NUMERIC MODELING OF A MICRO-SCALE DIFFERENTIAL THERMAL CALORIMETER FOR THE PURPOSE OF CRYOPRESERVATION

A Thesis

presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

By

Han Han

Dr. Solbrekken, Thesis Supervisor

DECEMBER 2017

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

NUMERIC MODELING OF A MICRO-SCALE DIFFERENTIAL THERMAL CALORIMETER FOR THE PURPOSE OF CRYOPRESERVATION

presented by Han Han,

a candidate for the degree of master of science,

and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Gary Solbrekken

Professor Adam Helfer

Professor Robert Tzou

ACKNOWLEDGEMENTS

I would never have been able to finish my thesis without the guidance of my committee members, help from friends, and support from my family and my husband.

I would like to acknowledge Dr. Gary Solbrekken, my adviser, for his assistance and dedicated involvement throughout every step of my research. It has been an honor to be his master's student. Dr. Solbrekken gave very generously of his time and knowledge. His vast knowledge and expertise have inspired me to pursue my goals with hard work and dedication. The knowledge and wisdom he has imparted upon me will forever remain a major contributor to my success and achievements. I would like to thank Dr. Adam Helfer and Dr. Robert Tzou for guiding me in writing my thesis and helping me to develop my background in quantum mechanics and heat transfer.

My sincere thanks go to my research partners, Manasi Kulrani and Christin Young, for their help conducting the experiments for the same research topic. I would like to thank John Kennedy, Jerome Rivers, Annemarie Hoyer, Gautham Manoharan, and Gerhard Schnieders for giving me good suggestions and honest feedback on my work.

I would also like to acknowledge my family for supporting me financially and mentally. I am very grateful for the love and support that they have given me.

Last but not least, I would like to thank my husband, Christopher Lenz. He has been extremely supportive of my work. His encouragement has guided me through the good times and the bad.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Nomenclature

C_p	Heat Capacity	T_w	Wall Temperature
h	Specific Enthalpy	f_l	Volume Fraction of Fluid
k	Thermal Conductivity	S	Heat Source
A	Area	R	Resistance
I	Current	$\vec{u}$	Velocity Component
L	Latent Heat	s	Thermal-layer Thickness
T	Temperature	ρ	Density
W	Thickness of the TE Module	N	Number of Junctions
c	Specific Heat	t	Time

Greek

δ	Node Spacing	Δx	Control Volume Size

Subscription

B	Bottom Node	s	Solid
N	North Node	e	East Boundary
S	South Node	eq	Equivalent Property
T	Top Node	w	West Boundary
E	East Node	s	South Boundary
P	Present Node	n	North Boundary
W	West Node	t	Top Boundary
m	Melting	b	Bottom Boundary
l	Liquid	c	Constant Value

Superscription

0	Previous Time Step	1	Current Time Step

Abstract

Cryopreservation requires biological material to be stored at temperatures well below the freezing point of water. During the process of cryopreservation, the cooling rate should be carefully chosen to avoid cell damage due to the unbalanced pressure and the solution effect. Cellular thermal analysis that determines the thermodynamics properties of a micro-scale biological material is necessary for a successful cryopreservation procedure.

A micro-scale differential thermal analyzer (μDTA) was previously developed to obtain accurate thermal properties measurements of freezing cells. However, the thermal signal needs to be properly interpreted in terms of how much ice is created. Also, the future cooling profile needed to experimentally control the cooling rate needs to be developed. So a 3D numeric model was built in STAR CCM+ to study the temperature change of the water droplets while being frozen by a thermoelectric module. The heat flux profile for the boundary condition was scaled to accommodate the heat spread effect. The numeric model solutions successfully matched the temperature distribution results from the experiments. By using additional heat flux to accommodate the heat spread effect, the STAR CCM+ model generated relatively accurate results on a smaller geometry within a very short computational time. This is a big improvement compared with other high fidelity computational simulation models used to solve similar problems. After the 3D model was calibrated, the same STAR CCM+ model was used to predict the temperature distribution of different sizes of water droplets during the freezing process. This model allowed us to better understand the temperature distribution of a water droplet when the water droplet was being cooled until frozen. The modeling technique developed

in this research will help establish the required cooling rates needed to control ice formation and establish thermodynamic properties of cell freezing solutions.

1. Introduction

1.1 Motivation

The world lost species at a rapidly increasing rate for the past 20 years. The current rate of biodiversity loss is much greater than background rates. Based on the World Wildlife Fund (WWF)'s most recent report, the rate that we are losing species is roughly between 1000 and 10,000 times higher than the natural extinction rate [1]. Between 1.4 and 1.8 million species have already been identified as extinct. Even if we estimate the number of species that has gone extinct every year by using the lowest extinction rate, we are facing 200 to 2,000 extinctions every year. In the face of human population growth, the extinction rate will keep increasing as predicted by scientists [2].

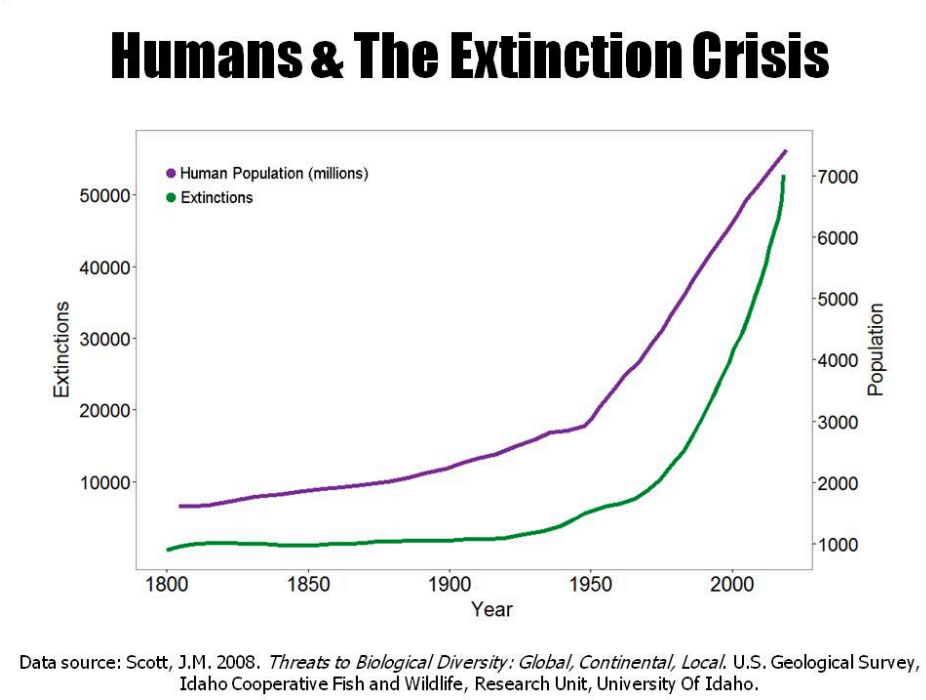


Figure 1.1 Human population and extinction rate

Human beings closely interact with wildlife in many different ways. Fish provide about 3 billion people with almost 20% of their intake of animal protein. Plants account for over 80% of the human diet. 30,000 terrestrial plants are known to be edible. Pollination services by insects and other animals affect 35% of the world's crop production.

It has been reported that a decrease in biological diversity would disrupt pollination between plants, water purification, the food chain and other important processes. A decrease in biodiversity would also make it harder for humans to harvest resources from ecosystems for the development of our society. In the end, a decrease in biological diversity will hurt human beings. Loss of genetic diversity has a great impact on the ability of populations to recover from natural disasters, which will lead to a higher chance of human extinction[3].

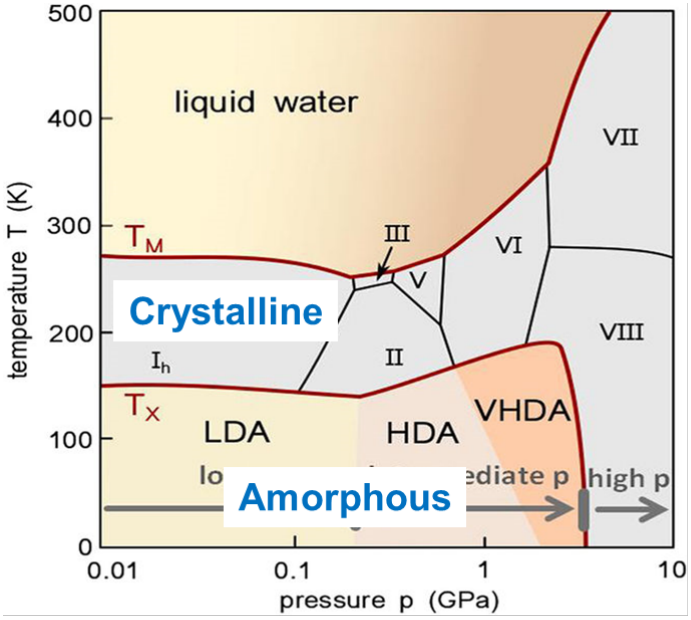
1.2 Cryopreservation Techniques

Cryopreservation has been proven to be an effective way to save genetic diversity. By conserving germ plasm of endangered species, human beings can slow down the process of endangered species going extinct. Cryopreservation is a very complex process. Certain principles of cryopreservation need to be followed carefully to increase the success rate. With the guidance of these principles, two main procedures will be introduced in this section.

1.2.1 Principles of cryopreservation

To store a biological material for a long time, the material has to be brought below a certain temperature to suspend all biological activities. There are two effects that

should be avoided during the cooling process of cryopreservation: ice crystal formation and solution effect [4].



J.Stern & T. Loerting, Crystallisation of the amorphous ices in the intermediate pressure regime, Scientific Reports 7, Article number: 3995 (2017)

Figure 1.2 Phase diagram of water[5]

First, cells contain a relatively large amount of water. When cells are cooled below the freezing temperature of water, the water inside cells can form crystalline or amorphous ice as shown in Fig 1.2. Ice crystal formation inside a cell disrupts the membranes and other structures.

The second major goal of a successful cryopreservation process involves avoidance of solution effects. Ice crystals forming outside a cell can cause internal dehydration and damage to sensitive proteins. Since solutes remain in the cytosol as

liquid water solidifies, the solutes can reach very high concentrations. High-level concentration of electrolytes and other solutes can be toxic to intracellular proteins.

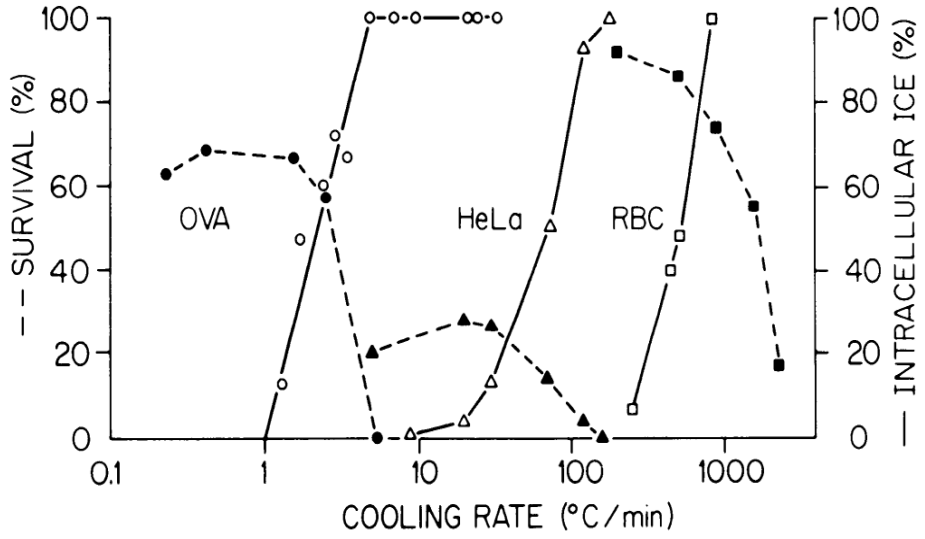


Figure 1.3 Survival rate and intracellular ice percentage vs cooling rate[6]

By cooling the cells at extremely high cooling rates around $10^5 K/s$, amorphous ice can be formed. The advantage of amorphous ice is that it will not extend as much as crystalline ice. However, we currently do not have the technology to achieve cooling rates on the order of $10^5 K/s$ for most cells and tissues. Therefore, cryoprotectants are used to displace some cellular water. The cooling rates needed to vitrify cryoprotectants are much lower. For example, in Fig 1.3, the cooling rates that ensure high survival rates of different cells range approximately from 0.2 K/s to 1000 K/s. By controlling ice formation and solute concentrations, it is possible to freeze biological samples.

1.2.2 Vitrification

Vitrification is a common method used in cryopreservation, which allows for fairly high survival rates. Cooling rates in vitrification are high compared to the cooling

rates used in freezing methods. Depending on the water content and the concentration of cryoprotectants in the biological samples, the required cooling rates could vary from less than 7 ℃/min to several thousand degrees per min.

Inside of biological materials, cytosol behaves much like water during a freezing process. During a vitrification process, the goal is to freeze the liquid into an amorphous solid. Early ice crystal formation and growth are found to have a negative impact on the survival rates of the preserved biomaterial. Unlike glass or ceramics, which will turn into amorphous solids naturally during a freezing process with high cooling rates, water becomes amorphous at near freezing temperatures with salt and cryoprotectants[7].

In several studies of the survival rates during vitrification of human oocytes, the samples were cooled close to -196 ℃ while suspended in different solutions of cryoprotectant. The entire group of specimens was then thawed to test the survival rate. The survival rates of vitrification are listed in Table.1.1 below.

Table 1.1 Human Oocytes Vitrification Studies and Results

Author	Number of Human Oocytes	Survival rate (%)
Chian(2005)[8]	180	93.9
Kuwayama(2005)[9]	64	90.8
Yoon (2003) [10]	474	68.7

(Data Source: J. K. Jain and R. J. Paulson, "Oocyte cryopreservation," Fertility and sterility, vol. 86, pp. 1037-1046, 2006.[4] and Yoon, T.K., et al., Live births after vitrification of oocytes in a stimulated in vitro fertilization–embryo transfer program. Fertility and sterility, 2003. 79(6): p. 1323-1326.[11])

1.2.3 Slow-freeze

Slow-freeze is another standard approach to preserve cells. And just as the name says, the cooling rate in this approach is very slow[11]. With the slow cooling rate, cytosol solidifies into its amorphous form directly. To test the survival rate of the biomaterials, cells also need to be frozen first and then thawed. A typical standard procedure is introduced below.

Before slowly freezing the cells, cryoprotectants are usually added at room temperature to minimize the toxic effects. After the slow-freeze process begins, the temperature is slowly reduced from room temperature to approximately -6 ℃ at first with the initial cooling rate being about 2 ℃/min. The nucleation is then induced manually during this period of time. When the biomaterial reaches the equilibrium state, the temperature will then be gradually reduced to -30 ℃ with a constant cooling rate of 0.3 ℃/min [12]. The concentrations of cryoprotectants will increase as cells cool down to -30 ℃. This is because the forming ice crystal excludes the solutes. With the slow-freeze method, the cells' metabolic rates become really slow, which can help the cells to avoid the toxic effect. In the end, the ice crystal gradually grows into the extracellular area. The slow cooling rates allow the additional cryopretectant to permeate into the cells to retain a balanced pressure between the inside and the outside of the cells. Once the cells and the solution is converted to a solid, the frozen biomaterial is stored in liquid nitrogen to stay at a vitrified state [4].

During the thawing process, the temperature is raised very rapidly to lower the risk of intracellular ice formation. The cryotube, with frozen cells in it, is usually plunged into a hot water bath at approximately 40 ℃ for 1 min [13]. Non-permeating cryoprotectant is added to create a pressure difference so that permeating cryoprotectant

can be diffused out of the cell. The overall survival rate results of the slow-freeze method are listed in Table.1.2.

Table 1.2 Slow-freeze studies and results

Author	Number of Human Oocytes	Survival rate (%)
Jain (1996-2004)[4]	4,000	55.4
Jain (2002-2004)[4]	688	69.2
Fabbri (2001)[14]	1,502	53.0
Chen (2005) [15]	159	74.8

1.2.4 Studies of the crystallization of water droplets

The slow-freeze method and vitrification method have been proven to be effective at conserving both plant cells and animal cells. However, in order to better preserve and characterize a single cell, the traditional methods are insufficient. A reliable way to develop a preservation protocol hasn't been found so far and the limit of the toleration of ice inside a single cell remains unknown. A thermal analyzer is a very good device to fulfill the goal of analyzing the thermal behavior of a single cell.

Each individual cell behaves differently in a freezing test. However, a study shows that the protoplast behaviors are similar between the animal cells and plant cells [16]. Despite the structural differences between the two types of cells, when biomaterials are plasmolyzed, animal cells and plant cells have a similar intracellular mechanism to prevent themselves from being damaged. All types of cells contain a large amount of cytosol, which has similar thermal properties to water.

Human body cells typically consist of cell membrane, cytoplasm, cell organelles, and nucleus. According to Geoffrey Cooper's research, approximately 70% or more of the mass of a cell is water. [17] In the early stages of a single cell thermal behavioral study, a water droplet sample is used as a common replacement for plant cells and animal cells in calibrating thermal analyzers.

During a high cooling rate solidification process, a micro-scale water droplet will experience four stages if it is initially at a temperature above its freezing point. A water droplet will first be cooled down to the freezing point in the liquid phase. Then, the water droplet will start to freeze while heat of fusion is released. The latent heat released during this stage can dramatically increase the water droplet temperature, therefore forming a peak in the temperature profile plot. During the third stage, the water droplet will continue cooling down slowly until it is fully solidified. Finally, the water droplet will be cooled down in its solid form. This process is well captured in Fig.1.4.

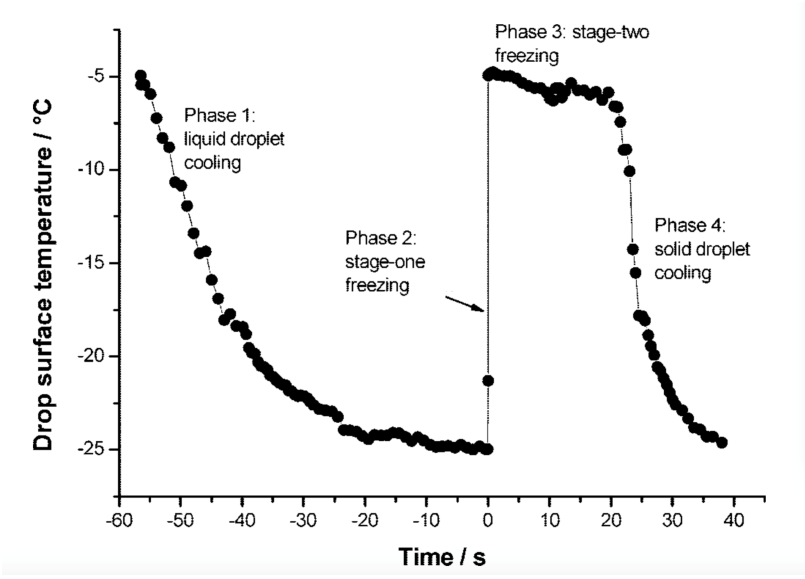


Fig.1.4 Water droplet surface temperature [18]

However, real cells still have 30% to 50 % of dry weight. The dry weight is made of proteins, RNA, lipids and other substance. Some cells even contain a type of protein called antifreeze protein (AFP). AFPs bind to small ice crystals to inhibit growth and recrystallization of ice. AFPs have been identified in fish, numerous plants, and insects living in areas susceptible to ice formation. There is a possibility of cells being partially frozen during a cryopreservation cooling process.

1.3 Differential Thermal Analyzers

Thermal analysis experiments were first conducted by Le Chatelier in 1887 to study the heating curve for clay minerals[19]. However, using DTAs to take heat of transition and heat of fusion measurements didn't start until the 1940s. Now, commercial Differential Scanning Calorimetry (DSC) can be purchased from several manufacturers at a wide variety of price. [20] DSCs are utilized to study the crystalline transition, melting, phase diagrams and heat capacity of biomaterials. For the purpose of cryopreservation, DTAs are designed to measure the change in weight of a biological sample as a function of a temperature profile, determine sample decomposition, oxidation, and loss of solvent or water. DTAs can also be used to obtain the temperature where phase change takes place. The commercial version DSCs are usually built based on two newer types of DTAs, heat-flux DSC (hf-DSC) and power compensation DSC (pc-DSC) [21].

The basic concept behind DTAs is that they measure the difference in temperature between a sample and a base-line reference during tests. Temperature programs are imported into computers to enable recent DTAs to take measurements automatically. The temperature results are recorded as a function of time. A heat-flux type DSC can capture

the heat flux alteration during a phase change process using a heat sensitive plate built inside of the hf-DSC. Enthalpy of transition then can be estimated from the incremental temperature fluctuation results[22]. On the other hand, pc-DSC usually has two individual heaters to maintain the temperature of the sample and the reference material. Energy or power that is input to maintain the temperature is then recorded as a function of time. The energy input is also proportional to the heat capacity of the testing sample.

DTAs are very useful when analyzing the thermal behavior of a single cell. When an ice crystal starts to form in a single cell, heat of fusion will be released. The water inside of the cell will go through the four stages as shown in Fig.1.4. A DTA can record the temperature results as a function of time and the heat capacity of the material. Given the heating or cooling rate, shape factor and other information, thermal conductivity of the sample can be calculated. DTAs will allow for the calculations of the thermal properties, therefore allowing us to better understand the thermal behaviors of a single cell.

1.4 Objective of Study

A micro-scale differential thermal analyzer (μDTA), shown in Fig.1.5, was developed by Campton and Armes[23] to obtain accurate thermal properties measurements. The μDTA setup includes three control systems and a data acquisition system. The control systems control the temperature of the air, the humidity of the air and the cooling rate of the thermoelectric module.

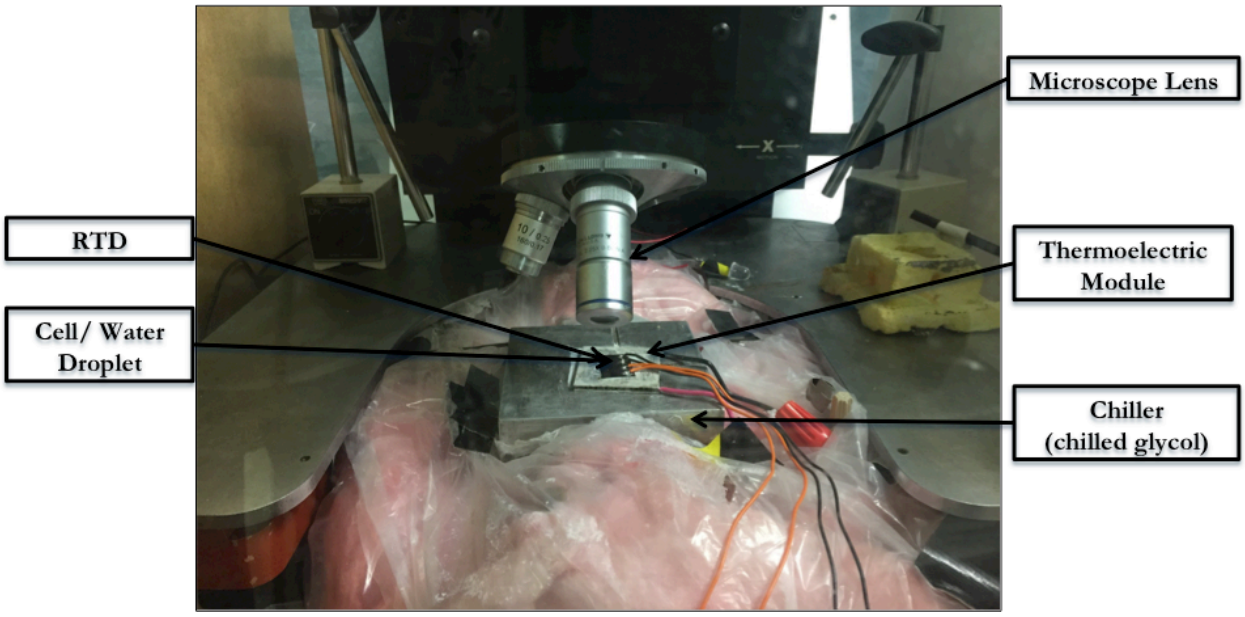


Fig. 1.5 Micro-scale differential thermal analyzer prototype

Gross temperature control is provided by an industrial chiller controls that circulates ethylene glycol through a cold block at the bottom. And a Proportional Integral Derivative (PID) controller controls a power supply to provide integrated cooling and heating rates. The cooling and warming rate of the μDTA is determined by a current profile output from the PID controller. The local humidity around the μDTA is controlled by the atmospheric control system with moist air being pumped out of the testing housing while the dry air is being pumped in simultaneously. A humidity sensor is constantly tracking the relative humidity of the air. A thermoelectric module is attached to the cold block and provides temperature control of the μDTA that is attached to the other side of the thermoelectric module.

In order to automate the data collection process, Campton and Armes developed a user interface using C#. This interface controls the power supply for the TE module and

the multi-meter that collect data from the thermocouple. Later on, the thermocouple was changed to a resistance temperature detector (RTD) for easier sample loading. The schematic of the data acquisition system is shown in Fig. 1.6.

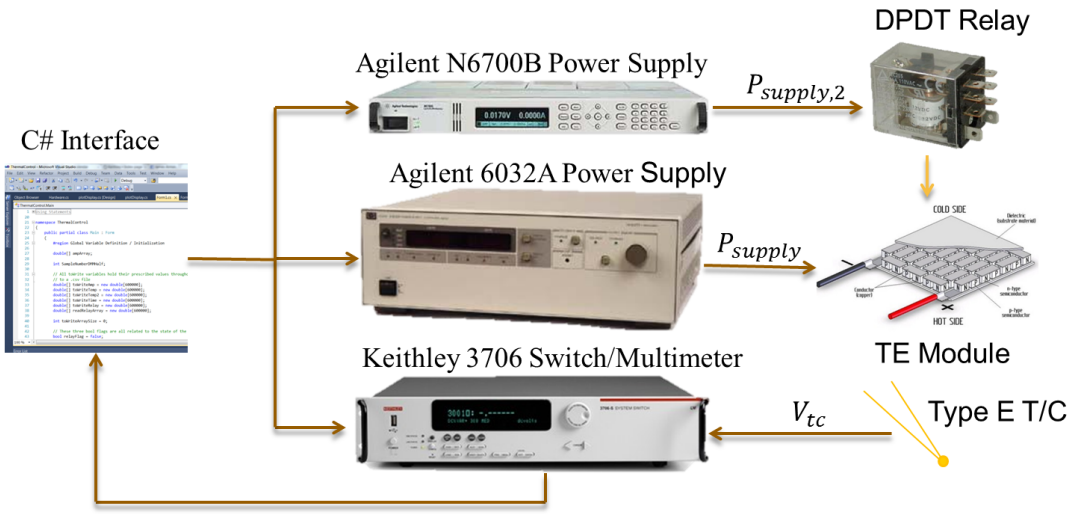


Fig. 1.6 Schematic of the data acquisition system

Prior to conducting the experiments on cells, water droplets were tested. The following steps were followed for each experiment. First, deionized water droplets were loaded in position and the initial temperature of the sample was held at 20 ℃. Enough time was allowed for the sample to reach thermal equilibrium and for the humidity of the experimental housing to drop below 20%. The chiller was set at −35 ℃ and then the measurement plan was executed while cooling down the TE module. The program would automatically stop the cooling process once the experiment was finished. Fig. 1.7 gives an example of the temperature profile of a water droplet sample after one complete experiment.

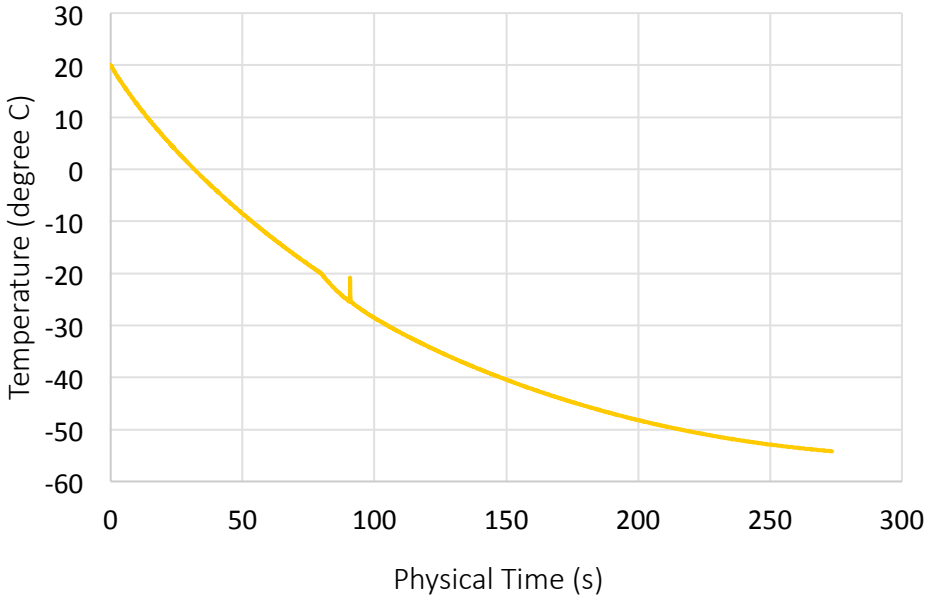


Fig. 1.7 Temperature Profile of a Water Droplet Sample

From the temperature profiles that were obtained in the past, the µDTA prototype allowed for calculations of enthalpy change, error in repeatability, heat of fusion, and volume measurements. Based on the performance of the µDTA prototype, a better generation of µDTA is required to enhance the prediction of the µDTA's transient thermal output.

To enhance the accuracy of the current profile, Compton made a one-dimensional, transient, conduction-only model in MATLAB. Armes developed a much more complex 2D numeric model using MATLAB to simulate the heat transfer process of the thermoelectric module. In Armes' model, a direct transient comparison was made. By comparing to Compton's 1D simulation results, it was found that both the one-dimensional and the two-dimensional models fail to capture the thermal lag phase at the

beginning of the cooling process. However, based on the experimental observation results, Armes' model did predict the cooling rate better.

A 3D numeric model built in STAR CCM+ is then proposed. This model will be calibrated with experimental data. In this numeric model, a water cube is built to simulate the cell activity when being frozen by a thermoelectric pad. This model will allow us to investigate the temperature distribution at different locations within the cell and help in the interpretation of the experimental data. It will also allow us to explore fractions of water cube that water freezes.

2. Literature Review

2.1 Introduction

Many analytical solutions for transient heat conduction profiles are reported in the literature. However, in many practical situations, analytical solutions haven't been obtained yet due to irregular geometry or unique boundary conditions. The thermal behavior of biomaterials has been studied using numerical methods. Computational Fluid Dynamics (CFD) is an effective and powerful tool to study thermal processes numerically. Among all the numerical techniques, the finite element method and the finite volume method are often used in commercial packages because of their high accuracy, general purpose application, and ability to consider complex geometries and physics.

In CFD, some physical models solve the Navier-Stokes equations while others solve the Boltzmann equations.[24] In order to obtain the temperature profiles and to predict the freezing time of phase change materials, the method so-called "Equivalent Heat Capacity Method" and the "Enthalpy Method" have been implemented often times. Both physical models are considered the classical engineering approaches to solve a phase-change heat transfer problem.

2.2 Available Numerical Techniques

2.2.1 Finite Element Method

Finite element modeling involves the discretization of the structure into elements that are defined by nodes as shown below. Since the nodes define the elements' boundaries, the shape of the elements could be different from case to case. Element shape

functions can range from linear interpolation functions to higher order polynomial functions. Generally, elements are formed in a local coordinate system. Elemental matrices are transformed from the local elemental coordinate system to the global coordinate system.

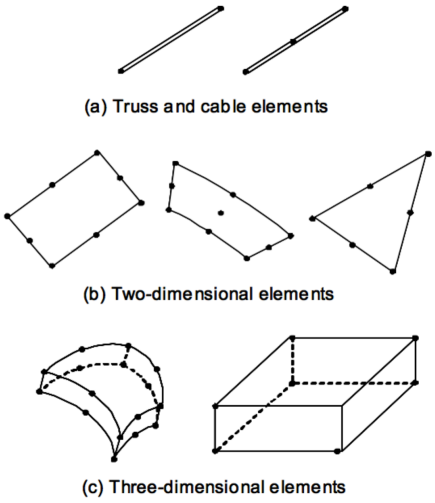


Fig. 2.1 Typical elements for finite-element method (Data source: F. Mahmoudian and G. F. Margrave, "A review of the finite-element method in seismic wave modelling," Technical report, CREWES Research Report2003.) [25]

The finite element method was first proposed by Y. Cheung to solve heat conduction problems[26] by developing partial differential equations of the function at each node. Field quantities are stored in each of the elements at the nodal points. Using the finite element method, thermal properties (such as thermal conductivity, heat capacity, density, latent heat and heat transfer coefficient) are approximated using discretized equations over each of the domains. The general solution for all elements results in an algebraic set of simultaneous equations of unknown nodal values. The equations are

calculated on a finite element basis after the application of boundary conditions using the variational principle and Euler minimum integral theorem.[27]

In numeric simulations involving phase change problems, simultaneous temperature dependence of thermal properties is assumed. These properties are set to change rapidly within a very narrow range of melting temperature. [28]

There is some literature that has solved the same heat conduction problems using the finite element method. For example, for a non-linear melting and solidification problem with geometry shown below, the analytical solution and numeric solution obtained by using the finite element method were compared. The material is at 283K initially and the wall is at constant 253K for the solidification process. Fig.2.2 shows the temperature distribution of the region after 200s and the freezing point of this material is 283.15K.

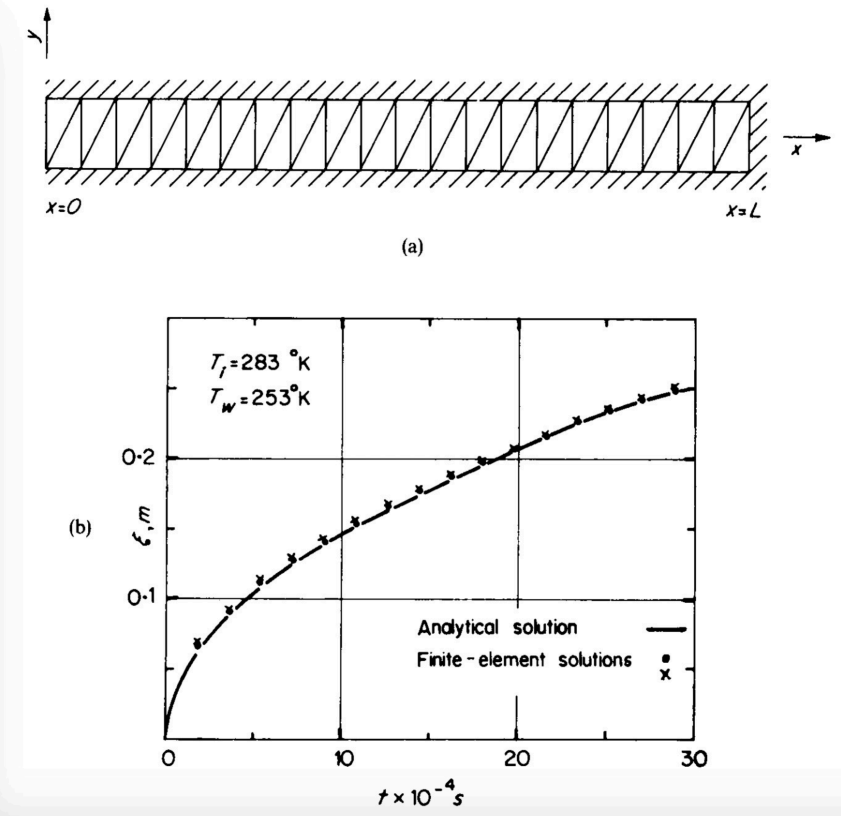


Fig.2.2 Solidification of a slab in liquid.(a) Finite-element mesh. Non-conductive boundaries are assumed throughout but at the face x = 0. Slab thickness: L = 1 m (b) Progress of freezing front through slab: distance vs time.(Data source: G. Comini,"Finite element solution of non-linear heat conduction problems with special reference to phase change," International Journal for Numerical Methods in Engineering, vol. 8, pp. 613-624, 1974.) [28]

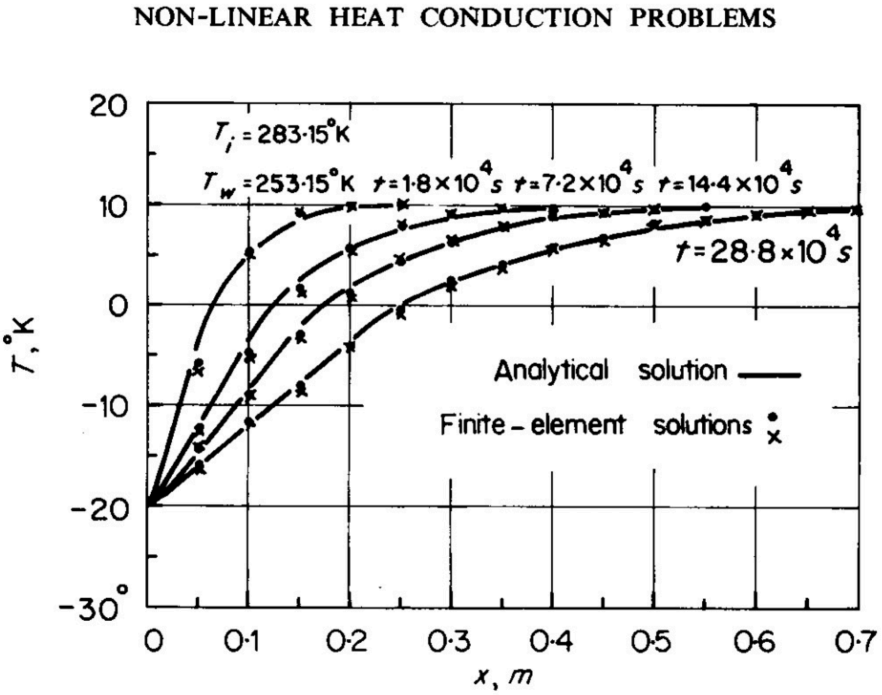


Fig.2.3 Temperature distributions at different time values during the freezing process referred to in Figure 2.2 (Data source: G. Comini,"Finite element solution of non-linear heat conduction problems with special reference to phase change," International Journal for Numerical Methods in Engineering, vol. 8, pp. 613-624, 1974.) [28]

There are many other applicable heat conduction problems that have been solved by the finite element method with decent accuracy. It has been proven that the finite element method has the capacity to deal with phase change problems, determination of thermal stresses, food freezing, thawing technology, etc.[28-30]

2.2.2 Finite Volume Method

Patankar and Spalding developed the finite volume method (FVM) using a physical approach for deriving the nodal equations.[31] When using FVM, the domain is divided into many small control volumes, usually in the shape of a rectangle or triangle. The integral conservation law is enforced for small control volumes (CV) defined by the computational mesh and the entire domain.

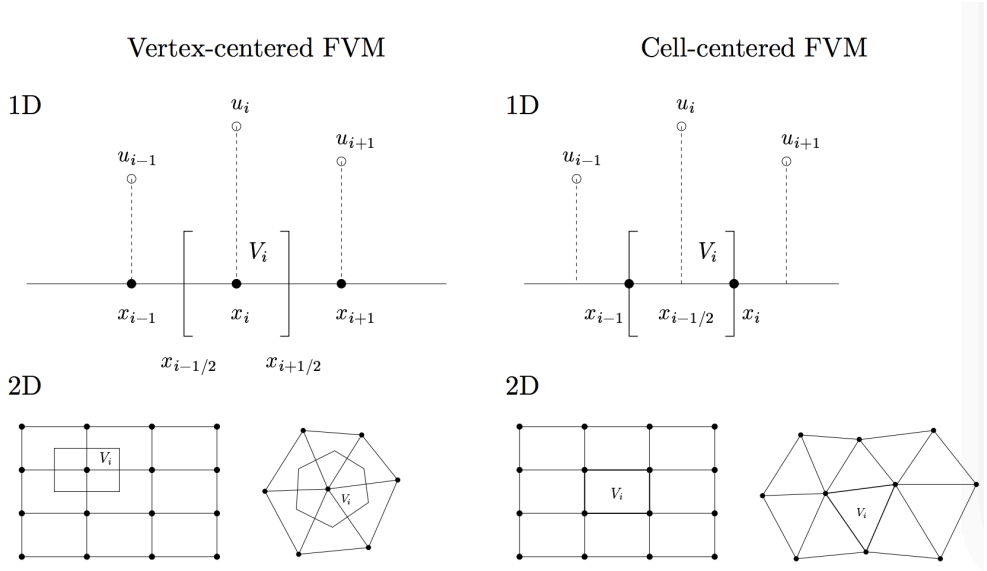


Fig.2.4 Typical control volumes in finite volume method[32]

This method requires a discretization of the governing equation in its integral form. In order to obtain a linear system, integrals must be expressed in terms of mean values. The solution is available only at the CV centers; interpolation is needed to obtain the function values at quadrature points. Common numerical integration rules and interpolation techniques are listed in the table below.

Table 2.1 Numerical integration rules and interpolation techniques

Newton-Cotes quadrature rules		**Interpolation techniques**
1D	2D or 3D	Upwind difference approximation
Midpoint rule Trapezoidal rule Simpson's rule	Midpoint Center of gravity	Central difference approximation Linear upwind difference scheme Quadratic upwind difference scheme

The second-order FVM usually combines the center difference scheme and the linear upwind difference scheme. High-order schemes can be derived from polynomial fitting.

Boundary conditions are applied with certain boundary temperatures or heat fluxes. The finite volume method can handle Neumann boundary condition as well as the Dirichlet boundary condition.

The finite volume method is oftentimes coupled with the front tracking approach for simulating the interface during a phase-change process, especially the solidification modeling[33-36]. Each phase is modeled separately with the use of fitted body coordinates. Two-phase equations are difficult to obtain due to the large variation of the thermal properties across the interface. A fine mesh is necessary to ensure high stability for the numeric simulation. However, even with fine mesh, the front tracking method becomes insufficient with the phase separation, and the fixed-grid approach or the "cut-cell" method is usually applied for better results [37-39].

2.2.3 Other Accelerated CFD Methods

There are adopted CFD methods built upon the conventional finite element method and the finite volume method in industrial applications. These methods usually accelerate the numeric simulations and output results with decent accuracy and stability. Conventional CFD methods use either the Eulerian approach or the Lagrangian approach, where the former one is a mesh-based method and the latter one is a mesh-free method. [24] Some accelerated CFD methods use one of the approaches, such as the reduced order modeling method and the smoothed particle hydrodynamics method. On the other hand, hybrid methods (such as the lattice Boltzmann method) are developed based on

both the Eulerian and Lagrangian approaches to accelerate the mathematical calculation process.

Reduced order modeling seeks to reduce the number of degrees of freedom to shorten the calculation time. The model relies on a basis called the Proper Orthogonal Decomposition (POD), which is used for data pattern analysis.[40] POD is the most popular procedure for extracting spatial basis functions from computational data. Detailed simulation work done for fluid flow and heat transfer can be found in Matthieu Rousset's [41] thesis and other literature[42].

Smoothed Particle Hydrodynamics (SPH) was initially developed by Gingold and Monaghan[43]. It has been a good method in solving multiphase problems. In SPH, the fluid properties are stored in discrete particles and are manipulated using kernel functions. Bockmann [44] and Valizadeh's work[45] showed the implementations of SPH in incompressible inviscid flows with free surfaces and two-phase flow modeling.

Lattice Boltzmann Method (LBM) is based on the Boltzmann equation, which was initially derived by Ludwig Boltzmann. It uses the particle-mesh approach and it models the fluid flow from single-phase to multi-phase with complex geometries such as porous media flows.[46] Complex heat transfer problems were solved using LBM in a lot of literature.[46-49] For instance, the thermal behavior of a droplet on a solid surface was studied by Taghilou, M. and using LBM.[50]

2.3 Phase Change Physical Models

2.3.1 Equivalent Heat Capacity Method

Dariusz Heim used a graph to explain the equivalent heat capacity method, as shown in Fig.2.5.[51]

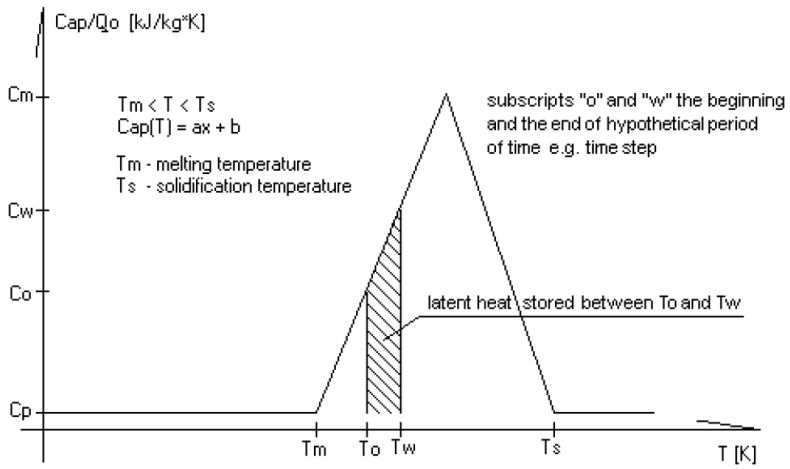


Fig.2.5 Graphical representation of Effective Heat Capacity Method (Figure source: D. Heim, "Two solution methods of heat transfer with phase change within whole building dynamic simulation")

The equivalent heat capacity method assumes that the latent heat of phase change can be treated as a function of the effective heat capacity and an artificial melting temperature range[52]. Since in reality, one substance has one specific melting temperature at a certain situation, the artificial melting temperature range is usually very narrow in order to accurately represent the real situation. The effective heat capacity can be calculated by,[53]

$$C_{p,eq} = \frac{L}{\Delta T} + C_p \qquad 2.1$$

As shown in Eqn.2.1, the effective/equivalent heat capacity is equal to the sum of the ratio between the latent heat and the melting range plus the real heat capacity of the material in its solid form. The effective heat capacity is then utilized in the energy balance equation shown below.

$$\rho_l C_{p,eq} \frac{\partial T}{\partial t} + h_c \nabla T = \nabla(k_l \nabla T) \qquad 2.2$$

Where, h_c is the convection heat transfer coefficient during the melting process. The equation that is used to calculate h_c can be found in Appendix A.

2.3.2 Enthalpy Method

The energy balance can be written in the form of the enthalpy.

$$\rho \frac{\partial h}{\partial t} + \rho \overrightarrow{u} \cdot \nabla h - \nabla \cdot (\alpha \nabla h) + S = 0 \tag{2.3}$$

Where $\vec{u}$ is the velocity vector, α is the thermal diffusivity, and S is a source term. The velocity vector,$\vec{u}$, equals to zero at the solid-liquid interface and also the solid region. The enthalpy method states that the total enthalpy (H) in a system during the phase-change stage can be determined by DSC measurements results. The specific enthalpy and the source term can be calculated by [54, 55]

$$\mathrm{h} = H - \Delta H \tag{2.4}$$

$$S = \rho \frac{\partial \Delta H}{\partial t} \tag{2.5}$$

For a isothermal solidification,

$$\Delta H = \begin{cases} L & T > T_m \\ 0 & T < T_m \end{cases} \tag{2.6}$$

For material with a melting temperature range, the amount of heat released can be calculated by,

$$\Delta H = \begin{cases} L & T > T_l \\ L \cdot f_l & T_l > T > T_s \\ 0 & T < T_s \end{cases} \tag{2.7}$$

Where f_l is the volume fraction of fluid and $f_l(T)$ could be a non-linear function[55].

2.4 Conclusions and Proposal

In summary, numeric simulation analysis relies on efficient mesh generation systems to accompany the geometrical complexity. The quality of the simulation depends strongly on the mesh conditions, in terms of smoothness, regularity, and distortion of the cells. There are several options for the grid geometry and different levels of density when we first meshed the geometry. Robust CFD algorithms will generate reliable results. Typically, reduced-order modeling can shorten the computational time while retaining the model's ability to capture physics of interest. However, a model with complex physical interactions requires more time in the face of high fidelity equation sets.

The simulation model in this study is based on the micro-scale calorimeter prototype created by James L. Armes. A two-dimensional, transient model was previously developed by James to predict the experimental performance of the μDTA prototype. The numeric model has allowed for a geometry upgrade.

In this study, a one-dimensional model was first established in MATLAB. Later on, a three-dimensional model was established in Star CCM+, which is a commercial application with implement of the adaptive finite volume method. To accompany multiphase heat transfer processes, a Eulerian fixed-grid technique called the volume of fraction scheme (VOF) was also introduced into the 3D model. The VOF scheme is based on the enthalpy method and it will ensure that the interfacial energy between the solid and liquid phases can be properly simulated.

The new numeric models will allow for further enhancement in current profile predictions that can ultimately help building the micro-scale calorimeter prototype. The details of the proposed numeric simulation can be found in the subsequent chapters.

3. Investigative Approach

3.1 Governing Equations

3.1.1 Combined Fundamental Differential Equation

The most fundamental laws that our numeric models are built upon are conservation of mass, conservation of energy and conservation of momentum. For a differential control volume, all the variables are calculated on a control volume basis by using the general differential equation proposed by S. V. Patankar[56].

$$\frac{\partial}{\partial t}(\rho\phi) + \nabla \cdot (\rho\vec{u}\phi) = \nabla \cdot (\Gamma\nabla\phi) + S \qquad 3.1$$

In Eqn 3.1, the velocity vector $\vec{u}$ = (u, v, w). The first term $\frac{\partial}{\partial t}(\rho\phi)$ represents the time rate of change of properties within the control volume, $\nabla \cdot (\rho\vec{u}\phi)$ represents the advection transport across the boundaries of the control volume, $\nabla \cdot (\Gamma\vec{u}\phi)$ is the diffusion term, and S is the source term.

When using different conservation equations, ϕ is a property and Γ is the diffusion coefficient. For example, in the conservation of mass equation, ϕ is equal to 1 and S equals 0. Thus, Eqn. 3.1 can be simplified to

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho\vec{u}) = 0 \qquad 3.2$$

The simplified conservation of energy and momentum equations are shown below.

$$\nabla \cdot (\rho\vec{u} \cdot h) = \nabla \cdot (\mathrm{k}\nabla T) + S \qquad 3.3$$

$$\frac{\partial}{\partial t}(\rho\mathrm{u}) + \nabla \cdot (\rho\vec{u}u) = \nabla \cdot (\mu\nabla u) + S$$

$$\frac{\partial}{\partial \mathrm{t}}(\rho\mathrm{v}) + \nabla \cdot (\rho\vec{\mathrm{u}}\mathrm{v}) = \nabla \cdot (\mu\nabla \mathrm{v}) + \mathrm{S}$$

$$\frac{\partial}{\partial t}(\rho w) + \nabla \cdot (\rho \vec{u} w) = \nabla \cdot (\mu \nabla w) + S \qquad 3.4$$

3.1.2 The Discretization Equation for Heat Conduction

A discretization equation is an algebraic relation that the numeric model follows to solve values for a group of node points. To derive the right discretization equation for the numeric model, let's consider a steady one-dimensional heat conduction problem first. Set ϕ equal to T and Γ equal to k in the general differential equation. Eqn. 3.1 then becomes,

$$\frac{d}{\partial x}(k\frac{dT}{dx}) + S = 0 \qquad 3.5$$

To discretize Eqn3.5, we shall employ the grid-point structure shown in Fig 3.1.

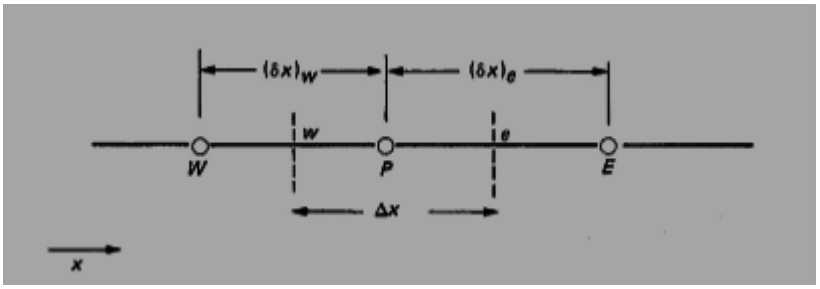


Figure 3.1 Grid-point structure for one-dimensional discretization equation (S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Taylor & Francis, 2014, p. 32.)

If we multiply dx on both sides of the equation 3.5 and then take an integral, it will become

$$\int_w^e d(k\frac{dT}{dx}) + \int_w^e S\,dx = 0 \qquad 3.6$$

If we approximate the value of $\frac{dT}{dx}$ using a linear profile, Eqn. 3.6 can be written as

$$\frac{k_e(T_E - T_P)}{(\delta x)_e} - \frac{k_w(T_P - T_W)}{(\delta x)_w} + \bar{S}\Delta x = 0 \qquad 3.7$$

where $\bar{S}$ is the average value of S over the control volume. Notice that for an infinitesimal change in x, the integral S can be evaluated by,

$$\int \mathrm{S(x)dx} \cong \bar{\mathrm{S}}\Delta\mathrm{x} \tag{3.8}$$

Rearrange Eqn. 3.7, the final form of the discretization equation becomes,

$$a_P T_P = a_E T_E + a_W T_W + b \tag{3.9}$$

Where,

$$a_E = \frac{k_e}{(\delta x)_e} \tag{3.10}$$

$$a_W = \frac{k_w}{(\delta x)_w} \tag{3.11}$$

$$a_P = a_E + a_W \tag{3.12}$$

$$b = \bar{S}\Delta x \tag{3.13}$$

The thermal conductivity k_e and k_w can be evaluated at the e or w axis shown in Fig 3.1.

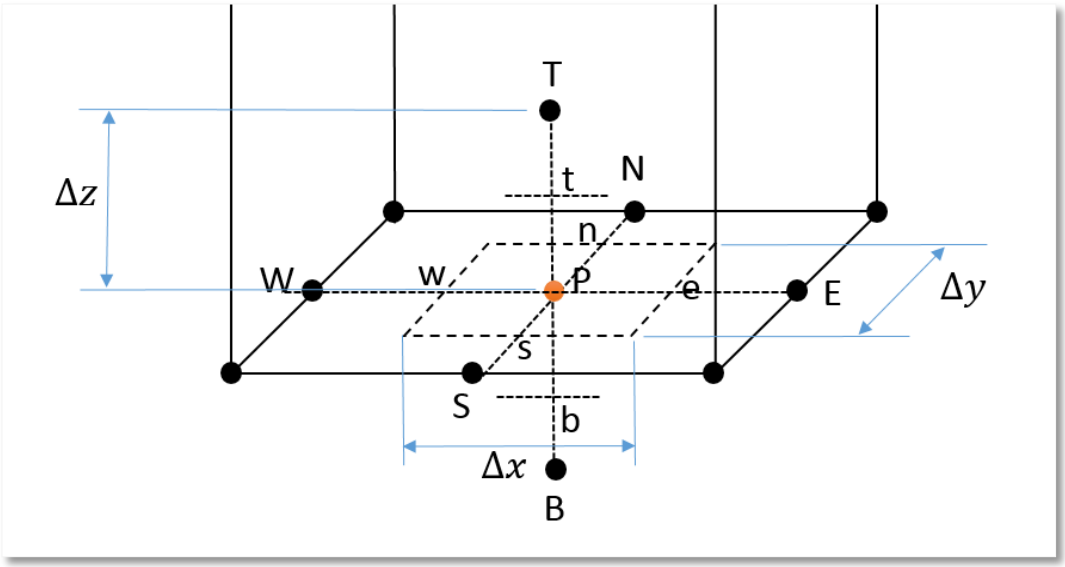


Figure 3.2 Grid-point structure for three-dimensional discretization equation

The same technique used to derive the one dimensional discretization equation can also be used to derive the three dimensional equation. As shown in Fig 3.2, by adding

the south, north, top and bottom direction, the general differential equation can be written as

$$a_P T_P = a_E T_E + a_W T_W + a_N T_N + a_S T_S + a_T T_T + a_B T_B + b \quad 3.14$$

$$a_P = a_E + a_W + a_N + a_S + a_T + a_B + a_P^0 - S_P \Delta x\, \Delta y\, \Delta z \quad 3.15$$

$$b = S_C \Delta x\, \Delta y\, \Delta z + a_P^0 T_P^0 \quad 3.16$$

$$a_E = \frac{\mathrm{k}_e \Delta y\, \Delta z}{(\delta x)_e} \quad 3.17$$

$$a_W = \frac{k_w \Delta y\, \Delta z}{(\delta x)_w} \quad 3.18$$

$$a_N = \frac{k_n \Delta x\, \Delta z}{(\delta y)_n} \quad 3.19$$

$$a_S = \frac{k_s \Delta x\, \Delta z}{(\delta y)_s} \quad 3.20$$

$$a_T = \frac{k_t \Delta y\, \Delta z}{(\delta x)_t} \quad 3.21$$

$$a_B = \frac{k_b \Delta y\, \Delta z}{(\delta x)_b} \quad 3.22$$

$$a_P^0 = \frac{\rho c\, \Delta x\, \Delta y\, \Delta z}{\Delta t} \quad 3.23$$

3.1.3 Tri-diagonal Matrix Algorithm (TDMA)

Suppose that we have a one-dimensional structure shown below,

Figure 3.3 Internal and boundary node points

For the convenience of presenting the algorithm, we will re-write Eqn. 3.9 to be,

$$a_n T_n = b_n T_{n+1} + a_n T_{n-1} + d_n \quad 3.14$$

Rearrange Eqn. 3.13, for each node n,

$$0 = c_n T_{n-1} - a_n T_n + b_n T_{n+1} + d_n \quad 3.15$$

In order to solve the temperature for each node, we will need to solve the matrix equation,

$$[0] = [A][T] + [d] \quad 3.16$$

Where,

$$[A] = \begin{bmatrix} c_1 & -a_1 & b_1 & 0 & \cdots & 0 \\ 0 & c_2 & -a_2 & b_2 & \cdots & 0 \\ 0 & 0 & c_3 & -a_3 & \cdots & 0 \\ \vdots & & & & & \vdots \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & c_N & -a_N & b_N \end{bmatrix} \quad 3.17$$

$$[T] = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \vdots \\ \vdots \\ T_N \end{bmatrix} \quad 3.18$$

$$, [d] = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ \vdots \\ d_N \end{bmatrix} \quad 3.19$$

If we write the current node temperature in term of the next node,

$$T_n = P_n T_{n+1} + Q_n \quad 3.20$$

$$T_n = \frac{b_n}{a_n - c_n P_{n-1}} T_{n+1} + \frac{c_n Q_{n-1} + d_n}{a_n - c_n P_{n-1}} \quad 3.21$$

Where,

$$P_1 = \frac{b_1}{a_1} \quad 3.22$$

$$, Q_1 = \frac{d_1}{a_1}, \quad 3.23$$

$$P_n = \frac{b_n}{a_n - c_n P_{n-1}} , \quad 3.24$$

$$Q_n = \frac{c_n Q_{n-1} + d_n}{a_n - c_n P_{n-1}} \quad 3.25$$

In summary, the first step of the tridiagonal-matrix algorithm (TDMA) is to calculateP_1 and Q_1. Then, one shall use Eqn. 3.22 to calculate P_n and Q_n for n=1, 2, 3 …,

N. Next, use boundary condition equations to solve T_N and set $T_N = Q_N$. And then use the sequence formula Eqn. 3.20 to calculate $T_{N-1}, T_{N-2}, \ldots, T_1$.

3.2 Numeric Model Development

3.2.1 One-Dimensional Heat Conduction Model Geometry

Thermal simulations of heat conduction have been preformed using MATLAB (MathWorks, R2009b) and Star CCM+ (CD-Adapco, STAR-CCM+9.06.011-R8). The continuity equation as well as the momentum and energy equations for the process has been solved in MATLAB and Star CCM+ using TDMA and finite volume methods respectively. The thermal simulations involve two kinds of structures; a single water cube and a multiple layered structure as shown in Fig. 3.4.

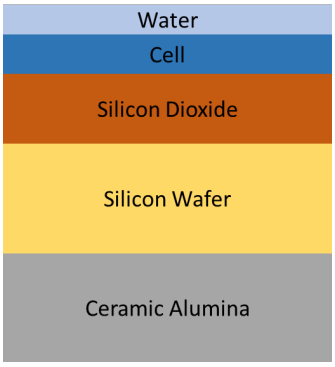


Figure 3.4 1D multi-layer geometry utilized in the construction of numeric model

A multi-layer solid region was created in Star CCM+. However, within the region, there was only heat conduction in the z direction as shown in Fig. 3.5.

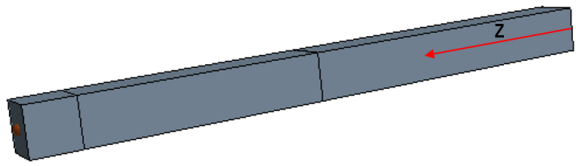


Figure 3.5 Multi-layer geometry utilized in Star CCM+

For the multi-layered geometry, the dimensions and thermal properties of different materials are listed in Table 3.1.

Table 3.1 Dimensions and thermal properties of different materials

Material Name		**Thickness** (mm)	**Heat Capacity** $C_p\ (J/kg \cdot K)$	**Thermal Conductivity** $k\ (W/m \cdot K)$	**Density** $\rho\ (kg/m^3)$
Water/cell	Solid	0.01	2050	2.22	916
	Liquid		4186	0.561	999.8
Silicon Wafer		0.5	745	1.38	2220
Silicon Dioxide		0.3	745	1.38	2220
Ceramic Alumina		0.68	77.5	35	3750

3.2.2 Boundary Condition

There are two types of boundary conditions assigned to the numeric models; constant temperature and constant heat flux boundary conditions. Each of the boundary conditions requires different values for coefficients used in the TDMA code written in

MATLAB. And each of the boundary conditions requires different physics conditions set up when building the regions in Star CCM+.

In MATLAB, for a constant temperature boundary condition applied to the beginning of the structure, the parameters for the first node point in TDMA can be set as shown below.

$$P_1 = 0\,, Q_1 = T_w\,,\ a_1 = 1\,, b_0 = 0\,, d_0 = T_w$$

If a constant temperature boundary condition is applied to the last node point,

$$P_N = 0\,, Q_N = T_E\,,\ a_N = 1\,, c_N = 0\,, d_N = T_E$$

For a constant heat flux boundary condition shown in Fig. 3.6,

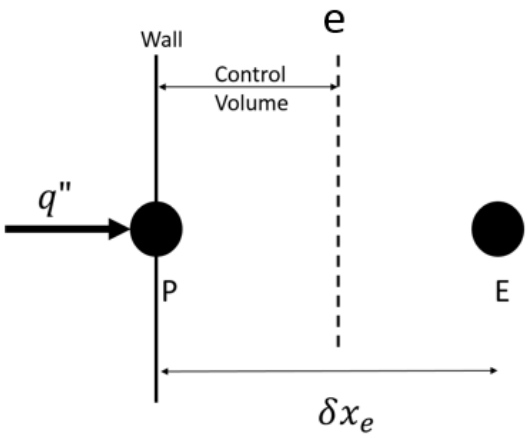


Figure 3.6 Half control volume near the boundary

Based on the energy balance equation,

$$\frac{k_e}{(\delta x)_e} T_2 + q^{"} + \bar{S}\Delta x = \frac{k_p}{(\delta x)_p} T_1 \qquad 3.26$$

Thus,

$$a_1 = \frac{k_p}{(\delta x)_p}\,, b_1 = \frac{k_e}{(\delta x)_e}\,, c_1 = 0, d_1 = q^{"} + \bar{S}\Delta x$$

Where k_e is evaluated at the e axis shown in Fig 3.6 and k_p is evaluated at the material boundary.

In Star CCM+, the boundary type can be selected to be wall, stagnation inlet and so on. To provide more information on the boundary condition, a user can apply a specified temperature or a specified heat flux thermal boundary condition.

3.2.3 Source Term

In the TDMA algorithm, when the heat source remains constant, it will be input as $\bar{S}$ in the discretization equation. When the heat source is a function of temperature, it can be linearized in the form of

$$S = S_c + S_p T_p \tag{3.27}$$

The value of S_c and S_p may depend on the temperature and there are recalculated during each iteration cycle. There are different values one can assign to S_c and S_p, but one should always choose the linearization of S that is a good representation of the $S\ (T)$ function and the slope of the linear relation should always remain negative.

For a phase change problem, the energy balance equation becomes

$$\rho C_p \frac{dT}{dt} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) - \rho L \frac{df_l}{dt} \tag{3.28}$$

The heat source term should include both heat released due to phase change and heat loss due to an increase in temperature. The heat source term for a single cell for an iteration cycle is

$$S = \rho C_p T_p^0 + \rho L \Delta x \Delta f_l \tag{3.29}$$

Since the fluid fraction f_l at a certain temperature T is

$$f_l = \begin{cases} 0 & T < T_s \\ \dfrac{T - T_s}{T_l - T_s} & T_s < T < T_l \\ 1 & T > T_l \end{cases} \tag{3.30}$$

Using Eqn 3.27, we can split our heat source term into S_c and S_p as shown below.

$$S_c = \rho C_p T_p^0 + \frac{\rho L T_p^0}{T_l - T_s} \quad 3.31$$

$$S_p = -\frac{\rho L}{T_l - T_s} \quad 3.32$$

The heat source terms S_c and S_p will be added to the a_P term and the b term in Eqn. 3.11 and Eqn. 3.12, respectively, for each control volume. The new a_P and b value can be computed by

$$a_P = a_E + a_W + a_P^0 - S_p \Delta x \quad 3.33$$

$$b = S_c \Delta x + a_P^0 T_P^0 \quad 3.34$$

Also, the thermal properties in the energy balance equation will be calculated based on the fluid fraction.

$$k = f_l \cdot k_l + (1 - f_l) k_s \quad 3.35$$

$$\rho = f_l \cdot \rho_l + (1 - f_l) \rho_s \quad 3.36$$

$$C_p = f_l \cdot C_{pl} + (1 - f_l) C_{ps} \quad 3.37$$

To simulate the phase change process in Star CCM+, a Volume of Fluid (VOF) model has been chosen from the Eulerian Multiphase Model group. This model allows the users to set the liquidus temperature and solidus temperature as well as other thermal properties; including the specific latent heat of fusion as shown in Fig 3.7.

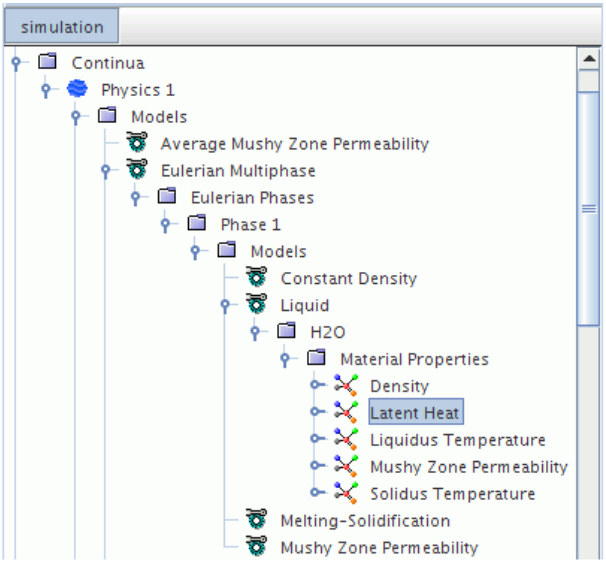


Figure 3.7 Melting-Solidification Simulation Window in Star CCM+

3.2.4 Thermal Properties in Composite Materials

The thermal properties of each node are stored in matrix [A] and [d] in Eqn. 3.15. For a multi-layer structure, thermal properties could change between two nodes. In order to derive a formula to calculate the equivalent thermal property, let's consider a one-dimensional node structure like Fig.3.8.

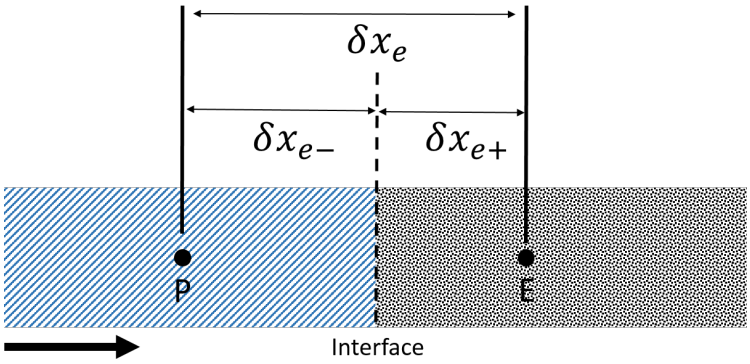


Figure 3.8 Composite Material Node Point Structure

Between node P and E, the length of material A (on the left hand side) is δx_{e-} and the length of material B (on the right hand side) is δx_{e+}. Therefore, between node P and E,

$\frac{\delta x_{e-}}{\delta x_e}$ percent of the material is A and $\frac{\delta x_{e+}}{\delta x_e}$ percent is B. Based on the energy balance at the interface, the equivalent thermal property X_{eq} can be calculated by

$$X_{eq} = \frac{X_A X_B}{\frac{\delta x_{e+}}{\delta x_e} X_A + \frac{\delta x_{e-}}{\delta x_e} X_B} \tag{3.38}$$

3.3 Mesh Refinement Study

3.3.1 MATLAB

A mesh independence study has been conducted using a steady-state one-dimensional conduction model. The three mesh densities are listed in Table 3.2.

Table 3.2 Mesh information for TDMA code developed in MATLAB

Mesh Density	Mesh Size (m)	Number of Nodes
Coarse	10^{-3}	700
Base Line	10^{-4}	7000
Fine	10^{-5}	70000

The temperature distribution results for different mesh densities have been compared at the same locations of the solid; the beginning of the solid, 1/7 length of the solid, 3/7 length of the solid, 5/7 length of the solid and the end of the solid. Average temperature differences have been calculated between one mesh and the previous one with lower numbers of nodes.

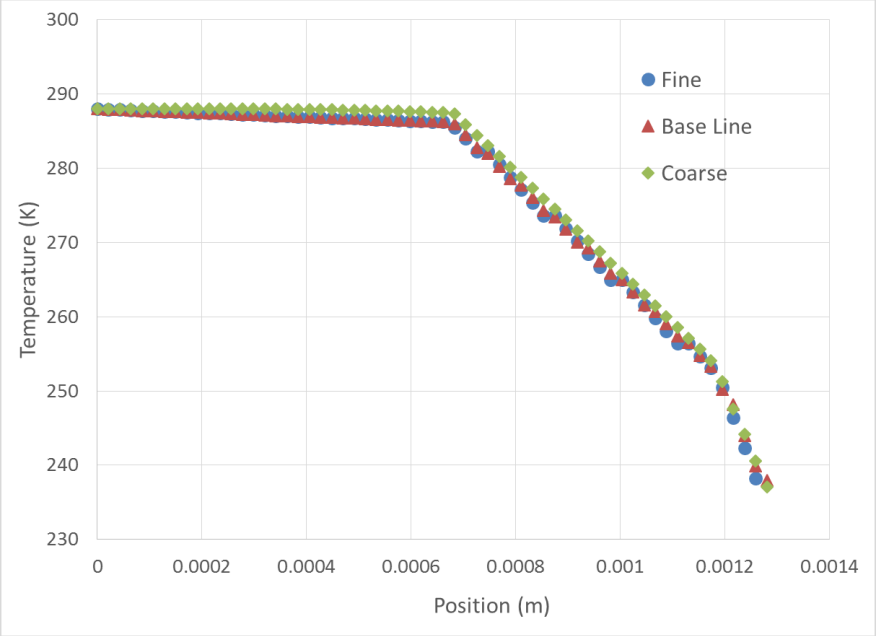


Figure 3.9 Temperature vs position for different mesh sizes

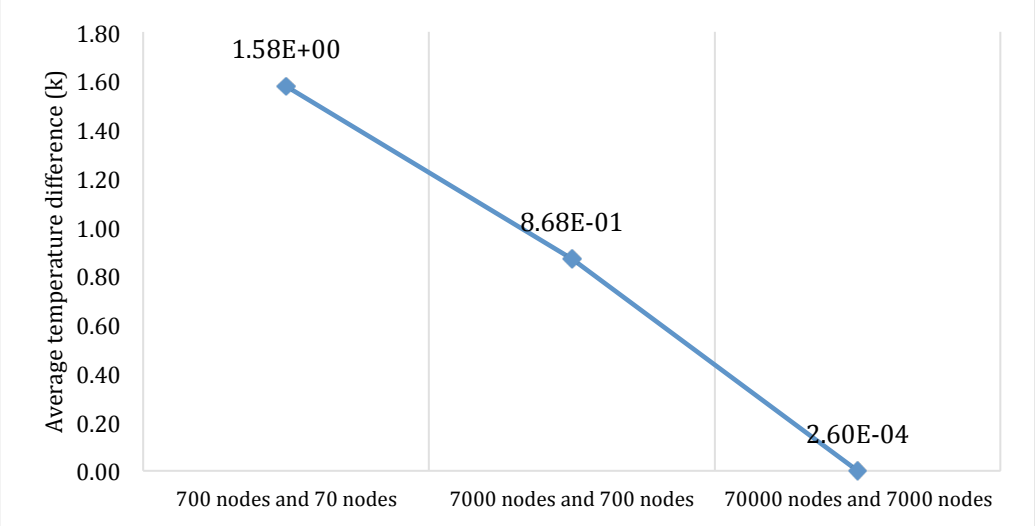


Figure 3.10 Average temperature differences between two mesh densities

3.3.2 Star CCM+

A mesh independence study has been conducted using the same transient one-dimensional conduction model that previously used in MATLAB. Detailed mesh information for numeric models developed in Star CCM+ is listed in Table 3.3.

Table 3.3 Mesh information for numeric models developed in Star CCM+

Mesh Density	Mesh Size (m)	Number of Cells
Coarse	1.28×10^{-5}	6,528
Base Line	6.14×10^{-6}	27,136
Fine	1.28×10^{-6}	6,096,198

Average temperature difference has been calculated between one mesh and the previous one with lower numbers of nodes. Based on the information in Table 3.4, we decide to use the fine mesh with the mesh size being 1.28×10^{-6} m.

Table 3.4 Mesh densities and the relative temperature differences

Mesh Density	Energy Residual (K)	Average Temperature Difference (K) Compare with Base Line
Coarse	1.17×10^{-14}	-0.87
Base Line	3.84×10^{-14}	0
Fine	8.98×10^{-06}	0.044

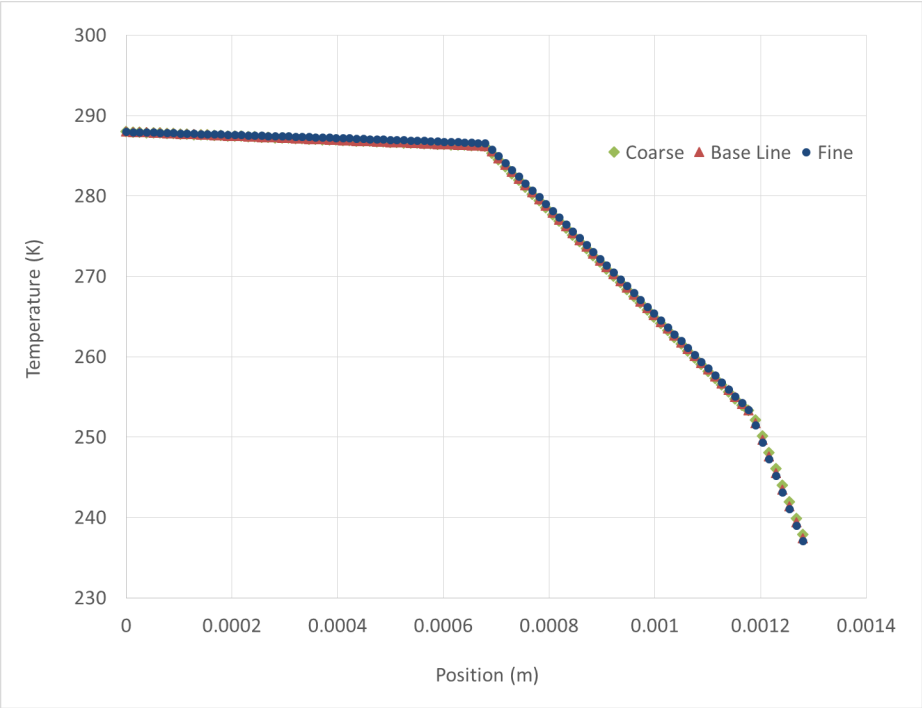


Figure 3.11 Temperature vs position for different mesh sizes

3.4 Time Step Study

3.4.1 MATLAB

Given the temperature of a cell at time t, a fully implicit scheme is utilized to calculate the temperature at t+Δt. The implicit scheme assumes the temperature at t, T_p^0, drops suddenly to temperature T_p^1 at t+Δt and stays at that value.

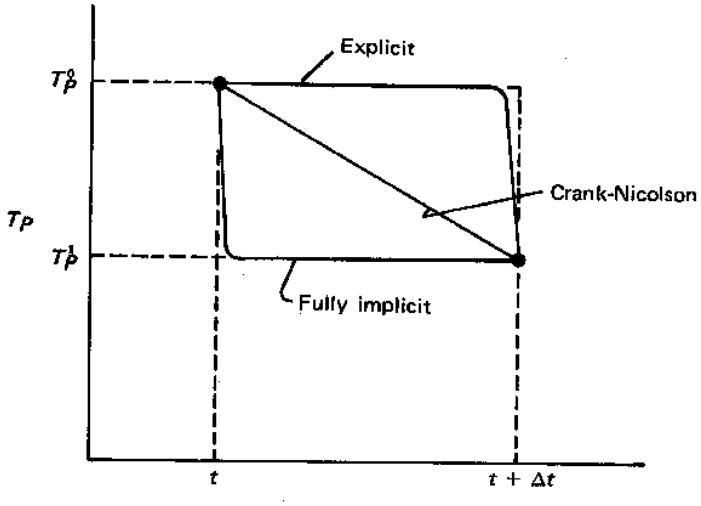


Figure 3.12 Variation of temperature with time for three different schemes (S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Taylor & Francis, 2014, p. 56.)

Using the fully implicit scheme, the values of a_P and b were re-calculated using different formulas.

$$a_P = a_E + a_W + a_p^0 - S_p\Delta x \quad 3.39$$

$$b = S_c\Delta x + a_p^0 T_p^0 \quad 3.40$$

Where,

$$a_p^0 = \frac{\rho c\Delta x}{\Delta t} \quad 3.41$$

It can be seen that as Δt goes to infinity, the value of a_P and b agree with the steady-state calculation results.

In order to make sure that the time step Δt would be as large as possible to reduce the computation time while also small enough to provide accurate and realistic results, a time step independence study has been conducted using a one-dimensional symmetric transient conduction model. This model has symmetric geometry and uniform thermal properties. Three different time-step values were chosen and the temperature at the 1/12, 1/6, and 1/4 lengths of the geometry have been compared after the simulations are converged.

Table 3.5 Time step information for TDMA code developed in MATLAB

Step Size	Time Step (s)	Temperature (K)		
		1 / 12 Length	1 / 6 Length	1 / 4 Length
Big	4×10^{-3}	16.8368	23.5623	24.8438
Base Line	4×10^{-4}	16.5645	23.6076	24.8924
Small	4×10^{-5}	16.5363	23.6122	24.8978
Very Small	4×10^{-6}	16.5334	23.6126	24.8984

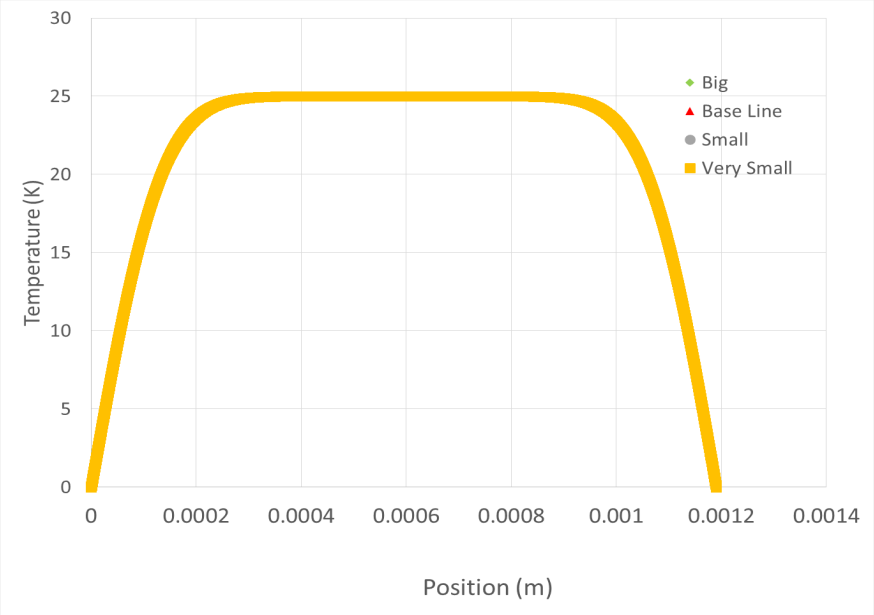


Figure 3.13 Temperature at 0.4 seconds for different time steps

3.4.2 Star CCM+

In Star CCM+, four different time steps have been tested using the same transient numeric model previously run in MATLAB. Temperature profiles were compared. The results are listed in Table 3.6 below. Based on the results, the time step was set to be 4×10^{-5} second.

Table 3.6 Time step information for numeric models developed in Star CCM+

Time Step Test	Time Step (s)	Average Temperature Difference (K)	Energy Residual
Big	4×10^{-3}	3.60×10^{-2}	2.97×10^{-13}
Base Line	4×10^{-4}	0	1.14×10^{-15}
Small	4×10^{-5}	4.0×10^{-3}	2.74×10^{-15}
Very Small	4×10^{-6}	3.65×10^{-4}	2.02×10^{-14}

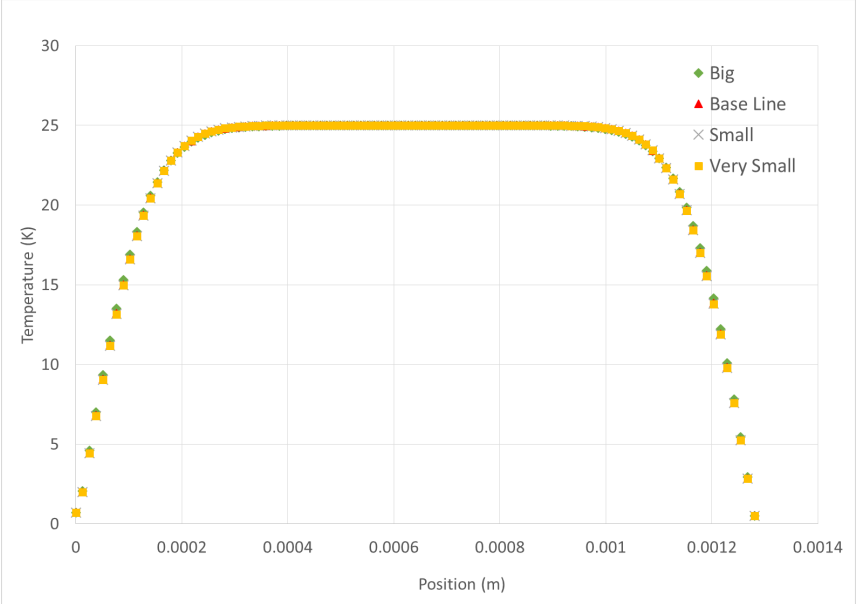


Figure 3.14 Temperature vs position for different time steps

3.5 Analytical Validation

3.5.1 Steady One-Dimensional Heat Conduction

The numeric model developed in MATLAB has been tested under a steady-state heat conduction condition. The bottom of the structure was set at a constant temperature of 288 K and the top of the structure was at 237 K constantly.

Figure 3.15 Boundary conditions of the one-dimensional heat conduction scenario

The same boundary conditions were applied to the geometry in Fig. 3.4 as in Star CCM+. The physical model of the geometry was set to be constant density and segregated energy.

The analytical solution can be obtained by solving,

$$\frac{d^2T}{dx^2} = -S \tag{3.42}$$

The exact solution of equation 3.39 is,

$$T = -\frac{S}{2}x^2 + C_1x + C_2 \tag{3.43}$$

Where C_1 and C_2 are constants which can be computed given the source term value and two boundary conditions. After solving Eqn. 3.39, temperature computed using the numeric method and the analytical method were compared.

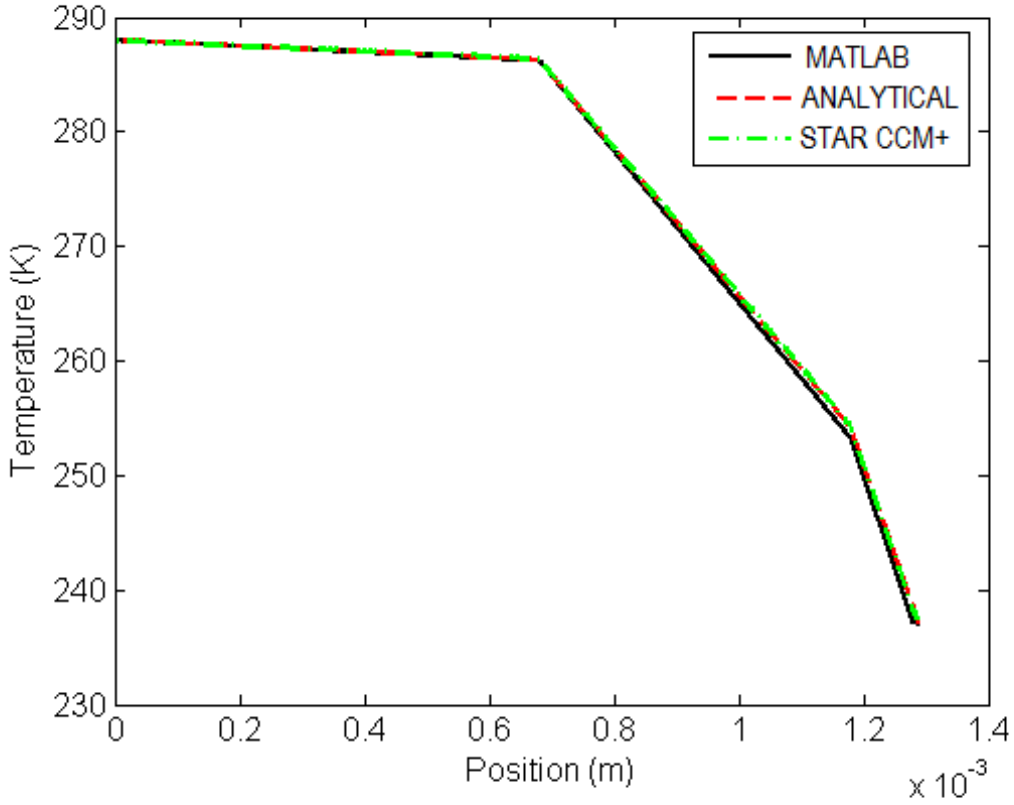


Figure 3.16 Numeric temperature profile and analytical result for one-dimensional conduction

The results obtained from all three methods agree with each other. The temperature difference between the MATLAB model results and the analytical solution is less than 0.34 K.

3.5.2 One-Dimensional Transient Heat Conduction

In order to further test the numeric model developed in MATLAB, a transient heat conduction problem was simulated using the geometry and boundary conditions shown below.

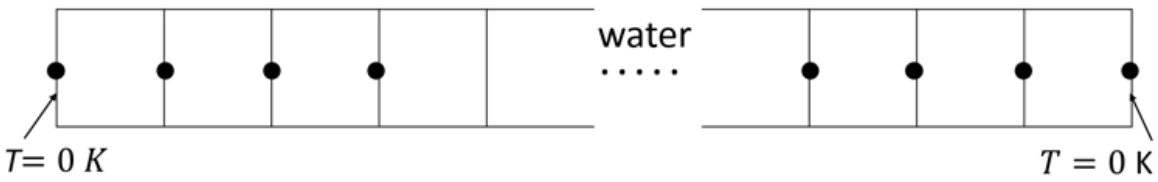


Figure 3.17 Geometry set up of the transient heat conduction scenario

This transient one-dimensional conduction problem is governed by the equation,

$$\frac{1}{\alpha}\frac{\partial T}{\partial t} = \frac{d^2T}{dx^2} \tag{3.44}$$

Based on M. Necati Özişik's separation of variables method[57], the solution of Eqn.3.44 can be approximated using the formula,

$$T(x,t) = \sum_{m=1}^{\infty} K(\beta_m, x) \cdot e^{-\alpha\beta_m^2 t} \cdot \int_{x'=0}^{L} K(\beta_m, x') \cdot F(x') \cdot dx' \tag{3.45}$$

Where, K and β are different constants that depend on how big the number of elements, m, is. More detailed formulas can be found in Appendix B. This study used 10 terms to calculate the analytic values of the temperature. The element size is 1×10^{-7}m, the time step is 4×10^{-5} second, the boundary temperature stays constantly at 0 K, and the initial temperature is at 25 K. The numeric simulation results and the analytical results are plotted in the figure below. The biggest temperature difference among the three models is about 0.2 K.

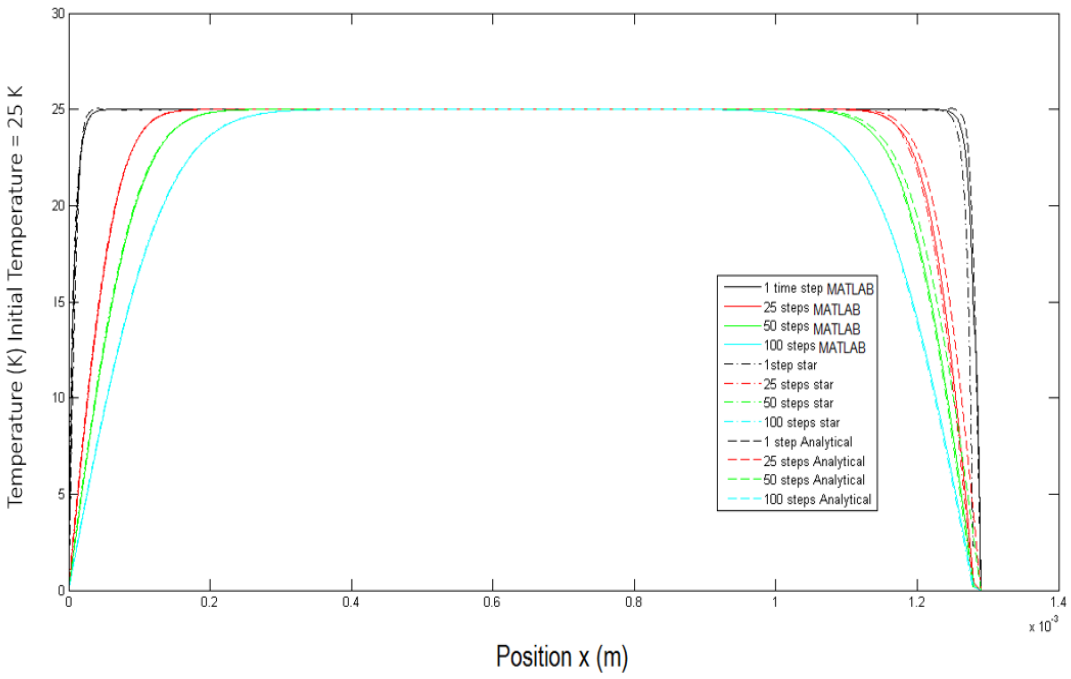


Figure 3.18 Transient heat conduction temperature distribution at different time

3.5.3 Semi-infinite Solid Melting Problem

In order to examine the numeric model for phase-change problems, the numeric models developed in MATLAB and Star CCM+ were used to calculate the temperature distribution of a semi-infinite solid during a melting process. Numeric results obtained from the TDMA algorithm and Star CCM+ were compared with the exact analytical solution and a second-order approximation solution.

Fig.3.18 shows the expected behavior at a certain moment for the one-dimensional melting problem in which the solid was initially at a temperature lower than the fusion temperature. A constant temperature boundary condition that is higher than the freezing point was applied at x=0.

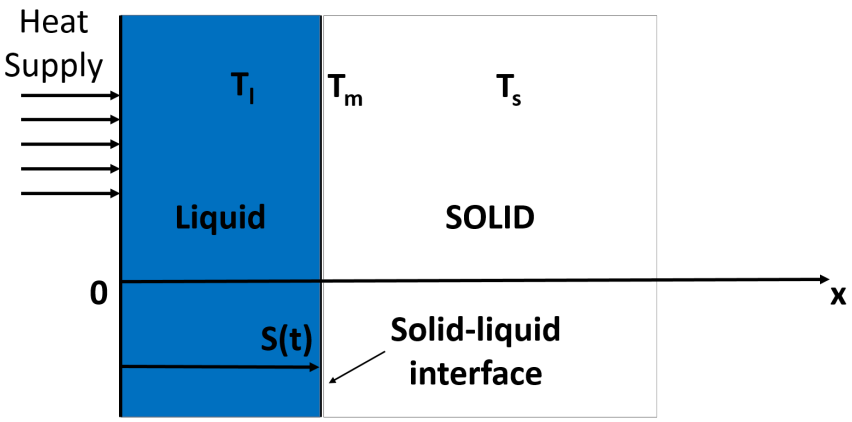


Figure 3.19 Sketch of semi-infinite solid melting problem

For the purpose of convenience, the fusion temperature for water was set to be 273.15 K, the temperature at x=0 was 278 K and the initial temperature of the region was 273.145 K. Since the thermal properties of water vary with phase and temperature, the volume of fraction method was used to determine the thermal properties of water. The thermal properties of the liquid and solid phase can be found in Table 3.1.

In order to understand the derivation of the exact solution and the approximated solution, let's first consider the governing physics equation for this phase-change problem,

$$\rho\, C_p \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} \tag{3.46}$$

The boundary conditions are

$$T(0,t) = T_0$$

$$T(s,t) = T_m$$

$$-k \frac{\partial T}{\partial x}\bigg|_{x=s} = \rho L \frac{ds}{dt}$$

The heat transfer problem above is also called the Stephan Problem, where the solution of this specific Stephan Problem can be obtained by solving a transcendental equation,

$$\frac{St_l}{e^{\lambda^2}\operatorname{erf}(\lambda)} - \frac{St_s}{ve^{(v\lambda)^2}\operatorname{erfc}(v\lambda)} = \lambda\sqrt{\pi} \tag{3.47}$$

$$St_l = \frac{c_l(T_l - T_m)}{L}$$

$$St_s = \frac{c_s(T_m - T_s)}{L}$$

$$v = \sqrt{\frac{\alpha_l}{\alpha_s}}$$

Where, λ is the root of the transcendental equation 3.47. The location of the interface $s(t)$ between the solid region and the liquid region is

$$s(t) = 2\lambda\sqrt{\alpha_l t} \tag{3.48}$$

The approximated temperature profile of the solid region and the liquid region was proposed by Vasilios Alexiades[58].

$$T(x,t) = T_s + (T_m - T_s)\frac{erfc(\frac{x}{2\sqrt{\alpha_s t}})}{erf\ c(\lambda\sqrt{\frac{\alpha_l}{\alpha_s}})} \tag{3.49}$$

$$T(x,t) = T_l - (T_l - T_m)\frac{erf\ (\frac{x}{2\sqrt{\alpha_l t}})}{erf\ (\lambda)} \tag{3.50}$$

In MATLAB, a mesh as shown in Fig 3.20 was established to solve the problem mentioned in Fig 3.19. Fig 3.20 also shows the boundary conditions that were applied at this MATLAB model.

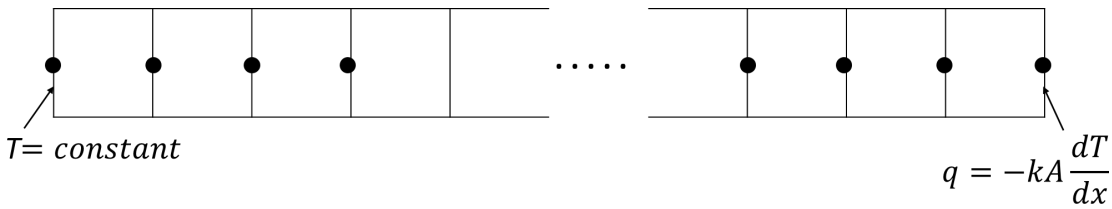


Figure 3.20 Sketch of the mesh and boundary conditions for phase-change problem

In order to solve this melting problem in MATLAB, a modified implementation of integral method derived by M. Necati Özişik[59] was utilized in addition to the TDMA algorithm as another analytical method. The integral method was initially derived by Goodman[60] to estimate the temperature in a liquid region and the method assumes that the temperature remains constant in a solid region. By integrating the differential equation of heat conduction over the thermal-layer thickness, which is s(t), the heat-balance integral equation for the melting problem becomes,

$$-\alpha\left[\frac{\rho L}{k}\frac{ds(t)}{dt}+\frac{\partial T}{\partial x}\Big|_{x=0}\right]=\frac{d}{dt}\left[\left(\int_0^{s(t)} T\cdot dx\right)-T_m\cdot s(t)\right] \qquad 3.51$$

To simplify the calculation, Özişik chose a second-degree polynomial to approximate the temperature profile in the form of

$$T-T_m=B_1(x-s)+B_2(x-s)^2 \qquad 3.52$$

Where B_1, B_2 and s are equations of thermal properties, initial temperature and boundary temperature. The formulas that were used to determine the value of those coefficients can be found in Appendix C. The 2nd approximation approach can only roughly predict the temperature profile trend before the liquid-solid interface and the location of the liquid-solid interface.

Those two analytical methods mentioned above have been compared in Özişik's book. The exact solution and the 2nd order polynomial output really similar predictions of the temperature profile of the substance as well as the liquid-solid interface location. In Özişik's second-degree approximation method, the root λ is calculated using

$$\lambda=\left[3\frac{1-(1+\mu)^{\frac{1}{2}}+\mu}{5+(1+\mu)^{1/2}+\mu}\right]^{1/2} \qquad 3.53$$

$$\mu=\frac{2k}{\alpha\rho L}\,(T_0-T_m)=\frac{2\,C_p(T_0-T_m)}{L} \qquad 3.54$$

The approximated λ is used in Eqn 3.48 to calculate the approximated liquid-solid interface location. The figure below shows the amount of deviation of the interface location relative to the exact solution as the physical time proceeds. It also shows the comparison of the exact interface location and the calculated interface location using Özişik's second-degree approximation method.

Interface Location Comparison

Between Exact Solution and Ozisik's Second-Degree Approximation

Exact Solution

$s(t) = 0.33606\sqrt{\alpha_L t}$

Second-Degree Approximation

$s(t) = 0.33524\sqrt{\alpha_L t}$

Physical time(S)	Deviation in Interface Location(mm)/% relative to Exact Solution	
1	0.000302	0.24%
60	0.002342	0.24%
3600	0.018144	0.24%

Exact Solution (mm)	2nd-Degree Approximation (mm)
0.123933	0.123630
0.959979	0.957637
7.435968	7.417824

Figure 3.21 Interface location comparison

As time proceeds, the difference between the exact solution and the second-degree approximation becomes larger. However, the percentage of deviation is determined by the λ used in the exact solution and the λ calculated from Eqn 3.53. Therefore, the percentage of deviation always remains constant once the two λ values are determined.

The same semi-infinite solid melting problem is also simulated using Star CCM+. A solid region was imported into the Star CCM+ model. The region was set to be symmetric along the x and y direction. There is only heat diffusion along the z axis. Since it is a transient model, the time is set to be implicitly unsteady. The physical properties of the region are calculated based on the volume of fluid method.

Figure 3.22 shows the temperature distribution results from the MATLAB model, the Star CCM+ simulation, and the exact solution equations.

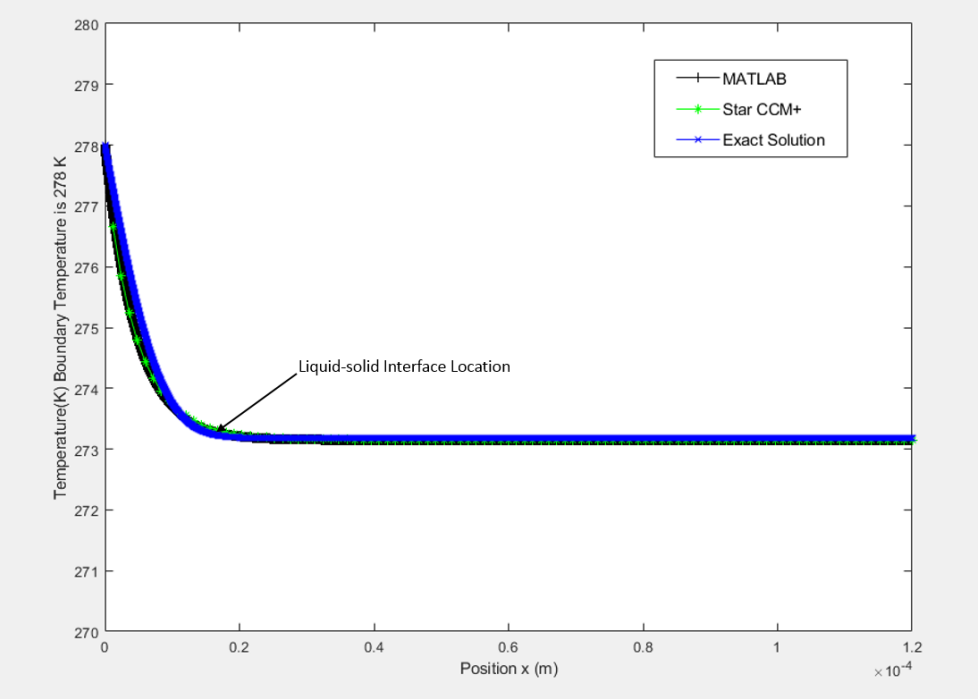


Figure 3.22 Temperature results at t = 0.0175 second for the phase-change problem

From Fig.3.22, it can be observed that melting starts at x=0 and the solid-liquid interface moves in the positive direction. Temperature decreases in the liquid region as the position moves away from x=0. After the position passed the liquid-solid interface, the temperature remains constant. The MATLAB model gives out a fairly accurate prediction on temperature in fluid region. Results from the Star CCM+ simulation agree with the exact solution.

4. Numeric Model

4.1 Geometry Discretization

As shown in Fig 4.1, a water droplet or the sample cell is dropped at the middle of the thermoelectric module for the physical experiment. The numeric model is mainly focused on simulating the heat transfer in the vertical direction of the water droplet or cell sample that sits on top of the thermoelectric module. Since the RTD thickness is very thin, it was neglected in the model geometry.

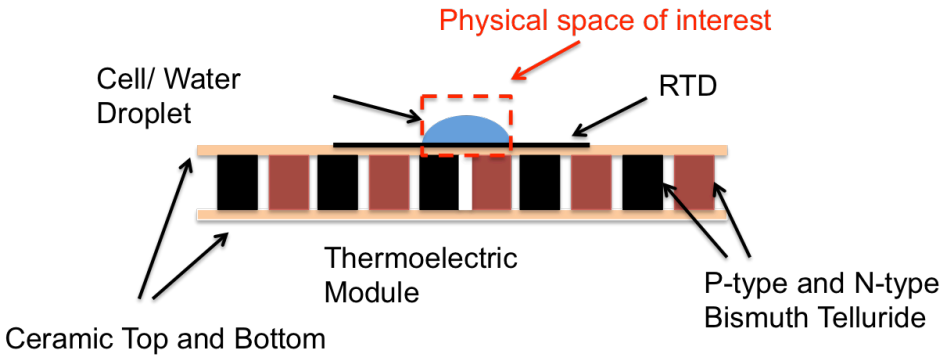


Figure 4.1 Two-dimensional Geometry of the experimental setup

A Parasolid file were generated in SolidWorks and then imported into Star CCM+. The Parasolid file has a water cube on the top and a layer of the ceramic surface of the thermoelectric module at the bottom as shown in Fig 4.2(a) The bottom area of the water cube equals to the ceramic layer surface on the top. In order to see how much of the ceramic layer is accounted for the numeric model, Fig 4.2(b) shows a water cube sitting on a real-scale TE ceramic layer.

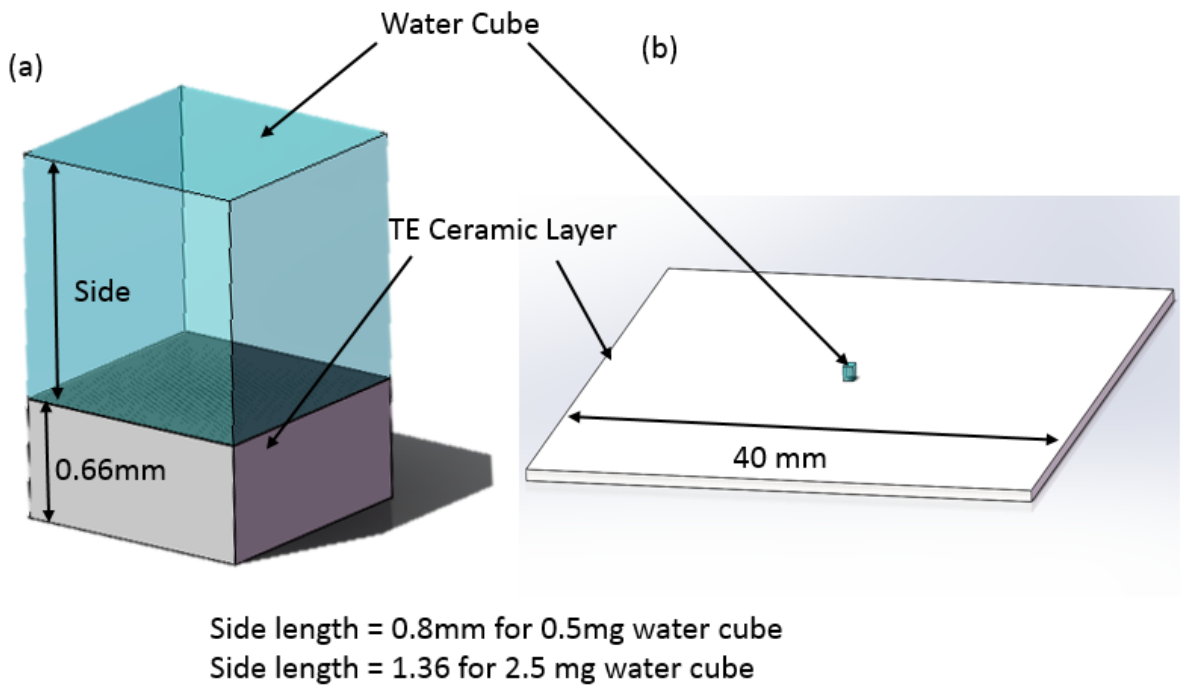


Figure 4.2 Three-dimensional geometry of the numeric model (a) water cube with a small TE ceramic layer (b) water cube with the real-scale ceramic layer

Christin Young [61] conducted some water droplet freezing experiments using 0.5mg and 2.5mg water droplets. He used a RTD that is located above the top of the TE ceramic layer and below the water droplet to take the temperature measurements. The sides of the water cubes were set to be 0.8mm and 1.36 mm in Star CCM+ to match the size of the water droplets used in the experiments. According to the most recent database of cell biology, the estimated size of human cells could be from 30 μm^3 to 4 $\times 10^6$ μm^3, as shown in Table 4.1 below. The equivalent edge length of the water droplet is from 3.11 μm to 158.74 μm. In addition to 0.8 mm and 1.36 mm, a cell with edge within the range from 3.11 μm to 158.74 μm was used to estimate the thermal behavior of human cells.

Table 4.1 Characteristic average volumes of human cells of different types[62]

cell type	average volume (μm^3)
sperm cell	30
red blood cell	100
lymphocyte	130
neutrophil	300
beta cell	1,000
enterocyte	1,400
fibroblast	2,000
HeLa, cervix	3,000
hair cell (ear)	4,000
osteoblast	4,000
alveolar macrophage	5,000
cardiomyocyte	15,000
megakaryocyte	30,000
fat cell	600,000
oocyte	4,000,000

Based on the previous mesh size study, the mesh element size for the 3D geometries was set to be 1×10^{-6}m. To provide more flexibility, the boundary growth rate was set to be slow and the relative value was set as a percentage of the base size. The advantage of allowing a relative size is that control volumes can be positioned in a way that reflects the interface between two regions. Then, in order to measure the temperature at different locations of the water cube, a temperature-detecting probe was inserted along each of the vertical axis of the geometry.

4.2 Initial Condition

Initially, the water cube region was set to be liquid and the ceramic layer of the TE module was set to be solid. From the experiments, water droplets were heated to room temperature and then cooled based on the current flow that went through the

thermoelectric module. In the numeric simulation, the initial temperature was set to be 293K.

4.3 Boundary Condition

4.3.1 Equivalent Heat Flux

The object of interest was the control volume at the center of the water droplet and the top layer of the thermoelectric module directly below the water droplet. Four side surfaces of the TE module were set to be adiabatic and all five exposed water cube surfaces were set to be natural convection with the heat transfer coefficient being $1\,W/m^2K$. The bottom surface of the water cube and the top surface of the TE module were set to be one overlapped interface. A heat sink was applied at the bottom of the structure to simulate the cooler in the experimental setup.

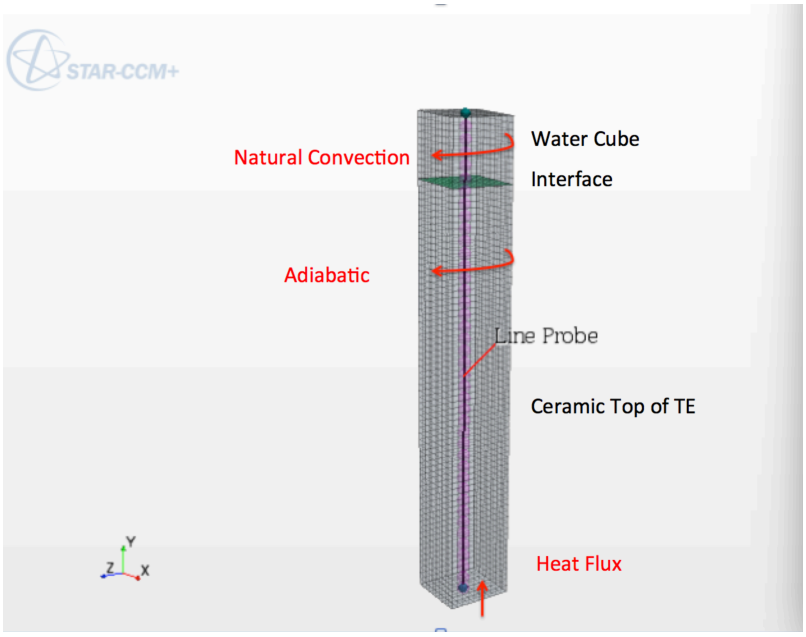


Figure 4.3 Boundary conditions schematic

The net heat loss that goes through the top of the ceramic layer is determined by the amount of heat that the TE module generates and the amount of heat loss due to natural convection as shown below.

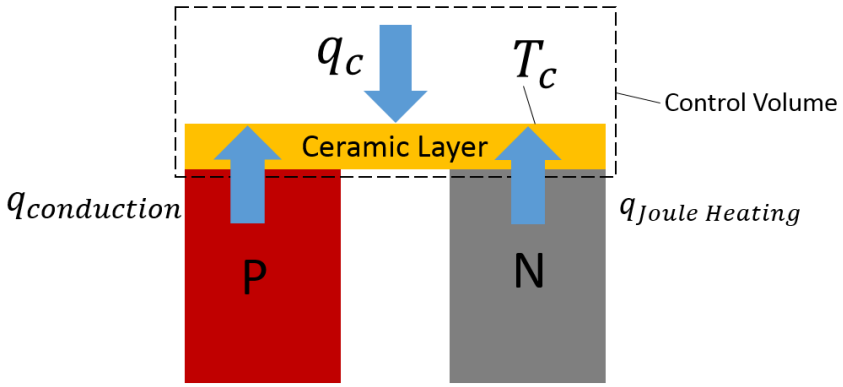


Figure 4.4 Energy balance diagram for the TE module

The total amount of heat going out of the control volume, q_{sink} , can be calculated by,

$$q_{sink} = q_c - q_{Joule\ Heating} - q_{conduction} \quad 4.1$$

Where, q_c is a function of T_c, the current that goes through the circuit 'I', number of junctions in the TE module 'N' and the Seabeck coefficient, α. Three heat terms can be calculated using the formulas below.

$$q_c = \alpha \cdot I(t) \cdot T_c \cdot N \quad 4.2$$

$$q_{Joule\ Heating} = \frac{1}{2} I(t)^2 R \quad 4.3$$

$$q_{conduction} = 2N \cdot \frac{T_h - T_c}{W} \cdot k\ A \quad 4.4$$

Knowing three terms in Eqn.4.1, the value of q_{sink} has been calculated and plotted in Fig.4.5.

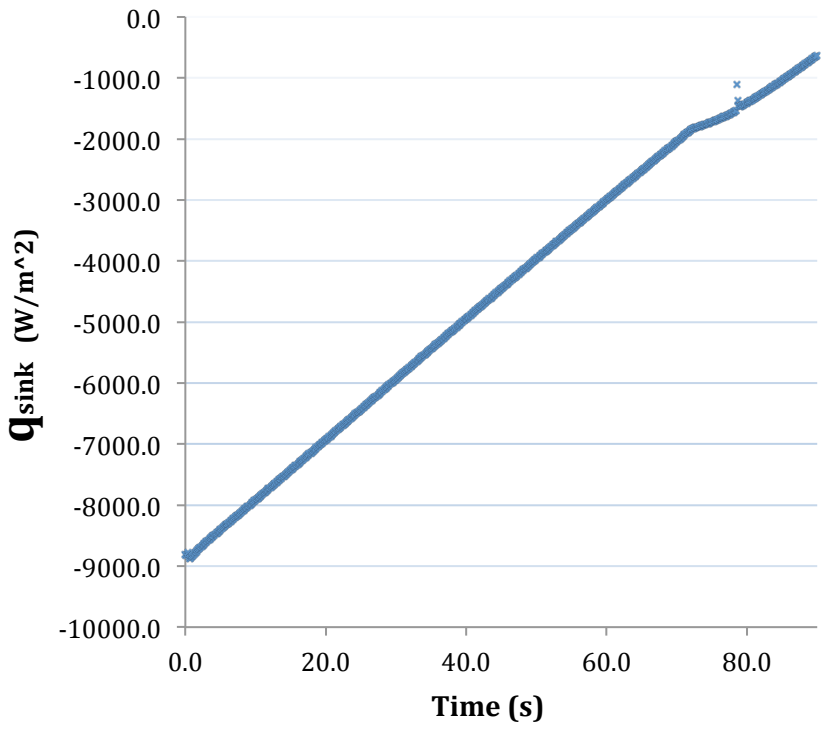


Figure 4.5 Instant amount of heat going out of the control volume at a certain time

This plotted $q_{sink}(t)$ profile shown in Fig 4.5 is applied at the bottom of the ceramic layer in the numeric models as a boundary condition.

4.3.2 Heat Spread Effect

The TE module below the water cube has a very complex geometry and a relatively large surface area compared with the water cube. In case the complex geometries of the TE module are not fully understood in the numeric model and shorten computing time, the heat spreading effect is used to simplify the geometry set up in the numeric model. Instead of using the exact geometry, the TE ceramic layer in the simulation was built much smaller, as shown in Fig.4.2. For geometries set up like Fig.4.2, the shape factor is 2 times the equivalent diameter of the TE ceramic layer area.

Table 4.2 shows all the dimension parameters and shape factors for each water cube size that were used in the simulation.

Table 4.2 Heat Spread Effect Parameters

Water Droplet Weight (mg)	0.003	0.5	1.0	2.0	2.5
Cube Side Length Size (m)	1.44E-04	7.94E-04	1.00E-03	1.26E-03	1.36E-03
Cube Surface Area (m^2)	2.08E-08	6.30E-07	1.00E-06	1.59E-06	1.84E-06
Effective Diameter of disk (m)	1.63E-04	8.96E-04	1.13E-03	1.42E-03	1.53E-03
Shape Factor	3.26E-04	1.79E-03	2.26E-03	2.84E-03	3.06E-03
Area of TE Ceramic Layer (m^2)	2.08E-08	6.30E-07	1.00E-06	1.59E-06	1.84E-06

Knowing the shape factor, the effective heat flux that was applied at the bottom TE ceramic layer can be computed with the equation,

$$q = Sk\Delta T \qquad 4.5$$

where S is the shape factor, k is the thermal conductivity of ceramic, and ΔT is the temperature difference between the top surface and the bottom surface of the TE ceramic layer.[63]

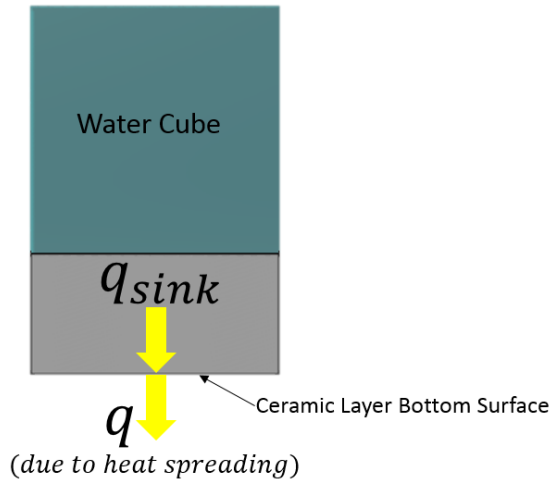


Figure 4.6 Boundary conditions applied at the bottom surface of the ceramic layer

Fig 4.6 shows the equivalent heat flux that should be applied to the bottom surface of the ceramic layer to accommodate for the heat spreading effect.

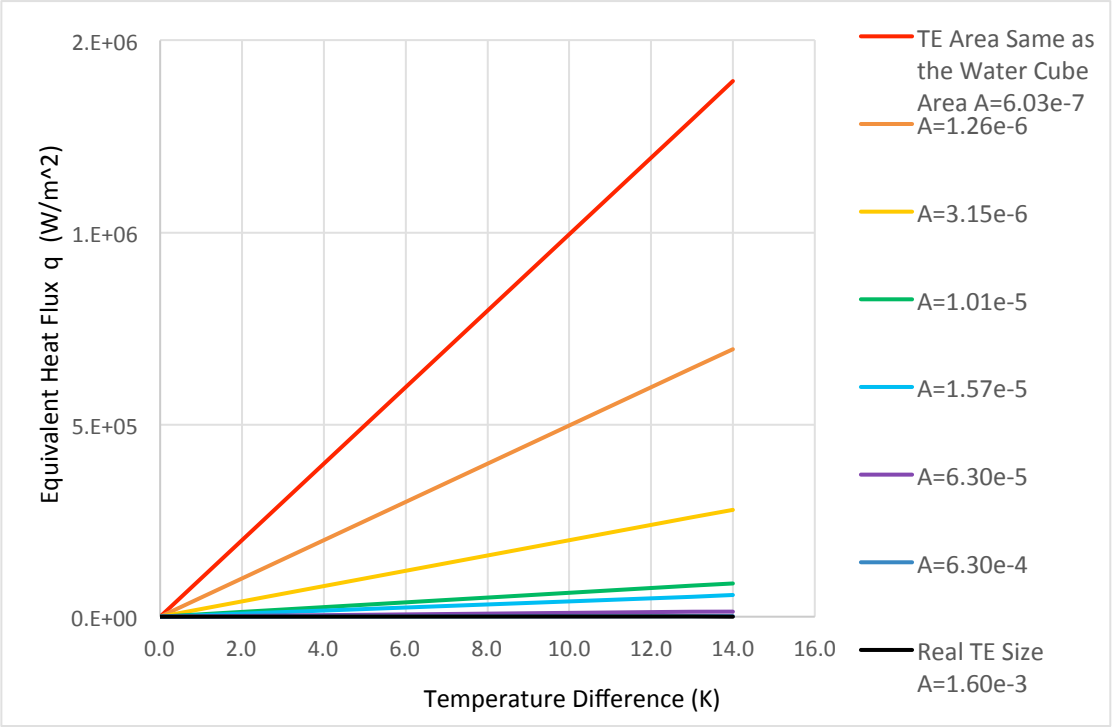


Figure 4.7 Equivalent heat fluxes that should be applied at the bottom of the ceramic layer to accommodate for heat spreading for a 0.5 mg water cube

Fig 4.7 shows the exact value of q that is needed for different temperature difference and different TE ceramic layer areas. The smaller the ceramic layer is, the bigger the heat flux value is. And as the temperature difference between the top surface and the bottom surface of the ceramic layer, the larger heat flux should be used. Just as shown in Fig. 4.7, when using the real size for the ceramic layer, the equivalent heat flux goes to zero.

All the TE layer sizes that were used in the numeric simulations were calculated. Fig 4.8 and Fig 4.9 show the heat fluxes that are required to apply at the bottom of the ceramic layer for different water cube sizes.

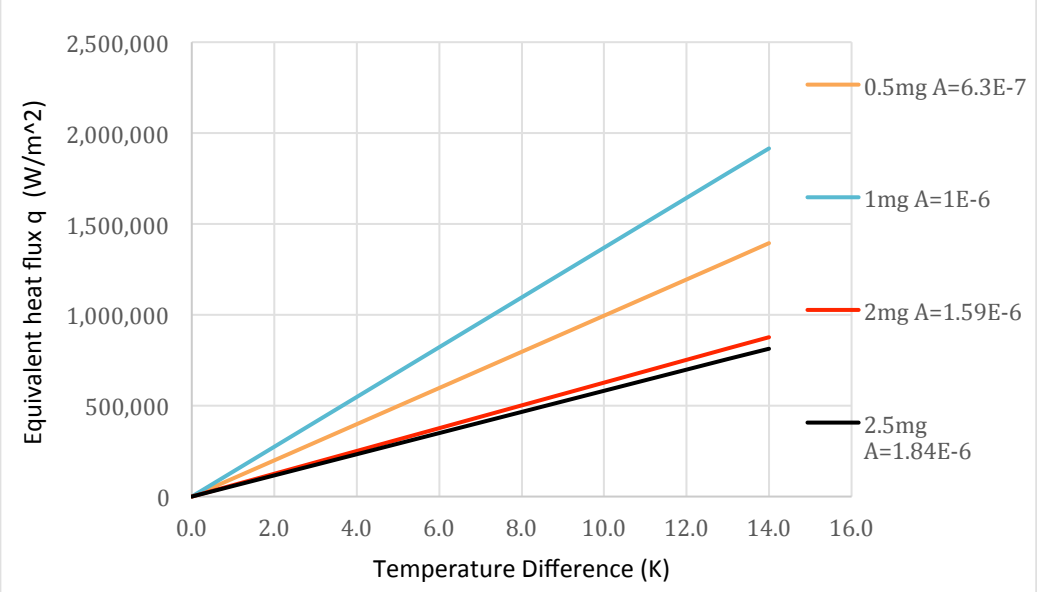


Figure 4.8 Equivalent heat fluxes that should be applied at the bottom of the ceramic layer to accommodate for heat spreading for a 0.5 mg, 1mg, 2mg or a 2.5 mg water cube

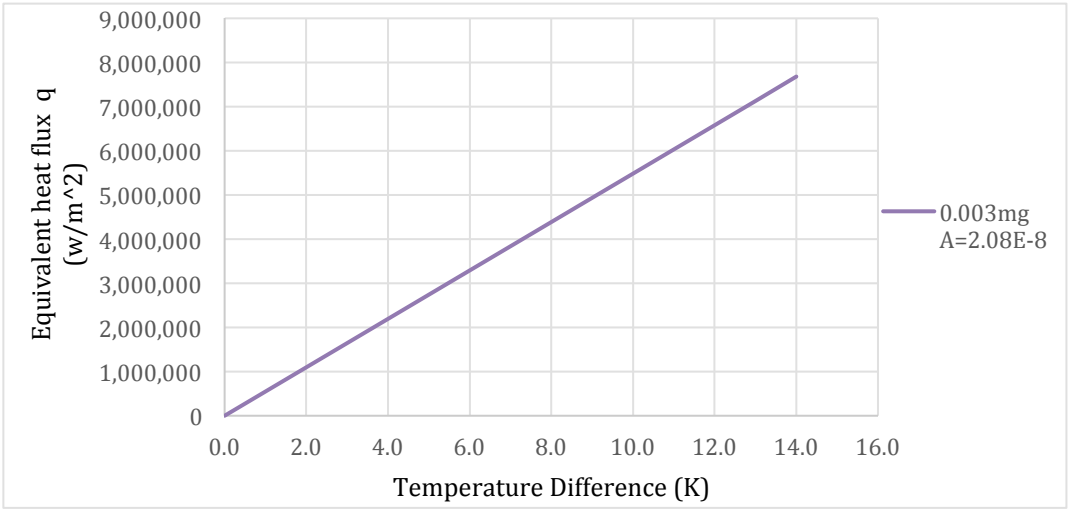


Figure 4.9 Equivalent heat fluxes that should be applied at the bottom of the ceramic layer to accommodate for heat spreading for a 0.003mg water cube

In order to accommodate the heat spreading effect in the simulation model, proper equivalent heat fluxes were applied at the bottom surface of the ceramic layer in additional to the $q_{sink}(t)$ profile mentioned in section 4.3.1.

4.4 Source Terms

4.4.1 Volume of Fluid

In order to accurately model the heat released due to phase change, the Volume of Fluid (VOF) model was utilized in the solver. The VOF formulation assumes that the liquid phase and solid phase are not interpenetrating. There are three conditions that could occur during the numeric simulation; pure liquid, pure solid, and two-phase mixture. To calculate the amount of latent heat released during the water droplet or cell sample solidification process for each cell, a variable called volume fraction has been introduced as mentioned in Chapter 3. In each cell, the sum of the volume fraction of the liquid and the volume fraction of the solid is equal to 1. The amount of latent heat can be calculated by the equation,

$$S = \rho L \, \Delta f_l \qquad 4.2$$

4.4.2 Crystallization Temperature

It has been found that water does not crystallize upon cooling at the liquid-solid equilibrium temperature but at a lower temperature.[64] The nucleation rate is the highest at the beginning of the freezing process. The difference between the liquid-solid equilibrium temperatures and the temperature at the highest nucleation rate increases as the water body size decreases. Based on some experimental data from different literatures, a plot was generated as shown below.

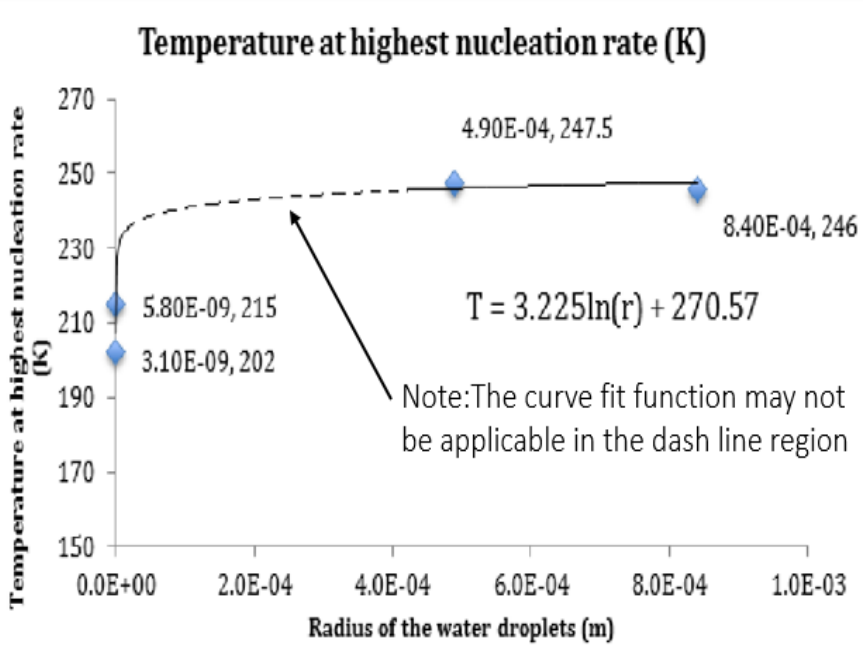


Fig.4.10 Size dependent temperature of water droplets at the highest nucleation rate[64, 65]

Based on the curve fit function, the freezing point of different size water droplets were calculated and listed in the table below.

Table 4.3 Calculated freezing temperature for water droplets with different size

Weight (mg)	Radius (mm)	Temperature Prediction (K)
0.003	9.00E-02	240.5
0.01	1.30E-01	241.7
1	6.20E-01	246.8
2	7.80E-01	247.5

4.4.3 Thermal Properties

The thermal properties such as thermal conductivity and specific heat are all functions of temperature. In Star CCM+, the freezing temperature was set to be a very small temperature range around the value predicted in Table 4.2. Within this range of temperature, thermal properties were interpolated with a linear relation and the value of

the thermal properties changed very rapidly from solidus value to liquidus value. Thus, the property for melting and solidification was implemented like a step function.

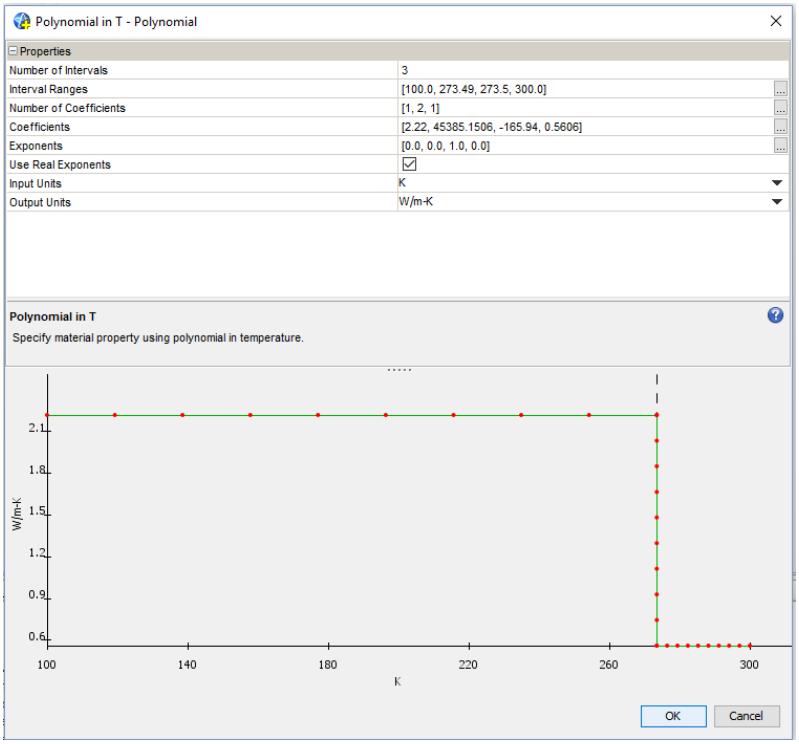


Figure 4.11 Thermal conductivity polynomial function in term of temperature

4.5 Solver

Fig.4.12 shows the solvers that have been implemented in the Star CCM+ model. Based on the previous time independence study, the time-step was set to be 1×10^{-5}s.

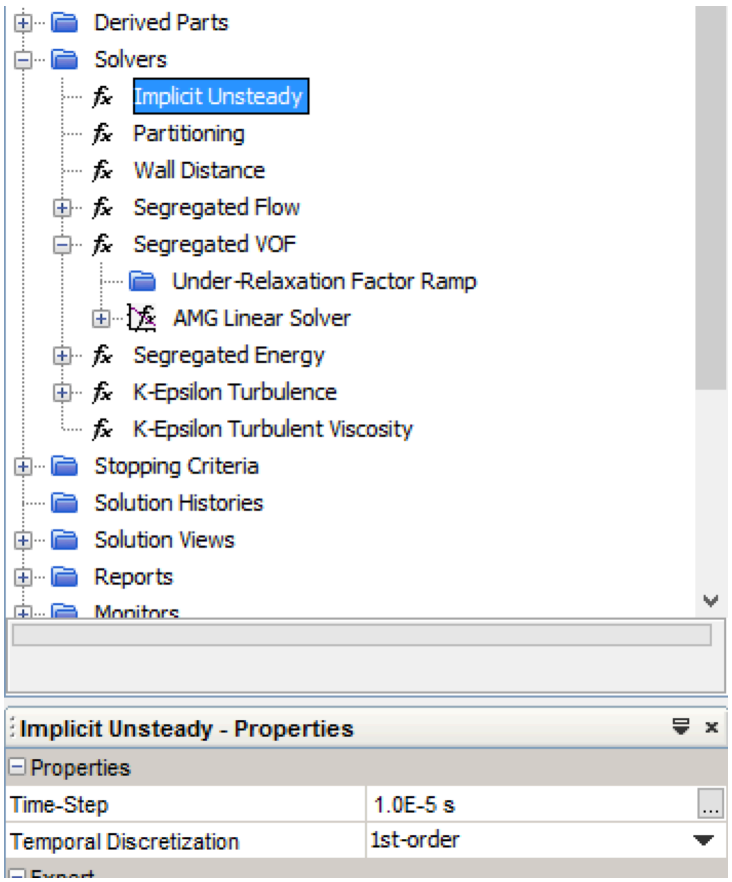


Figure 4.12 Solvers implemented in STAR CCM+

The solidification process is dynamic and many cells can release latent heat at subsequent iterations during one time-step. An accurate model requires a big number of iterations to ensure energy is balanced through each time-step. Usually, 200 to 1000 numbers of iterations is sufficient for a very small energy residual value.

4.6 Partially frozen water cube model

A simulation solely based on a frozen water cube model can only give some trends in the temperature profile. Since a cell contains about 70% water in mass, the rest 30% mass may have a different freezing point than water. It is then valuable to study the interaction between water and the rest of the cell mass and look at the temperature profile when a water cube in the numeric model is partially frozen. The partially frozen water

cube model can also help guide the design of the DTA to detect if a fraction of a cell freezes or not. The frozen percentage of the water cube was set to be 100% in the previous model. In this section, some minor modifications of the physical setup of the numeric models allows for a 50%, 25% or 0% frozen water in mass.

4.6.1 Half-frozen water cube

Half of the water cube geometry is set to be liquid phase in Star CCM+ regardless of the temperature change. The rest of the simulation model contains liquid that can change phase. Chapter 4.1 to 4.5. In order to better understand the correlation between the fraction of frozen water that is frozen and the change of temperature profile, three different locations of the half amount of water cube was set to be liquid only as shown in the picture below.

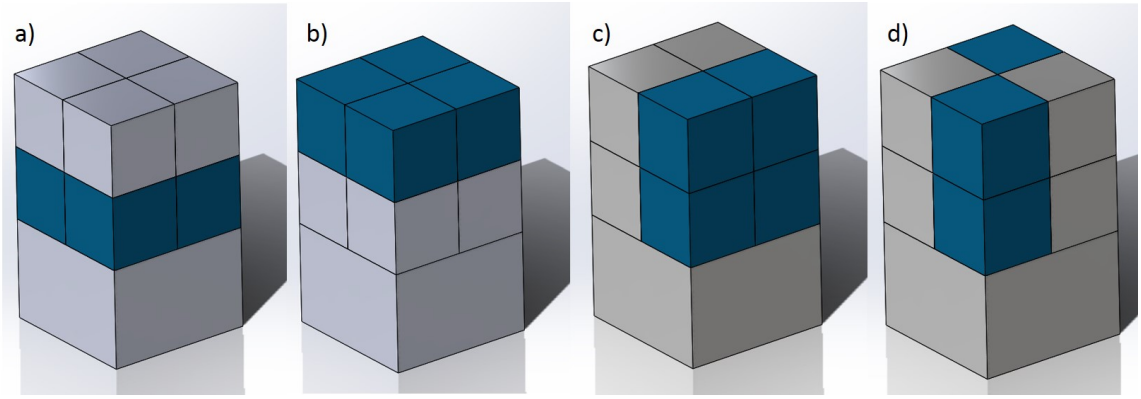


Figure 4.13 Locations of the frozen half of the water cubes

The solid halves of the water cubes in Fig. 4.13 are highlighted in blue. The frozen half volume in different simulations locates at the bottom, the top, one side and along one diagonal of the water cube geometry respectively. The initial condition and the boundary condition of the simulations remain the same. However, since only one half amount of

the water in each of the numeric model can become frozen, the total amount of latent heat released during the freezing process in each simulation was cut into half.

4.6.2 A-quarter-frozen water cube

Similar to the numeric models in Section 4.6.1, three quarters of the water cube geometry is set to be liquid phase in Star CCM+, which means that only one quarter of the water cube will be frozen at most in the model. Four different locations of the frozen quarter of the water cube were chosen as shown in the picture below.

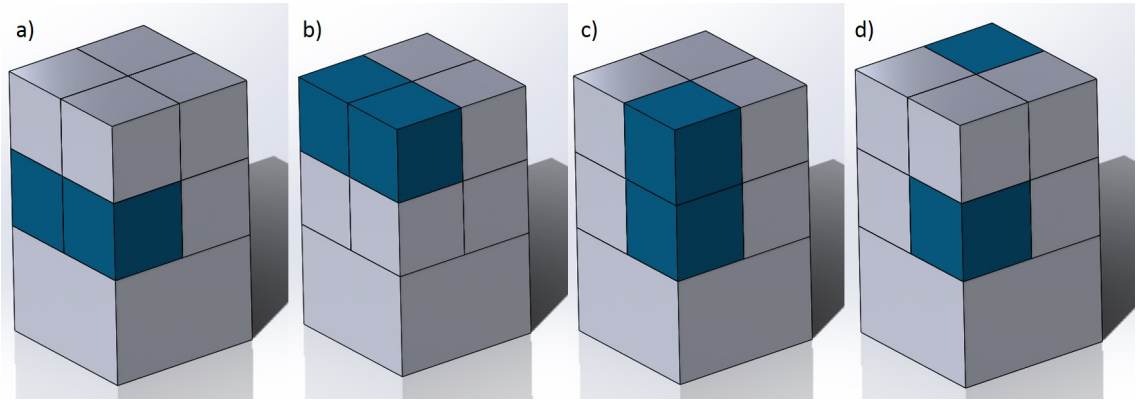


Figure 4.14 Locations of the frozen half of the water cubes

4.6.3 Liquid only water cube

The entire water cube in the simulation model was set to be liquid-only. The geometry, boundary conditions, initial condition and the solver remained the same like what were in the other partially frozen models. The modifications in the liquid phase only water cube model will allow us to see the strong influence of the heat flux boundary condition calculated based off the heat spread effect on the average volumetric temperature profile.

5. Results

5.1 Small TE Layer Module

The Star CCM+ results of the 0.5mg water cube model described in Fig. 4.2 (a) are shown in Fig.5.1 and Fig 5.2. The side length of the water cube is 0.8mm. The TE ceramic layer is set to have the same top and bottom surface area compare with the water cube and the layer thickness is at 0.66mm. In order to validate the results, several trials were run with different cell numbers and time steps before getting the results shown in Fig.5.1. Star CCM+ output energy residuals with an order of magnitude being 10E-7. Fig 5.1 and Fig 5.2 shows the volumetric average temperature profile of the 0.5mg water cube in Star CCM+ and Christin Young's RTD temperature results.

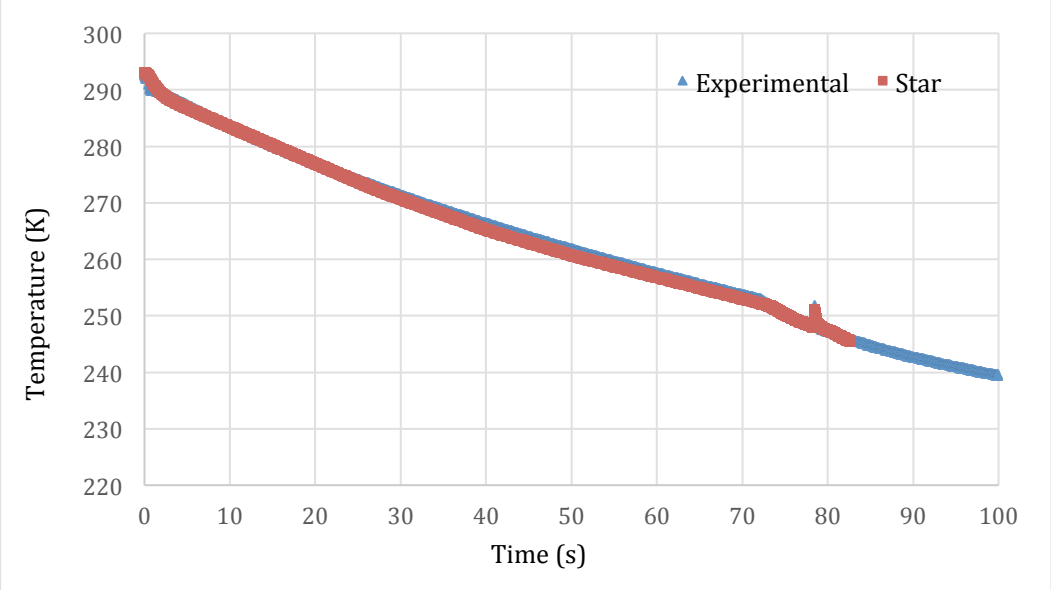


Figure 5.1 Temperature profile of 0.5mg water droplet sample with the TE layer top surface as big as the bottom of the water cube surface area

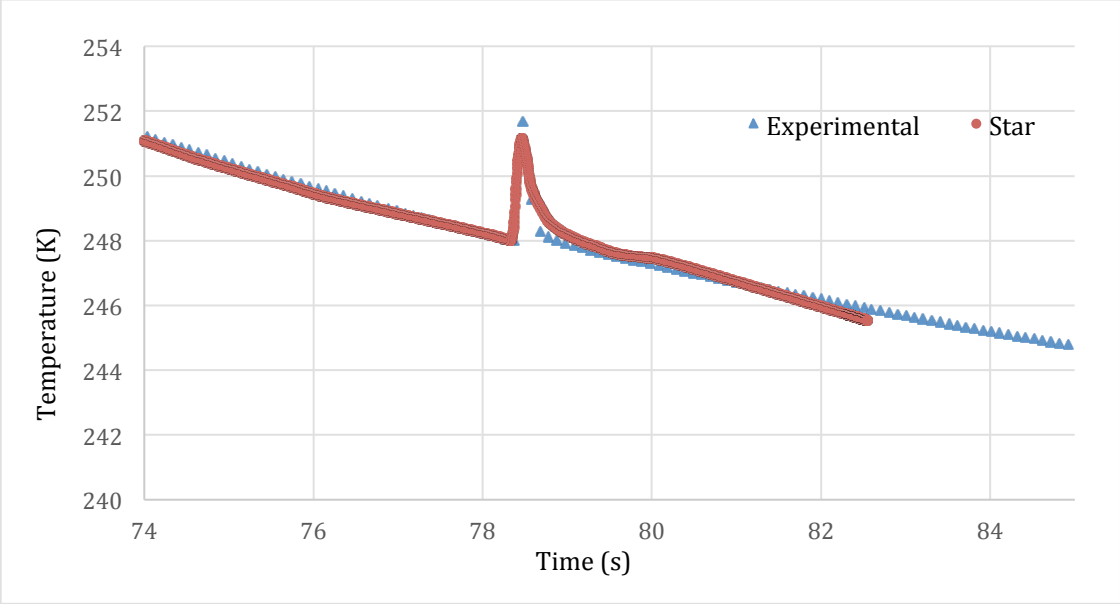


Figure 5.2 Focused view of the temperature profile of 0.5mg water droplet sample with the TE layer top surface as big as the bottom of the water cube surface area

The biggest difference between the numeric and the experimental data at a certain time was about 0.2 K. The numeric solution matches the temperature distribution trend from the experimental data. The moment when the rate of crystallization reached its highest value, the temperature of the water droplet increased rapidly, which is what happened in the experiments. After the temperature increased, the temperature fell back with a relatively high cooling rate. The high cooling rate from Fig.5.2 is because the temperature difference between the top surface and the bottom surface of the TE ceramic layer was increased. Therefore, according to Fig. 4.9, the additional heat flux applied at the bottom surface of the TE layer was increased. Eventually, the cooling rate went back to the value before the latent heat was released in the water droplet.

The numerically calculated temperature distribution of the 2.5mg water droplet is plotted in Fig.5.3 and Fig.5.4.

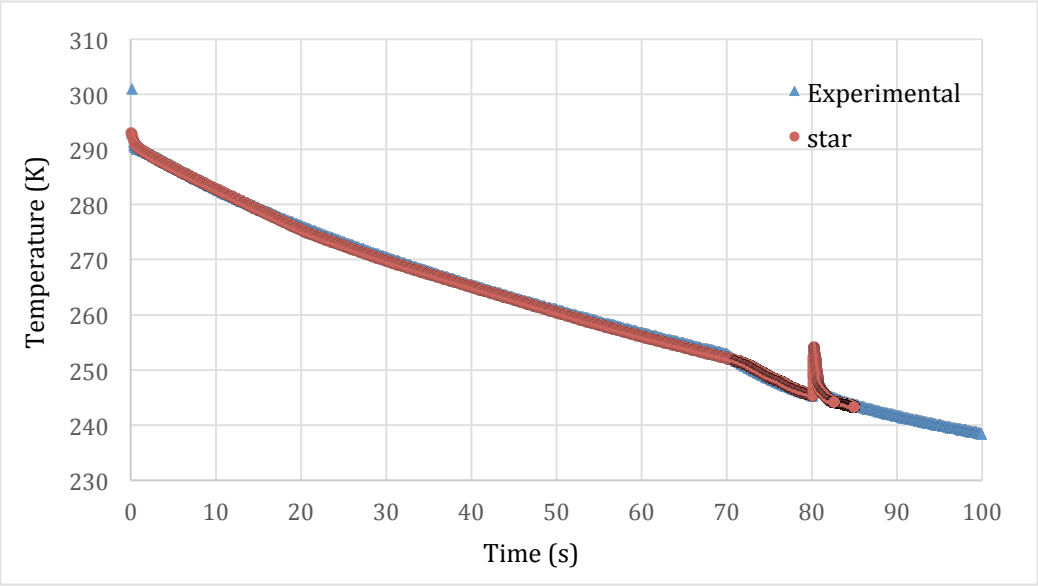


Figure 5.3 Temperature profile of 2.5mg water droplet sample with the TE layer top surface as big as the bottom of the water cube surface area

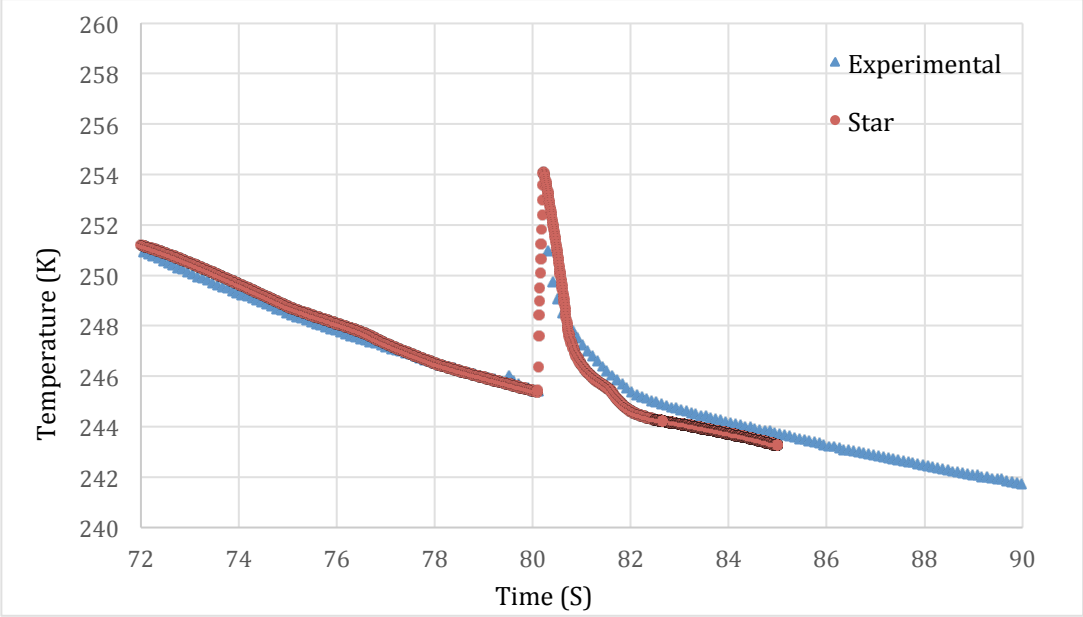


Figure 5.4 Focused view of the temperature profile of 2.5mg water droplet sample with the TE layer top surface as big as the bottom of the water cube surface area

The biggest temperature difference between the numeric and experimental data was about 0.82K. The 2.5mg simulation model was validated similarly to the 0.5mg water droplet model.

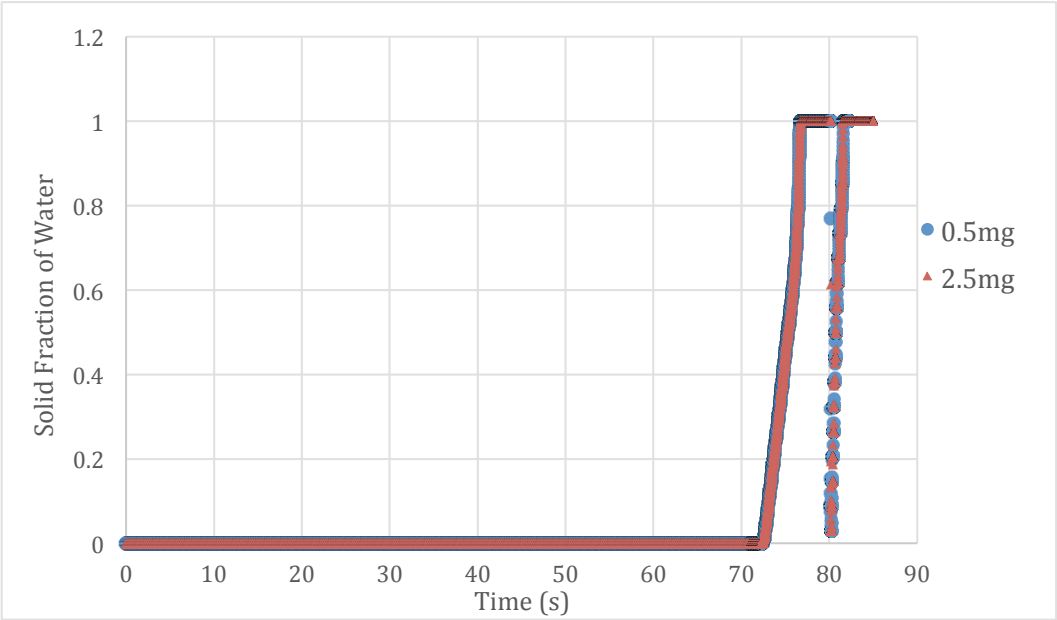


Figure 5.5 Solid fraction of water of the 0.5mg and the 2.5mg water droplet sample

As shown in the figures, the temperature difference between the experimental data and the numeric solution is very small. With the sudden latent heat being released after the water droplets reached the freezing temperature, the temperature profile formed a peak in the plot that is very similar to the experimental data. Additional heat flux was calculated based off of Eqn.4.5 and then applied at the bottom surface of the ceramic layer. The temperature in the numeric model was able to drop quickly like what was shown in the experimental data because of the additional heat flux.

The solid fraction of water remained to be zero until the phase change process started after 70 seconds. The fraction of ice went up to 100% when the temperature fell into the tight freezing temperature range set in the numeric model. When the temperature

rose due to the latent heat release, the solid fraction of water dropped around 80 seconds. Due to the steep cooling rate, the temperature quickly dropped below the freezing point of the water. And the solid fraction of water went back to 100% and remained unchanged.

5.2 Temperature Distribution Prediction

After the determination that the solution obtained from the Star CCM+ model is feasible, simulations were run to predict the temperature distribution of water droplets weighing 1mg, 2mg and 0.003mg. Fig.5.6 shows the focused view of the temperature profile of 1.0 mg, 2.0 mg, and 0.003mg water droplets when the samples were freezing.

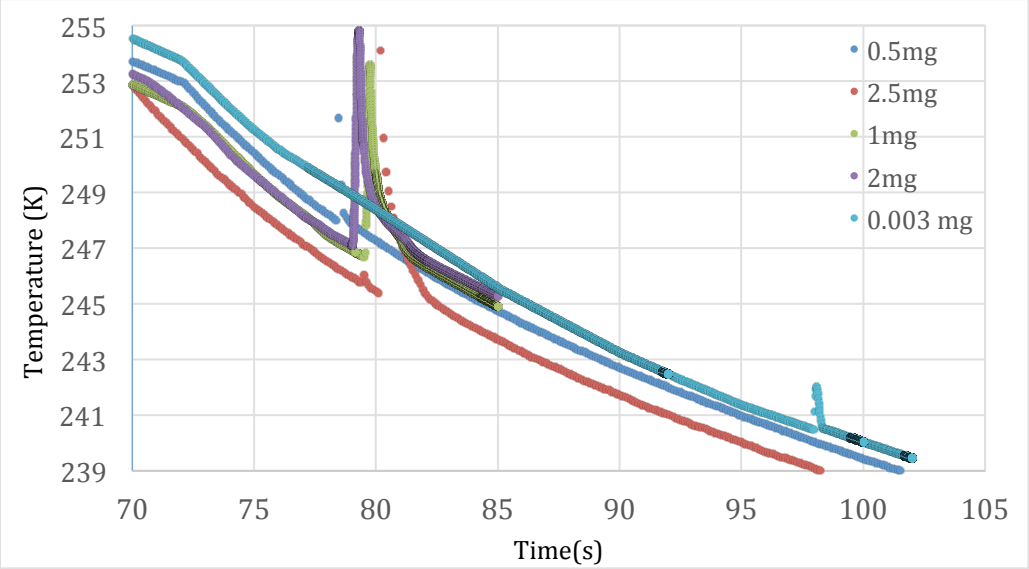


Figure 5.6 Focused view of the temperature profile of different size water cube

In Fig 5.6, water cubes with varies weight experience phase change processes at different temperature. The heavier the water cube is, the higher the temperature is when phase change processes begin. Since a heavier water cube releases more latent heat

during the entire phase change process, the temperature change is expected to be bigger for a heavier water cube after the phase change begins. In Fig 5.6, the temperature change is about 7 K for the 2.5 mg and the temperature change is about 2 K for the 0.003 mg curve. Other water cube curves with the cube weights between 0.003mg and 2.5mg suggest the same trend.

5.3 Partially frozen water cubes

5.3.1 Half-frozen water cube

Fig 5.12 was plotted after the physical time reached 100 second. This means that the phase change processes have already happened in the model. Fig 5.12 displays the solid volume fraction of water in the water cube geometry. Figure a, b and c within Fig 5.12 shows that the frozen part of the water cube matches with the physical set up in the model mentioned in section 4.6. The two one-quarter solid region set up as shown in Fig 4.13 part d is not included in the figure below because the two solid regions are set to be two different physical continuums. Star CCM+ is incapable of displaying discontinuous solid regions in the solid volume fraction plot for the water cube.

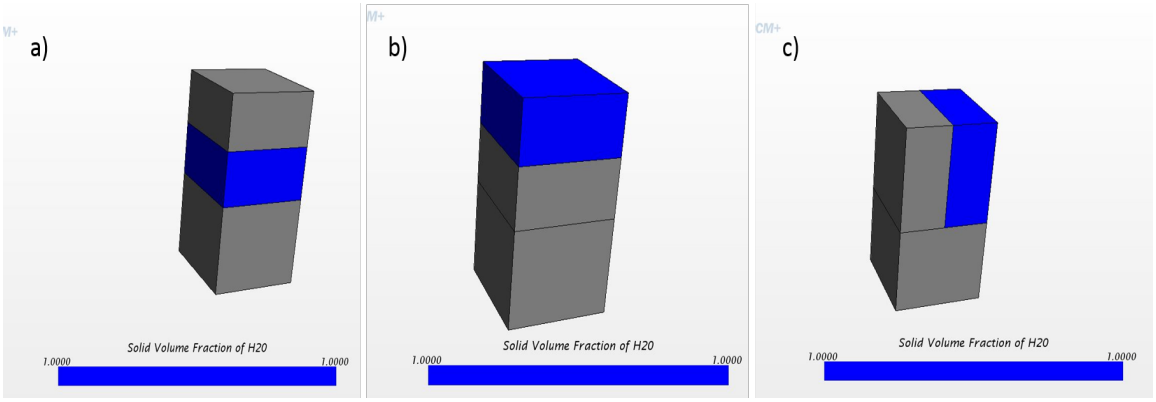


Figure 5.7 Solid volume fraction of water results for one-half frozen water droplets

Fig 5.7 does not have the solid volume fraction of water result for the set up in Fig 4.13 part d. Fig 5.7 only shows where the geometry was solid after cooling, however, the phase of the water at different locations of the water cube geometry is not the primary factor that influence the temperature profile. The amount of latent heat released during the phase change process is the primary factor.

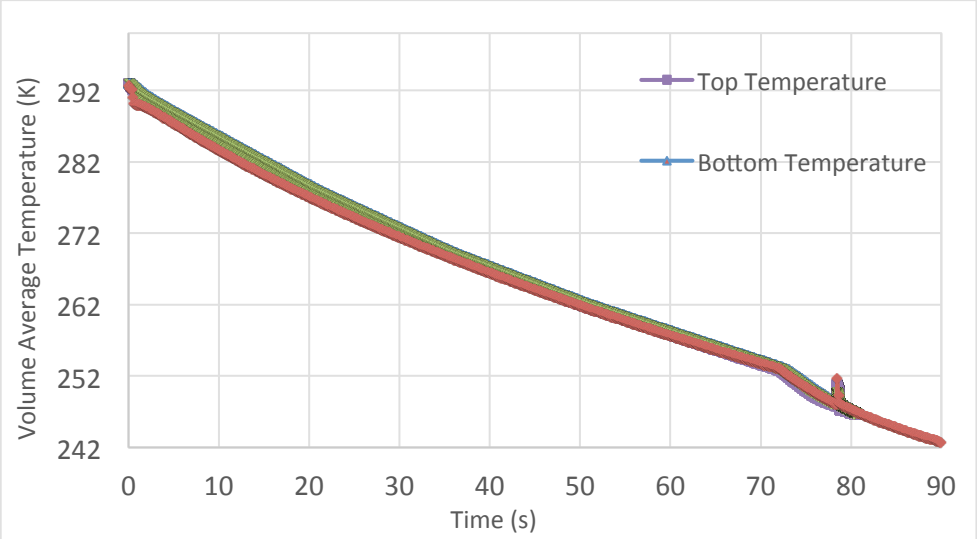


Figure 5.8 Temperature profile when only the bottom half of the water cube is frozen

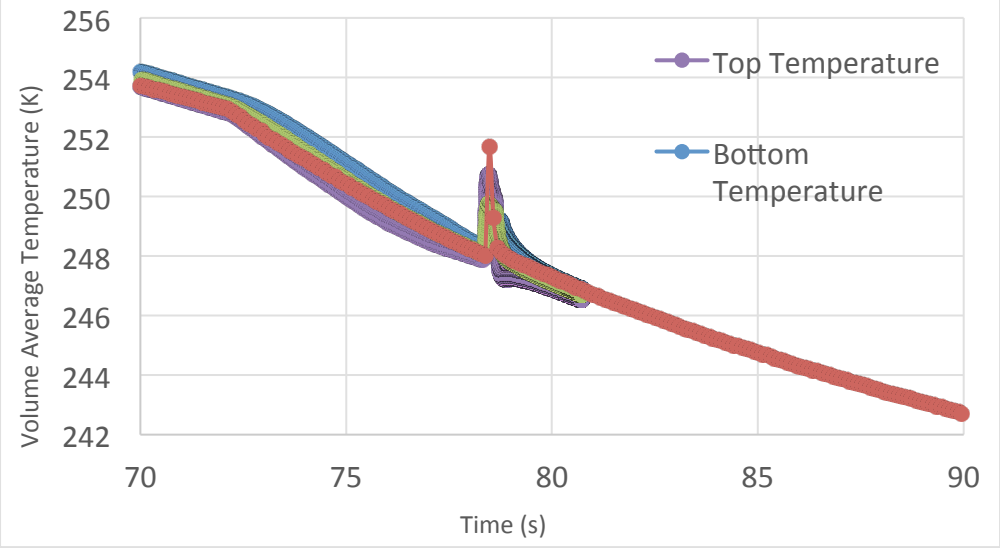


Figure 5.9 Detailed temperature profile when only the bottom half of the water cube is frozen

The numeric model temperature profile for the overall volume agrees with the experimental data before the water cube starts to freeze. In Fig 5.9, the height of the peak for the overall volume temperature profile is approximately half of the height of the peak formed in the experimental data. In order to better compare the temperature profiles in Fig 5.9, area under the peak is estimated treating the area as a triangle. The area under the T (t) curve is related to the amount of energy being released during phase change. Knowing the three vertices of the triangle, A (A_x, A_y), B(B_x, B_y) , C(C_x, C_y), the area of the triangle can be calculated by,

$$area = \left| \frac{A_x(B_y - C_y) + B_x(C_y - A_y) + C_x(A_y - B_y)}{2} \right| \qquad 5.1$$

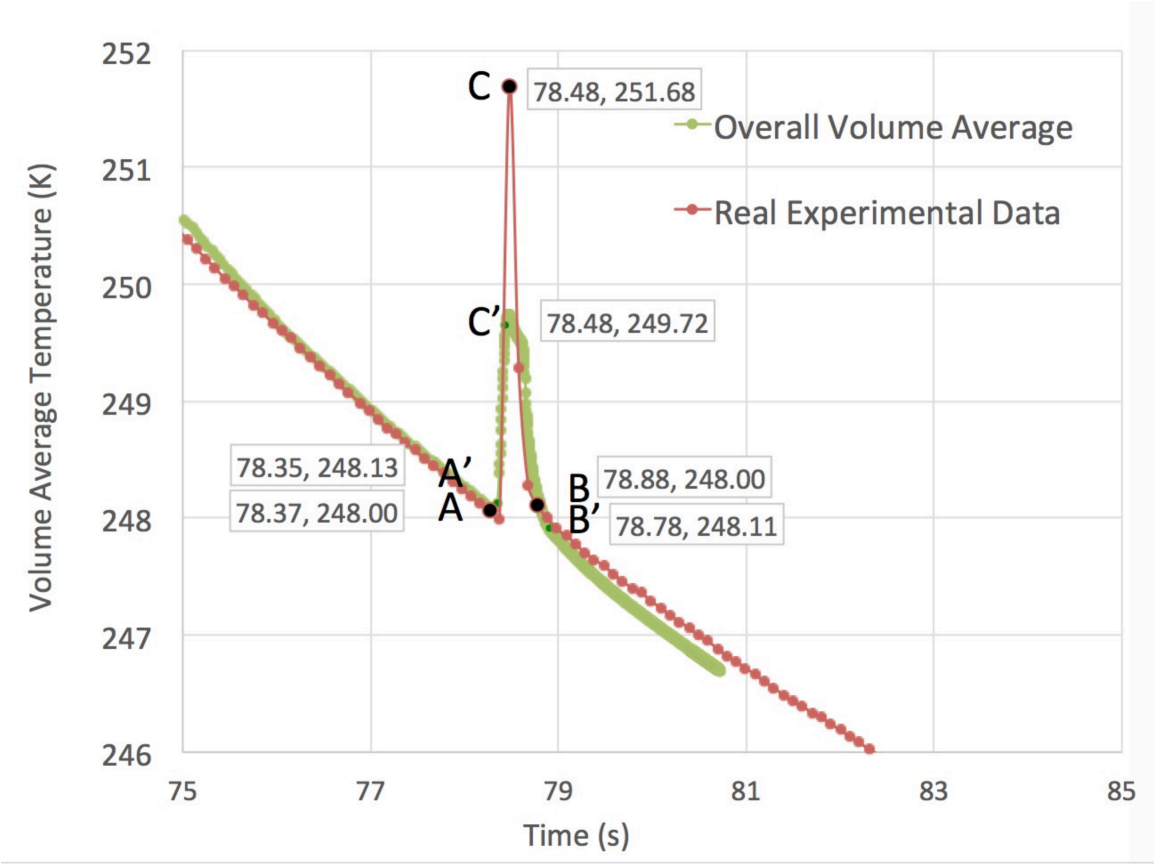


Figure 5.10 Temperature profile with some data point callouts

Fig 5.10 shows the temperature profile with six called out data points that are used to calculate the area under the peak. The estimated area under the peak for the experimental data is 0.84 and the estimated area under the peak for the overall volume average temperature is 0.44. The ratio of the area from the simulation result over the experimental result is 0.52. This is very close to 0.5, which means that half amount of latent heat was released during the phase change process.

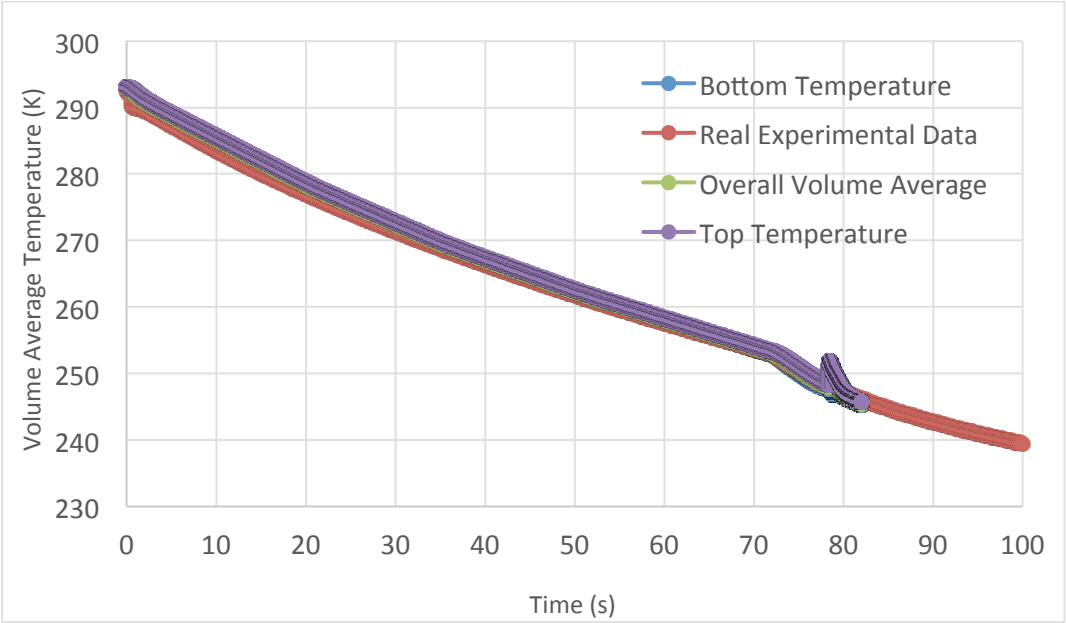


Figure 5.11 Temperature profile when only the top half of the water cube is frozen

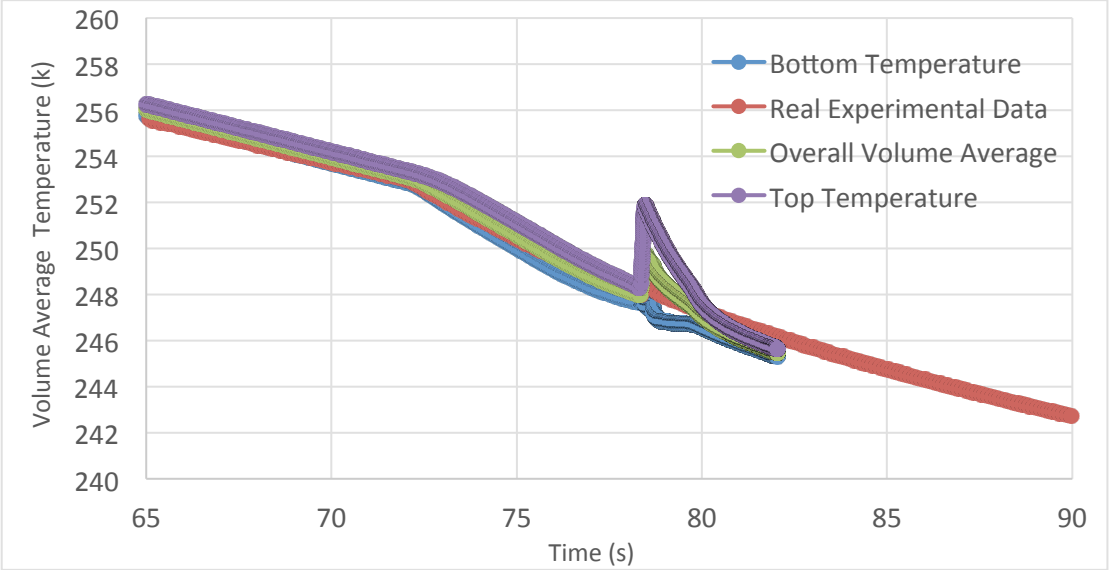


Figure 5.12 Detailed temperature profile when only the top half of the water cube is frozen

Similar to what Fig 5.8 and Fig 5.9 showed, the numeric model temperature profile for the overall volume in Fig 5.11 agrees with the experimental data before the water cube starts to freeze. In Fig 5.12, the height of the peak for the overall volume temperature profile is approximately half of the height of the peak formed in the

experimental data. However, since the frozen half of the water cube is at the top portion, the slope of the temperature drop due to the heat spreading heat flux applied at the bottom of the TE model is shallower compare with the results in Fig 5.9.

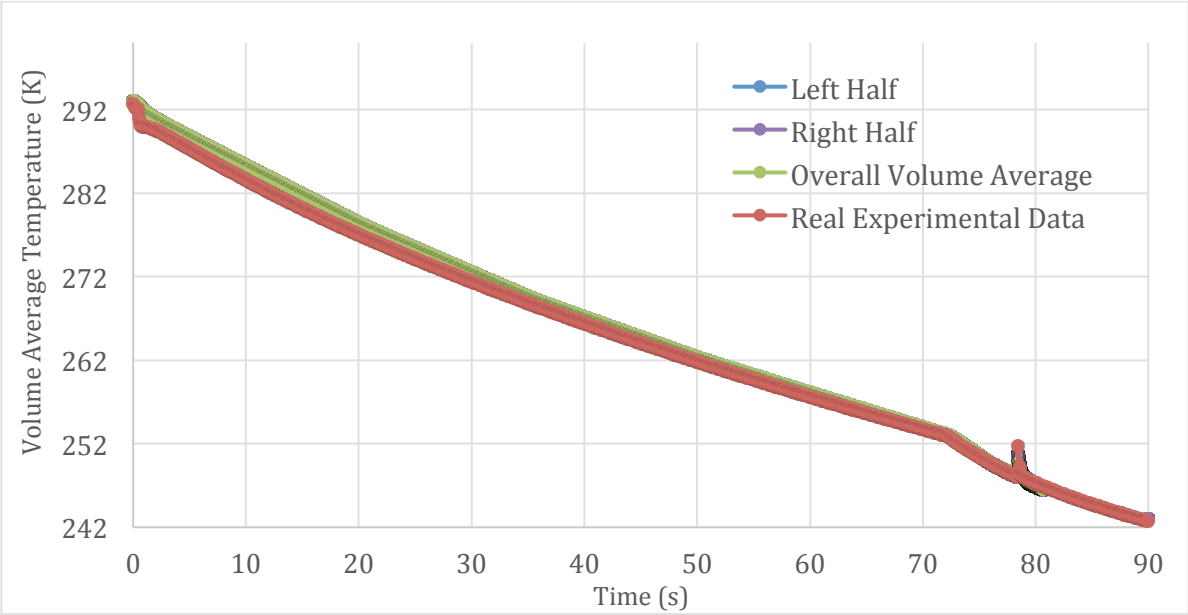


Figure 5.13 Temperature profile when only one half side of the water cube is frozen

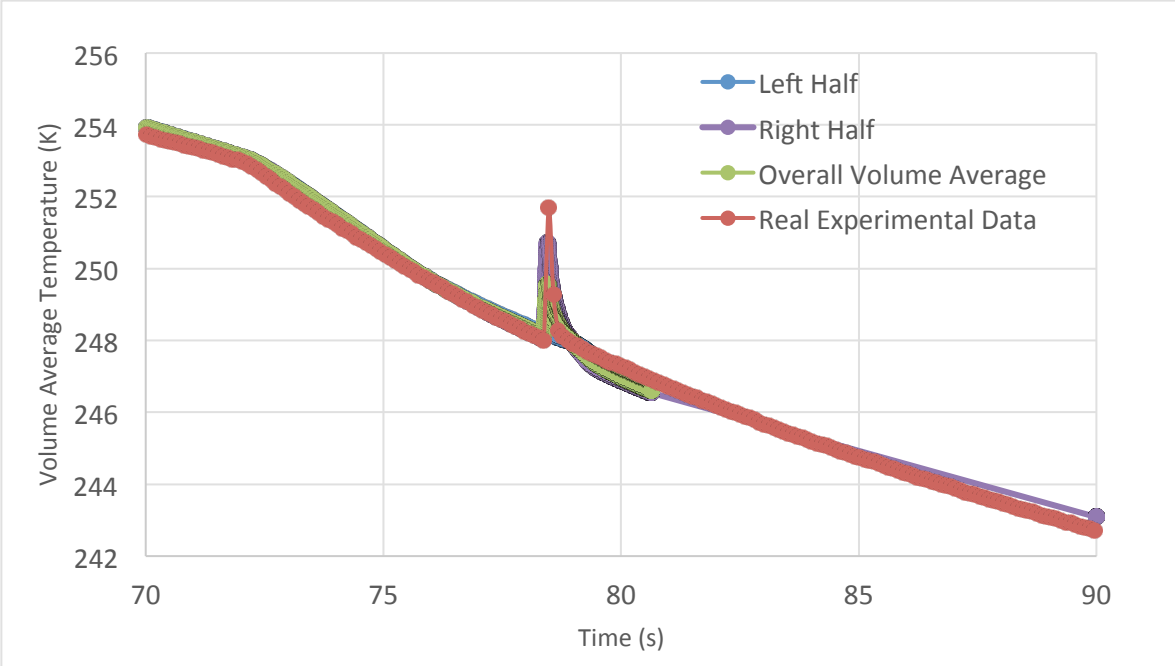


Figure 5.14 Detailed temperature profile when only one half side of the water cube is frozen

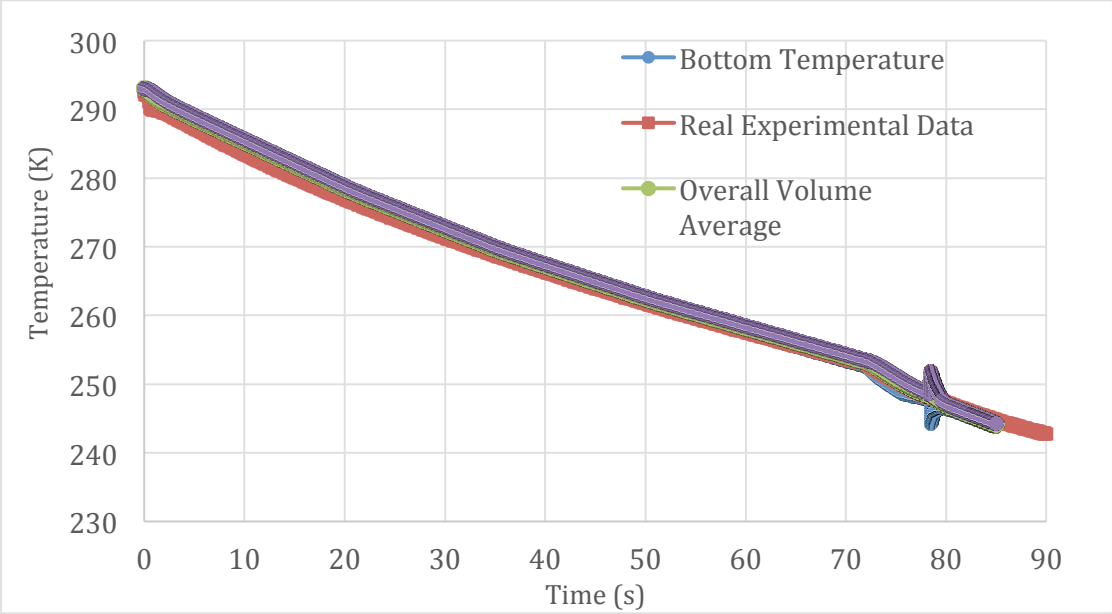


Figure 5.15 Temperature profile when two discontinuous quarters of the water cube is frozen

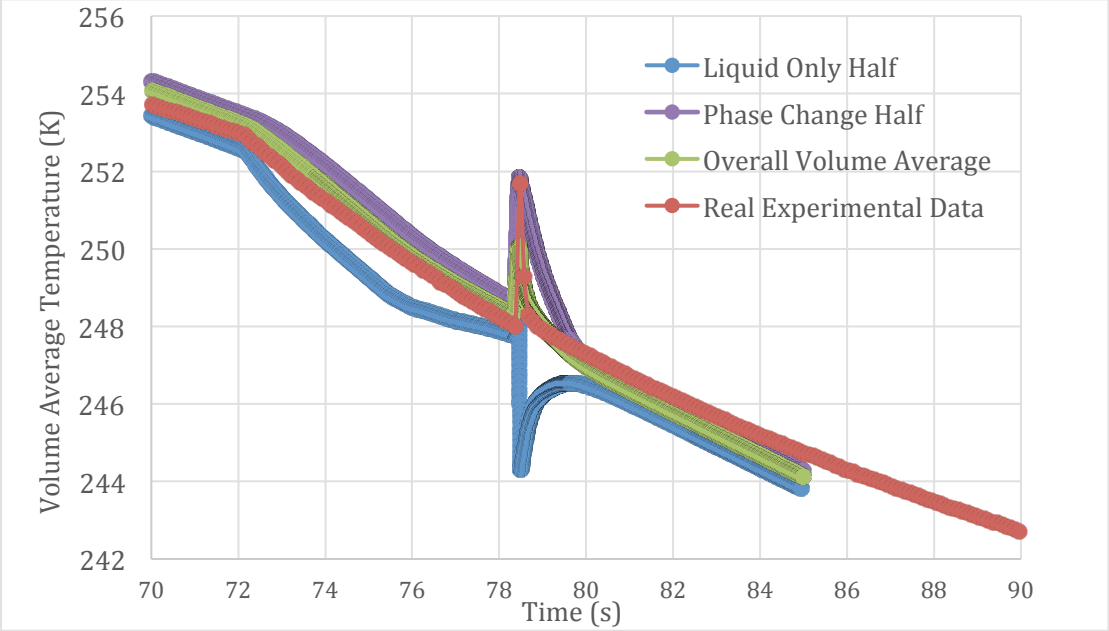


Figure 5.16 Detailed view of the temperature profile when two discontinuous quarters of the water cube is frozen

Table 5.1 Estimated areas under the peak for half frozen water cube models

Locations	Bottom	Top	Side	Two Quarters
Estimated Area Under the Peak	0.44	0.58	0.43	0.49
Simulation/Experiment Area ratio	0.52	0.69	0.51	0.59

The under the peak area ratio is a rough estimation of the latent heat energy ratio. Based on the ratio value in Table 5.1, the amount of heat released during the phase change process is about one half of the value in the experiment. The results in Fig 5.9, Fig 5.12, Fig 5.14 and Fig 5.16 shows that the locations of the frozen part of the water cubes have very little influence of the highest value on the volume average temperature curves. This can be explained by the same amount of latent heat released during the freezing process among all three half-frozen models.

However, the locations of the frozen part of the water cubes do influence the cooling rate right after the temperature profiles reach their highest peak value.

5.3.2 A-quarter-frozen water cube

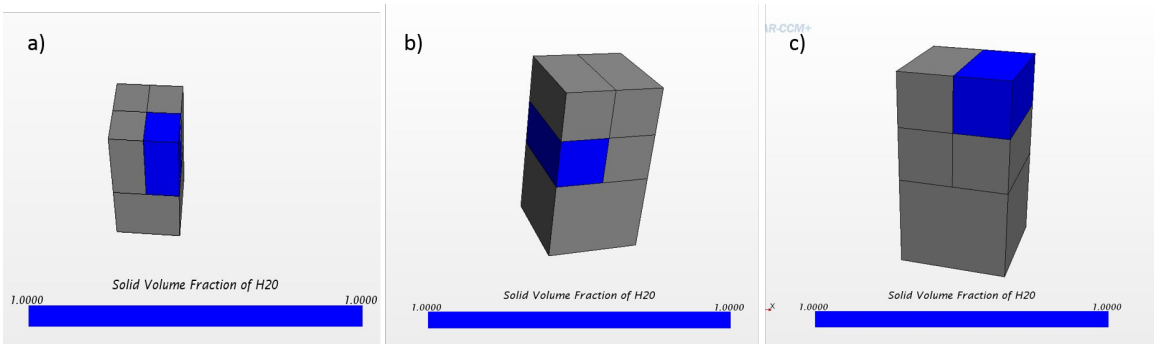


Figure 5.17 Solid volume fraction of water results for one-quarter frozen water droplets

The solid part of the water cubes shown Fig 5.17 matches with the physical model setup mentioned in the Section 4.6.2. Fig. 5.18 to Fig. 5.25 below shows the volume average temperature profile when a quarter of the water cube was frozen.

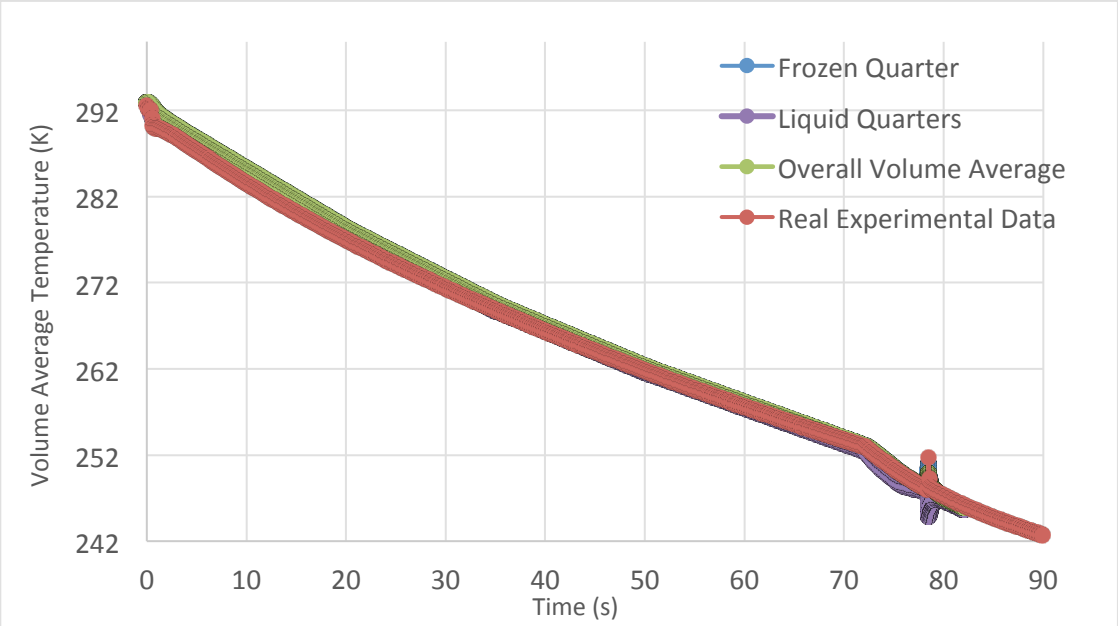


Figure 5.18 Temperature profile when only one vertical quarter of the water cube is frozen

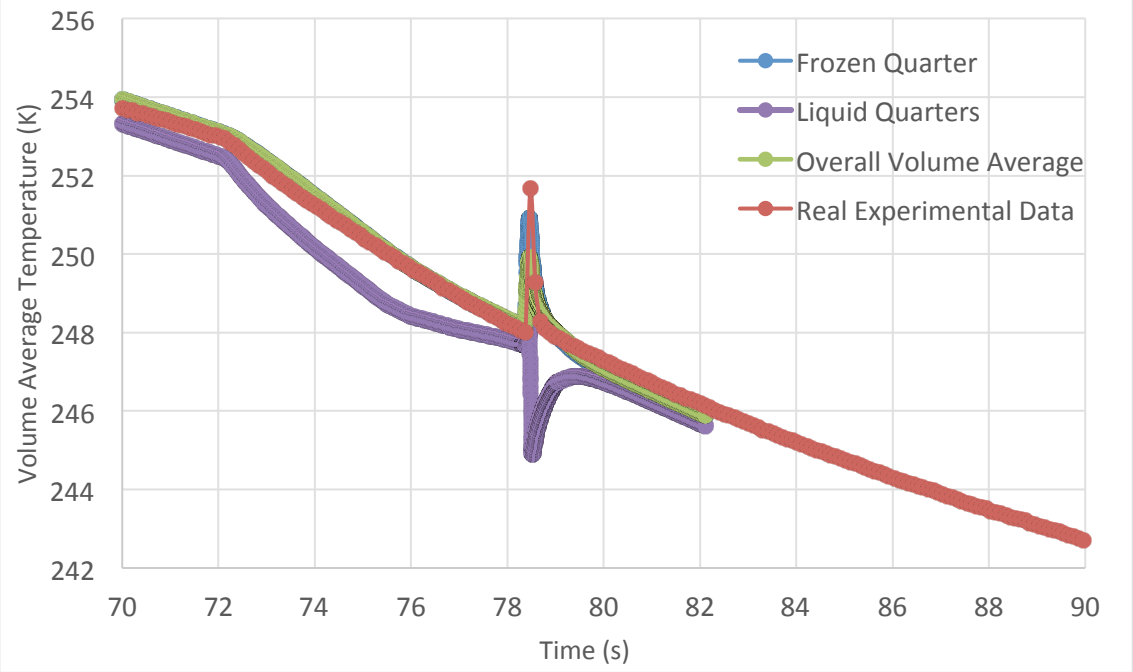


Figure 5.19 Detailed temperature profile when only one vertical quarter of the water cube is frozen

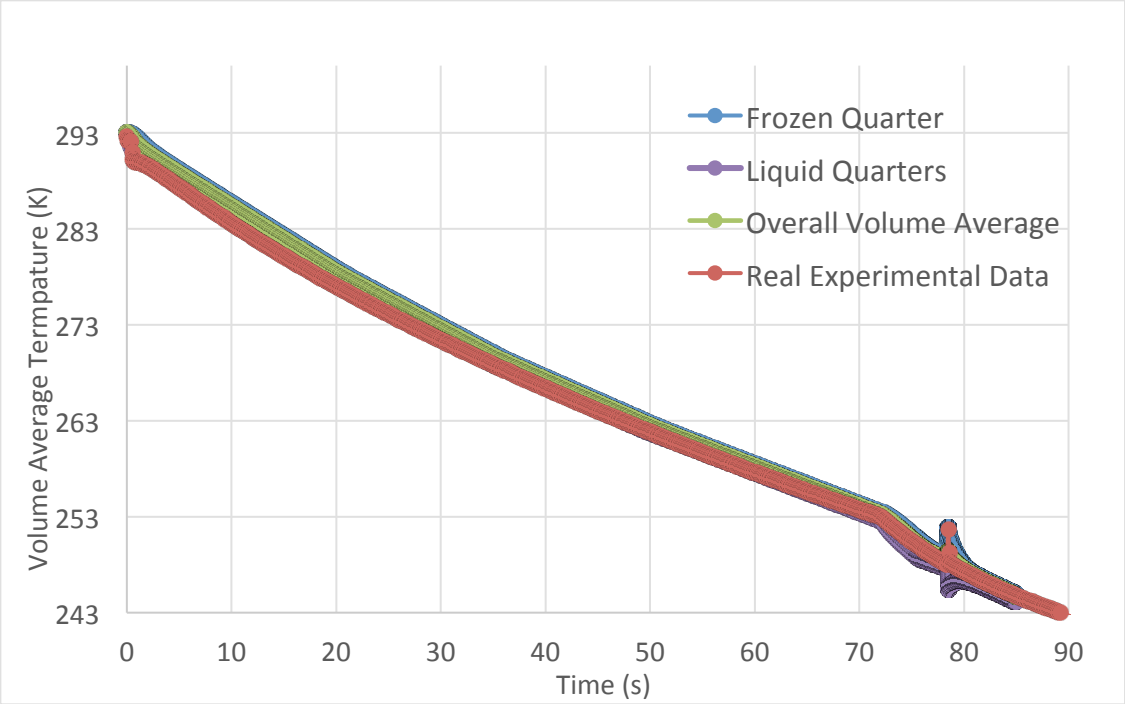


Figure 5.20 Temperature profile when only one horizontal quarter of the water cube is frozen

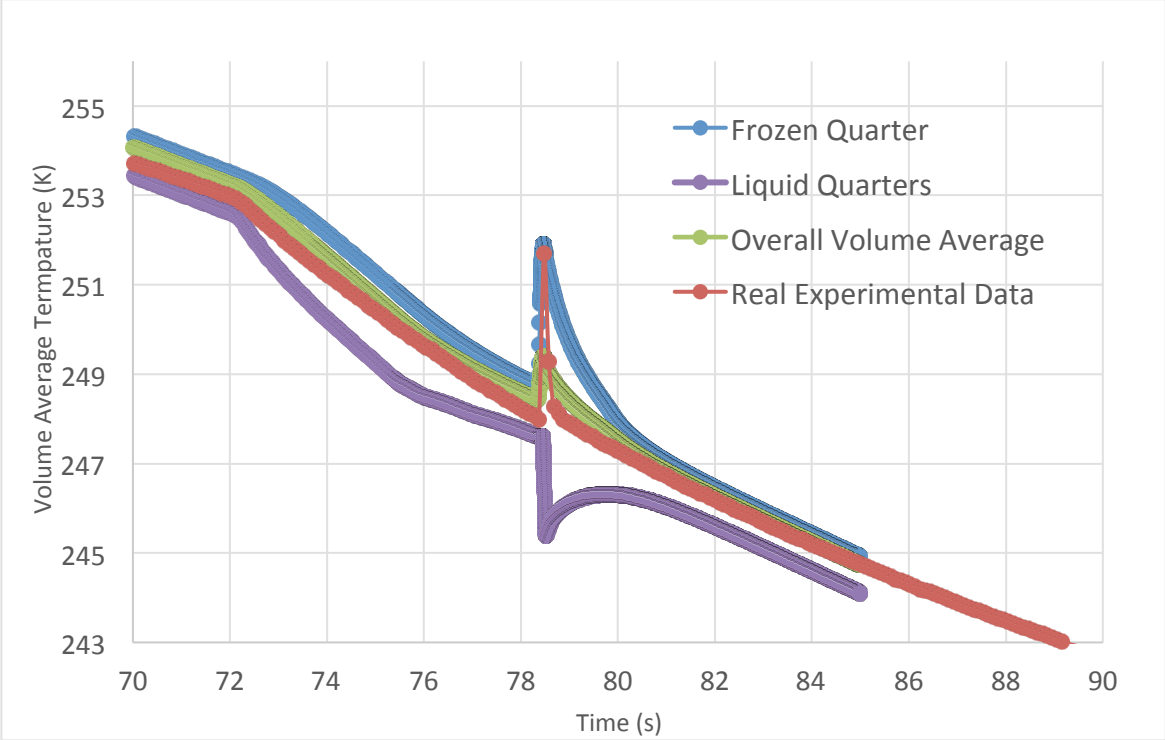


Figure 5.21 Detailed temperature profile when only one horizontal quarter of the water cube on the top is frozen

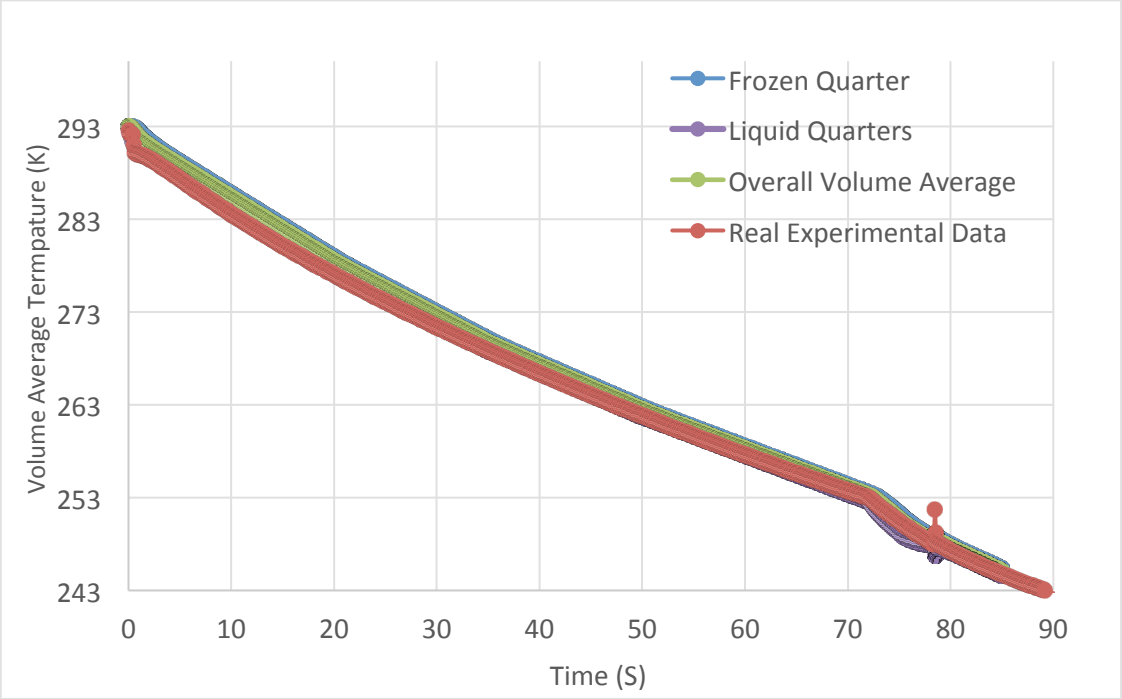


Figure 5.22 Temperature profile when only one horizontal quarter of the water cube on the bottom is frozen

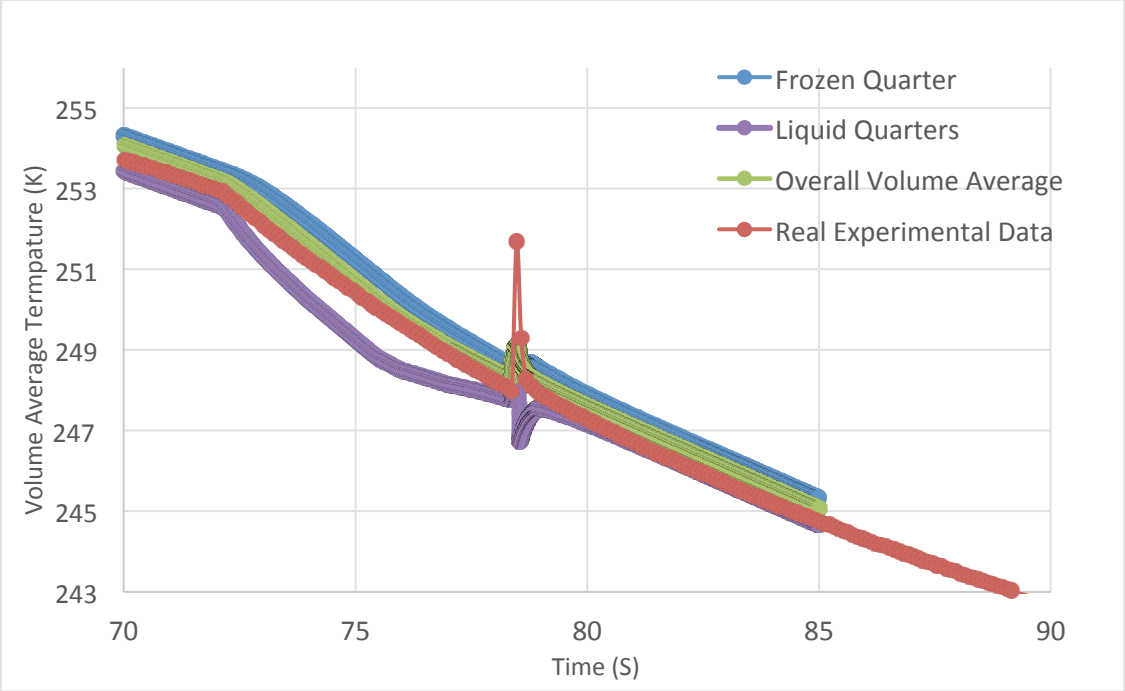


Figure 5.23 Detailed temperature profile when only one horizontal quarter of the water cube on the bottom is frozen

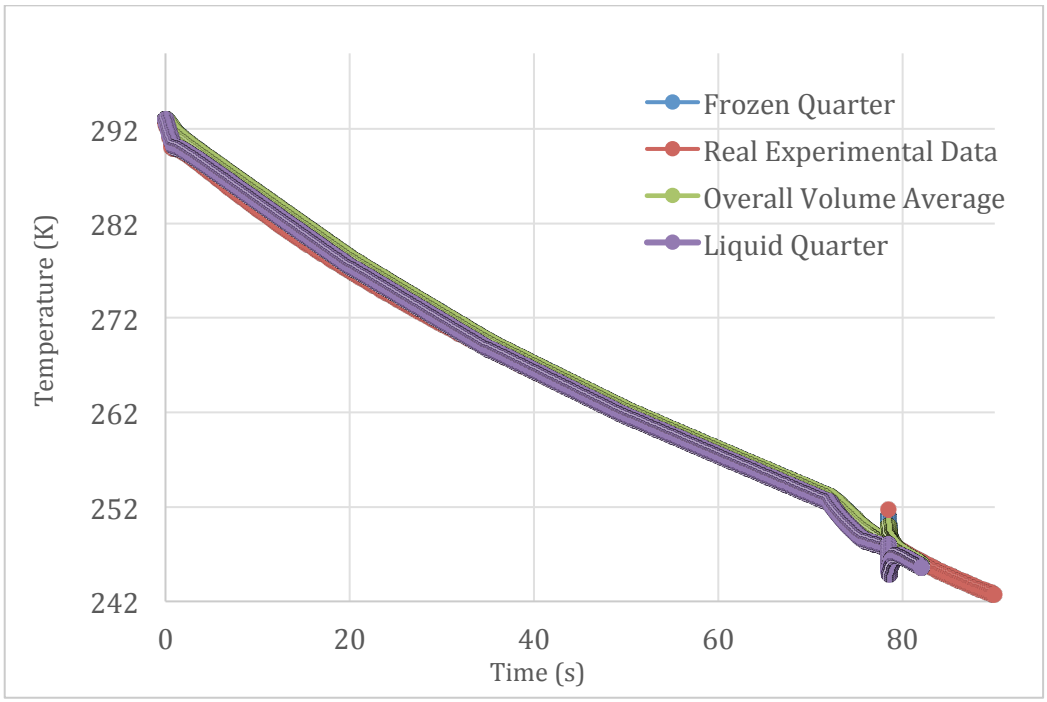


Figure 5.24 Temperature profile when 1/8 of the water cube at the top and the opposite bottom corner is frozen

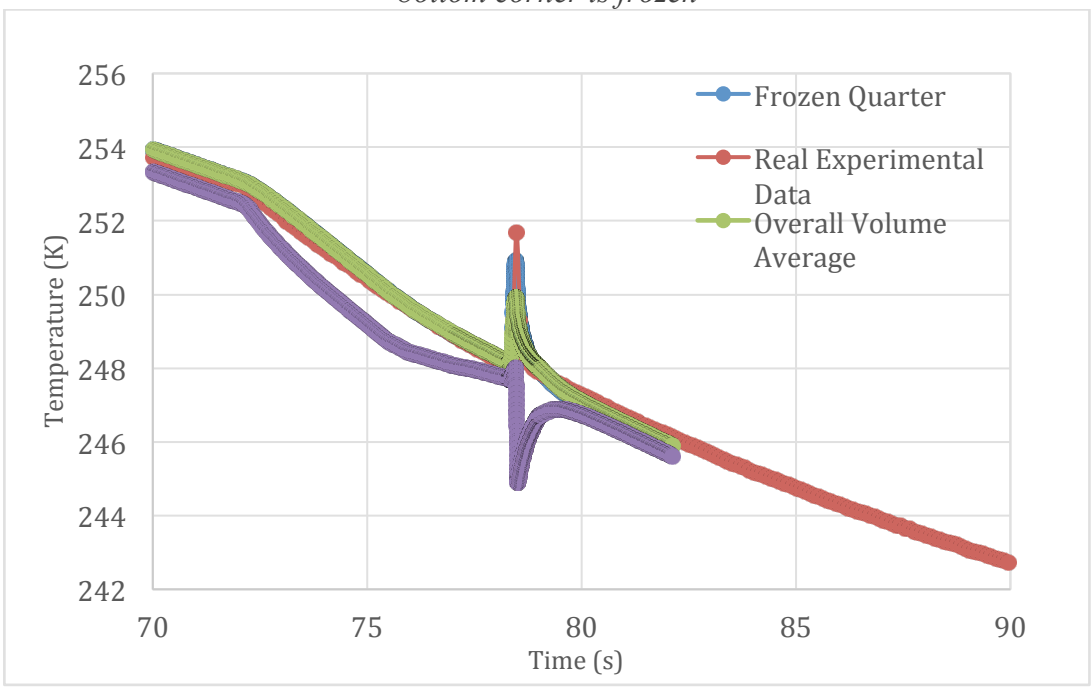


Figure 5.25 Detailed temperature profile when 1/8 of the water cube at the top and the opposite bottom corner is frozen

Table 5.2 Estimated area under the peak for 1/4 frozen water cube models

Locations	Vertical	Horizontal Top	Horizontal Bottom	Two 1/8 Water Cubes
Estimated Area Under the Peak	0.21	0.29	0.20	0.23
Simulation/Experiment Area ratio	0.25	0.35	0.24	0.27

When a quarter of the water cube is frozen at different locations within the water cube, the overall volume average temperature profiles are almost identical to each other. Comparing Fig 5.24, Fig 5.26, Fig 5.28 and Fig 5.30, the liquid quarter curves and the frozen quarter curves behave very differently during the phase change process among those four figures.

Since there is only a quarter of the mass being frozen for each scenario, the amount of latent heat being released should be the same. Therefore, the volume average temperature curves can be very similar for each location scenario. The artificially controlled frozen region locations can change the heat flow directions and the magnitude of the heat flux within the water cube during the phase change process, therefore cause different temperature curves for the frozen and the liquid regions.

5.3.3 Liquid only water cube

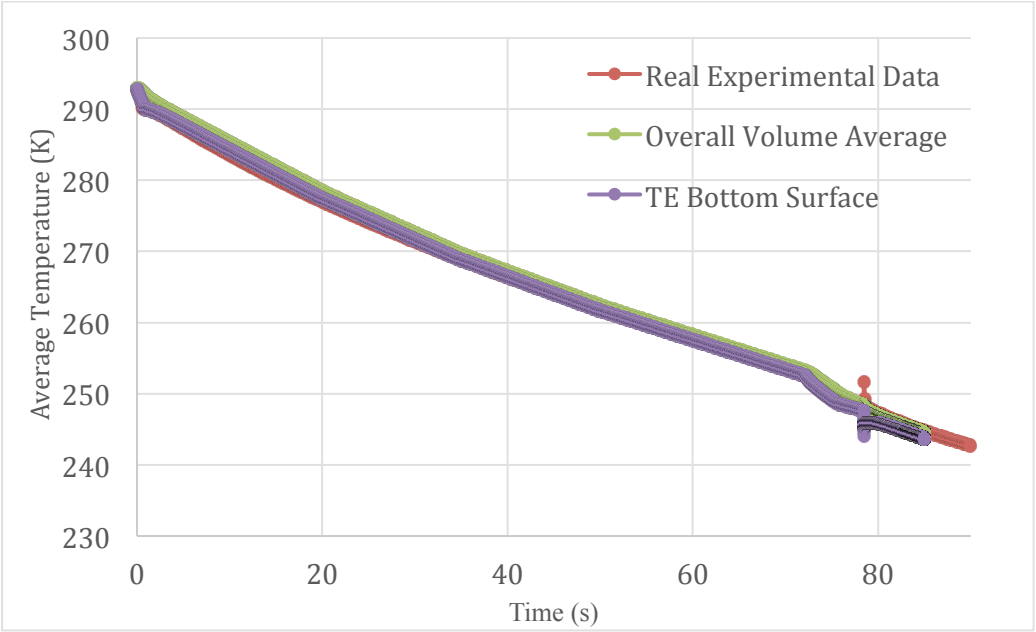


Figure 5.26 Temperature baseline for a liquid phase only water cube model

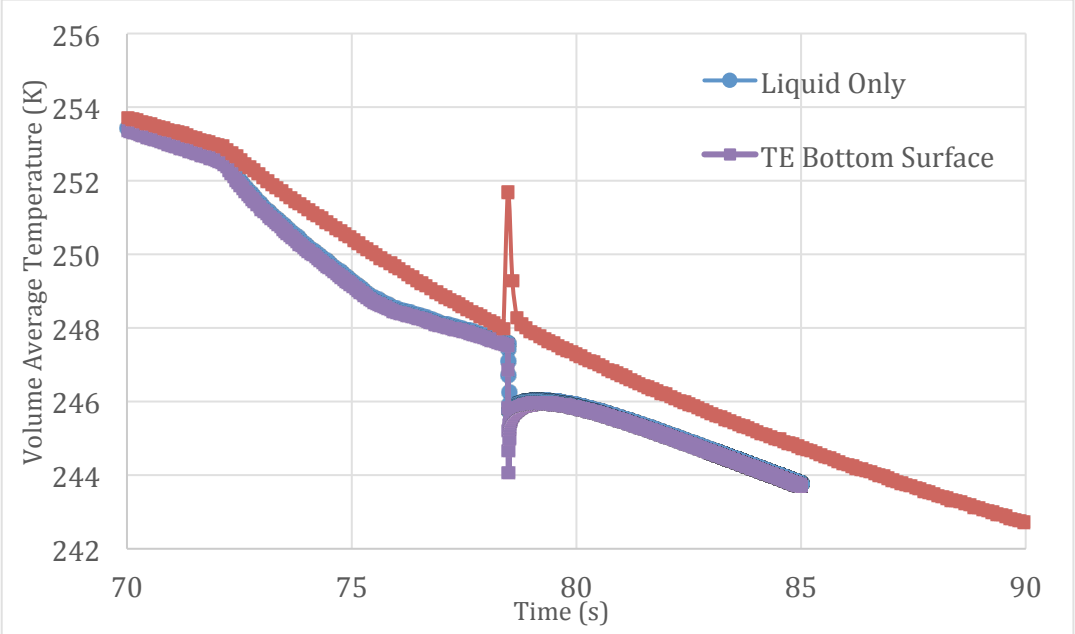


Figure 5.27 Focused temperature baseline for a liquid phase only water cube model

The liquid-phase water cube model result is a baseline to study how the amount of frozen water influences the temperature profile. As shown in Fig 5.26 and 5.27, the temperature of the entire cube drops rapidly around 78 second. This is due to the artificial

heat sink applied at the TE layer to incorporate the heat spreading effect. Similar behavior can be found in the liquid-phase only volume in partially frozen water cube simulations.

6. Conclusions

6.1 Conclusions

A 3D numeric model has been built using STAR CCM+ to simulate the freezing process in the experiments. In the numeric model, a geometry including water droplet and the TE ceramic layer was built to simulate the water droplet activity when being frozen by a thermoelectric pad. This numeric model was verified by comparing the simulation results with the analytical solutions, the simulation results generated from a MATLAB numeric model, and the experimental results.

After the 3D numeric model was calibrated, the same STAR CCM+ model was used to predict the temperature distribution of different sizes of water droplets during the freezing process. Simulations were run to predict the temperature distribution of water droplets weighing 1mg, 2 mg and 0.003 mg. In reality, it has been proven by experiments that the temperature at the highest nucleation rate increases as the water body size decreases. In the numeric result, the freezing temperature for different sizes of water droplets agrees with the experiments results.

In addition, the calibrated numeric model was modified to study fractional ice formation. Minor modifications of the physical setup of the numeric model were made to allow for 50%, 25% or 0% of the water mass being frozen. The temperature curves computed from this numeric model were validated by a rough energy conservation check. The behavior of the temperature curve for liquid only volume is very unique when 25% of water body is frozen. This is because to freeze only one quarter of the water body unsymmetrically in geometry can be physically unrealistic. When 25% of the water body is being frozen, the liquid part of the water body temperature drops very rapidly when the

freezing process starts. The artificial negative heat flux applied at the boundary that simulates the heat spread effect can draw energy from both the liquid-phase and the solid-phase part of the water body. However, being able to release latent heat during the freezing process made the solid-phase water body has a higher temperature than the part that was set to be liquid-only.

The calibrated numeric model allows us to better understand the temperature distribution of a water droplet when the water droplet is being cooled until frozen. The size of the thermoelectric pad influences the simulation result due to the heat spreading effect. By using additional heat flux to scale the simulated thermoelectric layer to a smaller size, the numeric model generated relatively accurate results within a very short computational time. This is a big improvement compared to other high fidelity computational simulation models that are used to solve similar problems. In addition to that, the partially frozen water body simulation helps to predict the behavior of a real cell when it is being frozen.

6.2 Recommendations and Future Work

The numeric model can be recreated using the same set up and geometry mentioned in the previous Chapters. The MATLAB code files that are used to validate the numeric model can be found in Appendix D.

Since other biomaterials, including cells, consist not only of water but other substances, it is very valuable to validate the fractionally frozen water cube models with carefully designed experiments in the future. Once the partially frozen water cube models are validated, those models will allow us to simulate more complex cell thermal behaviors. The artificial negative heat flux applied at the boundary that simulates the heat

spread effect can cause physically unrealistic behaviors for the quarterly frozen water cube model. Running the corresponding experiments and trying to only freeze a quarter of the water cube may provide useful hints in order to refine the numeric models.

In addition to the partially frozen model validation, the shape of the water droplet in the STAR CCM+ model is still a cube. The next step of the research is to change the cube to a semi-sphere or a geometry that is close to the real water droplet shape. The shape factor for the new model needs to be studied through simulation analysis. The quality of the simulation will depend strongly on the mesh conditions, in terms of smoothness, regularity, and distortion of the new water droplet geometry. The new model with complex physical interactions can be solved with high fidelity equation sets, although it will probably require more time.

Appendix

Appendix A

The convection heat transfer coefficient in Eqn.2.2 can be calculated by

$$\mathrm{h}_c = 0.072\left[\frac{\left[g(\frac{T_{wall}-T_{melting}}{2})\rho_l^2 c_l k_l^2 \beta\right]}{\mu}\right] \quad \text{A.(1)}$$

Where, β is the coefficient of the thermal expansion and μ is the dynamic viscosity of air.

Appendix B

The coefficients in Eqn.3.45 can be calculated by,

$$\mathrm{K}(\beta_m, x) = \sqrt{\frac{2}{L}} \cdot \sin(\beta_m x) \quad \text{B.(1)}$$

$$\beta_m \in \sin(\beta L) = 0 \cap \beta > 0 \quad \text{B.(2)}$$

$$F(x) = T_0 = const \quad \text{B.(3)}$$

Appendix C

$$B_1 = \frac{A\,\alpha}{s}\left[1-(1+\mu)^{1/2}\right] \quad \text{C.(1)}$$

$$B_2 = \frac{B_1 s + (T_0 - T_m)}{s^2} \quad \text{C.(2)}$$

$$A = \frac{\rho \cdot \mathrm{L}}{\mathrm{k}} \quad \text{C.(3)}$$

$$\mu = \frac{2(T_0 - T_m)}{\mathrm{A} \cdot \alpha} \quad \text{C.(4)}$$

$$s = K\sqrt{t} \quad \text{C.(5)}$$

$$K = \left[12\,\alpha\,\frac{1-(1+\mu)^{\frac{1}{2}}+\mu}{5+(1+\mu)^{1/2}+\mu}\right]^{1/2} \quad \text{C.(6)}$$

Appendix D

Appendix D.1

```
%% Writen by: Han Han 9-31-2014
% 1-D transient conduction validation
% Layering Scheme :Ceramic Alumina, Silicon Wafer and Silicon Dioxide,
% Titanium, Platinium, Cell, Water
%
=========================================================================
% ---------------- Section 1  Initialization of constants -------------
-
%
=========================================================================
close all,clear all,clc;
% Testing code
% Type I : check for k's. No heat source 1D conduction, constant
% temerature boundary condition.
% Type II: check for uniform distributed heat source.
% Type III: check for transient problem with: S=0,h=infinity
% Type IV : check for transient problem with: S=0,h=5
% Type V : transient problem with phase change in water layer

display('Type I : check for k. S=0,T_b=constant');
display('Type II: check for uniform distributed heat source.')
display('Type III: check for transient problem with: S=0,h=infinity.');
display('Type IV : check for transient problem with: S=0,h=5 W/m^2');
display('Type V: ozisic transient problem with: S=0,h=infinity.');
flag = input('What type of testing do you want? Please enter 1 to 5');

% cv = layer thickness in x direction [m]
cv = [6.8e-4 5e-4 3e-7 2e-9 1e-8 1e-7 1e-5];
if flag == 1
    % Testing code I
    delta_t = 999999999999;
    S = 0;
    % Define heat capacity, c_p [J/(kg*K)]
    c_p =[775 745 745 522 133 4217 4217];
    % Define density , rho = density [kg/m^3]
    rho=[3750 2220 2220 4500 21450 1000 1000];
    % Define thermal conductivity [W/m*K]
    k=[35 1.38 1.38 21.9 71.6 0.569 0.569];
    R_RTD = cv./k;
    % Num = Number of nodes in each control volume
    Num = [6800 5000 3 3 10 100 100];

    % Testing code II
elseif flag == 2
    delta_t = 999999999999;
    c_p =775*ones(1,length(cv));
    rho=3750*ones(1,length(cv));
    k= 35*ones(1,length(cv));
    Num = [6800 5000 3 3 10 100 100];

    % Testing code III
elseif flag == 3
```

```
    delta_t =0.0009;
    S = 0;
    c_p =775*ones(1,length(cv));
    rho=3750*ones(1,length(cv));
    k= 35*ones(1,length(cv));
    alpha = k./(rho.*c_p);
    Num = [6800 5000 3 3 10 100 100];
    L = sum(cv)/2;
    iteration =200;
elseif flag ==4
    delta_t = 0.01;
    S = 0;
    c_p =775*ones(1,length(cv));
    rho=3750*ones(1,length(cv));
    k= 35*ones(1,length(cv));
    Num = [6800 5000 3 3 10 100 100];
    h = 583333.33; % convection coefficient of air [W/m^2]
    L = sum(cv);
    iteration = 20;
else
    delta_t =0.0004;
    S = 0;
    c_p =4179*ones(1,length(cv));
    rho=1000*ones(1,length(cv));
    k= 0.569*ones(1,length(cv));
    alpha = k./(rho.*c_p);
    Num = [6800 5000 3 3 10 100 100];
    L = sum(cv);
    iteration =100;
end;
% ---------------------------------------------------------------------
%% Assign node and material interface position
delta_x(1)=cv(1)/(Num(1)-0.5);
delta_x(2:length(Num)-1)= cv(2:length(Num)-1)./(Num(2:length(Num)-1));
delta_x(length(Num)) = cv(end)/(Num(end)-0.5);

x = zeros(1,sum(Num));
x(1:Num(1))=(0:delta_x(1):cv(1));
for ii=1:length(Num)-2
    x(sum(Num(1:ii))+1:sum(Num(1:ii+1)))=...
    (sum(cv(1:ii))+0.5*delta_x(ii+1):delta_x(ii+1):sum(cv(1:ii+1)));
end;
x(sum(Num(1:length(Num)-1))+1:end)=...
(sum(cv(1:length(cv)-1))+0.5*delta_x(end):delta_x(end):sum(cv));
%---------------------------------------------------------------------
% Calculate the width of each cell
dx = zeros(1,length(x));
dx(1) = x(2)/2;
dx(2:Num(1))= delta_x(1);
for ii = 1:length(cv)-2
    dx(sum(Num(1:ii))+1:sum(Num(1:ii+1)))=delta_x(ii+1);
end;
dx(sum(Num(1:length(cv)-1)):end-1)=delta_x(end);
dx(end)=delta_x(end)/2;

%% ---------------------------------------------------------------------
```

```
% Define the source term S

% I = 10e-3; % current that goes through platinium layer [A]
% L = .1;     % length of RTD wire in [m]
% A = 1e-13;  % cross section area [m^2]
% R = rho(5)*L/A;
% S = zeros(1,length(x));
% % S(2201:3200)= I^2*R;
if flag==2
S =1e6;
fprintf('S = %e\n',S);
elseif flag ==5

end;
%-------------------------------------------------------------------------
--
%% Calculate equivalent Cp,rho,k values for each node
k_e=zeros(1,length(x));k_e(1:Num(1)-1)=k(1);
%k_eq is for analytic model
k_eq=zeros(1,length(x));

cp_eq=zeros(1,length(x));
cp_eq(1:Num(1))=c_p(1);
cp_eq(end)=c_p(end);
rho_eq=zeros(1,length(x));
rho_eq(1:Num(1))=rho(1);
rho_eq(end)=rho(end);

% Assign the ends of the nodes' thermal properties
k_w=zeros(1,length(x));k_w(2:Num(1))=k(1);k_w(end)=k(end);

for ii=1:length(Num)-1
    cp_eq(sum(Num(1:ii))+1:sum(Num(1:ii+1)))=c_p(ii+1);
    rho_eq(sum(Num(1:ii))+1:sum(Num(1:ii+1)))=rho(ii+1);

    k_e(sum(Num(1:ii))+1:sum(Num(1:ii+1))-1)=k(ii+1);
    k_w(sum(Num(1:ii))+2:sum(Num(1:ii+1)))=k(ii+1);

    delta_x_minus = sum(cv(1:ii))-x(sum(Num(1:ii)));
    delta_x_plus = x(sum(Num(1:ii))+1)-sum(cv(1:ii));
    delta_x_boundary = x(sum(Num(1:ii))+1)-x(sum(Num(1:ii)));
    f_minus = delta_x_minus/delta_x_boundary;
    f_plus = delta_x_plus/delta_x_boundary;

    k_e(sum(Num(1:ii)))=...
        k(ii)*k(ii+1)/(f_plus*k(ii)+f_minus*k(ii+1));

    k_w(sum(Num(1:ii))+1)=...
        k(ii)*k(ii+1)/(f_plus*k(ii)+f_minus*k(ii+1));

end;

k_eq=[k(1),k_e(1:end-1)];
%-------------------------------------------------------------------------
--
```

```
%% Pre Allocating memory
a_p0 = zeros(1,length(x));
a = zeros(1,length(x));
b = zeros(1,length(x));
c = zeros(1,length(x));
d = zeros(1,length(x));
P = zeros(1,length(x));
Q = zeros(1,length(x));
T = zeros(1,length(x));
%-------------------------------------------------------------------------
--
%% Chose boundary condition for testing
% Type I B.C. (Constant Temperature)
if flag == 1
T_boundary_1 =288; %cv1 boundary temperature [K]
T_boundary_2 =237; %cv7 boundary temperature [K]
T(1)= T_boundary_1;
T(end) = T_boundary_2;
T_initial = 273+20;

% Type II B.C. (Constant Heat Flux)
elseif flag == 2
T_boundary_1 =288; %cv1 boundary temperature [K]
T_boundary_2 =237; %cv7 boundary temperature [K]
T(1)= T_boundary_1;
T(end) = T_boundary_2;
T_initial = 273+20;

% Type III B.C. (Constant Temperature)
elseif flag == 3
T_inf =237; %T infinity [K]
T_initial =  273+20;% Initial temperature for all the nodes [K]

TT = T_initial*ones(iteration,length(x));
TT(:,1)= T_inf;
TT(:,end) = T_inf;
T_ana = TT;

elseif flag == 4

T_inf_1 =288; %cv1 boundary temperature [K]
T_inf_2 =237; %cv7 boundary temperature [K]
T_initial = 273+20;% Initial temperature for all the nodes [K]

TT = T_initial*ones(iteration,length(x));
TT(:,1)= T_inf_1;
TT(:,end) = T_inf_2;
T_ana = TT;

else

T_inf =0; %T infinity [K]
T_initial = 25;% Initial temperature for all the nodes [K]

TT = T_initial*ones(iteration,length(x));
```

```
TT(:,1)= T_inf;
TT(:,end) = T_inf;
T_ana = TT;

end;

%%
%
========================================================================
% --------- Section 2  Temperature distribution Calculation -----------
%
========================================================================
% Coefficients Calculation
if flag==1
             jj = 10;
            % Time [s]
            Time = jj*delta_t
            while jj>1
            a_p0 = rho_eq.*cp_eq.*dx./delta_t;

            a_E = k_e(1:end-1)./dx(2:end);
            a_E = [a_E,0];
            a_W = k_w(2:end)./dx(1:(end-1));
            a_W = [0,a_W];


            a = a_p0 + a_E + a_W;
            a(1) = 1;
            a(end)=1;
            b = a_E;
            b(1) = 0;
            c = a_W;
            c(end) = 0;
            d = a_p0.*T+S.*dx;
            d(1) = T(1);
            d(end) = T(end);
            P(1)= b(1)/a(1);
            Q(1)=d(1)/a(1);
            P(end) = 0;
            Q(end)= T(end);


            for ii= 2:length(x)-1
               P(ii) = b(ii)/(a(ii)-c(ii)*P(ii-1));
               Q(ii) = (c(ii)*Q(ii-1)+d(ii))/(a(ii)-c(ii)*P(ii-1));
            end;
               ii = length(x)-1;
            while ii>1
                T(ii)=P(ii)*T(ii+1)+Q(ii);
                ii=ii-1;
            end;

            jj = jj-1;
            end;
```

```
elseif flag == 2
            jj = 10;
            % Time [s]
            Time = jj*delta_t
            while jj>1
            a_p0 = rho_eq.*cp_eq.*dx./delta_t;

            a_E = k_e(1:end-1)./dx(2:end);
            a_E = [a_E,0];
            a_W = k_w(2:end)./dx(1:(end-1));
            a_W = [0,a_W];


            a = a_p0 + a_E + a_W;
            a(1) = 1;
            a(end)=1;
            b = a_E;
            b(1) = 0;
            c = a_W;
            c(end) = 0;
            d = a_p0.*T+S.*dx;
            d(1) = T(1);
            d(end) = T(end);
            P(1)= b(1)/a(1);
            Q(1)=d(1)/a(1);
            P(end) = 0;
            Q(end)= T(end);


            for ii= 2:length(x)-1
               P(ii) = b(ii)/(a(ii)-c(ii)*P(ii-1));
               Q(ii) = (c(ii)*Q(ii-1)+d(ii))/(a(ii)-c(ii)*P(ii-1));
            end;
               ii = length(x)-1;
            while ii>1
                T(ii)=P(ii)*T(ii+1)+Q(ii);
                ii=ii-1;
            end;

            jj = jj-1;
            end;

elseif flag==3
            Time = 0;
            kk=iteration-1;
            Time = delta_t+Time;
            a_p0 = rho_eq.*cp_eq.*dx./delta_t;
            a_E = k_e(1:end-1)./dx(2:end);
            a_E = [a_E,0];
            a_W = k_w(2:end)./dx(1:(end-1));
            a_W = [0,a_W];

            a = a_p0 + a_E + a_W;
            a(1) = 1;
            a(end)=1;
```

```
b = a_E;
b(1) = 0;
c = a_W;
c(end) = 0;
d = a_p0.*TT(1,:)+S.*dx;
d(1) = TT(1,1);
d(end) = TT(1,end);
P(1)= b(1)/a(1);
Q(1)=d(1)/a(1);
P(end) = 0;
Q(end)= TT(1,end);

for ii= 2:length(x)-1
   P(ii) = b(ii)/(a(ii)-c(ii)*P(ii-1));
   Q(ii) = (c(ii)*Q(ii-1)+d(ii))/(a(ii)-c(ii)*P(ii-1));
end;
ii = length(x)-1;
while ii>1
    TT(1,ii)=P(ii)*TT(1,ii+1)+Q(ii);
    ii=ii-1;
end;

while kk>0
        count = iteration-kk+1
        Time = delta_t+Time;
        a_p0 = rho_eq.*cp_eq.*dx./delta_t;
        a_E = k_e(1:end-1)./dx(2:end);
        a_E = [a_E,0];
        a_W = k_w(2:end)./dx(1:(end-1));
        a_W = [0,a_W];

        a = a_p0 + a_E + a_W;
        a(1) = 1;
        a(end)=1;
        b = a_E;
        b(1) = 0;
        c = a_W;
        c(end) = 0;
        d = a_p0.*TT(count-1,:)+S.*dx;
        d(1) = TT(count-1,1);
        d(end) = TT(count-1,end);
        P(1)= b(1)/a(1);
        Q(1)=d(1)/a(1);
        P(end) = 0;
        Q(end)= TT(count-1,end);
        for ii= 2:length(x)-1
           P(ii) = b(ii)/(a(ii)-c(ii)*P(ii-1));
           Q(ii) = (c(ii)*Q(ii-1)+d(ii))/(a(ii)-c(ii)*P(ii-
1));
        end;
        ii = length(x)-1;
        while ii>1
            TT(count,ii)=P(ii)*TT(count,ii+1)+Q(ii);
            ii=ii-1;
        end;
        kk = kk-1;
```

```
            T_ana(count,:) = (1.2733.*exp(-1*(1.57^2).*alpha(1).*...
            Time./(L.^2))).*cos(1.5708*((x-L)./L)).*...
            (T_initial-T_inf)+T_inf;
        end;



elseif flag ==4
    % Assign all the coefficients a,b,c,d,P and Q
        Time = 0;
        kk=iteration-1;
        Time = delta_t+Time;
        a_p0 = rho_eq.*cp_eq.*dx./delta_t;
        a_E = k_e(1:end-1)./dx(2:end);
        a_E = [a_E,0];
        a_W = k_w(2:end)./dx(1:(end-1));
        a_W = [0,a_W];

        a = a_p0 + a_E + a_W;
        b = a_E;
        c = a_W;
        d = a_p0.*TT(1,:)+S.*dx;

        a(1) = k_eq(1)/dx(1)+h;
        a(end)=k_eq(end-1)/dx(end);
        b(1) = k_eq(2)/dx(1);
        b(end)=k_eq(end)/dx(end)+h;
        c(1)=0;
        c(end) = 0;
        d(1) = h*T_inf_1+0.5*S*dx(end);
        d(end) =0.5*S*dx(end)-h*T_inf_2;

        P(1)= b(1)/a(1);
        Q(1)=d(1)/a(1);
        P(end) = 0;
        Q(end)= TT(1,end);

        for ii= 2:length(x)-1
            P(ii) = b(ii)/(a(ii)-c(ii)*P(ii-1));
            Q(ii) = (c(ii)*Q(ii-1)+d(ii))/(a(ii)-c(ii)*P(ii-1));
        end;
        % Calculate the first iteration of TT
        ii = length(x)-1;
        while ii>1
            TT(1,ii)=P(ii)*TT(1,ii+1)+Q(ii);
            ii=ii-1;
        end;

        while kk>0
                count = iteration-kk+1
                Time = delta_t+Time;
                a_p0 = rho_eq.*cp_eq.*dx./delta_t;
                a_E = k_e(1:end-1)./dx(2:end);
                a_E = [a_E,0];
                a_W = k_w(2:end)./dx(1:(end-1));
```

```
                    a_W = [0,a_W];

                    a = a_p0 + a_E + a_W;
                    b = a_E;
                    c = a_W;
                    d = a_p0.*TT(count-1,:)+S.*dx;

                    a(1) = k_eq(1)/dx(1)+h;
                    a(end)=k_eq(end-1)/dx(end);
                    b(1) = k_eq(2)/dx(1);
                    b(end)=k_eq(end)/dx(end)+h;
                    c(1)=0;
                    c(end) = 0;
                    d(1) = h*T_inf_1+0.5*S*dx(end);
                    d(end) =0.5*S*dx(end)-h*T_inf_2;

                    P(1)= b(1)/a(1);
                    Q(1)=d(1)/a(1);
                    P(end) = 0;
                    Q(end)= TT(count-1,end);

                    for ii= 2:length(x)-1
                       P(ii) = b(ii)/(a(ii)-c(ii)*P(ii-1));
                       Q(ii) = (c(ii)*Q(ii-1)+d(ii))/(a(ii)-c(ii)*P(ii-
1));
                    end;
                    ii = length(x)-1;
                    while ii>1
                        TT(count,ii)=P(ii)*TT(count,ii+1)+Q(ii);
                        ii=ii-1;
                    end;
                    kk = kk-1;
            end;

else
     Time = 0;
            kk=iteration-1;
            Time = delta_t+Time;
            a_p0 = rho_eq.*cp_eq.*dx./delta_t;
            a_E = k_e(1:end-1)./dx(2:end);
            a_E = [a_E,0];
            a_W = k_w(2:end)./dx(1:(end-1));
            a_W = [0,a_W];

            a = a_p0 + a_E + a_W;
            a(1) = 1;
            a(end)=1;
            b = a_E;
            b(1) = 0;
            c = a_W;
            c(end) = 0;
            d = a_p0.*TT(1,:)+S.*dx;
            d(1) = TT(1,1);
            d(end) = TT(1,end);
            P(1)= b(1)/a(1);
            Q(1)=d(1)/a(1);
```

```
            P(end) = 0;
            Q(end)= TT(1,end);

            for ii= 2:length(x)-1
               P(ii) = b(ii)/(a(ii)-c(ii)*P(ii-1));
               Q(ii) = (c(ii)*Q(ii-1)+d(ii))/(a(ii)-c(ii)*P(ii-1));
            end;
            ii = length(x)-1;
            while ii>1
                TT(1,ii)=P(ii)*TT(1,ii+1)+Q(ii);
                ii=ii-1;
            end;

            while kk>0
                    count = iteration-kk+1
                    Time = delta_t+Time;
                    a_p0 = rho_eq.*cp_eq.*dx./delta_t;
                    a_E = k_e(1:end-1)./dx(2:end);
                    a_E = [a_E,0];
                    a_W = k_w(2:end)./dx(1:(end-1));
                    a_W = [0,a_W];

                    a = a_p0 + a_E + a_W;
                    a(1) = 1;
                    a(end)=1;
                    b = a_E;
                    b(1) = 0;
                    c = a_W;
                    c(end) = 0;
                    d = a_p0.*TT(count-1,:)+S.*dx;
                    d(1) = TT(count-1,1);
                    d(end) = TT(count-1,end);
                    P(1)= b(1)/a(1);
                    Q(1)=d(1)/a(1);
                    P(end) = 0;
                    Q(end)= TT(count-1,end);
                    for ii= 2:length(x)-1
                       P(ii) = b(ii)/(a(ii)-c(ii)*P(ii-1));
                       Q(ii) = (c(ii)*Q(ii-1)+d(ii))/(a(ii)-c(ii)*P(ii-
1));
                    end;
                    ii = length(x)-1;
                    while ii>1
                        TT(count,ii)=P(ii)*TT(count,ii+1)+Q(ii);
                        ii=ii-1;
                    end;
                    kk = kk-1;

            end;

end;

%%
=========================================================================
%  ----------------- Section 3 Plot resluts for Testing ---------------
---
```

```
%
=========================================================================
if flag==1
    % R = dx./k_eq;
    R = zeros(1,length(x));
    T_ana = zeros(1,length(R));
    T_ana(1) = T_boundary_1;
    T_ana(end) = T_boundary_2;

    for ii = 2:length(R)
    R(ii) = dx(ii-1)/k_eq(ii-1)+0.5*dx(ii)/k_eq(ii);
    end;
    q =(T_boundary_1-T_boundary_2)/sum(R);
    for ii=1:length(R)
    T_ana(ii) = T_boundary_1-q*sum(R(1:ii));
    end;
    %calculate the mean difference between two lines
    T_diff = mean(abs(T_ana - T))

    figure
    plot(x,T,'k-'),hold on , plot(x,T_ana,'r--');
    title('Temperature vs x');
    legend('Numerical','Analytical');


elseif flag==2
    S = S/k(1);
    c2=T_boundary_1
    c1=(T_boundary_2 - T_boundary_1+S/2*(sum(cv)^2))/sum(cv)
    T_analytic = (-1*S)/2.*(x.^2)+c1.*x+c2;
    figure
    plot(x,T,'k-'),hold on, plot(x,T_analytic,'r--');
    legend('Numerical','Analytical');
    title('Temperature vs x while S ~10^6w/m^3');
    T_diff = mean(abs(T_analytic - T))

elseif flag==3

    figure,
    plot(x,TT(1,:),'k-'),hold on, plot(x,TT(iteration/4,:),'r-'),...
    plot(x,TT(iteration/2,:),'g-'), plot(x,TT(end,:),'c-');...
    plot(x,T_ana(1,:),'k--'),plot(x,T_ana(iteration/4,:),'r--'),...
    plot(x,T_ana(iteration/2,:),'g--'), plot(x,TT(end,:),'c--');
    legend('1 time step','1/4 time','1/2 time ','total time',...
        '1 time step Analytical','1/4 time Analytical',...
        '1/2 time Analytical ','total time Analytical');
    title('Temperature vs x at different time step 0.0009 X 200');

    T_diff = mean( mean(abs(T_ana - TT)))

elseif flag==4
    %Calculate the analytical results for Type VI   !!!!!!!!
    alpha = k_eq./(rho_eq.*cp_eq);
    H(1:2)=h./k_eq(1,end);
    B(1:2)=h*L./k_eq(1,end);
```

```
    % Find Xi,K,F
    m =4;
    Xi=zeros(1,m);
    K=zeros(m,length(x));
    X=K;
    F=T_initial*ones(m,length(x));
    T=K;
    intergral=K;
    Xi=[2.86 5.73 8.61 11.52];
    beta=Xi./L;

   for jj=1:length(x)
       X(:,jj) =beta.*cos(beta.*x(jj))+H(1).*sin(beta.*x(jj));
       K(:,jj) =sqrt(2).*(X(:,jj)')./(((beta.^2+H(1)^2).*(L+H(2)./...
           (beta.^2+H(2)^2))+H(1)).^0.5);
       intergral(:,jj)=sqrt(2).*(((beta.^2+H(1)^2).*(L+H(2)./...
           (beta.^2+H(2)^2))+H(1)).^-0.5).*(sin(beta.*x(jj))-H(1)./...
           beta.*cos(beta.*x(jj)));
   end;

    % Calculate T_ana(x,t)
   Time = 0;
   kk=iteration;

   for ii=1:kk
        Time = delta_t+Time;
        term1 = exp(-1.*alpha(jj).*Time.*((beta').^2));
        term2 = K.*F.*intergral;
        for jj=1:m
           T(jj,:)=term1(jj).*term2(jj,:);
        end;
        T_ana(ii,:)=sum(T);
   end;
    figure,
    plot(x,TT(1,:),'k-'),hold on, plot(x,TT(iteration/4,:),'r-'),...
    plot(x,TT(iteration/2,:),'g-'), plot(x,TT(end,:),'c-');...
    plot(x,T_ana(1,:),'k--'),plot(x,T_ana(iteration/4,:),'r--'),...
    plot(x,T_ana(iteration/2,:),'g--'), plot(x,TT(end,:),'c--');
    legend('1 time step','1/4 time','1/2 time ','total time',...
        '1 time step Analytical','1/4 time Analytical',...
        '1/2 time Analytical ','total time Analytical');
    title('Temperature vs x at different time step 0.0009 X 200');

    T_diff =mean( mean(abs(T_ana - TT)))
    figure,
    plot(x,TT(1,:),'k-'),hold on, plot(x,TT(iteration/4,:),'r-'),...
    plot(x,TT(iteration/2,:),'g-'), plot(x,TT(end,:),'c-');
    legend('1 time step','1/4 time','1/2 time ','total time');
    title('Temperature vs x at different time step 0.0002 X 100');
% ------------------------------------------------------------------------
-
% TESTING TYPE V
else
    %Calculate the analytical results
    alpha = k_eq./(rho_eq.*cp_eq);
```

```
    % Find Xi,K,F
    m=100;
    K=zeros(m,length(x));
    F=T_initial*ones(m,length(x));
    T=K;
    intergral=K;
    n = (1:1:m);
    beta=n.*pi./L;

   for jj=1:length(x)
       for ii=1:m
       K(ii,jj) = sqrt(2/L).*sin(beta(ii).*x(jj));
       intergral(ii,jj)=sqrt(2/L).*((1./beta(ii))-cos(beta(ii).*L)./...
           beta(ii));
       end;
   end;

    % Calculate T_ana(x,t)
   Time = 0;
   kk=iteration;

   for ii=1:kk

        Time = delta_t+Time;
        for jj = 1:m
        term1 = exp(-1.*alpha(1).*Time.*(beta(jj).^2));
        term2 = K(jj,:).*intergral(jj,:);
        T(jj,:)=term1.*term2;
        end;
        T_ana(ii,:)=T_initial.*sum(T);
   end;


    figure,
    plot(x,TT(1,:),'k-'),hold on, plot(x,TT(iteration/4,:),'r-'),...
    plot(x,TT(iteration/2,:),'g-'), plot(x,TT(end,:),'c-');...
    plot(x,T_ana(1,:),'k--'),plot(x,T_ana(iteration/4,:),'r--'),...
    plot(x,T_ana(iteration/2,:),'g--'), plot(x,TT(end,:),'c--');
    legend('1 time step','1/4 time','1/2 time ','total time',...
        '1 time step Analytical','1/4 time Analytical',...
        '1/2 time Analytical ','total time Analytical');
    title('Time step 0.0004 X 100 avg_T_d_i_f_f=0.1150');
    xlabel('position x (m)');
    ylabel('temperature (initial T =25 K)');
    T_diff =mean( mean(abs(T_ana - TT)))
end;
```

Appendix D.2

```
% Semi-infinite surface melting
%
========================================================================
% ---------------- Section 1  Initialization of constants -------------
-
%
========================================================================
close all,clear all,clc;
% For purpose of testing, assume:
% 1)constant thermal properties
% 2)semi-infinite surface

% cv = layer thickness in x direction [m]
% cv = [6.8e-4 5e-4 3e-7 2e-9 1e-8 1e-7 1e-5];
cv=[1e-4,1e-4,1e-4,1e-4,1e-4,1e-4,1e-4];

% c_p =4186*ones(1,length(cv));
% rho=997.561*ones(1,length(cv));
% k= 0.569*ones(1,length(cv));
%
c_p =2050*ones(1,length(cv));
rho=916.2*ones(1,length(cv));
k=2.22*ones(1,length(cv));

% Num = Number of nodes in each control volume
%  Num = [6800 5000 3 3 10 100 100];
Num=100.*[10,10,10,10,10,10,10];

% Define the thermal properties for phase change
Ts=273.15; % Temperature of ice in [K]
Tl=273.5;  % Temperature of water in [K]

ks=2.22;
kl=0.569;%W/mK
cps=2050;
cpl=4186; % J/Kg
rhos=916.2;
rhol=999.8;% Kg/m^3
lambda=334000; %[J/kg]

%
% ks=0.569;
% kl=0.569;%W/mK
% cps=4186;
% cpl=4186; % J/Kg
% rhos=997.561;
% rhol=997.561;% Kg/m^3
% lambda=334000; %[J/kg]


% delta_t =1.75e-2; % Time step
delta_t =1.75e-2; % Time step

iteration =3;
```

```
R_RTD = cv./k;
L=sum(cv);
% ------------------------------------------------------------------
--
%% Assign node and material interface position
delta_x(1)=cv(1)/(Num(1)-0.5);
delta_x(2:length(Num)-1)= cv(2:length(Num)-1)./(Num(2:length(Num)-1));
delta_x(length(Num)) = cv(end)/(Num(end)-0.5);

x = zeros(1,sum(Num));
x(1:Num(1))=(0:delta_x(1):cv(1));
for ii=1:length(Num)-2
    x(sum(Num(1:ii))+1:sum(Num(1:ii+1)))=...
    (sum(cv(1:ii))+0.5*delta_x(ii+1):delta_x(ii+1):sum(cv(1:ii+1)));
end;
x(sum(Num(1:length(Num)-1))+1:end)=...
(sum(cv(1:length(cv)-1))+0.5*delta_x(end):delta_x(end):sum(cv));
%------------------------------------------------------------------
-
% Calculate the width of each cell
dx = zeros(1,length(x));
dx(1) = x(2)/2;
dx(2:Num(1))= delta_x(1);
for ii = 1:length(cv)-2
    dx(sum(Num(1:ii))+1:sum(Num(1:ii+1)))=delta_x(ii+1);
end;
dx(sum(Num(1:length(cv)-1)):end-1)=delta_x(end);
dx(end)=delta_x(end)/2;

%% ------------------------------------------------------------------
-
% Define the source term S
Sc = zeros(1,length(x));
sc_mem=zeros(iteration,length(x));
%------------------------------------------------------------------
--
%% Calculate equivalent Cp,rho,k values for each node
k_e=zeros(1,length(x));k_e(1:Num(1)-1)=k(1);
%k_eq is for analytic model
k_eq=zeros(1,length(x));

cp_eq=zeros(1,length(x));
cp_eq(1:Num(1))=c_p(1);
cp_eq(end)=c_p(end);
rho_eq=zeros(1,length(x));
rho_eq(1:Num(1))=rho(1);
rho_eq(end)=rho(end);

% Assign the ends of the nodes' thermal properties
k_w=zeros(1,length(x));k_w(2:Num(1))=k(1);k_w(end)=k(end);

for ii=1:length(Num)-1
    cp_eq(sum(Num(1:ii))+1:sum(Num(1:ii+1)))=c_p(ii+1);
    rho_eq(sum(Num(1:ii))+1:sum(Num(1:ii+1)))=rho(ii+1);

    k_e(sum(Num(1:ii))+1:sum(Num(1:ii+1))-1)=k(ii+1);
```

```
    k_w(sum(Num(1:ii))+2:sum(Num(1:ii+1)))=k(ii+1);

    delta_x_minus = sum(cv(1:ii))-x(sum(Num(1:ii)));
    delta_x_plus = x(sum(Num(1:ii))+1)-sum(cv(1:ii));
    delta_x_boundary = x(sum(Num(1:ii))+1)-x(sum(Num(1:ii)));
    f_minus = delta_x_minus/delta_x_boundary;
    f_plus = delta_x_plus/delta_x_boundary;

    k_e(sum(Num(1:ii)))=...
        k(ii)*k(ii+1)/(f_plus*k(ii)+f_minus*k(ii+1));

    k_w(sum(Num(1:ii))+1)=...
        k(ii)*k(ii+1)/(f_plus*k(ii)+f_minus*k(ii+1));

end;

k_eq=[k(1),k_e(1:end-1)];
alpha =k_eq./(rho_eq.*cp_eq);
% A=rho_eq.*lambda./k_eq;
A=rho_eq.*lambda./k_eq;
Fo=alpha.*delta_t./L;
Fo(1)
%-------------------------------------------------------------------------
--
%% Pre Allocating memory
a_p0 = zeros(1,length(x));
a = zeros(1,length(x));
b = zeros(1,length(x));
c = zeros(1,length(x));
d = zeros(1,length(x));
P = zeros(1,length(x));
Q = zeros(1,length(x));
T = zeros(1,length(x));
%-------------------------------------------------------------------------
--
% %% Chose boundary condition
%
% T_boundary_1 =237; %cv1 boundary temperature [K]
% % Initial temperature for all the nodes [K]
% T_initial = 277;
% T_boundary_2 =T_initial; %cv7 boundary temperature [K]
%
%
% TT = T_initial*ones(iteration,length(x));
% TT(:,1)= T_boundary_1;
%
% T_ana = TT;
% miu=2.*(Ts-T_boundary_1)./(A.*alpha);
% K=sqrt(12.*alpha.*(1-sqrt(1+miu)+miu)./(5+sqrt(1+miu)+miu));
%% Chose boundary condition

T_boundary_1 =278; %cv1 boundary temperature [K]
% Initial temperature for all the nodes [K]
T_initial = 273.15;
T_boundary_2 =T_initial; %cv7 boundary temperature [K]
T0 = T_boundary_1;
```

```
Tm = 273.15;

TT = T_initial*ones(iteration,length(x));
TT(:,1)= T_boundary_1;

T_ana = TT;
miu=2.*(T0-Tm)./(A.*alpha);
K=sqrt(12.*alpha.*(1-sqrt(1+miu)+miu)./(5+sqrt(1+miu)+miu));

%%
%
========================================================================
% --------- Section 2  Temperature distribution Calculation -----------
%
========================================================================
% Coefficients Calculation
Time = 0;
kk=iteration-1;
Time = delta_t+Time;
a_p0 = rho_eq.*cp_eq.*dx./delta_t;
a_E = k_e(1:end-1)./dx(2:end);
a_E = [a_E,0];
a_W = k_w(2:end)./dx(1:(end-1));
a_W = [0,a_W];

a = a_p0 + a_E + a_W;
b = a_E;
c = a_W;
d = a_p0.*TT(1,:)+Sc.*dx;

a(1) = 1;
b(1) = 0;
c(1) = 0;
d(1) = TT(1,1);

a(end)=k_eq(end)/dx(end);
b(end)= 0;
c(end) =k_eq(end)/dx(end) ;
% d(end) =-0.5.*Sc(end).*dx(end);
d(end) =-0.5.*Sc(end).*dx(end);

P(1)= b(1)/a(1);
Q(1)=d(1)/a(1);
P(end)=0;
Q(end)=TT(1,end);

for ii= 2:length(x)-1
    P(ii) = b(ii)/(a(ii)-c(ii)*P(ii-1));
    Q(ii) = (c(ii)*Q(ii-1)+d(ii))/(a(ii)-c(ii)*P(ii-1));

end;

TT(1,end)=c(end)/a(end)*TT(1,end-1)+d(end);

ii = length(x)-1;
```

```
while ii>1
    TT(1,ii)=P(ii)*TT(1,ii+1)+Q(ii);

    ii=ii-1;
end;

S=K.*sqrt(Time);
B=A.*alpha.*(1-sqrt(1+miu))./S;
C=(B.*S+(T0-Tm))./(S.^2);
T_ana(1,:)=Tm+B.*(x-S)+C.*((x-S).^2);
%% ------------------------------------------------------------------------
----
% Define phase change relevant constants
% fl=fluid fraction
fl=zeros(1,length(x));
cpcount=zeros(iteration+1,length(x));
cpcount(1,:)=cp_eq;
if flag==1
for mm=1:length(x)
%     if TT(1,mm)<=Ts
%         fl(1,mm)=0;
%     elseif TT(1,mm)>=Tl
%         fl(1,mm)=1;
%     else
%         fl(1,mm)=(TT(1,mm)-Ts)/(Tl-Ts);
%     end
%     k_eq(1,mm)= fl(1,mm)*kl+(1- fl(1,mm))*ks;
%     rho_eq(1,mm)= fl(1,mm)*rhol+(1- fl(1,mm))*rhos;
%     cp_eq(1,mm)= fl(1,mm)*cpl+(1- fl(1,mm))*cps;
%
    if TT(1,mm)<=Ts
        fl(1,mm)=0;
    elseif TT(1,mm)>=Tl
        fl(1,mm)=1;
    else
        fl(1,mm)=(TT(1,mm)-Ts)/(Tl-Ts);
    end
    k_eq(1,mm)= fl(1,mm)*kl+(1- fl(1,mm))*ks;
    rho_eq(1,mm)= fl(1,mm)*rhol+(1- fl(1,mm))*rhos;
    cp_eq(1,mm)= fl(1,mm)*cpl+(1- fl(1,mm))*cps;


end;
end;
cpcount(2,:)=cp_eq;
%% ------------------------------------------------------------------------
----
% Start from the second iteration
while kk>0
    fl_old=fl;
    count = iteration-kk+1
    Time = delta_t+Time;

%     TT(count,end) = TT(count-1,end)-
Sc(end).*0.5.*dx(end).^2/k_eq(end);
```

```
a_p0 = rho_eq.*cp_eq.*dx./delta_t;
a_E = k_e(1:end-1)./dx(2:end);
a_E = [a_E,0];
a_W = k_w(2:end)./dx(1:(end-1));
a_W = [0,a_W];

a = a_p0 + a_E + a_W;
b = a_E;
c = a_W;
d = a_p0.*TT(1,:)+Sc.*dx;

a(1) = 1;
b(1) = 0;
c(1) = 0;
d(1) = TT(1,1);

a(end)=k_eq(end)/dx(end);
b(end)= 0;
c(end) = k_eq(end)/dx(end);
%     d(end) =-0.5.*Sc(end).*dx(end);
d(end) =-0.5.*Sc(end).*dx(end);
P(1)= b(1)/a(1);
Q(1)=d(1)/a(1);
P(end)=0;
Q(end)=TT(count,end);

for ii= 2:length(x)-1
    P(ii) = b(ii)/(a(ii)-c(ii)*P(ii-1));
    Q(ii) = (c(ii)*Q(ii-1)+d(ii))/(a(ii)-c(ii)*P(ii-1));
end;

TT(count,end)=c(end)/a(end)*TT(count-1,end-1)+d(end);

ii = length(x)-1;
while ii>1
    TT(count,ii)=P(ii)*TT(count,ii+1)+Q(ii);
    ii=ii-1;
end;

for mm=1:length(x)
    if TT(count,mm)<Ts
        fl(1,mm)=0;
    elseif TT(count,mm)>Tl
        fl(1,mm)=1;
    else
        fl(1,mm)=(TT(count,mm)-Ts)/(Tl-Ts);
    end
    k_eq(1,mm)= fl(1,mm)*kl+(1- fl(1,mm))*ks;
    rho_eq(1,mm)= fl(1,mm)*rhol+(1- fl(1,mm))*rhos;
    cp_eq(1,mm)= fl(1,mm)*cpl+(1- fl(1,mm))*cps;
end;

miu=2.*(T0-Tm)./(A.*alpha);
```

```
    K=sqrt(12.*alpha.*(1-sqrt(1+miu)+miu)./(5+sqrt(1+miu)+miu));
    S=K.*sqrt(Time);
    B=A.*alpha.*(1-sqrt(1+miu))./S;
    C=(B.*S+(T0-Tm))./(S.^2);
    T_ana(count,:)=Tm+B.*(x-S)+C.*((x-S).^2);

%     Sc=-lambda.*rho_eq.*abs(fl-fl_old)./delta_t;
    Sc=-lambda.*rho_eq.*abs(fl-fl_old)./delta_t;
    sc_mem(count,:)=Sc;
    cpcount(count+1,:)=cp_eq;
    kk = kk-1;

end;

%% Generate the plot

% plot(x,TT(2,:),'k-'),hold on, plot(x,TT(iteration/4,:),'r-'),...
% plot(x,TT(iteration/2,:),'g-'), plot(x,TT(end,:),'c-');
% plot(x,T_ana(1,:),'k--'),plot(x,T_ana(iteration/4,:),'r--'),...
% plot(x,T_ana(iteration/2,:),'g--'), plot(x,TT(end,:),'c--');
% legend('1 time step','1/4 time','1/2 time ','total time',...
%         '1 time step Analytical','1/4 time Analytical',...
%         '1/2 time Analytical ','total time Analytical');
% xlabel('postion x [m]');
% ylabel('Temperature [K]');
% title('Temperature vs x with phase change and time step 1 X 100');
T_diff = mean( mean(abs(T_ana - TT)));
T_absdiff=abs(TT-T_ana);

% figure,
% plot(x,TT(1,:),'k-'),hold on, plot(x,TT(2,:),'r-'),...
% plot(x,TT(4,:),'g-'), plot(x,TT(6,:),'c-');
% plot(x,T_ana(1,:),'k--'),plot(x,T_ana(2,:),'r--'),...
% plot(x,T_ana(4,:),'g--'), plot(x,TT(6,:),'c--');
% legend('1 time step','2nd step','4th step ','6th step',...
%         '1 step Analytical','2 step Analytical',...
%         '4th step Analytical ','6th step Analytical');
% xlabel('postion x [m]');
% ylabel('Temperature [K]');
%
%
% figure,

star_x = importdata('star_x5.txt');
% star_T6 :lambda=334 step 1,2,3,5,10,20 lambda=334000,step 1,2
% star_T7 :lambda=334,step 1,2,3,5,10,20
star_T = importdata('star_T6.txt');

plot(x,TT(1,:),'k-'),hold on, plot(x,T_ana(1,:),'r-'),...
plot(star_x,star_T(:,1),'g-');
legend('1 time step','analytical','star CCM');
xlabel('position x (m)');
axis([0 L 270 280]);
ylabel('temperature (initial T =278 K)');
```

```
figure,
plot(x,TT(2,:),'k-'),hold on, plot(x,T_ana(2,:),'r-'),...
plot(star_x,star_T(:,2),'g-');
legend('2 time steps','analytical','star CCM');
xlabel('position x (m)');
axis([0 L 270 280]);
ylabel('temperature (initial T =278 K)');

% figure,
% plot(x,TT(3,:),'k-'),hold on, plot(x,T_ana(3,:),'r-'),...
% plot(star_x,star_T(:,3),'g-');
% legend('3 time steps','analytical','star CCM');
% xlabel('position x (m)');
% ylabel('temperature (initial T =278 K)');
```

Reference

[1] J. Loh and M. Wackernagel, "Living planet report 2004," ed: Gland, Switzerland: WWF, 2016.

[2] P. M. Vitousek, H. A. Mooney, J. Lubchenco, and J. M. Melillo, "Human domination of Earth's ecosystems," *Science,* vol. 277, pp. 494-499, 1997.

[3] R. Frankham, "Genetics and conservation biology," *Comptes Rendus Biologies,* vol. 326, pp. 22-29, 2003.

[4] J. K. Jain and R. J. Paulson, "Oocyte cryopreservation," *Fertility and sterility,* vol. 86, pp. 1037-1046, 2006.

[5] V. Fuentes-Landete, C. Mitterdorfer, P. Handle, G. Ruiz, J. Bernard, A. Bogdan*, et al.*, "Crystalline and amorphous ices," in *Proceedings of the International School of Physics "Enrico Fermi*, 2015, pp. 173-208.

[6] S. Leibo, "Preservation of mammalian cells and embryos by freezing," Oak Ridge National Lab., Tenn.(USA)1976.

[7] M. Skyba, M. Faltus, J. Zámečník, and E. Čellárová, "Thermal analysis of cryopreserved Hypericum perforatum L. shoot tips: Cooling regime dependent dehydration and ice growth," *Thermochimica acta,* vol. 514, pp. 22-27, 2011.

[8] R. Chian, W. Son, J. Huang, S. Cui, W. Buckett, and S. Tan, "High Survival Rates and Pregnancies of Human Oocytes Following Vitrification: Preliminary Report," *Fertility and Sterility,* p. S36, 2005.

[9] M. Kuwayama, G. Vajta, O. Kato, and S. P. Leibo, "Highly efficient vitrification method for cryopreservation of human oocytes," *Reproductive biomedicine online,* vol. 11, pp. 300-308, 2005.

[10] T. K. Yoon, T. J. Kim, S. E. Park, S. W. Hong, J. J. Ko, H. M. Chung*, et al.*, "Live births after vitrification of oocytes in a stimulated in vitro fertilization–embryo transfer program," *Fertility and sterility,* vol. 79, pp. 1323-1326, 2003.

[11] G. Ambrosini, A. Andrisani, E. Porcu, E. Rebellato, A. Revelli, D. Caserta*, et al.*, "Oocytes cryopreservation: state of art," *Reproductive toxicology,* vol. 22, pp. 250-262, 2006.

[12] E. Porcu, R. Fabbri, G. Damiano, S. Giunchi, R. Fratto, P. Ciotti*, et al.*, "Clinical experience and applications of oocyte cryopreservation," *Molecular and cellular endocrinology,* vol. 169, pp. 33-37, 2000.

[13] K. N. Hamilton, S. E. Ashmore, and H. W. Pritchard, "Thermal analysis and cryopreservation of seeds of Australian wild Citrus species (Rutaceae): Citrus australasica, C. inodora and C. garrawayi," *CryoLetters,* vol. 30, pp. 268-279, 2009.

[14] R. Fabbri, E. Porcu, T. Marsella, G. Rocchetta, S. Venturoli, and C. Flamigni, "Human oocyte cryopreservation: new perspectives regarding oocyte survival," *Human reproduction,* vol. 16, pp. 411-416, 2001.

[15] S.-U. Chen, Y.-R. Lien, H.-F. Chen, L.-J. Chang, Y.-Y. Tsai, and Y.-S. Yang, "Observational clinical follow-up of oocyte cryopreservation using a slow-

freezing method with 1, 2-propanediol plus sucrose followed by ICSI," *Human Reproduction,* vol. 20, pp. 1975-1980, 2005.

[16] P. Durand-Smet, N. Chastrette, A. Guiroy, A. Richert, A. Berne-Dedieu, J. Szecsi*, et al.*, "A comparative mechanical analysis of plant and animal cells reveals convergence across kingdoms," *Biophysical journal,* vol. 107, pp. 2237-2244, 2014.

[17] G. M. Cooper and R. E. Hausman, *The cell* vol. 85: Sinauer Associates Sunderland, 2000.

[18] S. Bauerecker, P. Ulbig, V. Buch, L. Vrbka, and P. Jungwirth, "Monitoring ice nucleation in pure and salty water via high-speed imaging and computer simulations," *The Journal of Physical Chemistry C,* vol. 112, pp. 7631-7636, 2008.

[19] F. Norton, "Critical study of the differential thermal method for the identification of the clay minerals," *Journal of the American Ceramic Society,* vol. 22, pp. 54-64, 1939.

[20] J. L. Ford and P. Timmins, *Pharmaceutical thermal analysis: techniques and applications*: Halsted Press, 1989.

[21] T. Ozawa, "Thermal analysis—review and prospect," *Thermochimica Acta,* vol. 355, pp. 35-42, 2000.

[22] T. Hatakeyama and F. Quinn, "Fundamentals and applications to polymer science," *Thermal Analysis,* 1994.

[23] L. M. Compton, J. L. Armes, and G. L. Solbrekken, "Microfabrication of Single-Cell Scale Sample Holder for Scanning Thermal Analysis," in *ASME 2012 International Mechanical Engineering Congress and Exposition*, 2012, pp. 639-648.

[24] M. L. Hosain and R. B. Fdhila, "Literature Review of Accelerated CFD Simulation Methods towards Online Application," *Energy Procedia,* vol. 75, pp. 3307-3314, 2015.

[25] F. Mahmoudian and G. F. Margrave, "A review of the finite-element method in seismic wave modelling," Technical report, CREWES Research Report2003.

[26] Y. Cheung, "Finite Elements in the Solution of," *Engineer,* 1965.

[27] C. C. Ihueze, "Finite Elements in the Solution of Continuum Field Problems," *Journal of Minerals and Materials Characterization and Engineering,* vol. 9, p. 427, 2010.

[28] G. Comini, S. Del Guidice, R. Lewis, and O. Zienkiewicz, "Finite element solution of non‐linear heat conduction problems with special reference to phase change," *International Journal for Numerical Methods in Engineering,* vol. 8, pp. 613-624, 1974.

[29] H. Budhia and F. Kreith, "Heat transfer with melting or freezing in a wedge," *International Journal of Heat and Mass Transfer,* vol. 16, pp. 195-211, 1973.

[30] K. A. Rathjen and L. M. Jiji, "Heat conduction with melting or freezing in a corner," *Journal of Heat Transfer,* vol. 93, pp. 101-109, 1971.

[31] S. V. Patankar and D. B. Spalding, "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows," *International journal of heat and mass transfer,* vol. 15, pp. 1787-1806, 1972.

[32] D. Kuzmin, lecture notes, Introduction to Computational Fluid Dynamics 012600, Technische Universität Dortmund, 2017.

[33] C. W. Lan, C. C. Liu, and C. M. Hsu, "An adaptive finite volume method for incompressible heat flow problems in solidification," *Journal of Computational Physics,* vol. 178, pp. 464-497, 2002.

[34] K. Charn-Jung and M. Kaviany, "A numerical method for phase-change problems with convection and diffusion," *International journal of heat and mass transfer,* vol. 35, pp. 457-467, 1992.

[35] M. Lacroix, "Predictions of natural-convection-dominated phase-change problems by the vorticity-velocity formulation of the Navier-Stokes equations," *Numerical Heat Transfer, Part B Fundamentals,* vol. 22, pp. 79-93, 1992.

[36] M. Liang and C. Lan, "A finite-volume/Newton method for a two-phase heat flow problem using primitive variables and collocated grids," *Journal of Computational Physics,* vol. 127, pp. 330-345, 1996.

[37] C. Beckermann, H.-J. Diepers, I. Steinbach, A. Karma, and X. Tong, "Modeling melt convection in phase-field simulations of solidification," *Journal of Computational Physics,* vol. 154, pp. 468-496, 1999.

[38] H. Udaykumar, R. Mittal, and W. Shyy, "Computation of solid–liquid phase fronts in the sharp interface limit on fixed grids," *Journal of computational physics,* vol. 153, pp. 535-574, 1999.

[39] T. Ye, R. Mittal, H. Udaykumar, and W. Shyy, "An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries," *Journal of computational physics,* vol. 156, pp. 209-240, 1999.

[40] D. J. Lucia, P. S. Beran, and W. A. Silva, "Reduced-order modeling: new approaches for computational physics," *Progress in Aerospace Sciences,* vol. 40, pp. 51-117, 2004.

[41] M. Rousset, "Reduced-order modeling for thermal simulation," STANFORD UNIVERSITY, 2010.

[42] T. A. Brenner, R. L. Fontenot, P. G. Cizmas, T. J. O'Brien, and R. W. Breault, "A reduced-order model for heat transfer in multiphase flow and practical aspects of the proper orthogonal decomposition," *Computers & Chemical Engineering,* vol. 43, pp. 68-80, 2012.

[43] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: theory and application to non-spherical stars," *Monthly notices of the royal astronomical society,* vol. 181, pp. 375-389, 1977.

[44] A. Bøckmann, O. Shipilova, and G. Skeie, "Incompressible SPH for free surface flows," *Computers & Fluids,* vol. 67, pp. 138-151, 2012.

[45] A. Valizadeh, M. Shafieefar, J. Monaghan, and A. Neyshabouri, "Modelling two-phase flows using SPH method," *Journal of Applied Sciences [P],* vol. 8, pp. 3817-3826, 2008.

[46] S. Chen and G. D. Doolen, "Lattice Boltzmann method for fluid flows," *Annual review of fluid mechanics,* vol. 30, pp. 329-364, 1998.

[47] P. Yuan and L. Schaefer, "A thermal lattice Boltzmann two-phase flow model and its application to heat transfer problems—part 1. Theoretical foundation," *Journal of Fluids Engineering,* vol. 128, pp. 142-150, 2006.
[48] S. Muhamad Zuhairi, A. M Rosdzimin, and C. N Azwadi, "Simulation Ofmixed Convective Heat Transfer Using Lattice Boltzmann Method," *International Journal of Automotive and Mechanical Engineering (IJAME),* vol. 2, pp. 130-143, 2010.
[49] M. Geveler, D. Ribbrock, S. Mallach, and D. Göddeke, "A simulation suite for Lattice-Boltzmann based real-time CFD applications exploiting multi-level parallelism on modern multi-and many-core architectures," *Journal of Computational Science,* vol. 2, pp. 113-123, 2011.
[50] M. Taghilou and M. H. Rahimian, "Lattice Boltzmann model for thermal behavior of a droplet on the solid surface," *International Journal of Thermal Sciences,* vol. 86, pp. 1-11, 2014.
[51] D. Heim, "Two solution methods of heat transfer with phase change within whole building dynamic simulation," in *9th International IBPSA Conference, Montréal, Canada*, 2005.
[52] C. Bonacina, G. Comini, A. Fasano, and M. Primicerio, "Numerical solution of phase-change problems," *International Journal of Heat and Mass Transfer,* vol. 16, pp. 1825-1832, 1973.
[53] P. Lamberg, R. Lehtiniemi, and A.-M. Henell, "Numerical and experimental investigation of melting and freezing processes in phase change material storage," *International Journal of Thermal Sciences,* vol. 43, pp. 277-287, 2004.
[54] K. Morgan, "A numerical analysis of freezing and melting with convection," *Computer Methods in Applied Mechanics and Engineering,* vol. 28, pp. 275-284, 1981.
[55] V. Voller, M. Cross, and N. Markatos, "An enthalpy method for convection/diffusion phase change," *International journal for numerical methods in engineering,* vol. 24, pp. 271-284, 1987.
[56] S. Patankar, *Numerical heat transfer and fluid flow*: CRC press, 1980.
[57] N. Ozisik, *Heat Conduction*: Wiley, 1993.
[58] V. Alexiades, *Mathematical modeling of melting and freezing processes*: CRC Press, 1992.
[59] M. N. Özışık, *Boundary value problems of heat conduction*: Courier Corporation, 1989.
[60] T. R. Goodman, "The heat-balance integral and its application to problems involving a change of phase," *Trans. ASME,* vol. 80, pp. 335-342, 1958.
[61] C. J. Young, "Fabrication and Testing of a Tool for Thermodynamic Characterization of Micro-samples," 2016.
[62] R. P. Ron Milo, *CELL BIOLOGY by the NUMBERS*, 2015.
[63] T. L. Bergman, *Introduction to heat transfer*: John Wiley & Sons, 2011.
[64] M. Strub, O. Jabbour, F. Strub, and J. Bédécarrats, "Experimental study and modelling of the crystallization of a water droplet," *International journal of refrigeration,* vol. 26, pp. 59-68, 2003.

[65] T. Li, D. Donadio, and G. Galli, "Ice nucleation at the nanoscale probes no man's land of water," *Nature communications,* vol. 4, p. 1887, 2013.