THREE-DIMENSIONAL MODELING OF HEAT TRANSFER AND FLUID FLOW IN A FLAT-GROOVED HEAT PIPE

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

MECHANICAL ENGINEERING

By

Cem Kurt

September 2019

Three-Dimensional Modeling of Heat Transfer and Fluid Flow in a Flat-Grooved Heat Pipe
By Cem Kurt
September 2019

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Barbaros Çetin(Advisor)

Zafer Dursunkaya

Murat Köksal

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

THREE-DIMENSIONAL MODELING OF HEAT TRANSFER AND FLUID FLOW IN A FLAT-GROOVED HEAT PIPE

Cem Kurt
M.S. in Mechanical Engineering
Advisor: Barbaros Çetin
September 2019

Flat-grooved heat pipes (FGHP) are widely used in many applications from thermal management of electronic devices to space industry due to their robustness and ability of dissipating heat from the system effectively and reliably. FGHP is basically a container with micro grooves on the inner surfaces, and essentially a bridge that can transfer large amount of thermal energy between a heat source and a sink with small temperature differences by utilizing the phase change mechanism of the working fluid. Heat source evaporates the working fluid in the one end of the grooves, and due to the pressure difference, the composed vapor flows to the heat sink region in the other end. Then the vapor condenses back into the grooves before it flows to the evaporation region by the capillary force and repeat the cycle. Mathematical modeling of heat transfer and fluid flow of FGHP's is crucial to understand the effects of many parameters (dimensions, groove shape, working fluid filling ratio, material types) on their operational limits in order to design case-specific heat pipes. In the literature, many models are presented with some simplifications and assumptions. In this thesis, a computational methodology is proposed that models the heat transfer and fluid flow fully in 3D for the first time, by using COMSOL Multiphysics® *via* LiveLink™ for MATLAB® interface. Combining the flexibility of script environment of MATLAB with the benefits of using energy and momentum solvers of a commercial software gives a powerful and practical tool that can overcome great difficulties if this modeling was to be done in a CFD software or an in-house code alone. In the presented model, radius of curvature (R) variation of the working fluid in the groove, temperature gradient of the groove wall (T_w), and vapor temperature (T_v) are the essential working parameters of a heat pipe that reflects the efficiency. In this methodology, these variables are estimated initially, and are calculated by a set

of inter-bedded and subsequent iterations. The momentum equations are solved for the iteration of R, T_w is iterated by solving the energy equations, and lastly T_v is calculated by the secant method using the conservation of mass. Depending on the values of the variables, the solution domain is regenerated and the phase change boundary conditions are recalculated at each iteration. The presented model is compared with the literature for validation. Then, a parametric study for investigating the effect of groove depth on the performance of a flat-grooved heat pipe is conducted. Different power of heat sources are used for determining the dry-out in the grooves.

Keywords: Flat-Grooved Heat Pipe, Multidimensional Heat Transfer and Fluid Flow, Modeling.

ÖZET

DÜZ OLUKLU ISI BORULARINDA AKIŞIN VE ISI TRANSFERİNİN ÜÇ-BOYUTLU MODELLENMESİ

Cem Kurt
Makine Mühendisliği, Yüksek Lisans
Tez Danışmanı: Barbaros Çetin
Eylül 2019

Düz oluklu ısı boruları (DOIB), dayanıklı oluşlarından ve ısı taşıma konusunda yüksek etkinliğe ve güvenirliğe sahip olmalarından dolayı elektronik komponentlerin soğutulmasından uzay endüstrisine kadar geniş bir uygulama alanına sahiptirler. DOIB en basit tanımıyla, iç yüzeylerinde mikro olukları olan bir kutu ve bir ısı kaynağıyla gideri arasında, içindeki çalışan sıvının faz değişimi sayesinde düşük sıcaklık farklarında yüksek ısı iletimi yapabilen bir köprüdür. Isı kaynağının oluğun bir ucundaki çalışan sıvıyı buharlaştırması, oluşan buharın kendi oluşturduğu basınç sayesinde ilerleyerek diğer ucta tekrar yoğuşarak oluğun içine geri akması ve oluğun içinde, kılcal etki sayesinde başladığı yere tekrar akarak bu döngüyü devam ettirmesi ile çalışırlar. DOIB'larındaki ısı transferini ve akışı modellemek boyut, oluk şekli, çalışan sıvının doldurulma oranı ve materyal çeşitleri gibi bir çok parametrenin operasyonel limitler üzerine olan etkisini anlamak, kullanım şartlarına özgü tasarımlar yapabilmek açısından kritiktir. Literatürde bir takım basitleştirmelere ve varsayımlara dayanan bir çok model bulunmaktadır. Bu tezde ise, akışı ve ısı transferini ilk defa üç boyutta olarak modelleyen ve COMSOL Multiphysics® *via* LiveLink™ for MATLAB® arayüzü kullanmaya dayanan bir metot sunulmuştur. Bir programlama dilini kullanmanın yarattığı esnekliğin, bir ticari yazılımın enerji ve momentum çözücülerini kullanmanın getirdiği rahatlıkla birleşmesi ile, bu problemin tek başına bir HAD yazılımıyla veya bir programlama diliyle çözülmeye çalışılması durumunda ortaya çıkacak büyük engelleri aşabilecek kuvvetli ve kullanışlı bir araç ortaya çıkmıştır. Bu modelde eksenel yönde, oluk içindeki çalışan sıvının eğrilik yarı çapının (R) değişimi, oluk duvarının sıcaklık gradyenti (T_w) ve buhar sıcaklığı (T_v) ısı borularının esas çalışma parametreleri olmakla birlikte ısı borularının etkinliğini de gösterirler. Sunulan metot dahilinde, bu değişkenler için önce birer değer tahmin

edilir ve daha sonra da iç içe ve sıralı bir iterasyon seti ile hesaplanırlar. R iterasyonu için momentum denklemleri, T_w iterasyonu için enerji denklemleri çözülürken, T_v için objektif fonksiyonu, kütlenin korunumuna dayanan bir secant metotu kullanılır. İterasyonlar sırasında, değişkenler için hesaplanan her bir değere karşılık olarak da sürekli olarak çözüm bölgesi baştan oluşturulur ve sınır koşulları tekrar hesaplanır. Sunulan model, doğrulama amacıyla literatürle kıyaslanmıştır. Daha sonra, düz olukların derinliğinin ısı borusunun performansına olan etkisini gözlemleme amacıyla parametrik bir çalışma yürütülmüştür. Son olarak farklı ısı girdileri, verilen bir ısı borusundaki "dry-out"un saptanması amacıyla denenmiştir.

Anahtar sözcükler: Düz Oluklu Isı Borusu, Çok Boyutlu Isı Transferi ve Akış, Modelleme.

Acknowledgement

I would like to express my deep gratitude to Dr. Barbaros Çetin for his guidance with extensive knowledge and determination. He showed me well how to think like a scientist and an engineer at the same time. He set a good example of a professional that works both in detail and practically.

I would also want to offer my special thanks to Prof. Zafer Dursunkaya for his generous and valuable contribution to my thesis. The time I spent in our meetings with him has always been pleasant and productive thanks to his great experience and knowledge in the field, combined with the kindest attitude.

Dr. Yiğit Akkuş has supported me with his motivating advices and guidance in the most critical moments. His enthusiastic approach to science and appetite for everlasting learning are an inspiration to me.

Mr. Gökay Gökçe has enormous contributions to my project. He helped me with all his experience, and supplied me with the data I need, and spent his limited time to have meetings with me at his Ph.D. thesis phase.

I would also like to give my sincere thanks to the following people; Dr. Yildiray Yildiz and Dr. Mujdat Tohumcu for having confidence in me during my assistantship, Dr. Sakir Baytaroglu giving me warm encouragement for my work, and our department administrative assistant Ela Baycan for giving me all the help with a very friendly manner.

Writing this thesis was a painful process, and it would be impossible to accomplish it without the company of the best friend group in the Milky Way. Murat Yücelarslan, Mert Yüksel, Müge Özcan, Dilara Uslu, Cem Aygül and Orhun Ayar; one hell of a group indeed.

Last but not least, no matter what your grades are or how successful your thesis is, I believe family is the most important thing in a person's life. My M.Sc. studies came with some sacrifices, and not spending enough time with my family

is one of them. Thank you for everything you all have done for me; my psycho sister Ceren, my heroic father Nail, my beautiful mother Sibel, my best friend (also grandfather) Ziya, my beloved and beautiful fionce İrem, and everyone else. I love you all.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the increasingly emerging advances in the electronic component technology, it has been possible to fit more and more transistors in smaller microchips. Even though this possibility leads to manufacturing chips with superior performances and smaller volumes, higher transistor densities cause higher heat releases to be dissipated. This is simply due to the fact that more transistor means larger process capacity, but the release of heat by the continuous motion of transistors in smaller and denser volumes also means higher heat fluxes. In 2004, it was predicted that the heat flux produced by the electronic components was to increase up to 200 W/cm^2 in 2020 from 50 W/cm^2 in international electronics manufacturing initiative (iNEMI) technology roadmap [1]. Today, these expectations are already exceeded such that the electronics industry has been struggling to dissipate very high heat fluxes up to 300 W/cm^2 in 2007 [2,3]. To be able to cool down the advanced chips and keep the electronic devices in their operational temperature limits, electronic cooling technology has been trying to respond the cooling needs shaped by the increased and highly non-uniform heat releases. [4, 5].

There have been many proposals and methods introduced since the beginning of the electronic cooling field in order to meet the ever-growing electronics cooling demands, which can be categorized by their heat transfer efficiency as following [6]:

1. Free convection and radiation,

2. Forced air-cooling,

3. Forced liquid-cooling,

4. Liquid evaporation,

5. Newly emerging technologies.

The first four methods are the conventional ones and have been accepted as incapable for cooling down current microchips for various reasons. In terms of effective heat transfer coefficient, by using a gas, free convection provides approximately 15 W/m^2K to 100 W/m^2K and forced convection can go as high as 350 W/m^2K where both of them are insufficient in terms of effective heat transfer coefficient. For a liquid, free convection and forced convection can provide up to 1200 W/m^2K and 3000 W/m^2K respectively where effective heat transfer coefficients are still low in addition to using a liquid that can easily damage the electronics. Liquid evaporation techniques can provide up to 100,000 W/m^2K but are not convenient for cooling down small electronics and expensive [4, 5]. The fifth method in the list above includes techniques that can meet electronics cooling needs without having difficulties that conventional techniques experience such as insufficient heat transfer coefficient, direct liquid contact, occupying considerable space and need for an external source. Some major examples are plate fin heat exchangers, spray cooling and heat pipes .

Heat pipes are passive devices with very high thermal conductivity, and able to transfer large amount of heat with very small temperature differences by utilizing phase change mechanism of a working fluid circulating within the heat pipe. Some common kinds of heat pipes are tubular, pulsating, loop, sorption, and micro heat pipes [7, 8]. Heat pipes have many application areas from thermal management of electronic components to aerospace industry due to their robustness and heat transfer effectiveness [9–11]. They are closed systems that require no power input and their reliability lies in their customizable, simple, and passive structure with no moving parts, which make them convenient choice for electronics cooling, and

have attracted the most attention when first introduced by Cotter [12]. When a heat pipe is placed between a heat source and a heat sink, it transfers heat acting like a solid piece with high effective thermal conductivities that are hundred times larger than of copper [13, 14]. This high effective thermal conductivity is achieved by the latent heat released and absorbed as the phase change of the working fluid occurs. Accordingly, heat pipe's purpose is not cooling the object directly but dispersing the heat by effectively transferring it from the source to the sink. Heat pipes are sealed containers that consist of capillary alters for sustaining fluid flow, and conjunct volume for vapor flow. Correspondingly, flat grooved heat pipe (FGHP) is a specific kind that utilize micro grooves at the flat inner surfaces of the heat pipe for the fluid flow.

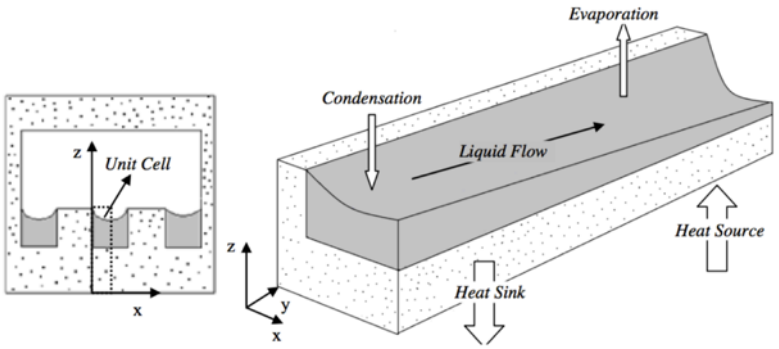


Figure 1.1: Working principle of a micro-grooved heat pipe [15]

During an operation, three main regions occur in heat pipe along its axis: evaporation region above the heat source, condensation region above the heat sink and the adiabatic region between them. Heat source evaporates the working fluid in the evaporation region and newly formed vapor creates a pressure difference that drives the vapor towards the condensation region. The arriving vapor condensates in the condensation region due to the colder temperatures caused by the heat sink and condensates into the grooves. Then, the condensed fluid flows back to the evaporation region due to the capillary forces acting in the, only to evaporate again and repeat the two-phase flow recurrently until the operation is stopped, as shown in the Fig. 1.1.

1.1 Operational Limits of Heat Pipes

As it is mentioned in the previous sections, heat pipes have a certain amount of working fluid in it which evaporates and condenses continuously throughout a cycling process. Phase change in a heat pipe is a tricky balance, so in order to sustain this cycle, a heat pipe should be designed carefully based on the operational conditions that they are intended to be used. Working fluid properties, groove geometry, dimensions and materials are all important design parameters and should be chosen according to the heat source and the sink of the application. This also means that, in order to remain operational and function effectively, the system needs to work under relatively predictable conditions rather than working in unstable conditions, such as the variation of the heat source power and the heat sink parameters. A heat pipe fail means a dry-out which means inadequate flow from evaporation to condensation region in the grooves. The following are the operational limits that in case of being exceeded, a heat pipe may dysfunction or loose efficiency [15, 23].

1) **Capillary Limit** Capillary force is an adhesion force, which is the action between the small size solid container (FGHP grooves in this case) and the liquid in it. If the capillary force overcomes the surface tension of the fluid, then the fluid will be forced to move upward. Eq. (1.1) is Young-Laplace equation that gives the capillary pressure created by two menisci (vapor-liquid interface) of different radius of curvatures, where σ is the surface tension.

$$P_c = \sigma \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \tag{1.1}$$

P_c is the main driving force on the liquid in the grooves and should be larger than the pressure loss along the heat pipe:

$$\Delta P_c > \Delta P_v + \Delta P_l + \Delta P_g \tag{1.2}$$

where ΔP_v is the pressure gradient required to drive the vapor from evaporation to condensation regions, ΔP_l is the pressure gradient required to drive the liquid from condensation to evaporation regions and ΔP_g is the

gravitational head. The radius of curvature of the working fluid in the grooves decreases from condensation to the evaporation region. If the differences between the radii get closer, P_c will be smaller and not be able to satisfy Eq. (1.2) that will result in the dry-out situation.

2) **Entrainment Limit** As it is previously told in the introduction the evaporated working fluid flows in the vapor channel to the condensation part and the condensed fluid flows in the micro-channels to the evaporation part. These opposite flows cause shear forces at their vapor-liquid intersection. This interaction may cause vapor to disjoin some liquid which will result in insufficient liquid flow in the micro-channels and may cause dry-out.

3) **Boiling Limit** The heat input to the heat pipe reaches to the liquid surface (liquid-vapor interface) partly by conduction through container and the liquid, and by convection. If the heat input is excessive the nucleate boiling will occur in liquid-container interface that will cause bubbles. These bubbles may block the liquid flow to the liquid surface where the evaporation occurs from, and cause dry-out by blocking the evaporation process.

4) **Viscous Limit** If the heat input is low or the heat pipe operates at low temperatures, the amount of vapor occurring in the evaporation region may not be enough to provide the necessary pressure gradient that drives the vapor to the condensation region. If such low pressure gradient occurs, the vapor flow will be suppressed by the opposing viscous forces in the vapor channel.

5) **Sonic Limit** This is a limit that mostly observed during low temperature operations or start-ups situations due the temperature associated very low vapor densities. As the vapor velocity increases towards the condensation region with the addition of the evaporated fluids, the inertial effects become significant and the vapor flow becomes choked (sonic). Due to this occurrence, nearly isothermal state of the vapor flow changes and a significant temperature gradient occurs. This does not result in a dry-out situation, however a deviation from isothermal behavior means decrease in FGHP's performance.

1.2 Modeling of Heat Pipes

Rather than being general purpose devices, heat pipes are to be designed for specific applications and operational conditions that show differences in terms of geometry, ambient temperature, power of the heat source etc. Type of the working fluid is case specific, where a liquid should comply with the geometry and the container material, and should be suitable for the thermo-chemical needs of the application. Amount of the working liquid loaded to a heat pipe is also important in order to avoid dry-out due to the low level of the fluid, and performance loss due to the excessive filling. Predicting the filling ratio of the working liquid is crucial in this sense. Accordingly, mathematical modeling of heat pipes accurately is critical for predicting the maximum heat transfer capacity of it and the compatibility of the design with the operational limits. For practical purposes, designing an optimum heat pipe requires a strong mathematical model rather than experimentation, considering the vast options in material, filling ratio and geometry.

Since the first introduction of heat pipes, many mathematical models with different approaches are presented by the researchers both analytically and numerically. Some common simplifications that the researchers employ for their studies to model the complicated nature of the problem can be summarized as the following:

- Downgrading the dimension of the actual problem,
- Neglecting axial temperature distribution,
- Uniform evaporation and condensation rates along the heat pipe axis,
- Neglecting the fluid profile variation along the axis,
- Fixed container temperature,
- Neglecting evaporation and condensation in the adiabatic zone.

The first model was introduced by Cotter that approximates the maximum heat load capacity of a triangular cross-section micro heat pipe, resulted from 1D momentum and continuity equations [12]. Then a energy approach that neglects the conduction in the container due to the hight thermal conductivity compare to that of the liquid, and calculates the phase change according to the heat input to the system, combined with a assumption of 1D flow of vapor and liquid was utilized for trapezoidal channels by Babil *et al.* [24]. Capillary pressure was correlated with the pressure drop in the channel in order to estimate dry-out limits. Later, Khrustalev *et al.* modeled the evaporation and condensation in a triangular groove using the kinetic theory and the variation of the liquid film thickness, and included the temperature gradient in the axial direction to calculate condensation and evaporation heat fluxes [25]. They also included the shear stress at the liquid-vapor interface and concluded that it decreases the thermal performance of heat pipes. El-Nasr *et al.* presented a thermal resistance analogy model for the estimation of effect of number of grooves on the thermal performance of a heat pipe and included the conduction in the container [26]. Peterson *et al.* presented a mathematical model that estimates the minimum radius of curvature of the meniscus and used it to calculate maximum heat transfer capacity [27]. They also introduced varying friction factor along the triangular micro channel, and showed that apex angles of the triangular grooves, liquid-solid contact angle, and heat pipe length affects maximum heat transport capacity dramatically. Anand *et al.* estimated the dry-out point by measuring axial temperature distribution as a function of heat input, both with dry and wet profiles of a v-grooved heat pipe [28]. They matched this experimental data with a theoretical analysis that includes inclination of the heat pipe. Kim *et al.* developed a mathematical model that includes the initial fluid charge along with the contact angle [29]. They introduced modified Shah model that accounts for the liquid-vapor interface shear stress to predict the maximum heat transfer capacity, and verified the model with experimentation. Launay *et al.* studied the thermal performance of triangular grooved heat pipes depending on the initial fill ratio [30]. They predicted the dry-out and flooding length by developing a model that counts for the coupling of micro and macro regions in the evaporation and condensation sections. Suman

et al. derived a mathematical model that predicts the dry-out region as a function of geometry, heat input inclination for any groove geometry [31]. In this model, they assumed 1D flow and temperature distribution, and compared their result with Anand [28]. Later, Suman *et al.* modified their existing model for v-shaped grooved heat pipe and included the conduction in the container and varying contact angle between the liquid and the solid [32]. They also studied the effect of angle of the v-groove on the thermal performance of the heat pipe. Lastly, Suman *et al.* extended their study by introducing transient terms both for the energy and momentum solutions, and succesfully compared their results with the existing data in the literature [33]. In their study, Lefevre *et al.* created a model that couples 2D hydrodynamic and 3D thermal equations for multiple heat sources and sinks, and derived isotherm patters for electronic components [34]. They based their calculations on a flat heat pipe with porous media without any directional grooves. In 2008, Do *et al.* developed an analytical model for FGHPs that solves one-dimensional conduction equation for the wall and the augmented Young–Laplace equation. Their results matched with the experimental data in literature by 20% [35]. Hyung *et al.* considered the variation of evaporation and condensation rates along the heat pipe in their model by coupling 1D conduction equation with augmented Young-Laplace equation [36]. The shear stress at the liquid-vapor interface, varying contact angle, and the liquid charge in their model, which is used for optimizing the maximum heat transfer capacity of a rectangular grooved heat pipe. Xiao *et al.* developed a fully three dimensional model for a heat pipe with rectangular grooves [37]. They assumed a fixed fluid profile along the wicks and considered an axial portion of the heat pipe for the solution domain in order to investigate the effects of structure of the wick column and the heat pipe size on the axial temperature and pressure gradients. Qu *et al.* investigated functional surfaces both experimentally and analytically by comparing regular surfaces with the hydrophilic ones [38]. They focused on triangular heat pipe and developed a model that take maximum heat input and contact angle into the account. They concluded their study by showing functional surfaces improves the maximum heat transfer capacity significantly. Lefevre *et al.* coupled thermal and hydrodynamic model for the prediction of temperature field of flat-grooved heat pipes [39]. In their model, both the vapor and pressure distribution is considered

in vapor along with the liquid. They investigated the effect of rectangular groove dimensions on the maximum heat transport capacity. In their work, Sonan *et al.* extended the work in [37] by introducing transient terms into both 2D momentum and 3D energy equations [40]. They investigated the start-up conditions of a flat heat pipe with porous media, and compared the results with the steady-state conditions. Aghvami *et al.* developed a model for predicting the 2D temperature distribution in the heat pipe wall depending on the vapor and liquid flow [41]. They included the viscous and inertial effects for the vapor, and the Darcian effect for the liquid flow in the porous wick, and compared their results with experimental data. Hung *et al.* developed a 1D model from the conservation of mass, energy, and continuity both for the liquid and the vapor flow [16]. They studied star-groove micro heat pipes with various configurations of the different number of apexes and apex angles. Maximum heat transfer capacities of different star-groove variations are compared . Later, Hung *et al.* adapted the same model for triangular grooves, and included the effect of the gravity on the thermal performance [42]. Thuchayapong *et al.* introduced a finite element based numerical solution of 2D momentum and energy equations [43]. They also investigated the effect of water jacket on the thermal performance of the heat pipe. Chauris *et al.* conducted one dimensional analysis of a hybrid groove consist of triangular and drop shapes [44]. Chang *et al.* modeled a triangular grooved heat pipe and proposed a circulation parameter that measures the circulation of the working fluid more accurate than the component of the Merit number [45]. Odabasi presented a model that solves the 3D heat transfer equations in both the solid and the liquid, coupled with a simplified 1D momentum equation [15]. The summary of the literature on heat pipe modeling can be found in Table 1.1.

1.3 Objectives and Motivation

In this study a new approach with the solution of momentum and energy equations in 3D combined with modeling phase change with the kinetic theory, without the simplifications aforementioned is proposed for a FGHP with rectangular grooves. Initially, a similar methodology was proposed by Odabasi, however flow

was solved 1D in a simplified geometry. Related to the study of Odabasi [15], a methodology of combined iterative processes is followed in the presented work. Vapor temperature (T_v), radius of curvature of the working fluid (R), the temperature gradient along the heat pipe wall (T_w), and the geometry of the problem are not known *a priori* and are estimated initially. During the solution process, T_v, R, and T_w are iterated consecutively by the solutions of energy and momentum equations. At each iteration, 3D solution geometry also changes related to the R. Boundary conditions, which are the functions of T_v, T_w and R also change at each of the iteration. To accomplish this difficult task, 3D momentum and energy equations are solved with COMSOL Multiphysics®. In order to use COMSOL automatically in such iterative process where the geometry and the boundary conditions are updated, COMSOL Multiphysics® *via* LiveLink™ for MATLAB® is utilized, in which COMSOL is controlled in algorithmic environment of MATLAB. This cooperation enables the opportunity of taking advantage of using a commercial CFD software in terms of modeling the complicated physics of FGHP's and using the software in the flexible script environment to iterate the geometry and the boundary conditions. This flexibility also gives the opportunity of studying any channel geometry without changing the equations.

1.4 Thesis Outline

The summary of the chapters of the complete thesis is the following:

Chapter 1; A brief introduction to the relationship between electronic component technology and electronics cooling industry is given. Different kinds of electronic cooling are told and heat pipes as a necessary and emerging technology are discussed. Application areas and the working principle of the FGHP are given. Different types of groove structures, materials and the various working fluid are mentioned. A literature review on the subject of the thesis is presented. Objectives and the hallmarks of the presented work are discussed.

Chapter 2; The procedure of generating the complex solution domain that

includes both the working fluid and the container is discussed. The derivation of evaporation and condensation models by the kinetic theory are explained. Boundary conditions for both the momentum and the energy equations are mentioned. Flowchart of the code written with COMSOL Multiphysics® *via* LiveLink™ for MATLAB® is given. Structure of the iteration is discussed.

Chapter 3; The results for momentum and energy solutions are presented. A detailed study for the verification of the selected boundary conditions is given. The model is validated by comparing it with the literature. The effects of different groove depths are compared with the validated model. The power of the heat source is studied for the detection of dry-out. Possible future directions for the study are discussed.

Table 1.1: Comparison of relative studies in literature

Reference	Year	Groove Cross-Section	Details
[12] Cotter	1984	Triangular	Analytical model for maximum heat transfer
[24] Babin *et al.*	1990	Trapezoidal	Analytical model for maximum heat transfer
[25] Khrustalev *et al.*	1994	Triangular	Inclusion of liquid-vapor interface shear stress and varying liquid profile
[26] El-Nasr *et al.*	1996	-	Thermal resistance analysis
[27] Peterson *et al.*	1996	Triangular	Inclusion of 2D model for liquid friction factor in a model
[28] Anand *et al.*	2002	V-shaped	Analytical model for dry-out prediction
[29] Kim *et al.*	2003	Circular heat pipe	Modified Shah method for 2D heat transfer analysis
[30] Launay *et al.*	2004	Triangular	Effect of micro region on evaporation is included in thermal analysis
[31] Suman *et al.*	2005	Any polygonal shape	Coupled, 1D flow and energy solution
[32] Suman *et al.*	2005	V-shaped	Uniform temperature distribution of the container is included in the analysis
[33] Suman *et al.*	2005	Any polygonal shape	Transient, 1D, and coupled flow and energy model
[34] Lefevre *et al.*	2006	Porous	3D conduction and 2D flow models are coupled and solved
[35] Do *et al.*	2008	Rectangular	1D Conduction equation / Augmented Young-Laplace equations are coupled
[36] Hyung *et al.*	2008	Rectangular	Inclusion of axial variation of evaporation and condensation in a model
[37] Xiao *et al.*	2008	Rectangular	3D Conduction and flow are coupled and solved for constant fluid geometry
[38] Qu *et al.*	2008	Triangular	Analysis of the effect of functional surfaces on the thermal performance
[39] Lefevre *et al.*	2008	Rectangular	Inclusion of equivalent thermal conductivity of the capillary structure
[40] Sonan *et al.*	2008	Porous	Modeling multiple heat source in a transient solution
[41] Aghvami *et al.*	2011	Porous	Inclusion of evaporation and condensation in the adiabatic region
[16] Hung *et al.*	2011	Star-groove	Thermal analysis of various star-groove geometries
[43] Thuchayapong *et al.*	2012	-	2D flow and energy solution with Finite Element Method
[42] Hung *et al.*	2012	Rectangular	Analysis of inclined orientation of FGHPs
[44] Chauris *et al.*	2013	Circular/triangular	Analysis on hybrid shaped grooves
[45] Chang *et al.*	2013	Triangular	Proposition of circulation effectiveness parameter
[15] Odabasi	2014	Rectangular	Full 3D energy solution with 1D flow model

Chapter 2

Computational Model

Mathematical modeling of a grooved heat pipe is a challenging task due to the complicated physics involved in the working mechanism and since the free surface of the liquid is part of the solution of the problem. Grooved heat pipes are sealed containers that transfer heat with high thermal efficiency, works between a heat source and a sink. There are three zones along the grooves; the evaporation region that is formed due to the heat source in contact with the heat pipe, the condensation region that is created by the external heat sink, and the adiabatic region between them that still has evaporation and condensation in them, but not aligned with neither the heat source nor the sink. The periodic operation of a FGHP can be described as following respectively:

1. Working fluid in the grooves evaporates into the vapor chamber due to the higher temperatures caused by the heat source at the evaporation region.
2. Emerging vapor creates a ΔP_v that drives the vapor towards the condensation region.
3. The vapor condensates in the condensation region due to the colder temperatures maintained by the heat sink, and flows back into the grooves.
4. The condensed working fluid in the grooves flows to the evaporation region by the capillary action ΔP_c, and completes a cycle.

It order to explain the complexity of the problem, it is important to note that both the evaporation and the condensation occur along the axis at inconstant rates. To be more clear, it can be generalized that if the groove wall temperature (T_w) is lower than the vapor temperature (T_v) condensation of the vapor occurs with a rate proportional to ΔT_{w-v}. Similarly, the evaporation of the liquid occurs if the T_w is higher than the T_v, at a rate proportional to ΔT_{w-v}. It should also be noted that fluid profile (the radius of curvature (R) gradient of the meniscus) does not remain constant along the groove and depends on the phase change mass flow rates, therefore also on the ΔT_{w-v}. Also, since R determines the geometry of the fluid domain, it also affects the temperature distribution in the domain, therefore T_w.

In this study, both the flow and the heat transfer are solved subsequently in the three dimensional, exact, and complex problem domain. This became possible due to the utilization of a commercial software, COMSOL Multiphysics® *via* LiveLink™ for MATLAB®. Beside the fixed geometric and material parameters, R, T_w, and T_v are the core variables that are the functions of each other as briefly mentioned in the previous paragraph, and are decisive in the calculations of the boundary conditions. The dependency of these variables on each other means that each time one of them is iterated the other variables change as well, as the example of the relationship between the geometry and the wall temperature distribution given in the previous paragraph.These variables are iterated: R by the momentum solution, T_w by the energy solution, and lastly T_v is modified by the Secant Method according to the results of these solutions which is briefly similar to a case of three equations with three unknowns.

2.1 Generating Heat Pipe Geometry

The proposed solution includes the usage of COMSOL for three dimensional momentum and energy equations. To do so, the solution domain should be formed. The problem domain consist of half of the rectangular groove/working liquid couple and the half fin top, due to the existence of symmetry. The domain includes

two subdomains: the liquid and the solid. The solid subdomain remains constant throughout the iteration process and depends on the values defined before the solution by the user. However the volume and the shape of the liquid subdomain are not constant and shaped by the evaporation and condensation mass fluxes going in and out. The liquid subdomain is used for the flow solution, where as both the liquid and the solid sub domains are used for convective and conductive heat transfer.

The change of the overall volume of the liquid domain by the iterations is not the only challenge but also the existence of the irregular shape at the liquid-vapor interface. The reason for this complex surface is the decreasing radius of curvature value of the liquid-vapor interface from condensation to the evaporation regions. Using LiveLink, this complex shape is generated by creating N many cross-sections with different radius of curvatures according to the R distribution (See Fig. 1.1 and Fig. 2.1) along the groove.

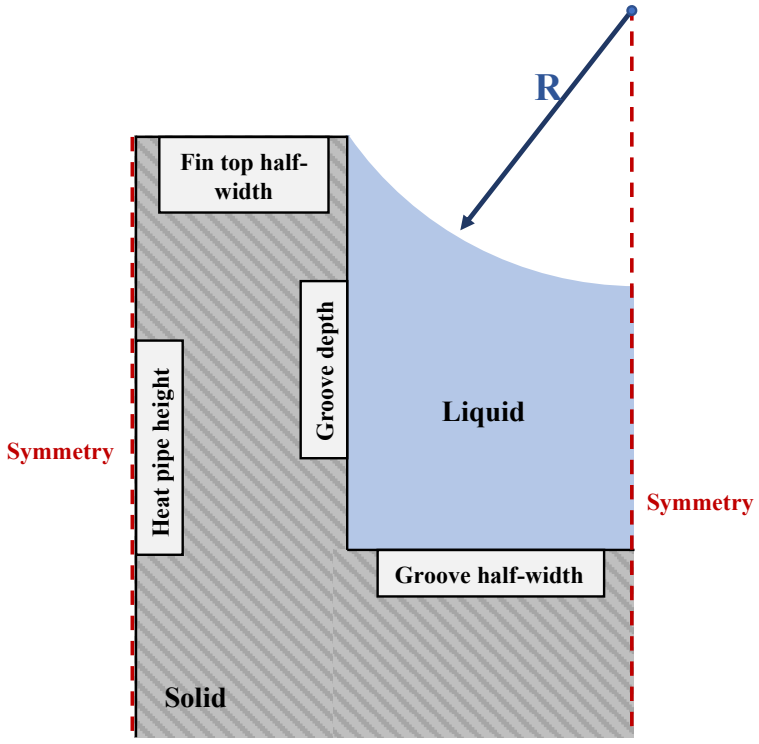


Figure 2.1: A cross-section of the solution domain

The heat pipe geometry in the cross-sections are created with the polygons, and

the arc-shaped liquid-vapor interface is with the interpolation curves. Note that any groove profile can be created easily with the curves and the polygons. The resolution of the geometry and the surface can also be increased by the number of points supplied for the curve. Conjoining these newly generated cross-sections later gives the desired 3D domain (See Fig. 2.2). Also, the resolution of the geometry and the model can also be boosted by increasing the number of the cross-sections along the FGHP. For every iteration of R, geometry is updated in the same way according to the new value of R.

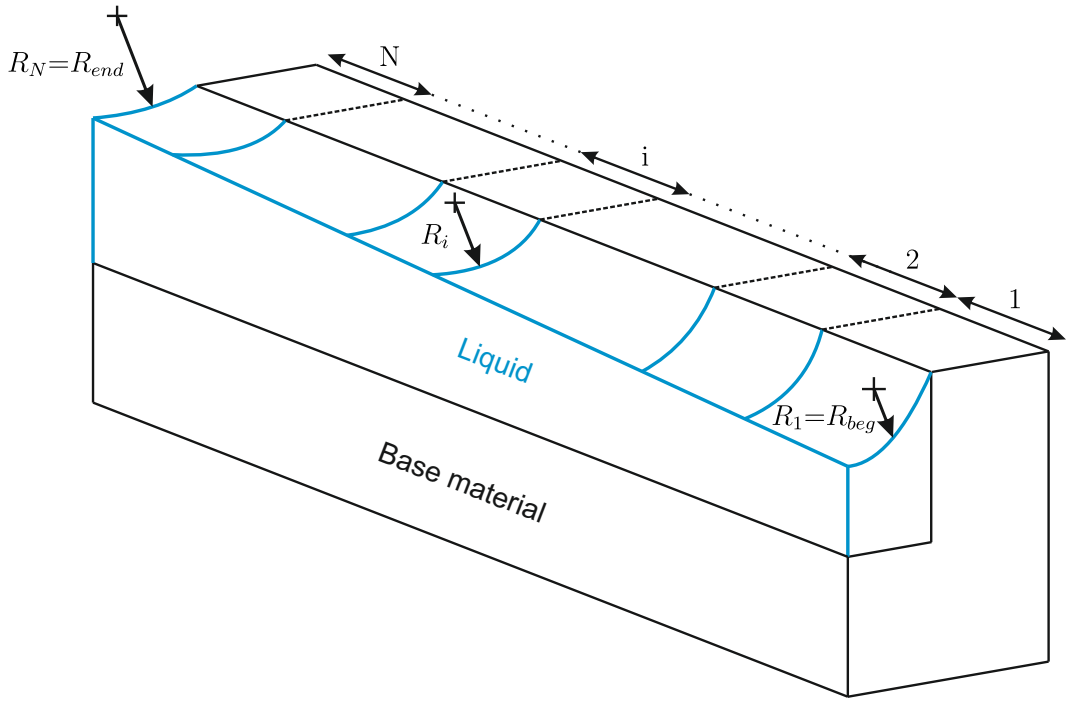


Figure 2.2: Solution domain with joined N-many cross-sections

2.2 Boundary Conditions

The proposed solution of the FGHP problem includes the 3D analysis of momentum and energy equations. Both of the analysis are solved in a subsequent but in a coupled manner since the same phase change boundary condition values are used as mass flow rate and convective heat flux for momentum and energy solutions respectively. In the following parts, details of the phase change and other

boundary conditions for both of the solutions are described.

2.2.1 Momentum Boundary Conditions

Momentum equations are solved for the fluid flow in the rectangular grooves of FGHP. To recall, this flow is sustained by the mass flow into the channels by the condensation of the fluid and the mass flow out due to the evaporation of the liquid out of the grooves. Difference between the resulting pressure distribution in the liquid (P_l) and the known vapor pressure (P_v) gives the new radius of curvature (R) distribution along the grooves, using the Young-Laplace equation and is used for updating the fluid profile. The boundary conditions that are used for the momentum solution are summarized in this section. For this part, steady-state Navier Stoke's equations for incompressible flow, and continuity equation are coupled and solved:

$$\rho(\mathbf{u} \cdot \nabla \mathbf{u}) = \nabla p + \nabla \cdot \boldsymbol{\tau} \tag{2.1}$$

$$\nabla \cdot (\rho \mathbf{u}) = 0 \tag{2.2}$$

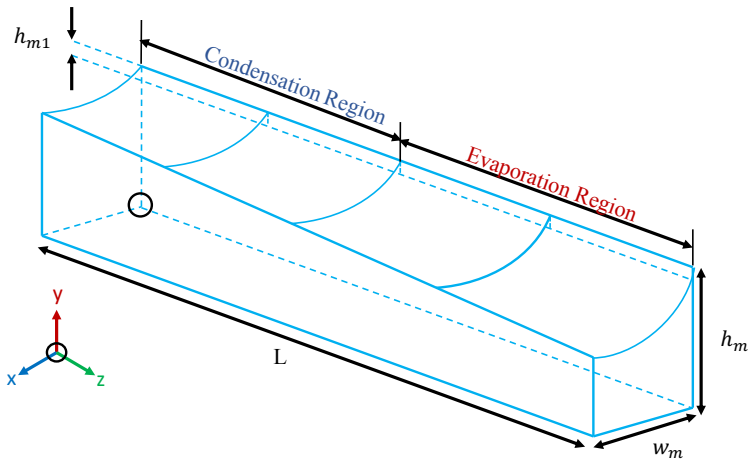


Figure 2.3: Momentum solution domain

1) Mass Flow Rate BC

The occurrence of evaporation and condensation of the working fluid is projected as mass flows into and out from the solution domain in the momentum solutions. At each cross-section that is made and used for forming the fluid geometry, phase change mass fluxes are calculated and applied as mass flux boundary condition for the corresponding unit cell next to the cross-section. Phase change mass fluxes occur in different rates dominated by the difference of Tv - Tw. Accordingly, as it can be correlated from the temperature distribution given in Fig. 2.4, phase change mass fluxes are highest at the end points of the grooves and decrease as T_w approaches to T_v.

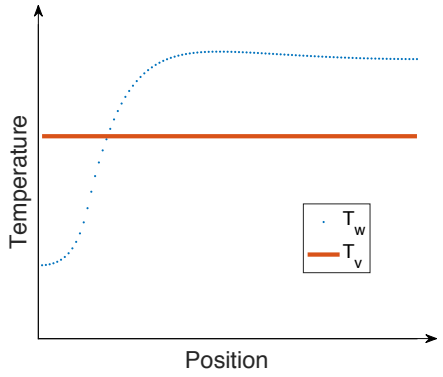


Figure 2.4: Representative T_w distribution and T_v

The mass flow rate boundary conditions are defined at the small portion of groove wall, right under the fin top corner as following, where i is the indice for unit cell:

$$\dot{m}^i = \dot{m}^i_{cond} \quad \text{at} \quad 0 < z < L_{cond}; \quad x = 0; \quad h_m - h_{m1} < y < h_m \tag{2.3a}$$

$$\dot{m}^i = \dot{m}^i_{evap} \quad \text{at} \quad L_{cond} < z < L_{total}; \quad x = 0; \quad h_m - h_{m1} < y < h_m \tag{2.3b}$$

2) Symmetry

The solution geometry is the three dimensional liquid domain (See Fig. 2.2), which is the axial half of a whole groove volume formed by the joint cross-sections. The geometry includes half width of the fluid filled micro-channel

due to the symmetry as it can be seen in Fig. 2.1. Therefore symmetry boundary condition is defined at the symmetry plane that divides micro channel in half in the axial direction, as following, where u is velocity and s is traction.

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{and} \quad \boldsymbol{\tau} \cdot \mathbf{t} = 0 \quad \text{at} \quad x = w_m \tag{2.4}$$

3) Slip Wall BC

In the momentum solution, the shear stress at the liquid-vapor interface that occurs due to opposite flows of both of the fluids is neglected and mass exchange at the interface is not allowed since phase change boundary conditions are defined at the fin top corner. For this reason, slip wall boundary condition is applied at the interface, which equals both shear stress and normal velocity to zero, identical to symmetry BC.

$$\mathbf{u} \cdot \mathbf{n} = 0 \quad \text{and} \quad \boldsymbol{\tau} \cdot \mathbf{t} = 0 \quad \text{at} \quad \text{liquid/vapor interface} \tag{2.5}$$

4) No-Slip Wall BC

The momentum solution geometry (fluid domain) shares boundaries with the groove wall as it is explained above. For this reason, all solid-liquid boundaries are defined as no-slip wall boundary condition except for the perpendicular wall at the end one at the evaporation region, marked by R_1 surfaces in Fig. 2.2.

$$\mathbf{u} = 0 \quad \text{at} \quad \text{solid/liquid interfaces} \tag{2.6}$$

5) Pressure Outlet BC

The reason for the perpendicular wall in the evaporation region is not set as no-slip boundary condition is to sustain the conservation of mass in the solution domain. As it is mentioned before, the solution algorithm includes the iteration of R, Tw and Tv values, which are used for the calculation of phase change mass fluxes. Until the convergence satisfies, evaporation and condensation mass flux boundary condition values will not be equal to each

other, therefore a boundary for compensating the inflow/outflow inequality is needed since mass flux boundary conditions are the only boundaries for mass inlet and outlet. Therefore, the pressure value at this boundary is defined by Young-Laplace equation using the corresponding R value:

$$P = P_v - \frac{\sigma}{R_{end}} \quad \text{at} \quad z = L \tag{2.7}$$

2.2.2 Energy Boundary Conditions

For the solution of the energy equations, both solid and the liquid domain are used as the problem domain. Fixed boundary conditions such as the heat sink and the source are applied, and as for the phase change boundary conditions -which are common inputs of both momentum and energy solutions- convective heat transfer boundary condition is used with $h_{PhaseChange}$ and T_v. The location of the boundary conditions are given in the Fig. 2.5 as supplementary.

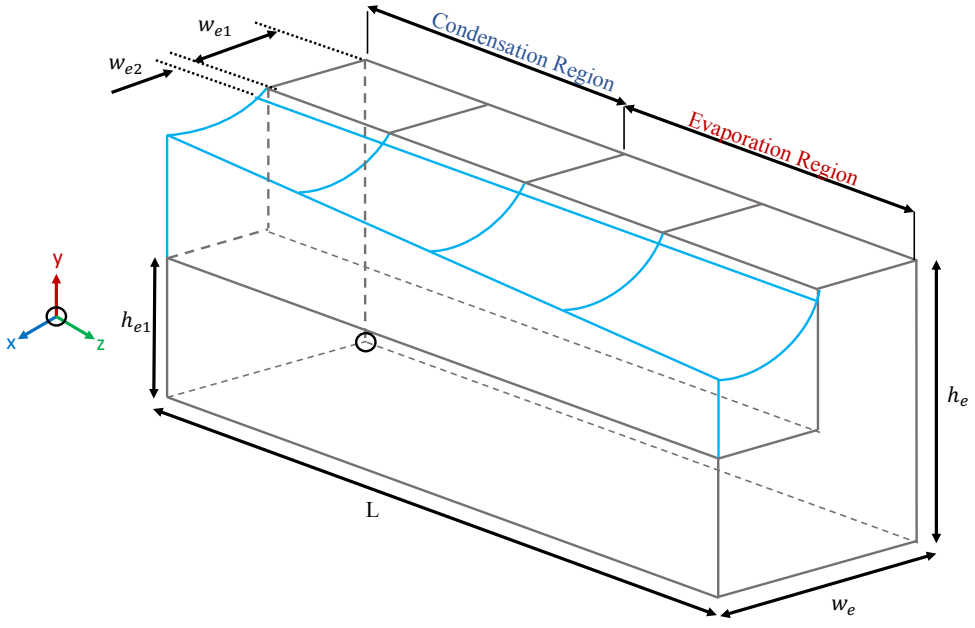


Figure 2.5: Energy solution domain

1) Symmetry BC

Solution domain presented for the energy domain includes the same groove volume used for the momentum solution, and corresponding portion of the container from mid-fin top to mid-groove with full height. Accordingly, unlike the momentum solution domain, the geometry is now symmetric in both sides, parallel to the axial direction.

$$-k\nabla_n T = 0 \quad \text{at} \quad x = 0; \quad x = w_e \tag{2.8}$$

$$k = k_s \quad \text{at} \quad 0 < y < h_{e1}; \quad x = w_e \tag{2.8a}$$

$$k = k_l \quad \text{at} \quad h_{e1} < y < h_e; \quad x = w_e \tag{2.8b}$$

$$k = k_s \quad \text{at} \quad x = 0 \tag{2.8c}$$

2) Heat Source BC

The working principle of a FGHP is to transfer heat from a source to a sink with high thermal efficiency. In order to model heat source, constant heat flux boundary condition is applied at the evaporation part of the bottom of the FGHP.

$$-k_s\nabla_n T = q''_{in} \quad \text{at} \quad y = 0; \quad L_{source} < z < L \tag{2.9}$$

3) Heat Sink BC

Heat sink is the location where heat transferred by the FGHP is expelled from the system. For this purpose, convective heat flux is defined at the condensation part of the bottom of the FGHP with predefined ambient temperature (T_∞) and heat transfer coefficient (h_∞).

$$-k_s\nabla_n T = h_{amb}(T - T_{amb}) \quad \text{at} \quad y = 0; \quad 0 < z < L_{sink} \tag{2.10}$$

4) Phase Change Convective Heat Flux BC

In the solution of the energy equations, the evaporation and condensation phase changes are projected as convective heat transfer. In the same way

with the momentum solution, occurrence of the evaporation and condensation at different rates depending on the temperature difference $T_w - T_v$ are applied. Micro region evaporation and condensation mass fluxes are calculated for each cross-section, and converted to a heat transfer coefficient value (h^*), which will be explained in the next chapter. Calculated h^* value is used as convective heat transfer boundary condition along with T_v at the corresponding unit cell. The evaporation heat transfer is defined at the thin sheet in the liquid region, near the fin top corner. The condensation heat transfer is given on the fin top surface and on a thin extension surface over the liquid region near the fin top corner. The reason for the selection of these surfaces for the evaporation and condensation boundary conditions will be discussed in the following sections and chapters.

$$\begin{aligned} -k_s \nabla_n T = h_{cond}(T - T_v) \quad \text{at} \quad & 0 < z < L_{cond} \\ & 0 < x < w_{e1} + w_{e2} \\ & y = h_e \end{aligned} \tag{2.11a}$$

$$\begin{aligned} -k_s \nabla_n T = h_{evap}(T - T_v) \quad \text{at} \quad & L_{cond} < z < L_{total} \\ & w_{e1} < x < w_{e1} + w_{e2} \\ & y = h_e \end{aligned} \tag{2.11b}$$

5) Temperature BC

In this solution procedure, vapor temperature is assumed to be constant along FGHP. In order to apply this, the liquid-vapor interface is defined as temperature boundary condition with T_v.

$$T = T_v \quad \text{at} \quad \text{liquid/vapor interface} \tag{2.12}$$

6) Thermal Insulation BC

Remaining surfaces, such as the adiabatic region between the heat source and the sink, both ends of the groove, and fin top at the evaporation region are set as thermal insulation.

2.3 Modeling of Evaporation and Condensation

Occurrence of evaporation and condensation is the basic driving force behind the very logic of heat pipes. High heat transfer capacity of FGHPs with very small temperature differences comes from the amount of transported heat during the phase change. Therefore accurate modeling of the phase change mechanism is crucial in order to calculate the thermal performance of a heat pipe. In this section, both the evaporation and the condensation models that are derived from the kinetic theory will be explained.

2.3.1 Evaporation Model

Evaporation is a vaporization process that transforms liquid into gas phase, and takes place at the surface of the liquid. The kinetic energy raises as the liquid molecules collide to each other. If the kinetic energy of the liquid molecules are high enough to suppress the intermolecular forces and overcome the vapor pressure, evaporation occurs. The kinetic energy is directly proportional to the temperature, so the evaporation process happens at faster rates at higher temperatures. For the derivation of the evaporation mass flow Odabasi's study is followed [15].

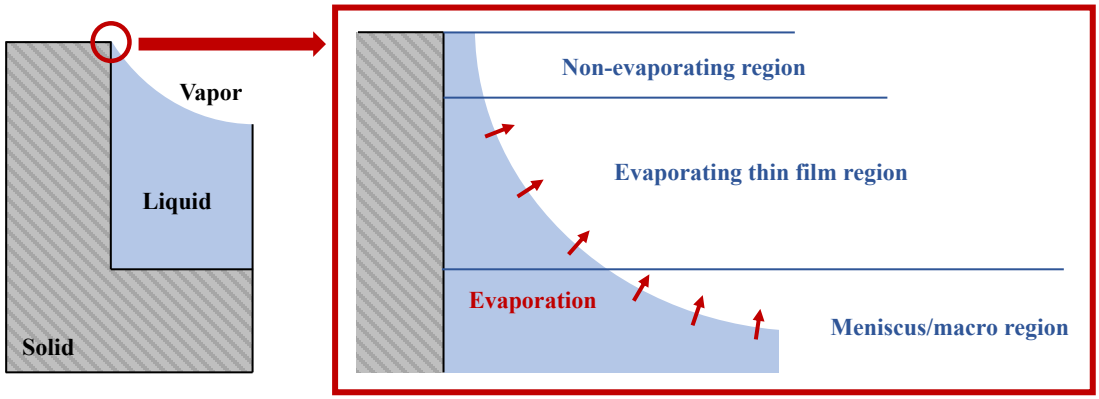


Figure 2.6: Evaporation subregions

The evaporation in a FGHP takes place in the grooves. The following three

subregions (See Fig. 2.6) occur at the liquid-vapor interface during evaporation process:

- *Meniscus/macro region*: Capillary force dominates the equilibrium in this region since the intermolecular forces have very low strength due to the very thick liquid layer. Capillary force formula, which was given before in Eq. (1.1) reduces to the following equation since the radius of curvature in the axial direction is very large compare to the one in the cross-sectional direction.

$$P_l - P_v = \frac{\sigma}{R} \tag{2.13}$$

  In this region, mass flux due to the evaporation can be calculated as following:

$$q_l" - q_v" = m_e" h_{lv} \tag{2.14}$$

- *Evaporating thin film region*: Capillary forces are still active due to the existence of the interface curvature and dominant together with the intermoleculer forces.

- *Non-evaporating region*: Thinnest layer of the liquid at the solid-liquid interface. Due to the nano-level thinness, the intermolecular forces between the solid-liquid interface are dominant and prevent the mass transfer due to phase change.

Due to the very low resistance to heat transfer that is caused by the very low thickness of the liquid layer in the micro region (evaporating thin film and non-evaporating regions), heat transfer rates, therefore the phase change mass fluxes, are considerable in this region [46]. The mass flux due to the phase change in the evaporating thin film region is defined as following:

$$m_e" = a(T_{lv} - T_v) + b(P_l - P_v) \tag{2.15}$$

and

$$a = \frac{2c}{2-c}\left(\frac{M}{2\pi R_u T_{lv}}\right)^{1/2}\left(\frac{MP_v h_{lv}}{R_u T_v T_{lv}}\right) \tag{2.15a}$$

$$b = \frac{2c}{2-c}\left(\frac{M}{2\pi R_u T_{lv}}\right)^{1/2}\left(\frac{P_v V_l}{R_u T_{lv}}\right) \tag{2.15b}$$

where M is the molecular weight, P_v is the vapor pressure, h_{lv} is the latent heat of evaporation, R_u is the universal gas constant, T_{lv} is the liquid-vapor interface temperature, T_v is the vapor temperature, V_l is the molar volume of the liquid, and c is the accommodation constant.

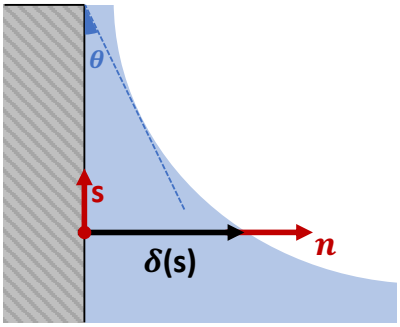


Figure 2.7: Coordinate system for the evaporating thin film calculations

The coordinate system that is defined for the solution of the kinetic theory equations can be seen in Fig. 2.7. Direction of s is the vertical direction in the FGHP cross-section, and its starting point is the macro-micro evaporation transition interface. n is the horizontal direction in the cross-section. δ is the thickness of the liquid layer which is a function of the direction s. Lastly, θ is the angle between the groove wall and the liquid.

According to the liquid-vapor interface that can be seen in Fig. 2.6, the pressure balance for the interface can be written as in Eq. (2.16)

$$P_{vapor} = P_{capillary} + P_{liquid} + P_{dispersion} \tag{2.16}$$

where $P_{dispersion}$ is the pressure caused by the hydration layers of polar molecules

near interfaces and significant in the micro evaporation region, and defined as:

$$P_d = \frac{A_d}{\delta^3} \tag{2.17}$$

where A_d is the dispersion constant. Following this, the capillary pressure P_c in Eq. (2.16) as a function of δ is:

$$P_c = \sigma \frac{d^2\delta/ds^2}{\left(1 + (d\delta/ds)^2\right)^{3/2}} \tag{2.18}$$

Using Eq. (2.17,2.18) and assuming P_v is constant along the s, the derivative of Eq. (2.16) with respect to s gives:

$$\frac{dP_l}{ds} = \frac{3A_d}{\delta^4}\frac{d\delta}{ds} - \sigma \frac{d^3\delta/ds^3}{\left(1 + (d\delta/ds)^2\right)^{3/2}} + 3\sigma \frac{(d^2\delta/ds^2)^2}{\left(1 + (d\delta/ds)^2\right)^{5/2}}\frac{d\delta}{ds} \tag{2.19}$$

Assuming the liquid flow from the macro region is one dimensional in s-direction, the equation and the boundary conditions for $\frac{dP_l}{ds}$ are:

$$\frac{dP_l}{ds} = \mu \frac{d^2u_l}{dn^2} \tag{2.20}$$

$$du_l/dn = 0 \qquad \text{at} \qquad n = \delta \tag{2.20a}$$

$$u_l = 0 \qquad \text{at} \qquad n = 0 \tag{2.20b}$$

Solving Eq. (2.20) with respect to boundary conditions Eq. (2.20a) and Eq. (2.20b), the liquid velocity in s-direction becomes:

$$u_l = \frac{1}{\mu}\frac{dP_l}{ds}\left(\frac{n^2}{2} - \delta n\right) \tag{2.21}$$

In order to get the evaporation mass flow rate, u_l is integrated according to Eq. (2.22):

$$m'_e = \int_0^{\delta} u_l dn \tag{2.22}$$

and differentiated with respect to s:

$$m"_e = -\frac{1}{3\nu}\frac{d}{ds}\left(\delta^3 \frac{dP_l}{ds}\right) \tag{2.23}$$

substituting Eq. (2.19) into Eq. (2.23) gives:

$$m"_e = -\frac{d}{ds}\left[\frac{\delta^3}{3\nu}\left(\frac{3A_d}{\delta^4}\frac{d\delta}{ds} - \sigma\frac{d^3\delta/ds^3}{\left(1+(d\delta/ds)^2\right)^{3/2}} + 3\sigma\frac{(d^2\delta/ds^2)^2}{[1+(d\delta/ds)^2]^{5/2}}\frac{d\delta}{ds}\right)\right] \tag{2.24}$$

Now that a equation for $m"_e$ as a function of δ alone is obtained, it can be used for coupling with Eq. (2.15) in order to get rid off T_{lv} that is unknown. To do so, $m"_e$ is written in terms of the heat flux between the wall and the liquid-wall interface, by adapting the formula Eq. (2.14):

$$m"_e = k_l \frac{T_w - T_{lv}}{\delta h_{lv}} \tag{2.25}$$

which reduces to a equation for T_{lv} when combined with Eq. (2.15):

$$T_{lv} = \frac{k_l T_w/\delta h_{lv} + aT_v + b(P_v - P_l)}{a + k_l/\delta h_{lv}} \tag{2.26}$$

Substituting T_{lv} into Eq. (2.15) gives:

$$m_e^{"} = \frac{a(T_w - T_v) + b(P_l - P_v)}{1 + a\delta h_{lv}/k_l} \tag{2.27}$$

which yields to the following equation when combined with the Eq. (2.24), subjected to the boundary conditions Eq. (2.28a) and (2.28b).

$$\begin{aligned} m_e^{"} &= \frac{a(T_w - T_v) + b(P_l - P_v)}{1 + a\delta h_{lv}/k_l} \\ &= -\frac{d}{ds}\left[\frac{\delta^3}{3\nu}\left(\frac{3A_d}{\delta^4}\frac{d\delta}{ds} - \sigma\frac{d^3\delta/ds^3}{\left(1+(d\delta/ds)^2\right)^{3/2}} + 3\sigma\frac{(d^2\delta/ds^2)^2}{\left(1+(d\delta/ds)^2\right)^{5/2}}\frac{d\delta}{ds}\right)\right] \end{aligned} \tag{2.28}$$

$$\delta = \delta_0 \qquad \text{at} \qquad s = 0 \tag{2.28a}$$

$$d\delta/ds = -tan\theta \qquad \text{at} \qquad s = 0 \tag{2.28b}$$

$$P_v - P_l = \sigma/R \qquad \text{at} \qquad s = 0 \tag{2.28c}$$

$$d(P_v - P_l)/ds = 0 \qquad \text{at} \qquad s = 0 \tag{2.28d}$$

$$m_e^{"} = 0 \qquad \text{at} \qquad s = l \tag{2.28e}$$

$$P_d = -\frac{\sigma}{R}10^{-5} = \frac{A_d}{\delta_0^3} \qquad \text{at} \qquad s = 0 \tag{2.28f}$$

where the last boundary conditions comes from the fact that the dispersion pressure is not effective at the macro region, therefore it is considered $1/10^5$ times of the capillary pressure at $s = 0$, where is the transition between the macro and the micro evaporation regions.

In the solution procedure, evaporation mass fluxes are calculated at each unit distance in the direction of s, starting from $s = 0$ to the non-evaporating micro region (see Figures 2.6 and 2.7) using the presented equations and finite difference method (FDM). Then the calculated mass fluxes are numerically integrated in the s-direction to obtain the total $m_{e,micro}^{'}$ for the cross-section. A function is created in MATLAB that applies this solution procedure and calculates the evaporating thin layer mass line flux (micro region evaporation mass line flux, $m_{e,micro}^{'}$). T_w and R values of each cross-section in the evaporation region is supplied to this

function along with T_v, material properties and the heat pipe dimensions which are fixed for all cross-sections. The computed value of $m^{'}_{e,micro}$ for each cross-section is multiplied with the length of the corresponding unit cell to obtain $m^{"}_{e,micro}$ which is the boundary condition for the unit cell.

2.3.2 Condensation Model

Condensation is the reverse process of the evaporation, where the gas transforms into liquid. Likewise, condensation is strongly proportional to temperature. In FGHPs, condensation occurs due to the temperatures generated by the heat sink that are lower than the saturation temperature of the gas. The condensation takes place mostly on the fin top, and fewer on the meniscus interface in the groove. The condensation occurs at the meniscus can be calculated as in Eq. (2.29), similar to the calculation of the evaporation at the meniscus side.

$$q_l" - q_v" = m^{"}_c h_{lv} \tag{2.29}$$

For the calculation of the fin top condensation, which will be called micro region condensation mass flow rate, Odabasi's study is taken as basis [15]. In Fig. 2.8 the coordinate system defined for the condensation calculations is given. Axis s is in the horizontal direction and zero at the half of the fin top. $\delta(s)$ is the film thickness above the fin top, and a function of s.

The pressure balance at the liquid-vapor interface above the fin top is similar to the one for the evaporation region except for the dispersion pressure:

$$P_{vapor} = P_{capillary} + P_{liquid} \tag{2.30}$$

Accordingly, condensation mass flux equation can be derived by modifying the evaporation equations Eq. (2.19) and Eq. (2.23) by assuming $d\delta/ds = 0$, where this assumption is made due to the fact that the variation of the film thickness (δ) is negligible. Following this, $m^{"}_c$ becomes:

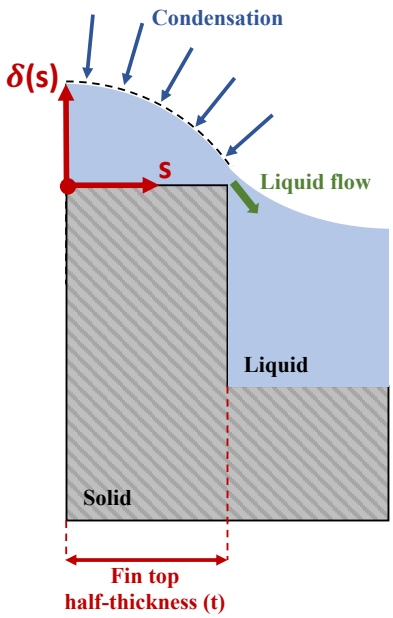


Figure 2.8: Coordinate system and the cross-section of the condensation region

$$\frac{\sigma}{3\nu}\frac{d}{ds}\left(\delta^3\frac{d^3\delta}{ds^3}\right) = -\frac{a(T_w - T_v) + b(P_l - P_v)}{1 + a\delta h_{lv}/k_l} \tag{2.31}$$

Same assumption also changes the previously given definition, Eq. (2.18), for the capillary pressure as following:

$$P_c = \sigma\frac{d^2\delta}{ds^2} \tag{2.32}$$

For the definition of the film thickness $\delta(s)$, a fourth degree polynomial equation is used and the boundary conditions for $\delta(s)$ are presented as following:

$$\delta(s) = c_0 + c_1(s-t) + c_2(s-t)^2 + c_3(s-t)^3 + c_4(s-t)^4 \tag{2.33}$$

$$d\delta/ds = 0 \qquad \text{at} \qquad s = 0 \tag{2.33a}$$

$$d^3\delta/ds^3 = 0 \qquad \text{at} \qquad s = 0 \tag{2.33b}$$

$$d\delta/ds = -tan(\pi/2 - \theta) \qquad \text{at} \qquad s = t \tag{2.33c}$$

$$d^2\delta/ds^2 = 0 \qquad \text{at} \qquad s = t \tag{2.33d}$$

where t is the fin top half thickness. When Eq. (2.33) is solved subjected to the boundary conditions (Equations (2.33a) through (2.33d)) the coefficients become:

$$\begin{aligned} c_1 &= -tan(\pi/2 - \theta) \\ c_2 &= 0 \\ c_3 &= tan(\pi/2 - \theta)/2t^2 \\ c_4 &= tan(\pi/2 - \theta)/(2t)^3 \end{aligned}$$

Since the equation Eq. (2.33) represents the film thickness above the fin top and considering the coordinate system defined for the condensation calculations, c_0 is the minimum value of the film thickness right above the fin top corner where the total mass flow (green arrow that shows the liquid flow in Fig. 2.8) is equal to the total condensation mass flow obtained for the fin top. To calculate the remaining coefficient c_0, left hand side of Eq. (2.32) is integrated along the fin top:

$$m_c' = \frac{\sigma}{3\nu} 6c_0^3 c_3 \tag{2.35}$$

In order to find the T_{lv} values of the Eq. (2.15a) and Eq. (2.15b) for the solution of right hand side of Eq. (2.31), heat flux at the liquid-vapor interface above the fin top is defined as following:

$$q_c^{"} = m_c^{"} h_{lv} = k_l \frac{T_{lv} - T_w}{\delta} \tag{2.36}$$

where T_{lv} is obtained by Secant method with the following objective function obtained by the combination of the right hand side of Eq. (2.31), and Eq. (2.36):

$$f_{T_{lv}} = k_l \frac{T_{lv} - T_w}{\delta h_{lv}} + \frac{a(T_w - T_v) + b(P_l - P_v)}{1 + a\delta h_{lv}/k_l} \tag{2.37}$$

with the known value of T_{lv}, the only unknown property is c_0 before the calculation of m'_c. For this purpose, Secant method is applied with the following objective function that is formed with the integral of right hand side of Eq. (2.31), and Eq. (2.35):

$$f_{c_0} = \frac{\sigma}{3\nu} 6c_0^3 c_3 + \int_0^t \frac{a(T_w - T_v) + b(P_l - P_v)}{1 + a\delta h_{lv}/k_l} ds \tag{2.38}$$

A MATLAB function is written for the computation of phase change mass fluxes at the condensation region. For each cross-section of heat pipe, there are unique T_w and R values which are supplied to the function along with T_v, material properties and the heat pipe dimensions that are fixed for all cross-sections. In the solution process, c_0 is numerically calculated by Secant method with the corresponding objective function. During this iteration, T_{lv} is also calculated numerically with Secant method and the related objective function, for each iterated value of c_0. Once c_0 is obtained, the function outputs the micro region condensation mass line flux $m'_{c,micro}$ by using Eq. (2.33) and Eq. (2.36). The result is given as mass flow rate boundary condition for the unit cell that corresponds to the cross-section after it is multiplied with the length of the unit cell to convert the calculated $m'_{c,micro}$ to $\dot{m}_{c,micro}$.

2.4 Algorithm

The challenging nature of the heat pipe mechanism requires a complex solution methodology. This complexity is further increased with the aim of including three dimensional momentum and energy equation. In the proposed solution, the difficulty of solving both of the three dimensional equations is overcome by using a finite element method based (FEM) commercial software COMSOL Multiphysics®. Eventhough COMSOL is an answer to some major challenges that comes with the

three dimensional solution of the problem, it is certainly not enough to fulfill the requirements of the proposed solution by itself. To recall, evaporation and condensation models that calculate phase change heat and mass flows are functions of three unknown variables: R, T_w, and T_v. These values are unknown in the beginning of the solution, therefore they have to be assumed first and then iterated in a coupled manner. R is guessed linearly decreasing from condensation to evaporation region. T_v is assumed based on the previous solutions. Since $\Delta_{T_w-T_v}$ is used for both condensation and evaporation mass flow rate calculations, this difference is guessed initially instead of T_w alone. Given that T_v is equal to the wall temperature at the evaporation/condensation transition, $\Delta_{T_w-T_v}$ is assumed as two pieces intersects at $\Delta_{T_w-T_v} = 0$, as can be seen in Fig. 2.9. For this assumption, only ΔT_e is guessed, and ΔT_c is correlated as following:

$$\Delta T_c = \frac{L_{source} - 0.5L_a}{L_{sink} - 0.5L_a} \times \Delta T_e \tag{2.39}$$

where

$$L_a = L_{total} - L_{sink} - L_{source} \tag{2.40}$$

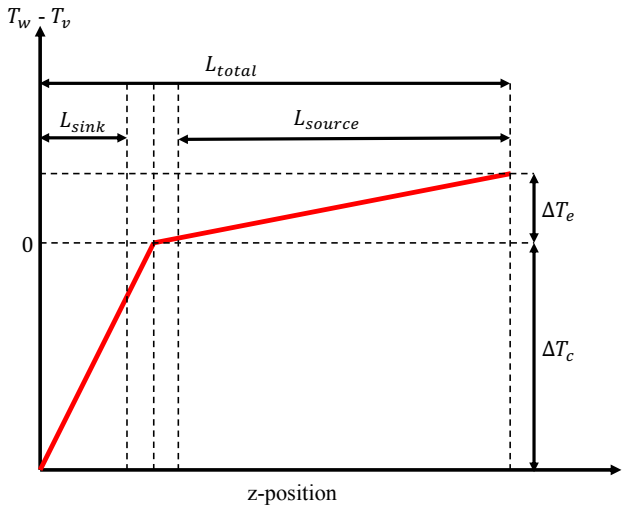


Figure 2.9: Initial guess procedure for T_w

During the iterations, the phase change fluxes that are used as boundary conditions have to be recalculated before the momentum and energy solutions due to the change in these variables. On top of it, fluid domain used for both of the solutions is generated depending on the distribution of R. Regeneration of the solution domain and recalculation of the boundary conditions in a set of iterative loops are the reasons for inadequacy of using COMSOL alone.

In order to overcome this challenge, COMSOL Multiphysics® *via* LiveLink™ for MATLAB® is utilized. LiveLink is an interface that allows to fully control and use all features of COMSOL in the scrip environment of MATLAB. Each action that can be taken in COMSOL's interface has a corresponding MATLAB code. Using these codes, COMSOL studies can be built from scratch in MATLAB with the following benefits:

- Working in a script environment: loops, conditions, cases,
- Ability to use custom functions that have complicated contents for boundary condition calculations, geometry generations, etc,
- Iterative studies: ability to obtain results from COMSOL, process and return them to COMSOL again,
- No need for COMSOL user interface,
- Ability to build any arbitrary geometry by supplying point data.

Due to this advantages, applying the geometry building procedure presented in Chapter 2 becomes possible. Moreover, evaporation and condensation functions written in MATLAB are used together with COMSOL in the script, such that the COMSOL results are provided as inputs to the functions and the functions results are sent back to COMSOL iteratively. In the following sections, the structure of this hybrid and iterative algorithm is explained in compliance with the flow chart in Fig. 2.10, which summarizes the solution procedure.

2.4.1 Momentum Solution

Momentum solution is performed in order to iterate R. The flow solution boundary conditions are stated in Chapter 2, and only the evaporation and condensation boundary conditions are subject to change at each iteration. Before the solution, these mass flows need to be calculated for a predicted Tv, Tw, and R values. Solution domain is also generated according to the initial R value. Pressure outlet BC on the last cross-section is calculated with the Young-Laplace formula:

$$R = \frac{\sigma}{P_v - P_l} \tag{2.41}$$

where P_v is the vapor pressure, σ is the surface tension and R is the radius of curvature of the cross-section. The evaporation and condensation mass flow rates are calculated with the models presented in Chapter 2. These models output m'_e and m'_c for each cross-section, which have to be multiplied with the length of the unit cell to get the micro region mass flow rate values. In order to include the macro region mass flow rates in the momentum solution, heat transfer values obtained for the macro region from the previous energy solution is used as following:

$$\dot{m} = \frac{q}{h_{lv}} \tag{2.42}$$

so that total evaporation and condensation mass flow rates become:

$$\dot{m}_{total} = \dot{m}_{macro} + \dot{m}_{micro} \tag{2.43}$$

Once the momentum equation is solved with the presented boundary conditions, pressure gradient along the axis is obtained. For each cross-section, R value is updated with Eq. (2.41) and the geometry is also regenerated accordingly. Note that this is a semi-iterative process, where the momentum equations are solved only once for each new value of T_v in order to update R.

2.4.2 Energy Iteration

Energy equation is solved in both the solid and the previously updated liquid domain. The models presented for the evaporation and condensation are also used in the energy solution and subjected to change as well. Before the solution, phase change mass flow rates are recalculated again due to the previously updated R, and implemented as convective heat transfer boundary condition, with heat transfer coefficient calculated as in Eq. (2.44) and T_v is set as T_∞:

$$h = \frac{q^{"}}{T_v - T_w} \tag{2.44}$$

Once the solution is obtained, T_w is extracted from the resulting temperature field of the solution domain. The energy solution iteration for T_w stops if the difference between the initial and new values of T_w is small enough. Otherwise, h is recalculated with Eq. (2.44) and the energy solution is repeated until convergence. Once the convergence is obtained, T_w is updated, and the up-to-date macro region heat transfer is extracted from the solution domain in order to be used in the upcoming momentum solutions by integrating surface for the normal heat flux.

Note that calculated $q^{"}$ is not used directly as heat flux boundary condition, but instead convective heat transfer boundary condition is used, due to the fact that at every iteration of T_w, calculated $q^{"}$ will change without any stopping criteria. But in the case of convective heat transfer, change in $q^{"}$ is balanced by the change in denominator which leads to the convergence of h (see Eq. (2.44)).

2.4.3 Secant Method

In the last two sequence, R and T_w values are iterated with the solutions of momentum and energy equations respectively for a given T_v value. For the iteration of T_v, the secant method is employed. In order for the combined iterations to

succeed, total evaporation and condensation heat transfer (or mass flow since Eq. (2.42) rates need to be equal to ensure conservation of mass. In the energy solution, there are four following boundaries that energy enters or leaves the solution domain:

- Heat source (Heat transfer into the system),
- Heat sink with convective heat transfer boundary condition (Heat transfer out of the system),
- micro and macro evaporative heat transfer region (Heat transfer out of the system),
- micro and macro condensative heat transfer region (Heat transfer into the system).

Since heat source is constant, once the total evaporation is equal to the total condensation, heat released by the heat sink will be equal to that of the heat source by the conservation of energy. Therefore, an objective function is used for the secant iterations as following:

$$f(T_v) = \frac{q_{out} - q_{in}}{q_{in}} \tag{2.45}$$

and by the secant method:

$$T_v^{i+1} = T_v^i - f^i \frac{T_v^i - T_v^{i-1}}{f^i - f^{i-1}} \tag{2.46}$$

note that in order for the secant method to generate a new T_v value, two previous sets of T_v and f are needed by the Eq. (2.46). Therefore, the overall iteration of the solution procedure is done separately for two different initial values of T_v before the secant iteration starts.

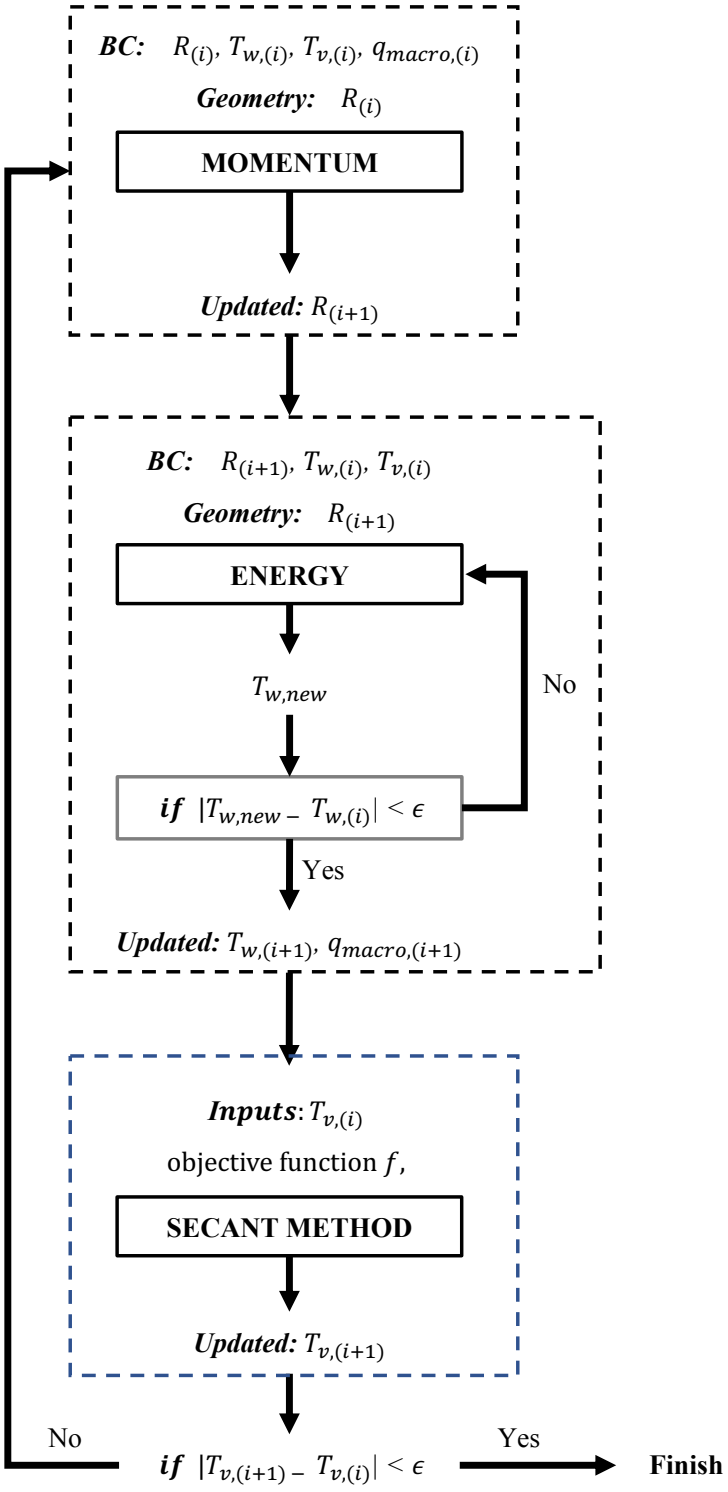


Figure 2.10: Flow chart of the model (BC and i mean phase change boundary condition and iteration index respectively.)

Chapter 3

Results & Discussion

In this chapter, the model presented in this thesis will be verified in two phases. First, the reason for the selection of boundary conditions for both the momentum and energy equations will be explained. Secondly, the result of the presented model will be compared with the study of Odabasi [15]. Then, the investigation of the effect of groove depth on the heat performance will be shown. Lastly, heat input is increased until dry-out occured, in order to obtain the maximum heat load for a given FGHP.

3.1 Validation of Mass Flow Rate BC

The phase change mechanism of the working fluid is projected as flow into and out of the fluid domain in the solution of the momentum equation. The evaporation and condensation mass flow rates are calculated by the model presented in Chapter 2. Calculated flows correspond to inlet and outlet boundary conditions for the condensation and evaporation respectively. In COMSOL, since the only choices for outlet boundary condition are velocity and pressure, fluid exit is defined as velocity boundary condition and so is the fluid entrance in order to be consistent with the exit; both are calculated as following:

$$U = \frac{\dot{m}_{pc}}{\rho A} \tag{3.1}$$

where ρ is the density of the liquid and A is the area of the boundary condition surface. When accordingly obtained channel velocity and R variation along the axis are compared with the results of Odabasi [15], a significant difference is observed. In order to understand the reason of this disparity, a strained alternative is tried, such that both of the fluid entrance and exit boundaries are defined as mass flow rate boundary condition, which is only applicable for inlet, but with negative values in the evaporation region. This approach succeeded as can be seen in Fig. 3.1 and Fig. 3.2.

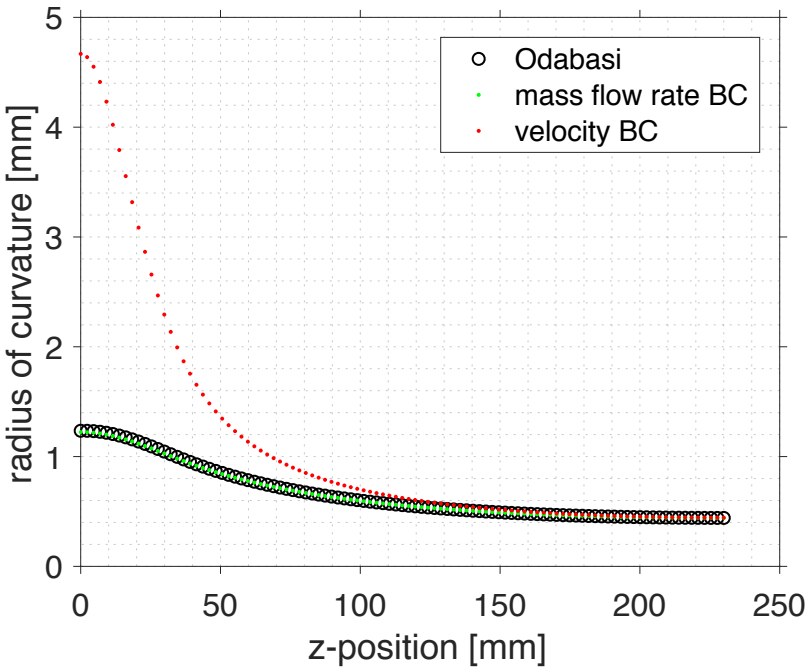


Figure 3.1: Comparison of velocity and mass flow rate boundary conditions (R)

The major difference between the velocity and mass flow rate boundary conditions is the dictation of the flow's direction in the velocity boundary condition. As it was previously told in Chapter 2.2.1, phase change flows are defined in a thin strip right under the fin top corner, which is small compare to the height of the groove wall (15-380μm), such that this boundary directly face with the liquid-vapor interface, that is defined as slip wall BC which zeroes both the shear stress

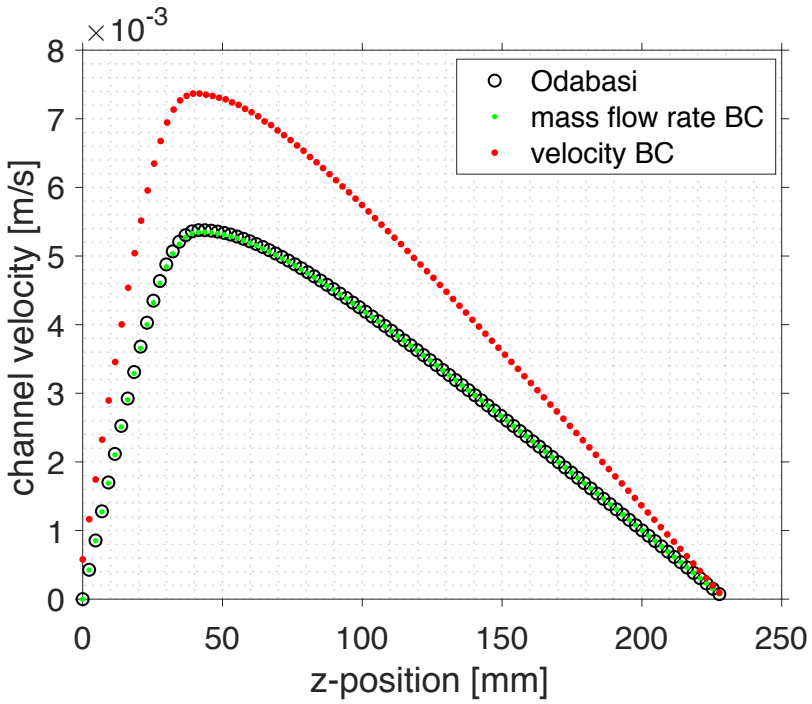


Figure 3.2: Comparison of velocity and mass flow rate boundary conditions (channel velocity)

and the normal flow. It is predicted that velocity boundary condition directing the flow into the slip wall in this very tiny region caused a compatibility issue.

3.2 Validation of Convective Heat Transfer BC Location

The introduction of the phase change mechanism into the energy solution is done with the utilization of convective heat transfer coefficient. The condensation convective heat transfer BC is defined at the fin top and a thin sheet that extends over the liquid as it was stated before in Chapter 2.2.1, eventhough it was previously mentioned in Chapter 2.3.2 that the condensation occurs alone at the fin top.

Before the final version of the proposed model, when energy equations were

solved without the extension over the liquid for the condensation BC, it was observed that the heat flux measured over the fin top in COMSOL was close to zero, and the wall temperature was equal to T_v in the same area. In other words, the effect of the convective heat flux boundary condition was neglected by COMSOL. In order to investigate this problem, a 2D domain that reflects a cross-section in the condensation region was investigated with the boundary conditions given in Fig. 3.3.

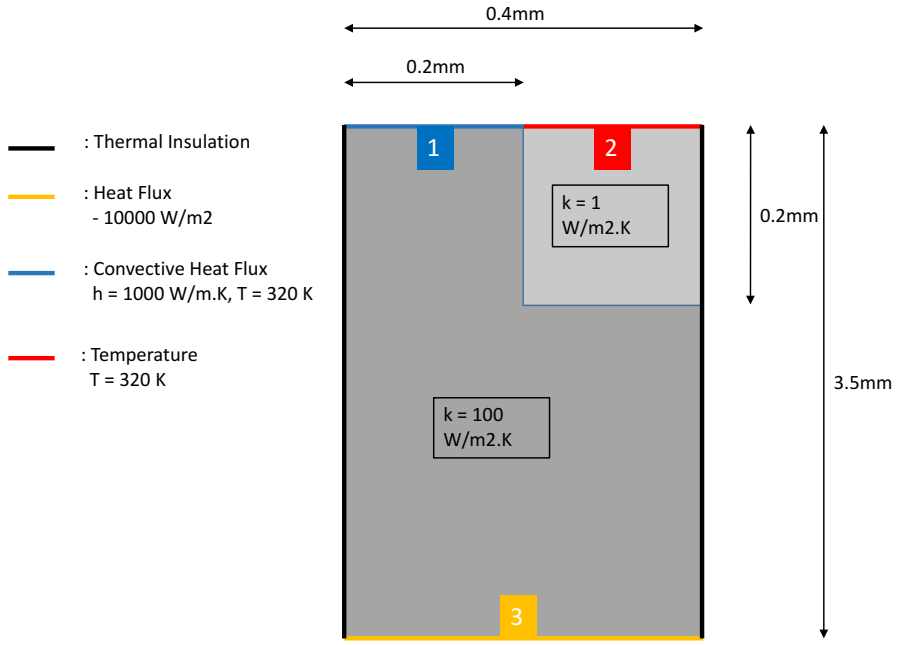


Figure 3.3: Boundary condition for COMSOL / ANSYS comparison

The 2D model was solved both in COMSOL and ANSYS, and normal total heat flux values are integrated over the walls that are tagged as $1, 2, 3$ in Fig. 3.3. This comparison gave some more than significant differences as can be seen in Table 3.1.

These results confirmed that COMSOL experienced some problems with the current BC's, which is interpreted as following:

- Convective BC, which is defined at the Wall 1 has a finite h value that is 1000 W/m.K, whereas Wall 2 has temperature boundary condition that

[Watt]	**COMSOL**	**ANSYS**
Wall 1	-0.0037	-0.2366
Wall 2	-3.9963	-3.7624
Wall 3	4.0000	4.0000
Net:	0	0

Table 3.1: COMSOL/ANSYS comparison results

also means h is infinite at Wall 2 in order to fix the temperature of Wall 2 to T_v.

- At the intersection point between Wall 1 and Wall 2, there is also a transition from the solid to the liquid domains, where thermal conductivity of the solid is 100 times higher than of the liquid.

as briefly explained above, there are two jumps in the intersection point of the Wall 1 and Wall 2; heat transfer coefficient h which is much bigger in the liquid region, and the thermal conductivity k which is much bigger in the solid region. It is suspected that the point having two opposite jumps creats a singularity that causes the problem. To solve this, an extension of the convective heat transfer boundary over the liquid domain is introduced in order to seperate the two jumps as it can be seen in Fig. 3.4:

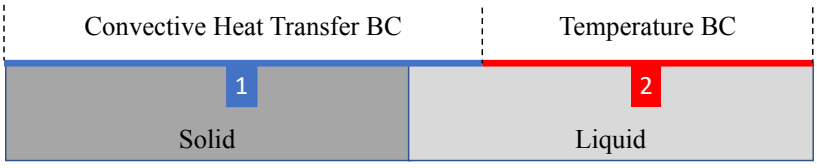


Figure 3.4: Modification for heat transfer BC of condensation

The effect of this proposed increase in the convective heat flux BC of condensation area worked, and can be seen in Table 3.2. The reason for this singularity happens in COMSOL but not in ANSYS is believed to be the different numerical methods employed by both of the softwares: FEM and FVM respectively. Of course,

this change also introduces a new question about the amount of this extension. Since this change is done only to break the singularity, setting the width of the extension as small as possible is the logical choice in order to keep the original problem intact.

[Watt]	**COMSOL (Initial)**	**COMSOL (Modified)**	**ANSYS**
Wall 1	-0.0037	-0.2441	-0.2366
Wall 2	-3.9963	-3.7559	-3.7624
Wall 3	4.0000	4.0000	4.0000
Net:	0	0	0

Table 3.2: COMSOL/ANSYS comparison results with modified BC

3.3 Comparison of Overall Model with Literature

So far, the reason for not using velocity boundary condition in the momentum solutions is discussed and the mass flow rate BC is verified. This is followed by the verification of the disputable areal modification done for the convective heat flux BC (conduction). In this part, the final results of the overall model will be compared with the study of Odabasi. The dimensions and geometries that are used for this comparison are given in Table 3.3.

In the following figures, the results of the algorithm are presented. Momentum and energy equations are solved with $1,348,160$ and $3.757.403$ number of elements respectively, and *quadratic element* mesh order is used for both. Accordingly:

- Momentum results that includes the axial channel velocity and R are shown in Fig. 3.6 and Fig. 3.7. As can be seen, the results of the presented model

is consistent with that of Odabasi [15] with $\sim 1\%$ difference. According to this result, effect of solving 3D full Navier-Stoke's equations made little difference. Nevertheless, this verification strengthen the potential of using COMSOL for other kinds of groove profiles, where the three dimensional effects of the flow might matter.

- The comparison of the results obtained for the Tw and Tv with Odabasi's study are presented in Fig. 3.8 and Fig. 3.9. Since ΔT_{w-v} gives efficiency of a heat pipe, Fig. 3.8 is plotted accordingly. In the results, there is $\sim 3\%$ error in the condensation regions, where as it is $\sim 13\%$ in the evaporation region. Compared to the momentum solutions, the difference between the results are higher in this case. The following are presented as potential reasons for this difference:

  1. The modified phase change heat transfer boundary explained in Chapter 3.2 should be considered. In Odabasi's study, there is no such extension of condensation heat flux boundary from solid to liquid domain. In Table 3.2, a slight difference between the modified COMSOL and ANSYS results can be observed. Such small differences might have been built up to give a larger difference.
  2. For this study, COMSOL's default mesh generator is utilized. Since the fluid geometry is comprised of many cross-sections, the mesh quality concentrated on the the intersections of the unit cells with 200 μm width, but not on the evaporation heat flux boundary of 5 μm width, which caused less accurate temperature results.

- The T_v iteration with secant method is presented in Fig. 3.5. It can be seen that the overall iteration took ten steps to achieve 0.1 K error in T_v results, where the duration is 10 hrs, 31mins.

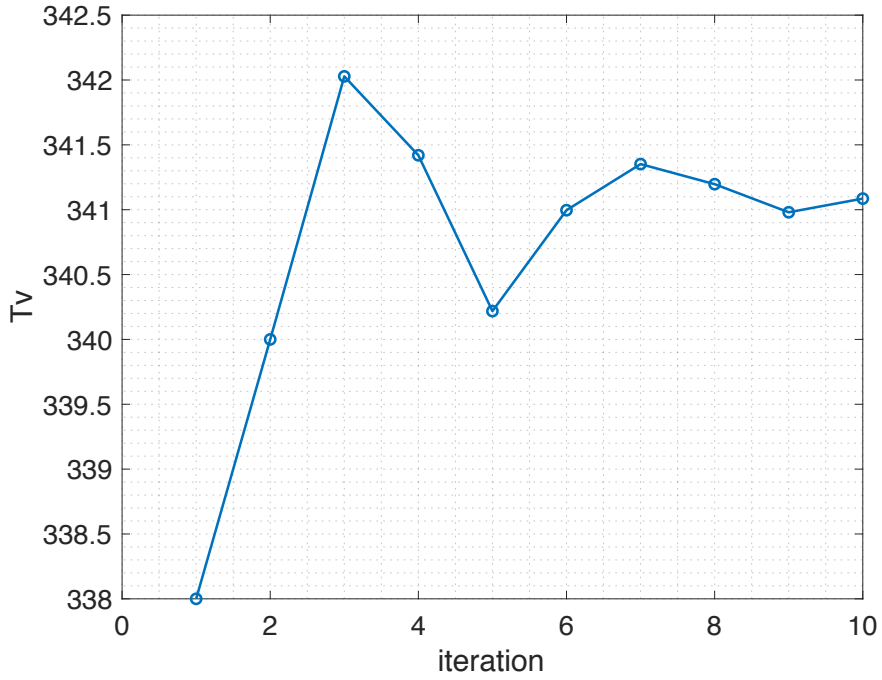


Figure 3.5: Tv iteration

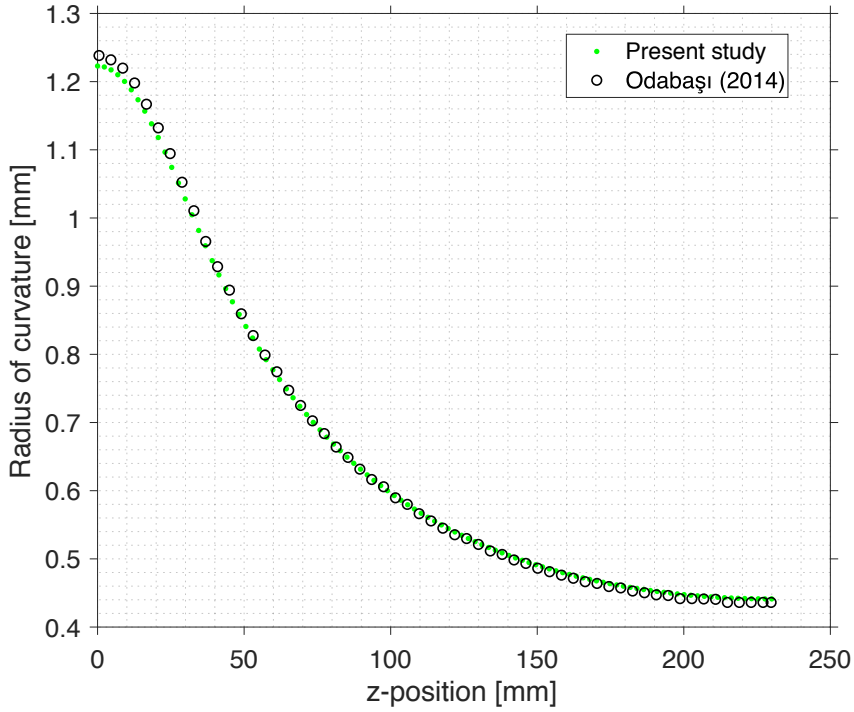


Figure 3.6: Radius of curvature verification

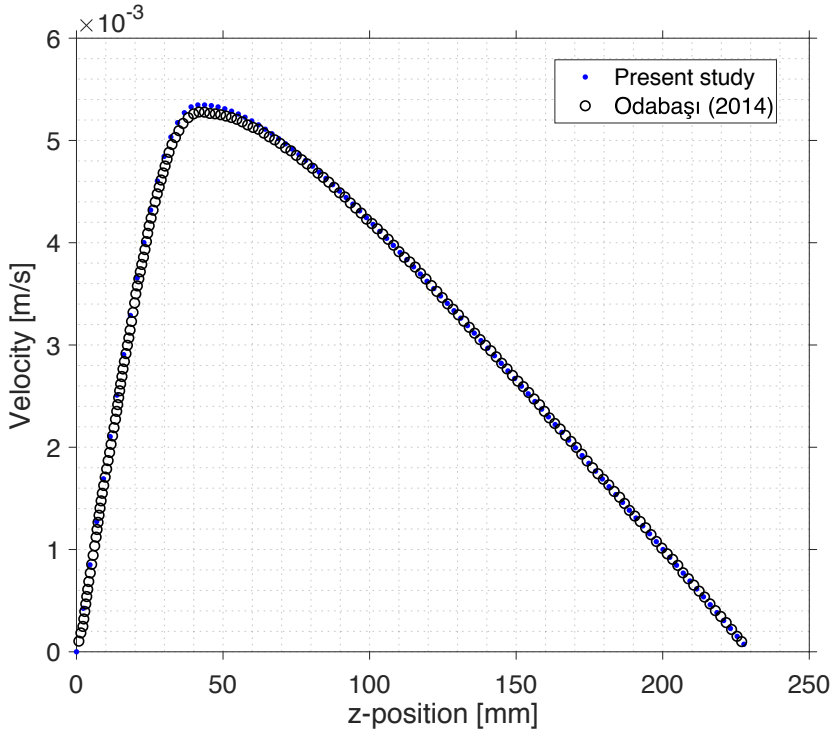


Figure 3.7: Channel velocity verification

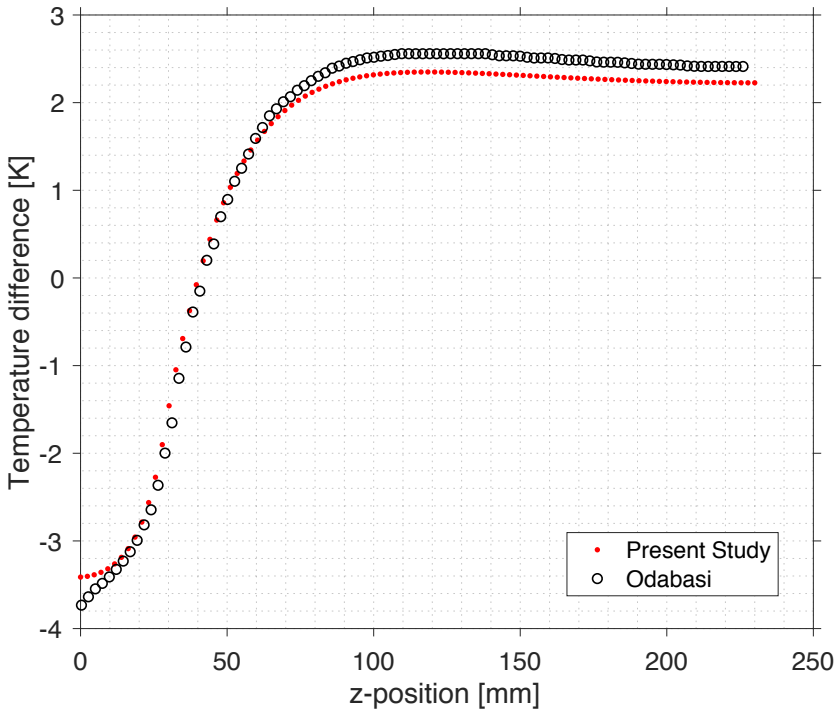


Figure 3.8: Wall temperature verification

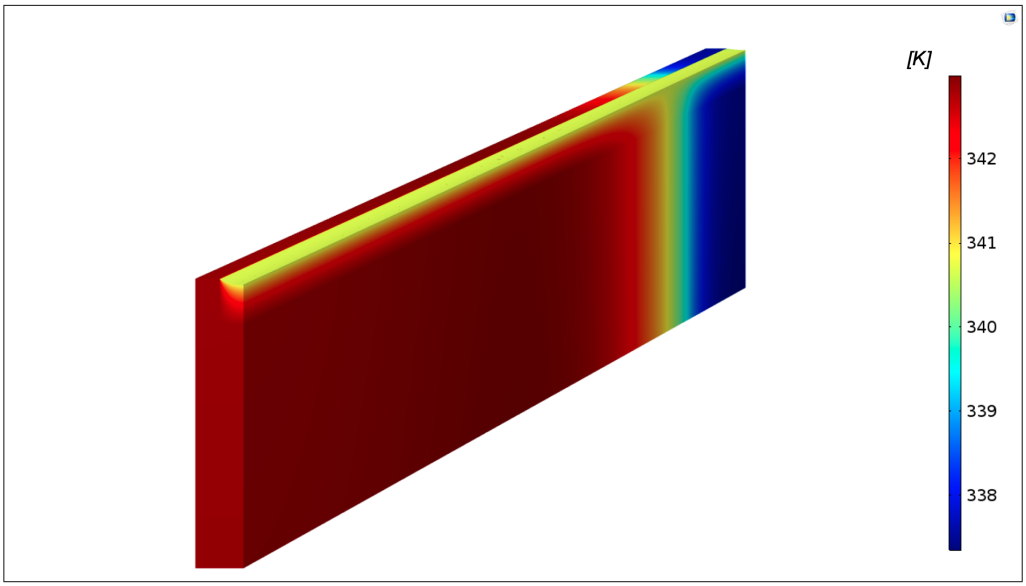


Figure 3.9: 3D temperature field

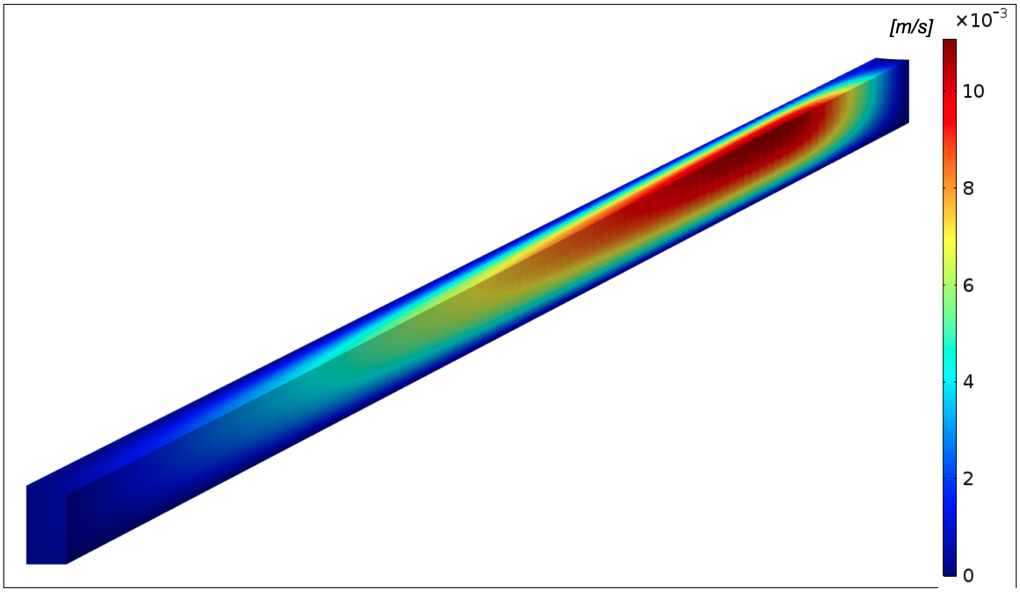


Figure 3.10: 3D velocity field

Fin top width (half)	0.2	mm
Channel width (half)	0.2	mm
Base height	2	mm
Channel height	0.38	mm
Channel length	230	mm
Length of heat source region	190	mm
Length of heat sink region	30	mm
Thermal conductivity (Solid)	400	$W/m.K$
Thermal conductivity (Liquid)	0.2	$W/m.K$
Vapor pressure	1.31×10^{-5}	Pa
Surface tension	18.5×10^{-3}	N/m
Latent heat of evaporation	1085×10^{3}	J/kg
Molar mass	32×10^{-3}	kg/mol
Molar volume	42×10^{-6}	m^3/mol
Density	792	kg/m^3
Dynamic Viscosity	3.14×10^{-4}	$Pa.s$
Dispersion constant	2×10^{-21}	-

Table 3.3: Material properties and dimensional parameters used for the parametric study

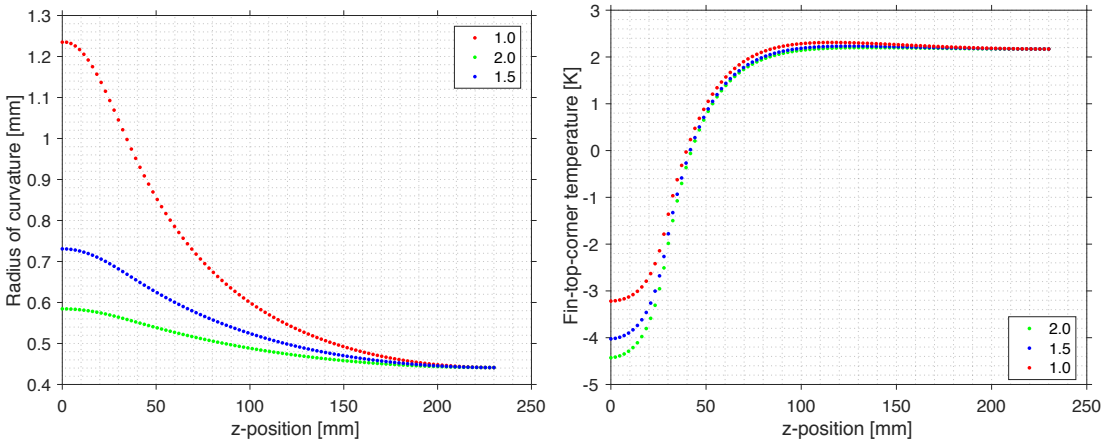
3.4 Parametric Study for Groove Depth

In this section, the previously compared model is used for investigating the effects of groove depth on the thermal performance of a FGHP. Used for dissipating a given heat source, the higher the effective thermal conductivity of a FGHP is, the lower the temperature difference will be, therefore the temperature gradient along the grooves can be considered as a performance criteria. For this investigation, the values given in Table 3.3 are used. Channel width of $380\mu m$ is used as baseline, and two alternative aspect ratios are defined at magnitudes of 1.5 and 2.0 linearly.

The R result of this comparison can be seen in Fig. 3.11a. It can be observed that the channel depth has an inversely and exponentially proportional relation with the pressure distribution in the groove if one consider the linearity of the aspect ratios. From the result in Fig. 3.11b it can be said that T_w continues the exponential behavior of R against the groove depth. From Fig. 3.11c, it can be observed that the change in mass flow rate is not significant for different groove depths. Lastly, channel velocity also decreases with the increasing groove width as it can be seen in Fig. 3.11d. From these figures following remarks can be made:

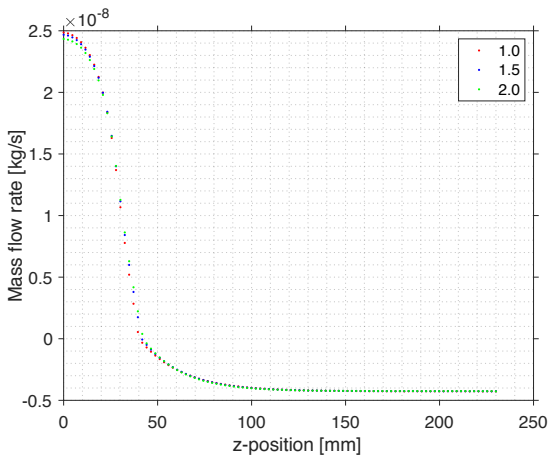
- Phase change mass fluxes are not strong functions of groove depth.
- Channel velocity decreases with the increasing depth, which is expected due to the non-significant change in mass flow rates, and increasing cross-sectional area. A negative impact of the groove depth can be observed here, such that lower velocities will start struggling to sustain the necessary mass flow to the evaporation region considering the entraintment limit.
- With the increasing depth, ΔR along the axis decreases, which means the pressure drop across the groove become smaller by Eq.(2.41). The ΔP is created by the phase change mass fluxes going in and out, therefore it can be said that the phase change fluxes become less effective at higher depths to create the necessary pressure difference. This indicates a negative impact considering the capillary limit.

- ΔT_w along the axis increases with the increasing groove width, which indicates a decrease in effective thermal conductivity as explained in the beginning of this section. In conclusion, increased groove depth affect the thermal performance of flat grooved heat pipe adversely considering the results obtained.

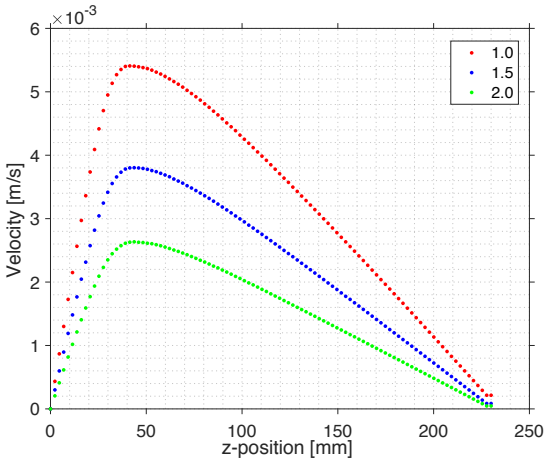


(a) R comparison (b) T_w comparison

Figure 3.11: Effect of groove width on various parameters



(c) Mass flow rate comparison

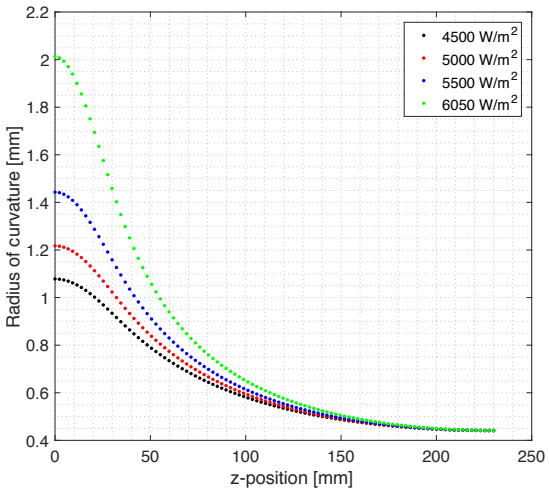


(d) Channel velocity comparison

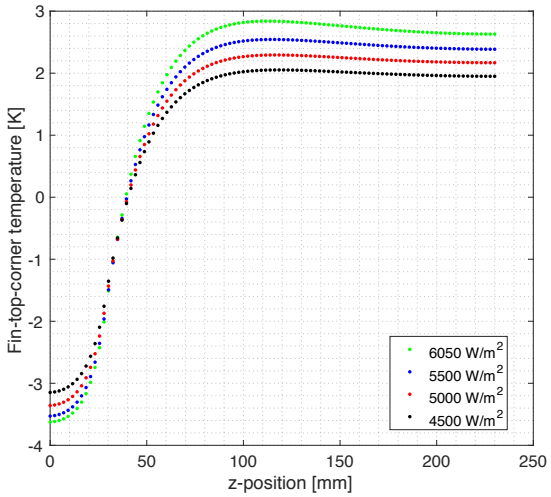
Figure 3.11: Effect of groove width on various parameters

3.5 Dry-out Investigation

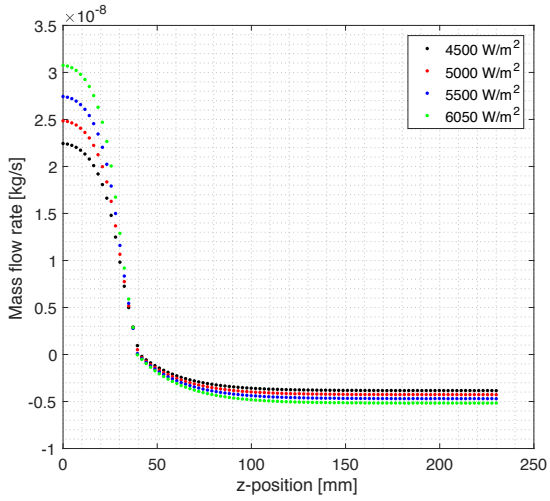
In this section, results for different heat loads are presented. For this purpose, heat load is increased up to a such value that convergence could not achieved. With the increase in heat load, mass flow rate values also increased as can be seen in Fig. 3.12c. Considering the calculation of the radius of curvature from the Young-Laplace equation Eq.(2.41), higher amounts of mass fluxes caused higher pressure difference along the channel as in Fig. 3.12a with the increasing heat load. In the case of not achieving convergence, higher heat load created excessive pressure values that caused P_l to be higher than P_v, which leads to a negative R value according to Young-Laplace equation. According to the results presented in Fig. 3.12, ∼6050 W/m^2 is the maximum heat load for this problem, and shows the maximum heat transfer capacity of the FGHP with the sustained temperature difference given in Fig. 3.12b. It should also be noted that increased heat load has a negative impact on the thermal performance of flat grooved heat pipes.



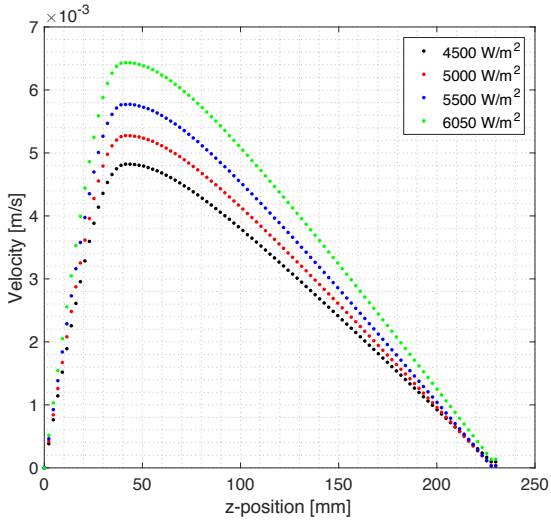
(a) R comparison



(b) T_w comparison



(c) Mass flow rate comparison



(d) Channel velocity comparison

Figure 3.12: Effect of groove width on various parameters

3.6 Conclusion & Future Studies

Heat pipes are passive containers that can transfer large amount of thermal energy with small temperature differences when placed between a heat source and sink. This high performance is due to the utilization of the continuous phase change mechanism of the working fluid contained in heat pipes. There are many operational limits -which depend on the amount and type of the working fluid, shape and number of grooves, geometry and material of heat pipe base, and environmental conditions- that affects the performance and ability to work of heat pipes. In order to design a case specific heat pipe with maximum efficiency, there are many proposed mathematical models in the literature. The mathematical model presented in this thesis is prominent due to the modeling the heat transfer and fluid flow in three-dimensions by benefiting from the ability of controlling a commercial CFD software in a script environment.

As for the results, the proposed mathematical model is compared with the results of Odabasi's study [15]. First, the unique boundary conditions that are used in this model are verified. Then the overall results of the model is validated by comparing it with the literature. With the validated model, three different channel heights are investigated for their effect on the thermal performance of a FGHP, and it was found that the increasing groove depth effects the FGHP performance negatively. Lastly, heat load is increased up for observing the dry-out.

Indeed, there is room for improvement and modification in the proposed model. The following list can be considered for the future directions of the study, considering the extremely modular and upgradeable structure of the model:

- Initial guesses for T_v, T_w, and R have an important role in the solution time and convergency. Depending on the results of the presented mathematical model, a correlation or an algorithm can be studied to produce more accurate and stable initial guesses.
- For both of the solution for momentum and energy equations, COMSOL's

built-in auto mesh feature is used. Only the quality of the mesh and the element order are introduced as user parameters into the MATLAB script. A fine study on optimizing the meshing of the solution domains is needed in order to increase the accuracy and decrease the solution time.

- COMSOL includes many features that can be introduced to the presented mathematical model, such as the ability of using obtained flow field in the energy solution, and vice versa by using the temperature field in the momentum solutions. Shear stress at the liquid-vapor boundary can also be introduced by using COMSOL's specialized boundary conditions. The effects of these changes on the overall results can be a part of a study.

- As it was discussed in Chapter 2.1, generating heat pipe geometry is done by joining 2D axial cross-sections of the heat pipe. This enables to obtain various channel geometries without trouble. A study on the effect of channel geometry on heat pipe performance is also valuable.

- It was stated that secant method is utilized for the iteration of T_v. Effects of the other methods on the stability and convergency rate can be discussed.

Bibliography

[1] R. C. Pfahl and J. McElroy, "The 2004 International Electronics Manufacturing Initiative (iNEMI) technology roadmaps," *2005 Conference on High Density Microsystem Design and Packaging and Component Failure Analysis, HDP'05*, 2006.

[2] B. Agostini, M. Fabbri, J. E. Park, L. Wojtan, J. R. Thome, and B. Michel, "State of the art of high heat flux cooling technologies," *Heat Transfer Engineering*, vol. 28, no. 4, pp. 258–281, 2007.

[3] P. Chen, S. Chang, K. Chiang, and J. Li, "High Power Electronic Component: Review," *Recent Patents on Engineering*, vol. 2, no. 3, pp. 174–188, 2008.

[4] S. M. Sohel Murshed and C. A. Nieto de Castro, "A critical review of traditional and emerging techniques and fluids for electronics cooling," *Renewable and Sustainable Energy Reviews*, vol. 78, no. May, pp. 821–833, 2017.

[5] M. C. Riofrío, N. Caney, and J. A. Gruss, "State of the art of efficient pumped two-phase flow cooling technologies," *Applied Thermal Engineering*, vol. 104, pp. 333–343, 2016.

[6] A. W. Scott, *Cooling of electronic equipment*. UMI Out-of-Print Books on Demand, 1991.

[7] H. N. Chaudhry, B. R. Hughes, and S. A. Ghani, "A review of heat pipe systems for heat recovery and renewable energy applications," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 4, pp. 2249–2259, 2012.

[8] A. Faghri, "Review and Advances in Heat Pipe Science and Technology," *Journal of Heat Transfer*, vol. 134, no. 12, p. 123001, 2012.

[9] K. N. Shukla, "Heat Pipe for Aerospace Applications?An Overview," *Journal of Electronics Cooling and Thermal Control*, vol. 5, no. 5, pp. 1–14, 2015.

[10] A. C. Kheirabadi and D. Groulx, "Cooling of server electronics: A design review of existing technology," *Applied Thermal Engineering*, vol. 105, pp. 622–638, 2016.

[11] H. N. Chaudhry, B. R. Hughes, and S. A. Ghani, "A review of heat pipe systems for heat recovery and renewable energy applications," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 4, pp. 2249–2259, 2012.

[12] T. Cotter, "Principles and prospects for micro heat pipes," *Proceedings of 7th International Heat Pipe Conference1*, vol. Vol 1, pp. 328–335, 1984.

[13] X. Chen, H. Ye, X. Fan, T. Ren, and G. Zhang, "A review of small heat pipes for electronics," *Applied Thermal Engineering*, vol. 96, pp. 1–17, 2016.

[14] J. Qu, H. Wu, P. Cheng, Q. Wang, and Q. Sun, "International Journal of Heat and Mass Transfer Recent advances in MEMS-based micro heat pipes," *International Journal of Heat and Mass Transfer*, vol. 110, pp. 294–313, 2017.

[15] G. Odabasi, "Modelling of multidimensional heat transfer in a rectengular grooved heat pipe," June 2014.

[16] Y. M. Hung and Q. Seng, "Effects of geometric design on thermal performance of star-groove micro-heat pipes," *International Journal of Heat and Mass Transfer*, vol. 54, no. 5-6, pp. 1198–1209, 2011.

[17] X. Yang, Y. Y. Yan, and D. Mullen, "Recent developments of lightweight, high performance heat pipes," *Applied Thermal Engineering*, vol. 33-34, no. 1, pp. 1–14, 2012.

[18] H. Jouhara, A. Chauhan, T. Nannou, S. Almahmoud, and B. Delpech, "Heat pipe based systems - Advances and applications," *Energy*, vol. 128, pp. 729–754, 2017.

[19] J. B. Marcinichen, J. A. Olivier, N. Lamaison, and J. R. Thome, "Advances in electronics cooling," *Heat Transfer Engineering*, vol. 34, no. 5-6, pp. 434–446, 2013.

[20] M. Shafahi, V. Bianco, K. Vafai, and O. Manca, "Thermal performance of flat-shaped heat pipes using nanofluids," *International Journal of Heat and Mass Transfer*, vol. 53, pp. 1438–1445, 03 2010.

[21] K. H. Do and S. P. Jang, "Effect of nanofluids on the thermal performance of a flat micro heat pipe with a rectangular grooved wick," *International Journal of Heat and Mass Transfer*, vol. 53, no. 9-10, pp. 2183–2192, 2010.

[22] M. Sigurdson, Y. Liu, P. Bozorgi, D. Bothman, N. MacDonald, and C. Meinhart, "A large scale Titanium Thermal Ground Plane," *International Journal of Heat and Mass Transfer*, vol. 62, no. 1, pp. 178–183, 2013.

[23] P. Nemec, A. Caja, and M. Malcho, "Mathematical model for heat transfer limitations of heat pipe," *Mathematical and Computer Modelling - MATH COMPUT MODELLING*, vol. 57, 01 2013.

[24] G. P. Babin, B. R.; Peterson and D. Wu, "Steady-State Modeling and Testing of a Micro Heat Pipe," vol. 112, no. August, pp. 595–601, 1990.

[25] D. Khrustalev and A. Faghri, "Thermal Analysis of a Micro Heat Pipe," vol. 116, no. FEBRUARY 1994, 1994.

[26] A. A. El-Nasr and S. M. El-Haggar, "Effective thermal conductivity of heat pipes," *Heat and Mass Transfer*, vol. 32, no. 1-2, pp. 97–101, 1996.

[27] G. P. Peterson and H. B. Ma, "Theoretical analysis of the maximum heat transport in triangular grooves: A study of idealized micro heat pipes," *Journal of Heat Transfer*, vol. 118, no. 3, pp. 731–739, 1996.

[28] S. Anand, S. De, and S. Dasgupta, "Experimental and theoretical study of axial dryout point for evaporation from V-shapes microgrooves," *International Journal of Heat and Mass Transfer*, vol. 45, no. 7, pp. 1535–1543, 2002.

[29] S. J. Kim, J. K. Seo, and K. H. Do, "Analytical and experimental investigation on the operational characteristics and the thermal optimization of a miniature heat pipe with a grooved wick structure," *International Journal of Heat and Mass Transfer*, vol. 46, no. 11, pp. 2051–2063, 2003.

[30] S. Launay, V. Sartre, and M. Lallemand, "Hydrodynamic and thermal study of a water-filled micro-heat-pipe array," *Journal of Thermophysics and Heat Transfer*, vol. 18, no. 3, pp. 358–363, 2004.

[31] B. Suman, S. De, and S. DasGupta, "A model of the capillary limit of a micro heat pipe and prediction of the dry-out length," *International Journal of Heat and Fluid Flow*, vol. 26, no. 3, pp. 495–505, 2005.

[32] B. Suman and N. Hoda, "Effect of variations in thermophysical properties and design parameters on the performance of a V-shaped micro grooved heat pipe," *International Journal of Heat and Mass Transfer*, vol. 48, no. 10, pp. 2090–2101, 2005.

[33] B. Suman, S. De, and S. DasGupta, "Transient modeling of micro-grooved heat pipe," *International Journal of Heat and Mass Transfer*, vol. 48, no. 8, pp. 1633–1646, 2005.

[34] F. Lefèvre and M. Lallemand, "Coupled thermal and hydrodynamic models of flat micro heat pipes for the cooling of multiple electronic components," *International Journal of Heat and Mass Transfer*, vol. 49, no. 7-8, pp. 1375–1383, 2006.

[35] K. H. Do, S. J. Kim, and S. V. Garimella, "A mathematical model for analyzing the thermal characteristics of a flat micro heat pipe with a grooved wick," 2008.

[36] K. Hyung, S. Jin, and S. V. Garimella, "A mathematical model for analyzing the thermal characteristics of a flat micro heat pipe with a grooved wick," vol. 51, pp. 4637–4650, 2008.

[37] B. Xiao and A. Faghri, "A three-dimensional thermal-fluid analysis of flat heat pipes," *International Journal of Heat and Mass Transfer*, vol. 51, no. 11-12, pp. 3113–3126, 2008.

[38] J. Qu, H. Wu, and P. Cheng, "Effects of functional surface on performance of a micro heat pipe," *International Communications in Heat and Mass Transfer*, vol. 35, no. 5, pp. 523–528, 2008.

[39] F. Lefèvre, R. Rullière, G. Pandraud, and M. Lallemand, "Prediction of the temperature field in flat plate heat pipes with micro-grooves - Experimental validation," *International Journal of Heat and Mass Transfer*, vol. 51, no. 15-16, pp. 4083–4094, 2008.

[40] R. Sonan, S. Harmand, J. Pellé, D. Leger, and M. Fakès, "Transient thermal and hydrodynamic model of flat heat pipe for the cooling of electronics components," *International Journal of Heat and Mass Transfer*, vol. 51, no. 25-26, pp. 6006–6017, 2008.

[41] M. Aghvami and A. Faghri, "Analysis of flat heat pipes with various heating and cooling configurations," *Applied Thermal Engineering*, vol. 31, no. 14-15, pp. 2645–2655, 2011.

[42] Y. M. Hung and K. K. Tio, "Thermal analysis of optimally designed inclined micro heat pipes with axial solid wall conduction," *International Communications in Heat and Mass Transfer*, vol. 39, no. 8, pp. 1146–1153, 2012.

[43] N. Thuchayapong, A. Nakano, P. Sakulchangsatjatai, and P. Terdtoon, "Effect of capillary pressure on performance of a heat pipe: Numerical approach with FEM," *Applied Thermal Engineering*, vol. 32, no. 1, pp. 93–99, 2012.

[44] N. Chauris, V. Ayel, Y. Bertin, C. Romestant, and D. Eysseric, "Hydraulic modelling of a flat heat pipe with two different groove shapes and a small vapour section," *Applied Thermal Engineering*, vol. 61, no. 2, pp. 311–326, 2013.

[45] F. L. Chang and Y. M. Hung, "The coupled effects of working fluid and solid wall on thermal performance of micro heat pipes," *International Journal of Heat and Mass Transfer*, vol. 73, pp. 76–87, 2014.

[46] P. C. Stephan and C. A. Busse, "Analysis of the heat transfer coefficient of grooved heat pipe evaporator walls," *International Journal of Heat and Mass Transfer*, vol. 35, no. 2, pp. 383–391, 1992.

Appendix A

LiveLink Code

```
function out = HP_CFD_v19

clc; clear; close all; tic;

disp('HP_CFD_final')

fprintf('\n* Date/Time: %s \n',datetime);

%% user defined values (
D1 = 1;  D1i = 1;         % cond. pc ht (1.18)
D2 = 1;    D2i = 1;         % evap. pc ht (2.125)
cdamp = 0;                  % Tv iteratýon damper coefficient
ite_max = 20;               % maximum iteration number
C1 = 1.1;                     % R initial guess amplification constant

%% BC
q_in = 5000;
htc = 2100;
Tamb = 323;

%% mesh
eo1 = 2;                   % element order (spf, P2+P2->4 )
eo2 = 2;                     % element order (ht, default->2 )
mq_spf = 6;                     % auto mesh quality (1.fine --> 8.course)
mq_ht = 5;
epsi = 1e-6;                % boundary/domain selection sensitivity
nx = 20;                    % number of points in x-direction
nz = 101;                   % number of cross-section

%% geometric parameters (in microns)
w1 = 200*1e-6;              % half-width of fin
```

```
w2 = 200*1e-6;              % half-width of groove
w3_e = 6*1e-6;              % insulating sheet cond
w3_c = 7*1e-6;              % insulating sheet evap
h1 = 2*380*1e-6;            % depth of microchannel
h2 = (2000-380)*1e-6;       % height of hp base
hmf = 15*1e-6;              % height of mass flow region
lt = 230000*1e-6;           % total length
lc = 30000*1e-6;            % cooler region length
lh = 190000*1e-6;           % heater region length

%% initial assumptions

c0 = 5*1e-6;                % c0
Tv(1) = 341.159;
Tv(2) = 341.2;

R = linspace(1500e-6,450e-6,nz);

la = lt-lh-lc;
nzc = round((nz-1)*lc/(lt-la));
nzh = round((nz-1)*lh/(lt-la));
Tw_min = -4;
Tw_max = abs(Tw_min)*lc/lh;
Tw = Tv(1) + [linspace(Tw_min,-0.1,nzc) linspace(0.1,Tw_max,nzh)];
Tw_ft = Tw;

%% error
err_ht = 0.03;              % error of DTw
err_Tv = 0.05;              % difference error of Tv

%% material properties
% fluid properties (water)
Pv = 1.31e5;                % vapor pressure (@40 celcius)
sft = 18.5e-3;              % surface tension
hlv = 1085e3;               % latent heat evaporation
M = 32e-3;                  % molar mass
Vl = 42e-6;                 % molar volume
Ru = 8.3145;                % universal gas constant
kl = 0.2;                   % thermal conductivity
rho = 792;                  % density
dvis = 3.14e-4;             % dynamic viscoity
kvis = dvis/rho;            % kinematic viscosity
A_const = 2e-21;            % dispersion constant (Joule Do et al water on copper) 21
cp = 2530;                  % specific heat
gamma = 1;                  % ratio of specific heats

% solid properties (aluminum)
cp_s = 910;                 % specific heat
rho_s = 2329;               % density
ks = 400;                   % thermal conductivity
```

```
%% initial calculations
dz = lt/(nz-1);             % length of unit cell
P = Pv - sft./R;            % pressure along axis[Pa]
P_fe = P(end);
% matrix initializing
yarc_end = zeros(1,nz);
yarc_start = zeros(1,nz);
P_new = zeros(1,nz);
q_mac = zeros(1,nz-1);      % macro region heat flux

%% initialization (spf & ht)

% model
import com.comsol.model.*
import com.comsol.model.util.*
model = ModelUtil.create('Model');
model.modelNode.create('comp1');

% geometry
model.geom.create('geom1', 3);
model.geom('geom1').lengthUnit('m');

% physics
model.physics.create('spf', 'LaminarFlow', 'geom1');
model.physics('spf').prop('PhysicalModelProperty').set('Compressibility', 'Incompressible');
% model.physics('spf').prop('PhysicalModelProperty').setIndex('StokesFlowProp', '1', 0);
model.physics('spf').prop('ShapeProperty').set('order_fluid', num2str(eo1));
model.physics.create('ht', 'HeatTransfer', 'geom1');
model.physics('ht').prop('ShapeProperty').set('order_temperature', num2str(eo2));
model.component('comp1').physics('ht').prop('ShapeProperty').set('boundaryFlux_temperature', true);

% creating study (spf)
model.study.create('std1');
model.study('std1').create('stat', 'Stationary');
model.study('std1').feature('stat').activate('spf', true);
model.study('std1').feature('stat').activate('ht', false);
model.study('std1').feature('stat').set('notlistsolnum', 1);
model.study('std1').feature('stat').set('notsolnum', '1');
model.study('std1').feature('stat').set('listsolnum', 1);
model.study('std1').feature('stat').set('solnum', '1');

% creating solver (spf)
model.sol.create('sol1');
model.sol('sol1').study('std1');
model.sol('sol1').create('st1', 'StudyStep');
model.sol('sol1').feature('st1').set('study', 'std1');
model.sol('sol1').feature('st1').set('studystep', 'stat');
model.sol('sol1').create('v1', 'Variables');
model.sol('sol1').feature('v1').set('control', 'stat');
```

```
model.sol('sol1').create('s1', 'Stationary');
model.sol('sol1').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol1').feature('s1').feature('fc1').set('initstep', 0.01);
model.sol('sol1').feature('s1').feature('fc1').set('minstep', 1.0E-6);
model.sol('sol1').feature('s1').feature('fc1').set('dtech', 'auto');
model.sol('sol1').feature('s1').feature('fc1').set('maxiter', 25);
model.sol('sol1').feature('s1').create('d1', 'Direct');
model.sol('sol1').feature('s1').feature('d1').set('linsolver', 'pardiso');
model.sol('sol1').feature('s1').feature('fc1').set('linsolver', 'd1');
model.sol('sol1').feature('s1').feature('fc1').set('initstep', 0.01);
model.sol('sol1').feature('s1').feature('fc1').set('minstep', 1.0E-6);
model.sol('sol1').feature('s1').feature('fc1').set('dtech', 'auto');
model.sol('sol1').feature('s1').feature('fc1').set('maxiter', 25);
model.sol('sol1').feature('s1').feature.remove('fcDef');
model.sol('sol1').attach('std1');

% creating study (ht)
model.study.create('std2');
model.study('std2').create('stat', 'Stationary');
model.study('std2').feature('stat').activate('ht', true);
model.study('std2').feature('stat').activate('spf', false);
model.study('std2').feature('stat').set('notlistsolnum', 1);
model.study('std2').feature('stat').set('notsolnum', '1');
model.study('std2').feature('stat').set('listsolnum', 1);
model.study('std2').feature('stat').set('solnum', '1');

% creating solver (ht)
model.sol.create('sol2');
model.sol('sol2').study('std2');
model.sol('sol2').create('st1', 'StudyStep');
model.sol('sol2').feature('st1').set('study', 'std2');
model.sol('sol2').feature('st1').set('studystep', 'stat');
model.sol('sol2').create('v1', 'Variables');
model.sol('sol2').feature('v1').set('control', 'stat');
model.sol('sol2').create('s1', 'Stationary');
model.sol('sol2').feature('s1').create('fc1', 'FullyCoupled');
model.sol('sol2').feature('s1').feature('fc1').set('dtech', 'auto');
model.sol('sol2').feature('s1').feature('fc1').set('initstep', 0.01);
model.sol('sol2').feature('s1').feature('fc1').set('minstep', 1.0E-6);
model.sol('sol2').feature('s1').feature('fc1').set('maxiter', 50);
model.sol('sol2').feature('s1').create('d1', 'Direct');
model.sol('sol2').feature('s1').feature('d1').set('linsolver', 'pardiso');
model.sol('sol2').feature('s1').feature('fc1').set('linsolver', 'd1');
model.sol('sol2').feature('s1').feature('fc1').set('dtech', 'auto');
model.sol('sol2').feature('s1').feature('fc1').set('initstep', 0.01);
model.sol('sol2').feature('s1').feature('fc1').set('minstep', 1.0E-6);
model.sol('sol2').feature('s1').feature('fc1').set('maxiter', 50);
model.sol('sol2').feature('s1').feature.remove('fcDef');
model.sol('sol2').attach('std2');
```

```
% creating mesh (spf&ht)
model.mesh.create('mesh1', 'geom1');

% material properties
% spf
model.physics('spf').feature('fp1').set('rho_mat', 'userdef');
model.physics('spf').feature('fp1').set('mu_mat', 'userdef');
model.physics('spf').feature('fp1').set('rho', num2str(rho));
model.physics('spf').feature('fp1').set('mu', num2str(dvis));
% ht
model.physics('ht').feature('solid1').set('k_mat', 'userdef');
model.physics('ht').feature('solid1').set('rho_mat', 'userdef');
model.physics('ht').feature('solid1').set('k', {num2str(ks) '0' '0' '0' num2str(ks) '0' '0' '0' num2str(ks)});
model.physics('ht').feature('solid1').set('Cp_mat', 'userdef');
model.physics('ht').feature('solid1').set('rho', num2str(rho_s));
model.physics('ht').feature('solid1').set('Cp', num2str(cp_s));
model.physics('ht').feature.create('fluid1', 'FluidHeatTransferModel', 3);
% model.physics('ht').feature('fluid1').set('minput_velocity_src', 'root.comp1.u');
model.physics('ht').feature('fluid1').set('rho_mat', 'userdef');
model.physics('ht').feature('fluid1').set('rho', num2str(rho));
model.physics('ht').feature('fluid1').set('Cp_mat', 'userdef');
model.physics('ht').feature('fluid1').set('Cp', num2str(cp));
model.physics('ht').feature('fluid1').set('k_mat', 'userdef');
model.physics('ht').feature('fluid1').set('k', {num2str(kl) '0' '0' '0' num2str(kl) '0' '0' '0' num2str(kl)});
model.physics('ht').feature('fluid1').set('gamma_mat', 'userdef');
model.physics('ht').feature('fluid1').set('gamma', num2str(gamma));

disp('* initializing complete')

%% initial figure config.
figure(1)
xlabel('axial position [mm]')
ylabel('R [mm]')
grid minor
set(gca,'fontsize',16)

figure(4)
xlabel('axial position [mm]')
ylabel('Tw [K]')
grid minor
set(gca,'fontsize',16)

%% --------------------- Tv Iteration -------------------------------------

for ite_Tv = 1:ite_max

display(['- Tv ite: ' num2str(ite_Tv)'] )

% R iteration figures
figure(1)
```

```
plot(linspace(0,lt,length(R))*1000,R*1000,'.')
hold on
legend;
saveas(gcf,'R_iteration.fig')
saveas(gcf,'R_iteration.pdf')

% Tw iteration figures
figure(4)
plot(linspace(0,lt,length(Tw_ft))*1000,Tw_ft-Tv(ite_Tv),'.')
hold on
legend;
saveas(gcf,'Tw_iteration.fig')
saveas(gcf,'Tw_iteration.pdf')

%% --------------------- R iteration ---------------------------------

for ite_R = 1:2

if ite_R == 1
disp('    - Momentum Solution')
end

%% building geometry (spf)

flag = 1; w3 = w3_c;

for i=1:nz

if flag == 1 && Tw(i)>Tv(ite_Tv)
w3 = w3_e;
flag = 2;
end

% arc point cloud
alpha = acosd((w2-w3)/R(i));
dmin  = pi - (alpha*pi)/180;
ang   = linspace(-dmin,-pi*0.5,nx);

xc = w1+w2;
h3 = sqrt(R(i)^2 - w2^2);
yc = h1 + h2 + h3;

x_arc = xc+R(i)*cos(ang);
y_arc = yc+R(i)*sin(ang);

yarc_end(i) = y_arc(end);   % point data evaluation coordinates
yarc_start(i) = y_arc(1);   % mc boundary coordinates

% work plane
model.geom('geom1').feature.create(['wp' num2str(i)], 'WorkPlane');
```

```
model.geom('geom1').feature(['wp' num2str(i)]).set('unite', true);
model.geom('geom1').feature(['wp' num2str(i)]).set('quickz', dz*(i-1));

% polygon
model.geom('geom1').feature(['wp' num2str(i)]).geom.create('pol1', 'Polygon');
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').set('source', 'table');
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').set('type', 'open');

model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').setIndex('table', x_arc(1), 0, 0);
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').setIndex('table', y_arc(1), 0, 1);
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').setIndex('table', w1, 1, 0);
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').setIndex('table', h1+h2, 1, 1);
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').setIndex('table', w1, 2, 0);
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').setIndex('table', h2, 2, 1);
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').setIndex('table', w1+w2, 3, 0);
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').setIndex('table', h2, 3, 1);
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').setIndex('table', w1+w2, 4, 0);
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('pol1').setIndex('table', y_arc(end), 4, 1);

% interpolation curve
model.geom('geom1').feature(['wp' num2str(i)]).geom.create('ic1', 'InterpolationCurve');

for xi=1:nx
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('ic1').setIndex('table', x_arc(xi), xi-1, 0);
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('ic1').setIndex('table', y_arc(xi), xi-1, 1);
end

% convert to solid
model.geom('geom1').feature(['wp' num2str(i)]).geom.create('csol1', 'ConvertToSolid');
model.geom('geom1').feature(['wp' num2str(i)]).geom.feature('csol1').selection('input').set({'ic1' 'pol1'});
model.geom('geom1').run(['wp' num2str(i)]);

% collecting work planes
men{i} = ['wp' num2str(i)];

wpt = i;
end

% creating loft
model.geom('geom1').create('loft1', 'Loft');
model.geom('geom1').feature('loft1').selection('profile').set(men);
model.geom('geom1').feature('loft1').set('facepartitioning', 'grid');

% creating mc boundary
wpt = wpt+1;
model.geom('geom1').feature.create(['wp' num2str(wpt)], 'WorkPlane');
model.geom('geom1').feature(['wp' num2str(wpt)]).set('unite', true);
model.geom('geom1').feature(['wp' num2str(wpt)]).set('quickplane', 'yz');
model.geom('geom1').feature(['wp' num2str(wpt)]).set('quickx', num2str(w1));
```

```
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.create('r1', 'Rectangle');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('r1').set('base', 'corner');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('r1').set('size', {num2str(hmf) num2str(lt)});
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('r1').set('pos', {num2str(h1+h2-hmf) num2str(0)});

% setting relative reapiring tolerance
model.geom('geom1').feature('fin').set('repairtol', '1.0E-7');
model.geom('geom1').run('fin');

%% termination
if ite_R == 2
disp(['      * R_old(1)/R_new(1): ' num2str(R_old(1)) ' / ' num2str(R_new(1))])
fprintf(['      * mfr @out: ' num2str(mfr_out) '\n'])
break
end

%% boundary conditions (spf)
b_mf = [];
for i=1:nz-1
dum = mphselectbox(model, 'geom1', [w1-epsi h1+h2-hmf-epsi dz*(i-1)-epsi; ...
w1+epsi h1+h2+epsi dz*i+epsi]', 'boundary');
b_mf = [b_mf dum];
end

b_sym = mphselectbox(model, 'geom1', [w1+w2-epsi h2-epsi -epsi; w1+w2+epsi h1+h2+eps lt+epsi]', 'boundary');
b_slip = mphselectbox(model, 'geom1', [w1-epsi h2+epsi -epsi; w1+w2+epsi h1+h2+epsi lt+epsi]', 'boundary');
b_out = mphselectbox(model, 'geom1', [w1-epsi h2-epsi lt-epsi; w1+w2+epsi h1+h2+epsi lt+epsi]', 'boundary');
b_slip = setdiff(b_slip,b_mf);

cmlf_mic = []; emlf_mic = [];
for i=1:nz-1
Rce = R(i);%(R(i)+R(i+1))/2;
if Tw(i)<Tv(ite_Tv)    % condensation region
[mlf_c,c0] = cmfr_v4(Tv(ite_Tv),Tw(i),Rce,w1,w2,c0,Pv,kl,sft,kvis,M,Ru,hlv,Vl);
cmlf_mic = [cmlf_mic mlf_c];
else                   % evaporation region
mlf_e = emfr_v4(Rce,Tw(i),Tv(ite_Tv),w2,Pv,sft,M,Vl,Ru,kl,kvis,hlv,A_const);
emlf_mic = [emlf_mic mlf_e];
end
end

mfr_mic_spf = [cmlf_mic*D1 -emlf_mic*D2]*dz;
mfr_mac = -q_mac/hlv;
mfr_tot = (mfr_mic_spf + mfr_mac);

for i=1:length(mfr_tot)
model.component('comp1').physics('spf').create(['inl' num2str(i)], 'InletBoundary', 2);
model.component('comp1').physics('spf').feature(['inl' num2str(i)]).selection.set(b_mf(i));
model.component('comp1').physics('spf').feature(['inl' num2str(i)]).set('BoundaryCondition', 'MassFlow');
model.component('comp1').physics('spf').feature(['inl' num2str(i)]).set('mfr', mfr_tot(i));
```

```
end

% outlet BC
model.physics('spf').feature.create('out1', 'OutletBoundary', 2);
model.physics('spf').feature('out1').set('SuppressBackflow', false);
model.physics('spf').feature('out1').selection.set(b_out);
model.physics('spf').feature('out1').set('p0', P_fe);

% interface BC
model.component('comp1').physics('spf').create('wallbc2', 'WallBC', 2);
model.component('comp1').physics('spf').feature('wallbc2').selection.set(b_slip);
model.component('comp1').physics('spf').feature('wallbc2').set('BoundaryCondition', 'Slip');

% symmetry BC
model.physics('spf').feature.create('sym1', 'Symmetry', 2);
model.physics('spf').feature('sym1').selection.set(b_sym);

%% running mesh (spf)
model.mesh('mesh1').autoMeshSize(mq_spf);
model.mesh('mesh1').run;

%% model solver (spf)
model.sol('sol1').runAll;

%% saving model (spf)
model.save(['C:\Users\Barbaros\Desktop\CemKurt\v19\spf_',num2str(ite_Tv),'_',num2str(ite_R)]); % saving model

%% results (R_new,mfr_out)

% R_new
model.result.dataset.create('cpt1', 'CutPoint3D');
model.result.dataset('cpt1').set('pointx', num2str(w1+w2/2));
model.result.dataset('cpt1').set('pointy', num2str(h2+h1/2));
for i=1:nz
model.result.dataset('cpt1').set('pointz', num2str(dz*(i-1)));
P_new(i) = mphinterp(model,'p','dataset','cpt1');
end
model.result.dataset.remove('cpt1');

% velocity & mass flow rate
model.result.dataset.create('cpl1', 'CutPlane');
model.result.dataset('cpl1').set('quickplane', 'xy');
for i=1:nz
model.result.dataset('cpl1').set('quickz', num2str(dz*(i-1)));
cell_mfr(i) = mphint2(model,'(u*cpl1nx+v*cpl1ny+w*cpl1nz)*spf.rho','surface','dataset','cpl1');
dum_1(i) = mphint2(model,'(u*cpl1nx+v*cpl1ny+w*cpl1nz)','surface','dataset','cpl1');
dum_2(i) = mphint2(model,'1','surface','dataset','cpl1');
velocity(i) = dum_1(i)/dum_2(i);
end
model.result.dataset.remove('cpl1');
```

```
mfr_out = cell_mfr(end);

% pressure
model.result.dataset.create('cpt1', 'CutPoint3D');
model.result.dataset('cpt1').set('pointx', num2str(w1+w2/2));
model.result.dataset('cpt1').set('pointy', num2str(h2+h1/2));
for i=1:nz
model.result.dataset('cpt1').set('pointz', num2str(dz*(i-1)));
P_new(i) = mphinterp(model,'p','dataset','cpt1');
end
model.result.dataset.remove('cpt1');

R_old = R;
R_new = sft./(Pv-P_new);
R_new(end) = sft/(Pv-P_fe);
R = R_new;

%% removing features
% BC's
for i=1:length(mfr_tot)
model.physics('spf').feature.remove(['inl' num2str(i)]);
end
model.physics('spf').feature.remove('out1');
model.physics('spf').feature.remove('wallbc2');
model.physics('spf').feature.remove('sym1');
% geometry
for i=1:wpt
model.geom('geom1').feature.remove(['wp' num2str(i)]);
end
model.geom('geom1').feature.remove('loft1');
model.geom('geom1').run('fin');

D1 = D1i;
D2 = D2i;

end

%% momentum figures

% velocity figure
figure(2)
plot(linspace(0,lt,length(velocity))*1000,velocity,'.')
hold on
xlabel('axial position [mm]')
ylabel('velocity [m/s]')
legend;
set(gca,'fontsize',16)
grid minor
saveas(gcf,'U_results.fig')
saveas(gcf,'U_results.pdf')
```

```
% mfr figure
figure(3)
plot(linspace(0,lt,length(mfr_tot))*1000,mfr_tot,'.')
hold on
legend;
xlabel('position [mm]')
ylabel('mass flow rate [kg/s]')
set(gca,'fontsize',16)
grid minor
legend;
saveas(gcf,'mfr_results.fig')
saveas(gcf,'mfr_results.pdf')

%% -------------------- Tw Iteration ---------------------------------

%% building geometry (ht)
% heat pipe base
wpt = wpt+1;
model.geom('geom1').feature.create(['wp' num2str(wpt)], 'WorkPlane');
model.geom('geom1').feature(['wp' num2str(wpt)]).set('unite', true);
model.geom('geom1').feature(['wp' num2str(wpt)]).set('quickz', 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.create('pol1', 'Polygon');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').set('source', 'table');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').set('type', 'open');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', 0, 0, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', 0, 0, 1);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', w1+w2, 1, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', 0, 1, 1);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', w1+w2, 2, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', h2, 2, 1);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', w1, 3, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', h2, 3, 1);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', w1, 4, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', h1+h2, 4, 1);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', 0, 5, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', h1+h2, 5, 1);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', 0, 6, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', 0, 6, 1);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.create('csol1', 'ConvertToSolid');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('csol1').selection('input').set({'pol1'});
model.geom('geom1').feature.create('ext1', 'Extrude');
model.geom('geom1').feature('ext1').set('workplane', ['wp' num2str(wpt)]);
model.geom('geom1').feature('ext1').selection('input').set({['wp' num2str(wpt)]});
model.geom('geom1').feature('ext1').setIndex('distance', num2str(lt), 0);
model.geom('geom1').run(['wp' num2str(wpt)]);

% heat load/sink regions
wpt = wpt + 1;
model.geom('geom1').feature.create(['wp' num2str(wpt)], 'WorkPlane');
```

```
model.geom('geom1').feature(['wp' num2str(wpt)]).set('unite', true);
model.geom('geom1').feature(['wp' num2str(wpt)]).set('quickplane', 'xz');
model.geom('geom1').feature(['wp' num2str(wpt)]).set('quicky', 0);

model.geom('geom1').feature(['wp' num2str(wpt)]).geom.create('pol1', 'Polygon');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').set('source', 'table');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').set('type', 'open');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', 0, 0, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', lc, 0, 1);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', w1+w2, 1, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', lc, 1, 1);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', w1+w2, 2, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', lt-lh, 2, 1);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', 0, 3, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature('pol1').setIndex('table', lt-lh, 3, 1);

model.component('comp1').geom('geom1').run;

wpt = wpt + 1; % for the condensation region heat flux boundary

% liquid domain selection
model.physics('ht').feature('fluid1').selection.set(2);

%% iteration for Tw

for ite_Tw = 1:ite_max

display(['    - Energy Iteration: ' num2str(ite_Tw)] )

% Tw iteration
figure(100+ite_Tv)
plot(linspace(0,lt,length(Tw))*1000,Tw-Tv(ite_Tv),'.')
hold on
xlabel('axial position [mm]')
ylabel('Tw [K]')
grid minor
set(gca,'fontsize',16)
legend;
saveas(gcf,['Tw_local_ite',num2str(ite_Tv) '.fig'])
saveas(gcf,['Tw_local_ite',num2str(ite_Tv) '.pdf'])

%% pc heat flux
cmlf_mic = []; emlf_mic=[];
for i=1:nz-1
Rce = R(i);
if Tw(i) < Tv(ite_Tv)    % condensation region (forward)
[mlf_c,c0] = cmfr_v4(Tv(ite_Tv),Tw(i),Rce,w1,w2,c0,Pv,kl,sft,kvis,M,Ru,hlv,Vl);
cmlf_mic = [cmlf_mic mlf_c];
else
% evaporation region
```

```
mlf_e = emfr_v4(Rce,Tw(i),Tv(ite_Tv),w2,Pv,sft,M,Vl,Ru,kl,kvis,hlv,A_const);
emlf_mic = [emlf_mic mlf_e];
end
end

% condensation & evaporation heat flux boundary width
for i=1:length(cmlf_mic)
theta_dum = acosd(w2/R(i));
w4_1(i) = w3_c./sind(theta_dum);
end
for i=1:length(emlf_mic)+1
theta_dum = acosd(w2/R(length(cmlf_mic)+i));
w4_1(length(cmlf_mic)+i) = w3_e./sind(theta_dum);
end
for i=1:nz-1
w4(i) = (w4_1(i) + w4_1(i+1))/2;
end
w4c = w4(1:length(cmlf_mic));
w4e = w4(length(cmlf_mic)+1:end);

% normal outputs
cmf_mic = cmlf_mic./(w1+w4c);
emf_mic = emlf_mic./(w4e);
qf_mic = [cmf_mic -emf_mic]*hlv;
h_mic = qf_mic./(Tv(ite_Tv)-Tw);

% auxillary outputs
q_mic_evap = emlf_mic*dz*hlv;
q_mic_cond = cmlf_mic*dz*hlv;

%% condensation geometry

% condensation region
model.geom('geom1').feature.create(['wp' num2str(wpt)], 'WorkPlane');
model.geom('geom1').feature(['wp' num2str(wpt)]).set('unite', true);
model.geom('geom1').feature(['wp' num2str(wpt)]).set('quickplane', 'xz');
model.geom('geom1').feature(['wp' num2str(wpt)]).set('quicky', h1+h2);
for i=1:length(cmlf_mic)
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.create(['pol' num2str(i)], 'Polygon');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature(['pol' num2str(i)]).set('source', 'table');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature(['pol' num2str(i)]).set('type', 'open');
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature(['pol' num2str(i)]).setIndex('table', 0, 0, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature(['pol' num2str(i)]).setIndex('table', dz*i, 0, 1);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature(['pol' num2str(i)]).setIndex('table', w1, 1, 0);
model.geom('geom1').feature(['wp' num2str(wpt)]).geom.feature(['pol' num2str(i)]).setIndex('table', dz*i, 1, 1);
end

model.component('comp1').geom('geom1').run;

%% boundary selection (ht)
```

```
b_dum1 = [];
for i=1:nz-1
dum = mphselectbox(model, 'geom1', [w1-epsi h1+h2-hmf-epsi dz*(i-1)-epsi; ...
w1+epsi h1+h2+epsi dz*i+epsi]', 'boundary');
b_dum1 = [b_dum1 dum];
end
b_dum2 = mphselectbox(model, 'geom1', [w1-epsi h2+epsi -epsi; w1+w2+epsi h1+h2+epsi lt+epsi]', 'boundary');
b_mesh = setdiff(b_dum2,b_dum1);

b_qout = mphselectbox(model, 'geom1', [-epsi -epsi -epsi; w1+w2+epsi +epsi lc+epsi]', 'boundary');
b_qin = mphselectbox(model, 'geom1', [-epsi -epsi lt-lh-epsi; w1+w2+epsi +epsi lt+epsi]', 'boundary');
dum1 = mphselectbox(model, 'geom1', [-epsi -epsi -epsi; +epsi h1+h2+epsi lt+epsi]', 'boundary');
dum2 = mphselectbox(model, 'geom1', [w1+w2-epsi -epsi -epsi; w1+w2+epsi h1+h2+eps lt+epsi]', 'boundary');
b_sym = [dum1 dum2];  dum1 = []; dum2 = [];

b_cond = [];
for i=1:length(cmlf_mic)
dum = mphselectbox(model, 'geom1', [-epsi min(yarc_start)-epsi dz*(i-1)-epsi; ...
w1+max([w3_c w3_e])+epsi epsi+h1+h2 dz*i+epsi]', 'boundary');
b_cond = [b_cond dum];
end

b_evap = [];
for i=1:length(emlf_mic)
dum = mphselectbox(model, 'geom1', [w1-epsi min(yarc_start)-epsi dz*length(cmlf_mic)+dz*(i-1)-epsi; ...
w1+max([w3_e w3_c])+epsi epsi+h1+h2 dz*length(cmlf_mic)+dz*i+epsi]', 'boundary');
b_evap = [b_evap dum];
end
b_evap = setdiff(b_evap,b_dum1);

b_Tv = [];
for i=1:nz-1
dum = mphselectbox(model, 'geom1', [w1+min([w3_c w3_e])-epsi min(yarc_end)-epsi dz*(i-1)-epsi; ...
w1+w2+epsi h1+h2+epsi dz*i+epsi]', 'boundary');
b_Tv = [b_Tv dum];
end

%% running mesh (ht)
model.mesh('mesh1').autoMeshSize(mq_ht);
model.mesh('mesh1').run;

%% boundary conditions (ht)

% symmetry BC
model.physics('ht').feature.create('sym1', 'Symmetry', 2);
model.physics('ht').feature('sym1').selection.set(b_sym);

% q"in BC
model.physics('ht').feature.create('hf1', 'HeatFluxBoundary', 2);
```

```
model.physics('ht').feature('hf1').selection.set(b_qin);
model.physics('ht').feature('hf1').set('q0', num2str(q_in));
model.physics('ht').feature('hf1').setIndex('materialType', 'solid', 0);

% q"out BC
model.physics('ht').feature.create('hf2', 'HeatFluxBoundary', 2);
model.physics('ht').feature('hf2').selection.set(b_qout);
model.physics('ht').feature('hf2').set('HeatFluxType', 'ConvectiveHeatFlux');
model.physics('ht').feature('hf2').set('h', num2str(htc));
model.physics('ht').feature('hf2').set('Text', num2str(Tamb));
model.physics('ht').feature('hf2').setIndex('materialType', 'solid', 0);

% vapor-liquid interface BC
model.physics('ht').feature.create('temp1', 'TemperatureBoundary', 2);
model.physics('ht').feature('temp1').selection.set(b_Tv);
model.physics('ht').feature('temp1').set('T0', num2str(Tv(ite_Tv)));

% phase change BC
for i=1:length(cmlf_mic)
model.physics('ht').feature.create(['hf' num2str(i+2)], 'HeatFluxBoundary', 2);
model.physics('ht').feature(['hf' num2str(i+2)]).set('HeatFluxType', 'ConvectiveHeatFlux');
model.physics('ht').feature(['hf' num2str(i+2)]).selection.set([b_cond(2*i-1) b_cond(2*i)]);
model.physics('ht').feature(['hf' num2str(i+2)]).setIndex('materialType', 'solid', 0);
model.physics('ht').feature(['hf' num2str(i+2)]).set('h', h_mic(i));
model.physics('ht').feature(['hf' num2str(i+2)]).set('Text', Tv(ite_Tv));
end
for i=1:length(emlf_mic)
model.physics('ht').feature.create(['hf' num2str(i+length(cmlf_mic)+2)], 'HeatFluxBoundary', 2);
model.physics('ht').feature(['hf' num2str(i+length(cmlf_mic)+2)]).set('HeatFluxType', 'ConvectiveHeatFlux');
model.physics('ht').feature(['hf' num2str(i+length(cmlf_mic)+2)]).selection.set(b_evap(i));
model.physics('ht').feature(['hf' num2str(i+length(cmlf_mic)+2)]).set('h', h_mic(length(cmlf_mic)+i));
model.physics('ht').feature(['hf' num2str(i+length(cmlf_mic)+2)]).set('Text', Tv(ite_Tv));
end

%% model solver (ht)
model.sol('sol2').runAll;

%% results
for i=1:nz-1
model.result.dataset.remove('cpt1');
model.result.dataset.create('cpt1', 'CutPoint3D');
model.result.dataset('cpt1').set('pointx', w1);
model.result.dataset('cpt1').set('pointy', h1+1*h2/2);
model.result.dataset('cpt1').set('pointz', dz/2+dz*(i-1));
model.result.dataset('cpt1').set('data', 'dset2');
Tw_ft(i) = mphinterp(model,'T','dataset','cpt1');
model.result.dataset.remove('cpt1');
end

for i=1:length(cmlf_mic)
```

```
model.result.dataset.create('surf1', 'Surface');
model.result.dataset('surf1').set('data', 'dset2');
model.result.dataset('surf1').selection.set([b_cond(2*i-1) b_cond(2*i)]);
Tw_new(i) = mphmean(model,'T','surface','dataset','surf1');
model.result.dataset.remove('surf1');
end
for i=1:length(emlf_mic)
model.result.dataset.create('surf1', 'Surface');
model.result.dataset('surf1').set('data', 'dset2');
model.result.dataset('surf1').selection.set(b_evap(i));
Tw_new(length(cmlf_mic)+i) = mphmean(model,'T','surface','dataset','surf1');
model.result.dataset.remove('surf1');
end

for i=1:nz-1
model.result.dataset.create('surf1', 'Surface');
model.result.dataset('surf1').set('data', 'dset2');
model.result.dataset('surf1').selection.set(b_Tv(i));
q_mac(i) = mphint2(model,'ht.ntflux','surface','dataset','surf1');
area_mac(i) = mphint2(model,'1','surface','dataset','surf1');
model.result.dataset.remove('surf1');
end

%% saving model
model.save(['C:\Users\Barbaros\Desktop\CemKurt\v19\ht_',num2str(ite_Tv),'_',num2str(ite_Tw)]);

%% error Tw
error_Tw = max(abs(Tw_new-Tw));

fprintf(['      error: ' num2str(error_Tw) '\n'])

if error_Tw < err_ht
Tw = Tw_new;
break
else
Tw = Tw_new;

%% removing features
% BC's
for i=1:nz+1
model.physics('ht').feature.remove(['hf' num2str(i)]);
end
model.physics('ht').feature.remove('sym1');
model.physics('ht').feature.remove('temp1');
% geometry
model.geom('geom1').feature.remove(['wp' num2str(wpt)]);
end

end     % iteration for Tw ends
```

```
%% energy figures

% q_mac figure
figure(5)
plot(linspace(0,lt,length(q_mac))*1000,-q_mac,'.')
hold on
xlabel('axial position [mm]')
ylabel('heat transfer (macro) [W]')
legend;
set(gca,'fontsize',16)
grid minor
saveas(gcf,'qmac_results.fig')
saveas(gcf,'qmac_results.pdf')

% h figure
figure(6)
plot(linspace(0,lt,length(h_mic))*1000,h_mic,'.' )
hold on
xlabel('axial position [mm]')
ylabel('heat transf. coeff. [W/ m^{2} K ]')
legend;
set(gca,'fontsize',16)
grid minor
saveas(gcf,'h_results.fig')
saveas(gcf,'h_results.pdf')

%% energy balance
for i=1:6
model.result.dataset.create(['surf' num2str(i)], 'Surface');
model.result.dataset(['surf' num2str(i)]).set('data', 'dset2');
end
model.result.dataset('surf1').selection.set(b_qin);
model.result.dataset('surf2').selection.set(b_qout);
model.result.dataset('surf3').selection.set(b_cond);
model.result.dataset('surf4').selection.set(b_evap);
model.result.dataset('surf5').selection.set(b_Tv(1:length(b_cond)));
model.result.dataset('surf6').selection.set(b_Tv(length(b_cond)+1:end));

q_IN(ite_Tv,1) = mphint2(model,'ht.ntflux','surface','dataset','surf1');
q_OUT(ite_Tv,1) = mphint2(model,'ht.ntflux','surface','dataset','surf2');
q_COND_MAC(ite_Tv,1) = mphint2(model,'ht.ntflux','surface','dataset','surf5');
q_EVAP_MAC(ite_Tv,1) = mphint2(model,'ht.ntflux','surface','dataset','surf6');

% measured by COMSOL (m)
q_COND_MIC_m(ite_Tv,1) = mphint2(model,'ht.ntflux','surface','dataset','surf3');
q_EVAP_MIC_m(ite_Tv,1) = mphint2(model,'ht.ntflux','surface','dataset','surf4');
q_COND_TOT_m(ite_Tv,1) = q_COND_MIC_m(ite_Tv,1) + q_COND_MAC(ite_Tv,1);
q_EVAP_TOT_m(ite_Tv,1) = q_EVAP_MIC_m(ite_Tv,1) + q_EVAP_MAC(ite_Tv,1);

% calculated by the model (c)
```

```
q_COND_MIC_c(ite_Tv,1) = -sum(q_mic_cond);
q_EVAP_MIC_c(ite_Tv,1) = sum(q_mic_evap);
q_COND_TOT_c(ite_Tv,1) = q_COND_MIC_c(ite_Tv,1) + q_COND_MAC(ite_Tv,1);
q_EVAP_TOT_c(ite_Tv,1) = q_EVAP_MIC_c(ite_Tv,1) + q_EVAP_MAC(ite_Tv,1);

q_TOT_m(ite_Tv,1) = q_IN(ite_Tv,1) + q_OUT(ite_Tv,1) + q_COND_TOT_m(ite_Tv,1) + q_EVAP_TOT_m(ite_Tv,1);
q_TOT_c(ite_Tv,1) = q_IN(ite_Tv,1) + q_OUT(ite_Tv,1) + q_COND_TOT_c(ite_Tv,1) + q_EVAP_TOT_c(ite_Tv,1);

for i=1:6
model.result.dataset.remove(['surf' num2str(i)]);
end

ENERGY_BALANCE_MEASURED =   ...
table(q_IN,q_OUT,q_COND_TOT_m,q_EVAP_TOT_m,q_COND_MIC_m,q_COND_MAC,q_EVAP_MIC_m,q_EVAP_MAC,q_TOT_m);

ENERGY_BALANCE_CALCULATED = ...
table(q_IN,q_OUT,q_COND_TOT_c,q_EVAP_TOT_c,q_COND_MIC_c,q_COND_MAC,q_EVAP_MIC_c,q_EVAP_MAC,q_TOT_c);

disp(ENERGY_BALANCE_MEASURED)
fprintf('\n')
%disp(ENERGY_BALANCE_CALCULATED)
%fprintf('\n')

%% TV secant method

% objective function
f(ite_Tv) = (q_OUT(ite_Tv,1) - abs(q_IN(ite_Tv,1)))/(abs(q_IN(ite_Tv,1)));

if ite_Tv > 1

del = -f(ite_Tv)*(Tv(ite_Tv)-Tv(ite_Tv-1))/(f(ite_Tv)-f(ite_Tv-1));

% damper
ra1 = cdamp*abs(del/Tv(ite_Tv));
ra2 = ra1^2;
ra3 = ra1*ra2;
damper(ite_Tv) = (1+ra2)/(1+ra2+ra3);

Tv(ite_Tv+1) = Tv(ite_Tv) + damper(ite_Tv)*del;

error_Tv = abs(Tv(ite_Tv+1)-Tv(ite_Tv));

%% display iteration
ITE = (1:ite_Tv)';
F = f';
DT(ite_Tv,1) = (Tv(ite_Tv)-Tv(ite_Tv-1));
DF(ite_Tv,1) = (f(ite_Tv)-f(ite_Tv-1));
DEL(ite_Tv,1) = del;
TV = Tv(1:ite_Tv)';
ERR(ite_Tv,1) = error_Tv;
```

```
Tv_ITERATION = table(ITE,DF,DT,F,TV,DEL,ERR);
disp(Tv_ITERATION)
fprintf('\n')

%% error
if error_Tv<err_Tv && ite_Tv>2
break
else
DTw = Tw - Tv(ite_Tv);
Tw = DTw + Tv(ite_Tv+1);
end
else
DTw = Tw - Tv(1);
Tw = DTw + Tv(2);
end

figure(7)
plot(Tv(1:end-1),f,'-o','linewidth',1.5)
xlabel('Tv')
ylabel('f')
set(gca,'fontsize',16)
grid minor
saveas(gcf,'Tv_vs_f.fig')
saveas(gcf,'Tv_vs_f.pdf')

figure(8)
plot(1:ite_Tv+1,Tv,'-o','linewidth',1.5)
xlabel('iteration')
ylabel('Tv')
set(gca,'fontsize',16)
grid minor
saveas(gcf,'Tv_iteration.fig')
saveas(gcf,'Tv_iteration.pdf')

%% removing features

% geometry
for i=1:wpt
model.geom('geom1').feature.remove(['wp' num2str(i)]);
end
model.geom('geom1').feature.remove('loft1');
model.geom('geom1').feature.remove('ext1');

% data points(spf)
for i=1:nz
model.result.dataset.remove(['cpt' num2str(i)]);
end
% data points(ht)
for i=1:nz-1
model.result.dataset.remove(['cpt' num2str(i+nz)]);
```

```
end
% boundary conditions(ht)
model.physics('ht').feature.remove('sym1');
model.physics('ht').feature.remove('temp1');
for i=1:nz+1
model.physics('ht').feature.remove(['hf' num2str(i)]);
end
% boundary conditions(spf)
model.physics('spf').feature.remove('wallbc2');
model.physics('spf').feature.remove('sym1');
for i=1:nz-1
model.physics('spf').feature.remove(['inl' num2str(i)]);
end
model.physics('spf').feature.remove('out1');

% %% constants
% D1 = D1i; D2 = D2i;

end % iteration for Tv ends

%% output

% R iteration figures
figure(9)
plot(linspace(0,lt,length(R))*1000,R*1000,'g.','linewidth',1.5)
xlabel('position [mm]')
ylabel('radius of curvature [mm]')
grid minor
set(gca,'fontsize',16)
saveas(gcf,'R_final.fig')
saveas(gcf,'R_final.pdf')

% U iteration figures
figure(10)
plot(linspace(0,lt,length(velocity))*1000,velocity,'b.','linewidth',1.5)
legend('Odabasi','Present Study')
xlabel('position [mm]')
ylabel('channel velocity [m/s]')
grid minor
set(gca,'fontsize',16)
saveas(gcf,'U_final.fig')
saveas(gcf,'U_final.pdf')

% Tw iteration figures
figure(11)
plot(linspace(0,lt,length(Tw_ft))*1000,Tw_ft-Tv(ite_Tv),'r.','linewidth',1.5)
legend('Odabasi','Present Study')
xlabel('position [mm]')
ylabel('fin-top-corner temperature [K]')
grid minor
```

```
set(gca,'fontsize',16)
saveas(gcf,'Tft_final.fig')
saveas(gcf,'Tft_final.pdf')

% saving model
model.save('C:\Users\Barbaros\Desktop\CemKurt\v19\final');

fprintf('\n* Done! \n')

% elapsed time
t_hr = floor(toc/3600);
t_min = floor((toc - t_hr*3600)/60);
t_sec = floor(toc - t_min*60 - t_hr*3600);
fprintf('\n* Ellapsed time is %d hour %d minute %d seconds. \n',t_hr,t_min,t_sec);
fprintf('\n* Date/Time: %s \n',datetime);

out = model;

end
```

Appendix B

MATLAB Function for the Condensation Model

```
function [mlf,c0] = cmfr_v4(Tv,Tw,R,w1,w2,c0_initial,Pv,kl,sft,kvis,M,Ru,hlv,Vl)


%% properties & initial values

theta = acosd(w2/R);

ds = 1e-7;

%% calculations

error_criter = 1e-11;

c0_1 = c0_initial;

c0_2 = c0_1*(1+1e-2);

f1_c0 = fnc_c0(theta,w1,sft,kvis,c0_1,kl,hlv,M,Ru,Pv,Vl,Tw,Tv,ds);

f2_c0 = fnc_c0(theta,w1,sft,kvis,c0_2,kl,hlv,M,Ru,Pv,Vl,Tw,Tv,ds);

while(1)

c0_3 = c0_2 - f2_c0*(c0_2-c0_1)/(f2_c0-f1_c0);
[f3_c0, Tlv] = fnc_c0(theta,w1,sft,kvis,c0_3,kl,hlv,M,Ru,Pv,Vl,Tw,Tv,ds);

error = abs(f3_c0-f2_c0);
```

```
if error < error_criter

c0 = c0_3;

c1 = -tand(90 - theta);
c3 = -c1/(2*w1^2);
mlf = 6*sft*c0^3*c3/(3*kvis);

break
else

c0_1 = c0_2;
c0_2 = c0_3;
f1_c0 = f2_c0;
f2_c0 = f3_c0;
end
end

% ------------------------------------------------------------------------ %

function [fc0, Tlv] = fnc_c0(theta,w1,sft,kvis,c0,kl,hlv,M,Ru,Pv,Vl,Tw,Tv,ds)


c1_f1 = -tand(90 - theta);
c3_f1 = -c1_f1/(2*w1^2);

fnc1 = sft*6*c0^3*c3_f1/(3*kvis);

n = w1/ds;

for i_f1=1:n+1

s = ds*(i_f1-1);

c_delta = delta(c0,theta,w1,s);
c_2delta = delta2(theta,w1,s);

Tlv(i_f1) = secant_Tlv(Tw,Tv,kl,hlv,sft,M,Ru,Pv,Vl,c_delta,c_2delta);
ac_f1 = constant_a(M,Ru,Tlv(i_f1),Pv,hlv,Tv);
bc_f1 = constant_b(M,Ru,Tlv(i_f1),Pv,Vl);

dmmy(i_f1) = (ac_f1*(Tw-Tv) + bc_f1*(-sft*c_2delta))/(1+ac_f1*c_delta*hlv/kl);
end

fnc2 = 0;
for i_f1=1:n
fnc2 = fnc2 + 0.5*(dmmy(i_f1)+dmmy(i_f1+1))*ds;
end
```

```
fc0 = fnc1 + fnc2;
end

% ---------------------------------------------------------------------- %

function sTlv = secant_Tlv(Tw,Tv,kl,hlv,sft,M,Ru,Pv,Vl,c_delta,c_2delta)

error_criter1 = 1e-12;

Tlv_1 = Tw + (Tv - Tw)*1e-2;

f1_Tlv = fnc_Tlv(Tlv_1,Tw,Tv,kl,hlv,sft,M,Ru,Pv,Vl,c_delta,c_2delta);

Tlv_2 = Tlv_1*(1+1e-4);

f2_Tlv = fnc_Tlv(Tlv_2,Tw,Tv,kl,hlv,sft,M,Ru,Pv,Vl,c_delta,c_2delta);

while(1)

Tlv_3 = Tlv_2 - f2_Tlv*(Tlv_2-Tlv_1)/(f2_Tlv-f1_Tlv);
f3_Tlv = fnc_Tlv(Tlv_3,Tw,Tv,kl,hlv,sft,M,Ru,Pv,Vl,c_delta,c_2delta);

error1 = abs(f3_Tlv-f2_Tlv);

if error1 < error_criter1

sTlv = Tlv_3;
break
else

f1_Tlv = f2_Tlv;
f2_Tlv = f3_Tlv;
Tlv_1 = Tlv_2;
Tlv_2 = Tlv_3;
end

end

end

% ---------------------------------------------------------------------- %

function fTlv = fnc_Tlv(Tlv,Tw,Tv,kl,hlv,sft,M,Ru,Pv,Vl,c_delta,c_2delta)

fnc1 = (Tlv-Tw)*kl/(hlv*c_delta);

ac_f3 = constant_a(M,Ru,Tlv,Pv,hlv,Tv);
bc_f3 = constant_b(M,Ru,Tlv,Pv,Vl);

fnc2 = (ac_f3*(Tw-Tv) + -bc_f3*sft*c_2delta)/(1+ac_f3*c_delta*hlv/kl);
```

```
fTlv = fnc1 + fnc2;

end

% ------------------------------------------------------------------------ %

function ac = constant_a(M,Ru,Tlv,Pv,hlv,Tv)

acom = 1;

ac = (2*acom/(2-acom))*(M/(2*pi*Ru*Tlv))^0.5*(M*Pv*hlv/(Ru*Tv*Tlv));

end

% ------------------------------------------------------------------------ %

function bc = constant_b(M,Ru,Tlv,Pv,Vl)

acom = 1;

bc = (2*acom/(2-acom))*sqrt(M/(2*pi*Ru*Tlv))*(Pv*Vl/(Ru*Tlv));

end

% ------------------------------------------------------------------------ %

function cdelta = delta(c0,theta,w1,s)

c1_f6 = -tand(90 - theta);
c2_f6 = 0;
c3_f6 = -c1_f6/(2*w1^2);
c4_f6 = -c1_f6/(2*w1)^3;

cdelta = c0 + c1_f6*(s-w1) + c2_f6*(s-w1).^2 + c3_f6*(s-w1).^3 + ...
c4_f6*(s-w1).^4;

end

% ------------------------------------------------------------------------ %

function c2delta = delta2(theta,w1,s)

c1_f7 = -tand(90 - theta);
c2_f7 = 0;
c3_f7 = -c1_f7/(2*w1^2);
c4_f7 = -c1_f7/(2*w1)^3;

c2delta = 2*c2_f7 + 6*c3_f7*(s-w1) + 12*c4_f7*(s-w1)^2;
```

```
end

% ------------------------------------------------------------------------ %
end
```

Appendix C

MATLAB Function for the Evaporation Model

```
function [xmd] = emfr_v4(R,Tw,Tv,w2,Pv,sft,M,Vl,Ru,kl,kvis,hlv,A_const)

%% user defined parameters

ds = 5e-12;                     % unit length, % assume micro region 1d-6 sonunda kontrol edilmeli

%% inputs

%     R:    radius of the arc
%     Tv:   vapor temperature
%     Tw:   wall temperature
%     Pv:   vapor pressure of water
%     w1:   half-width of theæ fin
%     w2:   half-width of the groove
%     kl:   thermal conductivity of water
%     sft   surface tension
%     kvis: kinematic viscosity
%     M:    molar mass of liquid
%     Ru:   universal gas constant
%     hlv:  latent heat evaporation
%     Vl:   molar volume of liquid

%% initial calculations

theta = acosd(w2/R);

%% initial values, deltas
```

```
% (2.15a), (2.16)
delta(1)     = ((-A_const*R*(1-1e5))/sft)^(1/3);

% (2.15b)
d_delta(1)   = -tand(theta);

% FDM(5)
delta(2)     = d_delta(1)*ds + delta(1);

% (2.15c), (2.7)
dd_delta(1)  = (1-1e-5)*((1+d_delta(1)^2)^1.5)/R;

% FDM(11)
delta(3)     = dd_delta(1)*ds^2 + 2*delta(2) - delta(1);

% (2.8), (2.15d)
trm0 = (1+d_delta(1)^2);
trm1 = 3*A_const*d_delta(1)/delta(1)^4;
trm2 = 3*sft*dd_delta(1)^2*d_delta(1)/trm0^2.5;
trm3 = sft/(trm0^1.5);
ddd_delta(1) = trm1/trm3 + trm2/trm3;

% FDM(p5)
delta(4)     = ds^3*ddd_delta(1)+delta(1)-3*delta(2)+3*delta(3);

% (2.14)
Tlv_ev  = Tv;   % meniscus side
acof    = acoefs(Pv,Tv,M,hlv,Ru,Tlv_ev);
bcof    = bcoefs(Pv,M,Vl,Ru,Tlv_ev);
P_dis   =  A_const/(delta(1)^3);
P_cap   = (sft*dd_delta(1))/((1+d_delta(1)^2)^1.5);
DP      = P_dis + P_cap;
xm1     = (acof*(Tw-Tv)+bcof*(-DP));
xm2     = acof*hlv*delta(1)/kl;

xmf(1)  = (xm1)/(1+xm2);

%% find delta's starting i=2

T_int = Tv + ((Tw - Tv)*1e-2);

for i=2:1e10

% FDM
d_delta(i)  =  (delta(i+1)-delta(i-1))/(2*ds);
dd_delta(i) =  (delta(i+1)-2*delta(i)+delta(i-1))/(ds^2);
P_dis       =  A_const/(delta(i)^3);
P_cap       = (sft*dd_delta(i))/((1+d_delta(i)^2)^1.5);
DP          =  P_dis+P_cap;
```

```
% (2.13)
T_int =  secant_T(DP,delta(i),Ru,Tw,Tv,T_int,Pv,M,hlv,Vl,kl);

% (2.14)
acof    = acoefs(Pv,Tv,M,hlv,Ru,T_int);
bcof    = bcoefs(Pv,M,Vl,Ru,T_int);
term1   = (acof*(Tw-Tv)+bcof*(-DP));
term2   = (acof*hlv*delta(i))/(kl);
xmf(i)  = term1/(1+term2);

% dpl/ds (2.8)
trm0 = (1+d_delta(i-1)^2);
trm1 = 3*A_const*d_delta(i-1)/delta(i-1)^4;
trm2 = 3*sft*dd_delta(i-1)^2*d_delta(i-1)/trm0^2.5;
trm3 = -sft*ddd_delta(i-1)/(trm0^1.5);
dplds(i-1) = trm1 + trm2 + trm3;

%
f1     = (-3*kvis*xmf(i-1))/(delta(i-1)^3) ;
f2       = (-3*d_delta(i-1)*dplds(i-1))/(delta(i-1));
func     = f1+f2;
dplds(i)   = dplds(i-1) + func*ds;

% (2.8)
trm0 = (1+d_delta(i)^2);
trm1 = 3*A_const*d_delta(i)/delta(i)^4;
trm2 = 3*sft*dd_delta(i)^2*d_delta(i)/trm0^2.5;
trm3 = sft/(trm0^1.5);
ddd_delta(i) = (trm1 + trm2 -dplds(i))/trm3;

% FDM
delta(i+3)   = ds^3*ddd_delta(i) +3*delta(i+2)-3*delta(i+1)+delta(i);

rat_m = xmf(i)/xmf(1);
rat_T(i) = (Tw-T_int);
rat_d(i) =abs((delta(i)-delta(i-1))/delta(i-1));

if   xmf(i)<0 || d_delta(i) == abs(d_delta(i))
break
end
end

% integrate to find me'
xmd = 0;
for j=1:i-1
xmd= xmd + (xmf(j)+xmf(j+1))*0.5*ds;
end

etfr_l = ds*(i-1);
```

```
%-----------------------------------------------------------------------
function T_int = secant_T(delp,delta,Ru,Tw,Tv,T_o,Pv,M,hlv,Vl,kl)

criter = 5e-12;

acoef1    = acoefs(Pv,Tv,M,hlv,Ru,T_o);
bcoef1    = bcoefs(Pv,M,Vl,Ru,T_o);
f_o     = T_o - find_Tlv(kl,hlv,acoef1,bcoef1,delp,delta,Tw,Tv);
T_n     = T_o*(1+1e-4);

while(1)

acoef1  = acoefs(Pv,Tv,M,hlv,Ru,T_n);
bcoef1  = bcoefs(Pv,M,Vl,Ru,T_n);
f_n    = T_n - find_Tlv(kl,hlv,acoef1,bcoef1,delp,delta,Tw,Tv);

result_T    =  T_n - f_n*(T_n-T_o)/(f_n-f_o);
error   = abs(result_T-T_n);

if error<=criter
T_int = result_T;
break
else
T_o = T_n;
T_n = result_T;
f_o = f_n;
end
end

end
%--------------function to calculate Tlv ----------------------------------
function Tlv = find_Tlv(kl,hlv,a,b,DP,delta,Tw,Tv)

tTlv1 = kl*Tw/(delta*hlv) + a*Tv + b*DP;
tTlv2 = a + kl/(delta*hlv);
Tlv  = tTlv1/tTlv2;

end
%--------------function to find coefficient a ----------------------------
function acoef = acoefs(Pv,Tv,M,hlv,Ru,Tlv)

ta1   = sqrt(M/(2*pi*Ru*Tlv));
ta2   = (Pv*M*hlv)/(Ru*Tv*Tlv);
% acom  = (2*0.13)/(2-0.13);    %lefevre
acom  = 1;                    % water
acomf = 2*acom/(2-acom);
acoef = acomf*ta1*ta2;

end
```

```
%--------------function to find coefficient b -----------------------------
function bcoef = bcoefs(Pv,M,Vl,Ru,Tlv)

tb1   = sqrt(M/(2*pi*Ru*Tlv));
tb2   = (Vl*Pv)/(Ru*Tlv);
% acom  = (2*0.13)/(2-0.13);  %lefevre
acom  = 1;                    % water
acomf = 2*acom/(2-acom);
bcoef = acomf*tb1*tb2;

end
% end

end
```