, respectively, for hemispherical droplets. Later, Zhao and Beysens (1995) studied condensation on a surface with a wettability gradient ranging from < 10° up to 90°. They observed that μ was still equal to 1/3 and 1 for individual and average droplet growth for all contact angles, though coalescence effects dominated earlier for hydrophilic surfaces. Grooten and van der Geld (2011) measured droplet growth on a vertical flat surface exposed to humid air flow. They determined that μ ranged from 0.85 to 1.64 as the air stream's absolute humidity increased from 0.05 to 0.14 kg/kg. These observations led to the conclusion that condensation, rather than coalescence, was dominant during their experiments. These large values of μ, which deviated from other published results, could have occurred because, unlike the previous studies where droplets did not grow to diameters larger than about 0.6 mm, droplets in Grooten and van der Geld's experiments had large terminal diameters between 1 and 4 mm. Very different results were observed by Castillo et al. (2015), who reported no change in the asymptotic value of μ for the growth of small droplets (D < 0.4 mm) under differing humidity conditions. Additionally, Boreyko and Collier (2013) demonstrated that the power-law relationship still held under condensation frosting conditions. They observed condensation on flat horizontal plates at surface temperatures of -10 °C and -20 °C and calculated μ to be between 0.5 and 0.71 for both the hydrophobic and superhydrophobic surfaces they tested (until droplets grew large enough to jump off the superhydrophobic surface).

2.4.3 Freezing Time

The point at which subfreezing liquid droplets turn into ice beads is an important quantity to predict. Understanding the factors that contribute to freezing delay can aid in system designs for longer working periods due to delaying the need for defrosting. In multi-stage frosting models, the freezing time determines the transition from the droplet growth phase to the crystal growth phase of frost growth. The freezing time was often measured and reported during condensation frosting studies, but it has seldom been modeled.

Bryant (1995) developed a correlation, presented below as Equation 2.1, for the freezing time using multiple linear regression of all the data gathered in his study. In Equation 2.1, A is the ratio of the surface contact angle to a contact angle of 90°, W is the humidity ratio difference between the bulk air and the surface, T_g is the temperature of the bulk air, and Nu is the Nusselt number calculated using a correlation developed for the data gathered in his study.

$$t_f = 41.3\, A^{2.6}\, Nu^{-1.4}\, W^{-0.6}\, T_g{}^{-0.25} \tag{2.1}$$

Kim et al. (2016) presented a correlation for a Fourier number associated with the freezing time. They used the results to identify conditions for which there was a strong frost retardation effect as contact angle increased. The optimum region lay between Fo of about 2.2 and 5; if Fo was above or below this band, contact angle effects on freezing time were weakened. The correlation is presented below in Equation 2.2, where $T^* = (T_{air} - T_{tp}) / (T_{air} - T_{cold})$. Similarly to Bryant’s case, this correlation was developed using only data from the authors’ study. This correlation helps understand the factors affecting droplet freezing time. However, it would be difficult to use in a droplet growth and freezing model because the droplet height at freezing is required to calculate an absolute freezing time.

$$Fo = \frac{\alpha \, t_f}{{y_{f,avg}}^2} = (4.29 \times 10^3)\, {Re_L}^{0.231}\, \omega^{-0.685}\, (T^*)^{1.644}\, (rad)^{0.232} \qquad (2.2)$$

Seki et al. (1985) took a different approach to predict the point of droplet freezing. Instead of specifying a freezing time, they developed a correlation using their experimental data to determine the water droplets' freezing temperature. This correlation is presented as Equation 2.3, where T_{cri} is the freezing temperature and V is the single droplet volume in cm^3.

$$\log_{10}|T_{cri}| = 0.1440 \ \log_{10} V + 0.3359 \qquad (2.3)$$

All three of the expressions above were developed using only data for their respective studies, so they were applicable only for the ranges of test conditions and surface types reported in those particular studies. All data reported in these studies had cold surface temperatures at or less than -5 °C, so their applicability to temperatures in the upper range of typical heat pump operating conditions is unknown.

2.5 Crystal Growth Modeling

Tao et al. (1993b) modeled the frost layer during the crystal growth period to be a forest of growing circular fins. It focused on the growth of individual ice columns rather than on the growth of a homogenous layer. It did not include frozen droplets, though the authors mentioned that the crystals grew at the liquid nucleation sites. Instead, this model required initial thickness, diameter, temperature, and crystal distribution to begin calculations. This model assumed different temperatures for the frost crystals and the neighboring air voids, with the average temperature at the frost surface falling somewhere in between. This model was capable of transitioning to a frost layer growth model (also developed and presented in the same paper) but needed the transition time to be a model input. The overall crystal growth/frost layer growth model was compared to experimental data with surface temperatures of -10 and -20 °C, with promising results.

The second model, developed by Sahin (1995), depicted the frost layer as identical cylindrical frost columns, which grew straight up from the cold surface. Unlike Tao's model, this model treated frost during crystal growth more as a homogeneous layer since heat and mass transfer were based on an average frost temperature. It was assumed that the frost columns and air voids were at the same temperature for a given point in the frost layer. As in frost layer growth models, diffusion mass flux led to an increase in column radius (leading to densification), and the remaining mass flux increased column height. The only empirical constant used was a volumetric ratio of crystals in the overall frost layer. No mention was made of the dimensions of the corresponding columns. The model was validated using the author's measured frost thickness data between 10 minutes and 100 minutes of testing. The most significant deviations occurred at very early and very late stages of frosting.

These two models were the only existing comprehensive models found in the current investigation that described the crystal growth frosting stage in particular. Shneider (1977) used a needle structure to describe the frost layer, and Cheikh and Jacobi (2014) used a hexagonal crystal structure to obtain a mass transfer conductance term for their overall model. However, both of these models' primary focus was to predict an overall homogeneous frost layer's behavior. These models were intended to be used as stand-alone models with no need to transition to any other model format. Additionally, Ismail et al. (1997) and Tahavvor and Yaghoubi (2009) included the crystal growth stage in their model for frost growth on cylindrical tubes. They utilized Tao's modeling format rather than developing a new method to describe crystal growth. There is still lacking a crystal growth model in which frost crystals deposit on top of frozen droplets and which can serve as a bridge between the droplet growth and frost layer growth stages of frosting.

2.6 Investigations into Wettability Effects on Frost Layer Growth

Some researchers investigated wettability effects on frost growth behavior but focused on how surface type affected the overall homogeneous layer. Shin et al. (2003) studied frost growth and densification on three aluminum surfaces treated to obtain contact angles of 23°, 55°, and 88°. They observed that frost thickness was low and density was high for surfaces with lower contact angles, while frost thickness was high and density was low for surfaces with high contact angles. Additionally, the surface characteristics had significant effects on frost layer properties during early frost growth, while environmental effects dominated after two hours of frosting. They developed correlations for frost thickness, density, and thermal conductivity that were dependent on the dynamic contact angle. The correlation they presented for frost mass was dependent only on time. In contrast to Shin et al.'s results, Hoke et al. (2000) reported larger frost thickness on a hydrophilic glass surface than on a hydrophobic PTFE surface. Kim and Lee (2011) studied frosting and defrosting behavior on hydrophilic ($\theta = 2.5°$), bare aluminum ($\theta = 75°$), and hydrophobic ($\theta = 142°$) fin surfaces. They observed very similar frost thicknesses on all three surfaces up to two hours of frosting, with that on the hydrophilic surface being slightly smaller. There was more stratification when considering frost density between one and two hours of frosting. The hydrophilic surface had the densest frost layer, and the hydrophobic surface had the least dense frost layer. The frost surface temperature was nearly the same for all three surface types. Rahimi et al. (2015) studied frost growth on a bare aluminum surface ($\theta = 78.1°$), a hydrophilic surface (θ = 36.9°), and two hydrophobic surfaces (θ = 116.1°, 123.9°). They observed that the hydrophilic surface had a thicker frost layer and lower density than both hydrophobic surfaces. In contrast, the bare aluminum surface had the thickest frost layer and lowest frost density of all the surface types. Hermes et al. (2018) investigated frost growth under natural convection conditions on three surface types with contact angles ranging from 45.3° to 158.9°. They presented a correlation for frost thickness, dependent on the contact angle, though this dependence was relatively small. The contact angle had a more significant impact at small frost thicknesses. Hermes et al. (2019) later studied frost growth under forced convection conditions on four surface types with contact angles ranging from 60° to 123°. In contrast to their natural convection results, they observed such a weak dependence of frost thickness on contact angle that contact angle dependence was removed from the correlation they developed for forced convection cases. The same research group (Sommers et al., 2017) observed that, under natural convection conditions, frost density on hydrophobic and hydrophilic conditions was not well predicted by a previously developed correlation from Hermes et al. (2014) that did not include the contact angle of the surface. They developed a new correlation, which included the effects of surface wettability, which described frost density much better.

Based on the above discussion, the effects of surface wettability on a frost layer's thickness and density were inconclusive and often directly contradictory between various studies. It cannot even be said that neutral contact angles produced frost behavior between those for the hydrophilic and hydrophobic extremes since Rahimi et al. (2015) observed the thickest frost and lowest density on a bare aluminum surface with a contact angle that fell between those of the other surfaces studied. Therefore, there is still a need for additional investigation into frost growth behavior on surfaces with different wettability. In these investigations, the entire three-stage frost growth process should be considered to more thoroughly asses the effects of surface wettability on all parts of the frost layer.

2.7 Frost Growth between Parallel Plates

Östin and Andersson (1991) studied frost growth in a long, aluminum parallel-plate channel under varied environmental conditions. Though frost grew on both the top and bottom surfaces, only single average values of frost thickness, density, mass, and thermal conductivity were provided for the channel. They observed that changing the cold plate temperature mainly affected frost thickness. A change in air velocity affected the rate of mass deposition primarily. Changes in the air relative humidity affected both the frost thickness and the mass deposition rate.

Ismail and Salinas (1999) coupled a crystal growth model based on Tao et al. (1993b) to a frost layer growth model to predict frosting behavior on flat and parallel plates. When modeling parallel plates, they assumed that frost behavior was the same on both plates. They used the same model for parallel plates as for a single flat plate, except that they increased air velocity as the flow was constricted due to frost growth on both the top and the bottom of the channel. Their model indicated that frost thickness was very similar for a flat plate and parallel plate spacing ranging from 20 mm to 40 mm. Density was similar on a flat plate and for a spacing of 40 mm, but it increased as plate spacing decreased.

Lüer and Beer (2000) performed an experimental and analytical investigation into frost growth in laminar convective flow on long aluminum parallel plates. They presented results for both average thickness and density of the frost layer and thickness at various locations along the channel in the direction of air flow. They observed that increasing the Reynolds number increased frost density while decreasing the cold surface temperature increased thickness and decreased porosity. Increasing air humidity increased both thickness and density of the frost layer, though, at low humidity values, frost density decreased for a time before it began to increase. For the study's analytical portion, air flow between the plates was modeled as internal laminar flow with constant properties along the channel's length. They assumed negligible gravitational effects, so the frost thickness and density were identical on the top and bottom plates for a given point along the channel. Model results were mostly satisfactory, though the model predicted a shallower slope of frost thickness change along the channel than was observed in the experiments.

Nascimento et al. (2015) studied frost growth and densification in a parallel plate channel under laminar forced convection conditions. They investigated the effects of air humidity, cold surface temperature, and channel spacing on frost thickness and density. They observed that while air humidity and cold surface temperature affected both thickness and density, channel spacing primarily affected density. They also performed a theoretical scaling analysis applicable to the period after the initial crystal growth period, showing that both frost thickness and density followed a $t^{1/2}$ scale. No specific information was provided on whether the frost layers had the same thickness and density values on both the top and bottom plates. However, frost layers looked similar in the photographs provided, and they only provided single values of frost thickness and density at each set of test conditions.

2.8 Summary of Primary Observations from the Literature Review

Observations of droplet growth and freezing data available in the literature revealed that very few data were offered for superhydrophilic surfaces or relatively high surface temperatures typical of heat pump operation. Additionally, no conclusive results were observed on how environmental conditions and surface characteristics controlled droplet freezing time. Coalescence effects on droplet growth were also rarely reported. Few papers carried their investigations past droplet growth and freezing to include crystal growth and frost layer growth data. Of those that did, data for different frosting stages were discussed separately and not compiled for all stages of a single experimental run. These observations led to the adoption of this research's first objective (as stated in Chapter 1), which was to present new data of early frost growth on plates of varying wettability. These data would fill some of the gaps in the literature by providing measurements for all three frosting stages during a single experimental run. More details would also be provided for how surface wettability and environmental conditions affected droplet characteristics such as distribution, coalescence effects, and freezing time. Implications of starting a test with a clean or dirty surface would also be investigated.

Existing methods for predicting droplet growth and coalescence behavior consisted primarily of using classical dropwise condensation theory to predict individual droplets' growth and coalescence at surface temperatures above freezing in a pure vapor field. These methods could be costly in terms of time, especially for large surfaces with many droplets, since the simulations must keep track of each droplet’s size and location at all times. Another method of estimating droplet size was to use a power-law method. However, exponents derived from experimental data often differed from traditional values, and environmental and surface effects on the exponent were inconclusive. Neither of these methods was often applied to droplets at subfreezing temperatures and did not take droplet freezing into account. A few separate studies developed correlations to predict the point of droplet freezing. However, the correlations were developed using only the authors’ data and did not apply to the high surface temperatures investigated during the current study. The second objective for this research was proposed to address some of these issues. A new model to predict droplet growth and freezing characteristics would be developed. This model would combine elements of classical dropwise condensation theory and the power-law method. It would be semi-empirical and predict average droplet characteristics on different wettability types to reduce time and computational costs and be useful for real-world design applications. Further investigation into the factors that affect droplet coalescence and freezing time would be performed during model development so the model could be broadly applicable to typical heat pump operating conditions instead of a single data set.

A review of the literature regarding modeling the crystal growth stage of frosting revealed no model that modeled crystals growing from the tops of frozen droplets. Instead, crystals grew straight up from the cold surface with assumed or empirical distributions. Since, under high-temperature conditions typical of heat pump operation, droplets grew large before freezing and could not be neglected when considering the overall frost layer, a crystal growth model was still needed to describe the frost layer more comprehensively. This research's third objective would fill this gap by providing a method to predict frost characteristics during the crystal growth stage of frosting when crystals were assumed to grow only from the frozen droplets. Experiments indicated that crystals grew in various shapes that were dependent on temperature and the humidity ratio gradient. However, several shapes were columnar (such as needles, solid hexagons, hollow prisms, etc.). Because of these observations and precedent set by previous models, the crystals in the current study’s model were assumed to be solid cylinders.

This research's fourth and fifth objectives primarily focused on extending the overall model's applicability to configurations more complex than a simple, flat plate. The specific configurations considered were segmentation in the airflow direction, biphilic surfaces, and parallel plates. While models already existed for some of these configurations, they were generally restricted to modeling only the frost layer growth stage. The current study aims to enhance the understanding of early frost growth on various surface configurations under test conditions where the droplet growth and crystal growth stages of frosting are not negligible by applying the overall model developed here to these complex geometries.

3. Experimental Work

The experimental portion of the research presented here was a continuation of the previous work reported by Cremaschi et al. (2019) and Adanur et al. (2019). The experimental facility, including all instrumentation, was described in great detail in Adanur's master's thesis (Adanur, 2019), but a summary is provided here for clarity. The majority of the experimental results presented in this section were also published in the *Applied Thermal Engineering* journal (Harges et al., 2020).

3.1 Summary of Experimental Facility

The test facility consisted of a closed airflow wind tunnel, shown in Figure 3.1, which controlled the air temperature, humidity, and velocity.

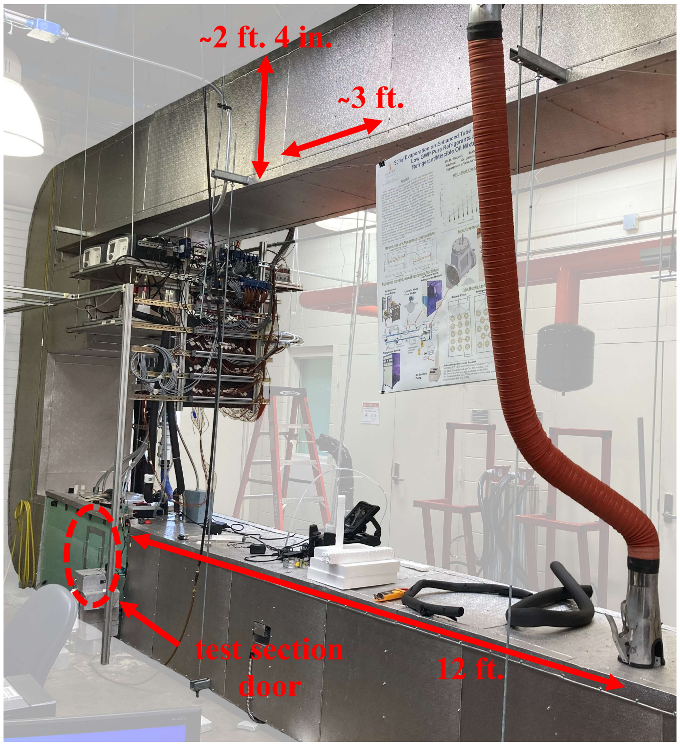
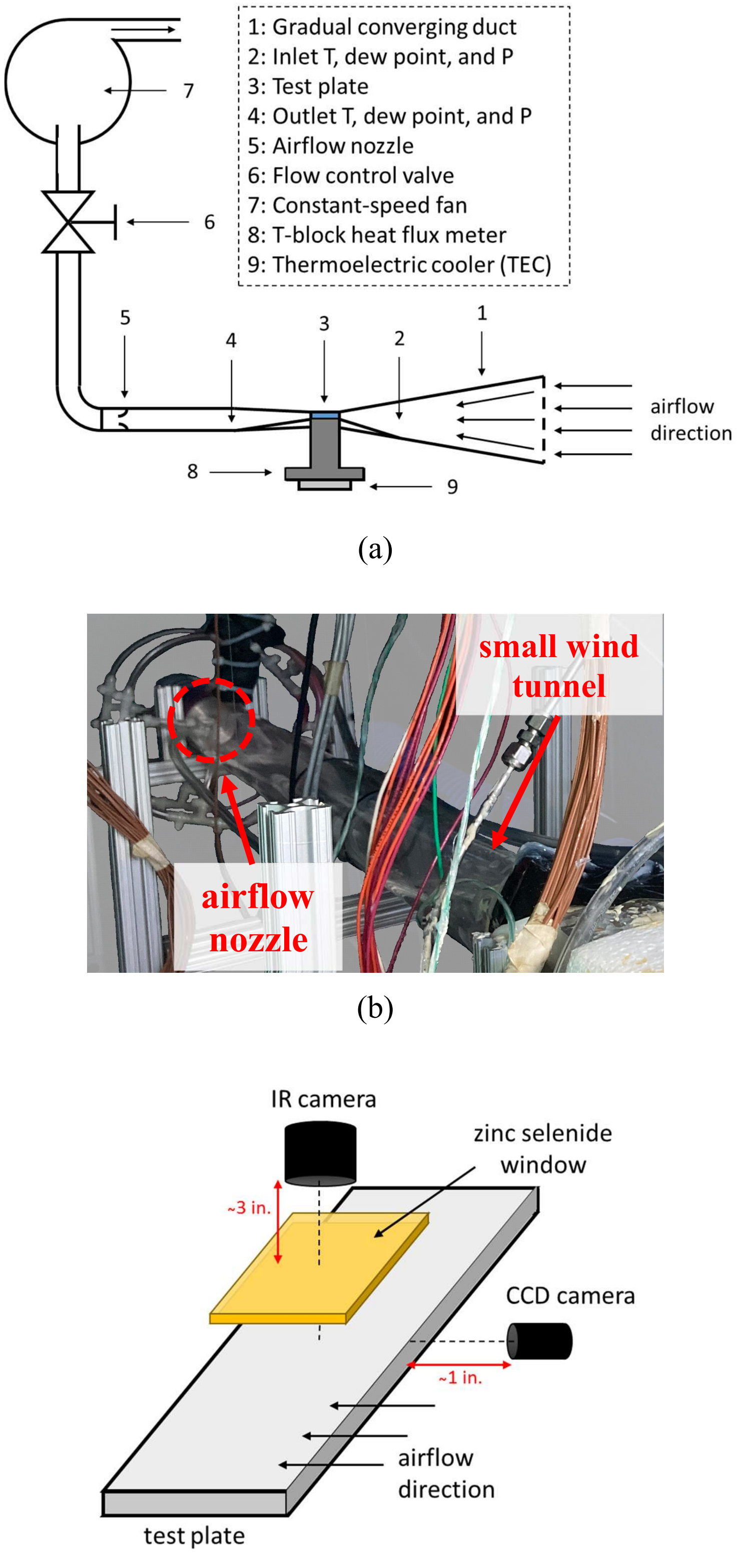


Figure 3.1: Photographic image of the closed airflow wind tunnel.

A second smaller airflow wind tunnel, whose schematic is shown in Figure 3.2(a), was installed inside the large wind tunnel and accommodated the cold flat test plates. Figure 3.2(b) is a photograph of a section of the small wind tunnel at the test section's outlet, including the airflow nozzle. The 12-ft. section indicated in Figure 3.1 is an estimate of the portion of the small wind tunnel that was used to develop the airflow before the test section inlet. An air filter of MERV 9 was installed before the test section to filter the incoming air. A steam humidifier was used to increase the humidity of the air for certain test conditions. Two thermoelectric coolers (TECs) and an in-house built stainless steel “T-block” heat flux meter controlled the test plates' surface temperature during frosting. Ethylene glycol flowed through the bottom of this T-block assembly to act as a heat sink. Smoothly converging and diverging duct sections were installed at the inlet and outlet of the test plate, keeping the angles low enough to reasonably assume that the air flow over the plate was initially well mixed and fully developed. T-type thermocouples, chilled mirror dew point meters, flow nozzles, and differential pressure transducers were used in the test rig. Twenty thermocouples were also embedded in the metal block underneath the test plate to measure heat flux and derive the plate's upper surface temperature.

A nitrogen displacement technique was used to limit the surface temperature variation during the test's first few minutes. Nitrogen gas was metered through the test section before the test started. The dew point temperature reduced significantly, and the thermoelectric cooling modules were energized without producing any water vapor condensation. When the test plate temperature was reached, the nitrogen gas flow was stopped, and the air was circulated on the top surface of the test plate. The operator adjusted a valve with 1% tolerance of the desired flow rate to maintain a constant flow rate.



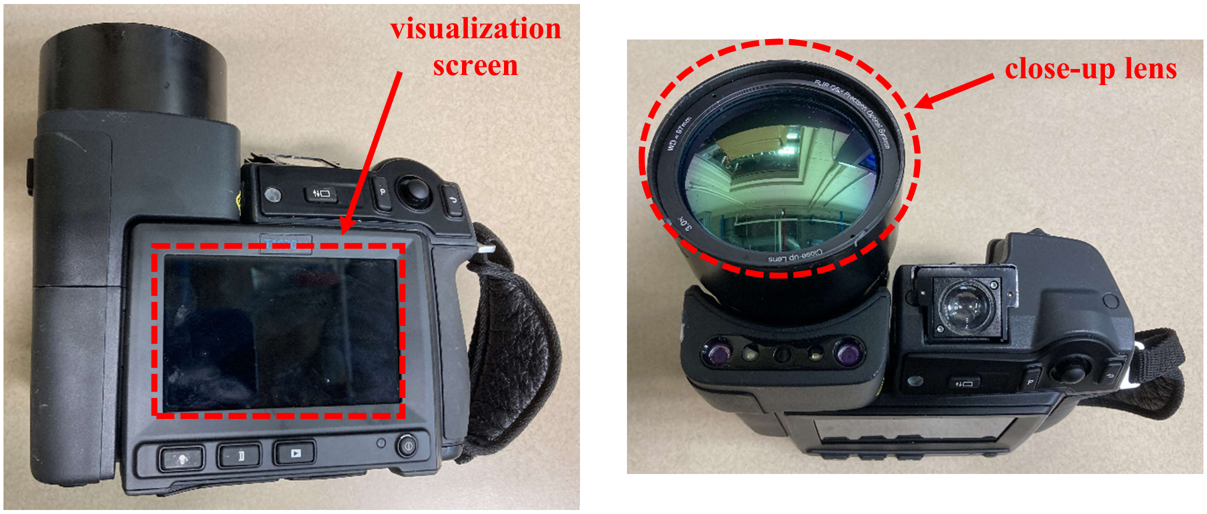
(a)

(b)

(c)

Figure 3.2: (a) Schematic of the main test apparatus. (b) Image of the outlet section and airflow nozzle. (c) Illustration of the CCD camera and IR camera in relation to the test plate.

An infrared (IR) camera with a magnifying zoom lens was used to view the droplets through a zinc selenide window (Figure 3.2(c)). It captured images every 2 s that were used to measure droplet size and surface area coverage. The minimum diameter of the smallest droplets that could be individually distinguished from each other was around 90-100 microns. An IR camera was used rather than an optical camera with a better resolution because frost surface temperature was measured during the test runs after a porous layer of frost was formed. An emissivity value of 0.9 was used for the IR camera when measuring the frost temperature. This value was chosen because it was near traditional values for snow, and it produced temperature measurements close to those of the test plate immediately after droplet freezing when the temperature at the top of the frost layer was only slightly higher than that of the test plate. Because of the difference in emissivity between the test plate and the frost layer surfaces, the frost temperature measured with the IR camera was considered when a first thin layer of frost covered the test plate. This first thin frost layer was achieved shortly after droplet freezing. The IR camera also helped identify droplet freezing because it captured the temperature increase in the individual droplets due to the release of latent heat. Photographic images of the IR camera used in the current study are provided in Figure 3.3. The camera was a FLIR T1030sc 28° model with serial number 72501374. A 51 µm close-up lens was used with the camera to improve droplet resolution.



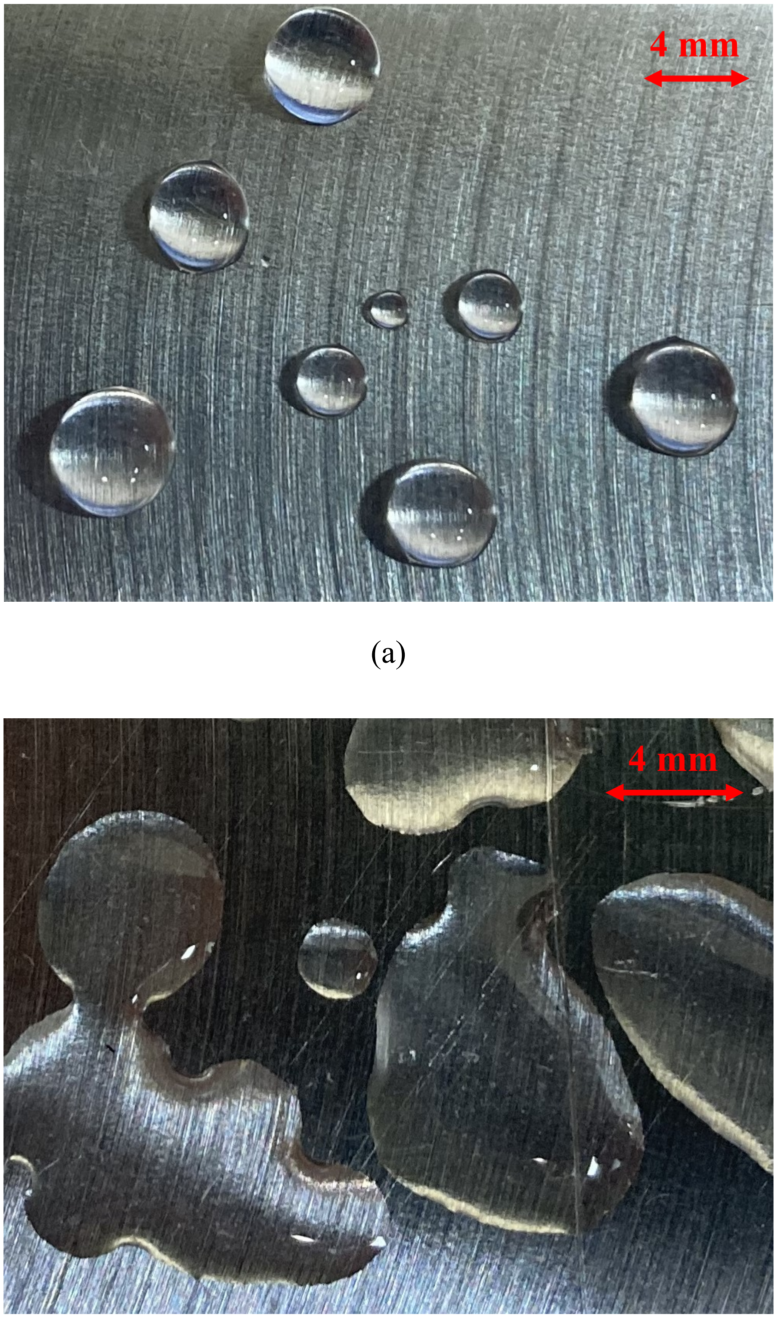
(a)

(b)

Figure 3.3: Photographic image of the IR camera used in the current study. (a) Visualization screen, (b) close-up lens.

A High Resolution (HR) CCD camera with a fiber-optic scope was used to view and measure the frost thickness at the test plate’s leading edge (Figure 3.2(c)). Real-time videos of frost growth were taken with this camera and were analyzed using multimedia processing software to create a series of jpeg images every 10 s. A Matlab function flattened the images using a checkerboard calibration pattern. Images were calibrated, and the frost profiles were hand-traced on the flattened images. Matlab calculated the number of pixels in each column of the frost profiles and generated the instantaneous frost height measurements. Frost profiles were drawn three times for each image, and averages were taken for the final frost height. Averages were taken of multiple profiles to reduce operator repeatability error since the frost layers' edges were often fuzzy and difficult to distinguish precisely.

The test plates were rectangular bars of about 6 mm (0.25 in.) in thickness. One side of each plate had a mirror finish; this side faced down and was attached to the T-block assembly by thermal paste to minimize contact resistance. The other side of each plate was a cold-rolled mill finish. The mill finish side was exposed to the airflow during the frosting tests and was treated to alter the surface wettability. The first plate had an uncoated aluminum surface and a contact angle, θ, of about 75°. The second plate was coated with a hydrophobic Teflon solution to produce a surface with a contact angle of about $\theta \approx$ 110-116° as described in Haque and Betz (2018). The third had a hydrophilic surface (θ ≈ 19-29°) formed by exposing an aluminum surface to ozone and UV light to promote its native oxide layer's growth. Area roughness (Sa) for the bare aluminum test plate surface was estimated to be in the range of 0.3 to 0.4 μm based on mill-finished aluminum sheet characteristics (Sanchez and Hartfield-Wunsch, 2011). This mill finish substrate was the one on which frost grew for the bare aluminum plate sample and which was treated to change wettability for the hydrophobic and hydrophilic samples. Photographs of macroscopic droplets on the hydrophobic and hydrophilic wettability types are provided in Figure 3.4 to illustrate the differences between the surface types. The test plates were exposed to convective airflow frosting conditions on their top surfaces with airflow cross-section dimensions of 4 mm high (perpendicular to the surface) and 152 mm wide (the test section's width). The airflow rate was constant for the entire test, that is, during both phases of frost nucleation and subsequent frost growth.



(a)

(b)

Figure 3.4: Images of droplets on the (a) hydrophobic and (b) hydrophilic wettability types

Table 3.1: Measurement devices, set points, ranges, accuracies, and control tolerances

Parameter Measured	**Measuring Device**	**Calibration**	**Set Point / Range**	**Accuracy**	**Control Tolerance**
Sensors for Controlled Variables					
Air Temp. (dry bulb)	thermocouple (grid)	in situ*	5 °C (41 °F)	±0.056 °C (±0.1 °F)	±0.28 °C (±0.5 °F)
Air Temp. (dew point)	chilled mirror dew point meter	manufacturer	0.56 °C (33 °F)	±0.28 °C (±0.5 °F)	±0.28 °C (±0.5 °F)
Plate Temperature	thermocouple (grid)	in situ*	-15 °C (5 °F)	±0.04 °C (±0.07 °F)	±0.28 °C (±0.5 °F)
Air Volume Flow Rate	flow nozzle	manufacturer	8.5 m^3/h (5 cfm)	±0.05 m^3/h (±0.03 cfm)	±0.09 m^3/h (±0.05 cfm)
Measured Variables					
Air Pressure Drop	pressure transducer	manufacturer	0 to 250 Pa (0 to 1” H_20)	0.25% full scale	(-)
Frost Surface Temperature	infrared camera	in-situ	-22 °C to 5 °C (-8 °F to 41 °F)	±2 °C (±3.6 °F)	±1.6 °C (±3.5 °F)
Frost Mass	high precision digital scale	manufacturer	0 to 5 g (0.011 lbm)	±0.1 mg (±0.0015 gr)	(-)
Frost Thickness	CCD camera	in-situ	0.2 to 3 mm	±80 µm @ 0.4 mm ±40 µm above 1 mm	
Heat Transfer Rate	conduction side	in-situ*	5 to 8 W (17 to 27.3 Btu/h)	15%	(-)

Note: *Temperature bath and temperature sensor with an accuracy of ±0.05 °C (±0.1 °F) were used for on-site calibration.

The uncertainty of the volumetric airflow rate was estimated to be within 3%. The frost thickness's uncertainty was about ±80 microns if the frost thickness was below 0.4 mm and within ±40 microns for frost thicknesses above 1 mm. For a few tests, the test plate was quickly taken to a high precision scale at the end of the frost test to verify the mass balance. The mass of water vapor leaving the air was measured by the inline dew point meters, while the mass of frost deposited on the surface was measured by the precision scale. This redundant method confirmed the mass balance. The air side measurements were used to calculate frost density because they were more accurate and instantaneously measured during the entire frosting period. Table 3.1 provides more details on the experimental uncertainty and sensor accuracy. Figure 3.5 provides a photographic image of one of the two identical dew point sensors used in the mass measurements. The dew point meters were Michell Instruments S8000 Remote models, with serial numbers 152647 and 153419.

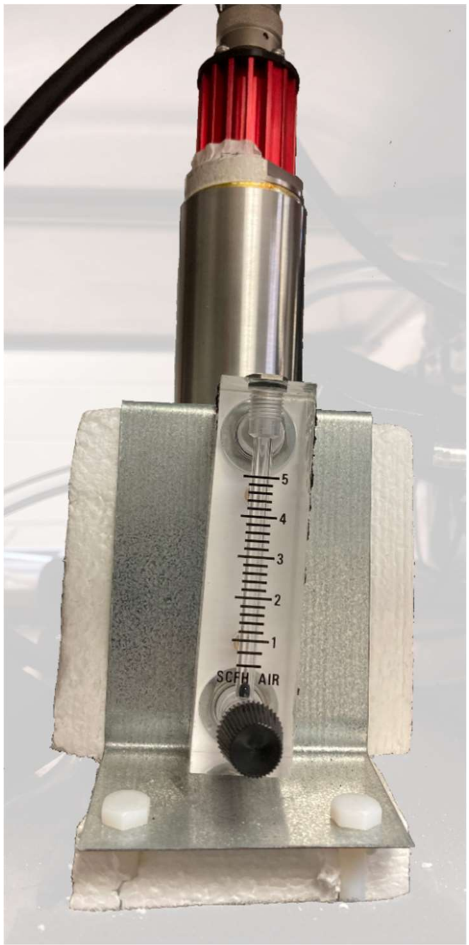


Figure 3.5: Photographic image of a dew point sensor used in mass measurements.

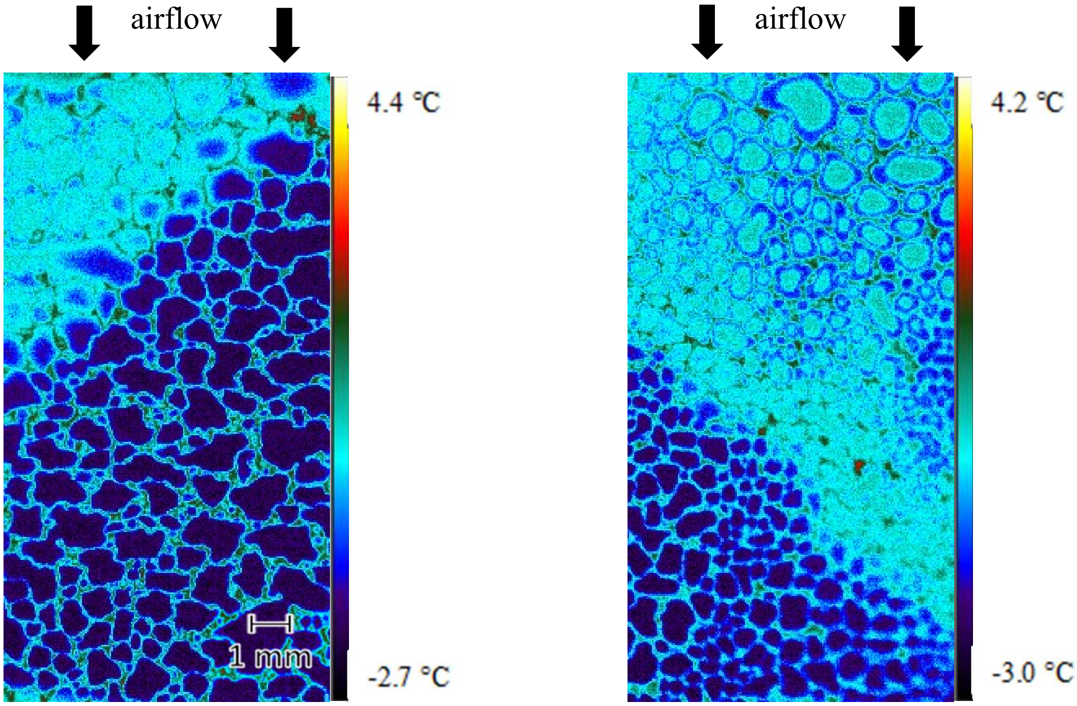
3.2 Methods Used to Clean and Dry the Surfaces after Each Frost Test

After each frost test, the thermoelectric modules were completely de-energized and the ethylene glycol flow to the T-block of the test set up was shut off. This allowed the test plate to gradually rise in temperature back to the temperature of the surrounding air in the wind tunnel, that is, to about 6 °C in a few hours. All frost melted, and the test plate was left to dry untouched through overnight exposure to the air flow in the wind tunnel. By the next day, all water had disappeared, and the test plate was ready for a new frost test. When repeating frost tests multiple times on the same test plate, it was observed that the surface became somewhat contaminated, resulting in significant differences in the frost nucleation and the freezing times between tests. Additional experiments were conducted to compare cleaning methods to achieve a more consistent and repeatable set of experimental measurements for frost nucleation. A comparison of the methods can be found in Adanur's thesis (2019). The final cleaning method adopted consisted of three steps: (a) cotton swabs dipped in isopropyl alcohol were used to swab the surface and remove contaminants; (b) new cotton swabs were dipped in distilled water to brush the test plate gently; and (c), lint-free cotton cloth was gently applied on the top of the test plate to absorb the remaining distilled water droplets from the surface. With this cleaning method, water marks left on the surface due to droplets evaporating back into the air stream were visually eliminated. Fine particulate matter with a diameter less than 2.5 μm (or $PM_{2.5}$) was the primary pollutant in outdoor ambient air where this work's experiments were performed. $PM_{2.5}$ is composed of combustion by-products such as soot, smoke, and automobile exhaust, and the concentration in ambient air in Auburn (AL, USA) was usually about 8 to 15 $\mu g/m^3$ on average. Since the MERV 9 air filter installed in the wind tunnel filtered out particles larger than 3 μm (such as textile fibers, dust mites, pollen, and hair), PM2.5 was also the primary contaminant that might have deposited on the test surfaces both during frost tests and left behind when the frost melted and water droplets re-evaporated back into the air stream. Nonpolar molecules, such as fats, oils, gasoline, and most organic molecules such as soot, were removed by the isopropyl alcohol, while most nonpolar molecules are insoluble in water. Thus, using isopropyl alcohol in step (a) was critical for removing $PM_{2.5}$ particulates deposited on the surfaces during testing. For the hydrophilic surface, the cleaning method was modified to omit step (a) because there was a concern that isopropyl alcohol may be adsorbed on the oxide layer and negatively affect the hydrophilicity of the surface. After multiple consecutive tests with the hydrophilic test plate, if the droplets were detected to be small and round instead of a large irregular dish-like shape and spread out on the surface, the test plate was physically removed from the test section. It was then submerged in a bath of 30% / 70% hydrogen peroxide/water solution for over one hour. This solution bath helped promote further oxide growth, reverting the droplets' appearance to their original sizes and shapes. For the bare aluminum and hydrophobic surfaces, the original cleaning procedure was performed after each frost test, and the bath was not required.

3.3 Infrared Technique to Observe Droplet Size and Freezing

Figures 3.6 and 3.7 provide examples of the IR images taken during freezing of each of the three surfaces after adopting the above cleaning method. The lighter-colored droplets were currently freezing. They appeared lighter than the other droplets because, during the capturing of the images, they were releasing latent heat during their freezing process. This phenomenon was also observed using infrared thermography by Alizadeh et al. (2012) during experiments of sessile droplets freezing on a cold flat surface. The change in color of the droplets determined the freezing time and freezing duration. The “freezing duration” was defined as the time from when the first droplet was observed to begin freezing until the last droplet in the field of view of the IR camera had completed its freezing process. This study's “freezing time” was defined as the period of subcooled liquid droplet growth plus half of the freezing duration and is expressed mathematically in Equation 3.1.

$$t_f = \left(t_{freezing\ onset} - t_{test\ start}\right) + \frac{1}{2} t_{freezing\ duration} \tag{3.1}$$

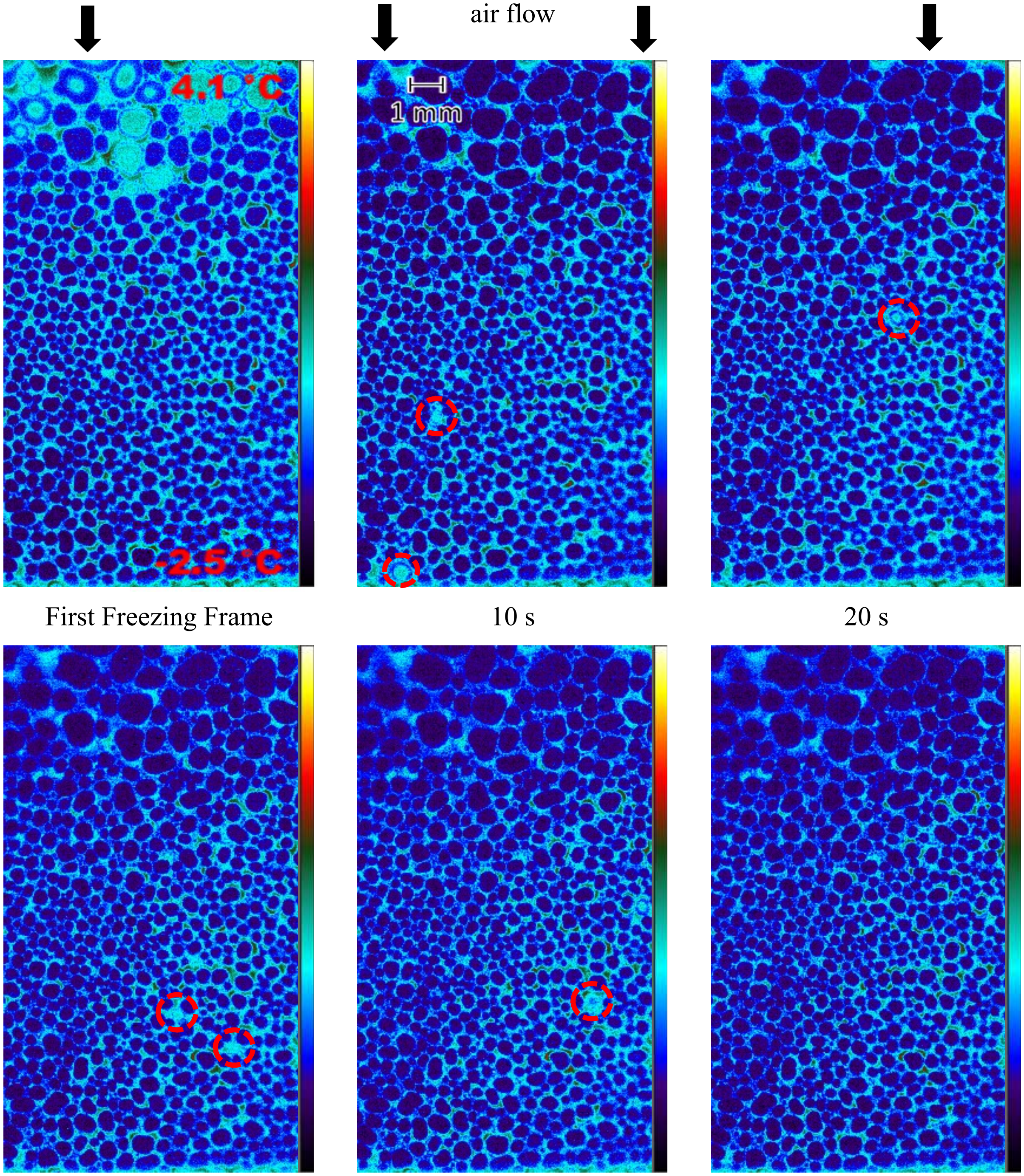


(a)

(b)

Figure 3.6: Images of droplets freezing on the cleaned test plates for (a) the bare aluminum surface and (b) the hydrophilic surface and for the same test conditions. Light-colored droplets were undergoing the liquid-to-solid phase transition, that is, freezing into ice beads. Dark blue droplets were either still in the liquid phase or had already frozen.

The freezing duration generally had ranges of a few seconds for the hydrophilic and bare aluminum surfaces. Since the IR camera captured images every two seconds, there was often only a single image of the freezing process for tests on these surfaces. Examples of these images on the bare aluminum and hydrophilic surfaces are shown in Figure 3.6. Droplets on both surfaces froze in bulk nearly instantly after the onset of freezing. The droplets looked very similar right before and after freezing, and there was no measurable preferred direction of the freezing propagation on the plates. The freezing duration was between thirty seconds and two minutes for the hydrophobic surface for most of the tests performed. This duration decreased closer to that of the other surfaces as the test plate's temperature dropped from -3.3 °C to -6.2 °C. Since the freezing durations were longer for the hydrophobic surface, several IR images were captured during the freezing process on this surface. Figure 3.7 presents a time series of images for a test at the same nominal experimental conditions as for Figure 3.6 on the hydrophobic surface, for which the freezing duration was about fifty-two seconds. This figure illustrates the differences between the freezing processes on the hydrophobic surface and the other two surfaces. Figure 3.7 shows that freezing began at the leading edge of the test plate. However, after the first few seconds, droplets froze in various, seemingly random locations on the plate surface by producing a sudden explosive color change detected by the IR camera. The number of droplets concurrently freezing decreased as the freezing process progressed until only one droplet was observed to still be in the freezing process by fifty seconds (though this droplet was out of the frame of the image in Figure 3.7). A few of the droplets in the midst of freezing were circled in the images for emphasis.



First Freezing Frame

10 s

20 s

30 s

40 s

50 s

Figure 3.7: Time series of experimental images for the freezing process on the hydrophobic surface. This test was conducted at the same nominal experimental conditions as for the surfaces in Figure 3.6.

Figure 3.8 gives a typical example of the freezing process of the droplets on the hydrophobic surface. The percentage of droplets in view that had frozen or were currently freezing was plotted against the elapsed portion of the freezing duration. Half of the droplets in view had frozen at a third of the freezing duration for this test. It took 80% of the total freezing duration for 95% of the droplets to freeze. Droplets on the bare aluminum and hydrophilic surfaces typically froze in a second or two; however, a few tests were exceptions to this rule in which the droplets did not all freeze at once. These exceptions comprised 11% of the bare aluminum surface tests used to determine freezing time averages for this research and 20% of the tests on the hydrophilic surface. The longer freezing durations on the hydrophilic surface were due to the degradation of the surface's hydrophilicity, as described in Section 3.2. These were some of the last tests conducted on the surface before it was removed and treated with the hydrogen peroxide solution. The extended freezing duration for the aluminum surface was due to a short freezing time and tiny droplets. With such small droplets, observation of a lighter-colored flash when they froze could not be made; instead, freezing was observed by tracking the frost front as it propagated across the surface. The freezing duration was determined to be the time it took for the frost front to travel across the field of view.

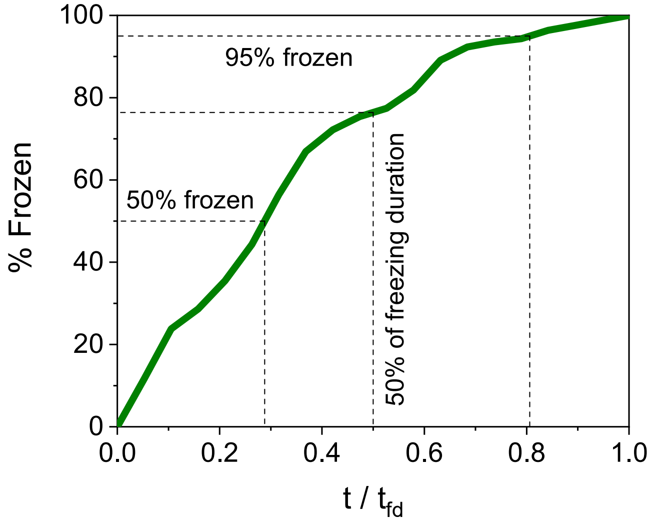


Figure 3.8: Droplet freezing distribution for the hydrophobic surface.

Figure 3.9 presents the results of a calibration test that was performed to explain the colors in the IR images in Figures 3.6 and 3.7. For this test, black acrylic paint with high emissivity was applied to a portion of one of the bare aluminum test plates. The test plate was then placed into the wind tunnel, and frosting and defrosting cycles were performed. Figure 3.9(a) shows a photograph of the bare aluminum (left) and painted (right) regions of the calibration test plate. Figure 3.9(b) shows an IR image of the same regions of the test plate while it was mostly dry, and 3.9(c) shows the same regions when they were covered with many condensed water droplets. Figure 3.9(d) shows the temperature legend associated with 3.9(c) in °C. Both regions of the test plate were about the same temperature, though the painted region appeared several degrees colder in Figure 3.9(b). This discrepancy occurred because the black paint had an emissivity close to 1, while the bare aluminum plate had a low emissivity (likely below 0.2). Since an emissivity value of 1 indicates that an object is a perfect absorber and emitter of radiation, the camera could pick up nearly the actual emitted radiation of the painted surface. Low emissivity values like that of aluminum indicate that most radiation incident on such a surface is reflected. For this test, the IR camera measured primarily the radiation emitted by the surroundings and reflected by the aluminum surface rather than the aluminum surface's actual radiation. Since the surroundings were the humid air and the wind tunnel walls, which were warmer than the test plate surface, the IR camera produced a temperature map in which the aluminum surface temperature appeared warmer than the painted surface temperature. Figure 3.9(c) further shows that the condensed droplets were visible on the bare aluminum surface but were not distinguishable on the painted surface. Since the droplets and the paint were at very similar temperatures, it was concluded that the droplets had emissivity values close to that of the paint. The significant difference in emissivity between the droplets and the test plate was very useful in observing droplet growth characteristics since it led to sharp contrasts between droplets and surface on the color maps.

The infrared thermography technique is a noninvasive way to measure the area and distribution of liquid on a solid surface that has been successfully used in the literature. Previous studies include de Lima et al. (2015), who used an infrared camera to measure droplet sizes during natural rain events, and Kim et al. (2013), who used an infrared technique to measure the wetted area fraction of water on a heater surface during subcooled pool boiling. In the current research, images such as those presented in Figures 3.6 and 3.7 were used to measure the droplet diameters during condensation and at freezing. The IR images were opened in commercially available imaging and digitizing software, significantly magnified, and scaled so that each droplet perimeter was visibly detected. The droplet perimeters were traced manually, and the software calculated the areas enclosed by each perimeter. Then, equivalent diameters were calculated for each droplet based on their measured areas. Finally, an area-weighted average was taken to calculate an average droplet diameter for the entire surface during frost nucleation. The geometric resolution on the droplet size was roughly ±50 µm, and it was limited by the pixel size of the images produced by the IR camera.

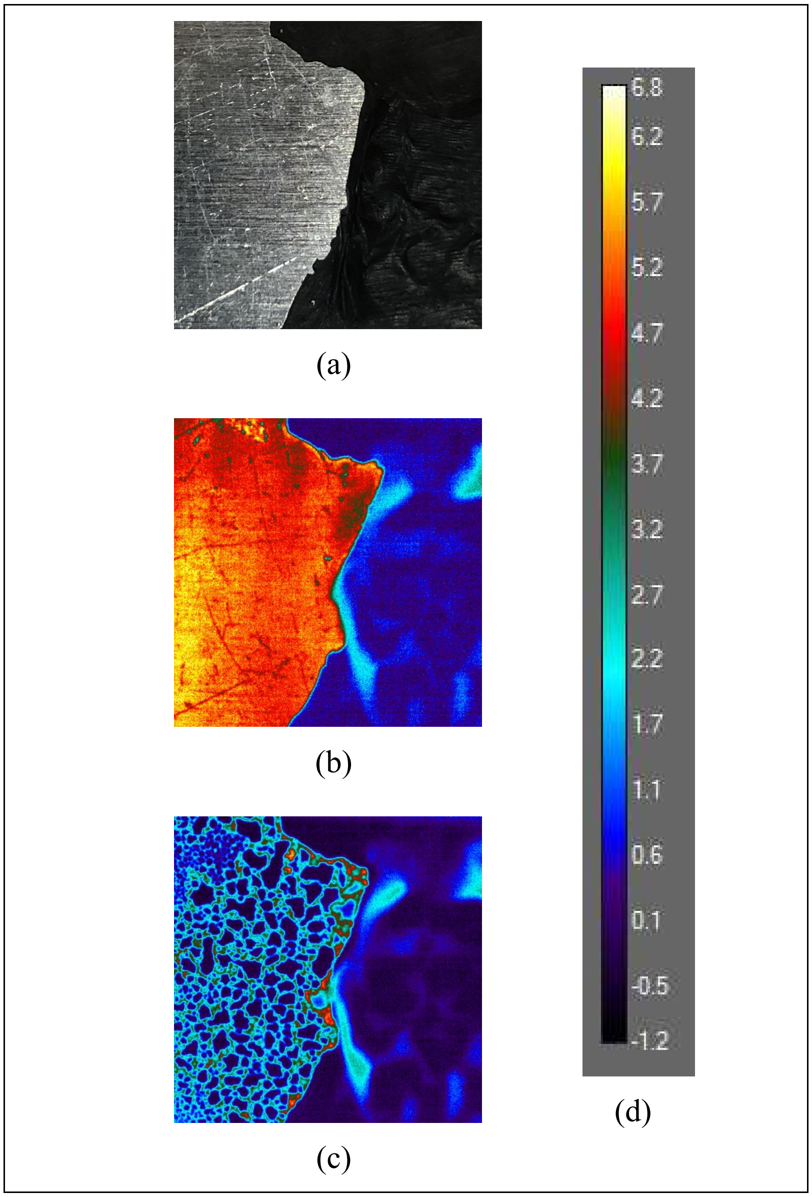


Figure 3.9: Example of images of the calibration surface, where the bare aluminum surface is on the left and the painted surface is on the right. (a) Photograph of the dry surface, (b) IR image of the mostly dry surface, (c) IR image of the surface with liquid water droplets, and (d) temperature scale associated with the image in (c), in °C.

3.4 Other Data Reduction

Data of air temperatures and humidity, pressure drop, and temperatures in the T-block assembly were measured and recorded by the National Instruments LabVIEW program every 2 seconds. A more detailed description of the data reduction can be found in Cremaschi et al. (2018), but the methods used to obtain some of the key values are provided here for clarity. The mass of the frost during the entire test run was calculated using Equation 3.2. The absolute humidity values in Equation 3.2 were calculated using the measurements obtained from the inlet and outlet dew point meters. Mass was calculated in this fashion to obtain instantaneous values in a noninvasive way throughout an entire frost growth experiment.

$$m_f = \sum \dot{m}_{air}(\omega_{in} - \omega_{out})\Delta t_{step} \tag{3.2}$$

The frost layer's volume and average density could then be calculated according to Equations 3.3 and 3.4, respectively.

$$Vol_f = A_s\delta_f \tag{3.3}$$

$$\rho_f = \frac{m_f}{Vol_f} \tag{3.4}$$

The frost layer volume was calculated with the assumption that the frost profile along the plate's depth, i.e., in the direction of the air flow, was horizontal. This assumption was confirmed by additional imaging analysis as part of this research and similar observations in the literature for convective flows (Hermes et al., 2018). The test plates' top surface temperature could be obtained using Equations 3.5 and 3.6, using an average value of the contact resistance between the heat flux meter and the test plates derived from calibration tests.

$$q''_{tee} = \frac{\Delta T_{tee}}{R_{tee}} \tag{3.5}$$

$$T_s = R_c\, q''_{tee} + T_{tee,bottom} \tag{3.6}$$

The air-side heat transfer coefficient was calculated according to Equation 3.7.

$$h = \frac{q''_{air}}{\left(T_{air,in} - T_{fs}\right)} \tag{3.7}$$

3.5 Test Conditions

Table 3.2 gives the test conditions and legend numbers for the seven different sets of nominal test conditions used for the bulk of the analysis and model development in this research. These test conditions were typical fins operating conditions in air-source heat pump systems, and they were adopted along with the new surface cleaning method described in Section 3.2. During winter-run operation of outdoor evaporators of air-source heat pump systems, and right after their defrost cycles, the finned structures of the heat exchangers often run at -3 to -4 °C (27 to 25 °F) for a few minutes, that is, during the period in which frost nucleation occurs. Then their surface temperatures drop to lower ranges of about -5 to -8 °C (18 to 22 °F) during the remaining run time, that is, during the period in which the frost layer thickens and densifies. These seven test conditions allowed for changing various environmental and surface set points in a parametric fashion. The analysis presented in the remainder of this chapter focused on test conditions 1-5. Test conditions 1 was the "baseline" test for this research, while test conditions 2-5 individually varied air relative humidity, test plate temperature, the temperature difference between the air and the test plate, and air velocity. Test conditions 6 and 7 both changed more than one set point from the baseline conditions. They were used for correlation development rather than drawing general conclusions about the effects of various environmental and surface conditions on frost nucleation and growth.

Table 3.2: Test conditions and legend numbers

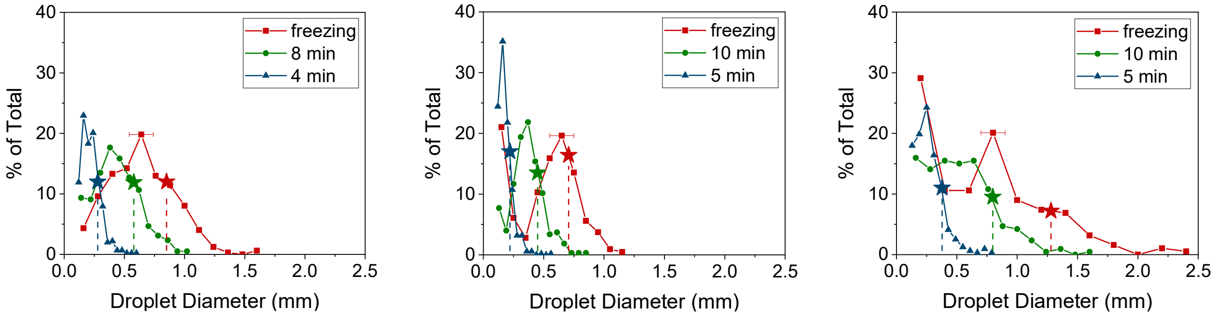
	LEGEND						
TEST CONDITIONS REFERENCE	1	2	3	4	5	6	7
Test Plate Temperature, °C (°F)	-3.3 (26)	-3.3 (26)	-6.1 (21)	-5.0 (23)	-3.3 (26)	-3.3 (26)	-6.1 (21)
Relative Humidity, %	75	85	75	68	75	85	85
Air Temperature, °C (°F)	5.0 (41)	5.0 (41)	5.0 (41)	8.3 (47)	5.0 (41)	5.0 (41)	5.0 (41)
Air Velocity, m/s (ft/s)	3.8 (12.5)	3.8 (12.5)	3.8 (12.5)	3.8 (12.5)	1.7 (5.6)	1.7 (5.6)	1.7 (5.6)

3.6 Droplet Growth and Freezing Characteristics

3.6.1 Droplet Distribution during Growth

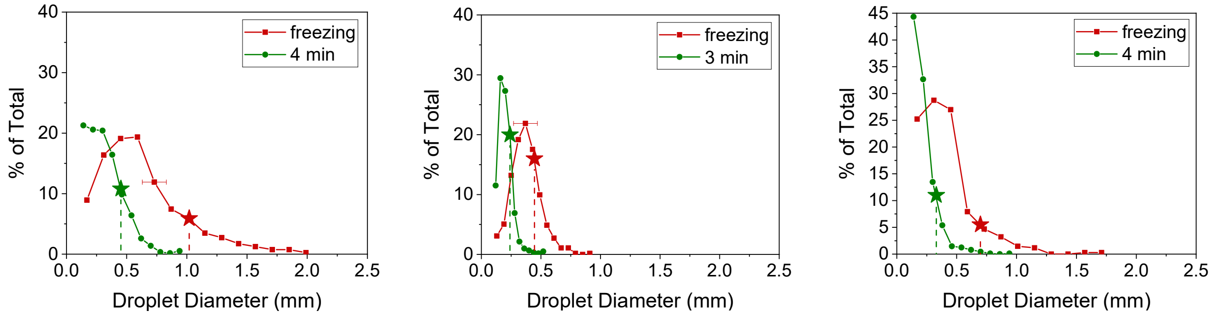
Water droplets condensed from the water vapor in the air within the surface thermal boundary region. Then these initial droplets grew until they froze and turned into ice beads. The projected droplet diameter was the equivalent circle's diameter projected from a droplet to the flat surface underneath it. Thus, tall and round droplets resulted in a lower projected diameter than flat disk-like elongated droplets that spread out on the surface with asymmetrical perimeters. Surface wettability not only can affect the droplet shape and size but also the droplet size distribution. The droplet size distribution, which unfortunately was seldom reported in the frost nucleation literature studies, provided information on how the frost nuclei formed on the surface and the dominant characteristic length scale during frost nucleation. The time-dependent size distributions of droplets on the three surfaces investigated during this research were measured. Data for test conditions 2 and 4 are presented in Figures 3.10 and 3.11, respectively, as examples. In these figures, the droplet diameters presented in the x-axis are the projected diameters. The number of droplets, expressed as percentages over the droplets' total number, is given on the y-axis. For each distribution curve, the summation of all the y-values is 100 percent. The star data point on each distribution curve represents the average diameter for that particular distribution, determined by using droplet surface coverage area-weighted averages. This weighted average method is commonly used in the literature to report a droplet diameter. However, these figures show that the average droplet diameter was always higher than the diameters of most of the droplets used to calculate the weighted averages. For example, in Figure 3.10(a), at the freezing time (red square data points), the average droplet diameter was about 0.8 mm. In contrast, the majority of the droplets had diameters of about 0.6 mm. This discrepancy could be considered a slight difference, which was only valid for the bare aluminum surface. If one considers the same type of analysis for the hydrophilic surface of Figure 3.10(c), the average droplet diameter was 1.3 mm while the majority of the droplets had much smaller diameters of about 0.3 mm (first peak of the red square data points in the distribution of Figure 3.10(c) and 0.7 mm (second peak of the distribution). In other words, for the hydrophilic surface, the droplets were more spread out, appearing large in diameter when they froze on the surface. However, many minute droplets were also present around the largest droplets at the freezing time. The differences between average (star) diameters and peak diameters affect the density of the first layer of frost during the onset of frost formation. The distribution curves do not extend to droplet diameters of 0 mm because, with the IR camera used in the current study, droplets with diameters less than 0.1 mm were too small to be distinguished from each other. It should also be noted that horizontal experimental error bars are shown on one representative point for each plot for clarity of the plots. However, similar error bars apply to all the data points of the distributions. These error bars represent a potential error in the measurements due to the uncertainty of the IR camera. Additionally, it is essential to point out that the curves in Figures 3.10 and 3.11 represent individual test runs and not averages over multiple repeated tests.

Figure 3.10 shows that, for test conditions 2, the hydrophobic surface's droplet distributions had the tallest, narrowest peaks, while those on the hydrophilic surface were more spread out. Additionally, the tails on the right sides of the distributions were quite large for the hydrophilic surface because a few large droplets were measured. The distribution curves on all three surfaces were narrow at small droplet sizes but widened as time went on. These trends support the current study’s visual observations of droplet coalescence phenomena before freezing. Large droplets were created during coalescence, and secondary nucleation of tiny droplets occurred in the regions left behind after larger droplets moved closer to each other and coalesced together. An interesting observation was that, for the case of freezing on both the hydrophobic and hydrophilic surfaces, there were spikes in the distributions at droplet diameters less than 0.3 mm in addition to peaks at larger droplet sizes. These spikes were due to the increasing dominance of the effects of secondary nucleation on the distribution. In these distributions, this occurred when the first peaks in the droplet diameters had passed projected diameters of 0.5 to 0.6 mm. Similar observations can be made concerning Figure 3.11. For test conditions 4, the distribution curves for the hydrophilic surface were not as spread out as for test conditions 2 but had shapes that were in between those for the hydrophobic and bare aluminum surfaces. There was also negligible secondary nucleation since only a single peak, and no additional spike, was measured for the distributions in Figure 3.11. Freezing occurred quickly for test conditions 4, and the droplets might not have had the time to grow and nucleate again under this specific set of high surface-to-air temperature differences. In other words, freezing on the surfaces occurred before coalescence and secondary nucleation were measurable and before they produced large enough effects to alter the droplet size distributions of these test conditions.



(a) (b) (c)

Figure 3.10: Time evolutions of droplet distributions for test conditions 2 on the (a) bare aluminum surface, (b) hydrophobic surface, and (c) hydrophilic surface.



(a) (b) (c)

Figure 3.11: Time evolutions of droplet distributions for test conditions 4 on the (a) bare aluminum surface, (b) hydrophobic surface, and (c) hydrophilic surface.

Figure 3.12 provides illustrations of the droplet distributions on all three surface types at test conditions 2. Images were provided at the same elapsed testing times for each surface to eliminate the effects of growth time on the droplet distributions. These times are 4 minutes, 8 minutes, and 12 minutes into the tests (though the droplets on the bare aluminum surface froze just before 12 minutes, so this surface's third image presented in Figure 3.12 was taken at 11.9

minutes). This figure confirms the results observed in Figure 3.10. The droplets on the hydrophobic surface were more uniform in shape and size than on the other surfaces (leading to narrow size distributions). In contrast, the droplets on the hydrophilic surface had the largest ranges of shapes and sizes (leading to wide distributions).

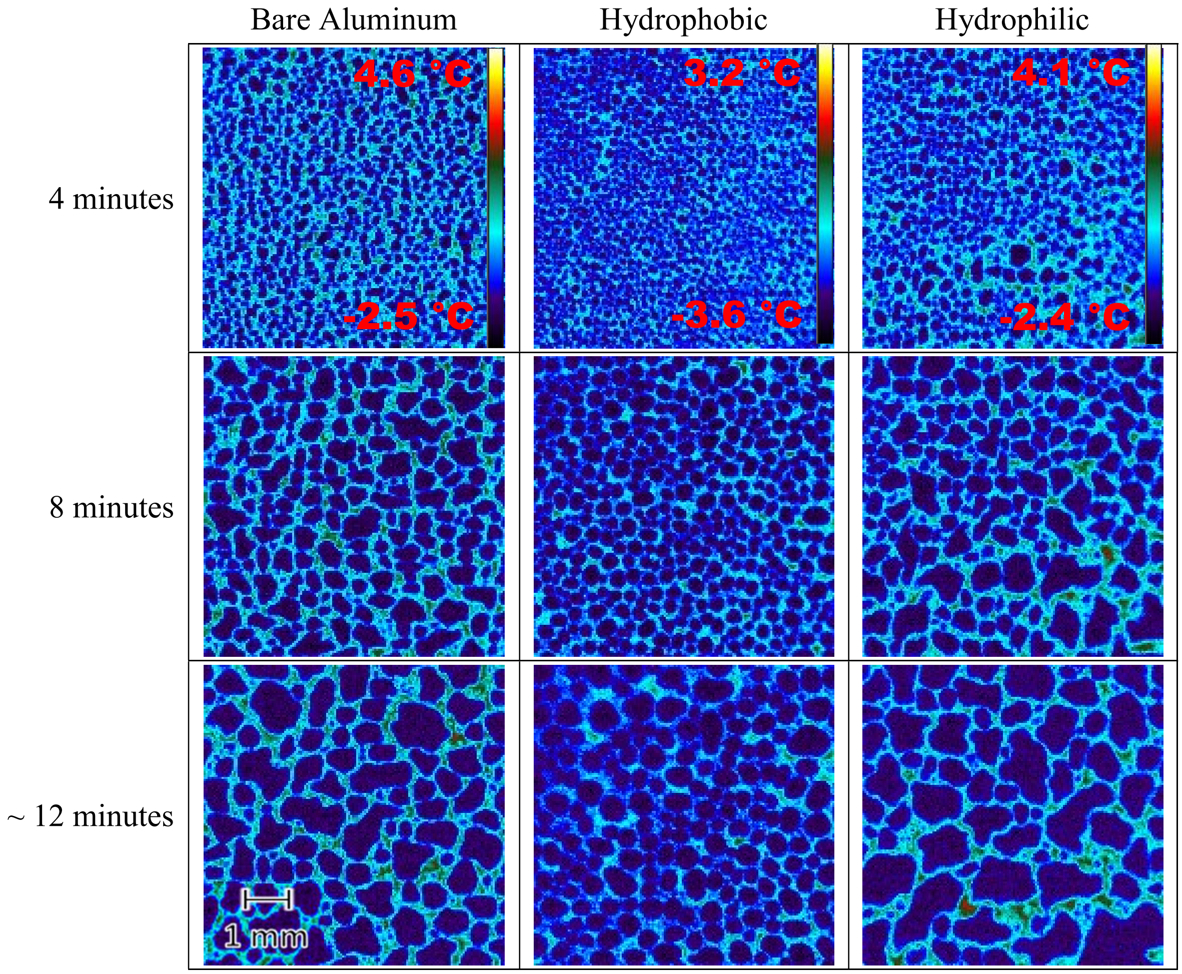
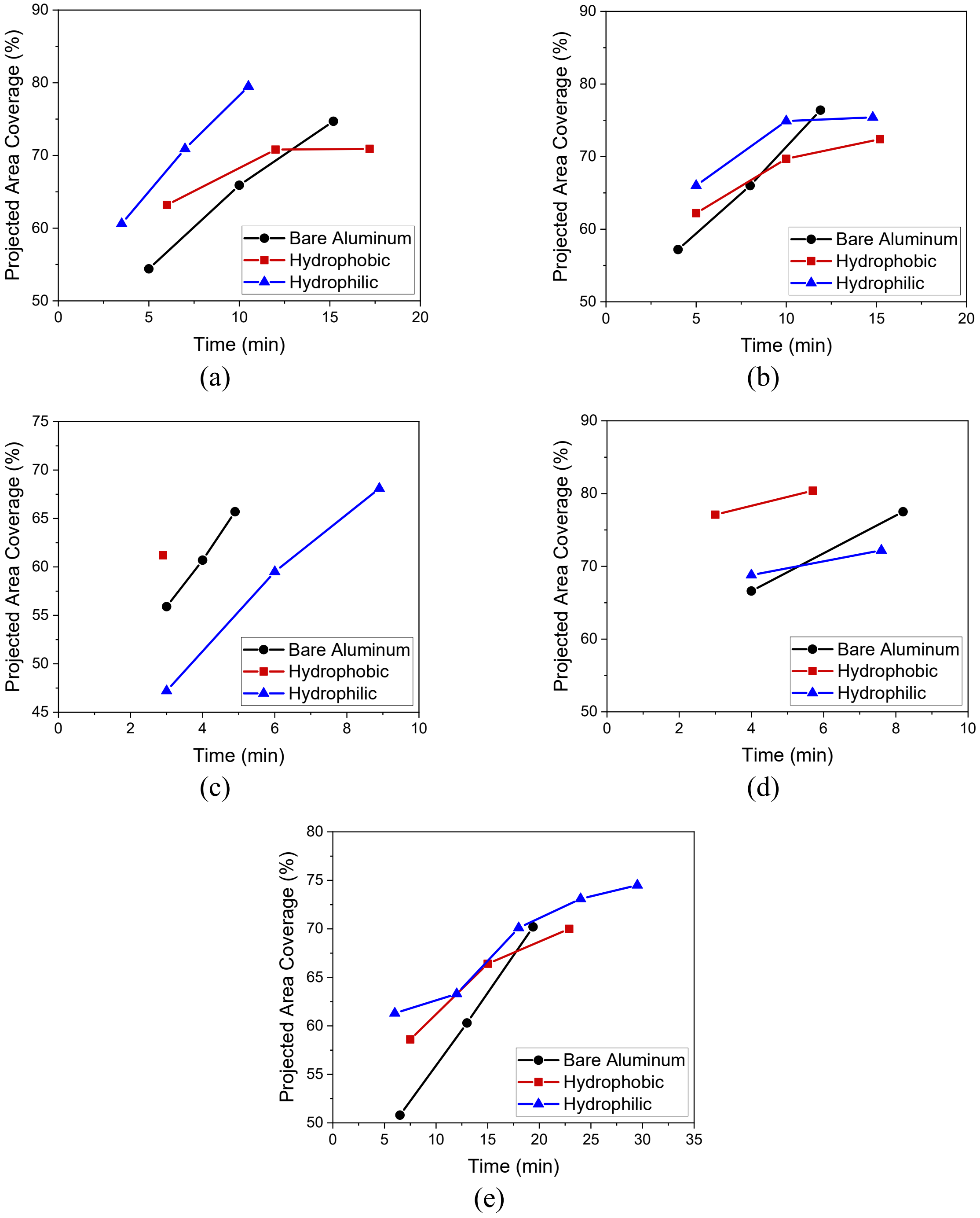


Figure 3.12: Droplet distributions on all three surface types at test conditions 2.

3.6.2 Droplet Surface Area Coverage during Growth

Just as surface wettability affected droplet size and distribution due to its ability to change droplet geometry, it also affected the projected area coverage of droplets on the test plate surface. Area coverage also impacts the frost layer's initial state because a larger area coverage indicates either flatter, more spread out droplets, or more liquid water on the surface. Both lead to a higher frost density immediately after the liquid droplets freeze and become ice beads. Figure 3.13 presents curves for projected droplet area coverage for test conditions 1 – 5, and Figures 3.13(b) and (d) correspond to the curves represented in Figures 3.10 and 3.11, respectively. In these figures, the point at the largest time value for each curve represented the point of freezing for that individual test run. All the plots showed that area coverage increased as the droplets grew over time. Figures 3.13(a), (b), and (e), which present area coverage for test conditions 1, 2, and 5, respectively, both showed very similar results. In all three plots, the hydrophilic surface had the largest area coverage. Also, in all three, the bare aluminum surface started with the smallest area coverage. However, it then crossed over the hydrophobic curve so that the hydrophobic surface had the smallest area coverage by the end of the condensation period. Interestingly, for the case of low air velocity (test conditions 5 and Figure 3.13(e)), the area coverage values were similar for all three surface types. Figures 3.13(c) and (d), which present area coverage results for test conditions 3 and 4, show very different results from the other test conditions. For these cases, the hydrophobic surface had the largest area coverage for a given elapsed testing time. These two test conditions represent lower test plate surface temperature, while the other three all had the same higher surface temperature. It appears likely that if freezing had not occurred so quickly for test conditions 3 and 4, area coverage on the bare aluminum and possibly on the hydrophilic surfaces would have crossed over the hydrophobic curve and been larger than on the hydrophobic surface later in the tests. Coalescence phenomena likely cause this intriguing behavior. When the droplets on all surfaces are still tiny, droplet coalescence does not have significant effects. However, coalescence effects grow over time and are more prominent at smaller contact angles (see Section 3.6.3), so it would be logical for area coverage on the bare aluminum or hydrophilic surface to overtake and exceed that on the hydrophobic surface.



(a)

(b)

(c)

(d)

(e)

Figure 3.13: Projected area coverage vs. time on all three surface types for (a) test conditions 1, (b) test conditions 2, (c) test conditions 3, (d) test conditions 4, and (e) test conditions 5.

3.6.3 Direct Droplet Growth and Coalescence Characteristics

This section highlights the effects of surface wettability and environmental conditions on droplet growth by coalescence compared to droplet growth by direct water vapor condensation. These effects are important to include during the development of droplet growth and freezing models, especially when predicting characteristics of large droplets such as those formed under typical heat pump operating conditions. The results of these droplet growth predictions affect predictions for droplet size at freezing and thus the thickness and density of the initial layer of frost. One method of approximating droplet growth is using a power-law (Beysens and Knobler, 1986; Meakin, 1992; Beysens, 1995). This method was initially based on experimental results and has proved to be quite robust over the years. The standard equations used in the literature to approximate average droplet diameter, $< D >$, and the diameter of a single droplet with no interactions with other droplets, D_s, are given as Equations 3.8 and 3.9, respectively. In these equations, μ_s was initially found to be 1/3, while $\mu_a = 3\mu_s = 1$.

$$< D > \propto t^{\mu_a} \tag{3.8}$$

$$D_s \propto t^{\mu_s} \tag{3.9}$$

Figure 3.14 presents single and average droplet growth for a single test run at test conditions 1 on the bare aluminum surface as an example of the droplet growth and coalescence data gathered during this research. The average droplet growth profile was generated by tracking a single average-sized droplet through the entire condensation process. Its equivalent projected diameter was calculated immediately before and after each coalescence event. For the single droplet growth curve, the segments with shallow slopes between the coalescence events were joined together to approximate the growth rate by direct condensation, as was done in Beysens and

Knobler (1986). For this particular droplet, μ_a was calculated to be 1.34 while μ_s was 0.32, leading to the ratio μ_a / μ_s of 4.2. Figure 3.15 presents IR images of the actual droplets from Figure 3.14 immediately before and after each coalescence event.

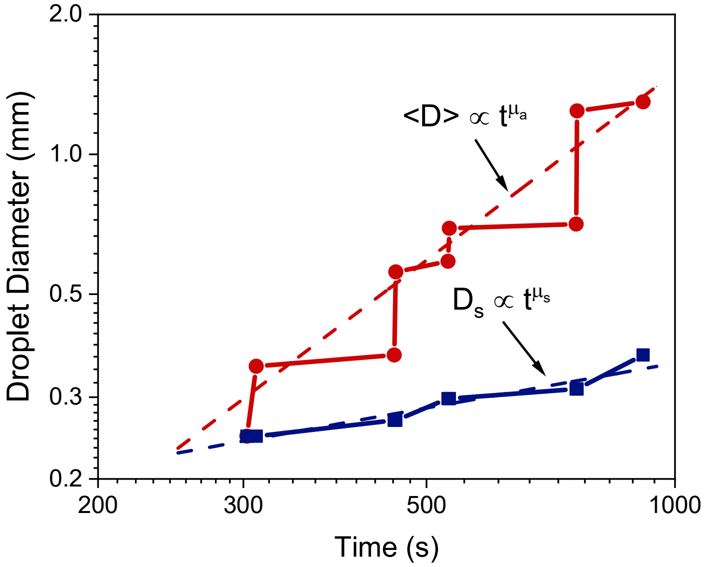


Figure 3.14: Droplet growth with (top line) and without (bottom line) coalescence effects for test conditions 1 on the bare aluminum surface.

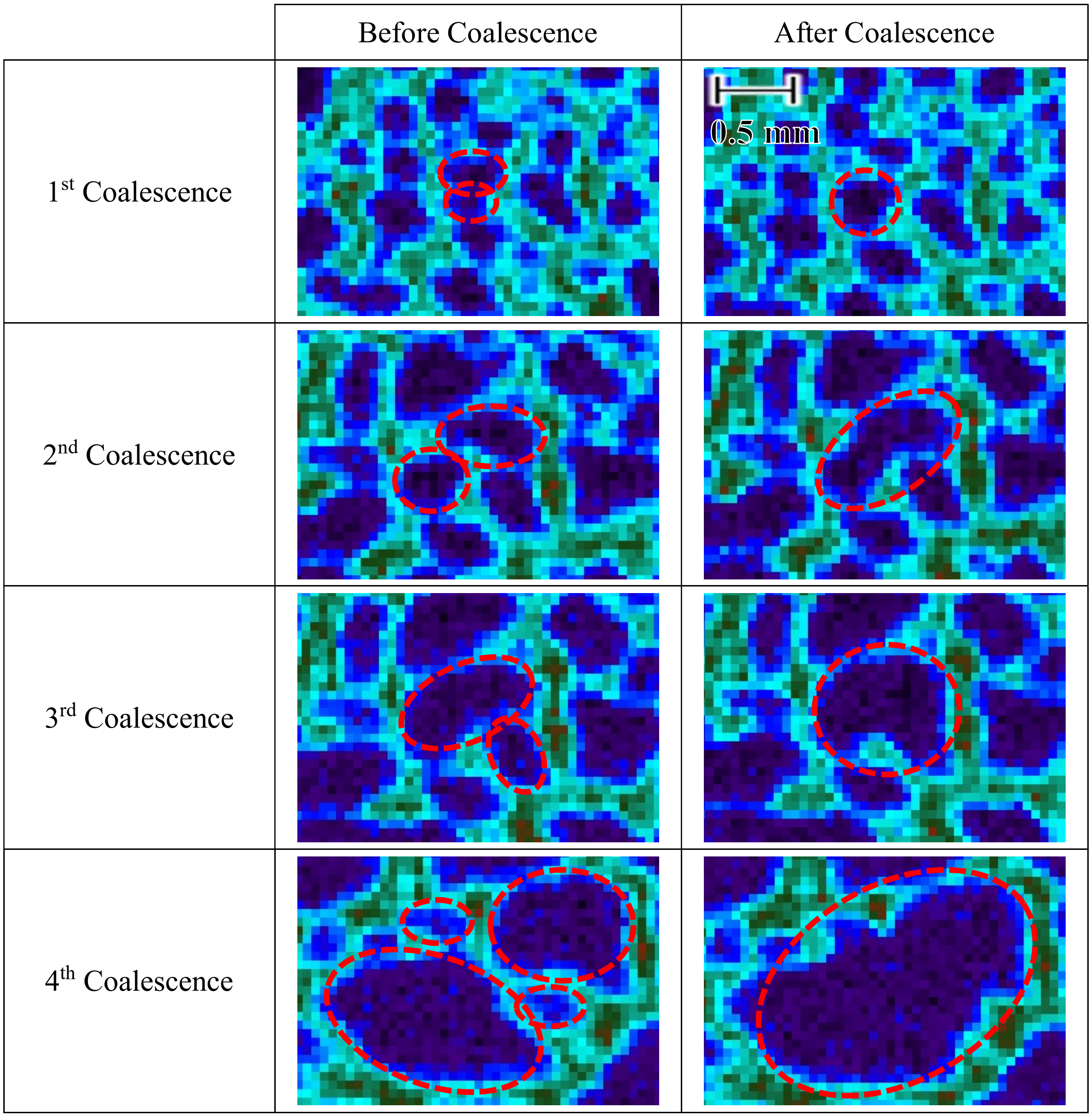


Figure 3.15: Coalescence events for test conditions 1 on the bare aluminum surface corresponding to the droplet profile in Figure 3.14.

Table 3.3 presents data for μ_a, μ_s, and their ratio by surface type. The values in Table 3.3 represent the results of several test runs at different experimental conditions averaged together. For the data in Table 3.3, test plate temperatures ranged from -3.2 to -6.2 °C, relative humidity values spanned 61 to 85%, air temperatures ranged from 4.8 to 8.2 °C, and air velocities were either 1.7 or 3.8 m/s. The principal results of interest were that (a) the ratio of the average to single droplet exponents decreased as contact angle increased, and (b) that this ratio was always higher than the conventional average value of 3 reported in the literature. The ratio of exponents between average and single droplet growth was much higher for the hydrophilic surface. It indicated that droplet coalescence played a more prominent role in droplet growth for the hydrophilic surface than for the hydrophobic. This discrepancy occurred because droplets were flatter and spread out on the hydrophilic surface and so grew more quickly in the radial direction than in the vertical; this led to droplets encountering each other and coalescing more quickly. It was also observed that the average to single droplet exponent ratio decreased as both the supersaturation degree and the temperature difference between the air stream and the test plate surface increased. In contrast, it increased slightly as the freezing time increased. Air velocity did not have a conclusive effect on the exponent ratio. These observations indicated that coalescence effects became more important at low heat and mass transfer driving potentials and when the droplets had more time to grow before freezing.

Table 3.3: Values for average and single droplet power-law exponents by surface type

Surface Type	μ_a	μ_s	μ_a / μ_s
Bare Aluminum	0.94	0.25	4.4
Hydrophobic	1.33	0.37	3.6
Hydrophilic	1.21	0.27	5.4

In Table 3.3, both μ_a and the ratio of exponents are higher than their traditional values of 1 and 3, respectively, for the hydrophobic and hydrophilic surfaces. Two possible reasons are proposed for this discrepancy. First, the fitting that produced μ_a was dependent on which numerical value was chosen as the smallest point in the data series, a fact also documented by Boreyko and Collier (2013). The data used to generate Table 3.3 began with the coalescence of the smallest droplets which could be distinguished; the diameters of these droplets at the time of the first reported coalescences were generally between 0.15 and 0.25 mm. Equations exist to calculate the minimum droplet diameter required for the onset of condensation. However, these equations were not used to calculate the smallest data point to avoid mixing measured and calculated data for the fitting. The smallest data points used for fitting were diameters of around 0.2 mm on average, larger than most of the droplet sizes used for fitting in the literature. Second, the droplets measured in this research grew much larger than most of those reported in the literature. Castillo et al. (2015) reported different regimes of droplet growth, where the fitted exponent was different in each regime and increased as the droplets grew (until the effects of new nucleation in bare spaces left between large coalesced droplets dominated the droplet growth process). Grooten and van der Geld (2011) also reported exponent values significantly greater than 1 for high humidity values and droplet diameters larger than 1 mm. Also, observations of data gathered in this research indicated that, while the droplet coalescences rate was fairly steady throughout a test, the droplet size increase due to coalescence increased as the tests progressed, while the size increase due to direct condensation did not. This phenomenon led to greater and greater dominance of coalescence effects on droplet size as the droplets grew. These observations and the literature results confirmed that it was possible for μ_a of this research to be larger than that of previous studies in the literature.

To further demonstrate that surface wettability played a role in droplet growth by coalescence, the coalescence rate was calculated for test conditions 1 on each of the three surfaces and is presented in Table 3.4. Coalescence rates were calculated by tracking a single droplet through its growth process and counting the number of coalescence events for that droplet. This number was then divided by the period over which the coalescences occurred, which for each droplet was the time immediately before its first distinguishable coalescence event up until freezing for that test run. The coalescence rates presented in Table 3.4 are ranges of the rates calculated for five different droplets from the same individual test run. The results in Table 3.4 agree with those in Table 3.3; both illustrate that coalescence effects on droplet growth increased as the surface's contact angle decreased.

Table 3.4: Number of coalescence events per minute for each surface type

	SURFACE TYPE		
TEST CONDITIONS	Bare Aluminum	Hydrophobic	Hydrophilic
1	0.39 – 0.43	0.27 – 0.39	0.47 – 0.67

3.6.4 Droplet Freezing Characteristics

Figure 3.16 presents the average freezing time for each of the three surface types at test conditions 1, 2, 4, and 5. The error bars represent the ranges of freezing times measured in multiple tests at the same nominal test conditions. These repeated tests were conducted on different days (and sometimes in different months) and on different test plates with the same nominal surface types. These repetitions were performed to assess the variability of the data due to the effectiveness of human operators to prepare the coatings and prepare for the preconditioning of the test plate before frost nucleation, due to possible contaminants in the ambient air, response to building electric power noise and minor variation of the lab environmental conditions of temperature and humidity. The average projected droplet diameters and their ranges measured during repeated tests are reported in Figure 3.17. All the freezing time and droplet diameter values are averages of measurements taken during two or three identical tests, except for a few cases where there was only a single test with usable measurements.

Increasing the air stream's nominal humidity (test conditions 2) did not have much effect on the freezing times for any of the test plates. Freezing times were slightly lower than for test conditions 1 for the bare aluminum and hydrophobic surfaces, while it was slightly higher for the hydrophilic surface. For all three test plates, increasing nominal air temperature and decreasing nominal test plate temperature (test conditions 4) reduced the freezing time. The increase in temperature difference increased the driving potential for both heat and mass transfers, ultimately accelerating the freezing process. Decreased air velocity drastically increased the freezing time for all three test plates. A decrease in airflow rate practically reduced the driving potential for heat and mass transfer, thereby slowing down the freezing process. The hydrophobic surface took longer to freeze than the other two surfaces except for test conditions 4. The freezing times for the hydrophilic surface were longer than for the bare aluminum surface for test conditions 2 and 5 but smaller for test conditions 1 and 4. Because the order in which droplets on the different surface types froze was not consistent for different test conditions, something other than surface wettability type was driving the differences in freezing time at the same nominal test conditions. Close observations of the data showed that two quantities, the supersaturation degree (SSD) defined in Equation 3.10 and the temperature difference between the bulk airflow and test plate surface, were good indicators of differences in freezing times even when absolute temperature and relative humidity values were quite similar. For example, a variation as small as 0.05 in SSD or 0.1 °C in ΔT had the potential to decrease freezing time by 100 seconds or more. Additionally, freezing times at low flow test conditions (test conditions 5) were more sensitive to these small changes than at other test conditions. Therefore, small fluctuations of air temperature, test plate temperature, and humidity even in well-controlled tests were responsible for much of the fluctuation in freezing times at the same nominal test conditions. At the same time, the surface wettability type had a small secondary effect on the freezing times.

$$SSD = \frac{\left(P_v - P_{v,sat,s}\right)}{P_{v,sat,s}} \tag{3.10}$$

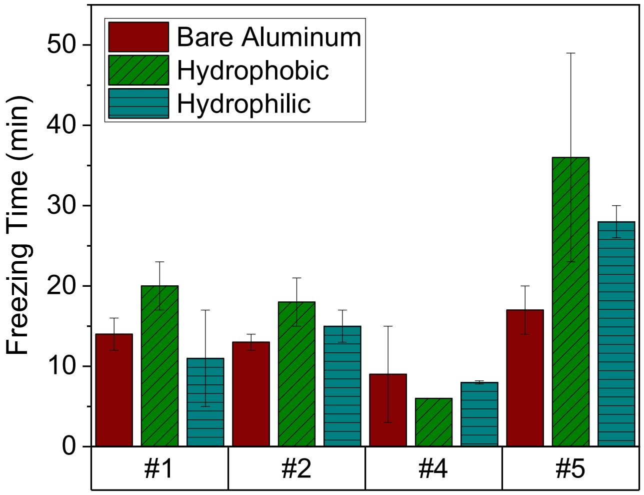


Figure 3.16: Freezing times for each surface at different test conditions.

Surface wettability had more measurable effects on the projected droplet diameters at the time of their freezing. The ice bead diameters on the hydrophobic surface were smaller than those for the hydrophilic surface for all test conditions except 1, while they were smaller than those for the bare aluminum surface for test conditions 2 and 4. This trend occurred even when freezing times were longer for the hydrophobic surface so the droplets had more time available to continue to grow larger and larger before turning into ice beads. But the contact angle for this surface was higher, leading to tall and round droplets. On the contrary, the droplets on the bare aluminum and the hydrophilic surfaces were flatter and spread out. Frozen droplet diameter was not significantly affected when the relative humidity was increased for the bare aluminum or hydrophobic surfaces, but it drastically increased for the hydrophilic surface. This increase was due to the increase in freezing time and the increase in the mass of water accumulated on the surface from vapor transfer in high humidity air. The hydrophilic test plate's low contact angle led to a much larger increase in droplet radius than height when more water mass was deposited on its coated surface.

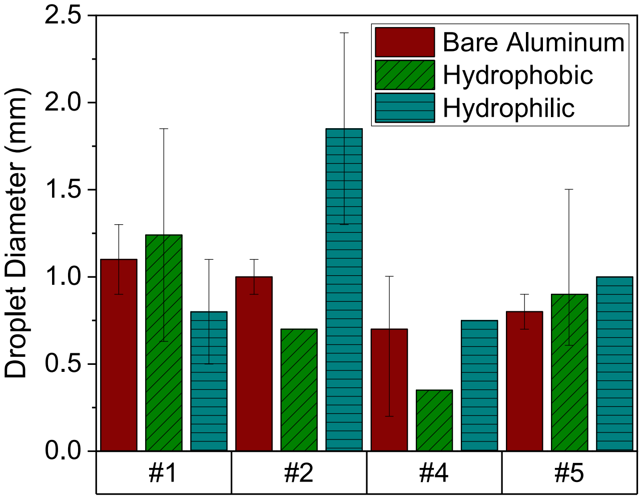


Figure 3.17: Average projected droplet diameters for each surface at different test conditions.

3.6.5 Effects of Surface Cleaning on the Droplet Growth and Freezing

Table 3.5 gives the test conditions and test legend numbers for five sets of nominal test conditions using uncleaned bare aluminum, hydrophobic, and hydrophilic surfaces. Data taken during this series of testing with uncleaned surfaces were first presented in Cremaschi et al. (2018). The test conditions in Table 3.5 are comparable with the corresponding test conditions in Table 3.2, except that the humidity levels in Table 3.5 are slightly higher than in Table 3.2.

Table 3.5: Test conditions and legend numbers for tests with no cleaning procedure

	LEGEND			
TEST CONDITIONS REFERENCE	1	2	4	5
Test Plate Temperature, °C (°F)	-3.3 (26)	-3.3 (26)	-4.4 (24)	-3.3 (26)
Relative Humidity, %	79	91	67	79
Air Temperature, °C (°F)	5.0 (41)	5.0 (41)	8.3 (47)	5.0 (41)
Air Velocity, m/s (ft/s)	3.8 (12.5)	3.8 (12.5)	3.8 (12.5)	1.7 (5.6)

Figures 3.18 and 3.19 present the freezing time and projected frozen droplet diameter for each of the three uncleaned surface types under the test conditions listed in Table 3.5. The test conditions had similar effects on freezing time for the uncleaned surfaces as for the corresponding tests with the cleaned test plates. For the hydrophobic and hydrophilic surfaces, freezing time decreased as humidity increased (just like in Figure 3.16). However, the decrease for the uncleaned surfaces was more drastic than for the cleaned surfaces. There was also a significant increase in freezing time as the airflow rate decreased, just like the cleaned surfaces. The main difference in test conditions effects between the cleaned and uncleaned surfaces was for test conditions 4. For the cleaned surfaces, this set of test conditions produced the shortest freezing time, while for the uncleaned surfaces, these freezing times were longer than for test conditions 2. Another similarity to the results for the cleaned surfaces was that surface type had a negligible effect on freezing time on the uncleaned surfaces. Once again, supersaturation degree and temperature difference were primary predictors of which surface would freeze first. For example, the tests on the bare aluminum surface at test conditions 2 and test conditions 4 had significantly smaller SSD than for the other two surfaces, which led to longer freezing times (or no freezing, as in the case of the high RH test conditions). For test conditions 5, the test on the

hydrophilic surface had a slightly smaller ΔT than for the other two surfaces. Additionally, in general, freezing times were longer for the uncleaned surfaces than for the cleaned surfaces. In Figure 3.19, frozen droplet diameter seemed to be directly correlated to how long the droplets grew before they froze, with only a couple of exceptions. This correlation is in direct contrast to the results in Figure 3.17, where frozen droplet size was dependent on surface wettability.

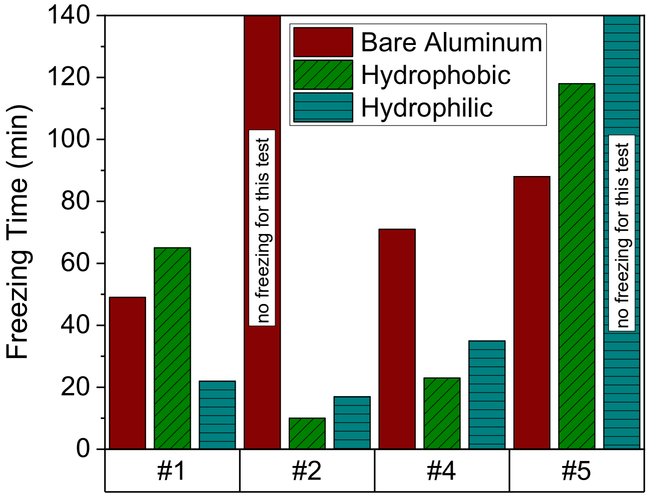


Figure 3.18: Freezing times for each uncleaned surface at each of the test conditions in Table 3.5

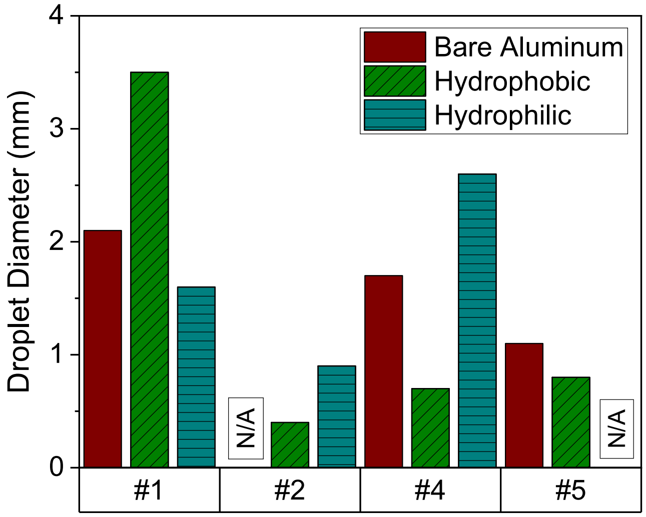


Figure 3.19: Average projected droplet diameters at freezing for each uncleaned surface at each of the test conditions in Table 3.5

Table 3.6 and Figure 3.20 present further droplet data that supported the hypothesis that contaminants that were left behind as residuals after frosting tests skewed the droplets' shapes and sizes as they grew. Table 3.6 presents the average projected droplet diameter for each surface, cleaned and uncleaned, for test conditions 2 at 8 minutes into the frost nucleation tests. Results were compared at the same testing time for all three surfaces to separate the effects of the surface wettability on freezing time from its effects on droplet size and shape. The data presented in Table 3.6 indicated that when the surfaces were cleaned, there was a marked difference in droplet diameter for each surface type. The droplets on the hydrophobic surface had the smallest average diameter, while those on the hydrophilic surface had the largest. In addition, the flat and irregularly shaped disk-like droplets that spread out on the hydrophilic surface had an average diameter almost double that of the hydrophobic surface. This result is supported by the small contact angle of water droplets on hydrophilic surfaces. In contrast, when the surfaces were not cleaned, they all had practically the same droplet diameter after 8 minutes. This observation indicated that when the surfaces became contaminated, the effect of the surfaces' wettabilities on the droplet contact angle significantly diminished.

Table 3.6: Average projected droplet diameter, in mm, after 8 minutes of testing for each surface, cleaned and uncleaned, for test conditions 2

	Bare Aluminum	**Hydrophobic**	**Hydrophilic**
Cleaned	0.58	0.36	0.65
Uncleaned	0.40	0.40	0.40

Figure 3.20 shows images of the droplets on the surfaces that correspond to the data presented in Table 3.6. The differences between the droplets' sizes and shapes on the hydrophilic and hydrophobic surfaces can be clearly seen for the cases where the surfaces were cleaned. For the uncleaned surfaces, however, the droplets all looked very similar. This observation was a

further indication that residual contaminants significantly inhibited the effects of surface wettability with regard to droplet size and shape.

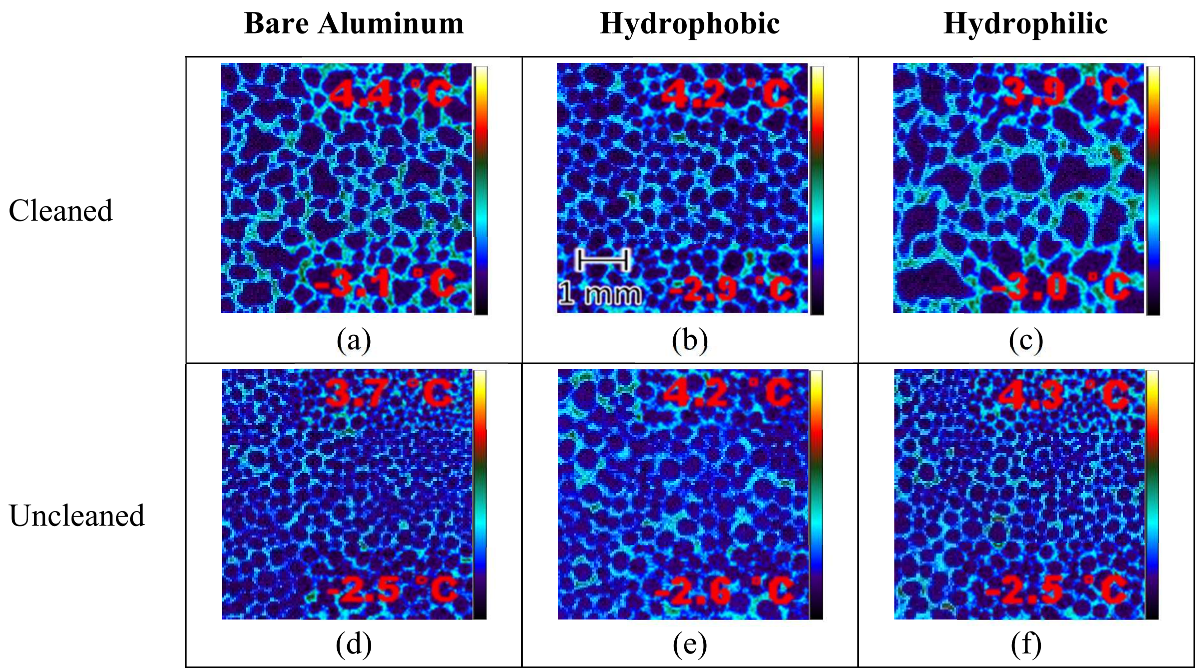
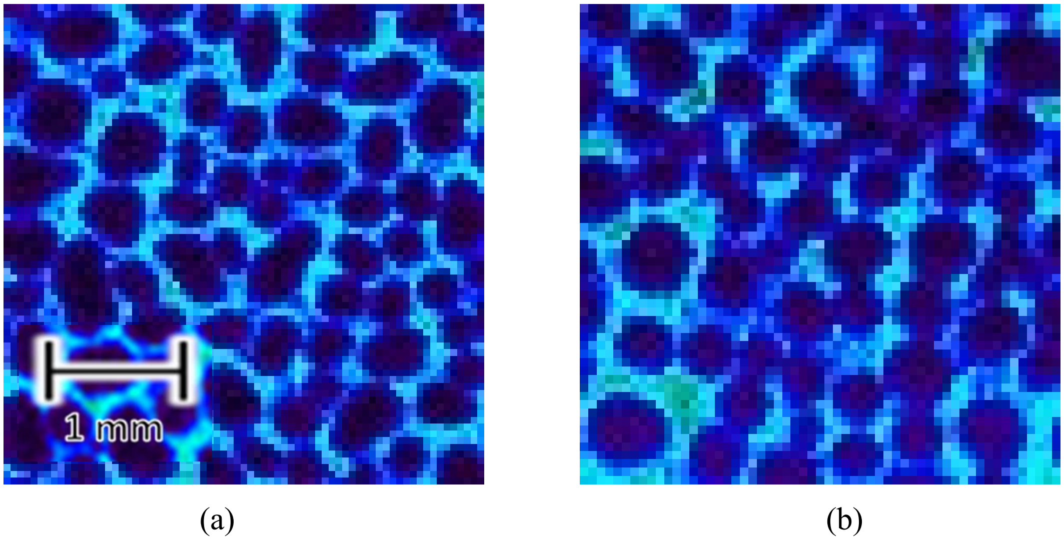


Figure 3.20: Images of droplets after 8 minutes of testing on each surface, cleaned and uncleaned, for test conditions 2

Figure 3.21 is presented to help explain why the uncleaned surfaces had longer freezing times than the cleaned surfaces. It shows that the droplets on the uncleaned surfaces were always much rounder than even droplets on the cleaned hydrophobic surface. Furthermore, after droplets coalesced on the uncleaned surfaces, the resulting droplet shrank quickly into a circular droplet whose shape was nearly unaffected by the original droplets' locations. In contrast, the droplets on the cleaned hydrophobic surface were smooth and rounded but not as circular as those on the uncleaned surfaces. This observation resulted from more “pinning” on the surface; when droplets coalesced, the resulting droplet was often a smooth and rounded but somewhat irregular shape that was loosely dependent on the original droplets’ sizes and locations. According to Chu et al. (2017) and Cha et al. (2020), pinning and irregular droplet shapes after coalescence are due primarily to large contact angle hysteresis (CAH). The larger the CAH, the more resistance there is for a droplet's triple line to shrink back to a circular state. Based on these observations of droplet shape on the cleaned and uncleaned surfaces, the contaminants on the uncleaned surfaces did not provide more nucleation sites but rather seemed to cover the surfaces with a film. This film smoothed surface imperfections and suppressed the effects of static contact angle on droplet size as well as the effects of contact angle hysteresis on droplet shape. Since heterogeneous nucleation is enhanced in the presence of surface inhomogeneity, smoothing surface imperfections would also delay the onset of droplet freezing by forming ice nuclei at the solid/liquid interface. This conclusion explains that the freezing time results for the uncleaned surfaces still responded to changes in test conditions even though freezing times were longer and had little dependence on surface wettability.



(a)

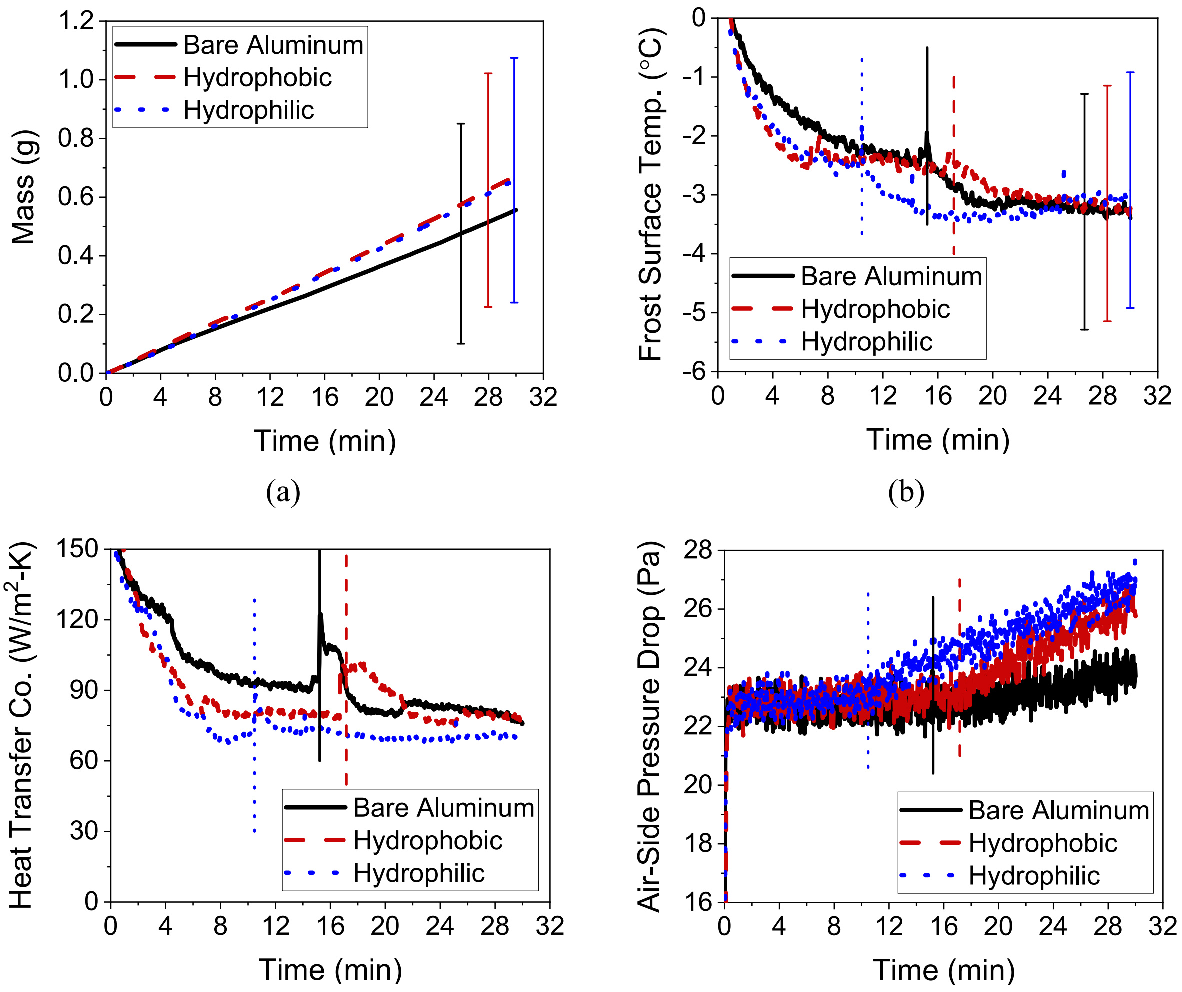
(b)

Figure 3.21: Examples of droplet shapes on the (a) cleaned and (b) uncleaned hydrophobic surfaces for the High ΔT test conditions.

3.7 Characteristics of Frost Growth after Initial Nucleation

Figures 3.22 – 3.27 present frost characteristics for the overall frost layer after initial nucleation. For all these figures, the profiles presented represent individual test runs and are not averages of multiple tests. The error bars in Figure 3.22(a) represent the frost mass variation as the difference between the inlet and outlet dew point temperatures changed by ±0.5 °F. This change in dew point temperature is within a reasonable range since, according to Table 3.1, the maximum uncertainty on the individual dew point temperature measurements was ±0.5 °F. The error bars in Figure 3.22(b) represent the accuracy uncertainty of using the IR camera to measure the frost surface temperature. Figure 3.22 presents the frost mass, frost surface temperature, air-side heat transfer coefficient, and airside pressure drop for test conditions 1 on the bare aluminum surface as an example. Figure 3.22(a) shows that the accumulated frost mass after 30 minutes for the hydrophobic and hydrophilic surfaces was nearly the same, while the mass accumulated for the bare aluminum surface was significantly less. Figure 3.22(b) shows that the frost surface temperatures were very similar for all three surfaces, especially after 20 minutes into each test. Early in the test, the temperature for the bare aluminum surface was higher than that for the hydrophilic surface; this indicated better conduction through the frost layer early into frosting for the hydrophilic surface, and thus a higher density of frost. Figure 3.22(c) shows the air-side heat transfer coefficient. Interestingly, both the frost surface temperature and the heat transfer coefficient for each surface spiked briefly when the droplets on the surface froze. Based on these spikes, it is evident that freezing occurred at around 10, 15, and 17 minutes for the hydrophilic, bare aluminum, and hydrophobic surfaces, respectively. Lastly, air side pressure drop is depicted in Figure 3.22(d). The figure shows that the hydrophilic surface had the highest pressure drop, while the bare aluminum surface had the lowest. The droplets' freezing time could also be determined from the pressure drop profiles; when freezing occurred, there was a marked increase in each profile's slope. In each of the plots below, freezing time is indicated for each surface by a dashed line.

Figure 3.23 presents frost thickness and frost density for experimental runs on all three surface types at test conditions 1. The error bars on the density measurements result from varying frost mass according to its uncertainty (as discussed above) and recalculating density. The error bars on frost thickness represent the uncertainty associated with interpreting the CCD video scope's images. Interestingly, even though freezing occurred at very different times for the different surfaces, the frost thickness values in Figure 3.23(a) for all three were very close from 20 to 30 minutes into the tests, with that for the hydrophobic surface being slightly higher at the end. In Figure 3.23(b), frost density for each surface is presented. Density was very high at the beginning of the tests and decreased as the frost crystals grew on the droplets. The highest density values were neglected in this plot so the focus could remain on the differences between the surfaces once density reached its minimum and began to increase slowly. The density values were highest for the hydrophilic surface and lowest for the bare aluminum surface, with those for the hydrophobic surface settling in between the other two. However, all density values were close after 22 minutes into the test and within the experimental uncertainty from each other. These results confirm the idea postulated earlier that the lower frost surface temperature observed for the hydrophilic surface was due to higher density and, therefore, more efficient heat conduction through the frost layer.



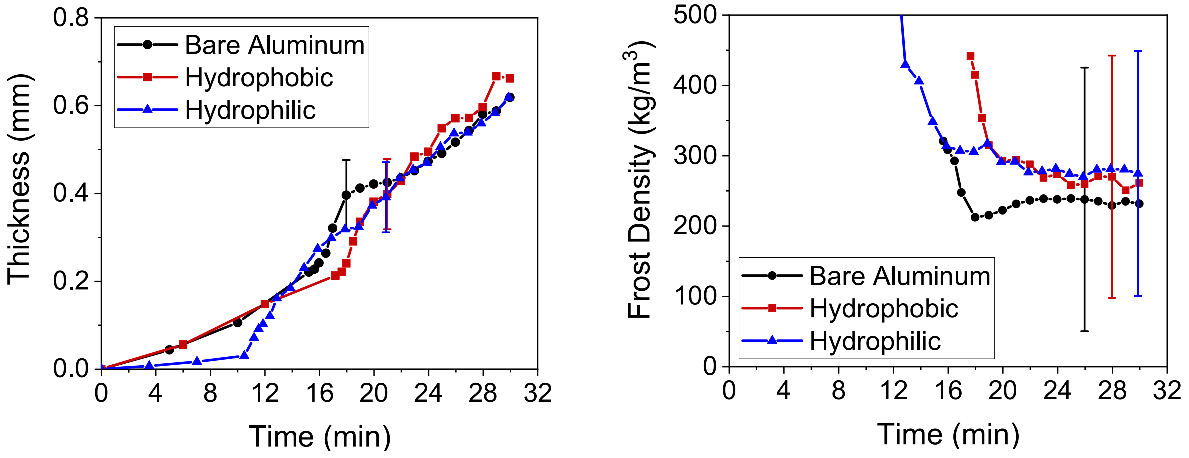
(a)

(b)

(c)

(d)

Figure 3.22: Frost layer characteristics for a test on the bare aluminum surface at test conditions 1. (a) Frost mass vs. time, (b) frost surface temperature vs. time, (c) heat transfer coefficient vs. time, and (d) air side pressure drop vs. time.



(a) (b)

Figure 3.23: (a) Frost thickness vs. time and (b) frost density vs. time for all three surface types at test conditions 1

Figure 3.24 presents frost thickness and density for test conditions 2, where the humidity of the bulk airstream was increased from the baseline. For this case, the frost thickness on the bare aluminum and hydrophobic surfaces increased significantly from that for test conditions 1, while the thickness on the hydrophilic surface remained nearly the same. The increase in humidity caused a decrease in density for the bare aluminum surface and a considerable increase in density for the hydrophilic surface, while that for the hydrophobic surface did not change much. The most significant difference between the plots for test conditions 1 and 2 is the stratification of density values as humidity increased; instead of being very similar, density values for the three surfaces spanned a range of over 200 kg/m^3 for test conditions 2.

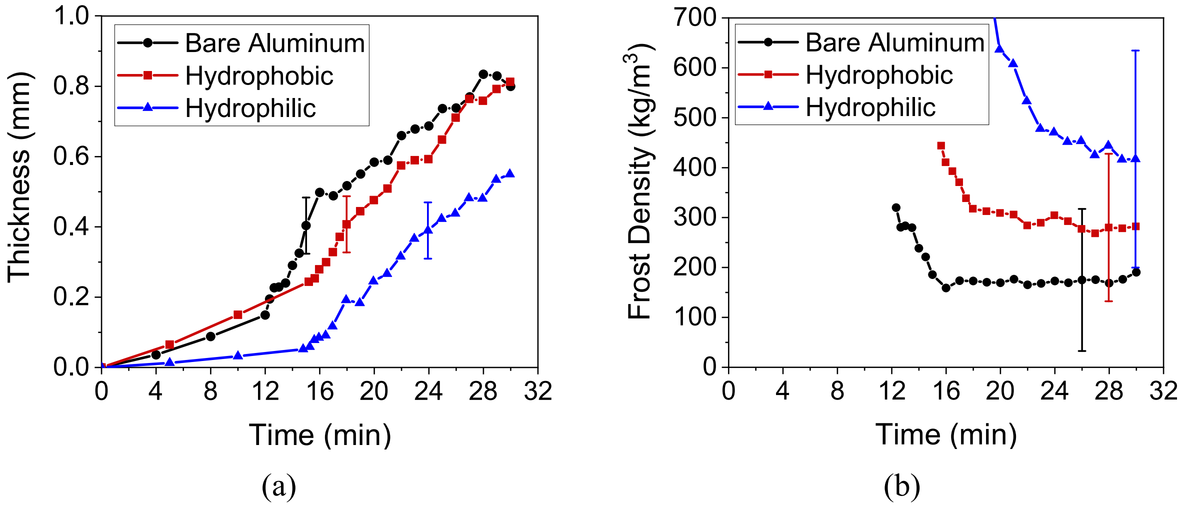
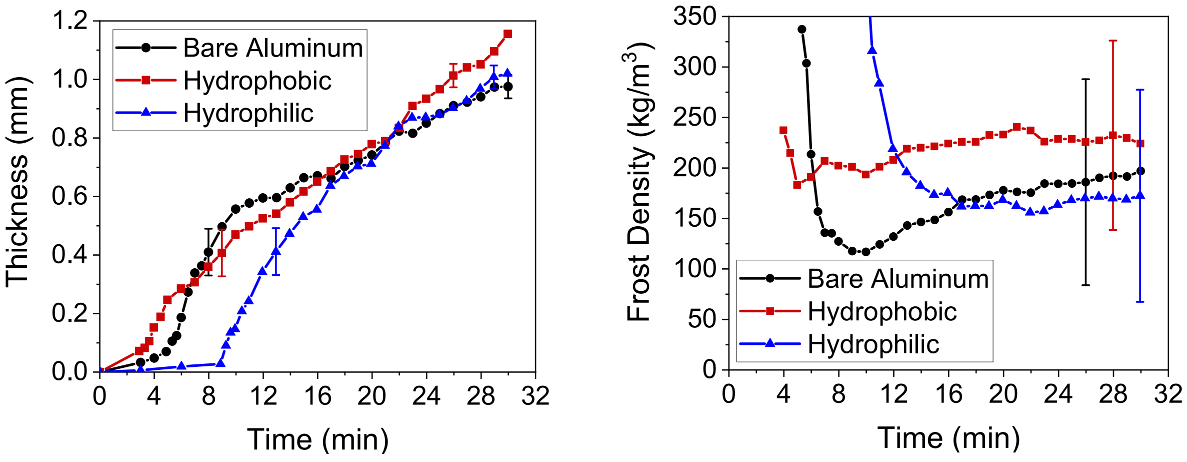


Figure 3.24: (a) Frost thickness vs. time and (b) frost density vs. time for all three surface types at test conditions 2

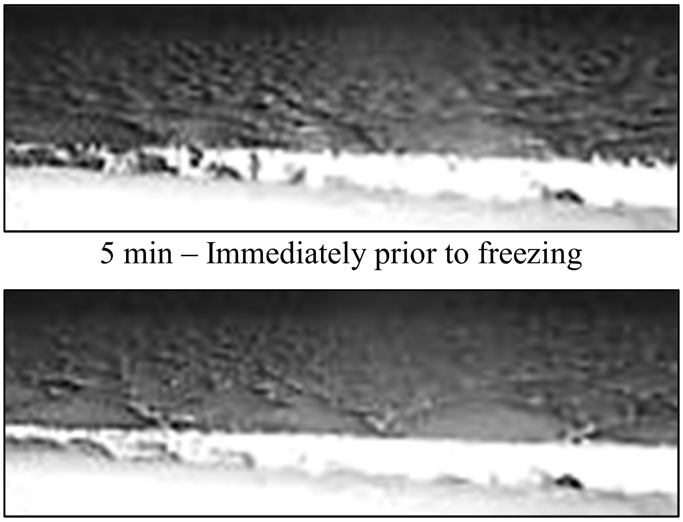
Figure 3.25 presents frost thickness and density for test conditions 3, where test plate temperature decreased from the baseline. For this case, the bare aluminum and hydrophobic surfaces had larger frost thicknesses than the hydrophilic surface for the first half of the test. However, the three had very similar thicknesses during the second half. Both surfaces had a much larger frost thickness after 30 minutes than they did for test conditions 1. Similarly to test conditions 1, the hydrophobic surface had a slightly larger frost thickness than the other two after 30 minutes. The density plots for test conditions 3 differed from those for test conditions 1 and 2 in that the density of the hydrophobic surface, not the hydrophilic surface, was the largest once it began increasing. All three surfaces had lower density after 30 minutes than they did for test conditions 1, however, and the values for each surface were not significantly different from each other.



(a) (b)

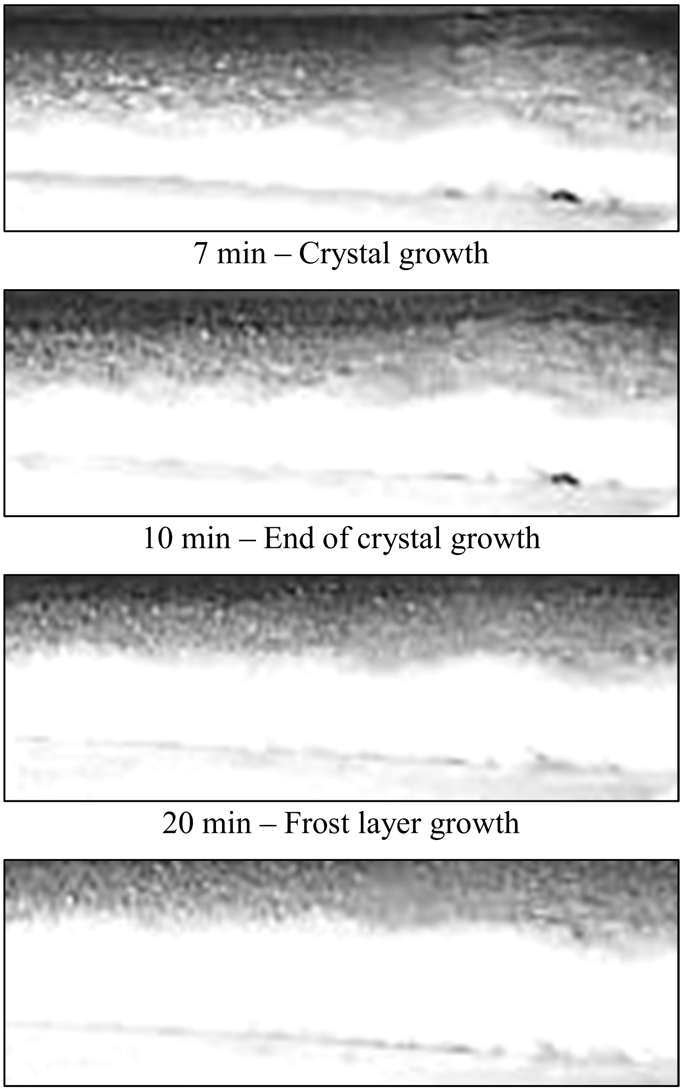
Figure 3.25: (a) Frost thickness vs. time and (b) frost density vs. time for all three surface types at test conditions 3

Figure 3.26 presents time-lapse images of frost growth after droplet freezing on the bare aluminum surface at test conditions 3. This figure is presented as a visual example of the frost growth process and corresponds to the hydrophobic surface data in Figure 3.25. The camera resolution was not high enough to detect individual crystals on the frozen droplets, but these images were useful for measuring the leading edge's average frost thickness.



5 min – Immediately prior to freezing

5.3 min – Freezing



7 min – Crystal growth

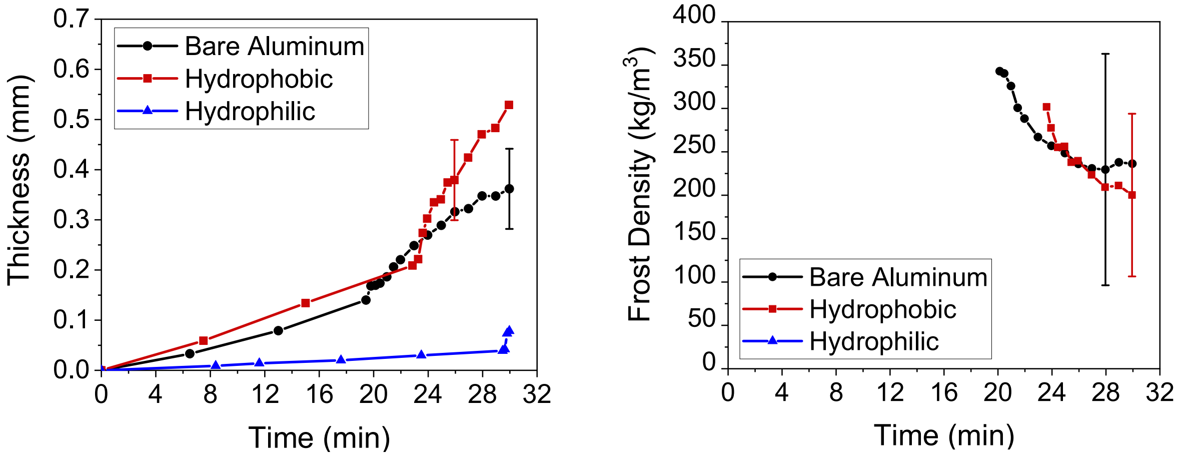
10 min – End of crystal growth

20 min – Frost layer growth

30 min – Frost layer growth

Figure 3.26: Time-lapse images of frost growth on the bare aluminum surface at test conditions 3

Figure 3.27 presents frost thickness and density for test conditions 5, where the air velocity was decreased from the baseline. Frost thickness was very low on all three surfaces for these test conditions, partially due to the very long freezing times and partially because of the low slope in the frost thickness profiles even immediately following freezing. Interestingly, there is significant stratification in the thickness plot. The hydrophobic surface had the largest thickness, and the hydrophilic had the smallest. Also, because the water on the hydrophilic surface was in liquid form for almost this entire test run, the density values for this surface were too great to fit within the plot range in Figure 3.27(b).



(a) (b)

Figure 3.27: (a) Frost thickness vs. time and (b) frost density vs. time for all three surface types at test conditions 5

4. Model for Droplet Growth and Freezing Stage

Based on the observations described in the literature review in Chapter 2, there is a need for one more comprehensive droplet model that accounts for both coalescence effects and different surface types under condensation frosting conditions typical of heat pump systems. This portion of the research fills this gap by combining modeling methods from both statistical physics and classical dropwise condensation theory into a new semi-empirical model for droplet growth and freezing characteristics. The current approach focused on surface temperatures a few degrees below the water freezing temperature. However, the basic principles presented in this paper could be extended to lower surface temperatures, providing that the frosting process began with condensation rather than ablimation. The droplet coalescence effects were significant only when the surface temperature was near and below the freezing point. At large sub-freezing temperatures, droplet coalescence might still be present, but its effects were minor due to the inherent fast freezing process.

4.1 Model Development

The model presented in this chapter consisted of a semi-empirical method for calculating droplet growth and freezing characteristics on flat plates at subfreezing temperatures under air-forced convection conditions. It employed classical nucleation and dropwise condensation theories along with statistical methods and new empirical correlations to predict droplet size and coverage throughout the condensation period leading up to freezing, as well as the time it takes for the droplets to freeze. The model did not require an initial droplet size or distribution, and the inputs to the model were the humid air conditions (temperature, pressure, relative humidity, and velocity), the characteristics of the cold surface (temperature, contact angle, length, and width), and the dimensions of the airflow channel. This model was implemented as a simulation tool in the C++ language. The main assumptions of the model are listed below:

(a) All droplets on the cold surface were a uniform average size at any given point during a simulation.

(b) Freezing of all droplets was instantaneous.

(c) Humid air was saturated at the cold surface.

(d) Droplets were very small compared to the cold surface's characteristic length, so they did not significantly alter the heat transfer geometry.

It is worth noting that when considering the symmetry line along the horizontal direction at the top of the plate, the flow-channel configuration investigated in the present paper represented the local center section of a heat exchanger finned geometry. Figure 4.1 illustrates this concept. The top symmetry line is the midpoint distance between two parallel fins of a microchannel type multi-folded fins heat exchanger of direct expansion air-source heat pump systems. Because the vertical microchannel tubes have smaller widths than the horizontal fins and are at temperatures close to their base fins' temperatures, transverse temperature gradients between the fins and the adjacent tubes are small. At the two sides of the plate, symmetry boundary conditions can be reasonably applied if the plate's width is much larger than its height, i.e., minor edge effects occur. This situation for microchannel type air-to-refrigerant heat exchanger finned structures was replicated in the experiments and the model of the present work.

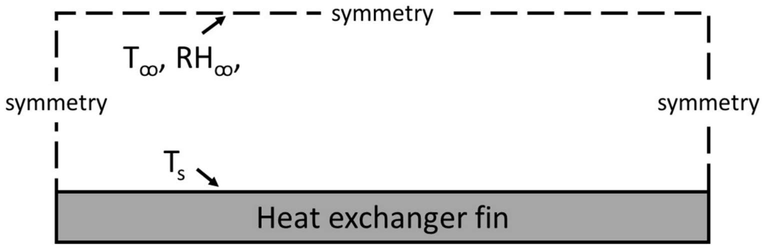


Figure 4.1: Illustration of how the flat plate model applies to heat exchanger fin geometry.

4.1.1 Fluid Property Calculations

Humid Air Properties

The model utilized the CoolProp library (Bell et al., 2014) to calculate the humid air's psychrometric properties. The dry-bulb temperature, pressure, and relative humidity of the air are the required inputs. The outputs are density, dynamic viscosity, thermal conductivity, specific heat capacity, Prandtl number, absolute humidity ratio, and partial pressure of the water vapor in the humid air. This same function was used to calculate humid air properties of the bulk air stream, at the cold surface, and at the film temperature as necessary. Using these humid air properties, the model calculated the humid air's vapor density using the ideal gas law for water vapor. A non-dimensional supersaturation degree (SSD) of the humid air at the cold surface (or at the frost surface, when applicable) with respect to the bulk humid air was calculated using Equation 4.1. In this equation, P_v represents the partial pressure of the water vapor in the bulk air stream, while $P_{v,sat,s}$ is the partial pressure at the cold surface temperature (or the frost surface temperature) under saturation conditions.

$$SSD = \frac{\left(P_v - P_{v,sat,s}\right)}{P_{v,sat,s}} \tag{4.1}$$

Assumption (c) above stated that humid air was saturated at the cold surface. This assumption was widely adopted for various frost models, but it has also been contested. In particular, Na and Webb (2004) investigated supersaturation at the frost surface and reported that the humid air could not be saturated at the frost surface but rather must be supersaturated. They presented an empirical correlation to calculate supersaturation at the frost surface. This correlation is presented as Equation 4.2 below. Note that Equation 4.2 represents supersaturation at the frost surface's particular location and not between the bulk humid air stream and the cold/frost surface as in Equation 4.1.

$$SSD_{fs} = 0.808\left(\frac{P_{v,\infty}}{P_{v,sat,\infty}}\right)\left(\frac{P_{v,sat,fs}}{P_{v,sat,\infty}}\right)^{-0.657} - 1 \tag{4.2}$$

Equation 4.2 was utilized to calculate supersaturation for test conditions 1 – 5 of the current study, using the cold plate temperature as the frost surface temperature. It predicted supersaturation values very close to, and sometimes below 0, for these conditions. The value of supersaturation decreased as the cold temperature increased, so it would be expected to continue to decrease as the frost layer grew. Since negative values of supersaturation are not physically feasible for frost growth, and yet frost growth occurred, Equation 4.2 was not utilized in the present model, which focused on high surface temperatures typical of heat pump operation. Additionally, since a supersaturation value of 0 indicated saturation and the values of supersaturation for test conditions 1 – 5 fluctuated around 0, the humid air was assumed to be saturated at the cold plate and frost surface.

Condensate Properties

The model utilized correlations from literature sources to calculate the density, surface tension, and thermal conductivity of liquid water for temperatures at or below 0 °C.

The correlation for density came from Hare and Sorenson (1987), where T is the water temperature in °C, and ρ_w is the water's density in g/mL. This correlation is presented below as Equation 4.3, with the parameter values given in Table 4.1.

$$\rho_w = a_0 + a_1\,T + a_2\,T^2 + a_3\,T^3 + a_4\,T^4 + a_5\,T^5 + a_6\,T^6 \tag{4.3}$$

Table 4.1: Parameter values for Equation 4.3

a_0	0.99986
a_1	6.690 x 10^{-5}
a_2	-8.486 x 10^{-6}
a_3	1.518 x 10^{-7}
a_4	-6.9484 x 10^{-9}
a_5	-3.6449 x 10^{-10}
a_6	-7.497 x 10^{-12}

The correlation that the model used for surface tension was the IAPWS (International Association for the Properties of Water and Steam) correlation presented in Vinš et al. (2015). This correlation was only applicable for temperatures as low as 0 °C, so the model set the droplet temperature to 0 °C for the calculation of surface tension of subcooled liquid droplets. The correlation is presented below in Equations 4.4 and 4.5, where the surface tension σ_w has units of mN/m and the liquid water temperature T has units of °C. Constants used in this correlation are given in Table 4.2.

$$\sigma_w = B\,\tau^{\varphi}\,(1 + b\tau) \tag{4.4}$$

where

$$\tau = \frac{(T_c - T)}{(T_c + 273.15)} \tag{4.5}$$

Table 4.2: Values of the constants in Equations 4.4 and 4.5

B	235.8 mN/m
φ	1.256
b	-0.625
T_c	373.946 °C

The equation used to calculate droplet thermal conductivity at subcooled temperatures was Eyring's formula as presented in Biddle et al. (2013) and is given below as Equation 4.6. Biddle et al. showed that this formula matched available experimental data with water temperatures as low as 250 K. In Equation 4.6, *k* is the thermal conductivity, *k_B* is the Boltzmann constant, *v* is the molecular volume, *ρ* is the density, and *K_T* is the isothermal compressibility.

$$k = 2.8\, k_B\, v^{-2/3} \left(\frac{1}{\rho\, K_T}\right)^{1/2} \tag{4.6}$$

The Boltzmann constant, molecular volume, and density were not difficult to determine for use in Equation 4.6, but the isothermal compressibility required additional research. The expression chosen for use with Equation 4.6 was a correlation developed by Speedy and Angell (1976), which could be used for water temperatures down to about -25 °C. This correlation is given as Equations 4.7 and 4.8, where K_T has units of 1/bar and T is in K. The constants used in these equations are given in Table 4.3.

$$K_T = A\, \varepsilon^{-\gamma} \tag{4.7}$$

where

$$\varepsilon = \frac{T}{T_s} - 1 \tag{4.8}$$

Table 4.3: Constants used in Equations 4.7 and 4.8

A	29.65 x 10^{-6} bar^{-1}
γ	0.349
T_s	228 K

Latent Heat of Condensation

The latent heat of condensation was calculated using a correlation from Henderson-Sellers (1984). This correlation is applicable for water temperatures ranging from 0 °C to 50 °C, so the model assumed a water temperature of 0 °C for cases where the droplets are subcooled. The correlation is presented below as Equation 4.9, where H_{fg} has units of J/kg and T is in K.

$$H_{fg} = 1.91846 \times 10^6 \left(\frac{T}{T - 33.91}\right)^2 \tag{4.9}$$

4.1.2 Initial Droplet Size

The droplet model began by calculating the minimum critical radius for the onset of condensation, which was set as the model's initial condensate droplet radius. This minimum critical radius was the smallest length scale that controlled the droplet growth model and was on the order of a few nanometers. Data were not available for how nanoscale features on relatively smooth surfaces such as those investigated in the current study affected droplet nucleation and initial distribution. Because of this, the model set the smallest length scale according to the critical radius for condensation and not according to minute surface features. According to classical nucleation theory, the critical droplet radius for the onset of condensation nucleation has been taken to be the minimum radius of curvature possible for a given wall subcooling (Glicksman and Hunt Jr., 1972). This approach was used in several theoretical studies, particularly those investigating dropwise condensation in pure vapor with no noncondensable gases (Burnside and Hadi, 1999; Abu-Orabi, 1998; Vemuri and Kim, 2006, Graham and Griffith, 1973). Recently, a few researchers took a closer look at this minimum radius and proposed new equations that accounted for condensation out of moist air (Piucco et al., 2008; Yang et al., 2014; Liu and Cheng, 2015). The investigation into critical droplet size for the current study was performed using two correlations: one from Piucco et al. (2008) and one from Yang et al. (2014). Unfortunately, the set of equations presented by Yang et al. (2014) was not thoroughly explained. Thus, the simpler equation from Piucco et al. (2008) was used in the current study's model. Fortunately, we verified Piucco et al.'s approach against the results presented by Yang et al. and noticed that both models provided similar results when considering their asymptotic values. The model from Piucco et al. used for calculations is summarized in Equation 4.10, where σ, ρ, and T are the surface tension, density, and temperature of the liquid droplet. For the calculations of critical radius, it was assumed that the liquid droplet's temperature was equal to the cold test plate's temperature.

$$r_{crit\ (Piucco\ et\ al.)} = \frac{2\sigma}{\rho RT \ln\left(\frac{\omega_\infty}{\omega_{sat,d}}\right)} \tag{4.10}$$

Equation 4.10 calculates the critical droplet radius of curvature for the onset of condensation nucleation. The projected droplet radius, which was the radius of the equivalent circle projected from a droplet to the flat surface underneath it, was calculated as well. For contact angles equal to 90° or above, the projected radius was equal to the radius of curvature. For contact angles less than 90°, the projected radius was calculated using Equation 4.11.

$$r_{proj} = r\,\sqrt{(1-\cos\theta)(1+\cos\theta)} \tag{4.11}$$

4.1.3 Heat Transfer through a Droplet

Heat transfer from the humid air stream to a droplet consisted of both sensible and latent parts and was calculated according to Equation 4.12. In this equation, T_i represents the interface temperature between the droplet and humid air. It was initially assigned a guess value and then determined through successive iterations as described next. Since mass transfer only occurred from the humid air to the droplets, which comprised the wetted area of the surface, the mass transfer term was modified by the *AC* variable, which represents the projected surface area coverage fraction of water on the cold surface. This surface area coverage was calculated using an empirical correlation as described later in Section 4.1.5.

$$\dot{Q}_d = \pi\, {r_{proj}}^2 \left[h\,(T_\infty - T_i) + \frac{h_m}{AC}\,\left(\rho_{v,\infty} - \rho_{v,i}\right) H_{fg}\right] \tag{4.12}$$

Next, the Nusselt number was calculated according to the flow characteristics. If the Reynolds number calculated using the air channel hydraulic diameter was less than 2300, the Nusselt number was calculated using Equation 4.13, which is the average Nusselt number for laminar flow over a flat plate.

$$Nu_{forced} = 0.664\, {Re_L}^{1/2}\, Pr^{1/3} \tag{4.13}$$

External flow was assumed here for several reasons. First, the cooled surface in most frosting experiments, performed in the current study and described in the literature, was only on the bottom of the duct instead of the whole duct being cooled to a uniform temperature. Second, calculations were performed for the thickness of the thermal boundary layer for the tests conducted during this study, and the results revealed that the boundary layers never reached the top of the channel. These results indicated that the isothermal top wall of the channel had a negligible effect on the cold surface’s heat transfer, so the flow regime for heat transfer calculations could reasonably be assumed to behave as external flow. If internal, thermally developing laminar flow was assumed, heat transfer coefficients were slightly too low for the current study's small-channel data. However, for data from other studies where the flow channel was larger, assuming internal flow provided heat transfer coefficient predictions that were much too small. The model kept the assumption of external flow over a flat plate to be more widely applicable, with the understanding that as the channel height decreased, the internal flow assumption produced more and more reasonable results. When the channel grew small enough for the top wall to interfere with the cold plate’s thermal boundary layer, external flow could no longer be assumed. Additionally, the flow itself was always assumed to be hydrodynamically fully developed because the experimental facilities were designed to generate fully developed laminar flow through their test sections. The current study's test section used a gradual converging duct to minimize flow disruption before the test section, thus maintaining fully developed laminar flow. Other studies used their own methods to preserve fully developed flow over the test plates.

Due to the very low air velocities often utilized in early frost growth experiments, it was judicious to include natural convection effects in the overall Nusselt number calculations. The expressions used to calculate the natural convection Nusselt number for horizontal surfaces used in this model came from Rohsenow et al. (1998) and are reproduced in Equations 4.14 and 4.15.

$$Nu_{free} = \frac{2.5}{\ln\left(1 + \frac{2.5}{Nu_T}\right)} \tag{4.14}$$

where

$$Nu_T = \frac{0.527\ Ra^{1/5}}{\left[1 + \left(\frac{1.9}{Pr}\right)^{9/10}\right]^{2/9}} \tag{4.15}$$

Equation 4.16, from Churchill and Chu (1975), was used for vertical surfaces to calculate the Nusselt number for natural convection.

$$Nu_{free} = 0.68 + \frac{0.670\ Ra^{0.25}}{\left[1 + \left(\frac{0.492}{Pr}\right)^{9/16}\right]^{4/9}} \tag{4.16}$$

Equation 4.17, from Cengel and Ghajar (2011), was then used to combine the effects of forced and natural convection into a single overall Nusselt number.

$$Nu = \left({Nu_{forced}}^4 + {Nu_{free}}^4\right)^{1/4} \tag{4.17}$$

If the Reynolds number was calculated to be above 2300, the flow was considered turbulent. For these cases, the Nusselt number was calculated using a method from Rohsenow et al. (1998) for thermally developing turbulent flow. These equations are presented as Equations 4.18 and 4.19 below, where Nu_∞ refers to the fully developed Nusselt number. For this model, the Dittus-Boelter equation was used for fully developed flow because of its applicability to low turbulent Reynolds numbers, as often encountered in frosting studies.

$$\frac{Nu}{Nu_\infty} = 1 + \frac{C}{x/D_h} \tag{4.18}$$

$$C = \frac{(x/D_h)^{0.1}}{Pr^{1/6}} \left(0.68 + \frac{3000}{Re^{0.81}}\right) \tag{4.19}$$

The heat transfer coefficient was then calculated according to Equation 4.20.

$$h = \frac{Nu\, k}{L} \tag{4.20}$$

The mass transfer coefficient was calculated using the Chilton-Colburn Analogy, as shown in Equation 4.21.

$$h_m = \frac{h}{\rho\, c_p\, Le^{2/3}} \tag{4.21}$$

The Lewis number has often been assumed to be equal to 1 for convective airflow. However, to achieve greater accuracy, the model calculated the Lewis number according to Equation 4.22, where D_{AB} is the mass diffusivity.

$$Le = \frac{k}{\rho\, c_p\, D_{AB}} \tag{4.22}$$

The mass diffusivity in Equation 4.22 was calculated using Equation 4.23, which was presented in Marrero and Mason (1972). This equation assumes a pressure of 1 atm, the units for T are K, and the units for D_{AB} are atm/cm^2-s.

$$D_{AB} = (1.87 \times 10^{-6})\, T^{2.072} \tag{4.23}$$

Heat transfer through a droplet was calculated using classical dropwise condensation theory to obtain the interface temperature, T_i. This theory assumes that the temperature difference between the water vapor directly above the surface of a droplet and the cold surface can be expressed as a sum of temperature differences associated with all the thermal resistances through the droplet. The most common and significant thermal resistances used in droplet condensation models were those due to conduction through the droplet, the curvature of the droplet, and mass transfer across the droplet interface. The same three resistances were used in this model, but the traditional interfacial resistance was replaced with resistance due to convection. Altering this resistance accounted for the difference between the pure vapor condensation models of classical dropwise condensation theory and the humid air condensation that initiates outdoor frosting. Traditional equations for droplet resistances assumed hemispherical droplets (as in Graham, 1969). However, more recently, researchers have proposed new equations for some of these resistances that depend on the droplet's contact angle. Kim and Kim (2011) proposed an equation for the temperature difference representing the conduction resistance through a droplet; this equation was developed by integrating over isothermal surfaces within a droplet from the test plate's surface up to the droplet surface. This equation was implemented in the model of the current study and is presented below as Equation 4.24.

$$\Delta T_{cond} = \frac{\dot{Q}_d \, \theta}{4 \, \pi \, r \, k_d \, \sin(\theta)} \tag{4.24}$$

The equation for the temperature difference that represented the curvature resistance came from a combination of the Kelvin-Helmholtz equation and the Clausius-Clapyron equation, along with the assumption that water vapor could be treated as an ideal gas. This derivation was presented in Graham (1969), and the final result, presented as Equation 4.25, was implemented in the current model.

$$\Delta T_{curv} = \frac{2 \, T_i \, \sigma}{r \, \rho_d \, H_{fg}} \tag{4.25}$$

The temperature difference that represented the resistance due to convection was set to be equal to the bulk humid air stream's temperature minus the air/water interface temperature, as shown in Equation 4.26.

$$\Delta T_{conv} = T_{\infty} - T_i \tag{4.26}$$

All of the individual temperature differences were added together and set equal to the total temperature difference between the bulk humid air stream and the temperature of the cold surface, as in Equation 4.27.

$$\Delta T_{total} = \Delta T_{cond} + \Delta T_{curv} + \Delta T_{conv} \tag{4.27}$$

After this calculation was performed, the C++ simulation iterated the heat transfer calculations until a value for T_i was obtained that caused the total temperature difference as calculated in Equation 4.27 to be equal to the temperature difference between the bulk humid air and the cold surface.

4.1.4 Droplet Radial Growth

The portion of the droplet heat transfer which contributed to droplet growth was the latent heat transfer. The latent heat transfer on the air side, presented as a time derivative, is given as Equation 4.28.

$$\frac{dQ_d}{dt} = \pi \, {r_{proj}}^2 \left(\frac{h_m}{AC}\right) H_{fg} \left(\rho_{v,\infty} - \rho_{v,i}\right) \tag{4.28}$$

This latent heat transfer rate contributed directly to an increase in the volume of an average-sized droplet. The volume of a hemispherical droplet and its derivative with respect to the radius are given in Equations 4.29 and 4.30.

$$Vol = \frac{2\pi}{3}\, r^3 \tag{4.29}$$

$$d(Vol) = 2\pi r^2\, dr \tag{4.30}$$

Equation 4.31 presents an expression for the latent heat transfer on the droplet side, substituting in the change in volume from Equation 4.30.

$$dQ_d = \rho_d\, H_{fg}\, d(Vol) = 2\pi r^2 \rho_d H_{fg}\, dr \tag{4.31}$$

By equating the two expressions for latent heat transfer from Equations 4.28 and 4.31, recognizing that the projected radius and radius of curvature are equal for a hemispherical droplet, and solving for change in droplet radius, the radial growth rate of a droplet was obtained and is presented as Equation 4.32.

$$\frac{dr}{dt} = \left(\frac{h_m}{AC}\right)\left(\frac{\rho_{v,\infty} - \rho_{v,i}}{2\,\rho_d}\right) \tag{4.32}$$

Integrating Equation 4.32 would indicate that droplet radius was linearly dependent on time. However, based on analyses of droplet growth dynamics from the literature and the current study’s experimental work, it was apparent that while average radial droplet growth was linearly dependent on time for specific situations, the dependence on time was not constant across different droplet growth scenarios. Instead, it could be represented by a power-law growth model as expressed in Equation 4.33. In this equation, μ accounts for droplet coalescence effects, and c, equal to the growth rate of a hemispherical droplet as expressed in the right-hand-side of Equation 4.32, accounts for the mass transfer driving potential. The assumption of a hemispherical droplet (i.e., neutral contact angle) for the variable c was made to avoid redundancy in contact angle dependence on the droplet growth expression. All contact angle dependence on the projected droplet radius was encapsulated in the exponent, μ. The controlling length scale for this portion of the droplet model was the change in radius with time, which was on the order of tenths of microns per second.

$$r_{proj,avg} = c\ t^{\mu} \tag{4.33}$$

It was reported in Chapter 3 that the effects of droplet coalescence as expressed by the exponent μ were strongly dependent on the contact angle. Droplets grew more quickly and had larger coalescence rates as the contact angle decreased. Comparisons of data from the current study with data from Tao et al. (1993) indicated that the temperature difference (*ΔT*) between the humid air and the cold surface also had significant effects on the shape of the droplet profile as expressed by μ; μ significantly decreased as *ΔT* increased. An expression for μ was developed using these observations by fitting it to the data for droplet size at freezing from both the current study, which provided data for variable contact angle and low *ΔT*, and from Tao et al. (1993), which provided data for an extensive range of *ΔT* all larger than in the data from the current study. The resulting expression is given as Equation 4.34, where θ is in degrees and *ΔT* is in °C.

$$\mu = 1.645 - 0.0017\ \theta - 0.0168\ \Delta T \tag{4.34}$$

The final expression for droplet size increase is presented as Equation 4.35, where *t* and *Δt* represent the absolute time and the time step used in the simulation, respectively.

$$r_{proj,avg,new} = r_{proj,avg,old} + c\ (t^{\mu} - (t - \Delta t)^{\mu}) \tag{4.35}$$

After the simulation returned the new droplet radius, the absolute time was increased by a single time step. The iteration process then repeated, beginning by taking a guess value for the larger droplet's new interface temperature.

4.1.5 Area Coverage

The projected area coverage of droplets on the surface was an essential parameter in determining average droplet growth. While this quantity was occasionally measured, it was rarely predicted for droplet growth under condensation frosting conditions. Hoke et al. (2000) presented expressions to calculate the surface area coverage for hydrophobic and hydrophilic surfaces dependent on droplet size, contact angle, and the number of droplets on a surface. The expressions used to calculate the number of droplets on a surface were empirical and applicable only to the surfaces used in that particular study, which had contact angles of about 50° and 90°. Equations 4.36 and 4.37 present their expressions for the number of droplets per mm^2 and area coverage, respectively, for the 90° surface. Equations 4.38 and 4.39 give the expressions for the number of droplets per mm^2 and area coverage for the 50° surface.

$$N" = (1.1036 \cdot 10^6)\, {D_v}^{-1.35} \tag{4.36}$$

$$AC = N"\,\pi \left(\frac{D_A}{2}\right)^2 \cos^2(\theta - 90°) \tag{4.37}$$

$$N" = (1.2584 \cdot 10^5)\, \left(\frac{D_v}{\cos(90° - \theta)}\right)^{-1.35} \tag{4.38}$$

$$AC = N"\,\pi \left(\frac{D_A}{2}\right)^2 \cos^2(90° - \theta) \tag{4.39}$$

In Equations 4.36 - 4.39, D_A and D_V represent the surface-area-averaged and volumetrically area-averaged diameters, respectively, and were defined according to Equations 4.40 and 4.41.

$${D_A}^2 = \sum_{i=1}^{N} \frac{{D_i}^2}{N} \tag{4.40}$$

$$D_V{}^3 = \sum_{i=1}^{N} \frac{D_i{}^3}{N} \tag{4.41}$$

These expressions were used with the current simulation tool to predict the data of the current study. To use the equations above, D_A and D_V were first calculated using Equations 4.40 and 4.41 for data from the current study, and those values were used in Equations 4.36 - 4.39. The results of this analysis are presented in Figure 4.2. These expressions could not predict the data, and large over and under-predictions were observed for data with higher and lower contact angles, respectively. These errors were likely because Equations 4.36 and 4.38 were not designed for use with any surface types or experimental conditions other than those investigated by Hoke et al. (2000).

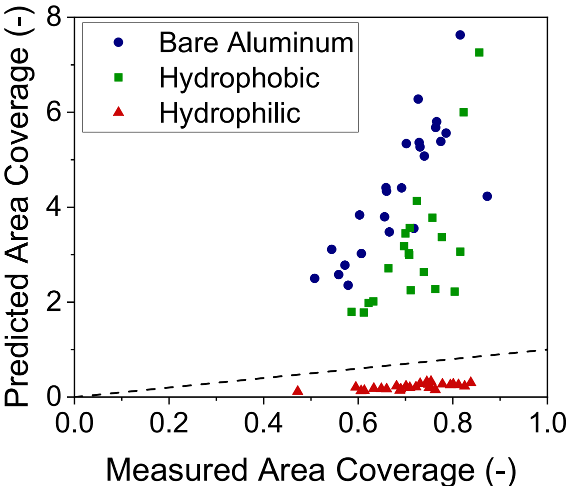


Figure 4.2: Results of using surface area coverage expressions from Hoke et al. (2000) to predict data from the current study

A new, more comprehensive correlation to predict surface area coverage was developed in this study using data from five different sources (current study; Tao et al., 1993; Chu et al., 2017; Haque and Betz, 2018; Li et al., 2010). All data except that from Chu et al. (2017) were taken under condensation frosting conditions. The data from Chu et al. was included because the surface temperature (2 °C) was only slightly above that of freezing, and the droplet sizes in this data set were comparable to those for the condensation frosting tests in the overall data set. The data indicated that surface area coverage was most strongly dependent on the average size of droplets at any given point during a test. It was also dependent on the contact angle, though of secondary magnitude. Based on these observations, a power law was determined to be a good prediction method. Nonlinear regression was performed to determine the fitting parameters. The resulting correlation is Equation 4.42, where r_{proj} is the projected radius in mm and θ is the contact angle in radians. Figure 4.3 shows the accuracy of the new correlation. Equation 4.42 is applicable for 24° < θ < 142°, 0.013 mm < r_{proj} < 1.219 mm, and 0.361 < AC < 0.873.

$$AC = 0.0386 \cdot (21.91 - \theta)\, {r_{proj}}^{0.1043} \tag{4.42}$$

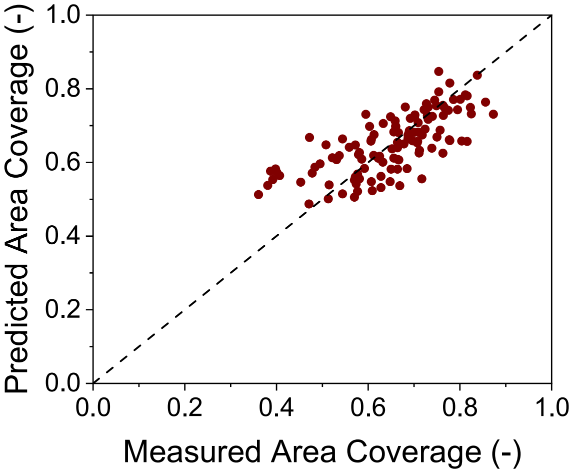
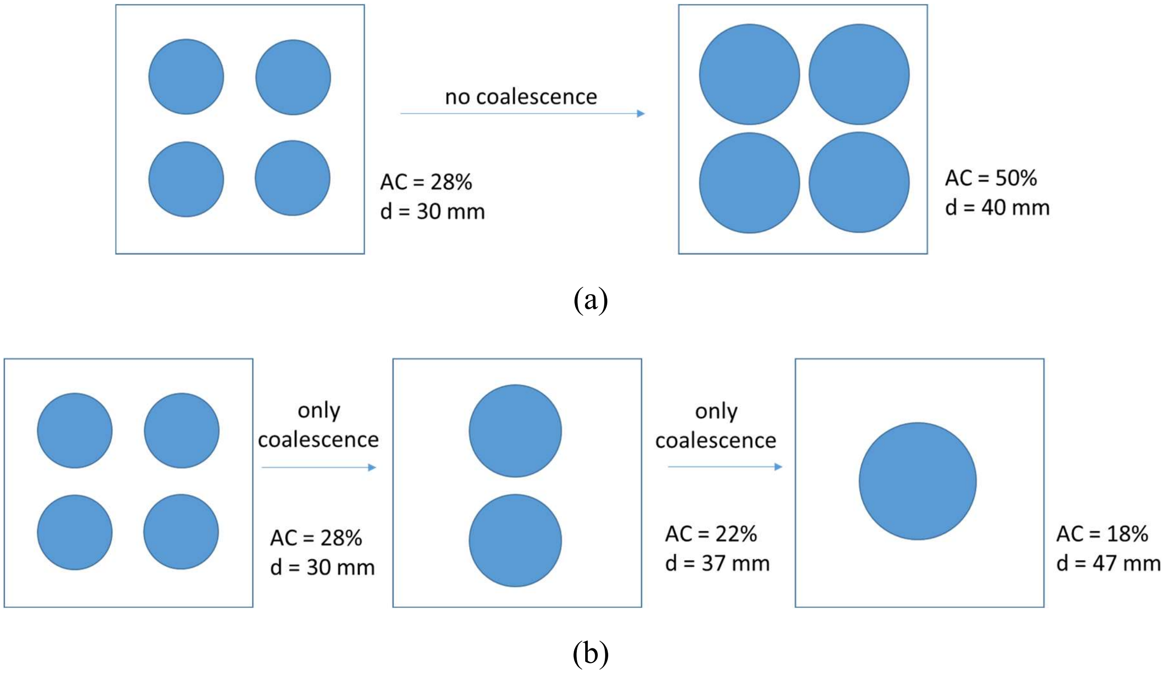


Figure 4.3: Results of Equation 4.42 as a predictor for surface area coverage data

Area coverage was recalculated using Equation 4.42 at each time step during droplet growth. It could not be calculated at the beginning of the simulation using the correlation and subsequently recalculated geometrically. This conclusion was because of the influence of secondary nucleation, which refers to the nucleation of tiny droplets in the spaces left open due to droplet coalescence, on the area coverage. Figure 4.4 further illustrates this point using two case studies where area coverage was calculated for droplets of specified initial diameter interacting on a square surface of 100 mm to a side. Figure 4.4(a) depicts droplets growing by direct condensation with no coalescence and no secondary nucleation. For this case, area coverage would always reach 100% as long as the simulation was sufficiently long enough. However, the time it would take to achieve total area coverage would vary as the mass transfer driving potentials changed. Figure 4.4(b) shows droplets growing purely by coalescing together, with no direct growth or secondary nucleation. For this case, area coverage would only decrease over time. However, experimental data shows that area coverage slowly increases with time and droplet size and does not reach 100%. Coalescence effects dominate, though individual droplet growth and secondary nucleation effects combine to lead to an overall increase in area coverage as tests progress. The interplay of these three effects on area coverage is not well understood, so Equation 4.42 was used at each time step to balance the three factors' contributions.



(a)

(b)

Figure 4.4: Illustration of droplet growth by (a) direct growth only and (b) coalescence only.

4.1.6 Droplet Freezing Time

The freezing time controls how long droplets grow before freezing occurs and thus directly affects the frozen ice-bead diameters and distribution. The frozen droplet size sets the initial frost thickness and density of the ensuing frost layer. As previously stated in Section 2.3.3, the current study only found three existing correlations to predict the point of droplet freezing. The correlation presented by Seki et al. (1985), which related the critical freezing temperature of a droplet to its volume, was not used in the model. Both critical droplet volume and droplet temperature to initiate freezing were unknown, so neither could be used to predict the point of freezing. The two freezing time correlations were used here to predict data for cleaned surfaces from the current study to investigate their accuracy for the surfaces of heat exchanger fins under heat pump systems operating conditions (i.e., lower but near freezing point surface temperatures). Figure 4.5(a) shows the results of using the freezing time correlation from Kim et al. (2016) as expressed in Equation 2.2. Predictions were made for data from the current study and Kim et al. as control data. This correlation could not predict the current study's data and showed over-predictions for the hydrophobic data and under-predictions for the hydrophilic data. The large scatter obtained in this comparison occurred for two primary reasons. The first is that the absolute freezing time predicted by Kim et al.'s correlation was proportional to the squared height of a frozen droplet. Droplets measured during the current study grew very large and so had very large heights on the hydrophobic surface. These large heights led to large freezing time predictions. Over-predictions were especially large at high surface temperatures where the droplets grew a long time before freezing. Kim et al. did not develop this correlation using such high surface temperatures, and their droplets did not grow as large as those in the current study. When the correlation was applied to droplets much larger than those used for correlation development, over-predictions were obtained. Droplet height was also primarily responsible for the under-predictions of the hydrophilic data. Droplets on the current study's hydrophilic surface had very small heights even when the radii were large and led to small freezing time predictions when using Kim et al.'s correlation. The correlation was only applicable for contact angles down to 75°, so it was not designed to account for these small droplet heights. Since freezing time was proportional to the squared height, very small heights produced significant under-predictions. The second reason for the large scatter was that the correlation was directly dependent on surface contact angle as well as on droplet height. The current study's data were negligibly dependent on the contact angle, so this direct dependence in the correlation enhanced the over and under-predictions already discussed. Figure 4.5(b) shows the results of using the correlation form from Bryant (1995) to predict the freezing time for the same data from the current study and Bryant as control data. The current study could not reproduce Bryant's data with the correlation exactly as reported by Bryant, and from his correlation form used regression to re-fit the parameters. Only one parameter changed due to the new regression; this new parameter was then used in place of the original to generate the predictions shown in Figure 4.5(b). The resulting correlation is given below as Equation 4.43.

$$t_f = 41.3\, A^{2.6}\, Nu^{-0.3}\, W^{-0.6}\, {T_g}^{-0.25} \tag{4.43}$$

This correlation could not predict the current study's data, specifically at surface temperatures between -3 and -4 °C. Better predictions were achieved as the surface temperature dropped. The poor predictions were likely because Bryant's correlation was developed only for his data set. The temperature of the cold surface was always at -10 °C, and it appeared that Bryant's correlation was not intended to be used for other surface temperatures.

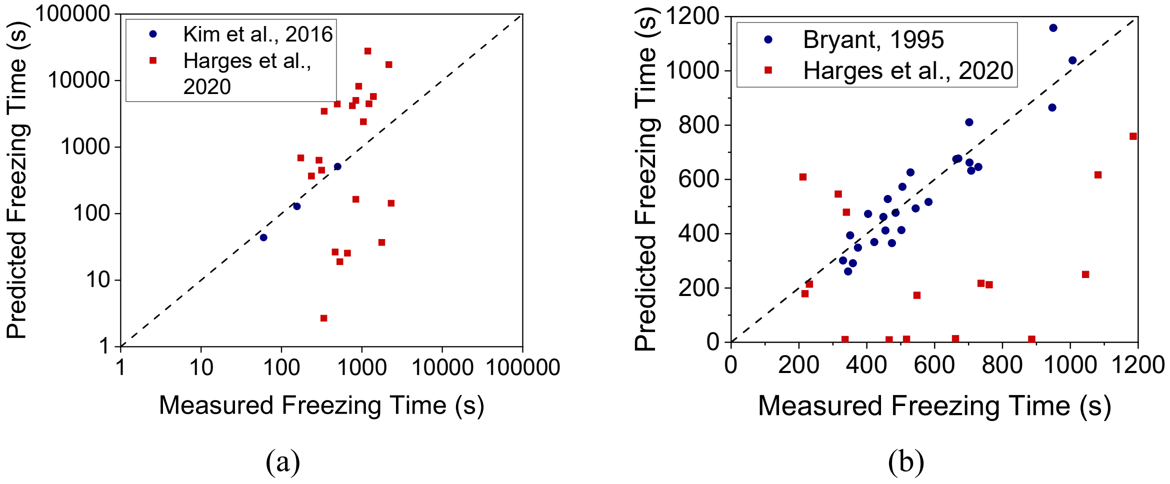


Figure 4.5: Results of using existing freezing time correlations to predict new data for cleaned surfaces from the current study. (a) Kim et al. (2016) correlation, (b) Bryant (1995) correlation.

Based on the results presented in Figure 4.5, it was evident that there was a need for a new freezing time correlation developed using multiple data sets and so applicable for a wide variety of test conditions and surface types. An initial correlation based on literature data was developed as a way to fill this gap. The data set used for developing this correlation consisted of 96 data points from four different sources (Tao et al., 1993; Kim et al., 2016; Hoke et al., 2000; Bryant, 1995). Nonlinear regression was performed on the data, with the resulting equation presented below as Equation 4.44. In this equation, *RH* is a decimal value, T_s and T_∞ have units of K, θ has units of radians, V is in m/s, and *Nu* is dimensionless. Nu was calculated using the current study's simulation tool during correlation development, but all other variables in the data set were gathered directly from the sources.

$$t_f = 0.01476 \cdot RH^{-1.0014} \cdot T_s^{21.406} \cdot T_\infty{}^{-18.616} \cdot \theta^{0.69053} \cdot (V+1)^{1.2681} \cdot Nu^{-1.7619} \qquad (4.44)$$

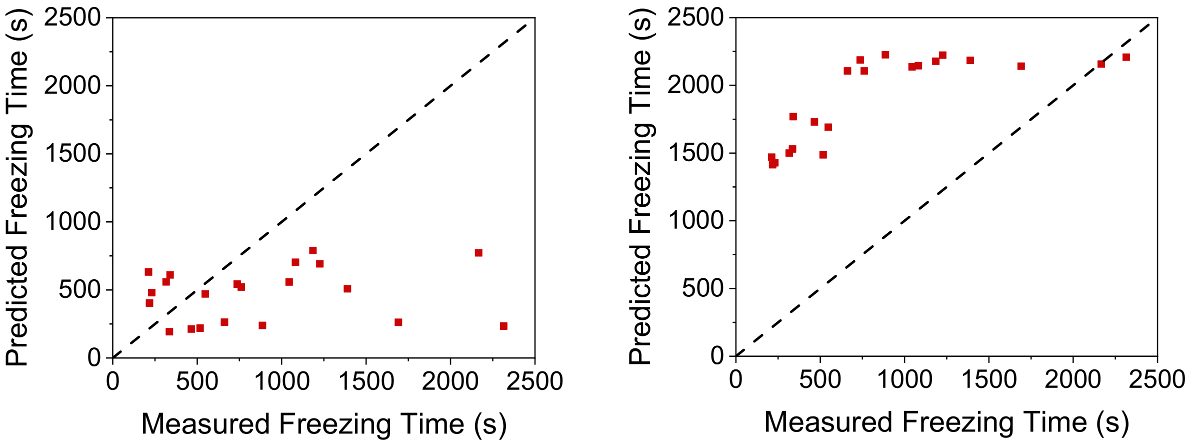
After the first round of data was gathered as part of this research (data for uncleaned surfaces and presented in Adanur, 2019), it was determined that Equation 4.44 could not predict the long freezing times observed in the current study. Furthermore, the overall form of Equation 4.44 did not work for this data set, regardless of what combination of predictor variables was used. Therefore, a new equation was generated using nonlinear regression techniques, this time in the form of a second-order polynomial. The data used to generate this equation included the 96 data points used to generate Equation 4.44 and the new data for uncleaned surfaces. The best fit was given by Equation 4.45 below. In Equation 4.45, *Ja* is the modified Jakob number defined in Hermes (2012), except the latent heat of vaporization was used in place of the latent heat of sublimation in this study. This expression is presented in Equation 4.46.

$$t_f = (2.42 \times 10^{-7}) \cdot Ja^2 + 10.35513 \cdot {T_s}^2 + 371.804 \cdot T_s + 0.28703 \cdot \theta + 3327.787 \tag{4.45}$$

$$Ja = \frac{c_p}{H_{fg}} \frac{(T_{dew} - T_s)}{(\omega_{air} - \omega_{s,sat})} \tag{4.46}$$

Equation 4.45 predicted the new data of uncleaned surfaces better than Equation 4.44 did. However, the fit was still relatively poor, and the investigation into new correlation forms that could better describe freezing for both literature data and the new experimental data of the current study was continued. After the second set of experimental data was gathered (new data for cleaned surfaces presented in Chapter 3 of this thesis and Harges et al., 2020), Equations 4.44 and 4.45 were used to predict average freezing time values for each set of nominal test conditions in Table 3.2 on each of the three surface types. The results are presented in Figure 4.6 below. Figure 4.6(a) shows that Equation 4.44 could not predict the data for high surface temperatures typical of heat pump operation. It failed particularly at predicting data gathered for low mass transfer driving potentials, i.e., at low air velocity and high surface temperature conditions. Figure 4.6(b) shows that Equation 4.45 also failed at predicting this data. Unlike

Equation 4.44, Equation 4.45 could predict a few of the data at low mass transfer driving potentials, but it severely over-predicted the rest of the data.



(a) (b)

Figure 4.6: Results of using (a) Equation 4.44 and (b) 4.45 to predict the new data of cleaned surfaces from Chapter 3.

Based on the results presented in Figure 4.6, it was evident that there was a need for a new correlation explicitly designed for use under heat pump operating conditions and for surfaces that were cleaned before use. A new correlation was developed using data for cleaned surfaces from the current study, data from Bryant (1995), and five data points from Kim et al. (2016), which had relatively high surface temperatures and low supersaturation degrees. It was discovered that freezing time had different dependencies for this subset of experimental conditions than for lower surface temperatures. For high surface temperatures, air velocity and the difference between the triple point and the surface temperature both strongly affected the freezing time, while these variables had minor effects on freezing time for data with low surface temperatures. The reverse was true of contact angle dependence; it was inconclusive for the development of this correlation, while for much of the data with lower surface temperatures, it was apparent that higher contact angles led to longer freezing times. The one variable that had a large effect on freezing time across the entire range of freezing time data was the supersaturation degree (*SSD*) as defined previously in Equation 4.1; as *SSD* increased, freezing time decreased. The new correlation applicable to forced convection data under typical heat pump operating conditions is presented as Equation 4.47.

$$t_f = 15{,}107\ (SSD)^{-0.6026}\ (V+1)^{-1.105}\ \left(T_{tp} - T_s\right)^{-1.224} \tag{4.47}$$

Equation 4.47 is applicable for $0.416 < SSD < 1.659$, 0.3 m/s $< V <$ 4.0 m/s, and -10 °C $< T_s <$ -3.3 °C. The results of Equation 4.47 are presented below in Figure 4.7. There were still prediction errors, but the new correlation predicted better the data relevant to conditions typical of heat pump operation. Additionally, some error was expected due to increased scatter in measured freezing times as surface temperatures increased and mass transfer driving potentials decreased (Hoke et al., 2000).

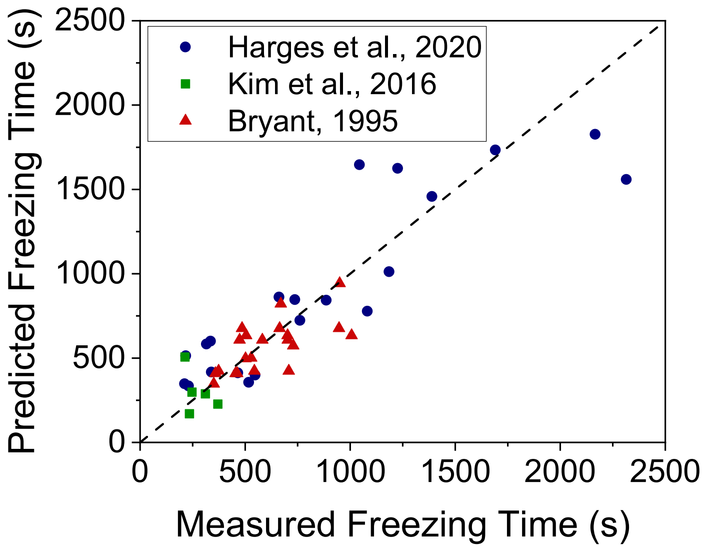


Figure 4.7: Results of using Equation 4.47 to predict freezing time data

4.2 Model Validation

The model's validity and applicability were evaluated by comparing model predictions with measured data of droplet growth and freezing characteristics from several sources. All data were for flat, horizontal surfaces under air-forced convection conditions except for Seki et al. (1985), who employed a vertical flat surface.

4.2.1 Current Study

Model validation was first performed using data for cleaned surfaces gathered during the experimental portion of the current study and presented in Chapter 3. Figure 4.8 compares model predictions with the measured growth profiles of individual droplets which were average-sized on the bare aluminum surface. The steep, nearly vertical jumps in the measured data profiles represent coalescence events for the droplet. The last symbol in each "measured" curve represents the moment of freezing for that particular experimental test run. The round symbol at the end of each "model" profile represents the predicted freezing of the droplets and the end of the simulation's droplet growth stage. The error bars in the figure represent the uncertainty in droplet size measurements due to the camera's resolution. Error bars are shown for representative data points only, but similar errors existed for all points in the droplet profiles. Error bars were not included for freezing time measurements because they could be determined accurately within a few seconds. Figure 4.8(a) presents results for the "baseline" test conditions (test conditions 1 as defined in Table 3.2) for the data set. Figures 4.8(b), (c), and (d) increase air humidity (test conditions 2), decrease cold surface temperature (test conditions 3), and decrease airflow rate (test conditions 5), respectively. The model could predict the trends for each droplet profile in Figure 4.8, though it over-predicted the freezing time for the low airflow conditions significantly.

This over-prediction led to an over-prediction of droplet size at freezing, even though the model followed the experimental trend of droplet size very closely.

Figure 4.9 compares model predictions with measured droplet size profiles for the hydrophobic surface from the current study and the same nominal test conditions as for Figure 4.8. The model could not predict droplet sizes on this surface as accurately as those on the bare aluminum surface. All profiles were over predicted except for that associated with test conditions 3, which was under predicted. Freezing time predictions were also less accurate for this surface. They were slightly under-predicted for the test conditions 1 and 2, while they were significantly over-predicted for test conditions 3 and 5. The poorer predictions of droplet size for this surface were due primarily to inaccuracies in the correlation for μ as presented in Equation 4.34. This correlation slightly over-predicted the experimental values for most of the tests on the hydrophobic surface, and small changes in μ led to significant changes in predicted droplet size, especially when freezing times were large, and the droplets had a long time to grow before freezing.

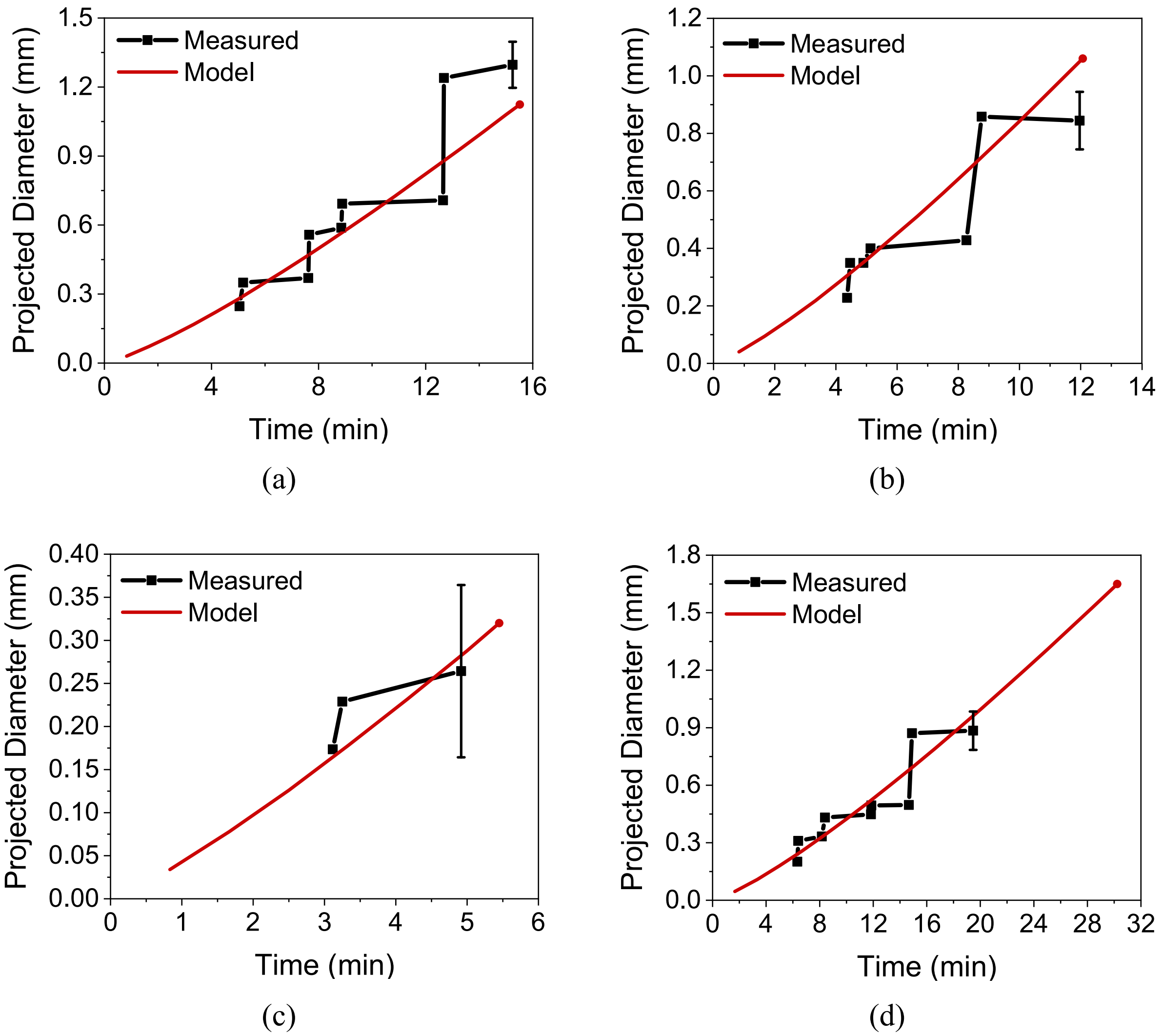


Figure 4.8: Comparison of model predictions with individual droplet profiles on a bare aluminum surface (θ ≈ 75°) from this research. (a) Test Conditions 1: T_a = 5.1 °C, RH = 77%, V = 3.8 m/s, T_s = -3.4 °C; (b) Test Conditions 2: T_a = 5.2 °C, RH = 84%, V = 3.8 m/s, T_s = -3.6 °C; (c) Test Conditions 3: T_a = 5.1 °C, RH = 75%, V = 3.8 m/s, T_s = -6.2 °C; (d) Test Conditions 5: T_a = 5.1 °C, RH = 75%, V = 1.7 m/s, T_s = -3.4 °C.

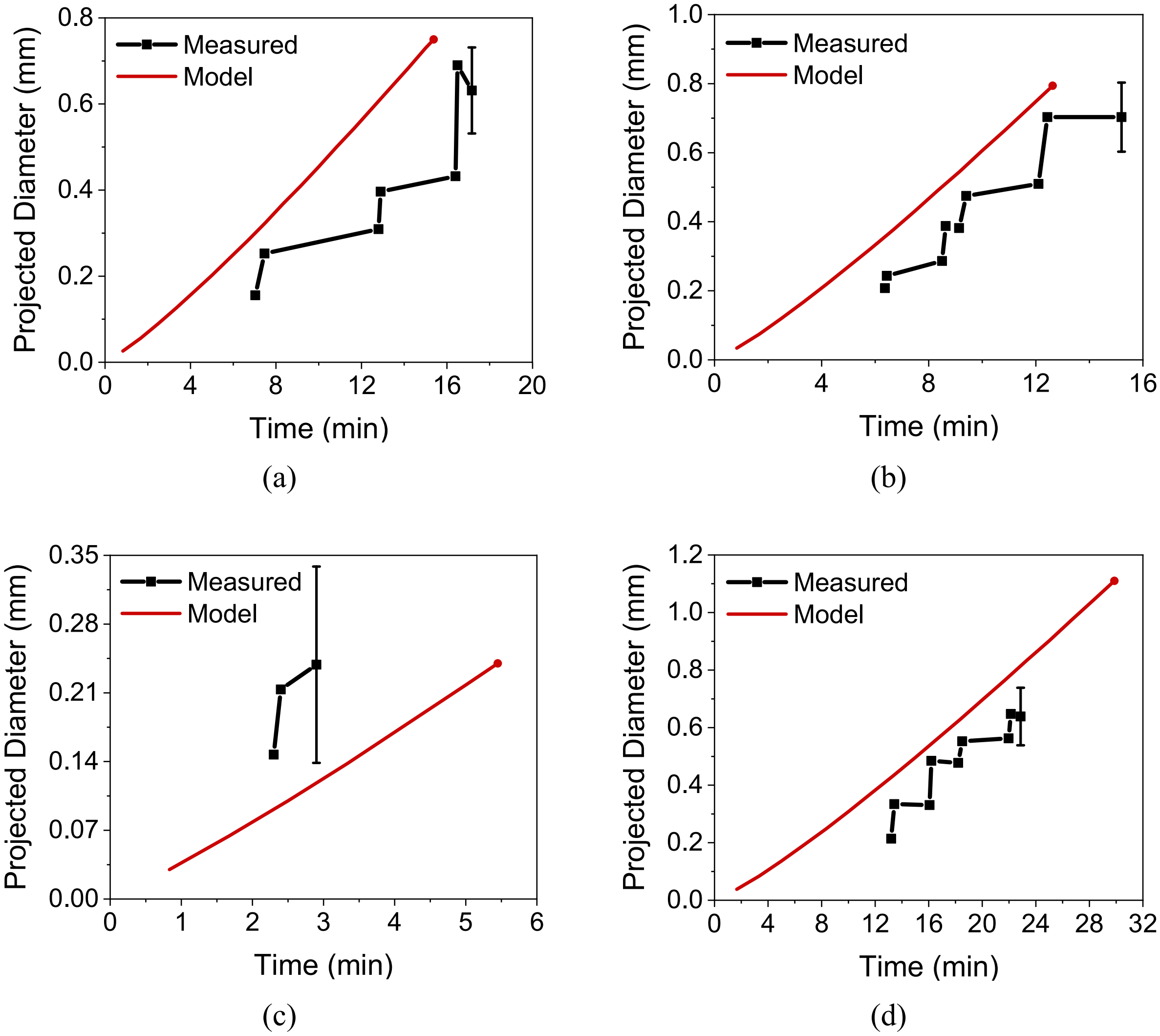
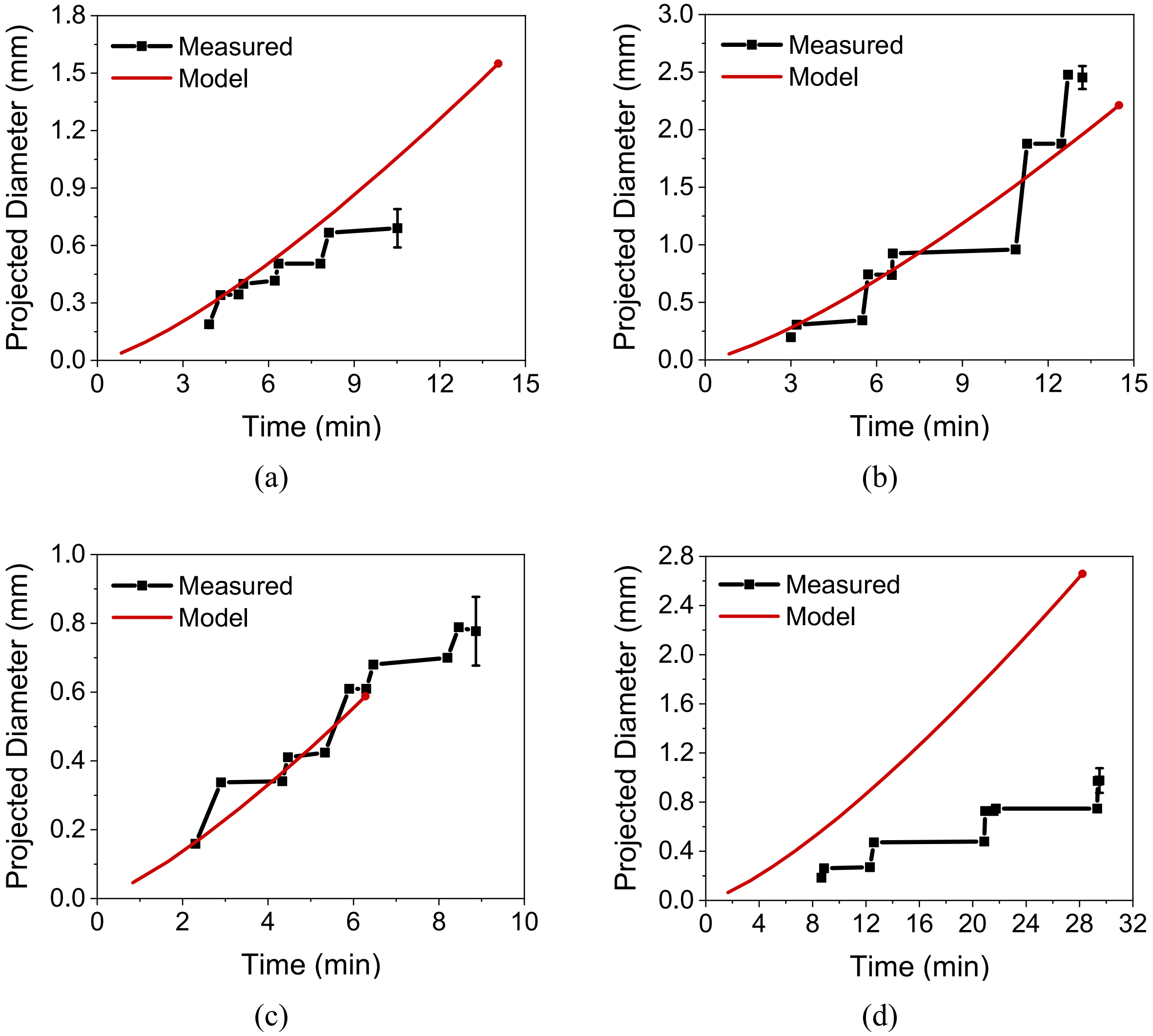


Figure 4.9: Comparison of model predictions with individual droplet profiles on a hydrophobic surface (θ ≈ 113°) from this research. (a) Test Conditions 1: T_a = 5.1 °C, RH = 77%, V = 3.8 m/s, T_s = -3.5 °C; (b) Test Conditions 2: T_a = 5.1 °C, RH = 85%, V = 3.8 m/s, T_s = -3.5 °C; (c) Test Conditions 3: T_a = 4.9 °C, RH = 76%, V = 3.8 m/s, T_s = -6.2 °C; (d) Test Conditions 5: T_a = 5.0 °C, RH = 76%, V = 1.7 m/s, T_s = -3.4 °C.

Figure 4.10 compares model predictions with measured droplet profiles for the hydrophilic surface from the current study. The model reasonably predicted the droplet size profiles for test conditions 2 and 3. However, it slightly over-predicted the trend for test conditions 1 and significantly over-predicted the trend for test conditions 5. This large over-prediction was due to the degradation of the hydrophilic surface, as described in Chapter 3. The experiments under test conditions 2 and 3 were performed first, while the surface was still fresh and undamaged. In contrast, the experiments for test conditions 1 and 5 were run after the surface had already received much use. Since the exponent μ in Equation 4.34 increased as contact angle decreased, assuming a constant contact angle for the hydrophilic surface would lead to over-predictions of μ for later tests if the surface's actual contact angle increased over time. There were not any noticeable trends when comparing freezing time predictions to measured data. Freezing times for test conditions 2 and 5 were predicted well, while those for test conditions 1 and 3 were over-predicted and under-predicted, respectively.

Figure 4.11 presents the predicted vs. measured droplet diameter at the time of droplet freezing. These data represent average droplet sizes and freezing times for multiple tests conducted on the same surface type at the same nominal test conditions. As in Figures 4.8, 4.9, and 4.10, the error bars in this figure represent measurement error due to camera resolution. Measurement uncertainty on droplet sizes and freezing times also existed due to repeatability error, but as this uncertainty was described in Chapter 3, it was not repeated here. Figure 4.11(a) shows the results when the measured freezing time was used as an input to the model to determine how long the droplets should grow, while 4.11(b) used Equation 4.47 to calculate the freezing time. When the measured freezing time was used, the model predicted most of the data within ±50%, except for some of the hydrophilic surface data (there was also one more hydrophilic data point not included in Figure 4.11 because the prediction was so far out of range). These over-predictions of the hydrophilic data were due to the degradation of the surface's hydrophilicity, as discussed in conjunction with Figure 4.10. When the freezing time was calculated using Equation 4.47, other data points also departed from the ±50% band. This departure indicated that error in freezing time predictions could lead to reasonably significant inaccuracies in determining droplet size at the point of freezing.



(a) (b)

(c) (d)

Figure 4.10: Comparison of model predictions with individual droplet profiles on a hydrophilic surface (θ ≈ 24°) from this research. (a) Test Conditions 1: T_a = 4.9 °C, RH = 76%, V = 3.8 m/s, T_s = -3.7 °C; (b) Test Conditions 2: T_a = 5.3 °C, RH = 85%, V = 3.8 m/s, T_s = -3.2 °C; (c) Test Conditions 3: T_a = 5.1 °C, RH = 76%, V = 3.7 m/s, T_s = -5.8 °C; (d) Test Conditions 5: T_a = 5.1 °C, RH = 75%, V = 1.7 m/s, T_s = -3.6 °C.

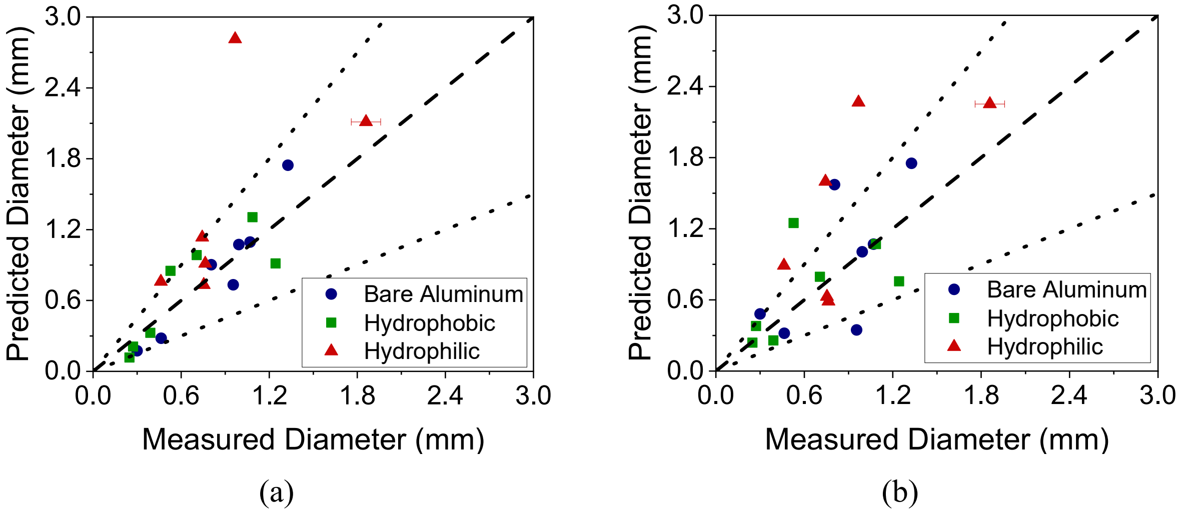
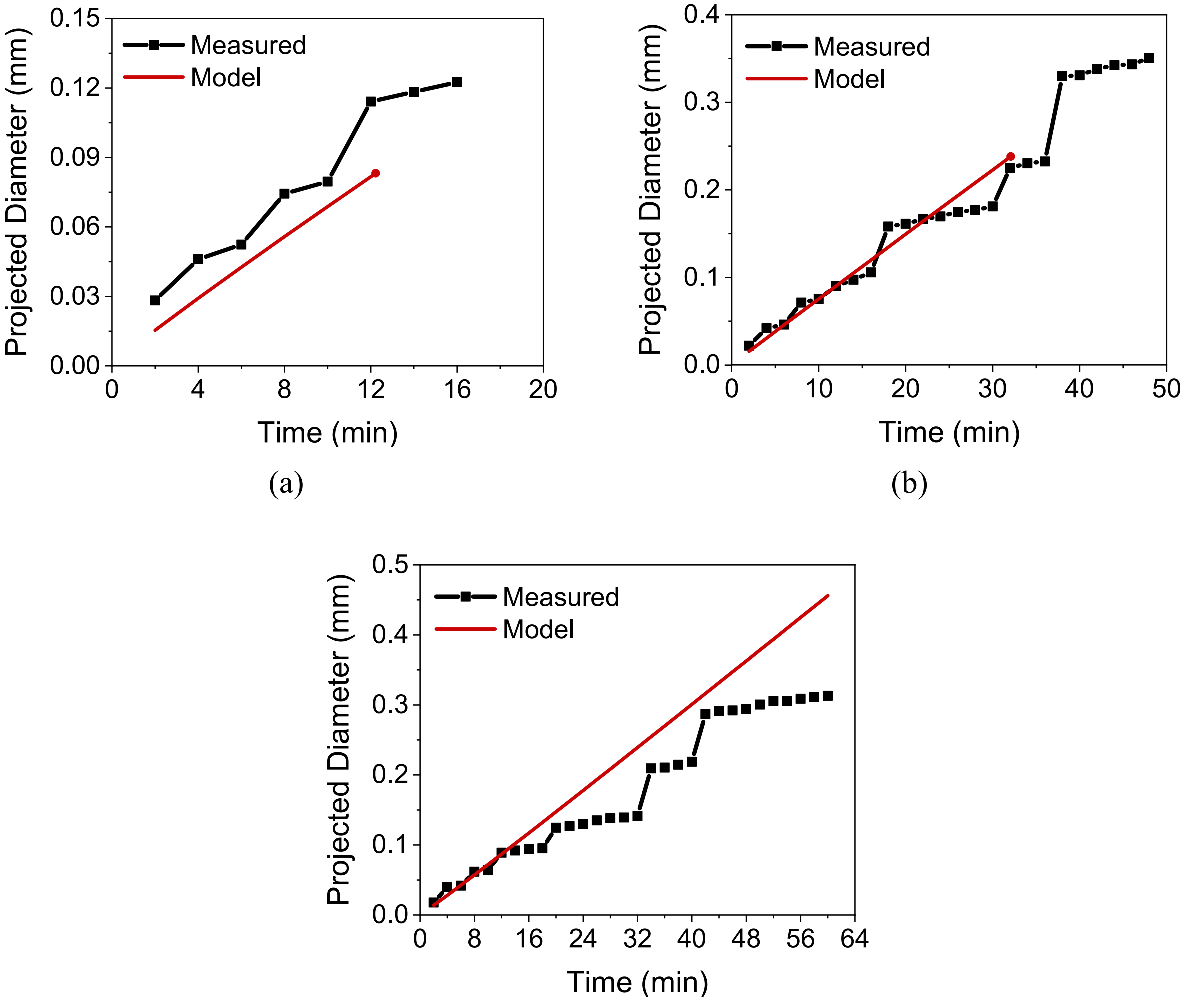


Figure 4.11: Predicted vs. measured frozen droplet radius using (a) the measured freezing time and (b) the freezing time predicted by Equation 4.47.

4.2.2 Sheng et al. (2020)

Model predictions were also compared with measured droplet growth and freezing data from Sheng et al. (2020). For these data, humid air was at 13 °C and 58% RH with a velocity of 0.4 m/s. The test surfaces were square flat plates with 40 mm to a side. Figure 4.12 compares the model predictions with three individual droplet size profiles on a superhydrophobic surface with a contact angle of 155°. Figures 4.12(a), 4.12(b), and 4.12(c) had surface temperatures of -7 °C, -4 °C, and -1 °C, respectively. The model under-predicted the freezing time for the -7 °C and -4 °C cases by small amounts. Freezing did not occur during one hour of testing for the physical experiment, nor was it predicted by the model during that hour for the case where the surface temperature was -1 °C. Additionally, the model closely predicted the droplet profiles for the -7 °C and -4 °C cases. In comparison, it slightly over-predicted the profile when the surface temperature was -1 °C, principally as the droplets grew larger.



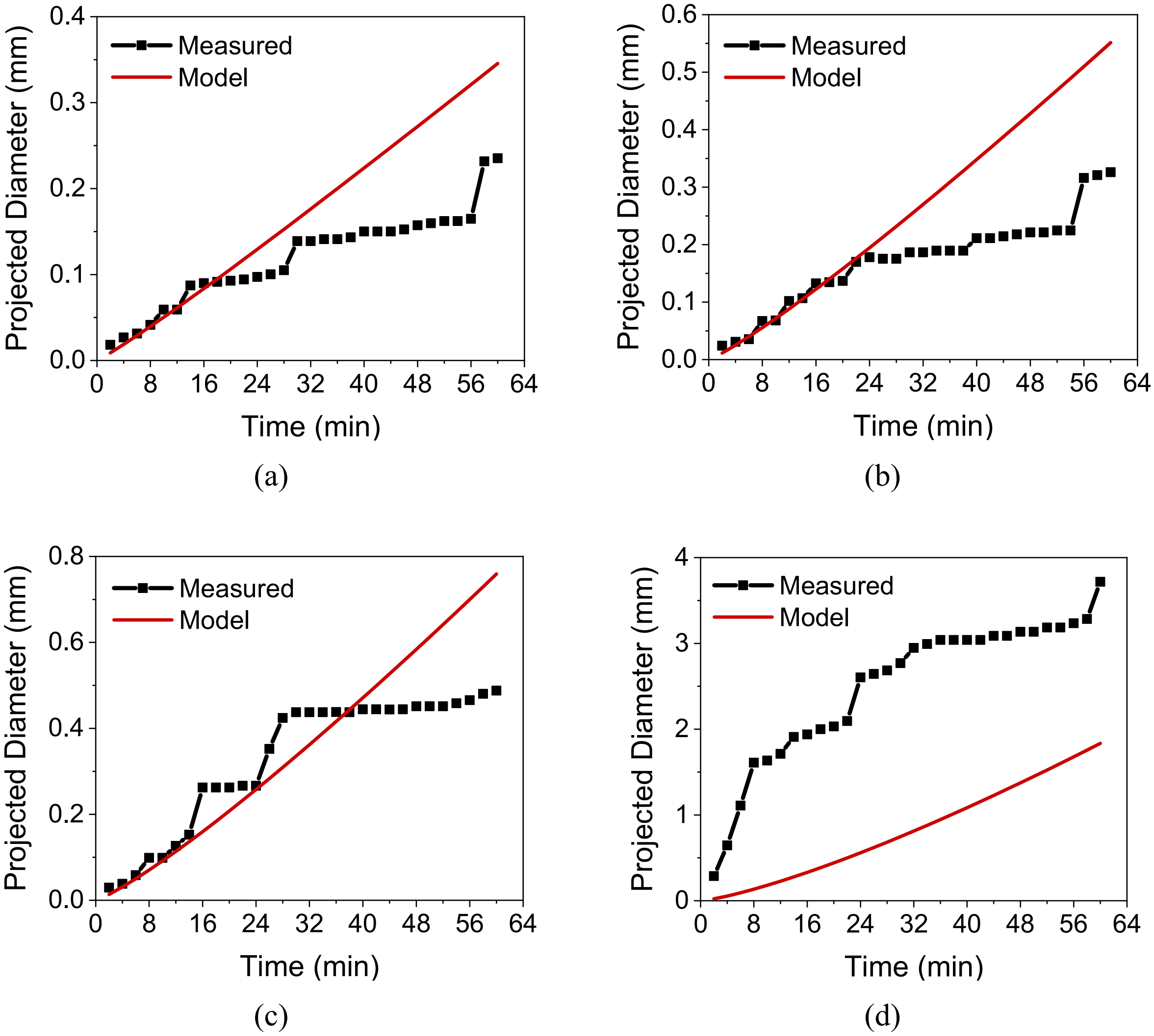
(a)

(b)

(c)

Figure 4.12: Comparison of model predictions with individual droplet profiles from Sheng et al. (2020). T_a = 13 °C, RH = 58%, V = 0.4 m/s, θ = 155°. (a) T_s = -7 °C, (b) T_s = -4 °C, (c) T_s = -1 °C.

Figure 4.13 tests the model's limits by comparing model predictions to measured data of condensation at a surface temperature above 0 °C from Sheng et al. (2020). Figures 4.13(a), (b), (c), and (d) present results for surfaces with contact angles of 155°, 116°, 89°, and 15°, respectively. The model could predict droplet size during the early stages of condensation for most of the surfaces but deviated from the measured data when the droplets' growth slowed and droplet size became more constant. However, the model could not predict droplet size for the hydrophilic surface, even at the very beginning of condensation.



(a)

(b)

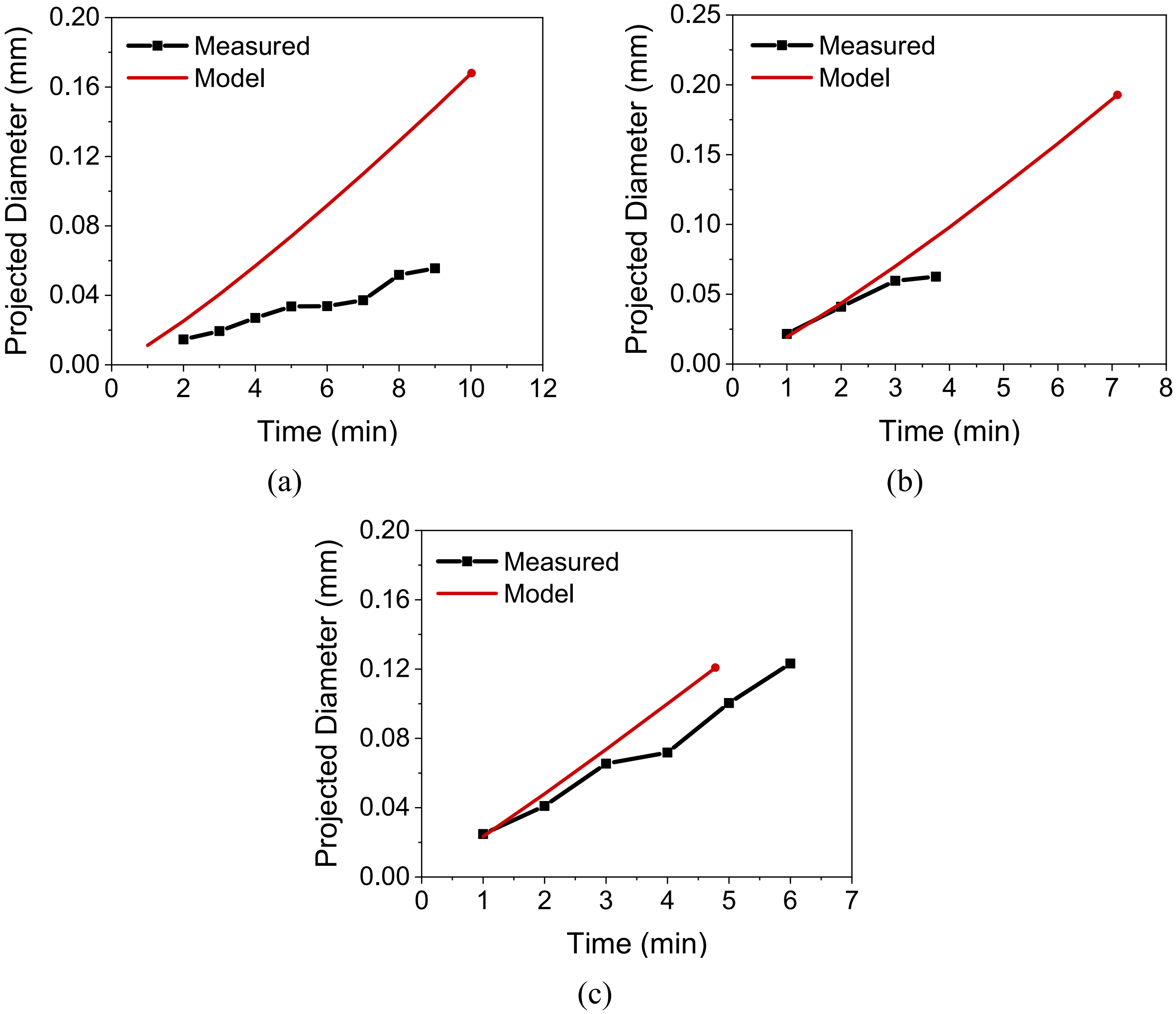
(c)

(d)

Figure 4.13: Comparison of model predictions with individual droplet profiles from Sheng et al. (2020). T_a = 13 °C, RH = 58%, V = 0.4 m/s, T_s = 2 °C. (a) θ = 155°, (b) θ = 116°, (c) θ = 89°, (d) θ = 15°.

4.2.3 Bryant (1995)

Figure 4.14 compares model predictions with individual droplet profile data from Bryant's (1995) doctoral dissertation. These data were gathered under forced convection conditions on surfaces with different wettabilities. All surfaces had 30 mm x 30 mm dimensions and were kept at a temperature of -10 °C during testing. The model was able to closely predict the freezing time for two of the three cases. It could also reasonably predict the droplet profiles for two of the three cases but over-predicted the low humidity profile (Figure 4.14(a)) by a significant amount.



(a)

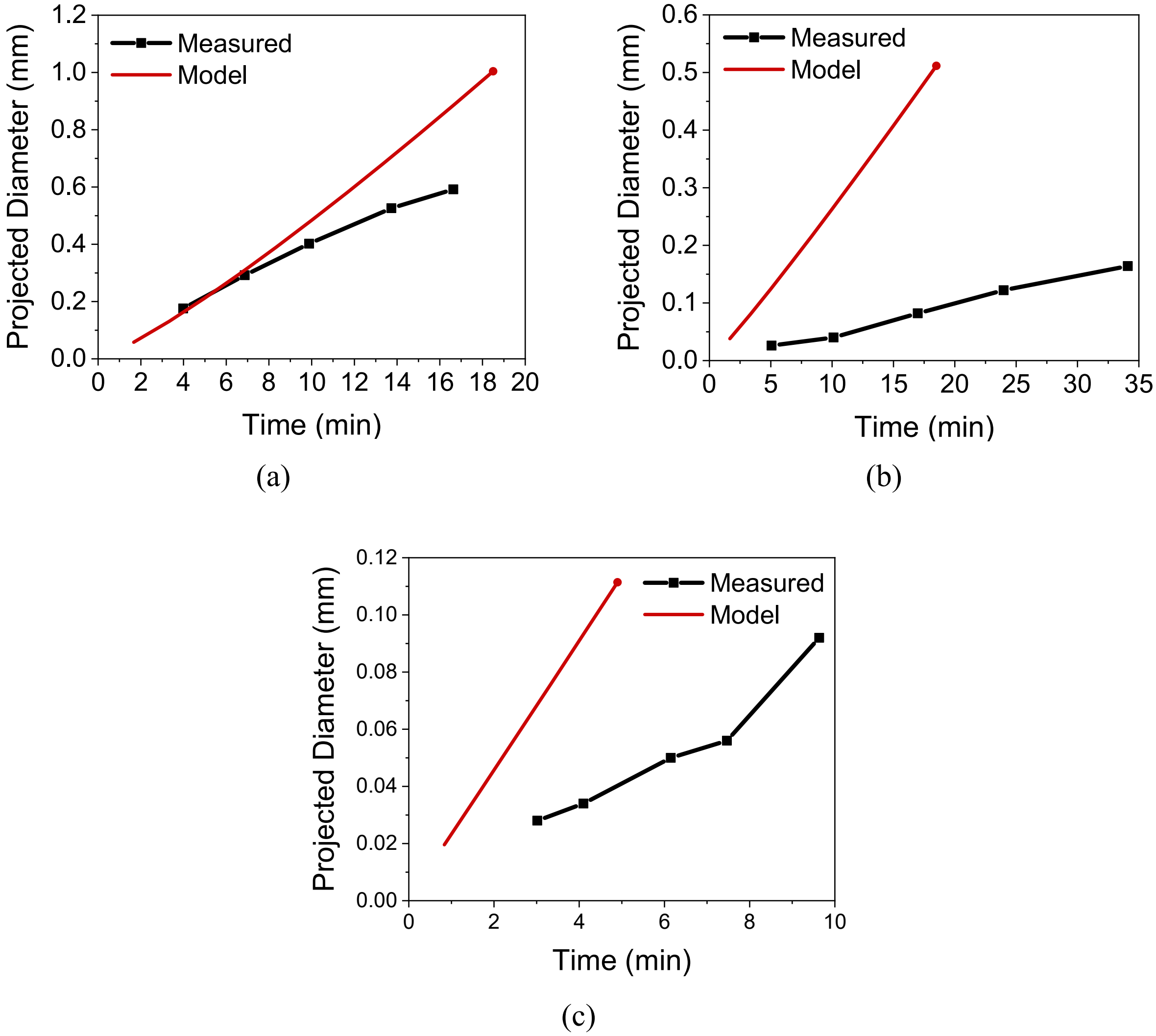
(b)

(c)

Figure 4.14: Comparison of model predictions with individual droplet profiles from Bryant (1995). T_s = -10 °C, V = 0.8 m/s. (a) T_a = 1.2 °C, RH = 65%, θ = 87.5°. (b) T_a = 1.2 °C, RH = 85%, θ = 87.5°. (c) T_a = 7.0 °C, RH = 85%, θ = 100°.

4.2.4 Seki et al. (1985)

Figure 4.15 compares model predictions to measured droplet profiles for data from Seki et al. (1985). These data were gathered under condensation frosting conditions with forced convection airflow. Air conditions were held constant for all tests at 10 °C, 59% RH, and 1 m/s, while surface temperature and type were varied. Validation with this data tested the model's limits because the experimental test plate had a vertical orientation while the model was primarily developed for horizontal flat surfaces. The model predicted both freezing time and droplet profile reasonably well for the case of high surface temperature and low contact angle (Figure 4.15(a)). The model could not predict droplet growth and freezing for the hydrophobic surface but under-predicted the freezing times and significantly over-predicted the size profiles.



(a)

(b)

(c)

Figure 4.15: Comparison of model predictions with individual droplet profiles from Seki et al. (1985). T_a = 10 °C, RH = 59%, V = 1 m/s. (a) T_s = -5 °C, θ = 43°, (b) T_s = -5 °C, θ = 110°, (c) T_s = -10 °C, θ = 110°.

4.2.5 Summary of Experimental Validation

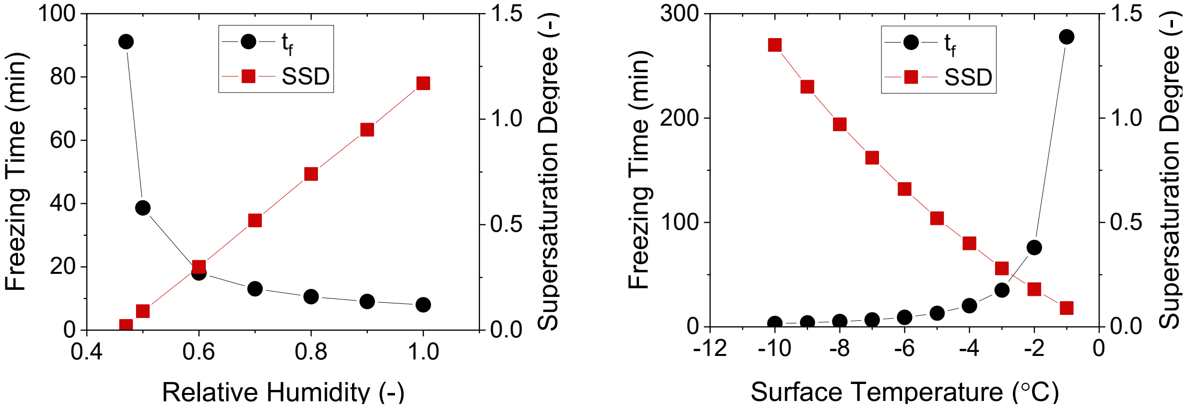
The analysis presented in Sections 4.2.1 through 4.2.4 determined that the model could predict droplet profiles and freezing times for many experimental cases spanning a range of typical heat pump operating conditions and surface types. Some limitations for the model were identified as well. First, the model was only applicable to cases where the cold surface temperature was below the triple point of water; predictions deviated from measured values as time progressed for cases where freezing never occurred. Second, the model could not predict surface decay. For the hydrophilic surface from this research, predictions grew worse as the surface's hydrophilicity degraded and the contact angle increased. Third, it is still inconclusive whether the model can be used for vertical surfaces. Some of Seki et al.'s (1985) data were reasonably well predicted, while other data from the same source were predicted very poorly. Fourth, there were still some inconsistencies in the accuracy of predictions, especially for superhydrophilic surfaces. For example, the model could predict the hydrophilic data from the current study that were gathered before the surface deteriorated, but it could not predict at all the data of the hydrophilic surface from Sheng et al. (2020).

In summary, the model presented here could be used to predict data of droplet growth and freezing on flat surfaces under forced convection conditions typical of heat pump operation, as long as the surface temperature was below the triple point of water and the surfaces were robust and not subject to deterioration. Further investigation is needed to determine if the model can produce accurate predictions for vertical surfaces. Further investigation is also needed (as new data become available) to refine the model for cases of very low contact angle.

4.3 Model Results and Discussion

Figure 4.16 highlights some of the challenges when predicting freezing time versus supersaturation degree, relative humidity, and surface temperature. Bulk air temperature, air velocity, and contact angle of the surface were held constant at 5 °C, 2.5 m/s, and 90°, respectively. Additionally, the surface temperature was held constant at -5 °C for Figure 4.16(a), while relative humidity was kept at 70% for Figure 4.16(b). Figure 4.16(a) indicates that frosting would not occur for every nonzero value of relative humidity. SSD was negative for relative humidity below 47% for these inputs. The freezing time approached infinity asymptotically as SSD approached zero, so the model indicated that freezing did not occur below this threshold value of relative humidity. The present model could not run when SSD was less than zero because this input resulted in a negative droplet growth rate, as expressed in Equation 4.32. A negative SSD means there is also a negative vapor pressure gradient and thus a negative vapor density gradient. A negative density gradient means that water vapor travels away from the surface and toward the bulk air stream under these conditions, leading to no net condensation on the surface. Since SSD depends on air humidity and the temperatures of the air and the cold surface, this condensation threshold changed as inputs to the model changed. Figure 4.16(b) illustrates the freezing time predictions' asymptotic behavior as the surface temperature changed; if the surface temperature increased to the triple point, SSD approached zero and freezing time approached infinity asymptotically. Note that Figure 4.16(b) also applies to specific model inputs. While freezing time always had asymptotic behavior when the surface temperature approached the triple point, SSD did not tend to zero for all cases. For example, if the bulk air stream's input temperature increased, SSD increased and was larger than zero at the triple point.

This trend indicated that the model predicted condensation, but not freezing, if the cold surface temperature reached the triple point with these inputs.



(a) (b)

Figure 4.16: Predictions of freezing time with variations in (a) relative humidity of the bulk air stream and (b) temperature of the cold surface.

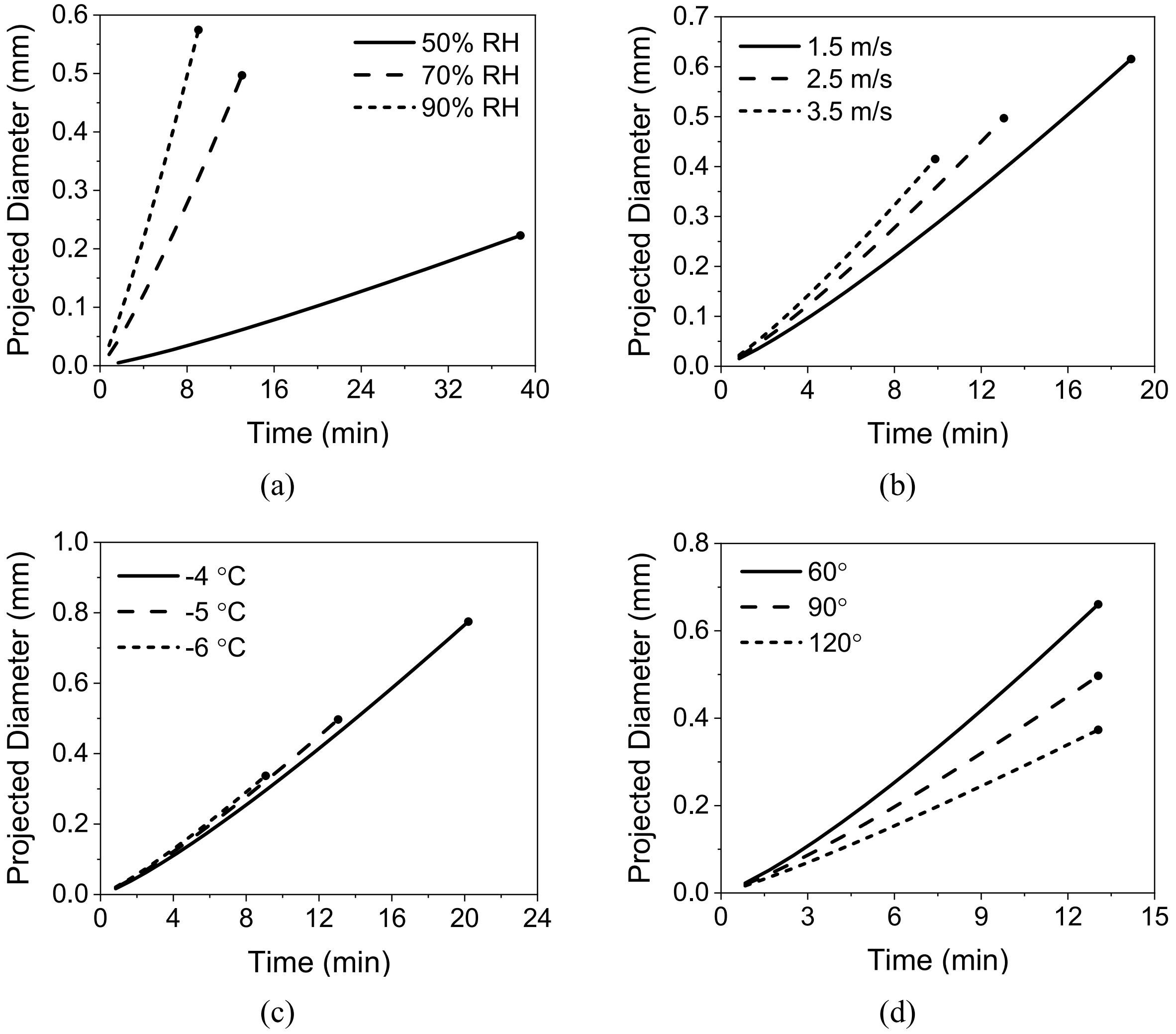
Figure 4.17 presents a parametric study of model predictions as air relative humidity, air velocity, cold surface temperature, and surface contact angle were varied. All test conditions used as inputs to the model for this study fell within the ranges for which the model is applicable and were the same as the baseline test conditions used to generate Figure 4.17. Figure 4.17(a) demonstrates how model predictions of droplet growth and freezing changed as relative humidity changed. The figure shows that both the droplet growth profile and the freezing time were dependent on the humidity level, and the effects were magnified at lower humidities. According to Figure 4.16, SSD was only slightly above zero, and freezing time was approaching asymptotic behavior at an RH of 50%. This small SSD indicated a slight vapor pressure gradient and thus very small mass transfer and droplet growth rates, leading to the shallow growth curve in Figure 4.17(a). The asymptotic behavior of freezing time at this humidity level also explains why the difference in freezing times between 50% and 70% RH was much larger than between 70% and 90% RH.

Figure 4.17(b) presents how model predictions responded to changes in the bulk air stream's velocity. As velocity increased, the droplet growth curve’s slope increased, indicating an increase in the droplet growth coefficient, *c*. The model used air velocity to calculate the Nusselt number, which was used to calculate the mass transfer coefficient. The droplet growth expression in Equation 4.17 is directly dependent on the mass transfer coefficient, so any increase in air velocity led to increased mass transfer driving potential, leading to droplets that grew larger more quickly. Freezing time was also inversely proportional to air velocity, leading to a decrease in freezing time as velocity increased.

Figure 4.17(c) shows that a change in temperature of the cold surface had a much more significant effect on the freezing time than the mass transfer potential. While the droplet growth profiles nearly overlapped for the three surface temperatures, freezing time decreased significantly as surface temperature decreased. This discrepancy was because the surface temperature only indirectly affected the mass transfer via the vapor pressure gradient, which was more strongly affected by humidity changes than by temperature. The temperature of the surface affects droplet cooling much more directly, and as droplets cool, the energy barrier required for ice nucleation decreases. This decrease in the energy required for ice nucleation leads to an increase in the probability of freezing as the surface temperature drops, and thus a shorter freezing time.

Figure 4.17(d) shows the model response to changes in the contact angle of the surface. As previously discussed, for the conditions of high surface temperature typical of heat pump operation, there has been no conclusive effect of variable contact angle on freezing time. This ambiguity led to identical predictions of freezing time for each contact angle considered. Contact angle had a significant effect on the slope of the droplet profile curve, however. As the contact angle increased, the projected droplet diameter on the surface decreased. There are two primary reasons for these results. The first is that at small contact angles, droplets are spread out on the surface and appear almost as puddles rather than as semi-spherical droplets. The droplets which are more spread out grow more quickly in the radial direction than vertically, while the rounder droplets grow more equally in all directions. This discrepancy leads to projected diameters increasing more quickly for smaller contact angles. The second reason is that more droplet coalescence events occur in a given amount of time at smaller contact angles since these droplets grow quickly in the radial direction and thus contact each other sooner. For this model, coalescence effects were quantified by μ (Equation 4.34), which increased with decreasing contact angle. μ directly influenced droplet size and had a more significant contribution the longer the droplets were allowed to grow.

In summary, changing humidity, air velocity, and surface temperature impacted droplet growth and freezing by affecting the heat and mass transfer driving potentials. However, changing the contact angle affected droplet growth by directly altering the individual droplets' geometry and coalescence behavior. Results such as those presented in Figure 4.17 can provide a basis for determining which environmental and surface conditions are most important to control when inhibiting and controlling frost growth on systems such as heat pumps that operate under high-surface-temperature condensation frosting conditions.



(a)

(b)

(c)

(d)

Figure 4.17: Parametric study of model response to changes in (a) air relative humidity, (b) air velocity, (c) cold surface temperature, and (d) surface contact angle. Baseline conditions for this study: T_a = 5 °C, RH = 70%, V = 2.5 m/s, T_s = -5 °C, θ = 90°.

4.4 Discussion about the Model's Ability to Capture the Physics of Droplet Growth and Freezing

The semi-empirical model presented in this chapter could predict droplet growth and freezing data for a range of test conditions and surface types, but it also failed in some respects. These successes and failures indicated where the model could correctly follow the physical processes governing droplet growth and freezing and where it could not. Comparing model predictions with droplet profile data for the current study's bare aluminum surface revealed good agreement between the two, indicating that the droplet growth predictions responded correctly to changing environmental conditions. Comparisons of model predictions with the hydrophobic and hydrophilic surfaces showed worse agreement, demonstrating that the model could not track all the effects of surface wettability on the droplet growth. The likely cause for much of the disagreement came from the inability to physically describe the nucleation of new droplets on the surfaces. Nucleation characteristics determine the initial distribution of droplets at the beginning of condensation. However, they also affect the entire growth process because there is secondary nucleation in the spaces left behind by coalescing droplets. While the model theoretically calculated the critical radius for the onset of condensation, it had no way of determining the number or spacing of the droplets as they nucleated or how these distributions changed with surface type. This gap led to the need for correlations to predict area coverage and the growth exponent, both of which are dependent on nucleation and coalescence behavior. The model could also track some, but not all, of the physics governing droplet freezing. The supersaturation degree was identified as a key parameter for determining freezing time across the entire data range. However, other parameters such as air velocity and surface contact angle only had significant effects under certain conditions. A correlation was proposed to calculate the freezing time for typical heat pump operating conditions. However, since the correlation was not applicable for all test conditions and error still existed in the predictions of data for which it was applicable, it was evident that the model could not identify all the physical processes that contributed to condensate freezing.

5. Modeling the Frost Layer after Initial Nucleation

After the model determined the freezing time and average diameter of droplets at freezing, it calculated the characteristics of the frost layer that deposited on top of the frozen droplets. It did this by first performing calculations specific to the crystal growth phase of frosting. This modeling stage was developed as part of the current study and relied heavily on data gathered during this study and presented in Chapter 3. The details of this stage of the model are presented in Section 5.1 below. Since several existing models available in the literature could predict subsequent frost layer growth characteristics, it was unnecessary to develop an entirely new calculation method for this last stage of the frost model. We began with the frost model presented by Padhmanabhan (2011) and modified it as described in Section 5.2. Finally, all three stages of the frost model (droplet growth and freezing, crystal growth, and frost layer growth) were coupled together to produce a single continuous simulation tool. Validation and results of the final version of the overall model are presented in Sections 5.4 and 5.5 below.

5.1 Crystal Growth Stage Model Development

The model's crystal growth stage was developed as a semi-empirical method of calculating frost layer properties before the frost layer could be approximated as a homogeneous porous layer. It employed a thermal resistance network in a similar fashion to the droplet growth model to calculate the heat transfer through a single crystal, which was used to calculate the crystal radial growth rate. Crystal size and distribution were used in conjunction with the droplet model results to calculate the overall frost layer's thickness and density and the point of transition to the frost layer growth stage. This stage of the model needed the results of the droplet model, primarily droplet size, droplet area coverage, and freezing time as inputs, as well as a few user-defined inputs, for which no currently existing methods of theoretical determination were found during the current investigation. The primary assumptions of this stage of the model are listed below:

(a) All crystals could be modeled as solid cylinders.

(b) All crystals were the same average size at all times.

(c) Crystals always grew from the tops of frozen droplets and not directly from the cold surface.

(d) Humid air was saturated at the cold surface.

(e) Ice crystals were small compared to the cold surface's characteristic length, so they did not significantly alter the heat transfer geometry.

A previous version of the model (presented in Harges and Cremaschi (2018b)) assumed that a single crystal grew from the top of each frozen droplet. This assumption was supported by photographs from the experiments of Cheikh and Jacobi (2014). However, based on observations of the data gathered during the current study, it was determined that this was not always the case and that many crystals grew out perpendicularly from large droplets. Because of these observations, this assumption was discarded for the final version of the model. Another assumption from the Harges and Cremaschi (2018b) version of the model was that the crystal aspect ratio, defined as crystal height divided by crystal radius, remained constant as the crystals grew. This assumption was also discarded after preliminary comparisons of model results to data gathered during this study. Figure 5.1 provides example diagrams of a crystal on a frozen droplet from a side view and a top view to illustrate assumptions (a), (b), and (c). The model predicts that multiple crystals grow from each droplet, but the crystals can be lumped together into a single crystal with equivalent height and diameter for practical purposes. The equivalent height would be equal to the heights of the individual crystals (which are all the same size), but the equivalent diameter would be such that the volume of the equivalent crystal is equal to the total volume of the individual crystals.

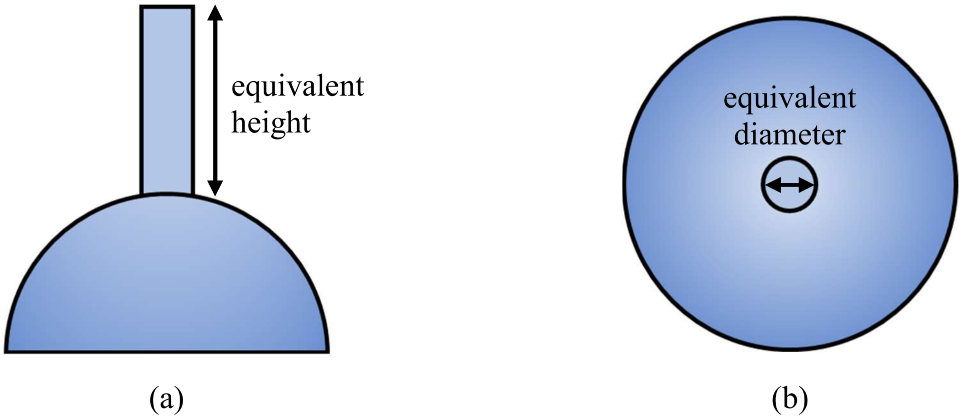


Figure 5.1: Diagram of crystals on a frozen droplet from a (a) side view and (b) top view.

5.1.1 Initialization

To start the crystal growth model, initial values for the crystal radius, r_c, crystal area coverage, AC_c, and crystal aspect ratio, $c = L_c / r_{c,}$ must be specified. In this work, the crystal area coverage was defined as the total projected area of crystals on a droplet divided by the projected area of an average-sized droplet. The investigation for this study did not reveal reports of these values in the literature, nor could they be determined directly from the current study's experiments because of inadequate camera resolution. For this model, constant values of 3 μm and 0.005 were used for the initial values of crystal radius and crystal area coverage, respectively, for all simulations. These values were chosen in order to provide the best fit for the early frost growth data. The aspect ratio was the only initial parameter that could not be held constant to fit data gathered under different environmental conditions. It was observed that the initial aspect ratio was most dependent upon the temperature of the cold surface, and a simple correlation was developed to obtain this value. This correlation was developed using data from the current study, Cheng and Wu (2003), and Hermes et al. (2009). It is presented as Equation 5.1, where T_s had units of °C.

$$c_0 = -\frac{(T_s + 3.5)}{3} + 2.5 \tag{5.1}$$

Unlike in the droplet model, for which the number of droplets was recalculated at each time step because of coalescence and renucleation effects, the number of crystals per droplet remained constant throughout the entire crystal growth period. The model recalculated crystal radius based on mass transfer and the new value for aspect ratio. Then crystal area coverage would be recalculated geometrically since both projected droplet area coverage and number of crystals remained constant.

5.1.2 Ice Properties

The model required the density of the frozen droplets and the ice crystals to calculate the frost layer's overall density. Since the freezing of liquid water formed the frozen droplets, their density was assumed to be equal to that of solid ice, about 917 kg/m^3. This assumption could not be made for ice crystals deposited by ablimation, however. Research performed by atmospheric scientists indicated that ice crystals formed by ablimation at temperatures below 0 °C generally had lower densities than solid ice. For this model's purposes, ice crystal density was calculated using a polynomial curve fit developed in the current study, based on data from Fukuta (1969) and Ryan et al. (1976). The resulting expression is given below as Equation 5.2. In Equation 5.2, ρ_c is only dependent on the crystal temperature, T_c, because that was the only experimental parameter reported by the sources. The order of the polynomial was chosen based on that which provided the closest fit to the data. In Equation 5.2, T_c has units of °C, while ρ_c had units of g/cm^3.

$$\rho_c = 0.969 - 0.220\, T_c - 0.175\, {T_c}^2 - 0.0372\, {T_c}^3 - 0.00342\, {T_c}^4 - (1.43 \times 10^{-4})\, {T_c}^5 - (2.21 \times 10^{-6})\, {T_c}^6 \tag{5.2}$$

The model also required the ice thermal conductivity to calculate the conduction heat transfer through the ice crystals. The expression used in this model came from Hobbs, 1974, and is presented below as Equation 5.3.

$$k_{ice} = \frac{488.19}{T_c} + 0.4685 \tag{5.3}$$

The latent heat of sublimation was calculated using a correlation developed by Feistel and Wagner (2007), presented in Equation 5.4 and Table 5.1. The correlation is applicable for temperatures between 20 and 273 K. In Equation 5.4, both temperatures have units of K, and H_{sg} has units of J/kg.

$$H_{sg} = \sum_{i=0}^{6} a_i \left(\frac{T}{T_{tp}}\right)^i \tag{5.4}$$

Table 5.1: Coefficients for use in Equation 5.4

Variable	Value
a_0	2,638,742.45418107
a_1	400,983.673912406
a_2	200,812.111806393
a_3	-1,486,203.38485336
a_4	2,290,451.50230789
a_5	-1,690,159.93521118
a_6	479,848.354373932

5.1.3 Heat Transfer through a Crystal

As with the heat transfer to a droplet, heat transfer to a crystal consisted of sensible and latent parts and is presented in Equation 5.5. This expression for heat transfer is nearly identical to the droplet expression, except that crystal radius was used, the latent heat of sublimation was used rather than the latent heat of vaporization, and the latent heat term was modified by the area coverage of crystals on a droplet as well as by the area coverage of droplets on the surface.

$$\dot{Q}_c = \pi\, r_c{}^2 \left[h\,(T_\infty - T_c) + \frac{h_m}{AC_d \cdot AC_c}\left(\rho_{v,\infty} - \rho_{v,c}\right)H_{sg}\right] \tag{5.5}$$

The heat and mass transfer coefficients were calculated next, but as the calculations were identical to those for the model's droplet growth portion, they are not repeated here.

Just as iterations were performed to determine the droplet model's interface temperature, iterations were performed to obtain the temperature at the top of the crystal, T_c, in the crystal growth model. Within each iteration, heat transfer through a crystal was calculated using a thermal resistance network as in the droplet model. There were two temperature differences for the crystal model that represented thermal resistances: one for conduction resistance and one for convection resistance. These temperature differences are expressed as Equations 5.6 and 5.7 below, with the total calculated temperature difference presented in Equation 5.8.

$$\Delta T_{cond} = \frac{\dot{Q}_c L_c}{\pi r_c{}^2 k_c} \tag{5.6}$$

$$\Delta T_{conv} = T_\infty - T_c \tag{5.7}$$

$$\Delta T_{total} = \Delta T_{cond} + \Delta T_{conv} \tag{5.8}$$

At this point, the C++ simulation iterated until a value of T_c was found that caused the calculated total temperature difference to be equal to the temperature difference between the bulk humid air and the crystal.

5.1.4 Crystal Radial Growth Rate

The portion of the crystal heat transfer that contributed to crystal growth was the latent heat transfer. This latent heat transfer, presented as a time derivative, is given in Equation 5.9.

$$\frac{dQ_c}{dt} = \pi\, r_c{}^2 \frac{h_m}{AC_d \cdot AC_c} \left(\rho_{v,\infty} - \rho_{v,c}\right) H_{sg} \tag{5.9}$$

This latent heat transfer rate contributed directly to an increase in the volume of a crystal. The volume of a cylindrical crystal and its derivative with respect to the radius are given in Equations 5.10 and 5.11.

$$Vol = \pi\, r_c{}^2 L_c \tag{5.10}$$

$$d(Vol) = 2\pi r_c L_c dr \tag{5.11}$$

Equation 5.12 presents an expression for the latent heat transfer on the crystal side, substituting the crystal volume change from Equation 5.11.

$$dQ_c = \rho_c H_{sg} d(Vol) = 2\pi r_c L_c \rho_c H_{sg} dr \tag{5.12}$$

The two expressions for latent heat transfer from Equations 5.9 and 5.12 were equated, and the definition of the crystal aspect ratio, *c*, was employed. The radial growth rate of a crystal was then obtained by solving for the differential change in crystal radius, and the resulting expression is presented as Equation 5.13.

$$\frac{dr_c}{dt} = \frac{h_m(\rho_{v,\infty} - \rho_{v,c})}{2\ c\ AC_d\ AC_c\ \rho_c} \tag{5.13}$$

The new crystal radius was then calculated according to Equation 5.14 below.

$$r_{c,new} = r_{c,old} + \frac{dr_c}{dt}\ \Delta t \tag{5.14}$$

Unlike the droplets' surface area coverage, the surface area coverage of crystals on a droplet could be recalculated geometrically at each time step. Since the crystals could not coalesce and reform but only grow in radius and height in their original positions, crystal area coverage could be obtained using Equation 5.15. In this equation, *N_c* refers to the number of crystals per droplet, which was calculated during the crystal growth stage's initialization.

$$AC_c = N_c\left(\frac{{r_c}^2}{{r_d}^2}\right) \tag{5.15}$$

5.1.5 Frost Layer Thickness and Density

Once the crystals' radial growth rate was determined, the frost layer's thickness and density could be calculated. The height of the crystals was calculated using the aspect ratio, as depicted in Equation 5.16.

$$L_c = c\ r_c \tag{5.16}$$

As discussed at the beginning of Section 5.1, the initial assumption that *c* was constant during the crystal growth period was not valid, but rather *c* increased as the crystals grew. An expression was developed for *c* and presented in Equation 5.17, where *t*, t_f, and *Δt* represent the current simulation time, freezing time, and the simulation's time step, respectively.

$$c_{new} = c_{old} + \frac{1}{e^{\left((t-t_f)/20\right)}} + 0.02\ (\Delta t) \tag{5.17}$$

The second term in Equation 5.17 caused significant increases in *c* at the beginning of the crystal growth stage. However, this term damped out quickly so that the increase in *c* was essentially constant for most of the crystal growth period. The specific form of Equation 5.17, particularly concerning the second term, was chosen to cause the crystal temperature to increase as the crystals grew. If the crystals grew too slowly at the beginning of crystal growth, the model would predict a crystal temperature that decreased with time, which makes no physical sense. However, crystal temperature would increase throughout the entire crystal growth stage with the proposed equation form. *c* was recalculated at the beginning of each time step and then used in Equations 5.13 and 5.16 to calculate crystal radial growth rate and crystal height.

Once crystal height was obtained, the frost layer's total height was calculated according to Equation 5.18, where $\delta_{f,0}$ refers to the initial frost layer's thickness as calculated at the end of the droplet growth stage.

$$\delta_f = \delta_{f,0} + L_c \tag{5.18}$$

The density of the frost layer during the crystal growth phase was calculated using a volumetrically averaged approach as shown in Equation 5.19, where the first term accounts for the frozen droplets, the second term refers to the crystals, and the third term represents the air in between the droplets and crystals.

$$\rho_f = \left(\frac{Vol_d}{Vol_{tot}}\right)\rho_{ice} + \left(\frac{Vol_c}{Vol_{tot}}\right)\rho_c + \left(\frac{Vol_{air}}{Vol_{tot}}\right)\rho_{air} \tag{5.19}$$

5.1.6 Effective Thermal Conductivity

In addition to calculating the frost layer's thickness and density during crystal growth, a method for calculating effective thermal conductivity was also developed. This method was based on calculations for thermal resistances through the frost layer. Similar methods have been employed by other researchers as well. However, this method is unique because it accounts for the frozen droplets by adding an equivalent ice layer between the crystals and the cold surface. Negrelli and Hermes (2015) compared calculations of "parallel" and "series" thermal resistance networks to data of frost layer thermal conductivity. They determined that measured thermal conductivity values fell between values predicted by the two kinds of networks. Based on these observations, the current study proposes calculating effective thermal conductivity by using a weighted average of "parallel" and "series" methods. Figure 5.2 illustrates the "parallel" and "series" thermal resistance networks used in the current model, which both include equivalent ice layers due to frozen droplets. In the figure, T_v represents the temperature of the air voids in between the crystals at the top of the frost layer. This temperature is approximate, but air's thermal conductivity is nearly constant in the frost layer's temperature range, so this temperature negligibly affects the final value of effective thermal conductivity.

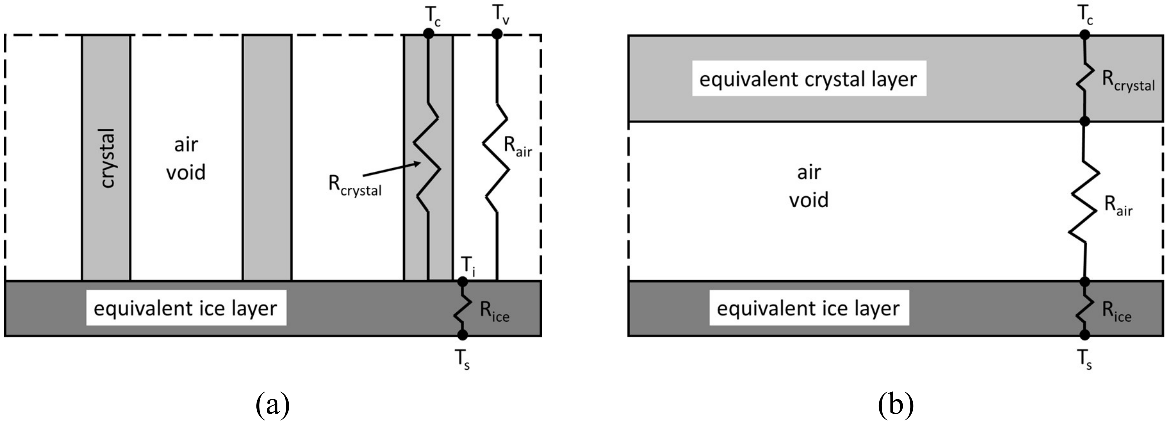


Figure 5.2: Examples of methods of calculating effective thermal conductivity. (a) Parallel method, (b) series method.

Equations 5.20 and 5.21 represent the thermal conductivities calculated using the “parallel” and “series” networks illustrated in Figure 5.2. In both equations, ε_c represents the volume ratio of crystals to the total frost layer above the equivalent ice layer.

$$k_{parallel} = \frac{\delta_f}{\frac{L_c}{(\varepsilon_c k_{ice} + (1 - \varepsilon_c) k_{air})} + \frac{\delta_0}{k_{ice}}} \tag{5.20}$$

$$k_{series} = \frac{\delta_f}{\left(\frac{\varepsilon_c L_c}{k_{ice}} + \frac{(1 - \varepsilon_c) L_c}{k_{air}} + \frac{\delta_0}{k_{ice}}\right)} \tag{5.21}$$

Equation 5.22 is the calculation for the weighted average effective thermal conductivity during crystal growth. In this equation, α_k represents the weighting factor, which was chosen in such a way as to match the values of effective thermal conductivity during crystal growth with frost conductivity during frost layer growth at the transition between the two. It was observed that α_k was primarily dependent on the cold surface temperature, and a simple correlation expressing this relationship was included in the model. It is presented as Equation 5.23 below, where T_s is in °C.

$$k_{eff} = \alpha_k k_{parallel} + (1 - \alpha_k) k_{series} \tag{5.22}$$

$$\alpha_k = -0.0351\, T_s + 0.4274 \tag{5.23}$$

5.1.7 Transition to Frost Layer Full Growth

Hayashi et al. (1977) was one of the first studies to acknowledge a crystal growth phase of frosting, distinct from the frost layer growth phase. They also defined a criterion for the transition point between crystal growth and frost layer growth: when the density of the frost layer reached a minimum value, the crystal growth phase ended, and the frost layer growth began. The model of the current study used the same criterion. The simulation remained in the crystal growth stage as long as the new calculated density was smaller than the previous time step's density. As soon as this was no longer true, the simulation switched to the frost layer growth stage, transferring the final thickness and density from the model's crystal growth portion to be the initial thickness and density necessary to start the frost layer growth model.

Tao et al. (1993b) proposed a method for calculating the frost surface temperature at the transition between the crystal growth period and the frost layer growth period. Their method was applied in the current model to check the overall model's cohesiveness at this transition. First, the temperature of the air voids between crystals at the frost surface was calculated according to Equation 5.24. Tao et al. used a value of 0.4 for β in their sample model results; they did not indicate how they chose this value, but only that it was larger for turbulent flow than for laminar flow. It was observed during the current study that β was not constant but rather varied with the temperature difference between the bulk air stream and the droplet/air interface temperature. A simple curve fit was developed to calculate β, which caused frost surface temperature at the end of crystal growth to match that calculated at the beginning of frost layer growth within ±0.5 °C. The resulting equation for β is presented as Equation 5.25.

$$T_{air,fs} = \beta\, T_\infty + (1 - \beta)\, T_s \tag{5.24}$$

$$\beta = 25.77\, (T_\infty - T_i)^{-2.42} \tag{5.25}$$

The results of Equation 5.24 were then used in Equation 5.26 to calculate the frost surface temperature. In Equation 5.26, ε_c is the volumetric ratio of crystals in the crystal/air layer on top of the frozen droplets. In Equations 5.24 – 5.26, all temperatures have units of K.

$$T_{fs} = \varepsilon_c \, T_c + (1 - \varepsilon_c) \, T_{air,fs} \tag{5.26}$$

The frost surface temperature calculated in Equation 5.26 was compared to the frost surface temperature calculated at the beginning of the frost layer full growth stage to check the homogeneity of the results produced by both model stages. This frost surface temperature calculated at the end of the crystal growth stage was not used in any subsequent calculations, so Equations 5.24 – 5.26 are of secondary importance to the model.

5.2 Frost Layer Full Growth Model Description

Since several frost layer growth models are already available in the literature, it was unnecessary to develop a completely new one for this model's final stage. The frost model implemented in this study was primarily based on the model previously presented by Padhmanabhan (2011), which was then modified to suit the current study's needs.

5.2.1 Mass Fluxes

In his model, Padhmanabhan treated the frost as a homogeneous layer with average density and thermal conductivity. He assumed that the total mass flux to the frost layer could be divided into two portions which increased overall frost density and thickness, respectively, as shown in Equation 5.27 below.

$$\dot{m}'' = h_m \left(\rho_{v,\infty} - \rho_{v,fs} \right) = \dot{m}''_\rho + \dot{m}''_\delta \tag{5.27}$$

To calculate the mass flux that increased the frost layer's density, Padhmanabhan developed an equation based on Fick's Law of Diffusion and the Clausius-Clapeyron equation. The first accounted for the diffusion of water vapor into the frost layer, and the second accounted for the vapor pressure gradient within the frost layer. The current study's model used a modified version of Padhmanabhan's equation to calculate the frost layer's diffusion mass flux, as presented in Equation 5.28. The primary difference between Equation 5.28 and Padhmanabhan's original equation was that the density of solid ice was replaced with the crystal deposition density (Equation 5.2) to obtain a more accurate representation of the density and porosity of the deposited frost layer.

$$\dot{m}_\rho'' = D_{AB}\left(\frac{1-\left(\rho_f/\rho_c\right)}{1+\left(\rho_f/\rho_c\right)^{0.5}}\right)\frac{P_{v,sat}}{R_v {T_{fs}}^2}\left(\frac{H_{sg}}{R_v {T_{fs}}^2}-1\right)\frac{\left(T_{fs}-T_s\right)}{\delta_f} \tag{5.28}$$

The mass flux that increased the thickness of the frost layer was then calculated using Equation 5.29.

$$\dot{m}_\delta'' = \dot{m}'' - \dot{m}_\rho'' \tag{5.29}$$

5.2.2 Frost Surface Temperature

The frost layer's surface temperature must be known to use Equation 5.27 to calculate the mass fluxes. Padhmanabhan presented an energy balance at the frost layer's surface, which, when rearranged, could be used to solve for the frost surface temperature using Equations 5.30 and 5.31.

$$T_{fs} = \frac{\frac{T_s}{\delta_f}\left(k_f + M\,H_{sg}\right) + h\,T_\infty + h_m\left(\rho_{v,\infty}-\rho_{v,fs}\right)H_{sg}}{\frac{k_f}{\delta_f} + M\,\frac{H_{sg}}{\delta_f} + h} \tag{5.30}$$

where

$$M = \dot{m}_\rho''\frac{\delta_f}{\left(T_{fs}-T_s\right)} \tag{5.31}$$

Since Equation 5.30 is implicit, iterations were performed in the simulation tool to determine the frost surface temperature. Equations 5.27 - 5.31 and all property calculations were performed at each time step until convergence was reached for the frost surface temperature.

5.2.3 Frost Thermal Conductivity

The frost layer's thermal conductivity is an essential variable for predicting the heat transfer through the frost layer when treated as a homogenous, porous layer. Observations during the current study indicated that very little experimental data exist for the thermal conductivity of frost at the high cold surface temperatures typical of heat pump system operation. It follows that there are also few methods aimed at calculating frost thermal conductivity in this temperature range. Negrelli and Hermes (2015) was one of a few studies that fitted a correlation specifically for a high-temperature range. Their correlation was developed using the few data available from published literature at the time, and it was the one adopted for use in the current study. It is reproduced below as Equation 5.32.

$$\frac{k_f}{k_{ice}} = 1.576 \left(\frac{k_{air}}{k_{ice}}\right)^{0.797\,\varepsilon} \tag{5.32}$$

In Equation 5.32, ε is the porosity of the frost, defined as in Equation 5.33 below. In this model, ice crystal density is used as the ice density since ice deposits in the frost layer as crystals instead of being formed by the freezing of liquid water.

$$\varepsilon = \frac{\rho_{ice} - \rho_f}{\rho_{ice} - \rho_{air}} \tag{5.33}$$

5.2.4 New Thickness and Density

After convergence was reached, the new frost thickness and density were calculated according to Equations 5.34 and 5.35, respectively.

$$\delta_{f,new} = \delta_f + \frac{\dot{m}''_\delta}{\rho_{f,dep}} \Delta t \tag{5.34}$$

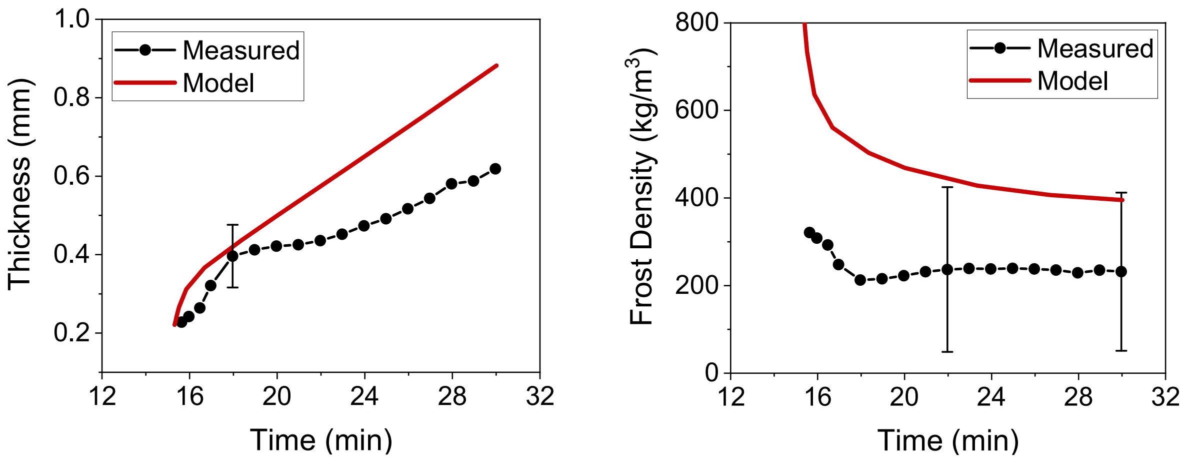
$$\rho_{f,new} = \frac{\delta_f}{\delta_{f,new}} \left(\rho_f + \frac{\dot{m}''_\rho}{\delta_f} \Delta t \right) + \frac{\left(\delta_{f,new} - \delta_f \right)}{\delta_{f,new}} \rho_{f,dep} \tag{5.35}$$

In Equations 5.34 and 5.35, $\rho_{f,dep}$ represents the density at which frost was deposited on top of the frost layer instead of the frost layer's average density. When this density was taken to be equal to the frost layer's average density, Equations 5.34 and 5.35 collapsed to be nearly identical to the standard equations found in several published works. Since there is very little information available on the density at which frost is deposited, the model used the average density as the deposition density. Frost deposition density is an area of interest for future investigation, however.

5.3 Model Validation – Frost Growth after Initial Nucleation

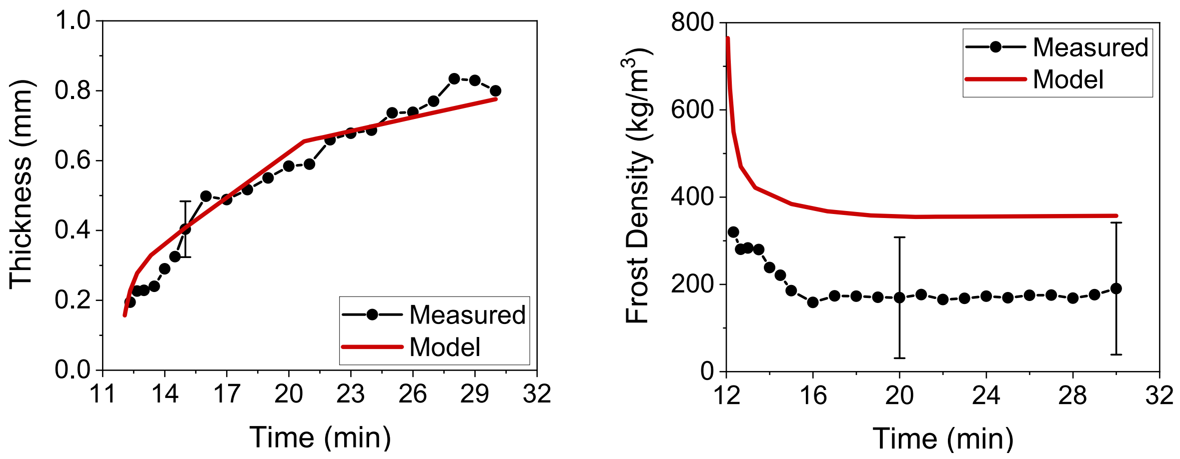
Model validation was performed first when the droplet growth model's primary results were bypassed to separate the droplet model's validation from validation for the crystal growth and frost layer growth models. Values of measured droplet radius and freezing time from the current study were used as inputs to the crystal growth model instead of calculated values. It was observed that frost mass (and thus also frost density) was over-predicted for the bare aluminum and hydrophobic surfaces, while it was under-predicted for the hydrophilic surface. If the frozen droplet mass, which was an input to the crystal growth model, was adjusted, predictions for total mass and density were markedly improved. This observation led to the conclusion that either 1) the average droplet size or distribution as measured by the IR camera at the center of the surface did not extend across the entire surface, or 2) the contact angle of the droplets on the surface was not as we assumed. This second point seems particularly likely. As described in Chapter 3, the hydrophilic surface's contact angle degraded over time, leading to a higher contact angle than was initially measured when the surface was created. A higher contact angle with the same projected radius would indicate more droplet volume and thus more mass.

Similarly, Haque and Betz (2018) observed degradation of a hydrophobic coating which was also used as the hydrophobic coating in this study. Contact angle on their surface decreased by roughly 10° over about 260 days / 70 frost tests. A decrease in contact angle would indicate a smaller mass for the same projected radius. Lastly, the contact angle of the bare aluminum surface is unknown. 75° was assumed for this study based on observation of reasonable values in the literature ranging from about 60° to 90°, but 75° may not be entirely suitable for this surface. Figures 5.3 through 5.5 below give examples for validating the crystal growth and frots layer growth stages on the bare aluminum surface when the droplet growth stage was bypassed. Based on the two points discussed above, the projected droplet radius and contact angle used as inputs varied by ± 0.025 mm and ± 5°, respectively, within their estimated uncertainties. When this was done, the values for frost density after it had stopped drastically decreasing were mainly predicted within the experimental error. However, predictions for test conditions 2 were slightly outside the range. Values for frost thickness were within the experimental uncertainty range at the early frosting stages, though they sometimes deviated from this range as time went on.



(a) (b)

Figure 5.3: Model predictions vs. measured data for (a) frost thickness and (b) frost density during crystal growth and frost layer growth for test conditions 1 on the bare aluminum surface.



(a) (b)

Figure 5.4: Model predictions vs. measured data for (a) frost thickness and (b) frost density during crystal growth and frost layer growth for test conditions 2 on the bare aluminum surface.

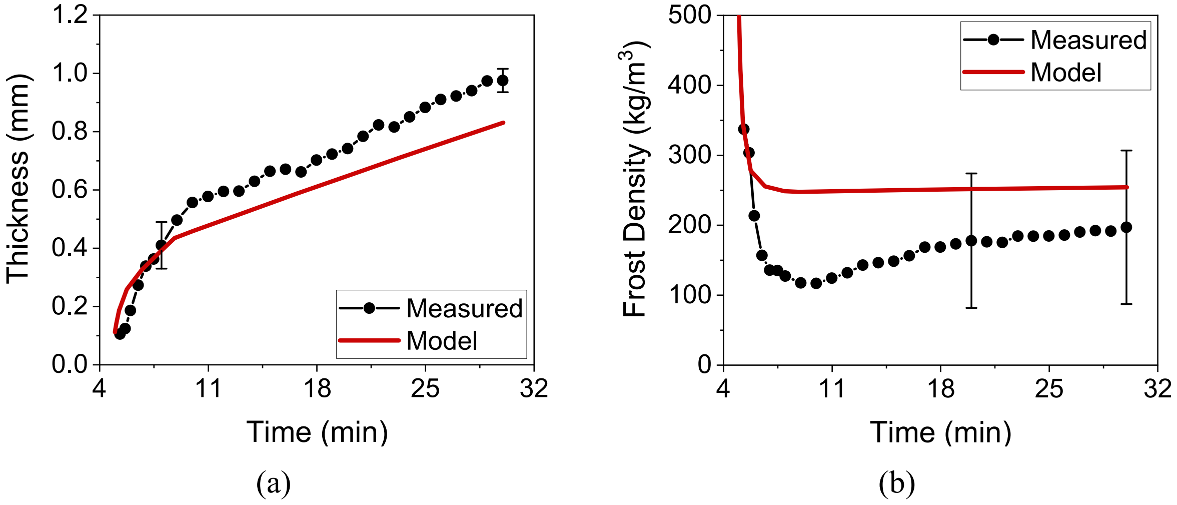


Figure 5.5: Model predictions vs. measured data for (a) frost thickness and (b) frost density during crystal growth and frost layer growth for test conditions 3 on the bare aluminum surface.

5.4 Model Validation – Overall Three Stage Frost Model

This section presents comparisons of predictions from the overall 3-stage model with measured data gathered during the current study and experimental data from two other literature sources. All data were gathered for flat, horizontal surfaces under forced convection conditions.

5.4.1 Current Research

Model validation was first performed using data for cleaned surfaces gathered during the experimental portion of this research and presented in Chapter 3. Figures 5.6, 5.7, and 5.8 compare model predictions to measured data of frost thickness and density for the bare aluminum surface at test conditions 1, 2, and 3 from Table 3.2, respectively. Arrows in Figures 5.6 and 5.7 indicate the various modeling stages. These figures correspond to Figures 5.3, 5.4, and 5.5, except that the droplet growth model was used to generate initial conditions for the crystal growth model instead of being bypassed. The results between the two sets of figures are similar but have some minor differences. Primary differences occurred in the density predictions.

Starting with the droplet model led to higher density predictions and more error at higher surface temperatures (test conditions 1 and 2). In contrast, at the lower surface temperature (test conditions 3), the density predictions were lower and closer to the measured values. Frost thickness predictions did not change much, though they were slightly better for test conditions 1 and slightly worse for test conditions 2 and 3. The main reasons for differences in the predictions when the droplet model was used were differences in the mass at the start of the model's crystal growth stage. Seemingly minor inaccuracies in droplet diameter at the end of droplet growth could lead to large over-predictions in the frost layer's overall density, especially in the early stages of frosting.

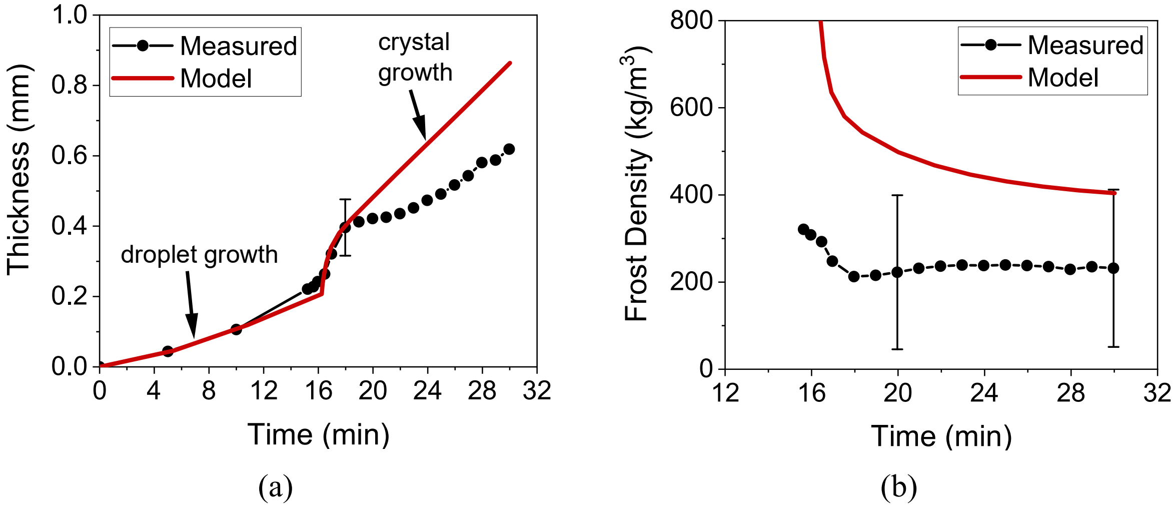
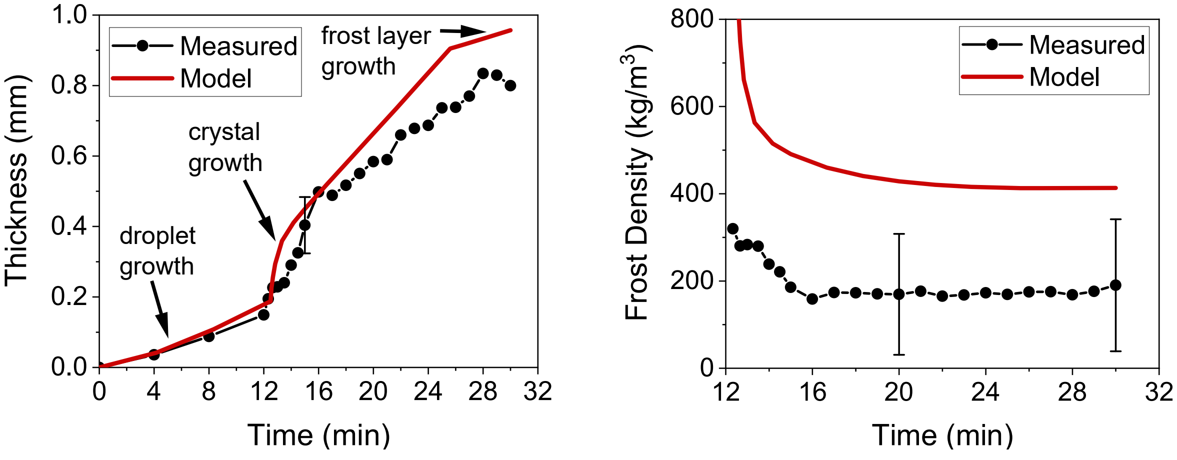
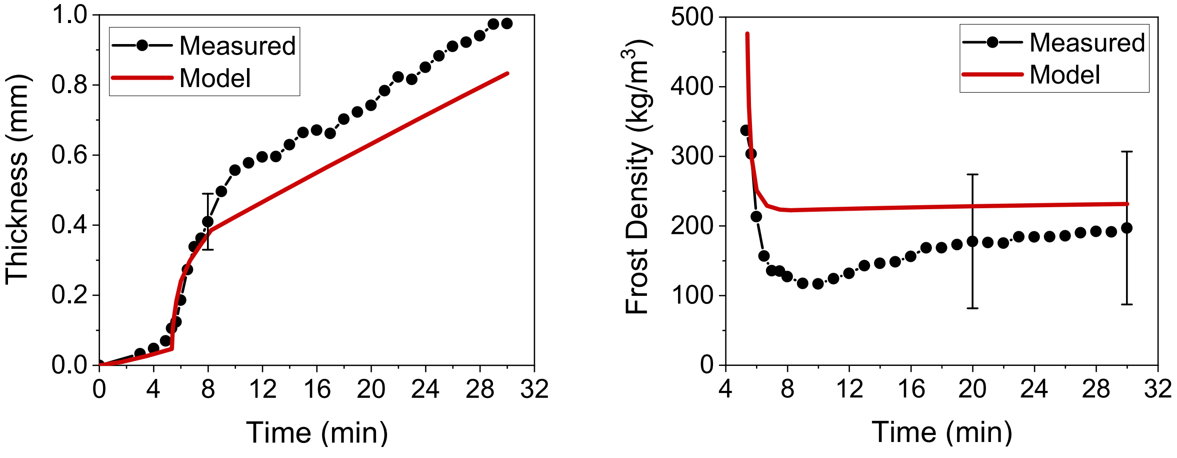


Figure 5.6: Model predictions vs. measured data for (a) frost thickness and (b) frost density for overall frost layer growth for test conditions 1 on the bare aluminum surface.



(a) (b)

Figure 5.7: Model predictions vs. measured data for (a) frost thickness and (b) frost density for overall frost layer growth for test conditions 2 on the bare aluminum surface.

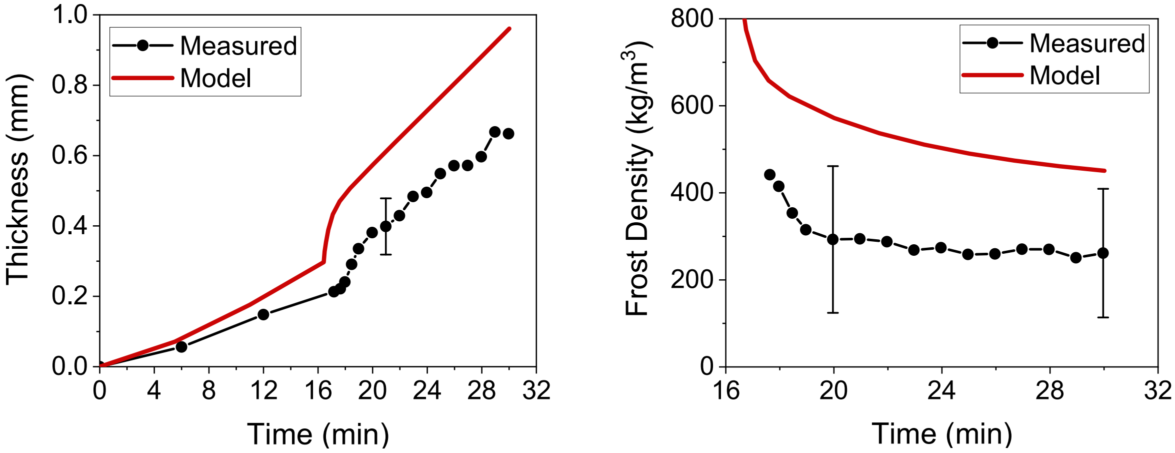


(a) (b)

Figure 5.8: Model predictions vs. measured data for (a) frost thickness and (b) frost density for overall frost layer growth for test conditions 3 on the bare aluminum surface.

Figures 5.9, 5.10, and 5.11 compare model predictions to measured data for test conditions 1, 2, and 3 on the hydrophobic surface. Density results were similar to those for the bare aluminum surface, where it was over-predicted for all three test conditions but only severely over-predicted for test conditions 1 and 2. Frost thickness was over-predicted for test conditions

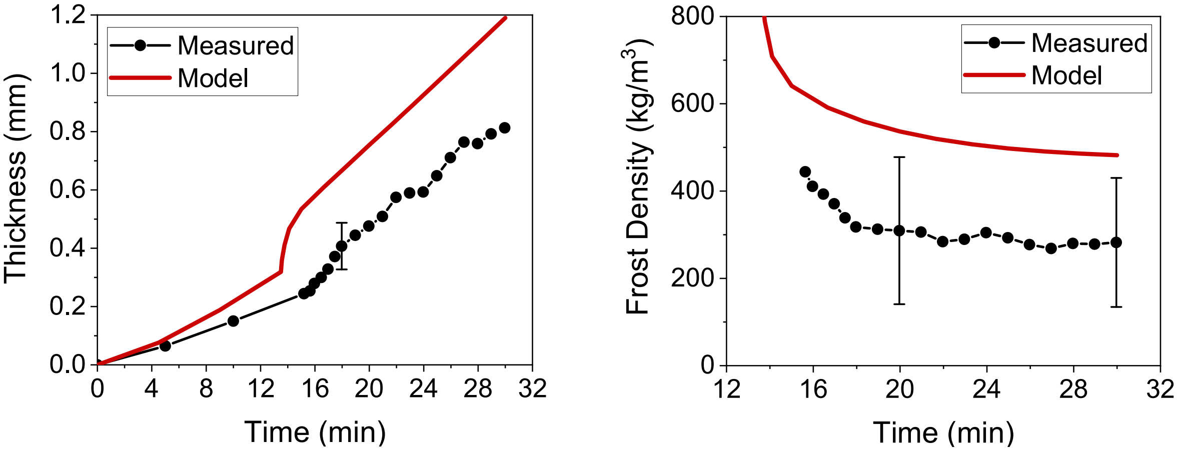
1 and 2, though this was primarily because droplet size (and thus initial thickness) was over-predicted. The thickness trends' shapes were close to the measured trends, but the crystal growth model's starting thickness values were too high. Frost thickness for test conditions 3 was well predicted, though it appears that predictions may deviate from measured values if compared for frosting times longer than 30 minutes.



(a)

(b)

Figure 5.9: Model predictions vs. measured data for (a) frost thickness and (b) frost density for overall frost layer growth for test conditions 1 on the hydrophobic surface.



(a)

(b)

Figure 5.10: Model predictions vs. measured data for (a) frost thickness and (b) frost density for overall frost layer growth for test conditions 2 on the hydrophobic surface.

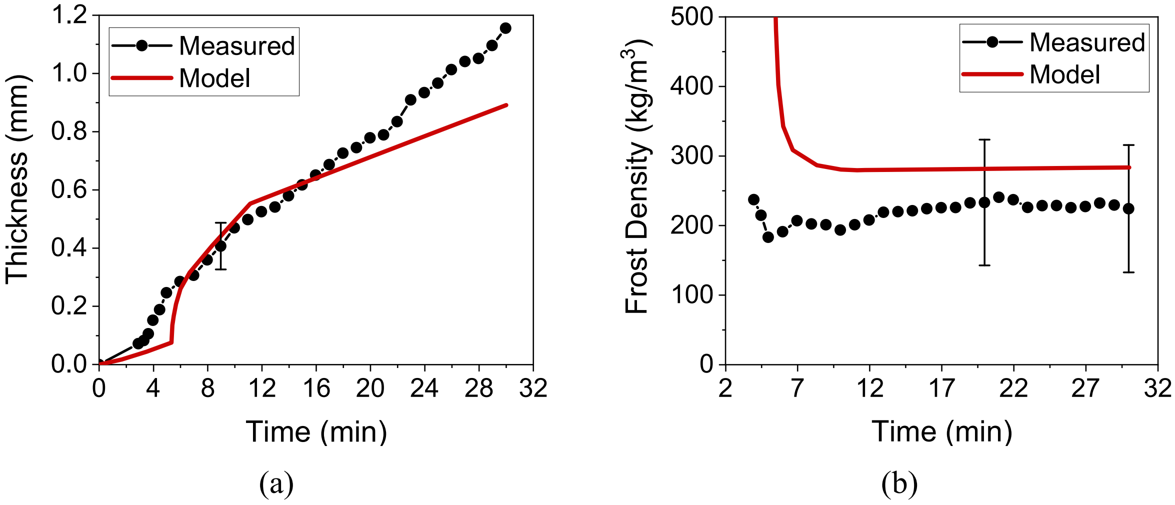


Figure 5.11: Model predictions vs. measured data for (a) frost thickness and (b) frost density for overall frost layer growth for test conditions 3 on the hydrophobic surface.

Figures 5.12, 5.13, and 5.14 compare model predictions to measured data for test conditions 1, 2, and 3 on the hydrophilic surface. Frost thickness predictions were better than for the other two surfaces, though thickness was over predicted for test conditions 2 and under predicted for test conditions 3 after the measured freezing time. Frost density predictions were also considerably better for this surface than the other two, with all predicted values falling well within the experimental uncertainty once density had stopped rapidly decreasing.

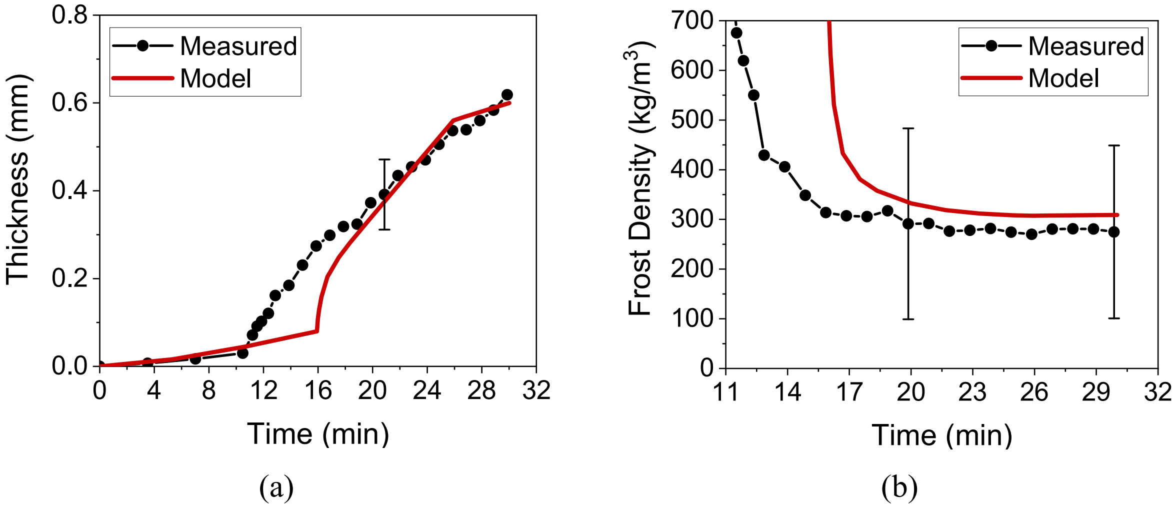
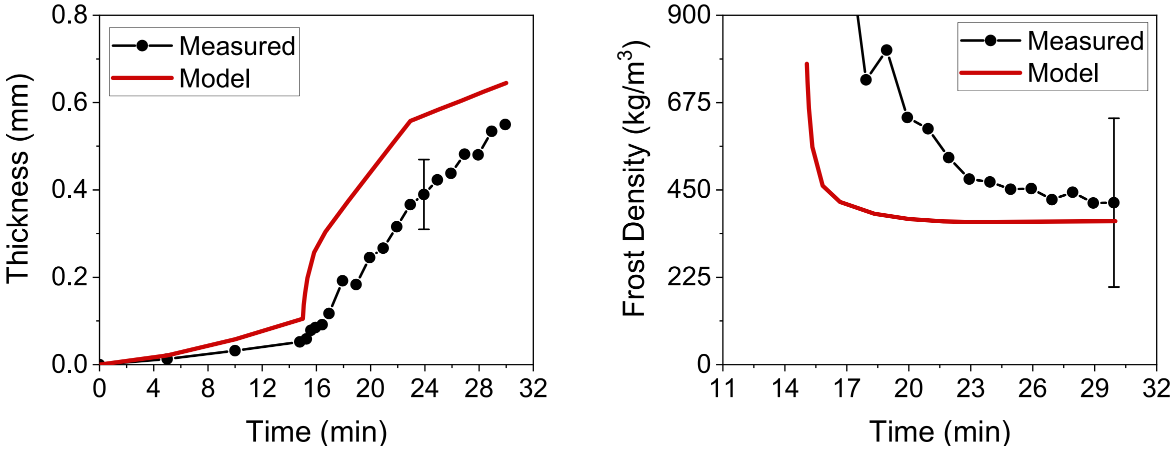
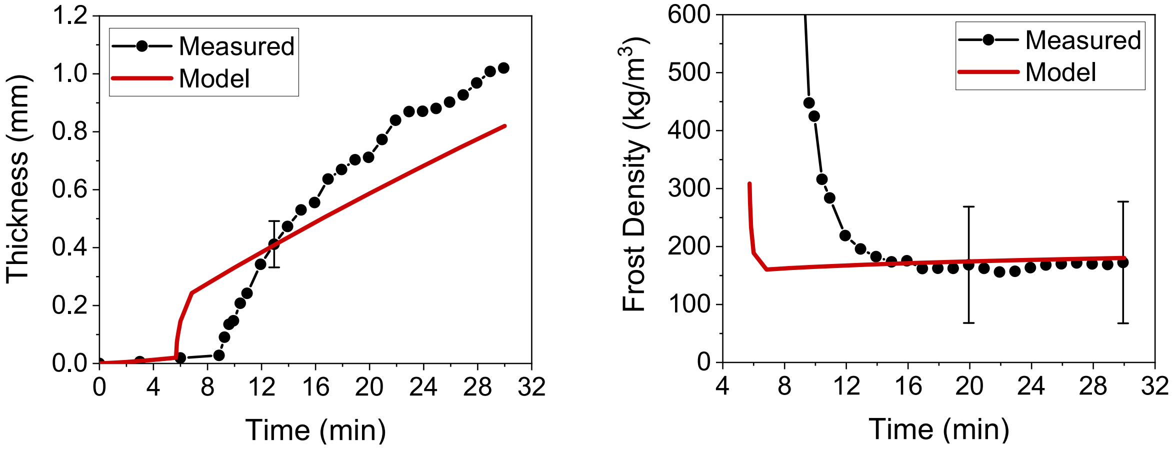


Figure 5.12: Model predictions vs. measured data for (a) frost thickness and (b) frost density for overall frost layer growth for test conditions 1 on the hydrophilic surface.



(a) (b)

Figure 5.13: Model predictions vs. measured data for (a) frost thickness and (b) frost density for overall frost layer growth for test conditions 2 on the hydrophilic surface.



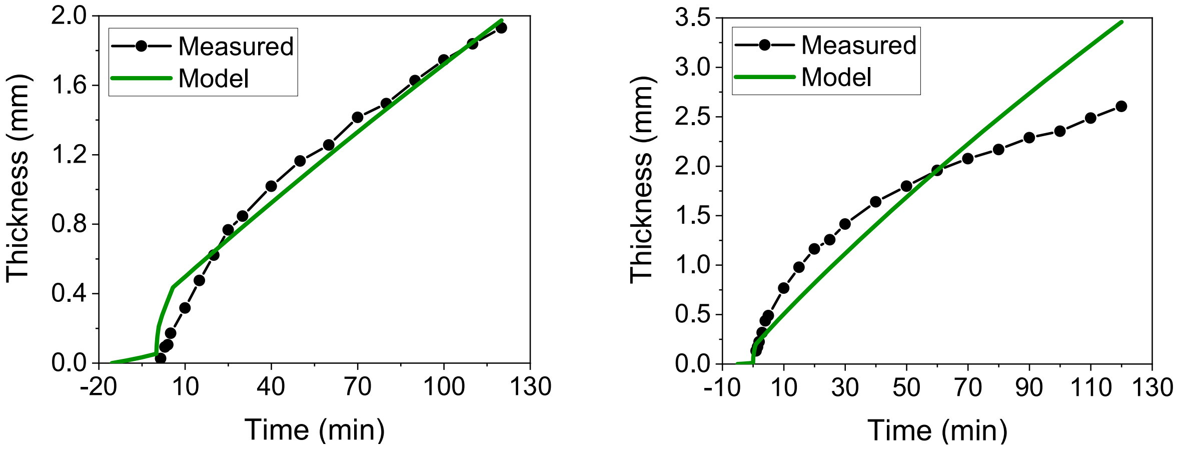
(a) (b)

Figure 5.14: Model predictions vs. measured data for (a) frost thickness and (b) frost density for overall frost layer growth for test conditions 3 on the hydrophilic surface.

5.4.2 Hermes et al. (2009)

Figure 5.15 compares model predictions to measured values of frost thickness from Hermes et al. (2009). In this study, frost growth and densification were studied on horizontal flat surfaces under convective airflow. The surface investigated was a square bare aluminum plate with dimensions 10 x 10 cm; for model validation, a contact angle of 75° was assumed for this surface. An assumption had to be made about when the authors considered the test to have begun (time 0) to compare model results to the data. There was no indication in this paper whether that was when airflow started, when surface cooling began, when ice was first observed, or whether there was some other criterion. However, in a later publication by the same research group (Hermes et al., 2019), they stated that at time 0, a plastic film, which had been covering the plate to impede frost growth, was removed. They also stated that condensation often occurred before this film was removed. Also, in the experimental data depicted in Figure 5.15, it appears that they do not take droplet growth into account in the thickness plots, even for their test with the cold surface temperature at -4 °C, where droplet growth should have had significant effects. For all these reasons, the assumption adopted for this model validation is that time 0 was the point at which ice crystals first appeared on the frozen droplets. When this assumption was made, the model predicted frost thickness very well, especially when the surface temperature was -4 °C (Figure 5.15(a)). Higher deviations were observed for the case of lower surface temperature (Figure 5.15(b)), particularly after one hour of frost growth.

Model predictions were also compared to measured values after one hour of testing for twelve sets of test conditions to more comprehensively validate the model. Again, the assumption was that time 0 for the measured data was when droplets froze and ice crystals appeared. Even though the current model focused primarily on the early stages of frosting, i.e., within the first 30 minutes, this validation was helpful since so many different measured variables were available for the same experimental run. The sets of experimental conditions used for validation are presented in Table 5.2.



(a) (b)

Figure 5.15: Comparison of model predictions with individual droplet profiles from Hermes et al. (2009). (a) T_s = -4 °C, V = 0.7 m/s, T_a = 16 °C, RH = 80%. (b) T_s = -8 °C, V = 0.7 m/s, T_a = 16 °C, RH = 80%.

Table 5.2: Twelve sets of experimental conditions from Hermes et al. (2009) used for model validation

Test #	**T_∞ [°C]**	**RH [%]**	**T_s [°C]**	**V [m/s]**
D-1	22.0	80	-15.0	0.7
D-2	22.0	80	-10.0	0.7
D-3	22.0	80	-5.0	0.7
D-4	22.0	50	-15.0	0.7
D-5	22.0	50	-10.0	0.7
D-6	22.0	50	-5.0	0.7
D-7	16.0	80	-15.0	0.7
D-8	16.0	80	-10.0	0.7
D-9	16.0	80	-5.0	0.7
D-10	16.0	50	-15.0	0.7
D-11	16.0	50	-10.0	0.7
D-12	16.0	50	-5.0	0.7

Table 5.3 presents model results for frost surface temperature, frost mass, frost thickness, and frost density at the test conditions from Table 5.2, as well as the associated errors on each value. Frost surface temperature was predicted within ±2 °C for surface temperatures equal to -10

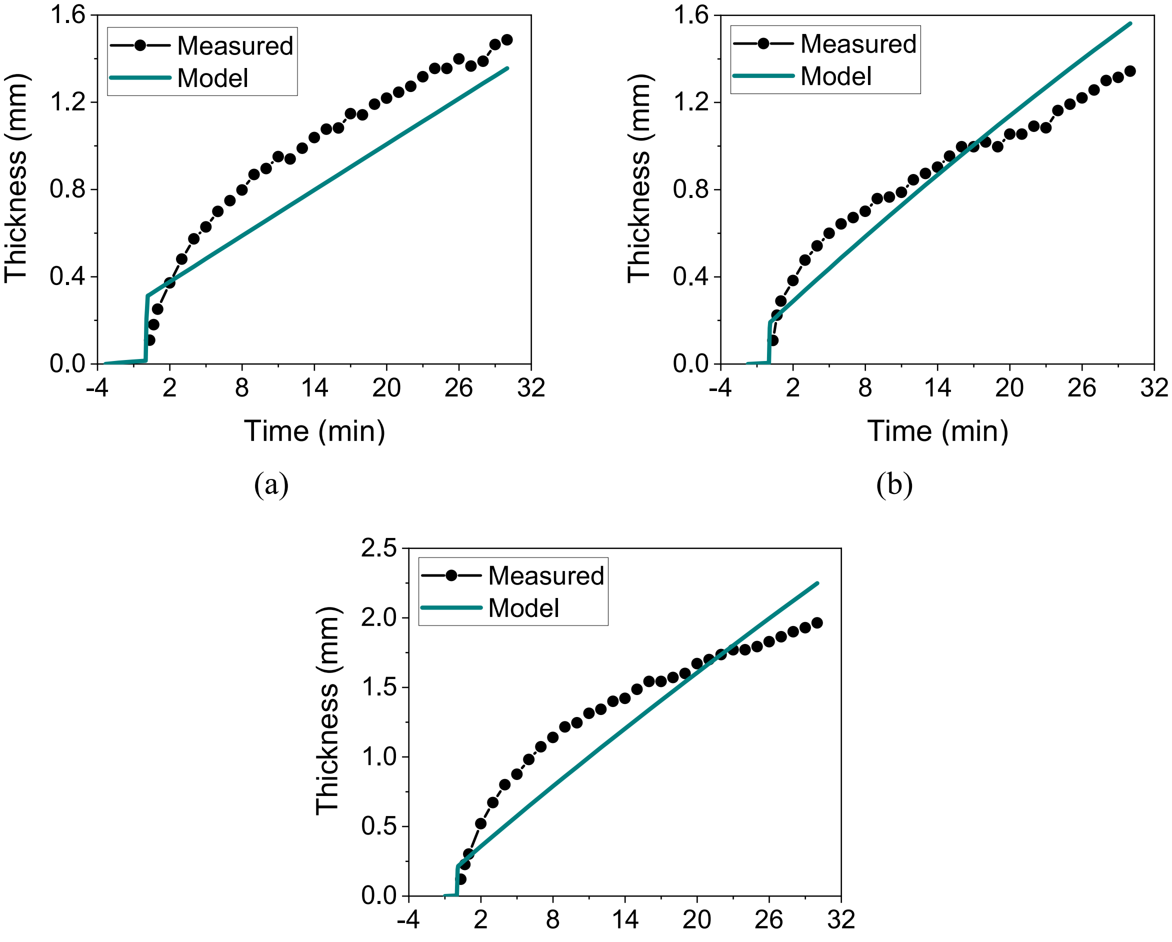
and -5 °C but was significantly over-predicted for 3 of the 4 cases with surface temperatures of -15 °C. These over-predictions at low temperatures were due to two factors: over-prediction of frost thickness and under-prediction of frost density. These over and under predictions, respectively, were worst at the lowest surface temperatures. Combined, they produced predictions for a thicker, more insulating frost layer than was measured, leading to predictions of frost temperature that were much higher than the actual values. Frost mass was predicted within ±25% for 10 of the 12 cases, with the largest over-predictions occurring when the surface temperature was -5 °C. These over-predictions were due to the longer freezing times associated with the highest surface temperatures. Longer freezing times led to more mass deposited during droplet growth, and if freezing times were slightly over-predicted, it could have a significant effect on the total mass predicted. Frost thickness was predicted within ±50% for 9 of the 12 cases, with the worst over predictions occurring for the lowest surface temperatures and higher air temperatures. These results show some of the limitations of the model, which was developed primarily for data with surface temperatures larger than or equal to -10 °C and air temperatures less than 20 °C. Frost density was well predicted, with 10 of the 12 cases falling within the ±40% range. Test conditions did not seem to affect how well the predictions matched the measured values significantly.

Table 5.3: Results of using the three-stage model to predict data of frost surface temperature, frost mass, frost thickness, and frost density from Hermes et al. (2009) at the test conditions in Table 5.2

	Model Results				Error			
Test #	**T_{fs} [°C]**	**Mass [g]**	**Thickness [mm]**	**Density [kg/m^3]**	**T_{fs} [°C]**	**Mass [%]**	**Thickness [%]**	**Density [%]**
D-1	-0.63	4.05	4.71	86.1	4.27	6.6	68.8	-36.7
D-2	-1.45	4.05	3.39	119.6	1.55	5.2	57.7	-33.1
D-3	-0.55	4.17	2.06	202.3	0.05	14.9	28.0	-10.1
D-4	-3.79	2.29	3.73	61.4	3.31	7.5	57.4	-31.8
D-5	-4.23	2.26	2.36	95.8	0.57	7.1	33.3	-19.6
D-6	-2.73	2.22	1.32	168.3	-0.03	25.4	-4.3	31.1
D-7	-3.89	2.68	4.08	65.7	2.41	8.1	40.2	-23.0
D-8	-4.92	2.70	2.37	113.7	0.28	14.9	5.3	8.7
D-9	-3.34	2.88	1.35	214.2	-0.94	24.1	-20.6	56.8
D-10	-7.66	1.52	2.80	54.4	-0.06	-1.3	48.9	-33.7
D-11	-6.95	1.44	1.50	96.5	-1.65	8.3	-4.5	13.9
D-12	-3.89	1.52	0.94	162.3	-0.49	36.9	-5.1	44.3

5.4.3 Cheng and Wu (2003)

Model predictions were also compared to measured thickness values during early frost growth from Cheng and Wu (2003). These data were gathered under forced convection conditions on a horizontal, square, bare copper surface with 56 x 56 mm dimensions. Once again, for model validation, a contact angle of 75° was assumed for the bare metal surface. Since no information was provided as to what time 0 of the tests represented, the assumption was made again that it was the point at which droplets froze and the first ice crystals appeared. This assumption made sense when observing the thickness trends of the measured data. Figure 5.16 compares model predictions to measured data of frost thickness during the first 30 minutes of frosting. The figure represents three different test conditions, with -11.3 °C ≤ T_s ≤ -5.1 °C, 41% ≤ RH ≤ 76%, 2.3 m/s ≤ V ≤ 4.2 m/s, 23.3 °C ≤ T_a ≤ 27.3 °C. Thickness values were predicted well for all test conditions at the early stages of frosting (the first 30 minutes). However, the trends matched best at high cold surface temperature and humidity (Figure 5.16(a)). No other data (such as frost mass, density, or surface temperature) were presented in this paper, so comparisons could only be made with frost thickness values.



(a)

(b)

(c)

Figure 5.16: Comparison of model predictions with individual droplet profiles from Cheng and Wu (2003). (a) T_s = -5.1 °C, V = 2.3 m/s, T_a = 27.1 °C, RH = 71%. (b) T_s = -7.2 °C, V = 4.2 m/s, T_a = 27.3 °C, RH = 41%. (c) T_s = -11.3 °C, V = 2.3 m/s, T_a = 23.3 °C, RH = 76%.

5.4.4 Hermes et al. (2019)

The model response to changing surface wettability was validated using data from Hermes et al. (2019). These data were gathered under forced convection conditions on test plates with four different surface types and dimensions of 60 mm in width and 120 mm in length in the airflow direction. As for the validation with data from Hermes et al. (2009) above, time 0 was assumed to be the point at which droplets froze and crystals began growing on the ice beads. Figure 5.17 compares predictions of the current model (solid lines, open symbols) with measured data from the experiments of Hermes et al. (2019) (closed symbols). The air temperature and relative humidity values were kept constant for all experimental and simulated runs at 5 °C and 80%, respectively. The values of air velocity varied between 0.97 and 1.12 m/s for the experimental runs, and the values of cold surface temperatures fell between -10.9 °C and -9.6 °C. The model used the exact inputs for a given experimental run when simulating the results, so the input values for environmental parameters were not exactly the same for each simulated frost thickness curve in Figure 5.17. The measured data indicated almost identical values for frost thickness between the surfaces, especially below 30 minutes of frosting. Model results showed similar results for all surfaces; however, all values were over-predicted at times below 30 minutes. The total spread in the thickness values at 60 minutes was 0.23 mm for the model predictions, which was very similar to the 0.20 mm spread for the experimental data. This figure indicates that, at least for these test conditions, the model predicted negligible differences in frost thickness between surfaces with different contact angles, agreeing with Hermes et al.'s (2019) measured results.

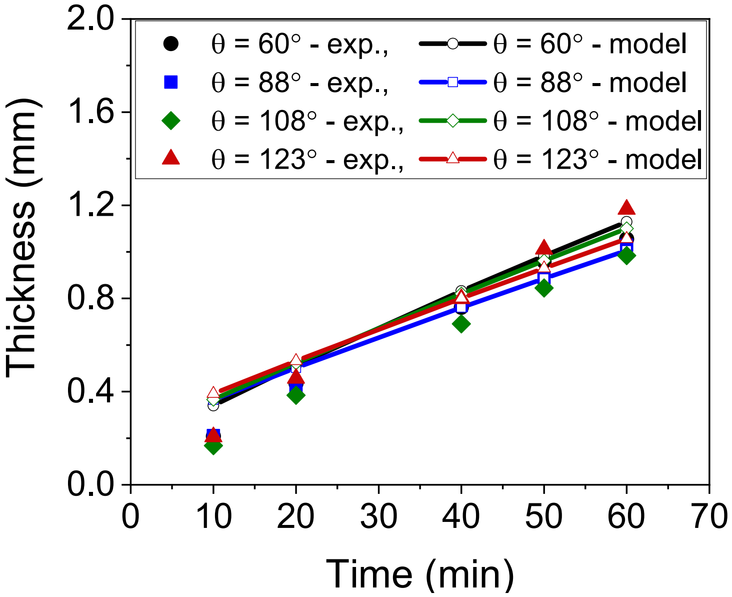


Figure 5.17: Comparisons between model predictions and measured data from Hermes et al. (2019). 60° surface: T_s = -10.3 °C, V = 0.97 m/s, 88° surface: T_s = -9.6 °C, V = 0.97 m/s, 108° surface: T_s = -10.9 °C, V = 1.04 m/s, 123° surface: T_s = -9.9 °C, V = 1.12 m/s.

5.4.5 Rahimi et al. (2015)

Model predictions were also compared to frost thickness data for surfaces of different wettability from Rahimi et al. (2015). These data were gathered under forced convection conditions on square surfaces with 15 mm to a side. They used four surfaces with different wettability types, and the air temperature and air velocity were held constant at 17 °F and 0.58 m/s, respectively. Table 5.4 presents model results and their associated error at 100% RH for the different surface types and temperatures investigated in Rahimi et al. (2015). It was stated in their paper that these results were gathered 5 minutes into the frosting process. However, no information was given as to whether the condensation period occurred before or after time 0. Because of this, results were presented for both assumptions. The column for frost thickness at 5 min assumes time 0 is at the beginning of condensation, while the column for t_f + 5 min assumes that time 0 was the instant when the droplets froze. Based on the results presented in Table 5.4, it appears that neither assumption worked well for all of the data. If time 0 was at the beginning of condensation, the most hydrophobic surface's thickness was predicted best, while that on the bare and hydrophilic surfaces were severely under-predicted. If time 0 was at the beginning of ice deposition, the thickness on the bare surface was predicted well, while that on the most hydrophobic surface was drastically over-predicted. Frost thickness consistently increased as surface temperature decreased for the measured data. It did the same for the modeled data after the condensation period when time 0 was set to be at the beginning of condensation. If time 0 was at the beginning of ice deposition, this trend did not always hold. This outcome occurred because results were reported at 5 minutes after droplet freezing. The cold surface temperature significantly impacted freezing time, so results were reported at different frosting times.

Table 5.5 compares model predictions with measured data for the same environmental conditions as Table 5.4 but at a frosting time of 10 minutes. Similar results were obtained for the various assumptions for time 0 for the experimental runs. However, the over and under-predictions were less drastic at 10 minutes since the droplet growth stage's effects on the frost thickness had begun to be tempered by this point. For both assumptions, model predictions of frost thickness agreed with experimental trends, consistently increasing as cold surface temperature decreased.

Table 5.4: Comparison of model predictions to data from Rahimi et al. (2015) at 100% RH and 5 minutes into frosting

		Measured	**Model**			
θ (°)	T_s (°C)	δ_f (mm)	δ_f (mm) 5 min	Error (%) 5 min	δ_f (mm) $t_f + 5$ min	Error (%) $t_f + 5$ min
78.1	-4.5	0.616	0.046	-92.5	0.594	-3.6
78.1	-6	0.719	0.041	-94.3	0.582	-19.1
78.1	-8	0.850	0.382	-55.1	0.601	-29.3
78.1	-10	1.001	0.396	-60.4	0.580	-42.1
36.9	-5	0.366	0.023	-93.7	0.537	46.7
36.9	-6	0.421	0.021	-95.0	0.527	25.2
36.9	-8	0.600	0.311	-48.2	0.559	-6.8
36.9	-9	0.779	0.332	-57.4	0.562	-27.9
36.9	-10	0.803	0.346	-56.9	0.554	-31.0
116.1	-6	0.310	0.062	-80.0	0.634	104.5
116.1	-8	0.437	0.417	-4.6	0.635	45.3
116.1	-10	0.600	0.437	-27.2	0.604	0.7
116.1	-11	0.707	0.466	-34.1	0.620	-12.3
123.9	-4	0.075	0.072	-4.0	0.675	800.0
123.9	-6	0.175	0.062	-64.6	0.637	264.0
123.9	-8	0.282	0.421	49.3	0.639	126.6
123.9	-10	0.421	0.440	4.5	0.606	43.9
123.9	-11	0.501	0.469	-6.4	0.622	24.2

Table 5.5: Comparison of model predictions to data from Rahimi et al. (2015) at 100% RH and 10 minutes into frosting

		Measured	Model			
θ (°)	T_s (°C)	δ_f (mm)	δ_f (mm) 10 min	Error (%) 10 min	δ_f (mm) t_f + 10 min	Error (%) t_f + 10 min
78.1	-4.5	0.792	0.100	-87.4	0.738	-6.8
78.1	-6	0.903	0.504	-44.2	0.761	-15.7
78.1	-8	1.089	0.631	-42.1	0.846	-22.3
78.1	-10	1.251	0.701	-44.0	0.880	-29.7
36.9	-5	0.582	0.337	-42.1	0.719	23.5
36.9	-6	0.705	0.438	-37.9	0.731	3.7
36.9	-8	0.911	0.593	-34.9	0.836	-8.2
36.9	-10	1.101	0.690	-37.3	0.891	-19.1
116.1	-6	0.562	0.563	0.2	0.797	41.8
116.1	-8	0.752	0.662	-12.0	0.859	14.2
116.1	-10	1.010	0.714	-29.3	0.878	-13.1
116.1	-11	1.196	0.773	-35.4	0.923	-22.8
123.9	-4	0.186	0.150	-19.4	0.794	326.9
123.9	-6	0.364	0.567	55.8	0.800	119.8
123.9	-8	0.582	0.666	14.4	0.862	48.1
123.9	-10	0.950	0.716	-24.6	0.879	-7.5
123.9	-11	1.089	0.774	-28.9	0.923	-15.2

Table 5.6 presents comparisons of model predictions with measured data from Rahimi et al. (2015) at -6 °C in order to compare results better when only the contact angle of the surface was changed. At both 5 and 10 minutes, the measured frost thickness was highest for the bare surface, then decreased for the hydrophilic surface and further decreased as hydrophobicity increased. The total range of frost thickness between the surface types was about 0.54 mm for both frosting times. The model showed different results, however. It predicted that the most hydrophobic surface had the highest frost thickness, and the thickness slightly decreased as the contact angle decreased. At 10 minutes of frosting, the range of frost thickness values predicted

by the model was about 0.13 mm across all four surface types. This small range of frost thicknesses agrees with the results of Hermes et al. (2019), discussed in the previous section. Based on Table 5.6, the model could not predict the trends of thickness vs. contact angle observed by Rahimi et al. (2015). However, this study's data is not well understood, and the observations they reported are contrary to expectations. If droplet condensation and freezing had occurred according to the observations presented in Chapter 3 and the model from Chapter 4 of this dissertation, droplets would have been taller and narrower on the hydrophobic surfaces than the hydrophilic surfaces. This discrepancy would lead to different crystal distributions and slightly larger thickness values on the hydrophobic surface, especially at 5 minutes of frosting when the droplets still substantially affected the overall frost layer properties. Rahimi et al. inserted the test plates after everything in the test section had come to the proper conditions when gathering this data. This procedure may have caused droplets on the surface to freeze immediately before having an opportunity to grow. Surface wettability effects of frost growth should then be essentially negated, leading to very similar frost layers on all surfaces. Because of the uncertainty in how these results were obtained, it is unclear in what areas the model fails when attempting predictions of these data trends.

Table 5.6: Comparison of model predictions with measured values of frost thickness at -6 °C from Rahimi et al. (2015)

	5 minutes			
θ (°)	36.9	78.1	116.1	123.9
measured δ_f (mm)	0.421	0.719	0.310	0.175
model δ_f (mm)	0.021	0.041	0.062	0.062
	10 minutes			
θ (°)	36.9	78.1	116.1	123.9
measured δ_f (mm)	0.705	0.903	0.562	0.364
model δ_f (mm)	0.438	0.504	0.563	0.567

5.4.6 Summary of Overall Model Validation

In summary, the model could predict frost characteristics for forced convection data from the current study and other literature sources. It was compared to data for bare metal surfaces and surfaces treated to change the surface wettability. It performed especially well for cases of high surface temperature and low air temperature. Deviations from measured values increased as the air temperature rose above 16 °C and the cold surface temperature dropped below -10 °C. It agreed with conclusions made by Hermes et al. (2019). They observed that contact angle had a negligible effect on frost thickness, predicting nearly the same spread of thickness values across all surface types at 60 minutes of frosting as were measured. In contrast, it could not predict data from Rahimi et al. (2015), whose findings contradicted those of Hermes et al. (2019). These data indicated stratification in thickness values by surface type and had a much larger spread in thickness values than the model predicted.

5.5 Results of the Overall Frost Model

Figures 5.18 – 5.21 present the results of a parametric study for the thickness and density as air relative humidity, air velocity, cold surface temperature, and surface contact angle were varied. The baseline test conditions were the same as for the parametric study presented in Section 4.3: T_a = 5 °C, RH = 70%, V = 2.5 m/s, T_s = -5 °C, and θ = 90°. The surface's length and width were held constant at 30 mm, while the channel's width and height were set to 30 mm and 5 mm, respectively. Since the current study is primarily concerned with the early stages of frost growth, the frost layer predictions were made only for the first 60 minutes of frosting.

Figure 5.18 shows the effects of air humidity on the model predictions. Figure 5.18(a) indicates that the air's humidity level affected the thickness of the simulated frost layer and the stage of the model the simulation was in at 60 minutes into the frosting. When the relative humidity was 50%, the droplet growth and crystal growth stages were long, leading to the model still being in the crystal growth stage at 60 minutes. As relative humidity increased, the model progressed through its stages more quickly. At 70% RH, the simulation switched from crystal growth to frost layer growth at 34 minutes, while it had moved to the frost layer growth stage by 20 minutes when the relative humidity was 90%. Additionally, Figure 5.18(b) shows that increasing relative humidity increased the simulated density. This increase was due to increased mass transfer driving potential, which led to more mass depositing on the surface. This conclusion is evident since even though the thickness values were similar at 60 minutes for the 70% and 90% RH cases, the density at 90% RH was significantly higher at this time than it was at 70% RH.

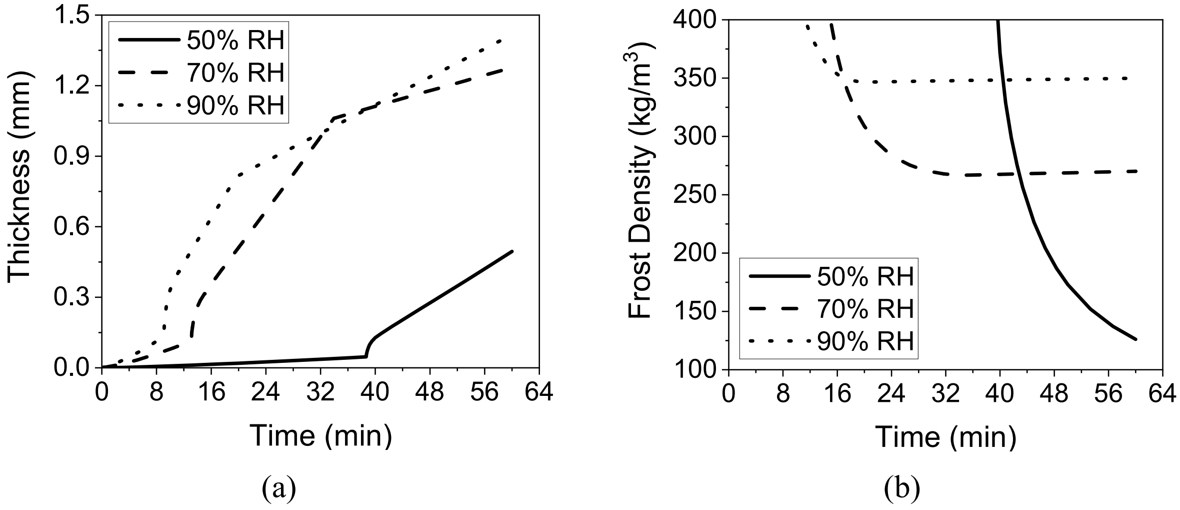


Figure 5.18: Model predictions of (a) thickness and (b) frost density as relative humidity varied.

Figure 5.19 presents simulation results of frost thickness and density when the humid air stream's velocity was varied. Increasing the air velocity affected the frost layer's growth primarily by decreasing the model stages' lengths. Figure 5.19(a) shows that the frost thickness profile slopes had nearly identical shapes for all three velocity values but were offset due to the differences in droplet freezing time and the transition from crystal growth to frost layer growth. Figure 5.19(b) indicates that density values were very similar for all three surfaces, though the lowest velocity produced a slightly lower density.

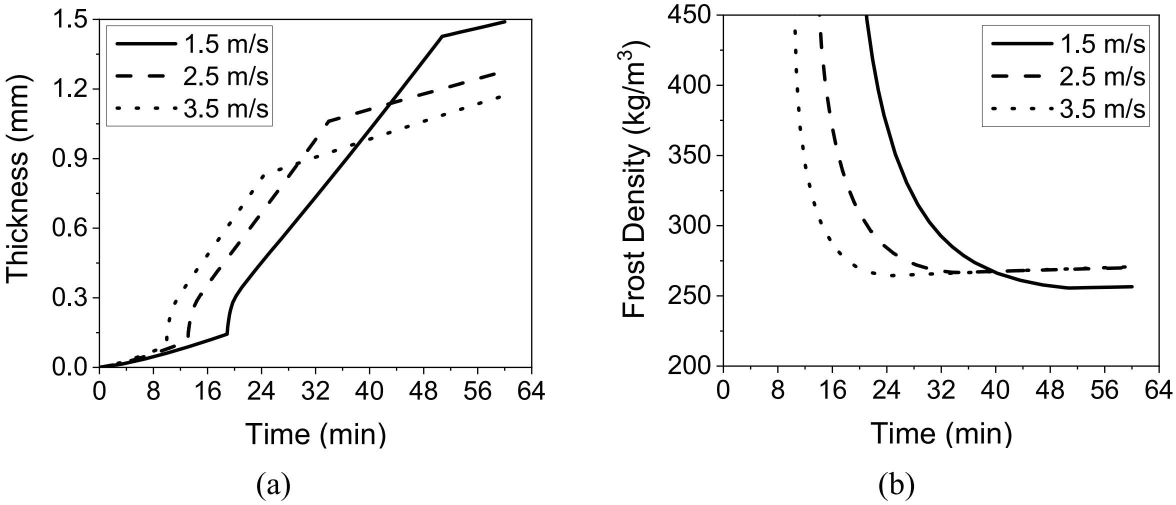
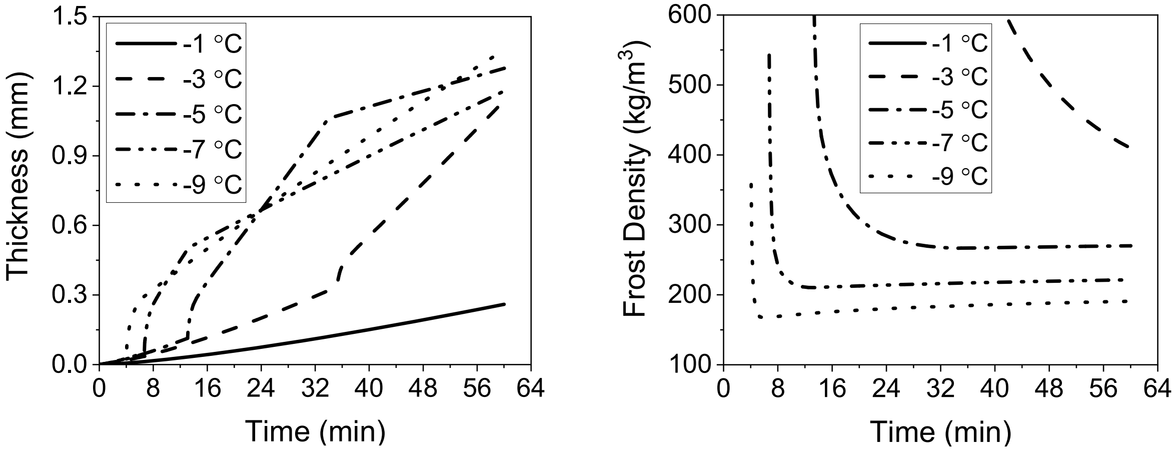


Figure 5.19: Model predictions of (a) thickness and (b) frost density as air velocity varied.

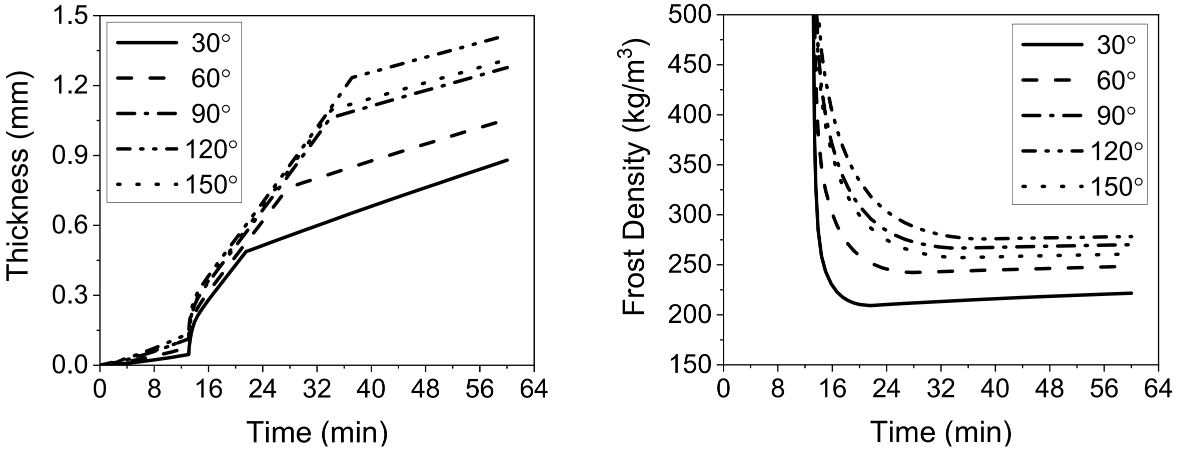
Figure 5.20 shows the model response to changing the temperature of the cold surface. At the high surface temperatures of -1 °C, the droplet freezing time was longer than 60 minutes, so the model was still in the droplet growth stage at the end of the simulation. It is evident even from results during the droplet growth stage that decreasing the surface temperature increased the mass transfer driving potential, however, since the thickness was higher after 30 minutes for the -3 °C case than for the -1 °C case. Progression through the three stages of the model became quicker as the cold surface temperature decreased. When the surface temperature was -3 °C, the simulation ended in the crystal growth stage, while it was in the frost layer growth stage for the -5, -7, and -9 °C cases. For the -5, -7, and -9 °C cases, frost thickness values were similar at 60 minutes because of the interplay between the various frosting stages. The case with the coldest surface temperature initially had the highest thickness. However, it then was overtaken by the other cases for which the simulation remained in the crystal growth stage for a more extended time. However, the thickness profile slope during the frost layer growth stage increased as surface temperature decreased. Eventually, the thickness for the coldest surface temperature overtook the others to maintain the largest thickness. This trend for the frost layer growth thickness slope is illustrated by comparing the -5, -7, and -9 °C cases. The thickness slope during frost layer growth was steepest at -9 °C, decreased at -7 °C, and was shallowest for -5 °C. The slopes' differences were significant enough that, if the simulation was allowed to run long enough, frost thickness would highest for the lowest surface temperature and lowest for the highest surface temperature. Figure 5.20(b) shows that no matter the frosting time, frost density consistently decreased as surface temperature decreased.



(a) (b)

Figure 5.20: Model predictions of (a) thickness and (b) frost density as the cold surface temperature varied.

Figure 5.21 presents the model results for frost thickness and density when the surface's contact angle was varied. Figure 5.21(a) shows that the thickness values were very similar for all contact angles until about 22 minutes into frosting when the most hydrophilic surface switched from the crystal growth stage to the frost layer growth stage and its thickness growth slowed considerably. The 60° surface switched model stages next, slowing its growth as well. It appeared that the thickness slope for the 30° surface was slightly shallower than for the 60° surface, so the two profiles would probably grow closer together again as time passed. Interestingly, the 120° surface remained in the crystal growth stage longest and had the highest frost thickness. Figure 5.21(b) also shows curious results. The 120° surface had the highest frost density. From there, density decreased both as contact angle decreased and as it increased. The decrease in density as the contact angle increased was expected. Droplets on hydrophobic surfaces had smaller area coverage values and did not fill up as much space in the overall frost layer due to their shapes as did the flatter droplets on hydrophilic surfaces. The decrease in density as contact angle decreased was unexpected, however. A further exercise of the model was needed to provide some explanation of these strange results.



(a)

(b)

Figure 5.21: Model predictions of (a) thickness and (b) frost density as contact angle varied.

While investigating the cause of the strange density predictions from Figure 5.21, it was discovered that there were significant discrepancies between the surfaces in the amount of mass deposited during the droplet growth stage. Less mass was deposited on the hydrophilic surfaces than on the hydrophobic surfaces, with the least mass deposited on the 30° surface. Data collected as part of the current study and data presented in literature sources indicated that similar mass deposition rates were present for all surface types if environmental conditions were held constant. The key model variable which accounted for surface contact angle during droplet growth was the growth exponent, μ (Equation 4.34). The correlation developed for this variable was based on experimental data from the current study and data presented in Tao et al. (1993). It was observed that altering this correlation altered significantly the amount of mass predicted by the model during droplet growth. The μ correlation's dependence on contact angle was altered so that similar amounts of mass were deposited for each surface type during droplet growth to check the model response. The altered correlation is presented as Equation 5.36.

$$\mu = 1.9 - 0.0168\,\Delta T + 0.0000187\,\theta^2 - 0.006225\,\theta \tag{5.36}$$

Figure 5.22 presents model results for different surface types when Equation 5.36 was used in place of Equation 4.34. The new results indicate that frost thickness values were nearly identical for all surface types, in agreement with Hermes et al. (2019). There was some stratification in density values, though the differences between surfaces were minor and mainly occurred at the extremes of the contact angle range. There were negligible differences in density for surfaces between 60° and 120°. However, the new density results seemed more reasonable. As expected, the lowest density was calculated at the highest contact angle, and the highest density was obtained for the lowest contact angle. These results obtained using Equation 5.36 seem reasonable. However, Equation 4.34 was developed using data gathered during laboratory experiments and should not be dismissed lightly. Because of this, Equation 4.34 was still used for the remainder of this study. Future investigations into droplet growth and coalescence behavior at subfreezing temperatures on surfaces with various contact angles would be helpful.

New data could improve the μ correlation's dependence on contact angle. Particular attention should be given to superhydrophilic surfaces since the most significant discrepancies in mass deposition occurred for the lowest contact angles, and there are still few data for droplet growth and freezing on superhydrophilic surfaces available in published literature.

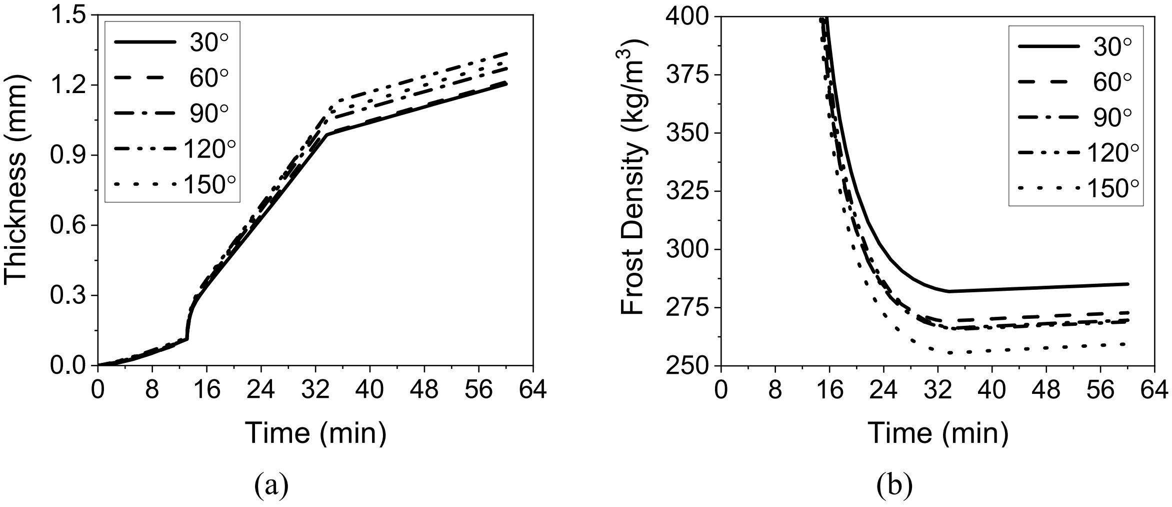


Figure 5.22: Model predictions with updated droplet growth exponent of (a) thickness and (b) frost density as contact angle varied.

5.6 Discussion about the Model's Ability to Capture the Physics of Crystal Growth

Like the droplet model, the crystal growth model presented in this chapter also had successes and failures that indicated which physical processes were captured and which were not. The model used theoretical methods to determine the volume change of a crystal due to mass transfer from the humid air stream, but determining volume change was not sufficient to obtain the new crystal dimensions. A method of calculating the crystal aspect ratio, and thus its dimensions, was developed. The expression's form was chosen to ensure that the crystal temperature changed in a way that made physical sense, not decreasing as the crystal grew. Aside from this physical limitation, the expression for aspect ratio was determined empirically.

Observations of the frost profile model validation results indicate that the frost thickness trends were often reasonably predicted. However, there were cases where absolute values of frost thickness were over or under-predicted or where the crystal growth modeling stage ended too early or too late. These frost thickness and stage transition predictions were both affected by the aspect ratio, so it is evident that physical processes occurred during experiments that were not captured by the proposed aspect ratio expression. Additionally, the assumption that crystals could be modeled as solid cylinders does not account for the crystal branching observed near the end of the crystal growth stage (or earlier for certain test conditions and surface types). The current study could not distinguish individual crystals on the droplets, and very little data were available in the literature to describe the branching, so a method of mathematically describing this phenomenon was not proposed here.

6. Model Segmentation

Model development for the baseline three-stage frost model, which predicted average frost characteristics for an entire cold surface with a single wettability type, was discussed in Chapters 4 and 5. The following sections describe extending that model to calculate frost characteristics for a cold surface divided into several segments. Two segmentation methods were applied: segmentation in the direction of airflow and segmentation by wettability type. The methods could be used individually or simultaneously.

6.1 Segmentation in the Direction of Air Flow

6.1.1 Model Description

This section describes how the model was segmented in the direction of airflow over a cold surface. It assumed a single contact angle for the entire surface and no segmentation in the direction perpendicular to the airflow since adjacent segments would have the same frost characteristics. Thus, the surface area of each segment is given by Equation 6.1, where N_{seg} is the total number of segments.

$$A_{seg} = W_{plate}\frac{L_{plate}}{N_{seg}} \tag{6.1}$$

Inlet air properties for a segment were set equal to the previous segment's outputs rather than specified by the user to perform calculations for the frost layer for sequential segments in the airflow direction. Outlet air temperature for a segment was calculated according to Equations 6.2 through 6.4.

$$q''_{sens} = h\left(T_{air,in} - T_f\right) \tag{6.2}$$

$$\dot{m}_{air,seg} = \rho_{air}\, W_{plate}\, \left(H_{ch} - \delta_f\right) V_{air} \tag{6.3}$$

$$T_{out,seg} = T_{in,seg} - \frac{q''_{sens}\, A_{seg}}{c_{p,air}\, \dot{m}_{air,seg}} \tag{6.4}$$

For a single segment in the airflow direction, the heat transfer coefficient was calculated using the average Nusselt number for laminar flow, as described in Chapter 4. For multiple segments, however, the local Nusselt number and heat transfer coefficient were used. These equations are given as Equations 6.5 and 6.6, with distance in the airflow direction expressed in Equation 6.7, where *n* represented the current segment for which calculations were being performed.

$$Nu_x = 0.332\, {Re_x}^{1/2}\, Pr^{1/3} \tag{6.5}$$

$$h_x = \frac{k}{x}\, Nu_x \tag{6.6}$$

$$x = \frac{(n + 0.5)}{N_{seg}}\, L_{plate} \tag{6.7}$$

The absolute humidity ratio at the outlet of a segment was calculated according to Equation 6.8. The humidity ratio was then used as an input into the CoolProp library to obtain the segment outlet's relative humidity.

$$\omega_{out,seg} = \omega_{in,seg} - \frac{\Delta m_w}{\dot{m}_{air,seg}\, \Delta t} \tag{6.8}$$

When the simulation reached the droplets' calculated freezing time, all segments switched from the droplet growth model to the crystal growth model. Similarly, when the average frost density over the entire cold surface reached a minimum value, all segments switched from the crystal growth model to the frost growth model. The flow chart in Figure 6.1 illustrates segmentation in the direction of air flow.

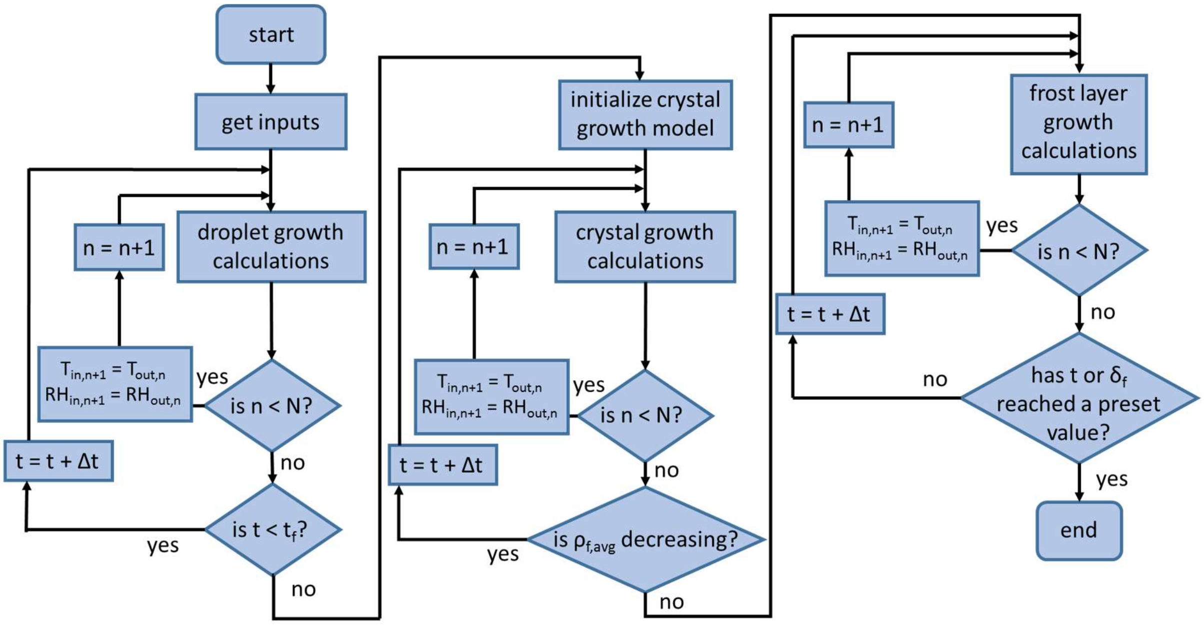

start
get inputs
droplet growth calculations
n = n+1
$T_{in,n+1} = T_{out,n}$
$RH_{in,n+1} = RH_{out,n}$
yes
is n < N?
no
t = t + Δt
is $t < t_f$?
yes
no
initialize crystal growth model
crystal growth calculations
n = n+1
$T_{in,n+1} = T_{out,n}$
$RH_{in,n+1} = RH_{out,n}$
yes
is n < N?
no
t = t + Δt
is $\rho_{f,avg}$ decreasing?
yes
no
frost layer growth calculations
n = n+1
$T_{in,n+1} = T_{out,n}$
$RH_{in,n+1} = RH_{out,n}$
yes
is n < N?
no
t = t + Δt
no
has t or δ_f reached a preset value?
yes
end

Figure 6.1: Flow chart for segmentation in the direction of airflow.

6.1.2 Segmentation Results and Validation

Figure 6.2 presents model results for thickness, density, and mass as functions of time when a cold surface was divided into three segments in the airflow direction. In Figure 6.2, Segment 1 refers to the segment at the plate's leading edge, while Segment 3 is the one at the trailing edge. The figure shows that thickness, density, and mass all decreased along the airflow direction. The decrease was sharp near the leading edge but became more gradual as the flow progressed across the plate.

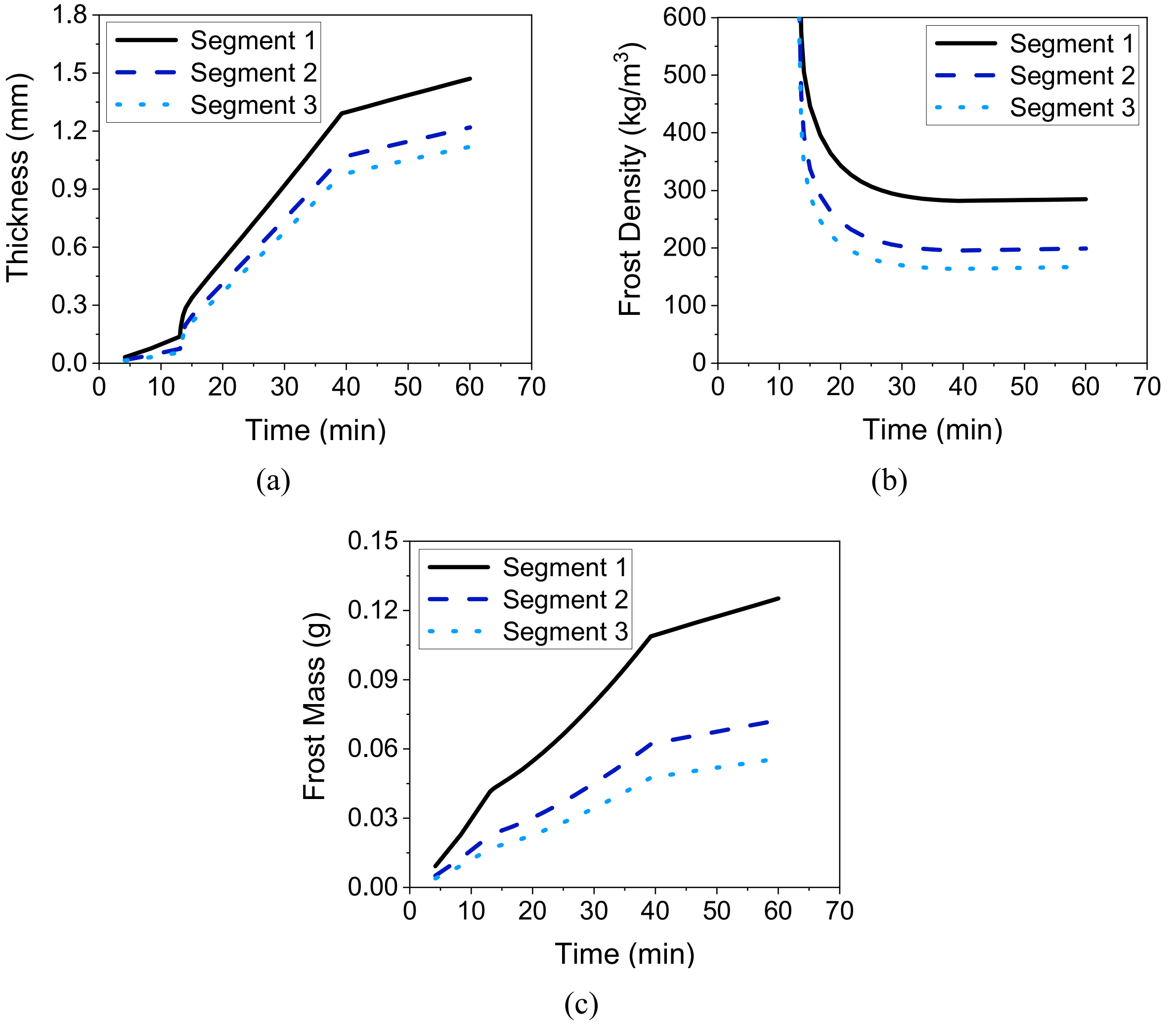


Figure 6.2: (a) Thickness, (b) density, and (c) mass as functions of time for three segments in the direction of air flow. T_a = 5 °C, RH = 70 %, V = 2.5 m/s, T_s = -5 °C, θ = 90°, L_{plate} = W_{plate} = 30 mm, $H_{channel}$ = 5 mm.

Table 6.1 presents sample calculations for variables of interest for all three segments from Figure 6.2 at the end of the droplet growth stage of the model, at the instant before droplet freezing. From these results, it appears that the heat and mass transfer coefficients were most responsible for the droplet size results. The heat and mass transfer coefficients decreased in the direction of airflow, and so did droplet size. Air temperature and humidity level also decreased along the plate, while the temperature at the top of the droplets (the interface temperature) changed very little. This decrease in driving potential combined with the decreasing heat and mass transfer coefficients to produce smaller droplets. The coalescence exponent, μ, increased in the direction of airflow since it was inversely related to air temperature difference. However, the increase was not enough to offset the significant disparity in heat and mass transfer coefficients between segments.

Table 6.1: Sample segmentation model calculations for the same inputs used to generate Figure 6.2. Values listed are at the instant before freezing (785 s).

#	Variables		Segment 1	Segment 2	Segment 3
1	heat transfer coefficient	h [W/m^2-K]	44.8	25.9	20.0
2	mass transfer coefficient	h_m [m/s]	0.0371	0.0214	0.0166
3	coalescence exponent	μ [-]	1.324	1.329	1.331
4	droplet growth rate	dr/dt [m/s]	3.97 x 10^{-8}	2.18 x 10^{-8}	1.62 x 10^{-8}
5	area coverage	AC [-]	0.692	0.654	0.636
6	droplet radius of curvature	r [mm]	0.300	0.173	0.132
7	droplet projected radius	r_{proj} [mm]	0.300	0.173	0.132
8	number of droplets	N_d [-]	735	2092	3467
9	interface temperature	T_i [°C]	-4.86	-4.95	-4.97
10	total frost mass	m_f [g]	0.042	0.023	0.017
11	initial frost thickness	δ_0 [mm]	0.138	0.075	0.056
12	outlet air temperature	$T_{air,out}$ [°C]	4.73	4.58	4.46
13	outlet air pressure	$P_{air,out}$ [Pa]	101,324.9	101,324.7	101,324.6

14	outlet humidity ratio	ω_{out} [g/kg]	3.645	3.568	3.512
15	outlet relative humidity	RH_{out} [-]	0.687	0.680	0.674

Table 6.2 presents calculations for all three segments at the end of the crystal growth stage, right before the transition to the frost layer growth stage. Very similar observations were made for these results as for those in Table 6.1. The heat and mass transfer coefficients decreased significantly in the airflow direction, leading to decreases in frost thickness, mass, and density. Decreases in crystal temperature in the direction of airflow were not enough to offset the heat and mass transfer coefficients' decreases. Also of note are the crystal dimensions. While the crystal aspect ratio remained the same for each segment due to its correlation being solely dependent on time, the crystal area coverage, height, radius, and volume ratio changed as the heat and mass transfer driving potentials changed.

Table 6.2: Sample segmentation model calculations for the same inputs used to generate Figure 6.2. Values listed are at the transition from crystal growth to frost layer growth (2355 s).

#	**Variables**		**Segment 1**	**Segment 2**	**Segment 3**
1	heat transfer coefficient	h [W/m^2-K]	51.3	29.6	22.9
2	mass transfer coefficient	h_m [m/s]	0.0425	0.0245	0.190
3	crystal area coverage	AC_c [-]	0.418	0.308	0.267
4	crystal aspect ratio	c [-]	42.0	42.0	42.0
5	crystal height	L_c [mm]	1.152	0.989	0.920
6	crystal radius	r_c [mm]	0.0274	0.0236	0.0219

7	number of crystals per droplet	N_c [-]	53	18	10
8	crystal volume ratio	ε_c	0.307	0.213	0.179
9	crystal temperature	T_c [°C]	-4.62	-4.78	-4.82
10	effective thermal conductivity	k_{eff} [W/m-K]	0.482	0.337	0.284
11	frost mass	m_f [g]	0.109	0.062	0.048
12	frost thickness	δ_f [mm]	1.290	1.064	0.976
13	frost density	ρ_f [kg/m^3]	281.6	195.4	163.5
14	outlet air temperature	$T_{air,out}$ [°C]	4.69	4.51	4.38
15	outlet air pressure	$P_{air,out}$ [Pa]	101,324.7	101,324.3	101,324.0
16	outlet humidity ratio	ω_{out} [kg/kg]	3.663	3.590	3.533
17	outlet relative humidity	RH_{out} [-]	0.692	0.687	0.682

Table 6.3 presents results for the same three segments at one hour of frosting, which occurred when the segments were in the model's frost layer growth stage. Interestingly, while total mass flux changed between the segments, the mass flux that increased frost density remained nearly the same. This result is likely because calculations for the densifying mass flux were dependent on the temperature gradient in the frost layer, which was also nearly identical between segments. This observation indicates that all three segments had similar decreases in frost density for a given time step, while the thicknesses grew at different rates. Additionally, these results also show that this frosting stage was not as efficient at dehumidifying the air stream as the other stages. In Tables 6.1 and 6.2, both the humidity ratio and relative humidity

decreased along the plate length, while in Table 6.3, the humidity ratio decreased and the relative humidity increased. This result indicates that the decrease in air temperature, rather than humidity, was the driving factor for the relative humidity change.

Table 6.3: Sample segmentation model calculations for the same inputs used to generate Figure 6.2. Values listed are at one hour into the simulation (3601 s).

#	Variables		Segment 1	Segment 2	Segment 3
1	heat transfer coefficient	h [W/m^2-K]	52.6	30.4	23.5
2	mass transfer coefficient	h_m [m/s]	0.0437	0.0252	0.0195
3	total mass flux	$\dot{m}''$ [g/m^2-s]	0.0433	0.0277	0.0228
4	density mass flux	$\dot{m}_\rho''$ [g/m^2-s]	0.0031	0.0033	0.0032
5	thickness mass flux	$\dot{m}_\delta''$ [g/m^2-s]	0.0402	0.0244	0.0191
6	frost mass	m_f [g]	0.125	0.073	0.056
7	frost thermal conductivity	k_f [W/m-K]	0.434	0.284	0.242
8	frost surface temperature	T_f [°C]	-3.16	-3.62	-3.83
9	frost thickness	δ_f [mm]	1.472	1.220	1.120
10	frost density	ρ_f [kg/m^3]	284.4	199.0	167.3
11	outlet air temperature	$T_{air,out}$ [°C]	4.73	4.57	4.45
12	outlet air pressure	$P_{air,out}$ [Pa]	101,324.6	101,324.2	101,323.8
13	outlet humidity ratio	ω_{out} [kg/kg]	3.760	3.742	3.728
14	outlet relative humidity	RH_{out} [-]	0.708	0.713	0.716

Figure 6.3 presents comparisons of model predictions with experimental data from Yoon et al. (2010) when the model split the surface into five equal segments in the airflow direction. These data are unique in that they were provided at different points along the plate length and can thus be used to validate the segmentation algorithm described above. In their paper, Yoon et al. used the distance along the plate as the characteristic length when predicting frost behavior on the surface. Because of this, and because they used flow homogenizers at the test section inlet and outlet to straighten the flow, the distance along the plate was also used as the characteristic length for Reynolds number calculations for this validation. Figure 6.3(a) shows that the model could predict the general trend of frost thickness decrease in the direction of airflow but under-predicted actual frost thickness values at all points. Figure 6.3(b) shows that the model over-predicted density at all points along the surface, but the over-predictions were only substantial at the plate's leading edge. Overall, the model produced similar trends to the experimental data, but it over-predicted mass transfer at the plate's leading edge. Additionally, mass distribution was not entirely accurate; the model applied too much of the total mass flux to frost layer densification and not enough to increasing thickness.

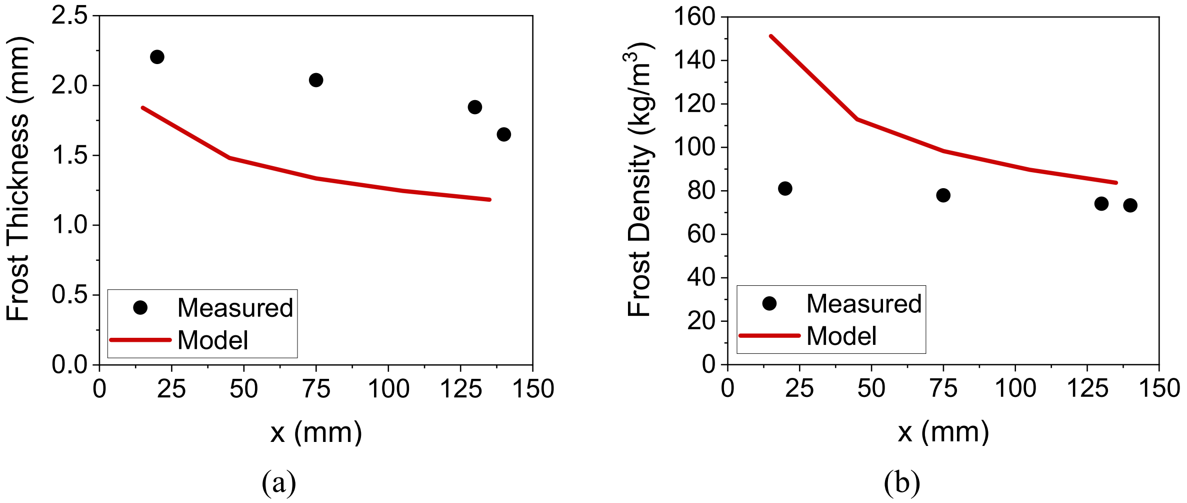


Figure 6.3: Model predictions vs. experimental data for (a) frost thickness and (b) frost density from Yoon et al. (2010)

6.2 Segmentation by Wettability Type

6.2.1 Model Description

This section describes how the model was segmented by wettability type, particularly in a direction perpendicular to the airflow when there is only a single segment in the airflow direction. The simulation tool accommodated two segments of different wettability types in the transverse direction while the user chose the portion of the plate width assigned to each type. The segmentation model in the airflow direction recalculated air properties at the outlet of a segment, but the segmentation model by wettability type recalculated air velocity through each segment. After frost calculations were performed for both segments in the transverse direction, the pressure drop across both segments was calculated and compared. If the pressure drops differed more than the convergence tolerance, air velocity was recalculated to maintain a constant total mass flow rate over the cold surface, and the frost calculations were repeated. These iterations were performed until the pressure drops for both segments were approximately equal, at which point the simulation would proceed to the next time step. To compare model predictions with experimental data gathered in the current study, the flow configuration used for pressure drop predictions was laminar flow between two parallel plates. Equations 6.9 through 6.11 below give the equations used for these calculations.

$$\Delta P = f\left(\frac{L_{plate}}{D_h}\right)\left(\frac{\rho_{air}}{2}\right) {V_{air}}^2 \tag{6.9}$$

where

$$f = \frac{64}{Re_h} \tag{6.10}$$

and

$$D_h = 2\left(H_{ch} - \delta_f\right) \tag{6.11}$$

If the pressure drop for Segment 1 was calculated to be larger than that in Segment 2, then the air velocity in Segment 1 was decreased by a pre-set amount. Then the air velocity in Segment 2 was recalculated to preserve the total mass flow rate over the surface, as defined in Equation 6.12.

$$\dot{m}_{tot} = \dot{m}_1 + \dot{m}_2 = \rho_{air}\, A_{c,seg,1}\, V_{air,1} + \rho_{air}\, A_{c,seg,2}\, V_{air,2} \tag{6.12}$$

where

$$A_{c,seg} = W_{seg}\, \frac{D_h}{2} \tag{6.13}$$

Equation 6.12 was rearranged as Equation 6.14 to solve for the air velocity in Segment 2.

$$V_{air,2} = \frac{\dot{m}_{tot} - \rho_{air}\, W_{seg,1}\left(H_{ch} - \delta_{f,1}\right) V_{air,1}}{\rho_{air}\, W_{seg,2}\left(H_{ch} - \delta_{f,2}\right)} \tag{6.14}$$

Figure 6.4 illustrates the process described above for the wettability segmentation, assuming no segmentation in the airflow direction.

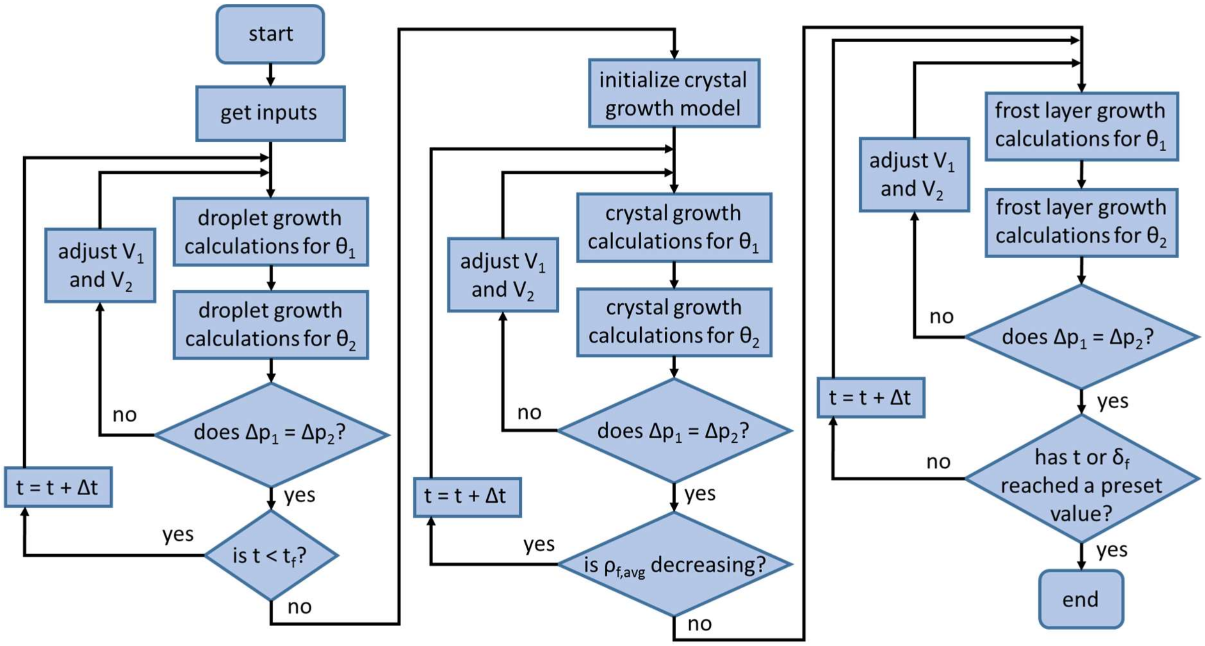

start
get inputs
droplet growth calculations for θ₁
droplet growth calculations for θ₂
adjust V₁ and V₂
no
does Δp₁ = Δp₂?
yes
t = t + Δt
yes
is t < t_f?
no
initialize crystal growth model
crystal growth calculations for θ₁
crystal growth calculations for θ₂
adjust V₁ and V₂
no
does Δp₁ = Δp₂?
yes
t = t + Δt
yes
is ρ_f,avg decreasing?
no
frost layer growth calculations for θ₁
frost layer growth calculations for θ₂
adjust V₁ and V₂
no
does Δp₁ = Δp₂?
yes
t = t + Δt
no
has t or δ_f reached a preset value?
yes
end

Figure 6.4: Flow chart for segmentation with two different contact angles.

6.2.2 Segmentation Results and Validation

Figure 6.5 presents results of thickness, density, mass, and air velocity with time when a cold surface was divided into two segments with different contact angles. The two segments were equal in size and had contact angles of 30° and 150°. The figure shows that both thickness and density were higher for the hydrophobic segment than for the hydrophilic, though the differences were not drastic. These density results seem unusual and occurred because the model predicted less mass deposition on the hydrophilic segment than the hydrophobic segment during the droplet growth stage (as described in Section 5.5). Figure 6.5(d) shows that the air velocity was always smallest through the hydrophobic segment. This result occurred because the frost thickness was larger on the hydrophobic segment, leading to a smaller cross-sectional area for the airflow. Air had to flow more quickly through the larger area over the hydrophilic segment to maintain equal pressure drops between the two segments. The slight difference between the two segments' airflow values did not significantly affect the thickness, density, or mass since the vapor density gradient primarily governed the driving potential.

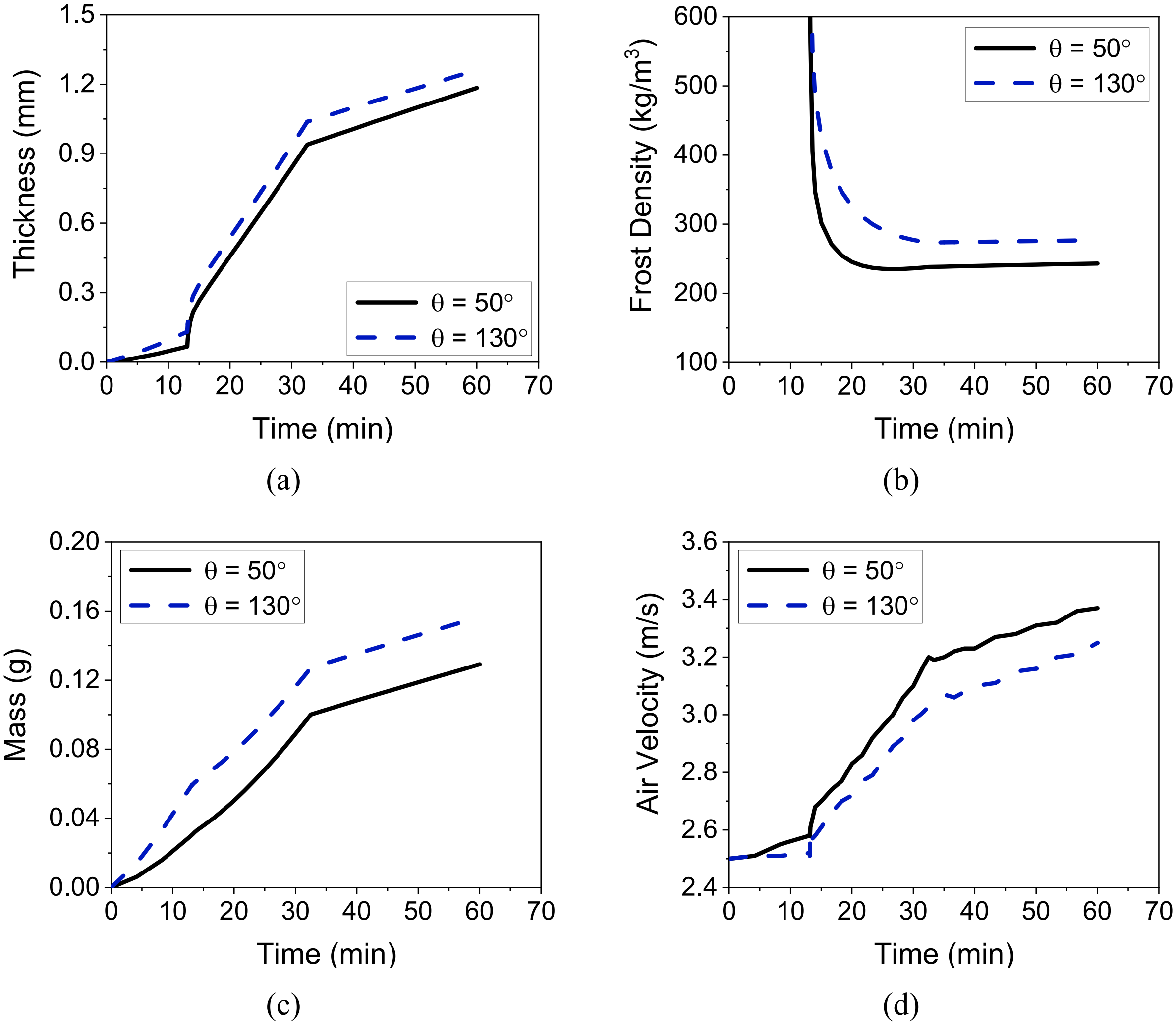


Figure 6.5: (a) Frost thickness, (b) frost density, and (c) frost mass as functions of time for equal segments with different contact angles. T_a = 5 °C, RH = 70 %, V = 2.5 m/s, T_s = -5 °C, L_{plate} = W_{plate} = 30 mm, $H_{channel}$ = 5 mm. θ = 50°, θ = 130°.

Table 6.4 presents sample calculations at the end of the segmented plate model's droplet growth stage from Figure 6.5. The table shows that heat and mass transfer coefficients were very similar for the two segments. The primary differences in the results came from droplet geometry and mass. The projected radius on the hydrophobic segment was much smaller than on the hydrophilic segment. Even though the projected area coverage on the hydrophilic segment was only about 10% higher than on the hydrophobic segment, the hydrophobic segment had many more droplets because their projected radii were smaller. The frost mass was also greater on the hydrophobic segment than the hydrophilic segment, as discussed in conjunction with Figure 6.5. The air velocity had similar values for both segments since, even though the hydrophobic segment had a larger frost thickness, the thickness on both segments was still low enough to have a slight effect on velocity.

Table 6.4: Sample segmentation model calculations for the same inputs used to generate Figure 6.5. Values listed are at the instant before freezing (785 s).

#	Variables		Segment 1: θ = 50°	Segment 2: θ = 130°
1	heat transfer coefficient	h [W/m^2-K]	36.7	36.3
2	mass transfer coefficient	h_m [m/s]	0.0303	0.0300
3	coalescence exponent	μ [-]	1.392	1.256
4	droplet growth rate	dr/dt [m/s]	3.11 x 10^{-8}	3.54 x 10^{-8}
5	area coverage	AC [-]	0.732	0.630
6	droplet radius of curvature	r [mm]	0.480	0.170
7	droplet projected radius	r_{proj} [mm]	0.368	0.170
8	number of droplets	N_d [-]	775	3109
9	interface temperature	T_i [°C]	-4.92	-4.88
10	total frost mass	m_f [g]	0.030	0.059
11	initial frost thickness	δ_0 [mm]	0.067	0.131
12	hydraulic diameter	D_h [mm]	9.87	9.74
13	friction factor	f [-]	0.0336	0.0347
14	air velocity	V [m/s]	2.58	2.52

Table 6.5 presents sample calculations at the end of the model's crystal growth stage for the same segmented surface as in Figure 6.5. Most of the table values varied somewhat between the two segments, but the most significant difference occurred in the frost layer density. The hydrophobic segment had a larger density than the hydrophilic segment because of its higher mass deposition. The larger mass on the hydrophobic surface was a direct result of the droplet growth stage's results since the masses on the two segments had nearly the same offset at the end of the crystal growth stage as at the end of the droplet growth stage. There was also a more significant disparity between frost thickness values by this point, leading to a greater air velocity difference between the segments.

Table 6.6 presents sample calculations during the frost layer growth stage at one hour of frosting. Because of the higher frost density on the hydrophobic segment, the frost thermal conductivity was higher and the frost surface temperature was lower for this segment than for the hydrophilic segment. This discrepancy led to a higher thickness mass flux and a lower diffusion mass flux on the hydrophobic segment than on the hydrophilic. The lower frost surface temperature also led to a slightly higher overall mass flux on that segment.

Table 6.5: Sample segmentation model calculations for the same inputs used to generate Figure 6.5. Values listed are at the transition from crystal growth to frost layer growth (1951 s).

#	**Variables**		**Segment 1: θ = 50°**	**Segment 2: θ = 130°**
1	heat transfer coefficient	h [W/m^2-K]	40.9	39.7
2	mass transfer coefficient	h_m [m/s]	0.0338	0.0329
3	crystal area coverage	AC_c [-]	0.367	0.397
4	crystal aspect ratio	c [-]	33.9	33.9

5	crystal height	L_c [mm]	0.872	0.906
6	crystal radius	r_c [mm]	0.0257	0.0267
7	number of crystals	N_c [-]	80	17
8	crystal temperature	T_c [°C]	-4.73	-4.67
9	effective thermal conductivity	k_{eff} [W/m-K]	0.438	0.429
10	frost mass	m_f [g]	0.100	0.127
11	frost thickness	δ_f [mm]	0.939	1.038
12	frost density	ρ_f [kg/m^3]	238.1	273.3
13	hydraulic diameter	D_h [mm]	8.12	7.93
14	friction factor	f [-]	0.0328	0.0356
15	air velocity	V [m/s]	3.20	3.03

Table 6.6: Sample segmentation model calculations for the same inputs used to generate Figure 6.5. Values listed are at one hour into the simulation (3601 s).

#	**Variables**		**Segment 1**	**Segment 2**
1	heat transfer coefficient	h [W/m^2-K]	41.9	41.2
2	mass transfer coefficient	h_m [m/s]	0.0348	0.0341
3	total mass flux	$\dot{m}''$ [g/m^2-s]	0.0381	0.0391
4	density mass flux	$\dot{m}''_\rho$ [g/m^2-s]	0.0033	0.024
5	thickness mass flux	$\dot{m}''_\delta$ [g/m^2-s]	0.0348	0.0368
6	frost mass	m_f [g]	0.129	0.157

7	frost thermal conductivity	k_f [W/m-K]	0.359	0.440
8	frost surface temperature	T_f [°C]	-3.50	-3.68
9	frost thickness	δ_f [mm]	1.184	1.261
10	frost density	ρ_f [kg/m^3]	243.1	276.7
11	hydraulic diameter	D_h [mm]	7.63	7.48
12	friction factor	f [-]	0.0333	0.0353
13	air velocity	V [m/s]	3.37	3.25

Validation of the wettability segmentation model was performed using data gathered for a biphilic surface composed of alternating hydrophobic and hydrophilic sections. The hydrophobic and hydrophilic sections were the same types as on the hydrophobic and hydrophilic surfaces described in Chapter 3. Testing this surface was a group effort among several students, using the same experimental facility as in the current study. Data were gathered first for this surface without cleaning by Burak Adanur, and the results were presented in Adanur et al. (2019). The second set of data was gathered for this surface by Fuqiang Ren that has not yet been published. For this data, cleaning was performed periodically. The surface was not cleaned with isopropyl alcohol after every test to avoid severe degradation of the hydrophilic sections. However, isopropyl alcohol was used occasionally in order to remove the PM2.5 contaminants. I assisted primarily with the analysis of raw data gathered during the experiments. For the testing series where no cleaning was performed between tests, droplets looked very similar on both the hydrophobic and hydrophilic sections. There was a noticeable difference in droplet sizes between the surface types for the data set where cleaning was occasionally performed. The newer data set was used to validate droplet size, thickness, and density since the surface wettability effects were less affected by surface contamination.

The biphilic surface had 1” x 6” dimensions. It had six equally-sized segments, three of which were hydrophilic ($\theta \approx 24°$) and three of which were hydrophobic ($\theta \approx 113°$). The hydrophobic and hydrophilic segments alternated across the width of the surface. A schematic of the surface is provided in Figure 6.6. For the tests from the newer data series used for model validation, environmental conditions were held at T_s = 5.3 °C, RH = 82.4%, V = 3.8 m/s, and T_s = -4.3 °C. Several tests were performed at these same nominal test conditions. The measured freezing time, which was calculated to be the average of the freezing times of thirteen similar tests, was about 49 min. A single freezing time could be assigned to the entire surface because it was observed that the onset of freezing occurred simultaneously for both the hydrophobic and hydrophilic segments. When the model was applied to the test conditions listed above, it predicted a freezing time of about 9.3 min, far shorter than the measured time, but comparable with the freezing times measured for surfaces with a single wettability under similar environmental conditions (see Chapter 3). Since these test conditions were within the range of applicable conditions for the freezing time correlation presented in Chapter 4, it appears that there was an as yet unexplained phenomenon that caused a delay of freezing for the biphilic surface. Freezing on the hydrophobic and hydrophilic surfaces could individually be predicted by the model, but the model failed when predicting a surface that was a combination of the two. This phenomenon was also observed in the older data set with no cleaning between tests. Droplets on this biphilic surface took longer to freeze than those on all other surfaces tested. Further investigation is necessary to determine why the combination of two predictable surface types led to unpredictable results.

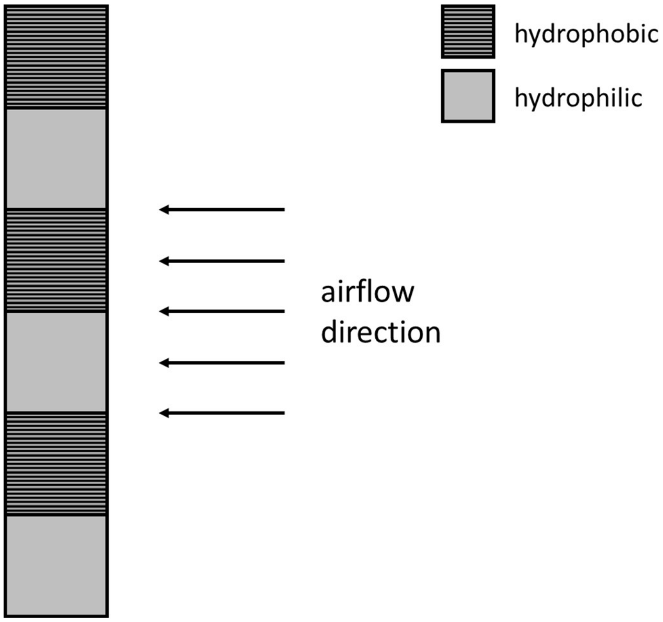


Figure 6.6: Schematic of biphilic surface used for model validation.

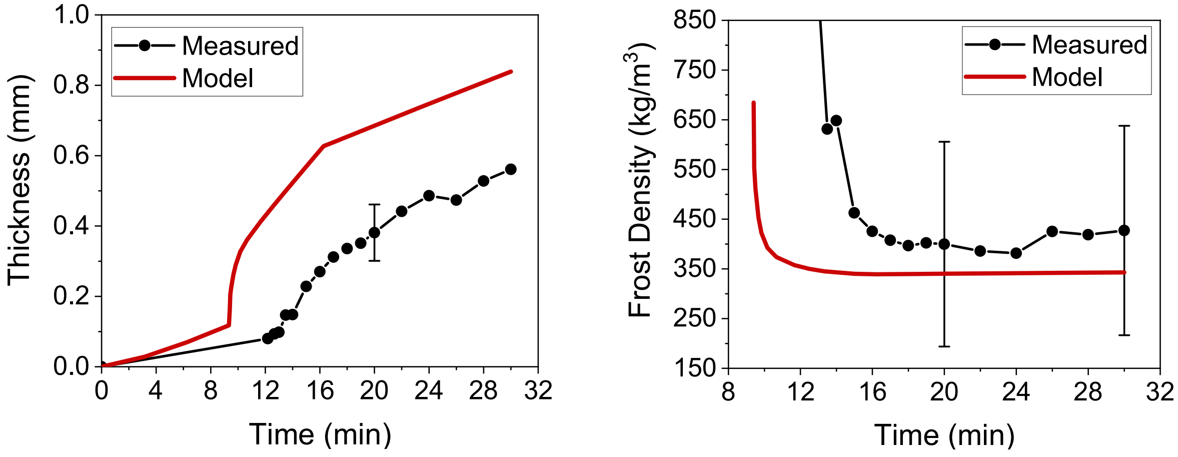
Table 6.7 compares the model predictions to measured results for average projected droplet radius at freezing on the hydrophilic and hydrophobic portions of the biphilic surface. Since it was already established that the model could not predict the freezing time on the biphilic surface, a test with a freezing time shorter than the average was chosen for comparison of droplet and frost layer properties. This choice was made so the error caused by inaccurate freezing time predictions would not propagate to the rest of the validation. There was a wide range of freezing times between the thirteen tests previously mentioned, and a test was selected for comparison that had a freezing time of about 12 minutes. Table 6.7 shows that the model over-predicted projected diameters on both portions of the surface. However, it over-predicted both by similar amounts (about 45%), so it appears that the model could correctly predict the ratio of droplet sizes between the two surfaces even if it did over-predict the mass accumulation.

Table 6.7: Comparisons of model predictions with measured data of projected droplet diameter at freezing on a biphilic surface

	Measured	**Model**	
Surface Type	**Projected Diameter (mm)**	**Projected Diameter (mm)**	**Error (%)**
Hydrophilic	0.844	1.204	42.7
Hydrophobic	0.360	0.530	47.2

Figure 6.7 compares model predictions of average thickness and density to values measured during the same experimental run as for Table 6.7. Error bars in the figure were calculated in the same way as described in Chapter 3. The video scope used to capture images of the frost layer did not have a wide enough field of view to allow for accurate measurement of frost thickness for each type of segment individually, so average values for the surface were collected that included both hydrophobic and hydrophilic portions. These average values are what are presented in Figure 6.7. Similarly, the model predicted separate values for the segments, but these were averaged to compare with the measured data. The figure shows that the model predicted the thickness profile's shape correctly, though it significantly over-predicted the actual thickness values. It slightly under-predicted frost density because of the thickness over-predictions, even though mass was over-predicted as well.

In summary, the model failed at predicting the freezing time for the biphilic surface, regardless of whether or not cleaning was performed between each test, even though it could predict freezing times on both surface types individually (see Chapter 4). It accurately predicted the ratio of droplet sizes between the surface types and the shapes of the thickness and density trends. However, it over-predicted the mass deposition on both segments. Future investigation into frost growth on biphilic surfaces would help understand the causes of the unexpected freezing delays and the lower mass transfer rates that were observed but not predicted during the current study.



(a)

(b)

Figure 6.7: Comparison of model predictions with measured data of average (a) thickness and (b) density on a biphilic surface

7. Model Extension to Folded-Flat Fin Configuration

After the three-stage frost model for average values was developed and then segmented by wettability and in the direction of airflow, it was further extended to folded-flat fin configurations. In order to approximate folded flat fins on a heat exchanger, a parallel plate configuration was adopted. The following sections discuss the additions to the model, which allow for application to parallel plates.

7.1 Model Development

The parallel-plate configuration used for model development can be visualized according to Figure 7.1. As in Figure 7.1, air flowed between two identical parallel cold plates. Thermal boundary layers began at the leading edges ($x = 0$) of both plates and met in the middle at some critical distance in the direction of airflow, x_{crit}. At distances smaller than x_{crit}, the boundary layers did not yet interfere with each other, and heat transfer was assumed to be governed by equations for external laminar flow over flat plates. Once the boundary layers met, external flow was no longer be assumed. At this point, heat transfer was assumed to be governed by equations for internal laminar flow between parallel plates. The model for segmentation in the airflow direction was a prerequisite to the parallel plate extension since the heat transfer equations changed along the channel's length. Additionally, the transition point changed as the frost grew. As the frost thickness increased, the flow area decreased, and the boundary layers joined more quickly.

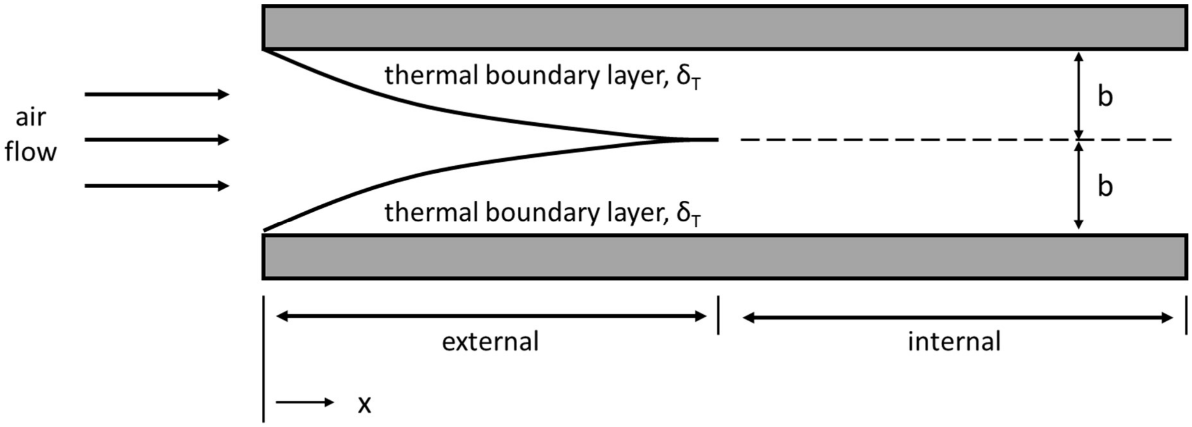

air
flow
thermal boundary layer, δ_T
thermal boundary layer, δ_T
b
b
external
internal
x

Figure 7.1: Illustration of the parallel-plate configuration used for model development.

The additional governing equations for the parallel plate configuration are presented below. First, the hydraulic diameter for the airflow was calculated according to Equation 7.1, assuming that since the two plates are identical, the frost thickness was equal for both.

$$D_h = 2\left(H_{ch} - 2\,\delta_f\right) \tag{7.1}$$

The thickness of the thermal boundary layer was calculated using Equation 7.2.

$$\delta_t = \frac{4.91x}{Pr^{1/3}\sqrt{Re_x}} \tag{7.2}$$

The critical distance for the transition from external to internal flow was obtained by solving Equation 7.2 for *x*. The resulting expression is presented in Equation 7.3.

$$x_{crit} = \frac{\rho\, V\, \delta_{t,crit}^{2}\, Pr^{2/3}}{24.1\,\mu} \tag{7.3}$$

The critical thermal boundary layer substituted into Equation 7.3 is given below as Equation 7.4, where the frost thickness is calculated in the first segment since that segment has the largest frost thickness and, therefore, the most flow restriction.

$$\delta_{t,crit} = \frac{H_{ch}}{2} - \delta_f \tag{7.4}$$

When the distance along the plate, x, was smaller than x_{crit}, the Nusselt number was calculated using Equation 6.5. If x was larger than x_{crit}, then the Nusselt number was calculated using an expression from Rohsenow et al. (1998), which was developed specifically for thermally developing laminar flow between isothermal parallel plates. This expression collapses to the fully-developed Nusselt number for laminar flow between parallel plates as the distance along the plate increases. These equations are presented below as Equations 7.5 and 7.6. The flow was assumed to be hydrodynamically fully developed because the literature studies used for model validation had test facilities explicitly constructed to produce fully developed laminar flow at the test section inlets.

$$Nu_x = \begin{cases} 1.233x^{*-1/3} + 0.4 & for\; x^* < 0.001 \\ 7.541 + 6.874(10^3 x^*)^{-0.488} e^{-245x^*} & for\; x^* > 0.001 \end{cases} \tag{7.5}$$

where

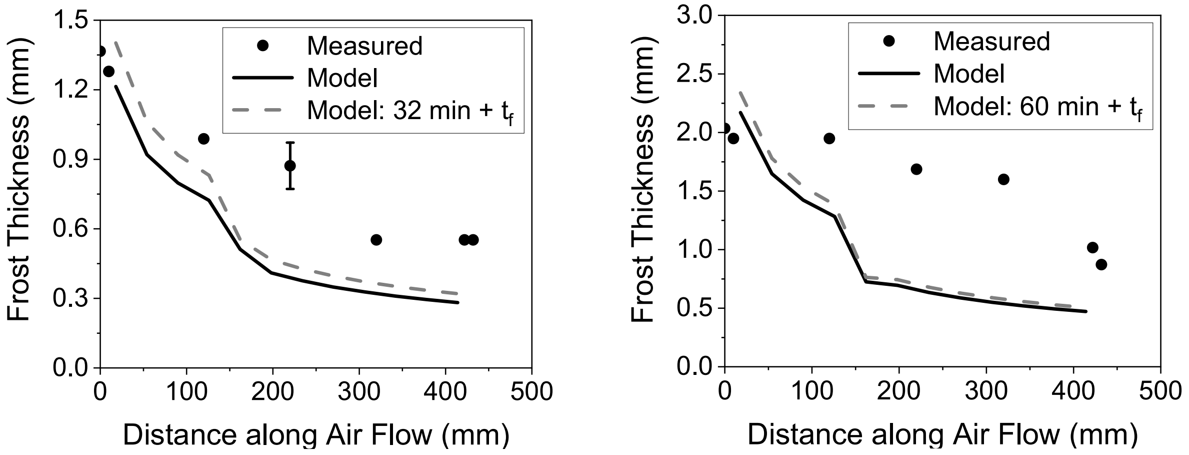
$$x^* = \frac{x/D_h}{Re_h\, Pr} \tag{7.6}$$

The other primary difference between the single-plate configuration and the parallel-plate configuration was the recalculation of air velocity at each time step. Equation 7.7 is a modification of Equation 6.14 applied to parallel plates. Equation 7.7 is slightly different in that two times the frost thickness is subtracted from the channel height since frost grows from both the top and the bottom of the channel. This alteration leads to a faster constriction of airflow and an air velocity that increases more rapidly.

$$V_{air,2} = \frac{\dot{m}_{tot} - \rho_{air}\, W_{seg,1}\, \left(H_{ch} - 2\,\delta_{f,1}\right) V_{air,1}}{\rho_{air}\, W_{seg,2}\, \left(H_{ch} - 2\,\delta_{f,2}\right)} \tag{7.7}$$

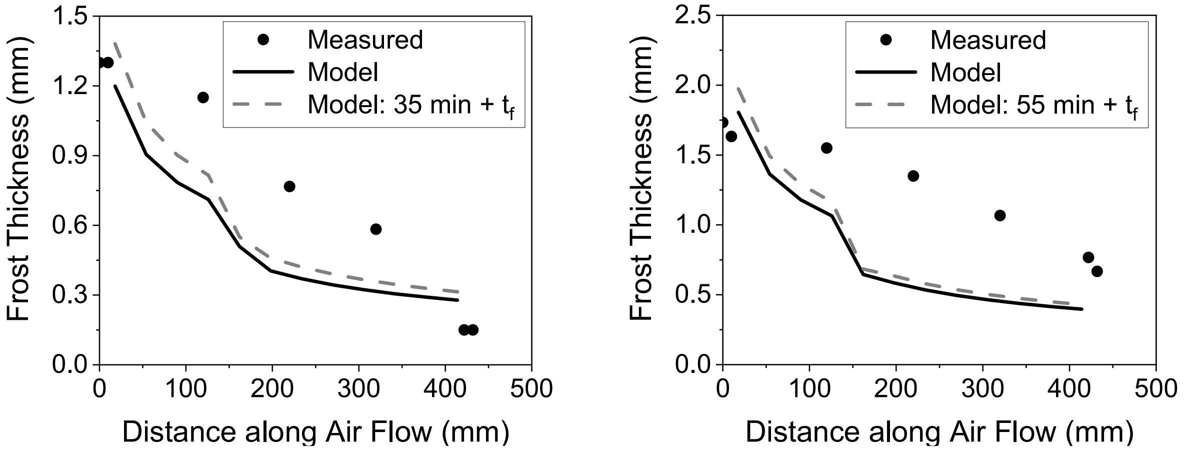
7.2 Model Validation and Results

The model extension to a parallel plate configuration was validated using measured data of frost thickness and density from Lüer and Beer (2000). In this paper, frost growth was investigated on parallel aluminum surfaces. Both surfaces were identical and were 432 mm in length (in the direction of airflow) and 114 mm in width, with a 24 mm space between the plates. Figures 7.2(a) and 7.2(b) compare model predictions to frost thickness data at 32 min and 60 min of frosting, respectively, when the humid air stream's relative humidity was 70%. Error on the frost thickness measurements was ±0.1 mm and is illustrated by a sample error bar in Figure 7.2(a). However, similar errors existed for each point in the figures. Since no information was given about what was considered to be the start of a test (time 0), model predictions were presented, assuming that both condensation began at time 0 (solid line) and crystal growth on top of frozen droplets began at time 0 (dashed line). The model assumed a contact angle of 75° for the aluminum surfaces and split the plate length into twelve segments in the airflow direction. Figure 7.2(a) shows that the model closely predicted frost thickness at the plates' leading edge after 32 min of frosting, while it under-predicted thickness over the rest of the plate. After 60 min (Figure 7.2(b)), frost thickness at the leading edge was still predicted well, but thickness in the plates' centers was severely under predicted. It was still under-predicted at the trailing edge, though not as severely. Figure 7.3 compares model predictions to measured data for the same parallel plates at a relative humidity of 60%. Predictions were better at this lower RH value for both 35 min and 55 min into the frost growth (Figure 7.3(a) and 7.3(b), respectively). Frost thickness values at both the leading and trailing edges were predicted reasonably well at 35 min, while under-predictions were less drastic at 55 min than at 70% RH for a similar frosting time. Both figures show a sudden decrease in frost thickness values between 100 and 200 mm. This decrease occurred when the simulation switched from modeling the air flow as external to modeling it as internal, indicating that the transition criterion may need adjustment.



(a) (b)

Figure 7.2: Comparison of model predictions with experimental data from Lüer and Beer (2000). Data were gathered after (a) 32 min and (b) 60 min of frost growth. T_a = 20 °C, RH = 70 %, Re_D = 1500, T_s = -8 °C.



(a) (b)

Figure 7.3: Comparison of model predictions with experimental data from Lüer and Beer (2000). Data were gathered after (a) 35 min and (b) 55 min of frost growth. T_a = 20 °C, RH = 60 %, Re_D = 1500, T_s = -8 °C.

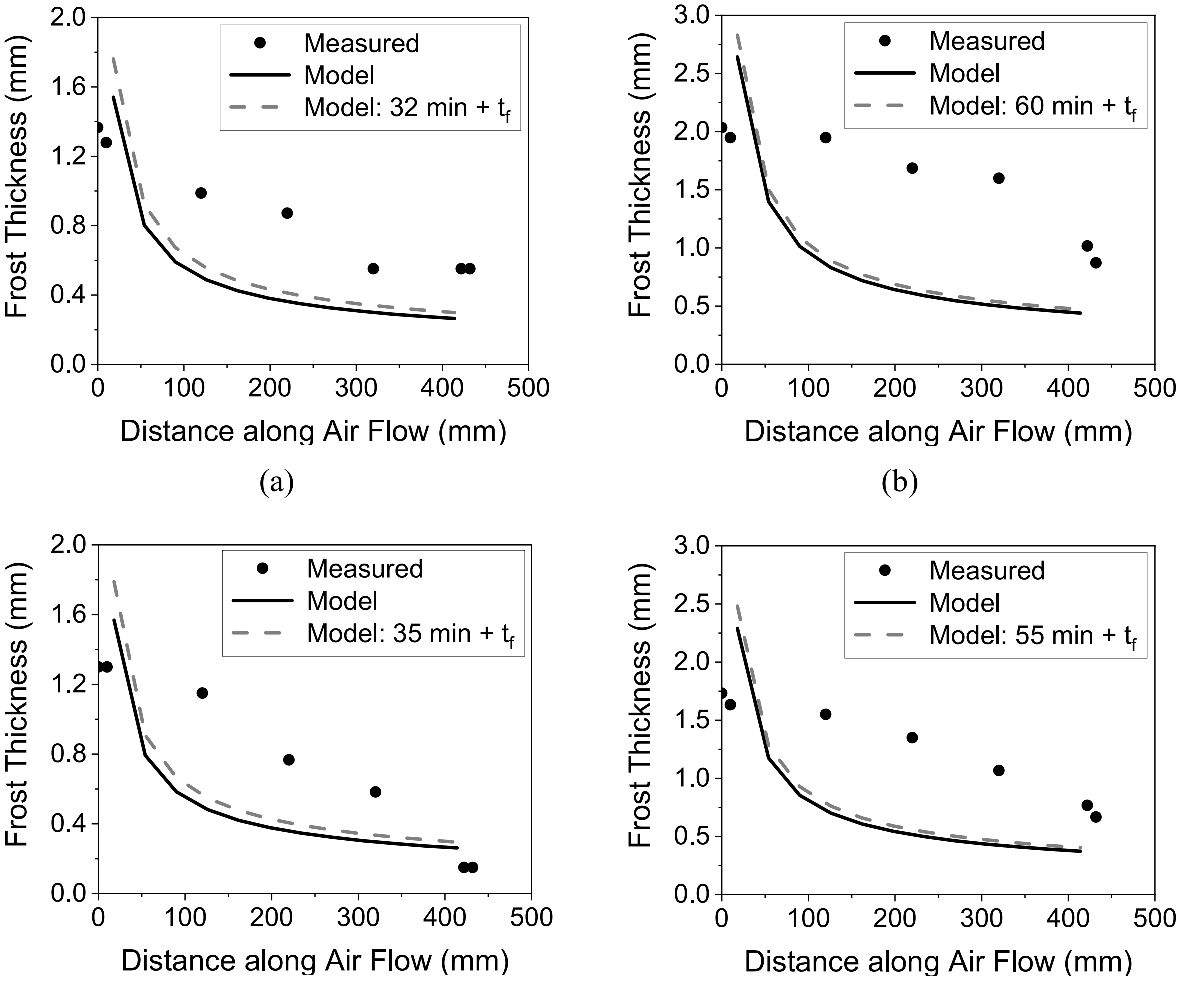
Table 7.1 compares measured density values to model predictions for the test conditions and frosting times from Figures 7.2 and 7.3. In the table, the values for average modeled density were obtained by taking an unweighted average of the density values for all twelve segments. For both relative humidity values, average frost density was predicted to be only about half of what was measured. Interestingly, however, density modeled in the first segment in the airflow direction was similar to, though slightly higher than, the average measured values.

Table 7.1: Comparisons of model predictions and measured density values for the environmental conditions in Figures 7.2 and 7.3

		Measured Density (kg/m^3)	Modeled Density (kg/m^3)	
	Frosting Time	Average	Average	First Segment
70% RH	32 min	122.9	53.4	125.6
	60 min	104.4	57.7	132.9
60% RH	35 min	80.8	48.4	112.2
	55 min	90.9	51.6	118.0

Since Figures 7.2 and 7.3 indicated that the criterion for transitioning from external flow to internal flow might not be accurate, further investigation was made into the flow type assumptions. Figures 7.4 and 7.5 compare model predictions to the same measured data from Lüer and Beer (2000) as above but when the flow was modeled as only internal and only external over the channel's entire length, respectively. Figure 7.4 indicates that assuming solely internal flow from the beginning of the channel produced modeled trends that did not match the experimental trends. Frost thickness was over-predicted at the leading edge and was severely under-predicted over the rest of the channel. This steep decrease was not observed in the measured frost thickness profiles. Figure 7.5 shows that assuming only external flow produced good predictions for both the channel's leading and trailing edges. The only exception was when

RH = 60% and t = 35 min, where the trailing edge's thickness was significantly over-predicted. Assuming only external flow produced much shallower decreases in frost thickness along the plate's length, a trend that more closely approximated the trend observed in the experiments.



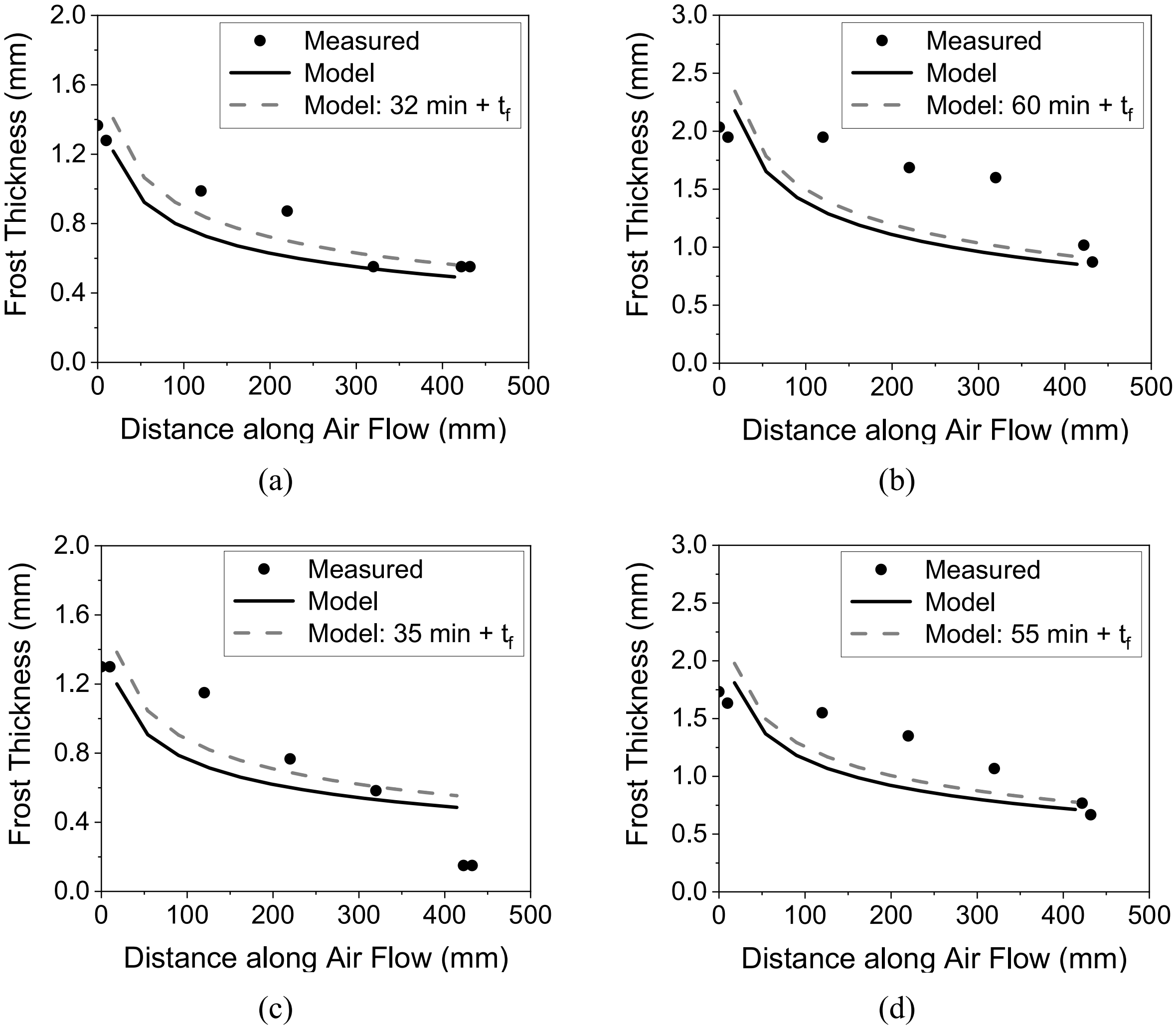
(a)

(b)

(c)

(d)

Figure 7.4: Comparison of model predictions with data from Lüer and Beer (2000) when flow was modeled as internal for the entire channel. (a) RH = 70%, t = 32 min, (b) RH = 70%, t = 60 min, (c) RH = 60%, t = 35 min, (d) RH = 60%, t = 55 min.

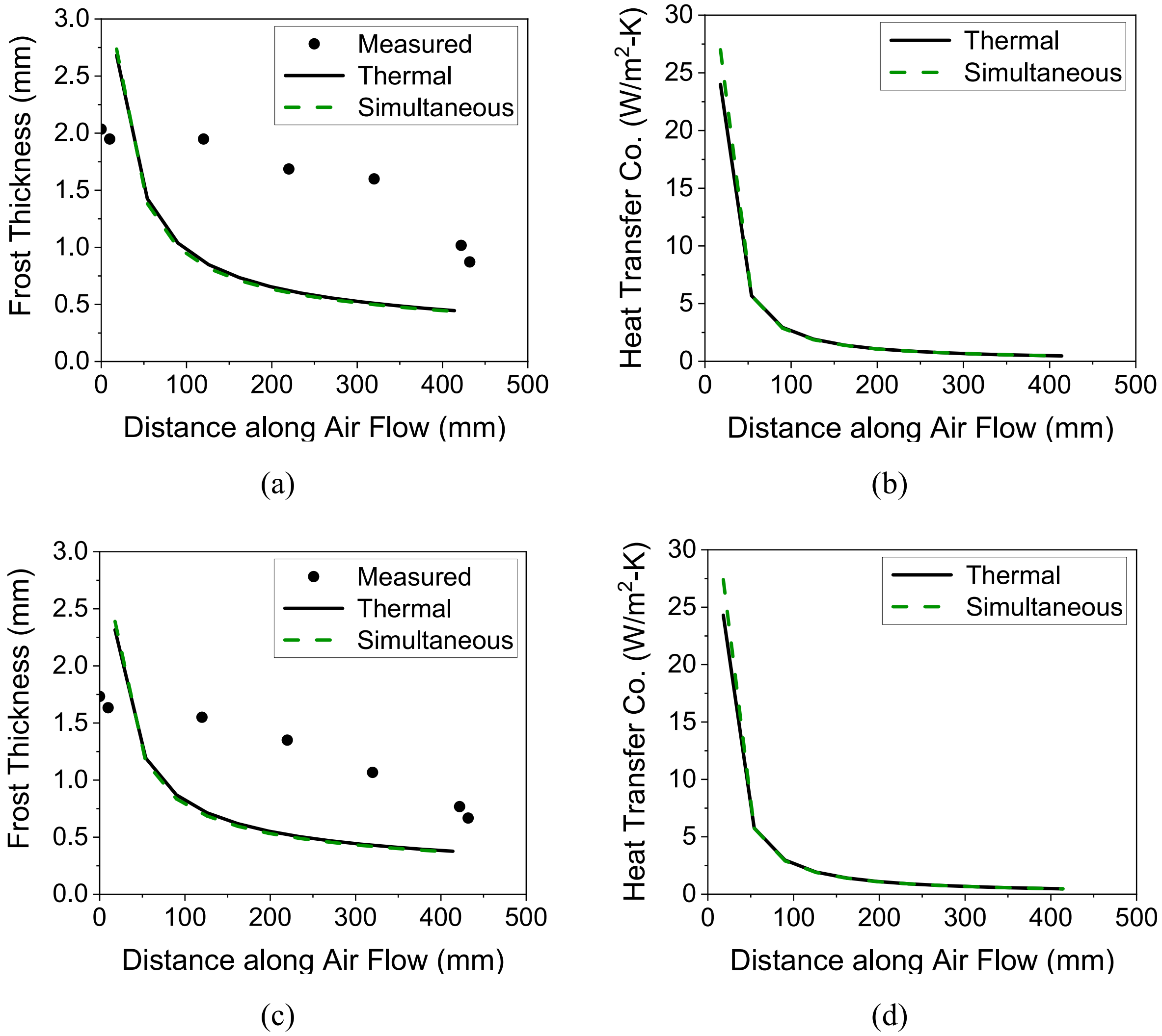


(a) (b)

(c) (d)

Figure 7.5: Comparison of model predictions with data from Lüer and Beer (2000) when flow was modeled as external for the entire channel. (a) RH = 70%, t = 32 min, (b) RH = 70%, t = 60 min, (c) RH = 60%, t = 35 min, (d) RH = 60%, t = 55 min.

Table 7.2 presents comparisons of measured and modeled average density values when assuming only internal or external flow. The results in this table correspond to those in Figures 7.4 and 7.5. The results show that while average density was under-predicted for both assumptions, the under predictions were less severe when only external flow was assumed, leading to errors within 30% for three of the four cases. Even though the assumption of hydrodynamically fully developed flow was made based on the authors' stated design of the experimental facility, this assumption was temporarily set aside to check if better results could be achieved without it. Figure 7.6 compares frost thickness and heat transfer coefficient for internal thermally developing flow and internally simultaneously developing flow. The figure shows that higher heat transfer coefficients were achieved at the leading edge when the flow was considered to be simultaneously developing, which led to slightly higher frost thickness values. However, after the first segment, both the heat transfer coefficient and the frots thickness were nearly identical for both cases. Thus, assuming simultaneously developing flow produced nearly identical, but slightly worse, results than assuming only thermally developing flow.



(a)

(b)

(c)

(d)

Figure 7.6: Comparisons of model predictions with data from Lüer and Beer (2000) when the flow was assumed to be thermally developing and simultaneously developing. (a), (b): RH = 70%, t = 60 min. (c), (d): RH = 60%, t = 55 min.

Table 7.2: Comparison of model predictions to measured average density values when the flow is assumed to be solely internal or external. Data correspond to those in Figures 7.4 and 7.5.

		Measured Density (kg/m^3)	Modeled Average Density (kg/m^3)	
	Frosting Time	Average	Internal	External
70% RH	32 min	122.9	59.3	72.5
	60 min	104.4	63.3	78.1
60% RH	35 min	80.8	52.4	65.7
	55 min	90.9	55.6	70.1

As a further check of the model, validation was also performed with data from Nascimento et al. (2015). In this paper, frost growth was investigated on parallel aluminum surfaces. Both surfaces were identical and were 40 mm in length (in the direction of airflow) and 60 mm in width. The spacing between the surfaces varied between tests. These data were for average values of frost thickness and density instead of local values along the airflow direction. Table 7.3 compares model predictions of average frost thickness and density with experimental data for two separate test runs. Both tests had roughly the same environmental conditions ($T_a \approx$ 5.3 °C, RH $\approx$ 68%, V $\approx$ 1.1 m/s, $T_s \approx$ -10.4 °C), but the spacing between the plates varied between 9.5 and 19 mm between the two. The model predicted frost thickness at all times for both tests within ±20% and frost density values within ±32%. Interestingly, the critical distance for the transition from external to internal flow was larger than the channel's length, so the flow was assumed to be external for the entire channel for both spacing values.

In summary, the model could reasonably predict average and local frost thickness values and average frost density values if the flow was assumed to be external over the channel's length. Under-predictions were observed when internal flow was assumed at any point along the channel, except for the leading edge. Based on this analysis, it appears that traditional expressions for the Nusselt number for airflow between parallel plates do not apply well to frosting conditions. Future research could be performed to investigate why moving from a single, external plate to a parallel plate channel does not significantly alter the frost layer characteristics.

Table 7.3: Comparisons of model predictions of average frost thickness and density with experimental data from Nascimento et al. (2015).

	H_{ch} = 19 mm			**H_{ch} = 9.5 mm**		
Property	**δ_{exp} (mm)**	**δ_{calc} (mm)**	**Error (%)**	**δ_{exp} (mm)**	**δ_{calc} (mm)**	**Error (%)**
30 min	0.7	0.58	-17.1	0.6	0.65	8.3
60 min	1.1	0.99	-10.0	1.2	1.14	-5.0
90 min	1.6	1.37	-14.4	1.8	1.55	-13.9
Property	**ρ_{exp} (mm)**	**ρ_{calc} (mm)**	**Error (%)**	**ρ_{exp} (mm)**	**ρ_{calc} (mm)**	**Error (%)**
90 min	112.0	129.1	15.3	101.3	133.1	31.4

8. Conclusions and Recommendations for Future Work

8.1 Conclusions

Frost growth on the fins of outdoor evaporators negatively impacts heat pump operation during winter months. It decreases system efficiency by adding a thermal resistance layer to the fins and increasing the air-side pressure drop by blocking the airflow through the fins. Several studies in the published literature reported that surface coatings with different wettability values could change the frost layer's characteristics, potentially inhibiting the frost growth and mitigating some of its adverse effects. These studies' cumulative results were sometimes inconclusive or contradictory, and most available frost models that accounted for surface wettability were developed solely using the researchers' measured data. This dissertation addresses some of these gaps by presenting new experimental and analytical investigations of how changing surface wettability affected early frost growth.

The experimental portion of this research revealed that the average droplet sizes were larger and droplet size distributions were wider for a hydrophilic surface than for hydrophobic and bare aluminum surfaces. These discrepancies were due to a coalescence phenomenon occurring before the droplets froze into ice beads. Surface wettability affected the droplet growth profiles due to droplet coalescence; as surface contact angle increased, the effects of droplet coalescence on droplet growth decreased. The temperature difference and supersaturation degree between the air and the surface affected droplet coalescence behavior as well. Because of the relatively high sub-freezing plate temperatures used in this work, the plates' surface wettability had a negligible impact on the freezing time. The supersaturation degree and the temperature difference between the air stream and the surface controlled freezing times for all surface wettability types. These observations indicated that, at typical heat pump operating conditions, the hydrophobic surface coating did not mitigate the frost’s negative effects. It did not delay frosting by increasing the freezing time, nor did it cause a thinner layer of frost to form. On the contrary, though the droplets had smaller radii on the hydrophobic surface than on the hydrophilic, the droplets were taller on the hydrophobic surface. Taller droplets would lead to points of locally high thickness that would inhibit the free stream airflow. If the nature of a specific hydrophobic coating or the surface configuration could cause droplets to roll off the surface before they froze, a hydrophobic coating could be useful under frosting conditions. However, the current study’s observations indicate that if droplets do not roll off the surface, hydrophobic coatings do little to alleviate frost’s adverse effects.

When testing was conducted without cleaning the surfaces between individual test runs, a film deposited on the surfaces that smoothed surface imperfections, inhibited surface wettability effects on droplet size and shape, and delayed freezing. However, a significant variation in the frost nucleation behavior was measured in the present work when surfaces were cleaned before each test. For these cases, surface wettability played a significant role in the droplets’ sizes and shapes, and droplet freezing was accelerated. Based on these observations of clean and dirty surfaces, it seems that surfaces contaminated with a thin film, as in the current study, had the potential to delay frosting. However, there was no indication of delayed condensation, leading to similar mass deposition as on the cleaned surfaces and very large droplets at the time of freezing. These large droplets may not adversely affect heat exchangers with relatively large fin spacing. However, they could present a problem for microchannel heat exchangers with louver fins with very small fin spacing. If the droplets grow large enough to bridge the gap between fins before they freeze, the fins may retain this water due to stronger adhesion forces even after defrosting, leading to some airflow blockage at the start of the next frosting cycle. Thus, this type of surface contamination may be useful for specific systems which would benefit from delayed frosting. However, it would likely have harmful effects on microchannel heat exchanger operation due to water retention on the fins. After initial nucleation, measurements of early frost layer characteristics showed initial dependence on the contact angle, but that this dependence mostly disappeared after the crystal growth stage of frosting. This initial dependence was due to differences in droplet geometries as the surface contact angle changed. However, the effects of droplet geometry on the initial frost layer had faded by 30 minutes into the frosting tests for most of the sets of test conditions considered.

The new, semi-empirical model for droplet growth and freezing presented in this dissertation predicted reasonably well the data of droplet growth and freezing on flat horizontal finned structures with various hydrophilicity even though some intriguing discrepancies between the model results and measurements were highlighted for some comparisons with experiments. Model results quantified how droplet freezing time asymptotically approached infinity if the supersaturation degree was reduced and if the surface temperature approached the triple point. Small SSD indicated small vapor pressure gradients and thus small mass transfer potentials and droplet growth rates. The cold surface temperature affected droplet cooling directly, and as droplets cooled, the energy barrier required for ice nucleation decreased. This reduction in the energy required for ice nucleation resulted in a higher probability of freezing and thus a shorter freezing time. There have been no conclusive observations of the effect of contact angle on the freezing time for the environmental conditions of high surface temperature typical of heat pump operation. However, it appears the static contact angle primarily affected droplet growth by directly altering the geometry and coalescence behavior of the liquid droplets formed on the cold surfaces. In particular, more droplet coalescence events occurred in a given amount of time at smaller contact angles because these droplets tended to grow quickly in the radial direction and contact each other faster.

The new, semi-empirical model for crystal growth presented in this dissertation predicted early frost thickness reasonably well for the data gathered in this research. However, it over-predicted density values due to over-predictions of frost mass at the end of the droplet growth period. If the droplet model was bypassed, the crystal growth model could predict frost density within the experimental error. The overall three-stage frost model reasonably predicted frost thickness, and sometimes density, data from several other sources in addition to the data gathered during this research. Primary failings occurred when the air temperature was high ($T_a > 16$ °C), the surface temperature was low ($T_s < -10$ °C), and when authors reported substantial differences between frost characteristics on surfaces with different wettability. The overall model results indicated that increasing air humidity increased both the frost layer's thickness and density. Increasing the air velocity served to accelerate the movement from one stage to another within the model without significantly affecting the frost layer characteristics. Decreasing the surface temperature decreased the frost layer's density, while its effects on thickness during early frosting were complicated. The thickness curves at different temperatures crossed over each other. However, steeper slopes of frost thickness during the frost layer growth stage indicated that the lowest surface temperature would ultimately achieve the highest frost thickness. Slightly higher frost thickness values were achieved for higher surface contact angles. Strange results were obtained for frost density as the contact angle was varied; a surface with a contact angle of 120° had the largest density, but then density decreased as contact angle increased and decreased. If the droplet growth exponent, μ, was adjusted so similar amounts of mass were deposited on each surface during the droplet growth stage, density results were as expected. The surface with the lowest contact angle had the highest density, and the surface with the highest contact angle had the lowest density. However, the spread in thickness and density values across all surfaces was very small. Further investigation would help refine μ's dependence on the contact angle, especially as it pertains to superhydrophilic surfaces.

The three-stage model developed during this research was segmented in the direction of airflow and by wettability to investigate changing frost layer behavior across a more complex surface. When a plate was segmented in the direction of airflow, both frost thickness and density were greatest at the leading edge and declined with each segment, a trend which is supported by literature results. When a surface was divided into equal hydrophobic and hydrophilic segments, the hydrophobic segment had the largest thickness and density. The hydrophilic segment had a larger air velocity since it had a lower thickness, and more air must flow through that segment to achieve the same pressure drop as for the hydrophobic segment. The model was also extended for use with a folded-flat-fin configuration modeled as humid air flowing between identical cold surfaces. The model could only predict frost layer characteristics when the flow was assumed to be external for the channel's entire length. If internal flow was assumed at any point past the leading edge of the channel, frost thickness and density were both severely under-predicted.

Some imperfections in the overall model exist because the model is a semi-empirical engineering model which predicts average values instead of a strictly physics-based model predicting frost characteristics at each point in a fine mesh. However, this model is useful for quickly predicting frost characteristics for all three frosting stages for various surface types, flow configurations, and environmental conditions. For example, predictions of frost layer characteristics for an hour of testing at a given set of test conditions could be produced in about

45 minutes. This time cost assumes time steps of 5 s and 2 s for the droplet growth and subsequent crystal and frost layer growth stages, respectively. It also assumes the simulations are run on a computer with a 64-bit operating system with a 3.5 GHz processor and 32 GB of RAM. Supercomputers and time-intensive CFD processing are not required. This model can provide helpful information about how frost will respond to various environmental conditions, surface types, and surface configurations that can aid in preliminary design decisions without prohibitive time or computational costs.

8.2 Recommendations for Future Work

The empirical correlations developed as part of the droplet growth model for this work were created using data from several sources. However, among those sources, data for superhydrophilic surfaces was very limited. Most of the data for superhydrophilic surfaces came from the current study's experimental work. As was discussed, the surface suffered some degradation of hydrophilicity throughout the testing. The droplet model would benefit from updated correlations for surface area coverage and the droplet growth exponent, μ, developed using additional data for superhydrophilic surfaces under forced convection conditions. Calculations using both of these correlations affect total mass during the droplet growth stage of frosting, which affects the density of the frost layer after initial nucleation, so improving predictions for these quantities would positively impact predictions for all stages of frosting.

Model validation with a biphilic surface indicated that the model could not predict the freezing time extension observed for that surface, though it could predict the freezing time for both individual surface types at similar test conditions. Additional research into droplet freezing on biphilic surfaces could help explain why two surface types with low freezing times could be combined into a single surface with an average freezing time 5 or 6 times greater.

Frost thickness and density were under-predicted when airflow between parallel plates was modeled as internal flow. Future investigation could explain why measured frost characteristics did not change significantly when the surface configuration changed from an external type to an internal type, and why predictions were consistently more accurate when airflow was treated as external flow.

The current model is useful for calculating average frost layer properties for flat surfaces, frost characteristics when a flat surface is segmented both in the direction of airflow and by wettability type, and frost behavior under folded-flat-fin configurations. The principles governing this model could be applied to many other surface geometries as well, including other fin configurations and entire heat exchangers.

References

Abu-Orabi, M., 1998. Modeling of heat transfer in dropwise condensation. *International Journal of Heat and Mass Transfer 41*(1), pp.81-87.

Adanur, B., 2019. *Experimental methods for the analysis of frost nucleation and frost growth on hydrophilic and hydrophobic coated flat plate surfaces in forced convection channel flows.* Master's thesis, Auburn University, Auburn, AL, USA.

Adanur, B., Cremaschi, L., Harges, E., 2019. "Effect of mixed hydrophilic and hydrophobic surface coatings on droplets freezing and subsequent frost growth during air forced convection channel flows." *Science and Technology for the Built Environment 25*(10), pp. 1302-1312.

Alizadeh, A., Yamada, M., Li, R., Shang, W., Otta, S., Zhong, S., Ge, L., Dhinojwala, A., Conway, K.R., Bahadur, V. and Vinciquerra, A.J., 2012. "Dynamics of ice nucleation on water repellent surfaces." *Langmuir*, *28*(6), pp.3180-3186.

Bell, I.H., Wronski, J., Quoilin, S., Lemort, V., 2014. Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library CoolProp. *Industrial & Engineering Chemistry Research 53*(6), pp.2498-2508.

Beysens, D., 1995. "The formation of dew." *Atmospheric Research 39*(1-3), pp. 215-237.

Beysens, D., Knobler, C.M., 1986. "Growth of breath figures." *Physical Review Letters*, *57*(12), pp.1433-1437.

Biddle, J.W., Holten, V., Sengers, J.V., Anisimov, M.A., 2013. Thermal conductivity of supercooled water. *Physical Review E*, *87*(4), pp.042302 (1-7).

Boreyko, J. B., Collier, C. P., 2013. "Delayed frost growth on jumping-drop superhydrophobic surfaces." *ACS Nano 7*(2), pp. 1618-1627.

Boulogne, F., Salonen, A., 2020. Drop freezing: fine detection of contaminants by measuring the tip angle. *Applied Physics Letters 116*(10), 103701.

Bryant, J., 1995. *Effects of hydrophobic surface treatments on dropwise condensation and freezing of water.* Doctoral dissertation, Texas A&M University.

Burnside, B.M., Hadi, H.A., 1999. Digital computer simulation of dropwise condensation from equilibrium droplet to detectable size. *International Journal of Heat and Mass Transfer 42*(16), pp.3137-3146.

Castillo, J. E., Weibel, J. A., Garimella, S. V., 2015. “The effect of relative humidity on dropwise condensation dynamics.” *International Journal of Heat and Mass Transfer 80*, pp. 759-766.

Cengel, Y.A., Ghajar, A.J., 2011. *Heat and Mass Transfer: Fundamentals & Applications, Fourth Edition*. McGraw-Hill, New York, NY.

Cha, H., Vahabi, H., Wu, A., Chavan, S., Kim, M.K., Sett, S., Bosch, S.A., Wang, W., Kota, A.K. and Miljkovic, N., 2020. “Dropwise condensation on solid hydrophilic surfaces.” *Science advances 6*(2), p.eaax0746.

Chen, Y., Lu, P., Shen, C., Zhang, Q., 2015. Experimental study on frost formation on a cold surface in low atmospheric pressure. *Applied Thermal Engineering 90*, pp. 86-93.

Cheng, C. H., Wu, K. H., 2003. Observations of early-stage frost formation on a cold plate in atmospheric air flow. *Journal of Heat Transfer 125*(1), pp. 95-102.

Chu, F., Wu, X., Ma, Q., 2017. Condensed droplet growth on surfaces with various wettability. *Applied Thermal Engineering 115*, pp.1101-1108.

Chu, F., Wu, X., Wang, L., 2017. Dynamic melting of freezing droplets on ultraslippery superhydrophobic surfaces. *ACS applied materials & interfaces 9*(9), pp.8420-8425.

Chu, F., Wu, X., Zhu, Y. and Yuan, Z., 2017. “Relationship between condensed droplet coalescence and surface wettability.” *International Journal of Heat and Mass Transfer 111*, pp.836-841.

Churchill, S.W., Chu, H.H., 1975. Correlating equations for laminar and turbulent free convection from a vertical plate. *International Journal of Heat and Mass Transfer 18*(11), pp.1323-1329.

Cremaschi, L., Harges, E., Adanur, B., Strong, A., 2018. “Frost nucleation and frost growth on hydrophobic and hydrophilic surfaces for heat exchangers fin structures.” *International Refrigeration and Air Conditioning Conference*, Paper 1994.

Cremaschi, L., Adanur, B. M., Harges, E., 2019. “Effect of humidity and airflow velocity on droplets elapsed time and radius at the onset of freezing and frost nucleation for super-hydrophilic and super-hydrophobic surfaces.” *ASTFE Digital Library*. Begel House Inc.

De Lima, J., Silva Jr., V., de Lima, M., Abrantes, J., Montenegro, A., 2015. “Revisiting simple methods to estimate drop size distributions: a novel approach based on infrared thermography.” *Journal of Hydrology and Hydromechanics 63*(3), pp. 220-227.

El Cheikh, A., Jacobi, A., 2014. A mathematical model for frost growth and densification on flat surfaces. *International Journal of Heat and Mass Transfer 77*, pp. 604-611.

Feistel, R., and Wagner, W., 2007. Sublimation pressure and sublimation enthalpy of H2O ice Ih between 0 and 273.16 K. *Geochimica et Cosmochimica Acta 71*(1), pp. 36-45.

Fritter, D., Knobler, C.M., Roux, D., Beysens, D., 1988. Computer simulations of the growth of breath figures. *Journal of Statistical Physics 52*(5-6), pp.1447-1459.

Fukuta, N., 1969. "Experimental studies on the growth of small ice crystals." *Journal of the Atmospheric Sciences 26*(3), pp. 522-531.

Graham, C., 1969. *The limiting heat transfer mechanisms of dropwise condensation.* Doctoral dissertation, Massachusetts Institute of Technology.

Glicksman, L.R., Hunt Jr, A.W., 1972. Numerical simulation of dropwise condensation. *International Journal of Heat and Mass Transfer 15*(11), pp.2251-2269.

Graham, C., Griffith, P., 1973. Drop size distributions and heat transfer in dropwise condensation. *International Journal of Heat and Mass Transfer 16*(2), pp.337-346.

Grooten, M.H.M., Van der Geld, C.W.M., 2011. "Dropwise condensation from flowing air–steam mixtures: Diffusion resistance assessed by controlled drainage." *International Journal of Heat and Mass Transfer*, *54*(21-22), pp.4507-4517.

Hao, Q., Pang, Y., Zhao, Y., Zhang, J., Feng, J., Yao, S., 2014. "Mechanism of delayed frost growth on superhydrophobic surfaces with jumping condensates: more than interdrop freezing." *Langmuir 30*(51), pp.15416-15422.

Haque, M. R., Betz, A. R., 2018. "Frost formation on aluminum and hydrophobic surfaces." *ASME 2018 16th International Conference on Nanochannels, Microchannels, and Minichannels*. American Society of Mechanical Engineers Digital Collection.

Haque, M., Zhu, C., Qu, C., Kinzel, E.C., Rachel Betz, A., 2020. Experimental investigation of condensation and freezing phenomenon on hydrophilic and hydrophobic titanium nanopillared glass surfaces. *Heat Transfer Engineering*, pp.1-16.

Hare, D.E., Sorensen, C.M., 1987. The density of supercooled water. II. Bulk samples cooled to the homogeneous nucleation limit. *The Journal of Chemical Physics*, *87*(8), pp.4840-4845.

Harges, E., Cremaschi, L., 2018b. Modeling of Frost Growth On Surfaces With Varying Contact Angle. In *ASTFE Digital Library*. Begel House Inc.

Harges, E., Cremaschi, L., Adanur, B., 2019. "A new model for droplets growth during their onset of freezing on flat surfaces under air forced convection." *Proceedings of the 25th IIR International Congress of Refrigeration*, Montreal, Canada. Paper 569.

Harges, E., Cremaschi, L., Adanur, B., 2020. Distribution, coalescence, and freezing characteristics of water droplets on surfaces with different wettabilities under subfreezing convective flow. *Applied Thermal Engineering 182*, 116052.

Hayashi, Y., Aoki, A., Adachi, S., Hori, K., 1977. Study of frost properties correlating with frost formation types. *Journal of Heat Transfer 99*, pp. 239-245.

He, M., Wang, J., Li, H., Song, Y., 2011. Super-hydrophobic surfaces to condensed micro-droplets at temperatures below the freezing point retard ice/frost formation. *Soft Matter 7*(8), pp. 3993-4000.

Henderson-Sellers, B., 1984. A new formula for latent heat of vaporization of water as a function of temperature. *Quarterly Journal of the Royal Meteorological Society*, *110*(466), pp.1186-1190.

Hermes, C.J.L., 2012. "An analytical solution to the problem of frost growth and densification on flat surfaces." *International Journal of Heat and Mass Transfer 55*(23-24), pp. 7346-7351.

Hermes, C. J., Loyola, F. R., Nascimento Jr, V. S., 2014. A semi-empirical correlation for the frost density. *International Journal of Refrigeration 46*, pp. 100-104.

Hermes, C.J., Nascimento Jr, V.S., Loyola, F.R., Cardoso, R.P., Sommers, A.D., 2019. A study of frost build-up on hydrophilic and hydrophobic surfaces under forced convection conditions. *Experimental Thermal and Fluid Science 100*, pp.76-88.

Hermes, C.J.L., Sommers, A.D., Gebhart, C.W., Nascimento, V.S., 2018. A semi-empirical model for predicting the frost accretion on hydrophilic and hydrophobic surfaces. *International Journal of Refrigeration 100*, pp. 76-88.

Hoffer, T. E., 1961. A laboratory investigation of droplet freezing. *Journal of Meteorology 18*(6), pp. 766-778.

Hoke, J.L., Georgiadis, J.G., Jacobi, A.M., 2000. *The interaction between the substrate and frost layer through condensate distribution*. Air Conditioning and Refrigeration Center. College of Engineering. The University of Illinois at Urbana-Champaign.

Iragorry, J., Tao, Y., Jia, S., 2004. A critical review of properties and models for frost formation analysis. *HVAC&R Research 10*(4)*,* pp. 393-420.

Ismail, K. A. R., Salinas, C., 1999. Modeling of frost formation over parallel cold plates. *International Journal of Refrigeration 22*(5), pp. 425-441.

Ismail, K. A. R., Salinas, C., & Goncalves, M. M., 1997. Frost growth around a cylinder in a wet air stream. *International Journal of Refrigeration 20*(2), pp. 106-119.

Jung, S., Dorrestijn, M., Raps, D., Das, A., Megaridis, C. M., Poulikakos, D., 2011. Are superhydrophobic surfaces best for icephobicity? *Langmuir 27*(6)*,* pp. 3059-3066.

Kim, S., Kim, K.J., 2011. Dropwise condensation modeling suitable for superhydrophobic surfaces. *Journal of Heat Transfer 133*(8).

Kim, H., Kim, D., Jang, H., Kim, D.R., Lee, K.S., 2016. Microscopic observation of frost behaviors at the early stage of frost formation on hydrophobic surfaces. *International Journal of Heat and Mass Transfer 97*, pp.861-867.

Kim, K., Lee, K. S., 2011. Frosting and defrosting characteristics of a fin according to surface contact angle. *International Journal of Heat and Mass Transfer 54*(13-14), .pp. 2758-2764.

Kim, H., Park, Y., Buongiorno, J., 2013." Measurement of wetted area fraction in subcooled pool boiling of water using infrared thermography." *Nuclear Engineering and Design 264*, pp. 103-110.

Lambregts, A. A., Nesemeier, G., Wilborn, J. E., Newman, R. L., 2008. Airplane upsets: Old problem, new issues. Presented at AIAA Modeling and Simulation Technologies Conference and Exhibit.

Lee, H., Shin, J., Ha, S., Choi, B., Lee, J., 2004. Frost formation on a plate with different surface hydrophilicity. *International Journal of Heat and Mass Transfer 47*(22), pp. 4881-4893.

Li, D., Chen, Z., Shi, M., 2010. Effect of ultrasound on frost formation on a cold flat surface in atmospheric air flow. *Experimental Thermal and Fluid Science 34*(8), pp.1247-1252.

Liu, X., Cheng, P., 2015. Dropwise condensation theory revisited: Part I. Droplet nucleation radius. *International Journal of Heat and Mass Transfer 83*, pp.833-841.

Lüer, A., Beer, H., 2000. Frost deposition in a parallel plate channel under laminar flow conditions. *International Journal of Thermal Sciences 39*(1), pp. 85-95.

Marrero, T.R., Mason, E.A., 1972. Gaseous diffusion coefficients. *Journal of Physical and Chemical Reference Data*, *1*(1), pp.3-118.

Meakin, P., 1992. "Droplet deposition growth and coalescence." *Reports on Progress in Physics 55*(2), pp. 157-240.

Miljkovic, N., Enright, R., Wang, E.N., 2012. Effect of droplet morphology on growth dynamics and heat transfer during condensation on superhydrophobic nanostructured surfaces. *ACS Nano 6*(2), pp.1776-1785.

Na, B., Webb, R. L., 2004. Mass transfer on and within a frost layer. *International Journal of Heat and Mass Transfer 47*(5), 899-911.

Nath, S., Ahmadi, S.F., Boreyko, J.B., 2017. A review of condensation frosting. *Nanoscale and Microscale Thermophysical Engineering 21*(2), pp.81-101.

Negrelli, S., & Hermes, C. J., 2015. A semi-empirical correlation for the thermal conductivity of frost. *International Journal of Refrigeration 58*, pp. 243-252.

Östin, R., & Andersson, S., 1991. Frost growth parameters in a forced air stream. *International Journal of Heat and Mass Transfer 34*(4-5), pp. 1009-1017.

Padhmanabhan, S., 2011. *Study of frost growth on heat exchangers used as outdoor coils in air source heat pump systems.* Doctoral dissertation, Oklahoma State University.

Petit, J., Bonaccurso, E., 2014. General frost growth mechanism on solid substrates with different stiffness. *Langmuir 30*(4), pp. 1160-1168.

Piucco, R.O., Hermes, C.J., Melo, C., Barbosa Jr, J.R., 2008. A study of frost nucleation on flat surfaces. *Experimental Thermal and Fluid Science 32*(8), pp.1710-1715.

Pruppacher, H. R., 1963. Some relations between the supercooling and the structure of aqueous solutions. *The Journal of Chemical Physics 39*(6), pp. 1586-1594.

Rahimi, M., Afshari, A., Fojan, P., Gurevich, L., 2015. The effect of surface modification on initial ice formation on aluminum surfaces. *Applied Surface Science 355*, pp. 327-333.

Rohsenow, W.M., Hartnett, J.P., Cho, Y.I., 1998. *Handbook of Heat Transfer* (Vol. 3). New York: McGraw-Hill.

Ryan, B.F., Wishart, E.R., Shaw, D.E., 1976. "The growth rates and densities of ice crystals between -3 °C and -21 °C." *Journal of the Atmospheric Sciences 33*(5), pp. 842-850.

Rykaczewski, K., 2012. Microdroplet growth mechanism during water condensation on superhydrophobic surfaces. *Langmuir 28*(20), pp.7720-7729.

Şahin, A. Z. 1995. An analytical study of frost nucleation and growth during the crystal growth period. *Heat and Mass Transfer 30*(5), pp. 321-330.

Sanchez, L.R., Hartfield-Wunsch, S., 2011. "Effects on surface roughness and friction on aluminum sheet under plain strain cyclic bending and tension." *SAE International Journal of Materials and Manufacturing 4*(1), pp.826-834.

Schneider, H. W., 1978. Equation of the growth rate of frost forming on cooled surfaces. *International Journal of Heat and Mass Transfer 21*(8), pp. 1019-1024.

Seki, N., Fukusako, S., Matsuo, K., Uemura, S., 1985. An analysis of incipient frost formation. *Wärme-und Stoffübertragung 19*(1), pp.9-18.

Shen, Y., Jin, M., Wu, X., Tao, J., Luo, X., Chen, H., Lu, Y., Xie, Y., 2019. “Understanding the frosting and defrosting mechanism on the superhydrophobic surfaces with hierarchical structures for enhancing anti-frosting performance.” *Applied Thermal Engineering 156*, pp.111-118.

Sheng, W., Pei, Y., Li, X., Ming, P., Zhao, W., 2020. Effect of surface characteristics on condensate droplets growth. *Applied Thermal Engineering*, p.115260.

Shin, J., Tikhonov, A.V., Kim, C., 2003. Experimental study on frost structure on surfaces with different hydrophilicity: density and thermal conductivity. *Journal of Heat Transfer 125*(1), pp. 84-94.

Sikarwar, B.S., Khandekar, S., Muralidhar, K., 2011. Dropwise condensation on horizontal substrates with and without a wettability gradient. In *8th Minsk International Seminar Heat Pipes, Heat Pumps, Refrigerators, Power Sources*.

Singha, S. K., Das, P. K., Maiti, B., 2018. Influence of salinity on the mechanism of surface icing: Implication to the disappearing freezing singularity. *Langmuir 34*(30), pp. 9064-9071.

Sommers, A. D., Napora, A. C., Truster, N. L., Caraballo, E. J., Hermes, C. J., 2017. A semi-empirical correlation for predicting the frost density on hydrophilic and hydrophobic substrates. *International Journal of Refrigeration 74*, pp. 313-323.

Speedy, R.J., Angell, C.A., 1976. Isothermal compressibility of supercooled water and evidence for a thermodynamic singularity at −45 C. *The Journal of Chemical Physics*, *65*(3), pp.851-858.

Tahavvor, A. R., Yaghoubi, M., 2009. Analysis of early-stage frost formation in natural convection over a horizontal cylinder. *International Journal of Refrigeration 32*(6), pp. 1343-1349.

Tao, Y.-X., Besant, R. W., Mao, Y., 1993. “Characteristics of frost growth on a flat plate during the early growth period.” *ASHRAE Transactions 99*, pp. 746-753.

Tao, Y. X., Besant, R. W., Rezkallah, K. S., 1993b. A mathematical model for predicting the densification and growth of frost on a flat plate. *International Journal of Heat and Mass Transfer 36*(2), pp. 353-363.

Tokura, I., Saito, H., Kishinami, K, 1983. Study on properties and growth rate of frost layers on cold surfaces. *Journal of Heat Transfer 105*, pp. 895-901.

Vemuri, S., Kim, K.J., 2006. An experimental and theoretical study on the concept of dropwise condensation. *International Journal of Heat and Mass Transfer 49*(3-4), pp.649-657.

Vinš, V., Fransen, M., Hykl, J., Hrubý, J., 2015. Surface tension of supercooled water determined by using a counterpressure capillary rise method. *The Journal of Physical Chemistry B, 119*(17), pp.5567-5575.

Viovy, J.L., Beysens, D., Knobler, C.M., 1988. Scaling description for the growth of condensation patterns on surfaces. *Physical Review A 37*(12), pp.4965-4970.

Wang, C. C., Huang, R. T., Sheu, W. J., Chang, Y. J., 2004. Some observations of the frost formation in free convection: with and without the presence of electric field. *International Journal of Heat and Mass Transfer 47*(14-16), pp. 3491-3505.

Wang, Y., Cheng, Y., 2019. Early stage condensation frosting characteristics on plain and nano Al2O3-epoxy mixture-coated brass. *Applied Thermal Engineering 160*, 113971.

Wen, H.W., Jer, R.M., 1976. On the heat transfer in dropwise condensation. *The Chemical Engineering Journal 12*(3), pp.225-231.

Wilson, P. W., Haymet, A. D. J., 2009. Effect of solutes on the heterogeneous nucleation temperature of supercooled water: an experimental determination. *Physical Chemistry Chemical Physics 11*(15), pp. 2679-2682.

Wu, X., Dai, W., Xu, W., Tang, L., 2007a. Mesoscale investigation of frost formation on a cold surface. *Experimental Thermal and Fluid Science 31*(8), pp. 1043-1048.

Wu, X., Dai, W., Shan, X., Wang, W., Tang, L., 2007b. Visual and theoretical analyses of the early stage of frost formation on cold surfaces. *Journal of Enhanced Heat Transfer 14*(3).

Xia, Y., Zhong, Y., Hrnjak, P. S., Jacobi, A. M., 2006. "Frost, defrost, and refrost and its impact on the air-side thermal-hydraulic performance of louvered-fin, flat-tube heat exchangers." *International Journal of Refrigeration 29*(7), pp. 1066-1079.

Yang, L., Quan, X., Cheng, P., Cheng, Z., 2014. A free energy model and availability analysis for onset of condensation on rigid and liquid surfaces in moist air. *International Journal of Heat and Mass Transfer 78*, pp.460-467.

Yoon, S., Hayase, G., Cho, K., 2010. Measurements of frost thickness and frost mass on a flat plate under heat pump condition. *Heat Transfer Engineering 31*(12), pp. 965-972.

Zhao, H., Beysens, D., 1995. From droplet growth to film growth on a heterogeneous surface: condensation associated with a wettability gradient. *Langmuir 11*(2), pp.627-634.

Appendix A – Frost Model C++ Script

The C++ script for the frost model was written in three modules and is presented here in those three sections:

1) Main – this module consists of variable declarations and the main structure of the droplet growth, crystal growth, and frost layer growth models
2) Properties – this module consists of humid air, liquid water, ice, and frost property calculations
3) Other Functions – this module contains all the other functions called from the "main" module, such as those for the heat transfer coefficient, droplet area coverage, frost surface temperature

The script presented here includes the parts of the three modules above actually used for the calculations in this dissertation. The entire script also includes sections that were previously written but "commented out" as obsolete; they were not deleted in case they needed to be referenced in the future, but they are not included here to promote clarity on the actual equations and functions used.

1) Main

```
// Opening source file for the frost model

#include "stdafx.h"
#include <iostream>
#include "CoolProp.h"
#include "HumidAirProp.h"
#include <stdlib.h>
#include <cmath>
#include <sstream>
#include <fstream>
#include <string>
#include <cstddef>

using namespace std;
using namespace CoolProp;
using namespace HumidAir;

// List of all functions used in this source file
void GetInputs(double& InletAirTemp, double& AirPressure, double& InletAirRH, double&
    InletAirVelocity, double& SurfaceTemp, double& ContactAngle1, double& ContactAngle2,
    double& InitialRadius, double& Length, double& Width, double& ChannelWidth, double&
    ChannelHeight, double& WidthRatio, double& NSegL, double& NumPlates);
void CondensateProperties(double& AirPressure, double& InterfaceTemp, double&
    VaporVolume, double& SurfaceTension, double& DropletDensity, double&
    DropletConductivity);
void HumidAirProperties(double& InletAirTemp, double& AirPressure, double& InletAirRH,
    double& Density, double& Viscosity, double& Conductivity, double& SpecificHeat,
    double& Prandtl, double& HumidityRatio, double& PartialPressure);
void ConvectionResistance(double InletAirTemp, double SurfaceTemp, double FilmTemp,
    double AirVelocity, double SideLength, double PlateWidth, double FilmPrandtl, double
    FilmDensity, double FilmViscosity, double FilmConductivity, double FilmSpecificHeat,
    double BulkPartialPressure, double InterfacePartialPressure, double InterfaceTemp,
    double ContactAngle, double Radius, double HeatTransfer, double H_fg, double LewisNo,
    double BulkVaporDensity, double InterfaceVaporDensity, double HTC, double MTC, double
    beta, double BaseRadius, double AreaCoverage, double& ConvectionResistance);
double ConductionResistance(double HeatTransfer, double ContactAngle, double
    Conductivity, double Radius);
double CurvatureResistance(double InterfaceTemp, double SurfaceTension, double Density,
    double H_fg, double Radius);
double InterfacialResistance(double InterfaceTemp, double H_fg, double VaporVolume,
    double ContactAngle, double HeatTransfer, double Radius, double alpha);
double CylinderInterfacialResistance(double CrystalTemp, double H_sub, double
    VaporVolume, double ContactAngle, double HeatTransfer, double Radius, double Height,
    double alpha);
double InterfacialHeatTransferCoefficient(double InterfaceTemp, double H_fg, double
    VaporVolume, double alpha);
```

```
double FreeHTC(double SideLength, double PlateWidth, double FilmViscosity, double
    FilmDensity, double InletAirTemp, double SurfaceTemp, double FilmPrandtl, double
    FilmTemp, double InterfaceTemp, double beta, double FilmConductivity, double
    InletAirRH);
double ForcedHTC(double SideLength, double FilmDensity, double AirVelocity, double
    FilmViscosity, double FilmPrandtl);
double DropletGrowthRate(double InletAirTemp, double InterfaceTemp, double SurfaceTemp,
    double SurfaceTension, double Density, double H_fg, double Radius, double
    Conductivity, double VaporVolume, double ContactAngle, double alpha, double MTC,
    double HTC, double BulkVaporDensity, double InterfaceVaporDensity, double
    AreaCoverage);
void IceProperties(double& Temperature, double& IceDensity, double& IceConductivity);
void FrozenDropletDimensions(double DropletDensity, double IceDensity, double
    ContactAngle, double& DropletHeight, double& Radius, double& DropletVolume, double&
    BaseRadius);
void InitialCrystalGrowth(double GrowthRate, double Radius, double ContactAngle, double
    TimeStep, double AspectRatio, double NumCylinders, double DropletDensity, double
    IceDensity, double& CylinderRadius, double& CylinderHeight);
void CylinderCrystalHeatTransfer(double ContactAngle, double DropletRadius, double
    CrystalRadius, double CrystalHeight, double CrystalTemp, double SurfaceTemp, double
    IceConductivity, double FilmDensity, double FilmViscosity, double AirVelocity, double
    FilmPrandtl, double SideLength, double PlateWidth, double AirTemp, double FilmTemp,
    double FilmConductivity, double NumCylinders, double beta, double InletAirRH, double
    FilmSpecificHeat, double FrostThickness, double BulkVaporDensity, double
    BulkViscosity, double CrystalVaporDensity, double SublimationHeat, double
    CrystalAreaCoverage, double SurfaceAreaCoverage, double ProjectedRadius, double& MTC,
    double& HeatTransfer);
void RadialGrowthRate(double CrystalHeatTransfer, double AirTemp, double CrystalRadius,
    double CrystalHeight, double IceConductivity, double InterfaceTemp, double IceDensity,
    double SideLength, double PlateWidth, double FilmViscosity, double FilmDensity, double
    FilmConductivity, double AirVelocity, double SurfaceTemp, double FilmPrandtl, double
    FilmTemp, double NumCylinders, double IntHTC, double beta, double InletAirRH, double
    FilmSpecificHeat, double FrostThickness, double& CrystalTemp, double& HTC, double&
    RadialGrowthRate);
double CrystalDensity(double CrystalTemp);
double MassTransferCo(double FilmDensity, double FilmSpecificHeat, double HTC, double
    LewisNo, double BulkDensity, double AirVelocity, double CharLength, double
    BulkViscosity, double InterfaceDensity, double FrostTemp);
void DensityMassFlux(double FrostDensity, double FrostTemp, double IceDensity, double
    AirPressure, double AirTemp, double AirHR, double SublimationHeat, double SurfaceTemp,
    double FrostThickness, double PartialPressureSat, double& FrostHR, double&
    DensityMassFlux);
double FrostLayerSurfaceTemp(double DiffusionFlux, double FrostTemp, double SurfaceTemp,
    double FrostThickness, double FrostConductivity, double SurfaceDensity, double
    AirTemp, double FrostDensity, double SublimationHeat, double HTC, double MTC);
double FrostThermalConductivity(double FrostDensity, double time, double ContactAngle,
    double IceDensity, double AirDensity, double IceConductivity, double AirConductivity);
```

```
void CrystalPhaseFrostDensity(double DropletVolume, double Length, double Width, double
    DropletRadius, double AspectRatio, double CrystalRadius, double FrostThickness, double
    IceDensity, double CrystalDensity, double AirDensity, double NumDroplets, double
    NumCylinders, double BaseRadius, double& TotalFrostMass, double& FrostDensity);
double DensityMassFluxHwangCho(double FrostTemp, double AirTemp, double FrostDensity,
    double CrystalDensity, double SublimationHeat, double PartialPressure, double
    Conductivity, double HTC, double MTC, double BulkDensity, double SurfaceDensity);
double LewisNumber(double FilmTemp, double FilmConductivity, double FilmDensity, double
    FilmSpecificHeat);
void Supersaturation(double BulkPartialPressure, double SatBulkPartialPressure, double
    SatSurfacePartialPressure, double& SuperSatDegree, double& SurfacePartialPressure);
void DropletDimensions(double Radius, double ContactAngle, double& DropletHeight, double&
    BaseRadius, double& DropletVolume);
double RoughnessHTC(double SideLength, double AirVelocity, double FilmDensity, double
    FilmViscosity, double FilmPrandtl, double FilmSpecificHeat, double FrostThickness);
double JakobNumber(double SpecificHeat, double H_fg, double Tdew, double Tsurface, double
    OmegaAir, double OmegaSat);
double DewPointTemp(double InletAirTempK, double AirPressurePa, double InletAirRH);
double DropletFreezingTime(double InletAirTemp, double SurfaceTemp, double
    SuperSatDegree, double InletAirVelocity, double ContactAngle);
double DropletCoalescence(double InletAirTemp, double SurfaceTemp, double ContactAngle);
double InitialDropletRadius(double SurfaceTension, double DropletDensity, double
    SurfaceTemp, double SurfaceHumidityRatio, double BulkHumidityRatio);
double SurfaceAreaCoverage(double ProjectedRadius, double ContactAngle);
double NusseltBryant(double Reynolds, double Grashof, double FilmPrandtl);
double NucleationTime(double ContactAngle, double NusseltBryant, double
    BulkHumidityRatio, double SurfaceHumidityRatio, double InletAirTemp);
double HokeDropletDiameter(double ContactAngle, double MTC, double BulkVaporDensity,
    double SurfaceVaporDensity, double DropletDensity, double NumDroplets, double Time,
    double SideLength, double PlateWidth);
double ASTFEDropletFreezingTime(double InletAirRH, double SurfaceTemp, double
    InletAirTemp, double ContactAngle, double InletAirVelocity, double Nusselt);
double CrystalConductionResistance(double CrystalHeatTransfer, double CrystalHeight,
    double CrystalRadius, double CrystalConductivity);
double CrystalRadialGrowthRate(double MTC, double DropletAreaCoverage, double
    CrystalAreaCoverage, double BulkVaporDensity, double CrystalVaporDensity, double
    AspectRatio, double CrystalDensity, double ProjectedRadius, double CylinderRadius,
    double NumCylinders);
double TurbulentNusselt(double HydraulicReynolds, double HydraulicDiameter, double
    PlateLength, double Prandtl);
double CrystalHeatTransfer(double CrystalRadius, double HTC, double MTC, double AirTemp,
    double CrystalTemp, double ProjectedRadius, double SurfaceAreaCoverage, double
    NumCylinders, double SublimationHeat, double BulkVaporDensity, double
    CrystalVaporDensity);
double SublimationLatentHeat(double Temperature);
double EffectiveFrostConductivity(double IceConductivity, double AirConductivity, double
    NumDroplets, double NumCylinders, double DropletVolume, double AspectRatio, double
    CrystalRadius, double Length, double Width, double FrostThickness, double
    SublimationHeat, double AirDensity, double FrostHumidityRatio, double
    SurfaceHumidityRatio, double FrostTemp, double SurfaceTemp, double CrystalVolumeRatio,
    double CrystalHeight, double InitialFrostThickness);
double LocalForcedHTC(double SideLength, double FilmDensity, double AirVelocity, double
    FilmViscosity, double FilmPrandtl);
double LaminarPressureDrop(double FrictionFactor, double PlateLength, double
    HydraulicDiameter, double AirDensity, double AirVelocity);
double LaminarNusselt(double Distance, double HydraulicDiameter, double
    HydraulicReynoldsNo, double Prandtl);
double LocalParallelPlateNusselt(double ChannelHeight, double xdistance, double
    InletAirVelocity, double FilmDensity, double FilmViscosity, double FilmPrandtl, double
    FrostThickness);

// Input variables
double InletAirTemp0;              // inlet humid air temperature for the first segment
                                   [C]
double InletAirRH0;                // inlet humid air relative humidity for the first
                                   segment [-]
double InletAirTemp[100][2];       // inlet humid air temperature for each segment [C]
double AirPressure[100][2];        // pressure of the humid air for each segment
                                   (atmospheric pressure) [kPa]
double AirPressure0;               // pressure of the humid air for the first segment
                                   [kPa]
double InletAirRH[100][2];         // inlet humid air relative humidity for each segment
                                   [-]
double InletAirVelocity[2];        // inlet humid air velocity [m/s]
double InletAirVelocity0;          // inlet humid air velocity at the first segment [m/s]
double SurfaceTemp;                // surface temperature of the fin/plate [C]
double ContactAngle[2];            // contact angle of water on the surface [degrees]
double ContactAngle1;              // primary contact angle [degrees]
double ContactAngle2;              // secondary contact angle [degrees]
double InitialRadius;              // initial radius of condensate droplet [microns]
double SideLength;                 // length of the test plate [m]
double PlateWidth;                 // width of the test plate [m]
double ChannelWidth;               // with of the air flow channel [m]
double ChannelHeight;              // height/thickness of the air flow channel [m]
double WidthRatio;                 // ratio of plate with that has the primary contact
                                   angle [-]
double FreezingTime[2];            // time until droplets freeze and condensation ends [s]


// Time variables
double NucTime;                    // time until droplets freeze and condensation ends
                                   (Bryant, 1995)
double ASTFEFreezingTime;          // freezing time as described in Harges and Cremaschi,
                                   2018b
```

```
double timeF;                     // time at which the model switched from droplet growth
                                  to crystal growth

// Droplet variables
double InterfaceTemp[100][2];     // interface temperature between liquid droplet and
                                  humid air [C]
double Radius[100][2];            // droplet radius of curvature [m]


// Fluid Property variables
double VaporVolume;               // specific volume of saturated water vapor [m^3/kg]
double SurfaceTension;            // surface tension of liquid condensate [N/m]
double DropletDensity;            // density of the condensate droplet [kg/m^3]
double DropletConductivity;       // thermal conductivity of the condensate droplet
                                  [W/m-K]
double H_fg;                      // enthalpy of vaporization of water [J/kg]


// Bulk humid air property variables
double BulkDensity;               // density of bulk humid air [kg/m^3]
double BulkViscosity;             // dynamic viscosity of bulk humid air [N/m^2-s]
double BulkConductivity;          // thermal conductivity of bulk humid air [W/m-K]
double BulkSpecificHeat;          // specific heat of bulk humid air [J/kg-K]
double BulkPrandtl;               // Prandtl number of bulk humid air [-]
double BulkHumidityRatio;         // humidity ratio of bulk humid air [kg/kg]
double BulkPartialPressure;       // partial pressure of the water vapor in the bulk
                                  humid air [Pa]
double BulkPartialPressure0;      // partial pressure of the water vapor in the bulk
                                  humid air at the leading edge [Pa]
double BulkDiffusivity;           // thermal diffusivity of the bulk humid air [m^2/s]
double OutletBulkHumidityRatio;   // humidity ratio at the outlet of a segment [kg/kg]
double OutletAirRH[100][2];       // relative humidity at the outlet of a segment [-]
double OutletAirPressure[100][2]; // air pressure at the outlet of a segment [kPa]


// Saturated humid air property variables at the bulk temperature
double SatBulkDensity;            // density of saturated bulk humid air [kg/m^3]
double SatBulkViscosity;          // dynamic viscosity of saturated bulk humid air
                                  [N/m^2-s]
double SatBulkConductivity;       // thermal conductivity of saturated bulk humid air
                                  [W/m-K]
double SatBulkSpecificHeat;       // specific heat of saturated bulk humid air [J/kg-K]
double SatBulkPrandtl;            // Prandtl number of saturated bulk humid air [-]
double SatBulkHumidityRatio;      // humidity ratio of saturated bulk humid air [kg/kg]
double SatBulkPartialPressure;    // partial pressure of the water vapor in the saturated
                                  bulk humid air [Pa]


// Film humid air property variables
double FilmTemp;                  // film temperature [C]
double FilmDensity;               // density of humid air at the film temperature
                                  [kg/m^3]
double FilmViscosity;             // dynamic viscosity of humid air at the film
                                  temperature [N/m^2-s]
double FilmConductivity;          // thermal conductivity of the humid air at the film
                                  temperature [W/m-K]
double FilmSpecificHeat;          // specific heat of the humid air at the film
                                  temperature [J/kg-K]
double FilmPrandtl;               // Prandtl number of the humid air at the film
                                  temperature [-]
double FilmHumidityRatio;         // humidity ratio of the humid air at the film
                                  temperature [kg/kg]
double FilmPartialPressure;       // partial pressure of the water vapor in the humid air
                                  at the film temperature [Pa]
double FilmRH;                    // relative humidity of the humid air at the film
                                  temperature [-]
double LewisNo;                   // dimensionless Lewis number
double FilmKinematicViscosity;    // kinematic viscosity at the film temperature [m^2/s]


// Droplet-air interface humid air property variables
double InterfaceDensity;          // density of humid air at the interface temperature
                                  [kg/m^3]
double InterfaceViscosity;        // dynamic viscosity of humid air at the interface
                                  temperature [N/m^2-s]
double InterfaceConductivity;     // thermal conductivity of the humid air at the
                                  interface temperature [W/m-K]
double InterfaceSpecificHeat;     // specific heat of the humid air at the interface
                                  temperature [J/kg-K]
double InterfacePrandtl;          // Prandtl number of the humid air at the interface
                                  temperature [-]
double InterfaceHumidityRatio;    // humidity ratio of the humid air at the interface
                                  temperature [kg/kg]
double InterfacePartialPressure;  // partial pressure of the water vapor in the humid air
                                  at the interface temperature [Pa]
double InterfaceRH;               // relative humidity of the humid air at the interface
                                  temperature [-]
double InterfaceVaporDensity;     // vapor density of the humid air at the interface
                                  temperature [kg/m^3]


// Crystal-air interface humid air property variables
double CrystalAirDensity;         // density of humid air at the crystal temperature
                                  [kg/m^3]
double CrystalVaporDensity;       // density of water vapor at the crystal temperature
                                  [kg/m^3]
double CrystalViscosity;          // dynamic viscosity of humid air at the crystal
                                  temperature [N/m^2-s]
double CrystalConductivity;       // thermal conductivity of the humid air at the crystal
                                  temperature [W/m-K]
```

```
double CrystalSpecificHeat;        // specific heat of the humid air at the crystal
                                   temperature [J/kg-K]
double CrystalPrandtl;             // Prandtl number of the humid air at the crystal
                                   temperature [-]
double CrystalHumidityRatio;       // humidity ratio of the humid air at the crystal
                                   temperature [kg/kg]
double CrystalPartialPressure;     // partial pressure of the water vapor in the humid air
                                   at the crystal temperature [Pa]
double CrystalRH;                  // relative humidity of the humid air at the crystal
                                   temperature [-]


// Surface humid air variables
double SurfaceDensity;             // density of humid air at the frost surface
                                   temperature [kg/m^3]
double SurfaceViscosity;           // dynamic viscosity of humid air at the frost surface
                                   temperature [N/m^2-s]
double SurfaceConductivity;        // thermal conductivity of the humid air at the frost
                                   surface temperature [W/m-K]
double SurfaceSpecificHeat;        // specific heat of the humid air at the frost surface
                                   temperature [J/kg-K]
double SurfacePrandtl;             // Prandtl number of the humid air at the frost surface
                                   temperature [-]
double SurfaceHumidityRatio;       // humidity ratio of the humid air at the frost surface
                                   temperature [kg/kg]
double SurfacePartialPressure;     // partial pressure of the water vapor in the humid air
                                   at the frost surface temperature [Pa]
double SatRH;                      // the saturated value of relative humidity (=1) [-]
double SurfaceRH;                  // relative humidity of the humid air at the surface
                                   temperature [-]


// Crystal growth frost surface temperature variables
double FrostTempDensity;           // density of humid air at the frost surface
                                   temperature [kg/m^3]
double FrostTempViscosity;         // dynamic viscosity of humid air at the frost surface
                                   temperature [N/m^2-s]
double FrostTempConductivity;      // thermal conductivity of the humid air at the frost
                                   surface temperature [W/m-K]
double FrostTempSpecificHeat;      // specific heat of the humid air at the frost surface
                                   temperature [J/kg-K]
double FrostTempPrandtl;           // Prandtl number of the humid air at the frost surface
                                   temperature [-]
double FrostTempHumidityRatio;     // humidity ratio of the humid air at the frost surface
                                   temperature [kg/kg]
double FrostTempPartialPressure;   // partial pressure of the water vapor in the humid air
                                   at the frost surface temperature [Pa]
double FrostTempVaporDensity;      // vapor density at the temperature of the frost
                                   surface during crystal growth [kg/m^3]
```

```
// Crystal growth average frost variables
double AvgTemp;                   // average temperature of the crystal phase frost layer
                                  [C]
double AvgDensity;                // density of humid air at the frost surface
                                  temperature [kg/m^3]
double AvgViscosity;              // dynamic viscosity of humid air at the frost surface
                                  temperature [N/m^2-s]
double AvgConductivity;           // thermal conductivity of the humid air at the frost
                                  surface temperature [W/m-K]
double AvgSpecificHeat;           // specific heat of the humid air at the frost surface
                                  temperature [J/kg-K]
double AvgPrandtl;                // Prandtl number of the humid air at the frost surface
                                  temperature [-]
double AvgHumidityRatio;          // humidity ratio of the humid air at the frost surface
                                  temperature [kg/kg]
double AvgPartialPressure;        // partial pressure of the water vapor in the humid air
                                  at the frost surface temperature [Pa]
double AvgDiffusivity;            // diffusivity at the average crystal phase frost
                                  temperature


// Crystal growth average air void variables
double AvgVoidTemp;               // average temperature of the crystal phase frost layer
                                  [C]
double AvgVoidDensity;            // density of humid air at the frost surface
                                  temperature [kg/m^3]
double AvgVoidViscosity;          // dynamic viscosity of humid air at the frost surface
                                  temperature [N/m^2-s]
double AvgVoidConductivity;       // thermal conductivity of the humid air at the frost
                                  surface temperature [W/m-K]
double AvgVoidSpecificHeat;       // specific heat of the humid air at the frost surface
                                  temperature [J/kg-K]
double AvgVoidPrandtl;            // Prandtl number of the humid air at the frost surface
                                  temperature [-]
double AvgVoidHumidityRatio;      // humidity ratio of the humid air at the frost surface
                                  temperature [kg/kg]
double AvgVoidPartialPressure;    // partial pressure of the water vapor in the humid air
                                  at the frost surface temperature [Pa]
double AvgVoidDiffusivity;        // diffusivity at the average crystal phase frost
                                  temperature
double VoidTemp2;                 // void temperature calculated using a different method
                                  [C]


// Ice property variables
double IceDensity;                // density of ice [kg/m^3]
double IceConductivity;           // thermal conductivity of ice [W/m-K]
double SublimationHeat;           // enthalpy of sublimation [J/kg]
```

```
// Resistance variables
double ConvResistance;            // convection resistance [K]
double CondResistance;            // conduction resistance through a single droplet [K]
double CurvResistance;            // curvature resistance for a single droplet [K]
double IntResistance;             // interfacial resistance for a single droplet [K]
double alpha;                     // condensation coefficient [-]
double CylinderCondResistance;    // conduction resistance through cylindrical crystal
                                  [K]
double CylinderConvResistance;    // convection resistance for cylindrical crystal [K]


// Droplet dimensions variables
double DropletHeight[100][2];     // height of condensate/ice droplet [m]
double DropletVolume[100][2];     // volume of condensate/ice droplet [m^3]
double DropletVolumeNew[100][2];  // additional droplet volume variable [m^3]
double BaseRadius[100][2];        // radius of the portion of the droplet contacting the
                                  surface [m]
double NumDroplets[100][2];       // number of droplets on the surface (assuming all are
                                  average size)
double NumDropletsNew[100][2];    // updated number of droplets on the surface [-]
double WettedArea;                // total projected area of all droplets [m^2]


// Crystal dimensions variables
double CylinderHeight[100][2];    // height of a cylinder ice crystal [m]
double CylinderRadius[100][2];    // radius of a cylinder ice crystal [m]
double AspectRatio[100][2];       // aspect ratio of the cylindrical ice crystal, equal
                                  to its height divided by its radius [-]
double AspectRatio2;              // aspect ratio as calculated an alternate way [-]
double InitialAspectRatio;        // starting value for the aspect ratio [-]
double NumCylinders[100][2];      // the number of crystals that forms on each frozen
                                  droplet [-]


// Crystal phase mass flux variables
double FrostThicknessOld[100][2]; // frost thickness at the previous time step [m]
double CylinderRadiusOld[100][2]; // crystal radius at the previous time step [m]
double FrostDensityOld[100][2];   // frost density at the previous time step [kg/m^3]
double MassOld;                   // frost mass at the previous time step [kg]
double MassNew;                   // frost mass at the current time step [kg]
double CrystalGrowthMassFlux;     // mass flux during the crystal growth stage [kg/m^2-s]
double VoidTemp;                  // temperature of the air voids between crystals at the
                                  frost surface [C]
double VoidTempRatio;             // weighting factor for calculating the temperature of
                                  the air voids [-]
double CrystalVolRatio[100][2];   // volume ratio of cylindrical crystals to the total
                                  porous frost layer on top of the droplets
double CrystalIntHTC;             // interfacial heat transfer coefficient for the
                                  crystal
```

```
// Other important variables
double DropletHeatTransfer;        // heat transfer through a single droplet
double SensibleHeatTransfer;       // sensible heat transfer through a droplet [W]
double CharLength;                 // characteristic length [m]
double CharLengthFree;             // characteristic length for natural convection [m]
double Nusselt;                    // Nusselt number [-]
double NusseltB;                   // Nusselt number developed in Bryant, 1995 [-]
double InternalTurbulentNusselt;   // Nusselt number calculated using equations for
                                   internal, turbulent flow [-]
double InternalLaminarNusselt;     // Nusselt number calculated using equations for
                                   internal, laminar flow [-]
double ConvectionHTC;              // convection heat transfer coefficient [W/m^2-K]
double TotalDeltaT;                // total temperature difference between the bulk air
                                   stream and the surface [K]
double TotalCrystalDeltaT;         // total temperature difference between the bulk air
                                   stream and the droplet-air interface [K]
double GrowthRate;                 // droplet radial growth rate [m/s]
double ProjectedGrowthRate;        // radial growth rate of the projected droplet radius
                                   [m/s]
double ProjectedRadius[100][2];    // projected radius of a droplet [m]
double AdjustedProjectedRadius[100][2];          // projected radius for adjusted mass [m]
double AdjustedRadius[100][2];     // radius for adjusted mass [m]
double MicroGrowthRate;            // droplet radial growth rate [microns/s]
double HokeDiameter;               // volume-averaged diameter as calculated by the method
                                   from Hoke et al., 2000 [m]
double time;                       // variable to keep track of how much time has passed
                                   [s]
double TimeStep;                   // time step variable to increment time during the
                                   condensation process [s]
double InitialFrostThickness[100][2];     // frost thickness during droplet growth [m]
double CylinderAreaCoverage[100][2];      // area coverage of cylinder cross-sectional
                                          area on projected droplet area [-]
double CylinderHeatTransfer;       // heat transfer used to grow a cylindrical crystal [W]
double CylinderRadialGrowthRate;   // growth rate of the radius of a cylindrical crystal
                                   [m]
double FrostThickness[100][2];     // total frost thickness [mm]
double CrystalTemp[100][2];        // temperature of a frost crystal [C]
double IceCrystalDensity;          // density of an ice crystal formed by deposition
                                   [kg/m^3]
double ARcheck;                    // variable to check the validity of the aspect ratio
                                   [-]
double HTC;                        // convection heat transfer coefficient [W/m^2-K]
double MTC;                        // mass transfer coefficient [kg/m^2-s]
double AreaCoverage[100][2];       // the ratio of area covered by frozen droplets to the
                                   total plate area
double TotalFrostMass[100][2];     // total frost mass for a segment at the current time
                                   step [kg]
double TotalFrostMassOld[100][2];         // total frost mass for a segment at the
                                          previous time step [kg]
```

```
double FrostMass;                  // temporary frost mass variable [kg]
double Grashof;                    // Grashof number [-]
double Jakob;                      // Jakob number [-]
double DewPoint;                   // dew point temperature [-]
double EffectiveConductivity[100][2];     // volumetrically averaged effective thermal
                                          conductivity of the frost layer during crystal
                                          growth [W/m-K]
double WaterMassTransfer;          // water mass deposited during a given time step [kg]


// Coalescence parameters
double Coalescences;               // number of coalescences [-]
double mu_basic;                   // exponent for average droplet growth for CA = 90
double mu;                         // exponent for average droplet growth


// Frost full growth variables
double FrostSurfaceTemp;           // temperature at the surface of the frost layer [C]
double AvgFrostTemp;               // average temperature in the frost layer [C]
double DiffusionFlux;              // mass flux by diffusion which increases frost layer
                                   density [kg/m^2-s]
double ThicknessFlux;              // mass flux which increases frost layer thickness
                                   [kg/m^2-s]
double TotalFlux;                  // total mass flux to the frost layer [kg/m^2-s]
double FrostTemp[100][2];          // surface temperature of the frost layer [C]
double FrostDensity[100][2];       // average density of the frost layer [kg/m^3]
double FrostConductivity[100][2];         // thermal conductivity of the frost layer
                                          [W/m-K]
double FrostHumidityRatio;         // humidity ratio of the air at the frost surface
                                   [kg/kg]
double FrostThicknessNew[100][2];         // new value of frost thickness after the
                                          calculations for one time step [m]
double FrostDensityNew[100][2];           // new value of frost density after the
                                          calculations for one time step [kg/m^3]
double FrostTempNew[100][2];              // new value of frost surface temperature after
                                          the calculations for one time step [C]
double BulkVaporDensity;           // density of the water vapor at the bulk humid air
                                   conditions [kg/m^3]
double SurfaceVaporDensity;        // density of the water vapor at the surface saturated
                                   conditions [kg/m^3]
double Rvap;                       // specific gas constant of water vapor [J/kg-K]
double FrostDensityDep;            // density at which frost is deposited on the top of
                                   the frost layer [kg/m^3]


// Heat flux variables
double SensibleFlux;               // sensible heat flux [W/m^2]
double LatentFlux;                 // latent heat flux [W/m^2]
double TotalHeatFlux;              // total heat flux [W/m^2]
```

```
// Plate segment variables
double SegmentLength;              // length of a segment [m]
double SegmentWidth;               // width of a segment [m]
double SegmentArea;                // area of a segment [m^2]
double SegmentOutletHumidityRatio;        // bulk humidity ratio at the outlet of a
                                          segment [-]


// Supersaturation variables
double SuperSatDegree;             // supersaturation degree [-]
double SurfaceSuperSatDegree;      // supersaturation degree at the test plate surface [-]
double ActualInterfacePartialPressure;           // interface partial pressure, taking
                                                 supersaturation into account [Pa]


// Reynolds and Nusselt variables
double Reynolds;                   // Reynolds number [-]
double HydraulicReynolds;          // Reynolds number for internal flow using hydraulic
                                   diameter [-]
double HydraulicReynoldsNo;        // temporary variable
double HydraulicDiameter;          // hydraulic diameter of a channel [m]
double beta;                       // thermal expansion coefficient [1/K]
double check;                      // temporary variable
double freeNu;                     // temporary variable for natural convection Nusselt
                                   number
double forcedNu;                   // temporary variable for forced convection Nusselt
                                   number
double Fourier;                    // Fourier number according to Kim et al., 2016
double VolumeFlowRate;             // volume flow rate of air through the test section
                                   [m^3/s]


// Variables for discretization in airflow direction
double NSegL;                      // number of segments the length is divided into
double SegL;                       // counter to keep track of which segment the code is
                                   working on
double LSeg;                       // length of a segment in the airflow direction


// Segmentation Variables
double OutletAirTemp[100][2];      // air temperature at the outlet of a segment [C]
double AirMassFlowRate;            // air mass flow rate over a segment [kg/s]
double WSeg[2];                    // widths of the segments with different contact angles
                                   [m]
double temp[100];                  // temporary variable
double AvgFrostDensity;            // average frost density of the overall frost layer
                                   [kg/m^3]
double AvgFrostDensityNew;         // new average frost density calculated at the current
                                   time step [kg/m^3]
```

```
double CumulativeFrostThickness;          // cumulative frost thickness for the entire
                                          plate [m]
double AvgFrostThickness;          // average frost thickness over the entire plate [m]
double CumulativeFrostDensityNew;         // cumulative frost density over the entire
                                          plate [kg/m^3]
double xdistance;                  // distance along the plate in the direction of
                                   airflow, calculated at the middle of each segment
double xcrit[2];                   // critical distance along air flow for transition from
                                   external to internal flow
double BLcrit[2];                  // critical boundary layer for transition from external
                                   to internal flow
double NumPlates;                  // number of plates (1 or 2)


// Pressure Drop Variables
double FlowHydraulicDiameter;      // hydraulic diameter when considering flow restriction
                                   due to frost [-]
double FlowHydraulicReynolds;      // Reynolds number calculated with the flow hydraulic
                                   diameter [-]
double LaminarFrictionFactor;      // friction factor for laminar flow through a duct
double PressureDrop;               // pressure drop for laminar flow through a duct [Pa]
double TotalPressureDrop[2];       // pressure drop from the front to the back of a plate
                                   [Pa]
double SegmentAirMassFlowRate;     // air mass flow rate for a transverse segment [kg/s]


int main()
{
     int n;                   // counter variable
     int j;                   // counter variable
     int m;                   // counter variable

     // Get primary input values from the user
     GetInputs(InletAirTemp0, AirPressure0, InletAirRH0, InletAirVelocity0, SurfaceTemp,
          ContactAngle1, ContactAngle2, InitialRadius, SideLength, PlateWidth,
          ChannelWidth, ChannelHeight, WidthRatio, NSegL, NumPlates);

     // Assign wettability segment values
     WSeg[0] = PlateWidth*WidthRatio;
     WSeg[1] = PlateWidth - WSeg[0];
     ContactAngle[0] = ContactAngle1;
     ContactAngle[1] = ContactAngle2;


     for (m = 0; m < 2; m++)       // wettability segmentation loop
     {
          // Initialize values at the first segment in the direction of air flow
          InletAirTemp[0][m] = InletAirTemp0;
          InletAirRH[0][m] = InletAirRH0;
          AirPressure[0][m] = AirPressure0;
```

```
InletAirVelocity[m] = InletAirVelocity0;
FrostThickness[0][m] = 0;

// Initialize some fluid properties
// Get condensate properties from CoolProp
CondensateProperties(AirPressure[0][m], SurfaceTemp, VaporVolume,
     SurfaceTension, DropletDensity, DropletConductivity);
// Get bulk humid air properties from CoolProp
HumidAirProperties(InletAirTemp[0][m], AirPressure[0][m], InletAirRH[0][m],
     BulkDensity, BulkViscosity, BulkConductivity, BulkSpecificHeat,
     BulkPrandtl, BulkHumidityRatio, BulkPartialPressure0);
// Get humid air properties at the surface temperature
SatRH = 1;
HumidAirProperties(SurfaceTemp, AirPressure[0][m], SatRH, SurfaceDensity,
     SurfaceViscosity, SurfaceConductivity, SurfaceSpecificHeat,
     SurfacePrandtl, SurfaceHumidityRatio, SurfacePartialPressure);

// Calculate total volume and mass flow rates for the channel
VolumeFlowRate = InletAirVelocity[m] * PlateWidth*ChannelHeight;
AirMassFlowRate = InletAirVelocity[m] * BulkDensity * PlateWidth*ChannelHeight;


for (n = 0; n < NSegL; n++)       // loop for segmentation in the direction of
                                  air flow
{
    SatRH = 1;
    HumidAirProperties(InletAirTemp[0][m], AirPressure[0][m], InletAirRH[0][m],
         BulkDensity, BulkViscosity, BulkConductivity, BulkSpecificHeat,
         BulkPrandtl, BulkHumidityRatio, BulkPartialPressure);
    HumidAirProperties(InletAirTemp[0][m], AirPressure[0][m], SatRH,
         SatBulkDensity, SatBulkViscosity, SatBulkConductivity,
         SatBulkSpecificHeat, SatBulkPrandtl, SatBulkHumidityRatio,
         SatBulkPartialPressure);

    // Calculate the supersaturation degree for the cold surface
    Supersaturation(BulkPartialPressure0, SatBulkPartialPressure,
         SurfacePartialPressure, SurfaceSuperSatDegree,
         ActualInterfacePartialPressure);

    // Calculate the freezing time for both wettability types
    FreezingTime[m] = DropletFreezingTime(InletAirTemp0, SurfaceTemp,
         SurfaceSuperSatDegree, InletAirVelocity0, ContactAngle[m]);
    //FreezingTime[m] = 29.9;

    // Initialize Radius
    InitialRadius = InitialDropletRadius(SurfaceTension, DropletDensity,
         SurfaceTemp, SurfaceHumidityRatio, BulkHumidityRatio);
    Radius[n][m] = InitialRadius;
    ProjectedRadius[n][m] = Radius[n][m] * pow(((1 - cos(ContactAngle[m] *
         3.1416 / 180))*(1 + cos(ContactAngle[m] * 3.1416 / 180))), 0.5);
         // Initialize TimeStep [s]
         TimeStep = 5;
         // Initialize the droplet/air interface temperature [C]
         InterfaceTemp[n][m] = SurfaceTemp + 0.102;

         // Calculate initial surface area coverage
         AreaCoverage[n][m] = SurfaceAreaCoverage(ProjectedRadius[n][m],
               ContactAngle[m]);
         // Calculate the initial wetted area for the segment
         WettedArea = AreaCoverage[n][m] * SideLength*WSeg[m] / NSegL;
         // Calculate the initial number of droplets on the segment
         NumDroplets[n][m] = WettedArea / (3.14159*pow(ProjectedRadius[n][m], 2));
         // Calculate the dimensions of the initial droplets
         DropletDimensions(Radius[n][m], ContactAngle[m], DropletHeight[n][m],
               BaseRadius[n][m], DropletVolume[n][m]);
     }
}

//-------DROPLET GROWTH MODEL-------

// Overall loop for the droplet model calculations
do
{
     time = time + TimeStep;

     do
     {
         // Loop for wettability segmentation
         for (m = 0; m < 2; m++)
         {

             // Loop for segmentation in the direction of air flow
             for (n = 0; n < NSegL; n++)
             {

                 // Calculate the distance along the plate as the midpoint of the
                 current segment
                 xdistance = SideLength / NSegL*(n + 0.5);
                 // Initialize the interface temperature for the new time step
                 InterfaceTemp[n][m] = InterfaceTemp[n][m] + 0.102;

                 // Loop for interface temperature iteration
                 do
                 {
                     InterfaceTemp[n][m] = InterfaceTemp[n][m] - 0.002;

                     // Get condensate properties from CoolProp
                     CondensateProperties(AirPressure[n][m], InterfaceTemp[n][m],
                          VaporVolume, SurfaceTension, DropletDensity,
                          DropletConductivity);
```

```
// Calculate the latent heat of vaporization
// This latent heat calculation comes from Henderson-Sellers,
1984
if (InterfaceTemp[n][m] > 0)
{
     H_fg = 1918460 * pow(((InterfaceTemp[n][m] + 273) /
          (InterfaceTemp[n][m] + 273 - 33.91)), 2);
}
else
{
     H_fg = 1918460 * pow(((0 + 273) / (0 + 273 - 33.91)), 2);
}

// Get bulk humid air properties from CoolProp
HumidAirProperties(InletAirTemp[n][m], AirPressure[n][m],
      InletAirRH[n][m], BulkDensity, BulkViscosity,
      BulkConductivity, BulkSpecificHeat, BulkPrandtl,
      BulkHumidityRatio, BulkPartialPressure);

// Calculate thermal diffusivity [m^2/s] of the bulk humid air
BulkDiffusivity = BulkConductivity /
      (BulkDensity*BulkSpecificHeat);

// Get saturated humid air properties at the bulk temperature
from CoolProp
SatRH = 1;
HumidAirProperties(InletAirTemp[n][m], AirPressure[n][m],
      SatRH, SatBulkDensity, SatBulkViscosity,
      SatBulkConductivity, SatBulkSpecificHeat, SatBulkPrandtl,
      SatBulkHumidityRatio, SatBulkPartialPressure);

// Get humid air properties at the surface temperature
HumidAirProperties(SurfaceTemp, AirPressure[n][m], SatRH,
      SurfaceDensity, SurfaceViscosity, SurfaceConductivity,
      SurfaceSpecificHeat, SurfacePrandtl, SurfaceHumidityRatio,
      SurfacePartialPressure);

// Get humid air properties at the film temperature from
CoolProp
FilmTemp = (InletAirTemp[n][m] + InterfaceTemp[n][m]) / 2;
// The film temperature is the average of the bulk and
interface temperatures
HumidAirProperties(FilmTemp, AirPressure[n][m],
      InletAirRH[n][m], FilmDensity, FilmViscosity,
      FilmConductivity, FilmSpecificHeat, FilmPrandtl,
      FilmHumidityRatio, FilmPartialPressure);
```

```
// Get humid air properties at the interface temperature from
CoolProp
HumidAirProperties(InterfaceTemp[n][m], AirPressure[n][m],
      SatRH, InterfaceDensity, InterfaceViscosity,
      InterfaceConductivity, InterfaceSpecificHeat,
      InterfacePrandtl, InterfaceHumidityRatio,
      InterfacePartialPressure);

// Calculate the supersaturation effects
Supersaturation(BulkPartialPressure, SatBulkPartialPressure,
      InterfacePartialPressure, SuperSatDegree,
      ActualInterfacePartialPressure);
Supersaturation(BulkPartialPressure0, SatBulkPartialPressure,
      SurfacePartialPressure, SurfaceSuperSatDegree,
      ActualInterfacePartialPressure);
// Calculate the densities of the water vapor at the bulk,
interface, and surface conditions
Rvap = 461.5;
BulkVaporDensity = BulkPartialPressure / (Rvap *
      (InletAirTemp[n][m] + 273.15));
InterfaceVaporDensity = InterfacePartialPressure / (Rvap *
      (InterfaceTemp[n][m] + 273.15));
SurfaceVaporDensity = SurfacePartialPressure / (Rvap *
      (SurfaceTemp + 273.15));

// Calculate the Lewis and Jakob numbers
LewisNo = LewisNumber(FilmTemp, FilmConductivity, FilmDensity,
      FilmSpecificHeat);
DewPoint = DewPointTemp(InletAirTemp[n][m], AirPressure[n][m],
      InletAirRH[n][m]);
Jakob = JakobNumber(BulkSpecificHeat, H_fg, DewPoint,
      SurfaceTemp, BulkHumidityRatio, SurfaceHumidityRatio);

// Calculate the hydraulic diameter of the bare channel
HydraulicDiameter = 4 * ChannelWidth*ChannelHeight / (2 *
      ChannelWidth + 2 * ChannelHeight);
// Calculate the hydraulic diameter of the channel assuming
parallel plates and nonzero frost thickness
FlowHydraulicDiameter = 2 * (ChannelHeight -
      NumPlates*InitialFrostThickness[0][m]);
// Calculate the Reynolds number using the parallel plate
hydraulic diameter
FlowHydraulicReynolds = (FilmDensity * InletAirVelocity[m] *
      FlowHydraulicDiameter) / FilmViscosity;
// Calculate the laminar friction factor
LaminarFrictionFactor = 64 / FlowHydraulicReynolds;
// Calculate the Reynolds number using the bare channel
hydraulic diameter
HydraulicReynoldsNo = (FilmDensity * InletAirVelocity[m] *
      HydraulicDiameter) / FilmViscosity;
//HydraulicReynolds = HydraulicReynoldsNo;
```

```
HydraulicReynolds = 1;        // Make this assignment if
                              assuming external laminar flow

// Calculate the heat transfer coefficients
if (HydraulicReynolds < 2300)        // laminar flow
{
    if (InletAirVelocity == 0)       // natural convection
    {
        CharLength = (SideLength * PlateWidth) / (2 * SideLength
        + 2 * PlateWidth);           // horizontal plates
        //CharLength = 4 * (SideLength * PlateWidth) / (2 *
              SideLength + 2 * PlateWidth);
        //CharLength = PlateWidth / 2;
        //CharLength = SideLength;          // vertical plate
        Nusselt = FreeHTC(CharLength, PlateWidth,
              FilmViscosity, FilmDensity, InletAirTemp[n][m],
              SurfaceTemp, FilmPrandtl, FilmTemp,
              InterfaceTemp[n][m], beta, FilmConductivity,
              InletAirRH[n][m]);
    }
    else       // forced convection
    {
        CharLength = SideLength;
        Reynolds = (FilmDensity * InletAirVelocity[m] *
              CharLength) / FilmViscosity;
        CharLengthFree = (SideLength * PlateWidth) / (2 *
              SideLength + 2 * PlateWidth);
        beta = 1 / (FilmTemp + 273.15);    // definition of beta
        FilmKinematicViscosity = FilmViscosity / FilmDensity;
        Grashof = (9.81 * beta * (InletAirTemp[n][m] -
              InterfaceTemp[n][m]) * pow(CharLengthFree, 3)) /
              pow(FilmKinematicViscosity, 2);
        //Grashof = (9.81 * beta * (InletAirTemp -
              InterfaceTemp) * pow(SideLength, 3)) /
              pow(FilmKinematicViscosity, 2);
        check = Grashof / pow(Reynolds, 2);

        if (check < 0.1)      // forced convection dominates
        {
            if (NumPlates == 2)      // parallel plates
            {
                // Calculate the boundary layer thickness for
                the boundary layers to meet in the middle
                BLcrit[m] = ChannelHeight / 2 -
                      InitialFrostThickness[0][m];
                // Calculate the critical distance to transition
                from internal to external flow
                xcrit[m] = pow(BLcrit[m], 2)*pow(FilmPrandtl,
                      0.667)*FilmDensity*InletAirVelocity[m] /
                      (FilmViscosity*24.1);
                if (xdistance < xcrit[m])      // external flow
                {
                    Nusselt = LocalForcedHTC((SideLength /
                         NSegL*(n + 0.5)), FilmDensity,
                         InletAirVelocity[m], FilmViscosity,
                         FilmPrandtl);
                }
                else      // internal flow
                {
                    Nusselt =
                         LocalParallelPlateNusselt(ChannelHeigh
                         t, xdistance, InletAirVelocity[m],
                         FilmDensity, FilmViscosity,
                         FilmPrandtl,
                         InitialFrostThickness[0][m]);
                }
            }
            else      // single plate
            {
                if (NSegL == 1)      // average values over
                entire plate
                {
                    Nusselt = ForcedHTC(SideLength, FilmDensity,
                         InletAirVelocity[m], FilmViscosity,
                         FilmPrandtl);
                }
                else      // multiple segments in air flow
                direction
                {
                    Nusselt = LocalForcedHTC((SideLength /
                         NSegL*(n + 0.5)), FilmDensity,
                         InletAirVelocity[m], FilmViscosity,
                         FilmPrandtl);
                }
            }
        }
        else
        {
            if (check > 10)      // natural convection dominates
            {
                Nusselt = FreeHTC(SideLength, PlateWidth,
                     FilmViscosity, FilmDensity,
                     InletAirTemp[n][m], SurfaceTemp,
                     FilmPrandtl, FilmTemp,
                     InterfaceTemp[n][m], beta,
                     FilmConductivity, InletAirRH[n][m]);
            }
            else      // mixed convection
            {
```

```
freeNu = FreeHTC(SideLength, PlateWidth,
      FilmViscosity, FilmDensity,
      InletAirTemp[n][m], SurfaceTemp,
      FilmPrandtl, FilmTemp,
      InterfaceTemp[n][m], beta,
      FilmConductivity, InletAirRH[n][m]);

if (NumPlates == 2)      // parallel plates
{
    // Calculate the boundary layer thickness
    for the boundary layers to meet in the
    middle
    BLcrit[m] = ChannelHeight / NumPlates -
          InitialFrostThickness[0][m];
    // Calculate the critical distance to
    transition from internal to external flow
    xcrit[m] = pow(BLcrit[m], 2) *
          pow(FilmPrandtl, 0.667) *
          FilmDensity*InletAirVelocity[m] /
          (FilmViscosity*24.1);

    if (xdistance < xcrit[m])   // external flow
    {
        forcedNu = LocalForcedHTC((SideLength /
              NSegL*(n + 0.5)), FilmDensity,
              InletAirVelocity[m],
              FilmViscosity, FilmPrandtl);
    }
    else      // internal flow
    {
        forcedNu =
              LocalParallelPlateNusselt(ChannelH
              eight, xdistance,
              InletAirVelocity[m], FilmDensity,
              FilmViscosity, FilmPrandtl,
              InitialFrostThickness[0][m]);
    }
}
else      // single plate
{
    if (NSegL == 1)      // average values over
    the entire plate
    {
        forcedNu = ForcedHTC(SideLength,
              FilmDensity, InletAirVelocity[m],
              FilmViscosity, FilmPrandtl);
    }
    else      // multiple segments in the air
    flow direction
    {
```

```
                        forcedNu = LocalForcedHTC((SideLength /
                             NSegL*(n + 0.5)), FilmDensity,
                             InletAirVelocity[m],
                             FilmViscosity, FilmPrandtl);
                    }
                }

                // expression for mixed convection Nusselt
                number
                // Cengel and Ghajar (2011)
                Nusselt = pow((pow(forcedNu, 4) + pow(freeNu,
                4)), 0.25);
            }
        }
    }

    if (NSegL == 1)      // average value for heat transfer
    coefficient
    {
        ConvectionHTC = Nusselt * FilmConductivity / CharLength;
    }
    else      // local value for heat transfer coefficient
    {
        ConvectionHTC = Nusselt * FilmConductivity / (SideLength
        / NSegL*(n + 0.5));
    }
}
else      // turbulent internal flow
{
    Nusselt = TurbulentNusselt(HydraulicReynolds,
          HydraulicDiameter, (SideLength/NSegL), FilmPrandtl);
    ConvectionHTC = Nusselt * FilmConductivity /
          HydraulicDiameter;
}

// Calculate Nusselt numbers using other methods for comparison
purposes
NusseltB = NusseltBryant(Reynolds, Grashof, FilmPrandtl);
InternalTurbulentNusselt =
      TurbulentNusselt(HydraulicReynoldsNo, HydraulicDiameter,
      (SideLength / NSegL), FilmPrandtl);
InternalLaminarNusselt = LaminarNusselt(SideLength,
      HydraulicDiameter, HydraulicReynoldsNo, FilmPrandtl);

// Calculate the mass transfer coefficient [kg/m^2-s]
MTC = MassTransferCo(FilmDensity, FilmSpecificHeat,
      ConvectionHTC, LewisNo, BulkDensity, InletAirVelocity[m],
      CharLength, BulkViscosity, InterfaceDensity,
      InterfaceTemp[n][m]);
    // Calculate the heat transfer through a droplet
    if (ContactAngle[m] >= 90)
    {
        DropletHeatTransfer = 3.1416 * pow(Radius[n][m], 2) *
              ConvectionHTC * (InletAirTemp[n][m] -
              InterfaceTemp[n][m]) + (1 / AreaCoverage[n][m]) *
              3.1416 * pow(Radius[n][m], 2) * MTC *
              (BulkVaporDensity - InterfaceVaporDensity) * H_fg;
        SensibleHeatTransfer = 2 * 3.1416 * pow(Radius[n][m], 2) *
              (1 - cos(ContactAngle[m] * 3.1416 / 180)) *
              ConvectionHTC * (InletAirTemp[n][m] -
              InterfaceTemp[n][m]);
    }
    else
    {
        DropletHeatTransfer = 3.1416 * pow(BaseRadius[n][m], 2) *
              ConvectionHTC * (InletAirTemp[n][m] -
              InterfaceTemp[n][m]) + (1 / AreaCoverage[n][m]) *
              3.1416 * pow(BaseRadius[n][m], 2) * MTC *
              (BulkVaporDensity - InterfaceVaporDensity) * H_fg;
        SensibleHeatTransfer = 2 * 3.1416 * pow(BaseRadius[n][m],
              2) * (1 - cos(ContactAngle[m] * 3.1416 / 180)) *
              ConvectionHTC * (InletAirTemp[n][m] -
              InterfaceTemp[n][m]);
    }

    // Calculate the droplet convection resistance
    ConvectionResistance(InletAirTemp[n][m], SurfaceTemp, FilmTemp,
          InletAirVelocity[m], SideLength, PlateWidth, FilmPrandtl,
          FilmDensity, FilmViscosity, FilmConductivity,
          FilmSpecificHeat, BulkPartialPressure,
          InterfacePartialPressure, InterfaceTemp[n][m],
          ContactAngle[m], Radius[n][m], DropletHeatTransfer, H_fg,
          LewisNo, BulkVaporDensity, InterfaceVaporDensity,
          ConvectionHTC, MTC, beta, BaseRadius[n][m],
          AreaCoverage[n][m], ConvResistance);
    // Calculate the droplet conduction resistance
    CondResistance = ConductionResistance(DropletHeatTransfer,
          ContactAngle[m], DropletConductivity, Radius[n][m]);
    // Calculate the droplet curvature resistance
    CurvResistance = CurvatureResistance(InterfaceTemp[n][m],
          SurfaceTension, DropletDensity, H_fg, Radius[n][m]);
    // Calculate the total temperature difference through a droplet
    TotalDeltaT = ConvResistance + CondResistance + CurvResistance;

// Compare the calculated temperature difference to the input
temperature difference
} while ((InterfaceTemp[n][m]) > (SurfaceTemp + 0.002) &&
      (InletAirTemp[n][m] - SurfaceTemp) > (TotalDeltaT + 0.002));
```

```
 // Calculate the droplet growth coefficient
 GrowthRate = DropletGrowthRate(InletAirTemp[n][m],
       InterfaceTemp[n][m], SurfaceTemp, SurfaceTension,
       DropletDensity, H_fg, Radius[n][m], DropletConductivity,
       VaporVolume, ContactAngle[m], alpha, MTC, ConvectionHTC,
       BulkVaporDensity, InterfaceVaporDensity, AreaCoverage[n][m]);

 // Calculate the Fourier number according to Kim et al., 2016
 Fourier = 4290 * pow(Reynolds, 0.231)*pow(BulkHumidityRatio, -
       0.685) * pow((InletAirTemp[n][m] / (InletAirTemp[n][m] -
       SurfaceTemp)), 1.644) * pow((ContactAngle[m] * 3.14159265 /
       180), 0.232);
 // Calculate the freezing time according to Bryant, 1995
 NucTime = NucleationTime(ContactAngle[m], NusseltB,
       BulkHumidityRatio, SurfaceHumidityRatio, InletAirTemp[n][m]);
 // Calculate the freezing time as presented in Harges and
 Cremaschi, 2018b
 ASTFEFreezingTime = ASTFEDropletFreezingTime(InletAirRH[n][m],
       SurfaceTemp, InletAirTemp[n][m], ContactAngle[m],
       InletAirVelocity[m], Nusselt);

 ProjectedGrowthRate = GrowthRate;

 // Calculate the droplet coalescence exponent
 mu = DropletCoalescence(InletAirTemp[n][m], SurfaceTemp,
       ContactAngle[m]);

// Calculate the droplet diameter using the method from Hoke et al.,
2000
HokeDiameter = HokeDropletDiameter(ContactAngle[m], MTC,
      BulkVaporDensity, SurfaceVaporDensity, DropletDensity,
      NumDroplets[n][m], time, SideLength, PlateWidth);

// Calculate the new projected radius according to the power law
formulation for droplet growth
ProjectedRadius[n][m] = ProjectedRadius[n][m] + (1000 *
      ProjectedGrowthRate*(pow(time, mu) - pow((time - TimeStep),
      mu))) / 1000;

// Calculate the radius of curvature
if (ContactAngle[m] < 90)
     Radius[n][m] = ProjectedRadius[n][m] / pow(((1 -
          cos(ContactAngle[m] * 3.1416 / 180))*(1 +
          cos(ContactAngle[m] * 3.1416 / 180))), 0.5);
else
     Radius[n][m] = ProjectedRadius[n][m];

if (ContactAngle[m] < 90)
     AdjustedRadius[n][m] = AdjustedProjectedRadius[n][m] / pow(((1
          - cos(ContactAngle[m] * 3.1416 / 180))*(1 +
          cos(ContactAngle[m] * 3.1416 / 180))), 0.5);
```

```
else
     AdjustedRadius[n][m] = AdjustedProjectedRadius[n][m];


DropletDimensions(Radius[n][m], ContactAngle[m],
      DropletHeight[n][m], BaseRadius[n][m], DropletVolumeNew[n][m]);

// Calculate surface area coverage using a nonlinear curve fit
AreaCoverage[n][m] = SurfaceAreaCoverage(ProjectedRadius[n][m],
      ContactAngle[m]);
// Calculate the wetted area and number of droplets per segment
WettedArea = AreaCoverage[n][m] * SideLength*WSeg[m] / NSegL;
NumDropletsNew[n][m] = WettedArea / (3.14159 *
      pow(ProjectedRadius[n][m], 2));

// Calculate the equivalent frost thickness and mass for the segment
InitialFrostThickness[n][m] = NumDropletsNew[n][m] *
      DropletVolumeNew[n][m] / (SideLength / NSegL*WSeg[m]);
TotalFrostMass[n][m] = NumDropletsNew[n][m] * DropletVolumeNew[n][m]
      * DropletDensity;

// Calculate the outlet air temperature
SensibleFlux = ConvectionHTC*(InletAirTemp[n][m] -
      InterfaceTemp[n][m]);
SegmentAirMassFlowRate = BulkDensity*InletAirVelocity[m] *
      WSeg[m]*(ChannelHeight-NumPlates*FrostThickness[0][m]);
SegmentArea = WSeg[m] * SideLength / NSegL;
OutletAirTemp[n][m] = InletAirTemp[n][m] – SensibleFlux *
      SegmentArea  * NumPlates / (BulkSpecificHeat *
      SegmentAirMassFlowRate);
//Calculate outlet relative humidity
WaterMassTransfer = (DropletVolumeNew[n][m]*NumDropletsNew[n][m] -
      DropletVolume[n][m]*NumDroplets[n][m]) * DropletDensity;
temp[n] = DropletVolumeNew[n][m] - DropletVolume[n][m];
OutletBulkHumidityRatio = BulkHumidityRatio – NumPlates *
      WaterMassTransfer / (SegmentAirMassFlowRate*TimeStep);
OutletAirRH[n][m] = HAPropsSI("R", "T", (OutletAirTemp[n][m] +
      273.15), "P", (AirPressure[n][m] * 1000), "Omega",
      OutletBulkHumidityRatio);
// Calculate the outlet pressure of the segment
PressureDrop = LaminarPressureDrop(LaminarFrictionFactor, SideLength
      / NSegL, FlowHydraulicDiameter, FilmDensity,
      InletAirVelocity[m]);
OutletAirPressure[n][m] = AirPressure[n][m] - PressureDrop / 1000;

// Assign outlet values as inlet values for the next segment
InletAirTemp[n + 1][m] = OutletAirTemp[n][m];
InletAirRH[n + 1][m] = OutletAirRH[n][m];
AirPressure[n + 1][m] = OutletAirPressure[n][m];
DropletVolume[n][m] = DropletVolumeNew[n][m];
NumDroplets[n][m] = NumDropletsNew[n][m];
```

```
        // Print simulation results as output
        cout << "Segment " << n+1 << " of " << NSegL << " and " << m+1 << "
             of 2" << "\n";
        cout << "Length along Plate: " << xdistance << "\n";
        cout << "Critical Distance: " << xcrit[m] << "\n";
        cout << "FreezingTime: " << FreezingTime[m] << "\n";
        //cout << "NucleationTime: " << NucTime << "\n";
        //cout << "InitialRadius [nm]: " << InitialRadius * 1000000000 <<
             "\n";
        //cout << "AreaCoverage: " << AreaCoverage[n][m] << "\n";
        //cout << "Number of Droplets: " << NumDroplets[n][m] << "\n";
        //cout << "Hoke Diameter: " << HokeDiameter * 1000000 << "\n";
        //cout << "InterfaceTemp: " << InterfaceTemp[n][m] << "\n";
        //cout << "H_fg: " << H_fg << "\n";
        //cout << "LewisNo: " << LewisNo << "\n";
        //cout << "SupersaturationDegree: " << SuperSatDegree << "\n";
        //cout << "SurfaceSupersaturationDegree: " << SurfaceSuperSatDegree
             << "\n";
        //cout << "Reynolds: " << Reynolds << "\n";
        //cout << "HTC: " << ConvectionHTC << "\n";
        //cout << "GrowthRate: " << GrowthRate * 1000000 << "\n";
        //cout << "ProjectedGrowthRate: " << ProjectedGrowthRate * 1000000
             << "\n";
        //cout << "DropletRadius: " << Radius[n][m] * 1000000 << "\n";
        //cout << "BaseRadius: " << BaseRadius * 1000000 << "\n";
        cout << "ProjectedRadius: " << ProjectedRadius[n][m] * 1000000 <<
             "\n";
        //cout << "Droplet Growth Exponent, mu: " << mu << "\n";
        //cout << "DropletVolume: " << DropletVolume[n][m] << "\n";
        cout << "Frost Height: " << InitialFrostThickness[n][m] * 1000 <<
             "\n";
        cout << "TotalFrostMass: " << TotalFrostMass[n][m] << "\n";
        cout << "InletTemp: " << InletAirTemp[n][m] << "\n";
        cout << "OutletTemp: " << OutletAirTemp[n][m] << "\n";
        cout << "InletRH: " << InletAirRH[n][m] << "\n";
        cout << "OutletRH: " << OutletAirRH[n][m] << "\n";
        cout << "inletOmega: " << BulkHumidityRatio << "\n";
        cout << "OutletOmega: " << OutletBulkHumidityRatio << "\n";
        cout << "Air Velocity: " << InletAirVelocity[m] << "\n";
        cout << "Time: " << time << "\n\n";

    }

}

// Calculate pressure drop for each wettability segment
for (m = 0; m < 2; m++)
{
    TotalPressureDrop[m] = AirPressure0 - AirPressure[n][m];
}
```

```
    // Adjust the air velocity through each wettability segment based on the
    pressure drops in each segment
    if (TotalPressureDrop[0] > TotalPressureDrop[1])
    {
        InletAirVelocity[0] = InletAirVelocity[0] - 0.01;
        InletAirVelocity[1] = (AirMassFlowRate - BulkDensity*WSeg[0] *
             (ChannelHeight-NumPlates*InitialFrostThickness[0][0]) *
             InletAirVelocity[0]) / (BulkDensity*WSeg[1] * (ChannelHeight-
             NumPlates*InitialFrostThickness[0][1]));

    }
    else if (TotalPressureDrop[1] > TotalPressureDrop[0])
    {
        InletAirVelocity[1] = InletAirVelocity[1] - 0.01;
        InletAirVelocity[0] = (AirMassFlowRate - BulkDensity*WSeg[1] *
             (ChannelHeight - NumPlates*InitialFrostThickness[0][1]) *
             InletAirVelocity[1]) / (BulkDensity*WSeg[0] * (ChannelHeight -
             NumPlates*InitialFrostThickness[0][0]));
    }
    else
    {
        InletAirVelocity[0] = AirMassFlowRate / (BulkDensity*PlateWidth*
             (ChannelHeight – NumPlates * InitialFrostThickness[0][0]));
        InletAirVelocity[1] = InletAirVelocity[0];
        cout << "pressure drop equal\n\n";
        break;
    }

// if pressure drop values are within the tolerance, exit the loop and move on
to the next time step
} while ((TotalPressureDrop[0] > (TotalPressureDrop[1] + 0.00001))||
      (TotalPressureDrop[0] < (TotalPressureDrop[1] - 0.00001)));

FrostMass = 0;

// Calculate the mass of the frost for the entire plate [kg]
for (m = 0; m < 2; m++)
{
    for (n = 0; n < NSegL; n++)
    {
        FrostMass = FrostMass + TotalFrostMass[n][m];
    }
}

cout << "Time: " << time << "\n";
cout << "FrostMass (one plate): " << FrostMass << "\n";
cout << "FrostMass (all plates): " << NumPlates*FrostMass << "\n\n";
```

```
// if the simulation time exceeds the calculated freezing time, exit the droplet
growth model
} while ((time < FreezingTime[0]) && (time < FreezingTime[1]));

// Assign a value for droplet radius to bypass the droplet growth model
//for (m = 0; m < 2; m++)
//{
//       for (n = 0; n < NSegL; n++)
//       {
//              ProjectedRadius[n][m] = 0.000207;
//              //PlateWidth = 0.1397;
//
//              if (ContactAngle[m] < 90)
//                     Radius[n][m] = ProjectedRadius[n][m] / pow(((1 -
                       cos(ContactAngle[m] * 3.1416 / 180))*(1 + cos(ContactAngle[m]
                       * 3.1416 / 180))), 0.5);
//              else
//                     Radius[n][m] = ProjectedRadius[n][m];
//
//              AreaCoverage[n][m] = SurfaceAreaCoverage(ProjectedRadius[n][m],
                ContactAngle[m]);
//              WettedArea = AreaCoverage[n][m] * SideLength*WSeg[m] /
                NSegL;//0.1474;
//              NumDroplets[n][m] = WettedArea / (3.14159*pow(ProjectedRadius[n][m],
                2));
//
//              DropletDimensions(Radius[n][m], ContactAngle[m],
                DropletHeight[n][m], BaseRadius[n][m], DropletVolume[n][m]);
//              InitialFrostThickness[n][m] = NumDroplets[n][m] *
                DropletVolume[n][m] / (SideLength*WSeg[m] / NSegL);
//       }
//}

TimeStep = 2;


//--------CRYSTAL GROWTH MODEL--------


// Calculate dimensions of droplets immediately after freezing
for (m = 0; m < 2; m++)
{

     for (n = 0; n < NSegL; n++)
     {
         // Get ice properties at the interface temperature
         IceProperties(InterfaceTemp[n][m], IceDensity, IceConductivity);
         // Calculate the dimensions of the frozen droplets using conservation of
         mass
```

```
        FrozenDropletDimensions(DropletDensity, IceDensity, ContactAngle[m],
              DropletHeight[n][m], Radius[n][m], DropletVolume[n][m],
              BaseRadius[n][m]);
        ProjectedRadius[n][m] = Radius[n][m]*pow(((1 - cos(ContactAngle[m]*3.1416 /
              180))*(1 + cos(ContactAngle[m]*3.1416 / 180))), 0.5);

        cout << "ProjectedRadius: " << ProjectedRadius[n][m] * 1000000 << "\n";
    }
}

time = time + TimeStep;


for (m = 0; m < 2; m++)      // loop for wettability segmentation
{
    for (n = 0; n < NSegL; n++)   // loop for segmentation in direction of air flow
    {
        // Initialize the crystal area coverage, radius, and aspect ratio
        CylinderAreaCoverage[n][m] = 0.005;
        CylinderRadius[n][m] = 0.000003;
        InitialAspectRatio = -(SurfaceTemp + 3.5) / 3 + 2.5;

        // Initialize the aspect ratio for each segment
        AspectRatio[n][m] = InitialAspectRatio;
        // Calculate the crystal height
        CylinderHeight[n][m] = AspectRatio[n][m] * CylinderRadius[n][m];
        // Calculate the constant number of cylinders per frozen droplet
        NumCylinders[n][m] = CylinderAreaCoverage[n][m] *
              pow(ProjectedRadius[n][m], 2) / pow(CylinderRadius[n][m], 2);

        // Recalculate ice properties
        IceProperties(InterfaceTemp[n][m], IceDensity, IceConductivity);
        // Calculate total frost thickness with the initial crystals
        FrostThickness[n][m] = CylinderHeight[n][m] + InitialFrostThickness[n][m];

        SatRH = 1;
        HumidAirProperties(InterfaceTemp[n][m], AirPressure[n][m], SatRH,
              InterfaceDensity, InterfaceViscosity, InterfaceConductivity,
              InterfaceSpecificHeat, InterfacePrandtl, InterfaceHumidityRatio,
              InterfacePartialPressure);

        // Calculate the density of the ice which composes the crystals
        IceCrystalDensity = CrystalDensity(InterfaceTemp[n][m]);

        // Calculate the average density of the overall frost layer
        CrystalPhaseFrostDensity(DropletVolume[n][m], SideLength / NSegL, WSeg[m],
              Radius[n][m], AspectRatio[n][m], CylinderRadius[n][m],
              FrostThickness[n][m], IceDensity, IceCrystalDensity, InterfaceDensity,
              NumDroplets[n][m], NumCylinders[n][m], BaseRadius[n][m],
              TotalFrostMass[n][m], FrostDensityNew[n][m]);
        // Initialize frost density to be high to ensure density decrease at the
        beginning of the crystal growth model
        FrostDensityNew[n][m] = 1000;      // large so convergence will work
        AvgFrostDensityNew = 1000;
        // Initialize temperature at the tops of the crystals
        CrystalTemp[n][m] = InterfaceTemp[n][m] + 0.1;

    }
}

do      // primary loop for the crystal growth model
{
    time = time + TimeStep;
    AvgFrostDensity = AvgFrostDensityNew;

    do
    {

        for (m = 0; m < 2; m++)     // loop for wettability segmentation
        {

            for (n = 0; n < NSegL; n++)    // loop for sementation in the direction
            of air flow
            {
                // calculate the current distance in the direction of air flow
                xdistance = SideLength / NSegL*(n + 0.5);
                // check value to ensure correct behavior of the crystal growth
                model
                ARcheck = AspectRatio[n][m] / CylinderRadius[n][m] / 1000000;

                FrostDensity[n][m] = FrostDensityNew[n][m];
                // Calculate the latent heat of sublimation
                SublimationHeat = SublimationLatentHeat(CrystalTemp[n][m]);
                // Adjust crystal temperature
                CrystalTemp[n][m] = CrystalTemp[n][m] + 0.1;

                HumidAirProperties(InletAirTemp[n][m], AirPressure[n][m],
                    InletAirRH[n][m], BulkDensity, BulkViscosity, BulkConductivity,
                    BulkSpecificHeat, BulkPrandtl, BulkHumidityRatio,
                    BulkPartialPressure);

                // Calculate the crystal aspect ratio
                AspectRatio[n][m] = AspectRatio[n][m] + 1 / exp((time -
                    FreezingTime[0]) / 20) + 0.02*TimeStep;

                do      // iteration loop for crystal temperature
                {
                    // Adjust crystal temperature
                    CrystalTemp[n][m] = CrystalTemp[n][m] - 0.001;
```

```
// Get crystal properties at the new crystal temperature
IceCrystalDensity = CrystalDensity(CrystalTemp[n][m]);
IceProperties(CrystalTemp[n][m], IceDensity, IceConductivity);

// Get humid air properties at the film temperature from
CoolProp
FilmTemp = (InletAirTemp[n][m] + CrystalTemp[n][m]) / 2;
// The film temperature is the average of the bulk and crystal
temperatures
HumidAirProperties(FilmTemp, AirPressure[n][m],
      InletAirRH[n][m], FilmDensity, FilmViscosity,
      FilmConductivity, FilmSpecificHeat, FilmPrandtl,
      FilmHumidityRatio, FilmPartialPressure);
// Get humid air properties at the crystal temperature from
CoolProp
HumidAirProperties(CrystalTemp[n][m], AirPressure[n][m], SatRH,
      CrystalAirDensity, CrystalViscosity, CrystalConductivity,
      CrystalSpecificHeat, CrystalPrandtl, CrystalHumidityRatio,
      CrystalPartialPressure);
// Calculate the vapor density at the crystal temperature
CrystalVaporDensity = CrystalPartialPressure / (Rvap *
      (CrystalTemp[n][m] + 273.15));
// Recalculate the Lewis number at the new film temperature
LewisNo = LewisNumber(FilmTemp, FilmConductivity, FilmDensity,
       FilmSpecificHeat);

// Calculate the hydraulic diameter of the channel assuming
parallel plates and nonzero frost thickness
FlowHydraulicDiameter = 2 * (ChannelHeight – NumPlates *
      FrostThickness[0][m]);
// Calculate the Reynolds number using the parallel plate
hydraulic diameter
FlowHydraulicReynolds = (FilmDensity * InletAirVelocity[m] *
      FlowHydraulicDiameter) / FilmViscosity;
// Calculate the laminar friction factor
LaminarFrictionFactor = 64 / FlowHydraulicReynolds;

 // Calculate the heat transfer coefficients
if (HydraulicReynolds < 2300)       // laminar flow
{
    // Initialize heat transfer using the initial interface
    temperature and the convection resistance
    if (InletAirVelocity == 0)      // natural convection
    {
        CharLength = (SideLength * PlateWidth) / (2 *
              SideLength + 2 * PlateWidth); // horizontal plates
        //CharLength = 4 * (SideLength * PlateWidth) / (2 *
              SideLength + 2 * PlateWidth);
        //CharLength = PlateWidth / 2;
        //CharLength = SideLength;         // vertical plate
    Nusselt = FreeHTC(SideLength, PlateWidth,
          FilmViscosity, FilmDensity, InletAirTemp[n][m],
          SurfaceTemp, FilmPrandtl, FilmTemp,
          CrystalTemp[n][m], beta, FilmConductivity,
          InletAirRH[n][m]);
}
else        // forced convection
{
    CharLength = SideLength;
    Reynolds = (FilmDensity * InletAirVelocity[m] *
          CharLength) / FilmViscosity;
    CharLengthFree = (SideLength * PlateWidth) / (2 *
          SideLength + 2 * PlateWidth);
    // definition of beta
    beta = 1 / (FilmTemp + 273.15);
    FilmKinematicViscosity = FilmViscosity / FilmDensity;
    Grashof = (9.81 * beta * (InletAirTemp[n][m] -
          InterfaceTemp[n][m]) * pow(CharLengthFree, 3)) /
          pow(FilmKinematicViscosity, 2);
    //Grashof = (9.81 * beta * (InletAirTemp[n] -
          CrystalTemp[n]) * pow(SideLength, 3)) /
          pow(FilmKinematicViscosity, 2);
    check = Grashof / pow(Reynolds, 2);

    if (check < 0.1)      // forced convection dominates
    {
        if (NumPlates == 2)      // parallel plates
        {
            // Calculate the boundary layer thickness for
            the boundary layers to meet in the middle
            BLcrit[m] = ChannelHeight / NumPlates -
                  FrostThickness[0][m];
            // Calculate the critical distance to transition
            from internal to external flow
            xcrit[m] = pow(BLcrit[m], 2)*pow(FilmPrandtl,
                  0.667)*FilmDensity*InletAirVelocity[m] /
                  (FilmViscosity*24.1);

            if (xdistance < xcrit[m])   // external flow
            {
                Nusselt = LocalForcedHTC((SideLength /
                      NSegL*(n + 0.5)), FilmDensity,
                      InletAirVelocity[m], FilmViscosity,
                      FilmPrandtl);
            }
            else        // internal flow
            {
                Nusselt =
                     LocalParallelPlateNusselt(ChannelHeigh
                     t, xdistance, InletAirVelocity[m],
                     FilmDensity, FilmViscosity,
                     FilmPrandtl, FrostThickness[0][m]);
            }
        }
        else        // single plate
        {
            if (NSegL == 1)      // average values over
                                 entire plate
            {
                Nusselt = ForcedHTC(SideLength, FilmDensity,
                     InletAirVelocity[m], FilmViscosity,
                     FilmPrandtl);
            }
            else  // multiple segments in air flow direction
            {
                Nusselt = LocalForcedHTC((SideLength /
                     NSegL*(n + 0.5)), FilmDensity,
                     InletAirVelocity[m], FilmViscosity,
                     FilmPrandtl);
            }
        }
    }
    else
    {
        if (check > 10)   // natural convection dominates
        {
            Nusselt = FreeHTC(SideLength, PlateWidth,
                 FilmViscosity, FilmDensity,
                 InletAirTemp[n][m], SurfaceTemp,
                 FilmPrandtl, FilmTemp, CrystalTemp[n][m],
                 beta, FilmConductivity, InletAirRH[n][m]);
        }
        else        // mixed convection
        {
            freeNu = FreeHTC(SideLength, PlateWidth,
                 FilmViscosity, FilmDensity,
                 InletAirTemp[n][m], SurfaceTemp,
                 FilmPrandtl, FilmTemp, CrystalTemp[n][m],
                 beta, FilmConductivity, InletAirRH[n][m]);

            if (NumPlates == 2)         // parallel plates
            {
                // Calculate the boundary layer thickness
                for the boundary layers to meet in the
                middle
```

```
}


            BLcrit[m] = ChannelHeight / 2 -
                 InitialFrostThickness[0][m];
            // Calculate the critical distance to
            transition from internal to external flow
            xcrit[m] = pow(BLcrit[m], 2) *
                 pow(FilmPrandtl, 0.667) *
                 FilmDensity*InletAirVelocity[m] /
                 (FilmViscosity*24.1);

            if (xdistance < xcrit[m])   // external flow
            {
                forcedNu = LocalForcedHTC((SideLength /
                     NSegL*(n + 0.5)), FilmDensity,
                     InletAirVelocity[m],
                     FilmViscosity, FilmPrandtl);
            }
            else        // internal flow
            {
                forcedNu = LocalParallelPlateNusselt(
                     ChannelHeight, xdistance,
                     InletAirVelocity[m], FilmDensity,
                     FilmViscosity, FilmPrandtl,
                     FrostThickness[0][m]);
            }
        }
        else       // single plate
        {
            if (NSegL == 1)         // average values
                                 over the entire plate
            {
                forcedNu = ForcedHTC(SideLength,
                     FilmDensity, InletAirVelocity[m],
                     FilmViscosity, FilmPrandtl);
            }
            else        // multiple segments in the air
            flow direction
            {
                forcedNu = LocalForcedHTC((SideLength /
                     NSegL*(n + 0.5)), FilmDensity,
                     InletAirVelocity[m],
                     FilmViscosity, FilmPrandtl);
            }
        }

        // expression for mixed convection Nusselt
        number
        // Cengel and Ghajar (2011)
        Nusselt = pow((pow(forcedNu, 4) + pow(freeNu,
             4)), 0.25);
    }
```

```
        }
    }
    if (NSegL == 1)    // average value for heat transfer
    coefficient
    {
        ConvectionHTC = Nusselt * FilmConductivity / CharLength;
    }
    else        // local value for heat transfer coefficient
    {
        ConvectionHTC = Nusselt * FilmConductivity /
              (SideLength / NSegL*(n + 0.5));
    }
}
else        // turbulent internal flow
{
    Nusselt = TurbulentNusselt(HydraulicReynolds,
          HydraulicDiameter, (SideLength/NSegL), FilmPrandtl);
    ConvectionHTC = Nusselt * FilmConductivity /
          HydraulicDiameter;
}

// Calculate the mass transfer coefficient [kg/m^2-s]
MTC = MassTransferCo(FilmDensity, FilmSpecificHeat,
      ConvectionHTC, LewisNo, BulkDensity, InletAirVelocity[m],
      CharLength, BulkViscosity, CrystalAirDensity,
      CrystalTemp[n][m]);

// Calculate the heat transfer through each cylindrical crystal
CylinderHeatTransfer = CrystalHeatTransfer(
      CylinderRadius[n][m], HTC, MTC, InletAirTemp[n][m],
      CrystalTemp[n][m], ProjectedRadius[n][m],
      AreaCoverage[n][m], NumCylinders[n][m], SublimationHeat,
      BulkVaporDensity, CrystalVaporDensity);

// Calculate the crystal conduction resistance
CylinderCondResistance = CrystalConductionResistance(
      CylinderHeatTransfer, CylinderHeight[n][m],
      CylinderRadius[n][m], IceConductivity);
// Calculate the crystal convection resistance
CylinderConvResistance = InletAirTemp[n][m] -
      CrystalTemp[n][m];
// Calculate the total temperature difference for the crystal
thermal resistance network
TotalCrystalDeltaT = CylinderCondResistance +
      CylinderConvResistance;
```

```
// Compare the calculated temperature difference to the input
temperature difference
} while ((CrystalTemp[n][m]) > (InterfaceTemp[n][m] + 0.001) &&
      (InletAirTemp[n][m] - InterfaceTemp[n][m]) >
      (TotalCrystalDeltaT));

// Update initial frost layer property values for the current time
step
TotalFrostMassOld[n][m] = TotalFrostMass[n][m];
FrostDensityOld[n][m] = FrostDensityNew[n][m];
FrostThicknessOld[n][m] = FrostThickness[n][m];

// Calculate the radial growth rate of a cylindrical crystal
CylinderRadialGrowthRate = CrystalRadialGrowthRate(MTC,
      AreaCoverage[n][m], CylinderAreaCoverage[n][m],
      BulkVaporDensity, CrystalVaporDensity, AspectRatio[n][m],
      IceCrystalDensity, ProjectedRadius[n][m],
      CylinderRadius[n][m], NumCylinders[n][m]);

// Calculate temperatures at the frost surface
//VoidTempRatio = 0.4;     // This is an empirical value from Tao et
al., 1993
VoidTempRatio = 27.52*pow((InletAirTemp[n][m] -
      InterfaceTemp[n][m]), -2.44);
// Calculate the temperature of the air between the crystals at the
frost surface
VoidTemp = VoidTempRatio*(InletAirTemp[n][m] + 273.15) + (1 -
      VoidTempRatio)*(InterfaceTemp[n][m] + 273.15) - 273.15;
// Calculate the volume ratio of crystals in the frost layer above
the frozen droplets
CrystalVolRatio[n][m] = 3.1416*pow(CylinderRadius[n][m], 2) *
      CylinderHeight[n][m] * (NumCylinders[n][m] *
      NumDroplets[n][m]) / (CylinderHeight[n][m] *
      SideLength*WSeg[m] / NSegL);
// Calculate the average temperature at the surface of the frost
layer
FrostTemp[n][m] = CrystalVolRatio[n][m] * (CrystalTemp[n][m] +
      273.15) + (1 - CrystalVolRatio[n][m])*(VoidTemp + 273.15) -
      273.15;

// Calculate humid air properties at the new frost surface
temperature
HumidAirProperties(FrostTemp[n][m], AirPressure[n][m], SatRH,
      FrostTempDensity, FrostTempViscosity, FrostTempConductivity,
      FrostTempSpecificHeat, FrostTempPrandtl,
      FrostTempHumidityRatio, FrostTempPartialPressure);
FrostTempVaporDensity = FrostTempPartialPressure / (Rvap *
      (FrostTemp[n][m] + 273.15));
```

```
// Calculate humid air properties at the average temperature in the
frost layer
AvgTemp = (FrostTemp[n][m] + SurfaceTemp) / 2;
HumidAirProperties(AvgTemp, AirPressure[n][m], SatRH, AvgDensity,
      AvgViscosity, AvgConductivity, AvgSpecificHeat, AvgPrandtl,
      AvgHumidityRatio, AvgPartialPressure);
AvgDiffusivity = 0.00000187 * pow((AvgTemp + 273.15), 2.072) /
      pow(100, 2);

// calculate the dimensions and area coverage of the cylindrical
ice crystals
CylinderRadius[n][m] = CylinderRadius[n][m] +
      CylinderRadialGrowthRate * TimeStep;
CylinderHeight[n][m] = CylinderRadius[n][m] * AspectRatio[n][m];
CylinderAreaCoverage[n][m] = NumCylinders[n][m] *
      pow(CylinderRadius[n][m], 2) / pow(ProjectedRadius[n][m], 2);

// Calculate the frost thickness by adding contributions from the
droplet and crystal models
FrostThickness[n][m] = CylinderHeight[n][m] +
      InitialFrostThickness[n][m];

// Calculate humid air properties at the crystal temperature
SatRH = 1;
HumidAirProperties(CrystalTemp[n][m], AirPressure[n][m], SatRH,
      CrystalAirDensity, CrystalViscosity, CrystalConductivity,
      CrystalSpecificHeat, CrystalPrandtl, CrystalHumidityRatio,
      CrystalPartialPressure);
IceCrystalDensity = CrystalDensity(CrystalTemp[n][m]);

//Calculate the average density of the frost layer
CrystalPhaseFrostDensity(DropletVolume[n][m], (SideLength / NSegL),
      WSeg[m], Radius[n][m], AspectRatio[n][m],
      CylinderRadius[n][m], FrostThickness[n][m], IceDensity,
      IceCrystalDensity, FilmDensity, NumDroplets[n][m],
      NumCylinders[n][m], BaseRadius[n][m], TotalFrostMass[n][m],
      FrostDensityNew[n][m]);

// Calculate the effective thermal conductivity of the frost layer
during crystal growth
AvgVoidTemp = (VoidTemp + InterfaceTemp[n][m]) / 2;
HumidAirProperties(AvgVoidTemp, AirPressure[n][m], SatRH,
      AvgVoidDensity, AvgVoidViscosity, AvgVoidConductivity,
      AvgVoidSpecificHeat, AvgVoidPrandtl, AvgVoidHumidityRatio,
      AvgVoidPartialPressure);
```

```
EffectiveConductivity[n][m] = EffectiveFrostConductivity(
      IceConductivity, AvgVoidConductivity, NumDroplets[n][m],
      NumCylinders[n][m], DropletVolume[n][m], AspectRatio[n][m],
      CylinderRadius[n][m], (SideLength / NSegL), WSeg[m],
      FrostThickness[n][m], SublimationHeat, AvgDensity,
      FrostTempHumidityRatio, SurfaceHumidityRatio, FrostTemp[n][m],
      SurfaceTemp, CrystalVolRatio[n][m], CylinderHeight[n][m],
      InitialFrostThickness[n][m]);

// Calculate the crystal radius at the previous time step
CylinderRadiusOld[n][m] = CylinderRadius[n][m] -
      CylinderRadialGrowthRate * TimeStep;
// Calculate the mass flux
CrystalGrowthMassFlux = (TotalFrostMass[n][m] -
      TotalFrostMassOld[n][m]) / (TimeStep * SideLength * WSeg[m] /
      NSegL);

// Calculate the outlet air temperature
SensibleFlux = ConvectionHTC*(InletAirTemp[n][m] -
      CrystalTemp[n][m]);
SegmentAirMassFlowRate = BulkDensity*InletAirVelocity[m] * WSeg[m]
      * (ChannelHeight - NumPlates*FrostThickness[0][m]);
SegmentArea = WSeg[m] * SideLength / NSegL;
OutletAirTemp[n][m] = InletAirTemp[n][m] – SensibleFlux *
      SegmentArea*NumPlates / (BulkSpecificHeat *
      SegmentAirMassFlowRate);
//Calculate outlet relative humidity
WaterMassTransfer = TotalFrostMass[n][m] - TotalFrostMassOld[n][m];
OutletBulkHumidityRatio = BulkHumidityRatio – NumPlates *
      WaterMassTransfer / (SegmentAirMassFlowRate*TimeStep);
OutletAirRH[n][m] = HAPropsSI("R", "T", (OutletAirTemp[n][m] +
      273.15), "P", (AirPressure[n][m] * 1000), "Omega",
      OutletBulkHumidityRatio);
// Calculate the outlet pressure of the segment
PressureDrop = LaminarPressureDrop(LaminarFrictionFactor,
      SideLength / NSegL, FlowHydraulicDiameter, FilmDensity,
      InletAirVelocity[m]);
OutletAirPressure[n][m] = AirPressure[n][m] - PressureDrop / 1000;

// Assign outlet values as inlet values for the next segment
InletAirTemp[n + 1][m] = OutletAirTemp[n][m];
InletAirRH[n + 1][m] = OutletAirRH[n][m];
AirPressure[n + 1][m] = OutletAirPressure[n][m];


// Print simulation results as output
cout << "Segment " << n + 1 << " of " << NSegL << " and " << m + 1
      << " of 2" << "\n";
cout << "Length along Plate: " << xdistance << "\n";
cout << "Critical Distance: " << xcrit[m] << "\n";
```

```
        //cout << "HeatTransfer: " << CylinderHeatTransfer << "\n";
        //cout << "CrystalRadialGrowthRate: " << CylinderRadialGrowthRate *
              1000000 << "\n";
        //cout << "CylinderRadius: " << CylinderRadius[n][m] * 1000000 <<
              "\n";
        //cout << "CylinderHeight: " << CylinderHeight[n][m] * 1000000 <<
              "\n";
        cout << "CrystalTemp: " << CrystalTemp[n][m] << "\n";
        cout << "FrostTemp: " << FrostTemp[n][m] << "\n";
        cout << "Frost Thickness: " << FrostThickness[n][m] * 1000 << "\n";
        cout << "FrostDensity: " << FrostDensityNew[n][m] << "\n";
        cout << "EffectiveConductivity: " << EffectiveConductivity[n][m] <<
              "\n";
        //cout << "TotalFlux: " << CrystalGrowthMassFlux << "\n";
        cout << "TotalFrostMass: " << TotalFrostMass[n][m] << "\n";
        cout << "InletTemp: " << InletAirTemp[n][m] << "\n";
        //cout << "OutletTemp: " << OutletAirTemp[n][m] << "\n";
        cout << "InletRH: " << InletAirRH[n][m] << "\n";
        //cout << "OutletRH: " << OutletAirRH[n][m] << "\n";
        cout << "BulkHumidityRatio: " << BulkHumidityRatio << "\n";
        cout << "OutletHumidityRatio: " << OutletBulkHumidityRatio << "\n";
        //cout << "ARcheck: " << ARcheck << "\n";
        //cout << "HTC: " << HTC << "\n";
        //cout << "TotalFrostMass: " << TotalFrostMass << "\n";
        cout << "InletAirVelocity: " << InletAirVelocity[m] << "\n";
        cout << "AspectRatio: " << AspectRatio[n][m] << "\n";
        cout << "Time: " << time << "\n\n";

    }

}

// Calculate pressure drop for each wettability segment
for (m = 0; m < 2; m++)
{
    TotalPressureDrop[m] = AirPressure0 - AirPressure[n][m];
}

// Adjust the air velocity through each wettability segment based on the
pressure drops in each segment
if (TotalPressureDrop[0] > TotalPressureDrop[1])
{
    InletAirVelocity[0] = InletAirVelocity[0] - 0.01;
    InletAirVelocity[1] = (AirMassFlowRate - BulkDensity*WSeg[0] *
          (ChannelHeight – NumPlates * FrostThickness[0][0]) *
          InletAirVelocity[0]) / (BulkDensity*WSeg[1] * (ChannelHeight -
          NumPlates*FrostThickness[0][1]));
}
else if (TotalPressureDrop[1] > TotalPressureDrop[0])
{
```

```
            InletAirVelocity[1] = InletAirVelocity[1] - 0.01;
            InletAirVelocity[0] = (AirMassFlowRate - BulkDensity*WSeg[1] *
                  (ChannelHeight - NumPlates*FrostThickness[0][1]) *
                  InletAirVelocity[1]) / (BulkDensity*WSeg[0] * (ChannelHeight –
                  NumPlates * FrostThickness[0][0]));
        }
        else
        {
            InletAirVelocity[0] = AirMassFlowRate / (BulkDensity*PlateWidth*
                  (ChannelHeight - NumPlates*FrostThickness[0][0]));
            InletAirVelocity[1] = InletAirVelocity[0];
            cout << "pressure drop equal\n\n";
            break;
        }

    // if pressure drop values are within the tolerance, exit the loop and move on
    to the next time step
    } while ((TotalPressureDrop[0] > (TotalPressureDrop[1] + 0.00001)) ||
          (TotalPressureDrop[0] < (TotalPressureDrop[1] - 0.00001)));

    FrostMass = 0;
    CumulativeFrostThickness = 0;
    CumulativeFrostDensityNew = 0;

    // Calculate the average mass, thickness, and density of the frost for the
    entire plate [kg]
    for (m = 0; m < 2; m++)
    {
        for (j = 0; j < NSegL; j++)
            {
                FrostMass = FrostMass + TotalFrostMass[j][m];
                CumulativeFrostThickness = CumulativeFrostThickness +
                      FrostThickness[j][m];
                CumulativeFrostDensityNew = CumulativeFrostDensityNew +
                      FrostDensityNew[j][m];
            }
    }

    AvgFrostThickness = CumulativeFrostThickness / (NSegL * 2);
    AvgFrostDensityNew = CumulativeFrostDensityNew / (NSegL * 2);

    cout << "Time: " << time << "\n";
    cout << "FrostMass (one plate): " << FrostMass << "\n";
    cout << "FrostMass (all plates): " << NumPlates*FrostMass << "\n";
    cout << "AverageFrostThickness: " << AvgFrostThickness << "\n";
    cout << "AverageFrostDensity: " << AvgFrostDensityNew << "\n\n";

// Criterion for exiting the crystal growth model (Hayashi et al., 1977)
} while (AvgFrostDensityNew < AvgFrostDensity);
```

```
for (m = 0; m < 2; m++)
{
     for (n = 0; n < NSegL; n++)
     {
         FrostTempNew[n][m] = FrostTemp[n][m];
     }
}


//-------FULL GROWTH MODEL--------
// Sankar's Thesis (2011)


// Recalculate the latent heat of sublimation
SublimationHeat = SublimationLatentHeat(FrostTemp[n][m]);

// Overall loop for the frost layer growth model
do
{
     time = time + TimeStep;

         do
         {
             // Loop for wettability segmentation
             for (m = 0; m < 2; m++)
             {
                 // Loop for segmentation in the direction of air flow
                 for (n = 0; n < NSegL; n++)
                 {
                     // Calculate the distance along the plate as the midpoint of
                     the current segment
                     xdistance = SideLength / NSegL*(n + 0.5);

                     // Loop for frost surface temperature iteration
                     do
                     {
                         // Update the frost surface temperature
                         FrostTemp[n][m] = FrostTempNew[n][m];

                         // Get humid air properties at the inlet air temperature,
                         the film temperature, and the frost surface temperature
                         FilmTemp = (InletAirTemp[n][m] + FrostTemp[n][m]) / 2;
                         // The film temperature is the average of the bulk and
                         interface temperatures
                         SatRH = 1;
                         HumidAirProperties(FilmTemp, AirPressure[n][m],
                               InletAirRH[n][m], FilmDensity, FilmViscosity,
                               FilmConductivity, FilmSpecificHeat, FilmPrandtl,
                               FilmHumidityRatio, FilmPartialPressure);
```

```
HumidAirProperties(InletAirTemp[n][m], AirPressure[n][m],
      InletAirRH[n][m], BulkDensity, BulkViscosity,
      BulkConductivity, BulkSpecificHeat, BulkPrandtl,
      BulkHumidityRatio, BulkPartialPressure);
HumidAirProperties(FrostTemp[n][m], AirPressure[n][m],
      SatRH, FrostTempDensity, FrostTempViscosity,
      FrostTempConductivity, FrostTempSpecificHeat,
      FrostTempPrandtl, FrostTempHumidityRatio,
      FrostTempPartialPressure);

// Get ice and air properties at the average frost
temperature
AvgFrostTemp = (FrostTemp[n][m] + SurfaceTemp) / 2;
// average temp in the frost layer [C]
IceProperties(AvgFrostTemp, IceDensity, IceConductivity);
HumidAirProperties(AvgFrostTemp, AirPressure[n][m], SatRH,
      AvgDensity, AvgViscosity, AvgConductivity,
      AvgSpecificHeat, AvgPrandtl, AvgHumidityRatio,
      AvgPartialPressure);

// Recalculate the Lewis number
LewisNo = LewisNumber(FilmTemp, FilmConductivity,
      FilmDensity, FilmSpecificHeat);

// Calculate the hydraulic diameter of the channel assuming
parallel plates and nonzero frost thickness
FlowHydraulicDiameter = 2 * (ChannelHeight – NumPlates *
      FrostThickness[0][m]);
// Calculate the Reynolds number using the parallel plate
hydraulic diameter
FlowHydraulicReynolds = (FilmDensity * InletAirVelocity[m]
      * FlowHydraulicDiameter) / FilmViscosity;
// Calculate the laminar friction factor
LaminarFrictionFactor = 64 / FlowHydraulicReynolds;

// Calculate the heat transfer coefficients
if (HydraulicReynolds < 2300)    // laminar flow
{
    if (InletAirVelocity == 0)   // natural convection
    {
        // horizontal plates
        CharLength = (SideLength * PlateWidth) / (2 *
              SideLength + 2 * PlateWidth);
        //CharLength = 4 * (SideLength * PlateWidth) / (2 *
              SideLength + 2 * PlateWidth);
        //CharLength = PlateWidth / 2;
        //CharLength = SideLength;      // vertical plate
    Nusselt = FreeHTC(SideLength, PlateWidth,
          FilmViscosity, FilmDensity,
          InletAirTemp[n][m], SurfaceTemp, FilmPrandtl,
          FilmTemp, FrostTemp[n][m], beta,
          FilmConductivity, InletAirRH[n][m]);
}
else        // forced convection
{
    CharLength = SideLength;
    Reynolds = (FilmDensity * InletAirVelocity[m] *
          CharLength) / FilmViscosity;
    CharLengthFree = (SideLength * PlateWidth) / (2 *
          SideLength + 2 * PlateWidth);
    // definition of beta
    beta = 1 / (FilmTemp + 273.15);
    FilmKinematicViscosity = FilmViscosity /
          FilmDensity;
    Grashof = (9.81 * beta * (InletAirTemp[n][m] -
          InterfaceTemp[n][m]) * pow(CharLengthFree, 3))
          / pow(FilmKinematicViscosity, 2);
    //Grashof = (9.81 * beta * (InletAirTemp[n] -
          FrostTemp[n]) * pow(SideLength, 3)) /
          pow(FilmKinematicViscosity, 2);
    check = Grashof / pow(Reynolds, 2);

    if (check < 0.1)  // forced convection dominates
    {
        if (NumPlates == 2)         // parallel plates
        {
            // Calculate the boundary layer thickness
            for the boundary layers to meet in the
            middle
            BLcrit[m] = ChannelHeight / NumPlates -
                  FrostThickness[0][m];
            // Calculate the critical distance to
            transition from internal to external flow
            xcrit[m] = pow(BLcrit[m], 2) *
                  pow(FilmPrandtl, 0.667)*FilmDensity *
                  InletAirVelocity[m] / (FilmViscosity *
                  24.1);

            if (xdistance < xcrit[m])   // external flow
            {
                Nusselt = LocalForcedHTC((SideLength /
                      NSegL*(n + 0.5)), FilmDensity,
                      InletAirVelocity[m],
                      FilmViscosity, FilmPrandtl);
            }
            else        // internal flow
            {
```

```
            Nusselt = LocalParallelPlateNusselt(
                 ChannelHeight, xdistance,
                 InletAirVelocity[m], FilmDensity,
                 FilmViscosity, FilmPrandtl,
                 FrostThickness[0][m]);
        }
    }
    else        // single plate
    {
        if (NSegL == 1)        // average values
        over entire plate
        {
            Nusselt = ForcedHTC(SideLength,
                 FilmDensity, InletAirVelocity[m],
                 FilmViscosity, FilmPrandtl);
        }
        else        // multiple segments in air flow
        direction
        {
            Nusselt = LocalForcedHTC((SideLength /
                 NSegL*(n + 0.5)), FilmDensity,
                 InletAirVelocity[m],
                 FilmViscosity, FilmPrandtl);
        }
    }
}
else
{
    if (check > 10)  // natural convection dominates
    {
        Nusselt = FreeHTC(SideLength, PlateWidth,
             FilmViscosity, FilmDensity,
             InletAirTemp[n][m], SurfaceTemp,
             FilmPrandtl, FilmTemp,
             FrostTemp[n][m], beta,
             FilmConductivity, InletAirRH[n][m]);
    }
    else        // mixed convection
    {
        freeNu = FreeHTC(SideLength, PlateWidth,
             FilmViscosity, FilmDensity,
             InletAirTemp[n][m], SurfaceTemp,
             FilmPrandtl, FilmTemp,
             FrostTemp[n][m], beta,
             FilmConductivity, InletAirRH[n][m]);

        if (NumPlates == 2)    // parallel plates
        {
```

```
    // Calculate the boundary layer
    thickness for the boundary layers to
    meet in the middle
    BLcrit[m] = ChannelHeight / 2 -
          InitialFrostThickness[0][m];
    // Calculate the critical distance to
    transition from internal to external
    flow
    xcrit[m] = pow(BLcrit[m],
          2)*pow(FilmPrandtl, 0.667) *
          FilmDensity*InletAirVelocity[m] /
          (FilmViscosity*24.1);

    if (xdistance < xcrit[m])
    // external flow
    {
        forcedNu =
             LocalForcedHTC((SideLength /
             NSegL*(n + 0.5)), FilmDensity,
             InletAirVelocity[m],
             FilmViscosity, FilmPrandtl);
    }
    else        // internal flow
    {
        forcedNu =
             LocalParallelPlateNusselt(
             ChannelHeight, xdistance,
             InletAirVelocity[m],
             FilmDensity, FilmViscosity,
             FilmPrandtl,
             FrostThickness[0][m]);
    }
}
else       // single plate
{
    if (NSegL == 1)     // average values
    over the entire plate
    {
        forcedNu = ForcedHTC(SideLength,
             FilmDensity,
             InletAirVelocity[m],
             FilmViscosity, FilmPrandtl);
    }
    else        // multiple segments in the
    air flow direction
    {
```

```
                        forcedNu =
                              LocalForcedHTC((SideLength /
                              NSegL*(n + 0.5)), FilmDensity,
                              InletAirVelocity[m],
                              FilmViscosity, FilmPrandtl);
                    }
                }

                // expression for mixed convection Nusselt
                number
                // Cengel and Ghajar (2011)
                Nusselt = pow((pow(forcedNu, 4) +
                      pow(freeNu, 4)), 0.25);
            }
        }
    }

    if (NSegL == 1)       // average value for heat transfer
    coefficient
    {
        ConvectionHTC = Nusselt * FilmConductivity /
              CharLength;
    }
    else        // local value for heat transfer coefficient
    {
        ConvectionHTC = Nusselt * FilmConductivity /
              (SideLength / NSegL*(n + 0.5));
    }
}
else        // turbulent internal flow
{
    Nusselt = TurbulentNusselt(HydraulicReynolds,
          HydraulicDiameter, (SideLength/NSegL),
          FilmPrandtl);
    ConvectionHTC = Nusselt * FilmConductivity /
          HydraulicDiameter;
}

// Calculate the mass transfer coefficient [kg/m^2-s]
MTC = MassTransferCo(FilmDensity, FilmSpecificHeat,
      ConvectionHTC, LewisNo, BulkDensity,
      InletAirVelocity[m], CharLength, BulkViscosity,
      FrostTempDensity, FrostTemp[n][m]);

// Calculate the densities of the water vapor at the bulk
and surface conditions
Rvap = 461.5;
BulkVaporDensity = BulkPartialPressure / (Rvap *
      (InletAirTemp[n][m] + 273.15));
    FrostTempVaporDensity = FrostTempPartialPressure / (Rvap *
          (FrostTemp[n][m] + 273.15));

    // Calculate the sensible, latent, and total heat fluxes
    [W/m^2]
    SensibleFlux = ConvectionHTC * (InletAirTemp[n][m] -
          FrostTemp[n][m]);
    LatentFlux = MTC * (BulkVaporDensity -
          FrostTempVaporDensity) * SublimationHeat;
    TotalHeatFlux = SensibleFlux + LatentFlux;

    // Calculate the thermal conductivity of the frost layer
    [W/m-K]
    IceCrystalDensity = CrystalDensity(AvgFrostTemp);
    FrostConductivity[n][m] = FrostThermalConductivity(
          FrostDensity[n][m], time, ContactAngle[m],
          IceCrystalDensity, AvgDensity, IceConductivity,
          AvgConductivity);

    // Calculate the diffusion mass flux that leads to
    densification of the frost layer [kg/m^2-s]
    IceCrystalDensity = CrystalDensity(FrostTemp[n][m]);
    DensityMassFlux(FrostDensity[n][m], FrostTemp[n][m],
          IceCrystalDensity, AirPressure[n][m],
          InletAirTemp[n][m], BulkHumidityRatio,
          SublimationHeat, SurfaceTemp, FrostThickness[n][m],
          FrostTempPartialPressure, FrostTempHumidityRatio,
          DiffusionFlux);

    // Calculate the total mass flux to the frost layer
    [kg/m^2-s]
    TotalFlux = MTC * (BulkVaporDensity -
          FrostTempVaporDensity);

    // Calculate new frost surface temperature [C]
    FrostTempNew[n][m] = FrostLayerSurfaceTemp(DiffusionFlux,
          FrostTemp[n][m], SurfaceTemp, FrostThickness[n][m],
          FrostConductivity[n][m], FrostTempVaporDensity,
          InletAirTemp[n][m], BulkVaporDensity, SublimationHeat,
          ConvectionHTC, MTC);

    // Calculate the mass flux which leads to thickening of the
    frost layer [kg/m^2-s]
    ThicknessFlux = TotalFlux - DiffusionFlux;

    cout << "\nFrostTempNew: " << FrostTempNew[n][m];

// iterate until the frost surface temperature reaches
convergence
} while ((FrostTempNew[n][m] - FrostTemp[n][m]) > 0.0001);
```

```
// Set a value for the density of deposition of frost [kg/m^3]
FrostDensityDep = FrostDensity[n][m];

// Calculate new frost layer thickness [m]
FrostThicknessNew[n][m] = FrostThickness[n][m] + ThicknessFlux
      / (FrostDensityDep)* TimeStep;

// Calculate new frost layer average density [kg/m^3]
//FrostDensityNew = FrostDensity + DiffusionFlux /
      FrostThickness * TimeStep;
FrostDensityNew[n][m] = FrostThickness[n][m] /
      FrostThicknessNew[n][m] * (FrostDensity[n][m] +
      DiffusionFlux / FrostThickness[n][m] * TimeStep) +
      (FrostThicknessNew[n][m] - FrostThickness[n][m]) /
      FrostThicknessNew[n][m] * (FrostDensityDep);

// Calculate the outlet air temperature
SensibleFlux = ConvectionHTC*(InletAirTemp[n][m] -
      FrostTemp[n][m]);
SegmentAirMassFlowRate = BulkDensity*InletAirVelocity[m] *
      WSeg[m] * (ChannelHeight – NumPlates *
      FrostThickness[0][m]);
SegmentArea = WSeg[m] * SideLength / NSegL;
OutletAirTemp[n][m] = InletAirTemp[n][m] – SensibleFlux *
      SegmentArea*NumPlates / (BulkSpecificHeat *
      SegmentAirMassFlowRate);
//Calculate outlet relative humidity
WaterMassTransfer = TotalFlux*SegmentArea*TimeStep;
OutletBulkHumidityRatio = BulkHumidityRatio – NumPlates *
      WaterMassTransfer / (SegmentAirMassFlowRate*TimeStep);
OutletAirRH[n][m] = HAPropsSI("R", "T", (OutletAirTemp[n][m] +
      273.15), "P", (AirPressure[n][m] * 1000), "Omega",
      OutletBulkHumidityRatio);
// Calculate the outlet pressure of the segment
PressureDrop = LaminarPressureDrop(LaminarFrictionFactor,
      SideLength / NSegL, FlowHydraulicDiameter, FilmDensity,
      InletAirVelocity[m]);
OutletAirPressure[n][m] = AirPressure[n][m] - PressureDrop /
      1000;

// Assign outlet values as inlet values for the next segment
InletAirTemp[n + 1][m] = OutletAirTemp[n][m];
InletAirRH[n + 1][m] = OutletAirRH[n][m];
AirPressure[n + 1][m] = OutletAirPressure[n][m];

// Iterate all values and move on to the next time step
FrostThickness[n][m] = FrostThicknessNew[n][m];
FrostDensity[n][m] = FrostDensityNew[n][m];
FrostTemp[n][m] = FrostTempNew[n][m];
```

```
       TotalFrostMass[n][m] = TotalFrostMass[n][m] + TotalFlux *
            (SideLength * WSeg[m] / NSegL * TimeStep);


       // Print simulation results as output
       cout << "\n\nSegment " << n + 1 << " of " << NSegL << " and "
            << m + 1 << " of 2" << "\n";
       cout << "Length along Plate: " << xdistance << "\n";
       cout << "Critical Distance: " << xcrit[m] << "\n";
       cout << "FrostThickness: " << FrostThickness[n][m] * 1000 <<
            "\n";
       cout << "FrostDensity: " << FrostDensity[n][m] << "\n";
       cout << "FrostTemp: " << FrostTemp[n][m] << "\n";
       //cout << "HTC: " << ConvectionHTC << "\n";
       //cout << "ThicknessFlux: " << ThicknessFlux << "\n";
       //cout << "DensityFlux: " << DiffusionFlux << "\n";
       //cout << "BulkHumidityRatio: " << BulkHumidityRatio << "\n";
       //cout << "OutletHumidityRatio: " << SegmentOutletHumidityRatio
            << "\n";
       cout << "FrostConductivity: " << FrostConductivity[n][m] <<
            "\n";
       cout << "TotalMass: " << TotalFrostMass[n][m] << "\n";
       cout << "InletTemp: " << InletAirTemp[n][m] << "\n";
       //cout << "OutletTemp: " << OutletAirTemp[n][m] << "\n";
       cout << "InletRH: " << InletAirRH[n][m] << "\n";
       //cout << "OutletRH: " << OutletAirRH[n][m] << "\n";
       cout << "BulkHumidityRatio: " << BulkHumidityRatio << "\n";
       cout << "OutletHumidityRatio: " << OutletBulkHumidityRatio <<
            "\n";
       cout << "InletAirVelocity: " << InletAirVelocity[m] << "\n";
       //cout << "SensibleFlux: " << SensibleFlux << "\n";
       //cout << "LatentFlux: " << LatentFlux << "\n";
       //cout << "TotalHeatFlux: " << TotalHeatFlux << "\n";
       cout << "Time: " << time << "\n";

    }

}

// Calculate pressure drop for each wettability segment
for (m = 0; m < 2; m++)
{
    TotalPressureDrop[m] = AirPressure0 - AirPressure[n][m];
}

// Adjust the air velocity through each wettability segment based on
the pressure drops in each segment
if (TotalPressureDrop[0] > TotalPressureDrop[1])
{
    InletAirVelocity[0] = InletAirVelocity[0] - 0.01;
```

```
         InletAirVelocity[1] = (AirMassFlowRate - BulkDensity*WSeg[0] *
               (ChannelHeight - NumPlates*FrostThickness[0][0]) *
               InletAirVelocity[0]) / (BulkDensity*WSeg[1] * (ChannelHeight
               - NumPlates*FrostThickness[0][1]));
     }
     else if (TotalPressureDrop[1] > TotalPressureDrop[0])
     {
         InletAirVelocity[1] = InletAirVelocity[1] - 0.01;
         InletAirVelocity[0] = (AirMassFlowRate - BulkDensity*WSeg[1] *
               (ChannelHeight - NumPlates*FrostThickness[0][1]) *
               InletAirVelocity[1]) / (BulkDensity*WSeg[0] * (ChannelHeight
               - NumPlates*FrostThickness[0][0]));
     }
     else
     {
         InletAirVelocity[0] = AirMassFlowRate / (BulkDensity*PlateWidth*
               (ChannelHeight - FrostThickness[0][0]));
         InletAirVelocity[1] = InletAirVelocity[0];
         cout << "pressure drop equal\n\n";
         break;
     }

// if pressure drop values are within the tolerance, exit the loop and move
on to the next time step
} while ((TotalPressureDrop[0] > (TotalPressureDrop[1] + 0.00001)) ||
      (TotalPressureDrop[0] < (TotalPressureDrop[1] - 0.00001)));

FrostMass = 0;
CumulativeFrostThickness = 0;
CumulativeFrostDensityNew = 0;

// Calculate the mass of the frost for the entire plate [kg]
for (m = 0; m < 2; m++)
{
     for (j = 0; j < NSegL; j++)
     {
         FrostMass = FrostMass + TotalFrostMass[j][m];
         CumulativeFrostThickness = CumulativeFrostThickness +
               FrostThickness[j][m];
         CumulativeFrostDensityNew = CumulativeFrostDensityNew +
               FrostDensityNew[j][m];
     }
}

AvgFrostThickness = CumulativeFrostThickness / (NSegL * 2);
AvgFrostDensityNew = CumulativeFrostDensityNew / (NSegL * 2);

cout << "\nTime: " << time << "\n";
cout << "FrostMass (one plate): " << FrostMass << "\n";
cout << "FrostMass (all plates): " << NumPlates*FrostMass << "\n";
```

```
            cout << "AverageFrostThickness: " << AvgFrostThickness << "\n";
            cout << "AverageFrostDensity: " << AvgFrostDensityNew << "\n\n";

    // criterion for stopping the simulation so it does not run infinitely
    } while (FrostThickness[n][m] < 0.005);


    return 0;
}
```

2) Properties

```
// This source file includes functions to calculate all fluid properties using CoolProp

#include "stdafx.h"
#include <iostream>
#include "CoolProp.h"
#include "HumidAirProp.h"
#include <stdlib.h>
#include <cmath>
#include <sstream>
#include <fstream>
#include <string>
#include <cstddef>

using namespace std;
using namespace CoolProp;
using namespace HumidAir;


double DewPointTemp(double InletAirTempK, double AirPressurePa, double InletAirRH);

// Function to calculate the condensate properties using CoolProp
void CondensateProperties(double &AirPressure, double &InterfaceTemp, double
&VaporVolume, double &SurfaceTension, double &DropletDensity, double
&DropletConductivity)
{
     double AirPressurePa;         // pressure of atmospheric air [Pa]
     double InterfaceTempK;        // interface temperature K]
     double VaporDensity;          // density of saturated water vapor at the interface
                                   temperature [kg/m^3]

     // Density Variables
     double Density;               // droplet density [g/mL]
     double a0, a1, a2, a3, a4, a5, a6;         // coefficients for density function

     // Surface Tension Variables
     double tau;
     double mu;
     double B;              // constants in the surface temperature equation
     double b;
     double T_c;
     double Temp;           // temperature of the liquid droplet [C]
     double Sigma;          // surface tension [mN/m]

     // Thermal Conductivity Variables
     double A_k;
     double gamma_k;
     double T_s;
double epsilon;        // variables and constants for use in calculating conductivity
double k_b;
double M;
double Mass;
double nu;
double temp;
double IsothermalCompressibility;          // isothermal compressibility

// Convert temperature and pressure to Pa and K, respectively
AirPressurePa = AirPressure * 1000;
InterfaceTempK = InterfaceTemp + 273.15;

// Calculate specific volume of saturated water vapor at the interface temperature
[m^3/kg]
VaporDensity = PropsSI("D", "T", 273.16, "Q", 1, "Water");
VaporVolume = 1 / VaporDensity;


// Calculate the density of the condensate at the interface temperature and air
pressure [kg/m^3]
// Correlation from Hare and Sorensen, 1987

a0 = 0.99986;
a1 = 0.00006690;
a2 = -0.000008486;
a3 = 0.0000001518;                  // initialization of density coefficients
a4 = -0.0000000069484;
a5 = -0.00000000036449;
a6 = -0.000000000007497;

// Calculate density [g/mL]
Density = a0 + a1 * InterfaceTemp + a2 * pow(InterfaceTemp, 2) + a3 *
     pow(InterfaceTemp, 3) + a4 * pow(InterfaceTemp, 4) + a5 * pow(InterfaceTemp, 5)
     + a6 * pow(InterfaceTemp, 6);

DropletDensity = 1000 * Density;       // convert density from [g/mL] to [kg/m^3]


// Calculate the surface tension of the condensate at 0 C (cannot be calculated for
lower temperatures) [N/m]
// IAPWS correlation as presented in Vins et al., 2015

if (InterfaceTemp > 0)
{
     Temp = InterfaceTemp;
}
else
{
     Temp = 0;
}
```

```
    T_c = 373.946;                                    // initialization of T_c [C]
    tau = (T_c - Temp) / (T_c + 273.15);              // calculate tau
    mu = 1.256;                                       // initialization of mu [-]
    B = 235.8;                                        // initialization of B [mN/m]
    b = -0.625;                                       // initialization of b [-]

    // Calculate surface tension [mN/m]
    Sigma = B*pow(tau, mu)*(1 + b*tau);        // surface tension [mN/m]

    SurfaceTension = Sigma / 1000;      // convert surface tension from [mN/m] to [N/m]


    // Calculate the thermal conductivity of the condensate at the interface temperature
    and air pressure [W/m-K]
    // Method from Biddle et al., 2013

    A_k = 0.00002965;                          // initialize A_k in [1/bar]
    gamma_k = 0.349;                           // initialize gamma_k [-]
    T_s = 228;                                 // initialize T_s [K]

    epsilon = InterfaceTempK / T_s - 1;

    // Calculate isothermal compressibility in [m^2/N], from Speedy and Angell, 1976
    IsothermalCompressibility = A_k*pow(epsilon, -gamma_k) / 100000;
    temp = pow(epsilon, -gamma_k);

    k_b = 1.38065e-23;                  // Boltzmann's constant [m^2-kg/s^2-K]
    M = 18.02;                          // mass of one water molecule [amu]
    Mass = M * (1.660539040e-27);              // convert mass from [amu] to [kg]
    nu = Mass / DropletDensity;                // define the variable nu

    temp = 1 / (DropletDensity * IsothermalCompressibility);        // set up a
                                                                    temporary variable

    // calculate thermal conductivity [W/m-K]
    DropletConductivity = 2.8 * k_b * pow(nu, -0.6666667) * pow(temp, 0.5);

}


// Function to calculate humid air properties using CoolProp
void HumidAirProperties(double &InletAirTemp, double &AirPressure, double &InletAirRH,
double &Density, double &Viscosity, double &Conductivity, double &SpecificHeat, double
&Prandtl, double &HumidityRatio, double &PartialPressure)
{

    double AirPressurePa;               // pressure of atmospheric air [Pa]
    double InletAirTempK;               // interface temperature [K]
    double Volume;                      // specific volume [m^3/kg]
    // Convert temperature and pressure to Pa and K, respectively
    AirPressurePa = AirPressure * 1000;
    InletAirTempK = InletAirTemp + 273.15;

    // Calculate the density of humid air [kg/m^3]
    Volume = HAPropsSI("Vha", "T", InletAirTempK, "P", AirPressurePa, "R", InletAirRH);
    Density = 1 / Volume;

    // Calculate the dynamic viscosity of humid air [N/m^2-s]
    Viscosity = HAPropsSI("mu", "T", InletAirTempK, "P", AirPressurePa, "R",
        InletAirRH);

    // Calculate the thermal conductivity of humid air [W/m-K]
    Conductivity = HAPropsSI("K", "T", InletAirTempK, "P", AirPressurePa, "R",
        InletAirRH);

    // Calculate the specific heat of humid air [J/kg-K]
    SpecificHeat = HAPropsSI("Cha", "T", InletAirTempK, "P", AirPressurePa, "R",
        InletAirRH);

    // Calculate the Prandtl number of the humid air [-]
    Prandtl = (SpecificHeat*Viscosity) / Conductivity;

    // Calculate the humidity ratio of humid air [kg/kg]
    HumidityRatio = HAPropsSI("Omega", "T", InletAirTempK, "P", AirPressurePa, "R",
        InletAirRH);

    // Calculate the partial pressure of the water vapor in the air [Pa]
    PartialPressure = HAPropsSI("P_w", "T", InletAirTempK, "P", AirPressurePa, "R",
        InletAirRH);

}


// Function to calculate properties of solid ice
void IceProperties(double &Temperature, double &IceDensity, double &IceConductivity)
{
    // convert the inlet temperature from [C] to [K]
    double TemperatureK;
    TemperatureK = Temperature + 273.15;

    // Ice density [kg/m^3]; it changes very little within our temperature range, so a
    constant value is acceptable
    IceDensity = 917;

    // Thermal conductivity [W/m-K]; equation from "Ice Physics" - Hobbs, 1974
    IceConductivity = 488.19 / TemperatureK + 0.4685;

}
```

```
// Function to calculate density of deposited ice crystals
double CrystalDensity(double CrystalTemp)
{
	// the equation used here is a curve fit taken from data presented in Fukuta, 1969
	and Ryan et al., 1976

	double rho;		// density of an ice crystal formed by deposition [g/cm^3]
	double CrystalDensity;	// density of an ice crystal formed by deposition [kg/m^3]

	// calculate crystal density using a curve fit
	rho = 0.96940936337 - 0.22034440158*CrystalTemp - 0.17528591346*pow(CrystalTemp, 2)
		- 0.037187835992*pow(CrystalTemp, 3) - 0.0034192623107*pow(CrystalTemp, 4) -
		0.00014272100058*pow(CrystalTemp, 5) - 0.0000022142177193*pow(CrystalTemp, 6);

	// convert density from [g/cm^3] to [kg/m^3]
	CrystalDensity = rho * 1000; // -100;

	return CrystalDensity;
}


// Function to calculate thermal conductivity of the frost layer using expressions from
various sources
double FrostThermalConductivity(double FrostDensity, double time, double ContactAngle,
double IceDensity, double AirDensity, double IceConductivity, double AirConductivity)
{
	double FrostConductivity; // average thermal conductivity of the frost layer [W/m-K]
	double CACoefficient;
	double Porosity;	// porosity of the frost layer
	double a;	// empirical parameter for use with Negrelli and Hermes correlation
	double b;	// empirical parameter for use with Negrelli and Hermes correlation

	// Van Dusen frost thermal conductivity as presented in Sankar's thesis
	//FrostConductivity = (0.0209 + 0.0000403 * FrostDensity + 0.00000000237 *
		pow(FrostDensity, 3));

	// Frost thermal conductivity equation developed and presented in Shin et al., 2003
	//CACoefficient = 1.09 - 0.00008 * pow((ContactAngle - 55), 2);
	//FrostConductivity = CACoefficient * (0.0448 + 0.00000004426 * pow(FrostDensity,
		3));

	// Frost thermal conductivity from O'Neal and Tree, 1985, as presented in Shin et
	al., 2003
	//FrostConductivity = 0.001202 * pow(FrostDensity, 0.963);

	// Frost thermal conductivity from Yonko and Sepsy, 1967, as presented in Shin et
	al., 2003
	//FrostConductivity = 0.024248 + 0.00072311 * FrostDensity + 0.000001183 *
		pow(FrostDensity, 2);
```

```
    // Frost thermal conductivity from Ostin and Andersson, 1991, as presented in Shin
    et al., 2003
    //FrostConductivity = -0.00871 + 0.000439 * FrostDensity + 0.00000105 *
        pow(FrostDensity, 2);

    // Frost thermal conductivity from Oscarsson et al., 1990, as presented in Shin et
    al., 2003
    //FrostConductivity = 0.202 * FrostDensity * (1 - FrostDensity / 1860) / (295 -
        0.189 * FrostDensity);

    // Frost thermal conductivity from Negrelli and Hermes, 2015
    //IceDensity = 917;
    Porosity = (IceDensity - FrostDensity) / (IceDensity - AirDensity);
    a = 1.576;
    b = 0.797;
    FrostConductivity = IceConductivity*a*pow((AirConductivity / IceConductivity),
        (b*Porosity));

    return FrostConductivity;
}


// Function to calculate frost layer density during the crystal growth stage
void CrystalPhaseFrostDensity(double DropletVolume, double Length, double Width, double
DropletRadius, double AspectRatio, double CrystalRadius, double FrostThickness, double
IceDensity, double CrystalDensity, double AirDensity, double NumDroplets, double
NumCylinders, double BaseRadius, double &TotalFrostMass, double &FrostDensity)
{
    //double NumDroplets;                   // number of droplets on the surface [-]
    //double DropletVolume;                 // volume of a single frozen droplet [m^3]
    double SurfaceArea;                     // total available surface area [m^2]
    double TotalDropletVolume;              // volume of all frozen droplets combined [m^3]
    double TotalCrystalVolume;              // volume of all frost crystals combined [m^3]
    double AirVolume;                // volume of the air present in the frost layer [m^3]
    double TotalVolume;                     // volume of the total frost layer [m^3]
    double ContactAngleRad;                 // contact angle [rad]
    double TotalDropletMass;                // mass of all frozen droplets [kg]
    double TotalCrystalMass;                // mass of all ice crystals [kg]

    // convert the contact angle from [deg] to [rad]
    //ContactAngleRad = ContactAngle * 3.1416 / 180;

    // calculate surface area [m^2]
    SurfaceArea = Length * Width;

    // calculate number of droplets on the surface
    //if (ContactAngle > 90)
    //{
    //    NumDroplets = AreaCoverage * SurfaceArea / (3.1416 * pow(DropletRadius, 2));
    //}
```

```
     //else
     //{
     //    NumDroplets = AreaCoverage * SurfaceArea / (3.1416 * pow(BaseRadius, 2));
     //}

     // calculate the volume of a single droplet [m^3]
     //DropletVolume = 3.1416 / 3 * pow(DropletRadius, 3) * pow((1 -
          cos(ContactAngleRad)), 2) * (2 + cos(ContactAngleRad));

     // calculate the total droplet volume [m^3]
     TotalDropletVolume = NumDroplets * DropletVolume;

     // calculate the droplet mass [kg]
     TotalDropletMass = TotalDropletVolume * IceDensity;

     // calculate the total crystal volume [m^3]
     TotalCrystalVolume = (NumCylinders * NumDroplets) * 3.1416 * AspectRatio *
          pow(CrystalRadius, 3);

     // calculate the crystal mass [kg]
     TotalCrystalMass = TotalCrystalVolume * CrystalDensity;

     // calculate the total volume of the space [m^3]
     TotalVolume = Length * Width * FrostThickness;

     // calculate the total frost mass [kg]
     TotalFrostMass = TotalDropletMass + TotalCrystalMass;

     // calculate the volume of the air in the frost layer [m^3]
     AirVolume = TotalVolume - TotalDropletVolume - TotalCrystalVolume;

     // Calculate the density of the frost layer by taking a weighted average of the
     densities of the different parts of the frost layer
     FrostDensity = (TotalDropletVolume / TotalVolume) * IceDensity + (TotalCrystalVolume
          / TotalVolume) * CrystalDensity + (AirVolume / TotalVolume)*AirDensity;
}


// Function to calculate the dew point temperature of the humid air
double DewPointTemp(double InletAirTemp, double AirPressure, double InletAirRH)
{
     double DewPoint;
     double DewPointK;
     double AirPressurePa;               // pressure of atmospheric air [Pa]
     double InletAirTempK;               // interface temperature [K]

     // Convert temperature and pressure to Pa and K, respectively
     AirPressurePa = AirPressure * 1000;
     InletAirTempK = InletAirTemp + 273.15;
```

```
    // Calculate the dew point temperature of humid air [C]
    DewPointK = HAPropsSI("D", "T", InletAirTempK, "P", AirPressurePa, "R", InletAirRH);

    DewPoint = DewPointK - 273.15;

    return DewPoint;
}


// Function to calculate the latent heat of sublimation
double SublimationLatentHeat(double Temperature)
{
    // written 1/7/21 by EH
    // correlation from Feistal and Wagner, 2007

    double SublimationHeat;             // specific latent heat of sublimation [J/kg]
    double TemperatureK;                // input temperatue in units of K
    double TriplePointTempK;            // temperature at the triple point of water [K]
    double a0;                          // constants for the correlation
    double a1;
    double a2;
    double a3;
    double a4;
    double a5;
    double a6;

    // convert the temperature from C to K
    TemperatureK = Temperature + 273.15;

    // define the triple point of water
    TriplePointTempK = 273.16;

    // set the values of the constants
    a0 = 2638742.45418107;
    a1 = 400983.673912406;
    a2 = 200812.111806393;
    a3 = -1486203.38485336;
    a4 = 2290451.50230789;
    a5 = -1690159.93521118;
    a6 = 479848.354373932;

    // Calculate the latent heat of sublimation
    SublimationHeat = a0 + a1*pow((TemperatureK / TriplePointTempK), 1) +
        a2*pow((TemperatureK / TriplePointTempK), 2) + a3*pow((TemperatureK /
        TriplePointTempK), 3) + a4*pow((TemperatureK / TriplePointTempK), 4) +
        a5*pow((TemperatureK / TriplePointTempK), 5) + a6*pow((TemperatureK /
        TriplePointTempK), 6);

    return SublimationHeat;
}
```

```
// Function to calculate the effective conductivity of the frost during the crystal
growth stage
double EffectiveFrostConductivity(double IceConductivity, double AirConductivity, double
NumDroplets, double NumCylinders, double DropletVolume, double AspectRatio, double
CrystalRadius, double Length, double Width, double FrostThickness, double
SublimationHeat, double AirDensity, double FrostHumidityRatio, double
SurfaceHumidityRatio, double FrostTemp, double SurfaceTemp, double CrystalVolumeRatio,
double CrystalHeight, double InitialFrostThickness)
{
     // written 1/8/21 by EH
     // calculate the effective thermal conductivity of the frost layer
     // during crystal growth using a volumetric average

     double EffectiveConductivitySeries;         // effective thermal conductivity using a
                                                 series resistance network [W/m-K]
     double EffectiveConductivityParallel;       // effective thermal conductivity using a
                                                 parallel resistance network [W/m-K]
     double alpha;             // averaging factor for effective thermal conductivity [-]
     double EffectiveConductivity;               // effective thermal conductivity [W/m-K]
     double EffectiveConductivitySahin;          // effective thermal conductivity with
                                                 added diffusivity term [W/m-K]
     double TotalDropletVolume;            // total volume of the frozen droplets [m^3]
     double TotalCrystalVolume;            // total volume of the crystals [m^3]
     double AirVolume;              // total volume of the air in the frosty layer [m^3]
     double TotalVolume;                   // total volume of the frost layer [m^3]
     double AvgFrostTempK;                 // average temperature in the frost layer [K]
     double MassDiffusivity;               // binary diffusion coefficient [m^2/s]

     // calculate the total droplet volume [m^3]
     TotalDropletVolume = NumDroplets * DropletVolume;
     //TotalDropletVolume = 0.3 * NumDroplets * DropletVolume;

     // calculate the total crystal volume [m^3]
     TotalCrystalVolume = (NumCylinders * NumDroplets) * 3.1416 * AspectRatio *
          pow(CrystalRadius, 3);

     // calculate the total volume of the space [m^3]
     TotalVolume = Length * Width * FrostThickness;

     // calculate the volume of the air in the frost layer [m^3]
     AirVolume = TotalVolume - TotalDropletVolume - TotalCrystalVolume;

     // Calculate the thermal conductivity of the frost layer by taking a weighted
     average of the thermal conductivities of the different parts of the frost layer.
     //EffectiveConductivity = (TotalDropletVolume / TotalVolume) * IceConductivity +
          (TotalCrystalVolume / TotalVolume) * IceConductivity + (AirVolume /
          TotalVolume)*AirConductivity;
     //EffectiveConductivity = CrystalVolumeRatio*IceConductivity + (1 -
          CrystalVolumeRatio) * AirConductivity;
```

```
    // Calculate effective thermal conductivity using both parallel and series methods
    EffectiveConductivityParallel = FrostThickness / ((CrystalHeight /
         (CrystalVolumeRatio*IceConductivity + (1 -
         CrystalVolumeRatio)*AirConductivity)) + (InitialFrostThickness /
         IceConductivity));
    EffectiveConductivitySeries = FrostThickness / (CrystalVolumeRatio*CrystalHeight /
         IceConductivity + (1 - CrystalVolumeRatio)*CrystalHeight / AirConductivity +
         InitialFrostThickness / IceConductivity);

    // Calculate effective thermal conductivity using a weighted average of the parallel
    and series methods
    //alpha = 0.7;
    alpha = 0.4274 - 0.0351*SurfaceTemp;
    EffectiveConductivity = alpha*EffectiveConductivityParallel + (1 - alpha) *
         EffectiveConductivitySeries;

    // calculate the binary diffusion coefficient in the frost layer
    AvgFrostTempK = (FrostTemp + SurfaceTemp) / 2 + 273.15;
    MassDiffusivity = 0.00000187 * pow(AvgFrostTempK, 2.072) / pow(100, 2);

    // Add in the diffusivity term as in Sahin, 2000
    EffectiveConductivitySahin = (TotalDropletVolume / TotalVolume) * IceConductivity +
         (TotalCrystalVolume / TotalVolume) * IceConductivity + (AirVolume /
         TotalVolume) * AirConductivity + (AirVolume /
         TotalVolume)*SublimationHeat*MassDiffusivity*AirDensity* (FrostHumidityRatio -
         SurfaceHumidityRatio) / (FrostTemp - SurfaceTemp);

    return EffectiveConductivity;
}
```

3) Other Functions

```
// Definitions of functions used in the frost model

#include "stdafx.h"
#include <iostream>
#include <cmath>

using namespace std;

// List of functions used in this source file
double InterfacialHeatTransferCoefficient(double InterfaceTemp, double H_fg, double
      VaporVolume, double alpha);
double FreeHTC(double SideLength, double PlateWidth, double FilmViscosity, double
      FilmDensity, double InletAirTemp, double SurfaceTemp, double FilmPrandtl, double
      FilmTemp, double InterfaceTemp, double beta, double FilmConductivity, double
      InletAirRH);
double ForcedHTC(double SideLength, double FilmDensity, double AirVelocity, double
      FilmViscosity, double FilmPrandtl);
double CrystalDensity(double CrystalTemp);
double MassTransferCo(double FilmDensity, double FilmSpecificHeat, double HTC, double
      LewisNo, double BulkDensity, double AirVelocity, double CharLength, double
      BulkViscosity, double InterfaceDensity, double FrostTemp);
double RoughnessHTC(double SideLength, double AirVelocity, double FilmDensity, double
      FilmViscosity, double FilmPrandtl, double FilmSpecificHeat, double FrostThickness);
double JakobNumber(double SpecificHeat, double H_fg, double Tdew, double Tsurface, double
      OmegaAir, double OmegaSat);


// Function to take primary inputs from the user
void GetInputs(double &InletAirTemp, double &AirPressure, double &InletAirRH, double
&InletAirVelocity, double &SurfaceTemp, double &ContactAngle1, double &ContactAngle2,
double &InitialRadius, double &Length, double &Width, double &ChannelWidth, double
&ChannelHeight, double &WidthRatio, double &NSegL, double &NumPlates)
{
     cout << "Enter the inlet air temperature in degrees Celsius: ";
     cin >> InletAirTemp;
     cout << "Enter the pressure of the humid air in kPa: ";
     cin >> AirPressure;
     cout << "Enter the inlet air relative humidity as a decimal: ";
     cin >> InletAirRH;
     cout << "Enter the inlet air velocity in m/s: ";
     cin >> InletAirVelocity;
     cout << "Enter the surface temperature in degrees Celsius: ";
     cin >> SurfaceTemp;
     cout << "Enter the primary contact angle in degrees: ";
     cin >> ContactAngle1;
     cout << "Enter the secondary contact angle in degrees: ";
     cin >> ContactAngle2;
```

```
    //cout << "Enter the initial radius of the condensate droplet in microns: ";
    //cin >> InitialRadius;
    cout << "Enter the plate length in m: ";
    cin >> Length;
    cout << "Enter the plate width in m: ";
    cin >> Width;
    cout << "Enter the channel width in m: ";
    cin >> ChannelWidth;
    cout << "Enter the channel height in m: ";
    cin >> ChannelHeight;
    cout << "Enter the fraction of the plate width that has the primary contact angle:
         ";
    cin >> WidthRatio;
    cout << "Enter the number of segments in the airflow direction: ";
    cin >> NSegL;
    cout << "Enter the number of plates (1 for a single plate, 2 for parallel plates):
         ";
    cin >> NumPlates;
    //cout << "Enter the time until condensate droplets freeze: ";
    //cin >> FreezingTime;
    cout << endl;
}


// Function to calculate the convection resistance
void ConvectionResistance(double InletAirTemp, double SurfaceTemp, double FilmTemp,
double AirVelocity, double SideLength, double PlateWidth, double FilmPrandtl, double
FilmDensity, double FilmViscosity, double FilmConductivity, double FilmSpecificHeat,
double BulkPartialPressure, double InterfacePartialPressure, double InterfaceTemp, double
ContactAngle, double Radius, double HeatTransfer, double H_fg, double Lewis, double
BulkVaporDensity, double InterfaceVaporDensity, double HTC, double MTC, double beta,
double BaseRadius, double AreaCoverage, double &ConvectionResistance)
{
    //double ConvectionResistance;                    // convection heat transfer
                                   resistance expressed as a temperature difference [K]
    double Nusselt;                // Nusselt number [-]
    double CharLength;             // characteristic length [m]
    //double HTC;                  // heat transfer coefficient [W/m^2-K]
    //double MTC;
    double Rvap;
    //double BulkVaporDensity;
    //double InterfaceVaporDensity;
    double ContactAngleRad;

    ContactAngleRad = ContactAngle * 3.1416 / 180;
     Rvap = 461.5;
    if (ContactAngle >= 90)
    {
        ConvectionResistance = HeatTransfer / (3.1416 * pow(Radius, 2) * HTC) - ((1 /
            AreaCoverage) * MTC * (BulkVaporDensity - InterfaceVaporDensity) * H_fg)
            / HTC;
    }
    else
    {
        ConvectionResistance = HeatTransfer / (3.1416 * pow(BaseRadius, 2) * HTC) - ((1
            / AreaCoverage) * MTC * (BulkVaporDensity - InterfaceVaporDensity) *
            H_fg) / HTC;
    }

}


// Function to calculate the Nusselt number for natural convection
double FreeHTC(double SideLength, double PlateWidth, double FilmViscosity, double
FilmDensity, double InletAirTemp, double SurfaceTemp, double FilmPrandtl, double
FilmTemp, double InterfaceTemp, double beta, double FilmConductivity, double InletAirRH)
{
    double Nusselt;                     // Nusselt number [-]
    double gravity = 9.81;              // acceleration due to gravity [m^2/s]
    //double beta;                      // variable used to calculate the Nusselt number
    double FilmTempK;                   // film temperature [K]
    double FilmKinematicViscosity;      // kinematic visosity [m^2/s]
    double Rayleigh;                    // Rayleigh number [-]
    double CharLength;                  // characteristic length [m]
    double HTC;
    double WidthFunc;
    double FunctionPrandtl;
    double NuTemp;

    FilmTempK = FilmTemp + 273.15;      // convert the film temperature from [C] to [K]
    beta = 1 / FilmTempK;                            // definition of beta
    // horizontal plate
    CharLength = (SideLength * PlateWidth) / (2 * SideLength + 2 * PlateWidth);
    //CharLength = SideLength;                 // vertical plate
    FilmKinematicViscosity = FilmViscosity / FilmDensity;
    Rayleigh = (gravity * beta * (InletAirTemp - InterfaceTemp) * pow(CharLength, 3)) *
         FilmPrandtl / pow(FilmKinematicViscosity, 2);
    //WidthFunc = 2.1*exp((-48)*PlateWidth) + 1.2;
    //Nusselt = 1 * (0.27*pow(Rayleigh, 0.25));              // horizontal plate
    //Nusselt = WidthFunc * pow(Rayleigh, 0.2);
    //Nusselt = 0.52*pow(Rayleigh, 0.2);
    //Nusselt = 0.59*pow(Rayleigh, 0.25);                    // vertical plate

    // Alamdari and Hammond, 1983
    // CharLength = 4 * (SideLength * PlateWidth) / (2 * SideLength + 2 * PlateWidth);
    // HTC = 0.60*pow(((InletAirTemp - InterfaceTemp) / pow(CharLength, 2)), 0.2);
    // Nusselt = HTC*CharLength / FilmConductivity;

    // Thomas et al., 2012
    //HTC = 110.96 + 203.11 * pow(InletAirRH, 2.17) * pow((InletAirTemp - SurfaceTemp),
        0.1);
    //Nusselt = HTC*CharLength / FilmConductivity;

    // USE THIS ONE FOR A HORIZONTAL PLATE
    // Rohsenow et al., 1998
    NuTemp = 0.527 * pow(Rayleigh, 0.2) / pow((1 + pow((1.9 / FilmPrandtl), 0.9)),
        0.2222222);
    Nusselt = 2.5 / log(1 + 2.5 / NuTemp);

    // VERTICAL PLATE
    //CharLength = PlateWidth;
    //Rayleigh = (gravity * beta * (InletAirTemp - InterfaceTemp) * pow(CharLength, 3))
        * FilmPrandtl / pow(FilmKinematicViscosity, 2);
    //Nusselt = 0.68 + 0.670*pow(Rayleigh, 0.25) / pow((1 + pow((0.492 / FilmPrandtl),
        (9 / 16))), (4 / 9));

    return Nusselt;
}


// Function to calculate the average Nusselt number for forced convection
double ForcedHTC(double SideLength, double FilmDensity, double AirVelocity, double
FilmViscosity, double FilmPrandtl)
{
    double Nusselt;                             // Nusselt number [-]
    double Reynolds;                            // Reynolds number [-]
    double CharLength;                          // characteristic length [m]

    CharLength = SideLength;
    Reynolds = (FilmDensity * AirVelocity * CharLength) / FilmViscosity;
    Nusselt = 0.664 * pow(Reynolds, 0.5) * pow(FilmPrandtl, 0.333333);
    //Nusselt = 0.034 * pow(Reynolds, 0.8);         // O'Neal and Tree, 1985
    //Nusselt = 0.394 * pow(Reynolds, 0.509);

    return Nusselt;
}


// Function to calculate the local Nusselt number for forced convection
double LocalForcedHTC(double SideLength, double FilmDensity, double AirVelocity, double
FilmViscosity, double FilmPrandtl)
{
    double Nusselt;                             // Nusselt number [-]
    double Reynolds;                            // Reynolds number [-]
    double CharLength;                          // characteristic length [m]
```

```
     CharLength = SideLength;
     Reynolds = (FilmDensity * AirVelocity * CharLength) / FilmViscosity;
     Nusselt = 0.332 * pow(Reynolds, 0.5) * pow(FilmPrandtl, 0.333333);

     return Nusselt;
}


// Function to calculate the Nusselt number for thermally developing internal turbulent
flow
double TurbulentNusselt(double HydraulicReynolds, double HydraulicDiameter, double
PlateLength, double Prandtl)
{
     // ENH - 11/16/2020
     // Calculates Nusselt number for thermally developing turbulent flow in a duct
     // Equation from Rohsenow et al. (1998) - "Handbook of Heat Transfer"

     double Nusselt;              // Nusselt number [-]
     double DevelopedNusselt;     // Nusselt number for fully developed turbulent flow
     double temp;                 // temporary variable

     // Use the Dittus-Boelter correlation to calculate the Nusselt number for fully
     developed flow
     DevelopedNusselt = 0.026*pow(HydraulicReynolds, 0.8)*pow(Prandtl, 0.3);

     // Calculate a temporary variable
     temp = pow((PlateLength / HydraulicDiameter), 0.1) / pow(Prandtl, 0.1667)*(0.68 +
          3000 / pow(HydraulicReynolds, 0.81));

     // Calculate the Nusselt number for thermally developing flow (Rohsenow, 1998)
     Nusselt = DevelopedNusselt*(1 + temp / (PlateLength / HydraulicDiameter));

     return Nusselt;
}


// Function to calculate the Nusselt number for laminar internal flow
double LaminarNusselt(double Distance, double HydraulicDiameter, double
HydraulicReynoldsNo, double Prandtl)
{
     // ENH - 2/5/2021
     // Calculates Nusselt number for thermally developing laminar flow in a duct
     // Equation from Rohsenow et al. (1998) - "Handbook of Heat Transfer"

     double Nusselt;
     double x;

     // dimensionless distance
     x = (Distance / HydraulicDiameter) / (HydraulicReynoldsNo*Prandtl);
     Nusselt = 3.66 + 0.0668 / (pow(x, 0.333)*(0.04 + pow(x, 0.667)));

     return Nusselt;
}


// Function to calculate the local Nusselt number between parallel plates
double LocalParallelPlateNusselt(double ChannelHeight, double xdistance, double
InletAirVelocity, double FilmDensity, double FilmViscosity, double FilmPrandtl, double
FrostThickness)
{
     // ENH - 2/13/2021
     // Calculates Nusselt number for thermally developing laminar flow between parallel
     plates
     // Equation from Rohsenow et al. (1998) - "Handbook of Heat Transfer"

     double HydraulicDiameter;
     double HydraulicReynolds;
     double xstar;
     double Nusselt;

     // Calculate the hydraulic diameter and Reynolds number
     HydraulicDiameter = 2 * (ChannelHeight - 2 * FrostThickness);
     HydraulicReynolds = (FilmDensity * InletAirVelocity * HydraulicDiameter) /
          FilmViscosity;

     // Calculate the dimensionless distance
     xstar = xdistance / (HydraulicReynolds*HydraulicDiameter*FilmPrandtl);

     //Calculate the local Nusselt number
     if (xstar < 0.001)
     {
          Nusselt = 1.233 / pow(xstar, 0.333) + 0.4;
     }
     else
     {
          Nusselt = 7.541 + 6.874*pow((1000 * xstar), -0.488)*exp(-245 * xstar);
     }

     return Nusselt;
}


// Calculate the Nusselt number according to the correlation presented by Bryant, 1995
double NusseltBryant(double Reynolds, double Grashof, double FilmPrandtl)
{
     // EH - 8/24/2020
     // Function to calculate the Nusselt number developed by Brant, 1995 for his data

     double NusseltB;              // Nusselt number correlated by Bryant's data
     NusseltB = 246 * pow(Reynolds, 0.16)*pow(Grashof, -0.24)*pow(FilmPrandtl, 0.33);

     return NusseltB;
}


// Function to calculate the conduction resistance through a droplet
double ConductionResistance(double HeatTransfer, double ContactAngle, double
Conductivity, double Radius)
{
    // expression taken from Kim and Kim, 2011

    double ConductionResistance;         // conduction heat transfer resistance expressed
                                         as a temperature difference [K]
    double ContactAngleRad;              // contact angle [rad]

    // convert contact angle from degrees to radians
    ContactAngleRad = ContactAngle * 3.1416 / 180;

    // calculate the conduction resistance [K]
    ConductionResistance = (HeatTransfer * ContactAngleRad) / (4 * 3.1416 * Radius *
         Conductivity * sin(ContactAngleRad));

    return ConductionResistance;
}


// Function to calculate the curvature resistance through a droplet
double CurvatureResistance(double InterfaceTemp, double SurfaceTension, double Density,
double H_fg, double Radius)
{
    // expression taken from Graham, 1969

    double CurvatureResistance;          // curvature heat transfer resistance expressed
                                         as a temperature difference [K]
    double InterfaceTempK;               // interface temperature [K]

    // convert interface temperature from [C] to [K]
    InterfaceTempK = InterfaceTemp + 273.15;

    // calculate the curvature resistance [K]
    CurvatureResistance = (2 * InterfaceTempK * SurfaceTension) / (Density * H_fg *
         Radius);

    return CurvatureResistance;
}
```

```
// Function to calculate the droplet growth coefficient
double DropletGrowthRate(double InletAirTemp, double InterfaceTemp, double SurfaceTemp,
double SurfaceTension, double Density, double H_fg, double Radius, double Conductivity,
double VaporVolume, double ContactAngle, double alpha, double MTC, double HTC, double
BulkVaporDensity, double InterfaceVaporDensity, double AreaCoverage)
{
     double GrowthRate;            // droplet radial growth rate [m/s]
     double InterfaceTempK;        // interfacial temperature in K
     double ContactAngleRad;       // contact angle in radians
     double InterfacialHTC;        // interfacial heat transfer coefficient [W/m^2-K]
     double S;

     InterfaceTempK = InterfaceTemp + 273.15;          // convert interface temperature
                                                       from [C] to [K]
     ContactAngleRad = ContactAngle * 3.1416 / 180;          // convert contact angle
                                                              from [deg] to [rad]

     GrowthRate = MTC*(BulkVaporDensity - InterfaceVaporDensity) / (2 * AreaCoverage *
          Density);

     return GrowthRate;
}


// Function to calculate the droplet freezing time
double DropletFreezingTime(double InletAirTemp, double SurfaceTemp, double
SuperSatDegree, double InletAirVelocity, double ContactAngle)
{
     // EH - 6/19/2020
     // function to calculate the freezing time using the in-house built regression
     correlation

     double FreezingTime;          // time that it takes droplets on a surface to freeze
     double DeltaT;                // temperature difference between air and surface
     double ContactAngleRad;              // contact angle in units of radians

     double a;                     // fitting parameter
     double b;                     // fitting parameter
     double c;                     // fitting parameter
     double d;                     // fitting parameter

     DeltaT = InletAirTemp - SurfaceTemp;
     ContactAngleRad = ContactAngle * 3.1416 / 180;

     // Optimization for whole data set
     //a = 446.3;
     //b = -0.7723;
     //c = 4.346;
     //d = 298.5;
```

```
	// Optimization for our data
	//a = 324.9;
	//b = -1.490;
	//c = 2.825;
	//d = 13.38;

	// Optimization for our data (2)
	//a = 11324;
	//b = -0.6653;
	//c = -0.6117;
	//d = -1.587;

	// Optimization for data in our range from us, Bryant, and Kim 2016
	a = 15107;
	b = -0.6026;
	c = -1.105;
	d = -1.224;

	//FreezingTime = a*pow(SuperSatDegree, b)*pow(ContactAngleRad, (pow(ContactAngleRad,
		c) / d));
	//FreezingTime = a*pow(SuperSatDegree, b)*pow((ContactAngleRad+1),
		(pow((ContactAngleRad+1), c) / d));
	//FreezingTime = a*pow(SuperSatDegree, b)*pow(DeltaT, c)*pow((InletAirVelocity + 1),
		d);
	FreezingTime = a*pow(SuperSatDegree, b)*pow((InletAirVelocity + 1), c)*pow((0 -
		SurfaceTemp), d);

	  return FreezingTime;
}


// Function to calculate the freezing time from the correlation presented in Harges and
Cremaschi, 2018b
double ASTFEDropletFreezingTime(double InletAirRH, double SurfaceTemp, double
InletAirTemp, double ContactAngle, double InletAirVelocity, double Nusselt)
{
	double ASTFEFreezingTime;
	double SurfaceTempK;
	double InletAirTempK;
	double ContactAngleRad;
	double a;
	double b;
	double c;
	double d;
	double e;
	double f;
	double g;

	SurfaceTempK = SurfaceTemp + 273.15;
	InletAirTempK = InletAirTemp + 273.15;
```

```
    ContactAngleRad = ContactAngle*3.1416 / 180;

    // exponent values
    a = 0.01476;
    b = -1.0014;
    c = 21.406;
    d = -18.616;
    e = 0.69053;
    f = 1.2681;
    g = -1.7619;

    ASTFEFreezingTime = a*pow(InletAirRH, b)*pow(SurfaceTempK, c)*pow(InletAirTempK, d)
        * pow(ContactAngleRad, e)*pow((InletAirVelocity + 1), f)*pow(Nusselt, g);

    return ASTFEFreezingTime;
}


// Function to calculate droplet freezing time according to Bryant, 1995
double NucleationTime(double ContactAngle, double NusseltBryant, double
BulkHumidityRatio, double SurfaceHumidityRatio, double InletAirTemp)
{
    // EH - 8/24/2020
    // Function to calculate freezing time using correlation from Bryant, 1995

    double NucTime;                 // time until liquid nucleates to ice
    double A;                       // term used for contact angle dependence
    double W;                       // term used for humidity dependence
    double InletAirTempK;           // inlet air temperature in K

    A = ContactAngle / 90;
    W = BulkHumidityRatio - SurfaceHumidityRatio;
    InletAirTempK = InletAirTemp + 273.15;

    //NucTime = 41.3*pow(A, 2.6)*pow(NusseltBryant, -1.4)*pow(W, -0.6) *
    pow(InletAirTempK, -0.25);
    NucTime = 41.3*pow(A, 2.6)*pow(NusseltBryant, -0.3)*pow(W, -0.6) * pow(InletAirTemp,
        -0.25);

    return NucTime;
}


// Function to calculate exponent for power-law formulation
double DropletCoalescence(double InletAirTemp, double SurfaceTemp, double ContactAngle)
{
    // EH - 6/20/2020
    // function to calculate the exponent/power governing droplet coalescence according
    to the power law formulation
    double mu;                    // exponent governing droplet coalescence

    //mu = 1.66 - 0.0025*ContactAngle - 0.011*(InletAirTemp - SurfaceTemp);
    //mu = 1.77 - 0.004*ContactAngle - 0.011*(InletAirTemp - SurfaceTemp);
    //mu = 1.53 - 0.00098*ContactAngle - 0.0105*(InletAirTemp - SurfaceTemp);
    //mu = 1.59 - 0.0017*ContactAngle - 0.0105*(InletAirTemp - SurfaceTemp);
    mu = 1.645 - 0.0017*ContactAngle - 0.0168*(InletAirTemp - SurfaceTemp);

    return mu;
}


// Function to calculate droplet diameter according to Hoke et al., 2000
double HokeDropletDiameter(double ContactAngle, double MTC, double BulkVaporDensity,
double SurfaceVaporDensity, double DropletDensity, double NumDroplets, double Time,
double Length, double Width)
{
    double ContactAngleRad;                    // Contact angle in radians
    double DropletDiameter;                    // Volume averaged droplet diameter
    double ThetaPhobic;                        // Equal to ContactAngle - 90 degrees
    double ThetaPhilic;                        // Equal to 90 degrees - ContactAngle

    ContactAngleRad = ContactAngle * 3.1416 / 180;
    ThetaPhobic = ContactAngleRad - 3.1416 / 2;
    ThetaPhilic = 3.1416 / 2 - ContactAngleRad;

    if (ContactAngle < 90)

        DropletDiameter = pow((3 * MTC*(BulkVaporDensity - SurfaceVaporDensity)*Time /
            (DropletDensity*NumDroplets /(Length*Width)*3.1416) * 8 /
            pow((1 - sin(ThetaPhilic)), 2) / (2 + sin(ThetaPhilic))), (0.3333));

    else

        DropletDiameter = pow((3 * MTC*(BulkVaporDensity - SurfaceVaporDensity)*Time /
            (DropletDensity*NumDroplets/ (Length*Width)*3.1416) * 8 /
            pow((1 + sin(ThetaPhobic)), 2) / (2 - sin(ThetaPhobic))), (0.3333));

    return DropletDiameter;
}


// Function to calculate initial droplet radius
double InitialDropletRadius(double SurfaceTension, double DropletDensity, double
SurfaceTemp, double SurfaceHumidityRatio, double BulkHumidityRatio)
{
    // EH - 6/20/2020
    // This function calculates the critical droplet radius for the onset of
    condensation
    // Reference: Piucco, R. O., Hermes, C. J., Melo, C., Barbosa, J. R., 2008.
    // “A study of frost nucleation on flat surfaces: Theoretical model and experimental
    validation.” International Refrigeration and Air Conditioning Conference, Paper 878.

    double GasConstant;                 // Gas constant of water vapor
    double SurfaceTempK;                // Surface temperature in units of K
    double CriticalRadius;              // Critical droplet radius for condensation

    // Initialize the gas constant
    GasConstant = 461.5;                // units of J/kg-K

    // Convert SurfaceTemp from C to K
    SurfaceTempK = SurfaceTemp + 273.15;

    // Calculate the critical radius in units of m
    CriticalRadius = 2 * SurfaceTension / (DropletDensity*GasConstant*SurfaceTempK*
         log(BulkHumidityRatio / SurfaceHumidityRatio));

    return CriticalRadius;
}


// Function to calculate the surface area coverage of droplets
double SurfaceAreaCoverage(double ProjectedRadius, double ContactAngle)
{
    // EH - 6/20/2020

    double ContactAngleRad;             // Contact angle in units of radians
    double AreaCoverage;                // Surface area coverage fraction

    ContactAngleRad = ContactAngle*3.14159265 / 180;

    // Calculate the surface area coverage fraction using an empirical correlation
    //AreaCoverage = (-0.2817 * ContactAngle + 79.067) / 100;

    // This correlation behaves incorrectly at very small values of droplet radius (< 60
    microns)
    //AreaCoverage = (1 - exp(-21.8*ProjectedRadius*1000)) * 0.841 *
         pow((ProjectedRadius*1000), 0.151) + exp(-21.8*(ProjectedRadius*1000)) *
         15.8*pow(ContactAngle, -0.722);

    // This correlation has a little less accuracy, but behaves correctly from the
    beginning
    AreaCoverage = 0.0386*(21.91 - ContactAngleRad)*pow(ProjectedRadius*1000, 0.1043);

    return AreaCoverage;
}
```

```
// Function to calculate all droplet dimensions using radius of curvature and contact
angle
void DropletDimensions(double Radius, double ContactAngle, double &DropletHeight, double
&BaseRadius, double &DropletVolume)
{
	double ContactAngleRad;

	ContactAngleRad = ContactAngle * 3.1416 / 180;

	DropletHeight = Radius * (1 - cos(ContactAngleRad));
	BaseRadius = pow((DropletHeight * (2 * Radius - DropletHeight)), 0.5);
	DropletVolume = 3.1416 / 3 * pow(DropletHeight, 2) * (3 * Radius - DropletHeight);
}


// Function to calculate dimensions of the frozen droplets
void FrozenDropletDimensions(double DropletDensity, double IceDensity, double
ContactAngle, double &DropletHeight, double &Radius, double &DropletVolume, double
&BaseRadius)
{
	// This function calculates the dimensions of droplets after freezing and is based
	on mass conservation

	double ContactAngleRad;                    // contact angle [rad]
	double DropletMass;                 // mass of droplet [kg]
	double temp;                  // temporary variable for use with the "pow" function

	// convert contact angle from [deg] to [rad]
	ContactAngleRad = ContactAngle*3.1416 / 180;

	// calculations for droplet height [m] and volume [m^3] using supercooled water
	properties
	DropletHeight = Radius*(1 - cos(ContactAngleRad));
	DropletVolume = (3.1416*pow(DropletHeight, 2))*(3 * Radius - DropletHeight) / 3;

	// calculation of droplet mass [kg], which remains constant
	DropletMass = DropletDensity*DropletVolume;

	// calculations for droplet dimensions using ice properties
	DropletVolume = DropletMass / IceDensity;
	temp = 3 * DropletVolume / (3.1416*(3 / (1 - cos(ContactAngleRad)) - 1));
	DropletHeight = pow(temp, 0.333333);
	Radius = DropletHeight / (1 - cos(ContactAngleRad));

	// calculate the radius of the base of the droplet (the part which actually contacts
	the surface) [m]
	temp = DropletHeight*(2 * Radius - DropletHeight);
	BaseRadius = pow(temp, 0.5);
}
```

```
// Function to calculate the Lewis number
double LewisNumber(double FilmTemp, double FilmConductivity, double FilmDensity, double
FilmSpecificHeat)
{
     double MassDiffusivity;              // diffusion coefficient of water in air [m^2/s]
     double FilmTempK;                    // film temperature in degrees Kelvin
     double Lewis;                        // dimensionless Lewis number

     FilmTempK = FilmTemp + 273.15;

     // calculate diffusion coefficient
     MassDiffusivity = 0.00000187 * pow(FilmTempK, 2.072) / pow(100, 2);

     Lewis = FilmConductivity / (FilmDensity * FilmSpecificHeat * MassDiffusivity);
     //Lewis = 1;

     return Lewis;
}


// Function to calculate the heat transfer rate through a cylindrical crystal
double CrystalHeatTransfer(double CrystalRadius, double HTC, double MTC, double AirTemp,
double CrystalTemp, double ProjectedRadius, double SurfaceAreaCoverage, double
NumCylinders, double SublimationHeat, double BulkVaporDensity, double
CrystalVaporDensity)
{
     double HeatTransfer;         // heat transfer through a cylindrical crystal [W]

     HeatTransfer = 3.1416*pow(CrystalRadius, 2)*HTC*(AirTemp - CrystalTemp) +
          3.1416*pow(ProjectedRadius, 2)*MTC*(BulkVaporDensity - CrystalVaporDensity)
          *SublimationHeat / (SurfaceAreaCoverage*NumCylinders);

     return HeatTransfer;
}


// Function to calculate the conduction resistance through a cylindrical crystal
double CrystalConductionResistance(double CrystalHeatTransfer, double CrystalHeight,
double CrystalRadius, double CrystalConductivity)
{
     // written Oct. 24, 2020, by ENH

     double DeltaTCond;

     DeltaTCond = CrystalHeatTransfer*CrystalHeight / (3.1416*pow(CrystalRadius,
          2)*CrystalConductivity);

     return DeltaTCond;
}
```

```
// Function to calculate the radial growth rate of a cylindrical crystal
double CrystalRadialGrowthRate(double MTC, double DropletAreaCoverage, double
CrystalAreaCoverage, double BulkVaporDensity, double CrystalVaporDensity, double
AspectRatio, double CrystalDensity, double ProjectedRadius, double CrystalRadius, double
NumCylinders)
{
     double RadialGrowthRate;

     //RadialGrowthRate = (MTC / (DropletAreaCoverage*CrystalAreaCoverage)) *
           (BulkVaporDensity - CrystalVaporDensity) / (2 * AspectRatio*CrystalDensity);

     RadialGrowthRate = (pow(ProjectedRadius,2) * MTC /
           (DropletAreaCoverage*NumCylinders))*(BulkVaporDensity - CrystalVaporDensity) /
           (2 * AspectRatio*CrystalDensity*pow(CrystalRadius,2));

     return RadialGrowthRate;
}


// Function to calculate the mass transfer coefficient
double MassTransferCo(double FilmDensity, double FilmSpecificHeat, double HTC, double
Lewis, double BulkDensity, double AirVelocity, double CharLength, double BulkViscosity,
double InterfaceDensity, double FrostTemp)
{
     double Diffusivity;
     double Reynolds;
     double Schmidt;
     double Sherwood;
     double FrostTempK;
     double MTC;                         // mass transfer coefficient [m/s]

     // initialize the Lewis number
     //Lewis = 1;                        // this is often a good approximation in real-
                                         life situations, so I use it for now
     FrostTempK = FrostTemp + 273.15;

     // calculate the mass transfer coefficient [kg/m^2-s]
     MTC = HTC / (FilmDensity * FilmSpecificHeat * pow(Lewis, 0.6666667));

     // Calculate the mass transfer coefficient using the Sherwood number
     Diffusivity = 0.00000187 * pow(FrostTempK, 2.072) / pow(100, 2);
     Reynolds = BulkDensity * AirVelocity * CharLength / BulkViscosity;
     Schmidt = BulkViscosity / (InterfaceDensity * Diffusivity);
     // The following correlation is from Fontaine et al. (?)
     Sherwood = 0.225 * pow(Reynolds, 0.6667) * pow(Schmidt, 0.33333);
     //MTC = Sherwood * Diffusivity / CharLength;

     return MTC;
}
```

```
// Function to calculate the mass flux which goes toward densification of the frost layer
void DensityMassFlux(double FrostDensity, double FrostTemp, double IceDensity, double
AirPressure, double AirTemp, double AirHR, double SublimationHeat, double SurfaceTemp,
double FrostThickness, double PartialPressureSat, double &FrostHR, double
&DensityMassFlux)
{
	double MassDiffusivity;			// mass diffusive coefficient [m^2/s]
	double Rvap;				// specific gas constant of water vapor [J/kg-K]
	double FrostTempK;			// frost surface temperature [K]
	double AirPressurePa;			// atmospheric pressure in [Pa]

	// convert the atmospheric air pressure from [kPa] to [Pa]
	AirPressurePa = AirPressure * 1000;

	// convert the frost surface temperature from [C] to [K]
	FrostTempK = FrostTemp + 273.15;

	// initialize the specific gas constant of water vapor [J/kg-K]
	Rvap = 461.5;

	// calculate the diffusion coefficient [m^2/s]
	// Marrero and Mason, 1972 - "Gaseous Diffusion Coefficients"
	MassDiffusivity = 0.00000187 * pow(FrostTempK, 2.072) / pow(100, 2);   // standard
		diffusion coefficient for air-water mixtures [m^2/s]

	// calculate the partial pressure of the water vapor at the frost surface [Pa]
	// Assume saturated at the frost surface
	// PartialPressureSat = exp(77.3450 + 0.0057 * FrostTempK - 7235 / FrostTempK) /
		pow(FrostTempK, 8.2);

	// calculate the humidity ratio at the frost surface
	// FrostHR = 0.62198 * (PartialPressureSat / (AirPressurePa - PartialPressureSat));

	// calculate the diffusion mass flux which goes to increase density of the frost
	layer [kg/m^2-s]
	DensityMassFlux = MassDiffusivity * ((1 - (FrostDensity / IceDensity)) / (1 +
		pow((FrostDensity / IceDensity), 0.5))) * PartialPressureSat / (Rvap *
		pow(FrostTempK, 2)) * (SublimationHeat / (Rvap * FrostTempK) - 1) * (FrostTemp
		- SurfaceTemp) / FrostThickness;

}


// Function to calculate the temperature of the frost layer surface
double FrostLayerSurfaceTemp(double DiffusionFlux, double FrostTemp, double SurfaceTemp,
double FrostThickness, double FrostConductivity,double SurfaceVaporDensity, double
AirTemp, double BulkVaporDensity, double SublimationHeat, double HTC, double MTC)
{
	double M;   // variable used for simplification of a complex expression [kg/m-s-K]
	double FrostTempNew;		// new frost layer surface temperature [C]
```

```
    // Calculate M
    M = DiffusionFlux * FrostThickness / (FrostTemp - SurfaceTemp);

    FrostTempNew = (SurfaceTemp / FrostThickness * (FrostConductivity + M *
         SublimationHeat) + HTC * AirTemp + MTC * (BulkVaporDensity-SurfaceVaporDensity)
         * SublimationHeat) / (FrostConductivity / FrostThickness + M * SublimationHeat
         / FrostThickness + HTC);

    return FrostTempNew;
}


// Function to calculate the supersaturation degree
void Supersaturation(double BulkPartialPressure, double SatBulkPartialPressure, double
SatSurfacePartialPressure, double &SuperSatDegree, double &SurfacePartialPressure)
{
    // calculate the supersaturation degree and surface partial pressure using equations
    from Na and Webb (2004)
    //SuperSatDegree = 0.808 * (BulkPartialPressure / SatBulkPartialPressure) *
         pow((SatSurfacePartialPressure / SatBulkPartialPressure), -0.657) - 1;
    SuperSatDegree = (BulkPartialPressure - SatSurfacePartialPressure) /
         SatSurfacePartialPressure;
    //SurfacePartialPressure = SuperSatDegree * SatSurfacePartialPressure +
         SatSurfacePartialPressure;
}


// Function to calculate the Jakob number
double JakobNumber(double SpecificHeat, double H_fg, double Tdew, double Tsurface, double
OmegaAir, double OmegaSat)
{
    double Jakob;

    // calculate the modified Jakob number
    Jakob = SpecificHeat / H_fg*(Tdew - Tsurface) / (OmegaAir - OmegaSat);
    return Jakob;
}


// Function to calculate the pressure drop for laminar flow through a channel
double LaminarPressureDrop(double FrictionFactor, double PlateLength, double
HydraulicDiameter, double AirDensity, double AirVelocity)
{
    double PressureDrop;          // Pressure drop through a length of duct [Pa]

    PressureDrop = FrictionFactor*(PlateLength / HydraulicDiameter)*(AirDensity /
         2)*pow(AirVelocity, 2);
```

```
     return PressureDrop;
}



// ****** OLD FREEZING TIME CALCULATIONS ******

// Correlation for total data
// FreezingTime = 0.0006863 * pow(InletAirRH, -0.961812) * pow((SurfaceTemp + 273.15),
   23.887166) * pow((InletAirTemp + 273.15), -20.99232) * pow((ContactAngle * 3.1416 /
   180), 0.1242667) * pow((InletAirVelocity + 1), 0.2800613) * pow(Nusselt, -0.659322);

//if (InletAirVelocity == 0)
//{
//      FreezingTime = 0.00003195 * pow(InletAirRH, -0.76344) * pow((SurfaceTemp +
        273.15), 17.659982) * pow((InletAirTemp + 273.15), -14.12583) * pow((ContactAngle
        * 3.1416 / 180), -0.233429) * pow((InletAirVelocity + 1), 1) * pow(Nusselt, -
        1.553953);
//}
//else
//{
//      FreezingTime = 0.0147642 * pow(InletAirRH, -1.001356) * pow((SurfaceTemp +
        273.15), 21.406133) * pow((InletAirTemp + 273.15), -18.61644) * pow((ContactAngle
        * 3.1416 / 180), 0.690532) * pow((InletAirVelocity + 1), 1.2680848) * pow(Nusselt,
        -1.761863);

//      FreezingTime = 0.00003295 * pow(InletAirRH, -1.547818) * pow((SurfaceTemp +
        273.15), 14.218657) * pow((InletAirTemp + 273.15), -11.81691) * pow((ContactAngle
        * 3.1416 / 180), 0.1677869) * pow((InletAirVelocity + 1), -2.358216) *
        pow(Nusselt, 1.4025199);

//      FreezingTime = 0.000000242 * pow(Jakob, 2) + 10.35513 * pow(SurfaceTemp, 2) +
        371.804 * SurfaceTemp + 0.28703 * (ContactAngle*3.1416 / 180) + 3327.787;

//      if (SurfaceTemp < -10)
//              FreezingTime = 16.4584 * pow(Jakob, 2) + 0.89399 * pow(SurfaceTemp, 2) +
                50.1545 * SurfaceTemp + 0.23141 * (ContactAngle*3.1416 / 180) + 675.12;
//      else
//              FreezingTime = 0.000028 * pow(Jakob, 2) + 83.5179 * pow(SurfaceTemp, 2) +
                1478.45 * SurfaceTemp + 0.00021*pow((ContactAngle*3.1416 / 180), 2) +
                0.28902 * (ContactAngle*3.1416 / 180) + 6962.08;
//}



// ****** OLD COALESCENCE CALCULATIONS ******

// Calculate the number of coalescences
//Coalescences = round(2.3324*log(FreezingTime) - 12.194);
//
// Calculate parameters for coalescence times
```

```
//alphaT = round((FreezingTime / 4) / 2) * 2;
//betaT = round(((FreezingTime - alphaT) / Coalescences) / 2) * 2;
//
// default condition
// C = 1;
//
// Calculate new droplet size
//for (i = 0; i < 20; i = i + 1)
//{
//      if (time == alphaT + i*betaT)
//             C = 1.33;
//      if (C == 1.33)
//             break;
//}
//
//Radius = C*Radius + GrowthRate * TimeStep;                  // use calculated growth
rate to find new droplet radius
//
//if (ContactAngle < 90)
//      C = 16.064*pow(GrowthRate * 1000000, 3) - 3.1093*pow(GrowthRate * 1000000, 2) +
       0.1561*GrowthRate * 1000000 - 0.0001;
//      C = 0.2252*pow((GrowthRate * 1000000), 2.54);
//else
//      C = 9.2781*pow(GrowthRate * 1000000, 3) + 1.067*pow(GrowthRate * 1000000, 2) -
       0.0169*GrowthRate * 1000000 - 0.00002;
//      C = 0.0938*GrowthRate * 1000000 - 0.0014;
//      C = (ContactAngle/200)*pow((GrowthRate * 1000000), 2.54);
//      C = (0.000000005*pow(ContactAngle, 4.0308))*pow((GrowthRate * 1000000), 2.54);
//      C = (0.00003*pow(ContactAngle, 2) + 0.0023*ContactAngle - 0.0708)*pow((GrowthRate
       * 1000000), 2.6);
//      C = (0.000000003*pow(ContactAngle, 4.2784))*pow((GrowthRate * 1000000), 2.6);
//
//if (ContactAngle < 90)
//      ProjectedGrowthRate = GrowthRate*pow(((1 - cos(ContactAngle*3.1416 / 180))*(1 +
       cos(ContactAngle*3.1416 / 180))), 0.5);
//else
//      ProjectedGrowthRate = GrowthRate;
//
//      C = (0.02061*pow(ContactAngle, 2) - 1.88169*ContactAngle +
       35.2)*pow((ProjectedGrowthRate * 1000000), 3.63646);
//
// Calculate the transition time between droplet growth modes
//time_trans = pow((GrowthRate * 1000000 / C), 1.8);
//
//if (ContactAngle < 90)
//      radius_trans = BaseRadius;
//else
//      radius_trans = Radius;
//
//if (radius_trans < 0.000080)
```

```
//     Radius = Radius + GrowthRate * TimeStep;
//else
//     Radius = Radius + (C * (pow(time, 1.677)-pow((time-TimeStep),1.677)))/1000000;
//
//if (ProjectedRadius < 0.000080)
//     ProjectedRadius = ProjectedRadius + ProjectedGrowthRate * TimeStep;
//else
//     ProjectedRadius = ProjectedRadius + (C * (pow(time, 1.677) - pow((time -
       TimeStep), 1.677))) / 1000000;
//
//if (ContactAngle < 90)
//     Radius = ProjectedRadius / pow(((1 - cos(ContactAngle*3.1416 / 180))*(1 +
       cos(ContactAngle*3.1416 / 180))), 0.5);
//else
//     Radius = ProjectedRadius;
//
//if ((time == 786))
//     Radius = 1.33*Radius;
```