# An Efficient Computational Model for Solidification of Liquids in Large Partially Filled Tanks

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy
in the Graduate School of The Ohio State University

By

Shashank Terala

Graduate Program in Mechanical Engineering

The Ohio State University

2023

Dissertation Committee

Professor Sandip Mazumder, Advisor

Professor Marcello Canova

Professor Datta Gaitonde

Professor Seung Hyun Kim

# ABSTRACT

A 32.5% water-urea mixture, commercially known as AdBlue®, is stored onboard diesel vehicles as a liquid within storage tanks and is used for exhaust aftertreatment. In cold weather conditions, the mixture may freeze over the span of several hours or days. Since its freezing is accompanied by expansion, it often results in damage of the enclosing tank. In the interest of avoiding such outcomes, the freezing process within these tanks is of interest. Computational modeling can shed light on the exact freezing process and provide insight into mitigation strategies. However, computational modelling of the solidification/melting process in tanks of such "large" size and over such "long" durations is a challenging task. This is partly due to the simultaneous presence of all three phases (solid, liquid and gas). Furthermore, as natural convection plays an important role during the freezing process, it cannot be ignored. Capturing the dynamics of natural convection requires the use of extremely small time-step sizes, in relation to the overall freezing time scales, which significantly affects the computational speed of these simulations. This fact is demonstrated by the first study in this thesis, where the limitations of the capabilities of commercial CFD codes are highlighted when used to solve solidification/melting problems. This led to the main objective of this work: the development, validation, and demonstration of an efficient 3D computational model that can be used to model the solidification process in large, partially-filled tanks containing either water or Adblue®.

This objective is achieved using two newly proposed models: the first is a reduced natural convection model for heat transfer during solidification/melting, and the second is a diffusion-based volume transport model used to account for the expansion of ice during the freezing process.

In the preliminary assessment phase of this study, the in-built models of the commercial CFD solver ANSYS Fluent$^{TM}$ are utilized to study the freezing process in a simple, partially filled 2D tank, which is a quarter of the size of tanks used in the application at hand. Results show that though the models are able to provide great physical details of the solidification process, they result in impractically long simulation run times (~year) due to the requirement of very small time-step sizes. Subsequently, the two new models mentioned earlier were developed.

The first phase of this work developed a new “reduced” model that accounts for the heat transfer due to natural convection during solidification/melting but, ignores the movement of the gas-(solid/liquid) interface due to expansion of ice. This new reduced natural convection model bypasses solving for flow and reduces the energy equation to a pure conduction equation by modeling convective heat fluxes using an equivalent conductive heat flux via an artificial thermal conductivity. The idea is borrowed from turbulence modeling wherein turbulent transport is modeled using an eddy diffusivity. The physical laws governing natural convection are used to obtain an appropriate expression for this artificial thermal conductivity. Additionally, the latent heat release due to solidification is represented in this model using a volumetric source term. The reduced model generates an unknown material constant that is ultimately calibrated by matching

temperature vs. time data collected during freezing experiments of a partially-filled tank. This experimental data was collected at the Ford Motor Company. In the experimental study, additional experiments were carried out using combinations of three different fill levels (25%, 50% and 80%) and two working liquids (water and AdBlue®) to provide data for validation of the models. Validation studies were then performed and showed good agreement with measured temperature data, while also providing significant improvement in simulation run times: reduction from ~year to a few days. To test the capabilities of the reduced model when applied to a general natural convection problem, a study involving heat transfer in a differentially heated cavity was also undertaken. Results for three Rayleigh numbers comparing predictions to those from high-fidelity calculations show good agreement.

The second phase of the work involved accounting for the expansion of ice and its coupling to thermal transport and phase change. As flow is not calculated as part of the reduced natural convection model, conventional methods for tracking phase boundaries, such as the volume-of-fluid (VOF), are incompatible with this model and cannot be used to track the expansion of ice and the resulting movement of the gas-(solid/liquid) interface. Therefore, a new model, which is a diffusion-based form of the VOF method and solves a new conservation equation derived from volume (mass) conservation, is instead proposed. As this model introduces a new governing equation outside the suite of equations normally solved by ANSYS Fluent$^{TM}$, a parallel, unstructured conjugate gradient squared solver with Jacobi pre-conditioning, written within Fluent's User-Defined Function (UDF) framework, is developed from ground up. The measured temperature data previously used for validation of the reduced model is used once again for validation of the model, but now with the inclusion of ice expansion. It is found that the implementation of the formation and rise of the ice dome primarily improves agreement with experimental data at locations closer to the surface of the solid/liquid. The ice dome itself is also seen to clearly rise above the initial liquid surface. As a test of the ability of the models to handle more complex geometries, a simulation of the freezing process within a production DEF (Diesel Exhaust Fluid) tank used by the Ford Motor Company was also conducted successfully.

Key contributions of the work covered in this thesis include the development of two new computational models. The first model is a model to account for the heat transfer due to natural convection during freezing of water in large tanks but without solving for flow. The second is a model to account for the expansion of ice and its effects during the freezing process. The models were integrated into ANSYS Fluent$^{\text{TM}}$ using UDFs making them completely general-purpose and ready for commercial use.

# DEDICATION

To my family and my friends, thank you for everything.

# ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Sandip Mazumder, for his guidance and support over the course of my doctoral studies. I greatly appreciate the patience and trust he has had in me during this time. I am grateful to have had the opportunity to work under his tutelage. He has played an invaluable role in helping me grow both as a researcher and as a person.

I would like to express my gratitude to Prof. Marcello Canova, Prof. Datta Gaitonde and Prof. Seung Hyun Kim for taking the time and effort to serve on my doctoral committee.

I am thankful to the Ford Motor Company for sponsoring this work. Specifically, I would like to extend my gratitude to the Ford engineers, both past and current, Dr. Syed Ali, Dhaval Vaishnav, Mohsen Ehteshami, Joseph Demaria for the invaluable inputs and support over the course of this work. I would like to thank Eric Myers in particular, for assisting with running some of the large-scale CFD/CHT simulations in this work at the computational facilities provided by the Ford High Performance Computing (HPC) facility.

I would like to thank the Ohio Supercomputer Center (OSC) for providing the computational resources required for the large-scale CFD/CHT simulations in this work. I would also like to thank the Department of Mechanical and Aerospace Engineering at The

Ohio State University for providing additional computational resources alongside licenses for ANSYS' mesh generation package, ICEM-CFD$^{TM}$, and solver, FLUENT$^{TM}$.

Lastly, I am grateful to my family and friends for their constant support and encouragement. I would like to thank my parents for all their efforts and sacrifices to provide me and my brothers with every opportunity for success and happiness. To my brothers, I will forever be proud to be your older brother. Thank you to my lab mates, Vishal Ramesh, Navni Verma, Siddharth Saurav and Nehal Jajal for their support and companionship. Finally, thank you to Shruthanth, Rajan, Karthik, Rishi, Nikilesh, Vinay and Joel for their friendship and the shared experiences over the years.

# VITA

| | |
|---|---|
| July 2015 | B.Tech & M.Tech in Aerospace Engineering<br>Indian Institute of Technology Madras, Chennai |
| January 2017 to May 2019 | Graduate Teaching Associate,<br>Department of Mechanical and Aerospace Engineering,<br>The Ohio State university |
| August 2019 to present | Graduate Research Associate,<br>Fluids and Thermal Analysis Lab<br>The Ohio State University |

## Journal Publications

Terala, S., Mazumder, S., Matharu, G., Vaishnav, D., and Ali, S. (2022). "Efficient Simulation of Freezing of Water in Large Tanks Including Expansion of Ice." ASME. *J. Thermal Sci. Eng. Appl*. November 2022; 14(11): 111006.

Ramesh, V., Terala, S., Mazumder, S., Matharu, G., Vaishnav, D., and Ali, S. (2020). "Development and Validation of a Model for Efficient Simulation of Freezing of Water in Large Tanks." ASME. *J. Thermal Sci. Eng. Appl*. February 2021; 13(1): 011008.

## Conference Publications

Terala, S., Mazumder, S., Matharu, G., Vaishnav, D., & Ali, S. (2022, October). A Reduced Three-Phase Model for Solidification of Liquid in Large Tanks. In *ASME International Mechanical Engineering Congress and Exposition* (Vol. 86700, p. V008T11A053). American Society of Mechanical Engineers..

Terala, S., Mazumder, S., Matharu, G., Vaishnav, D., Ehteshami, M., & Ali, S. (2021). An Efficient Computational Model for Solidification of Water in Large Tanks. In *Proceedings of the 26thNational and 4th International ISHMT-ASTFE Heat and Mass Transfer Conference December 17-20, 2021, IIT Madras, Chennai-600036, Tamil Nadu, India*. Begel House Inc..

## Fields of Study

Major Field: Mechanical Engineering

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

# CHAPTER 1. INTRODUCTION

## 1.1 Background

Selective catalytic reduction is a standardized, efficient technology used in diesel vehicles to reduce nitric oxide (NOx) emissions [1]. In this technology, a 32.5% urea solution in water serves as a reducing agent for the catalytic reduction of NOx. This liquid solution, called Diesel Exhaust Fluid (DEF) or AdBlue®, is generally stored in a plastic tank onboard the vehicle. AdBlue® is directly injected from these tanks into the exhaust gas stream. Subsequently, the ammonia formed from urea reacts with the NOx within the SCR catalytic converter to form nitrogen and water. This process claims to reduce NOx emissions by 85% [2]. Figure 1.1 shows a schematic of a typical SCR system with the corresponding AdBlue® injection system.

AdBlue® has a freezing point of -11°C (12°F) [6]. In parts of the world with colder climates, particularly those in northern Europe and North America, ambient temperatures below -11°C are a regular occurrence. When this happens, the AdBlue® present in the tank freezes and, similar to water, expands. The expansion leads to a rise in pressure within the tank which can damage both the internal components as well as the tank itself. Additionally, the mixing and reaction of the AdBlue® spray with the exhaust gases plays a crucial role in the performance of the SCR system [3],[5]. Therefore, sufficient liquid

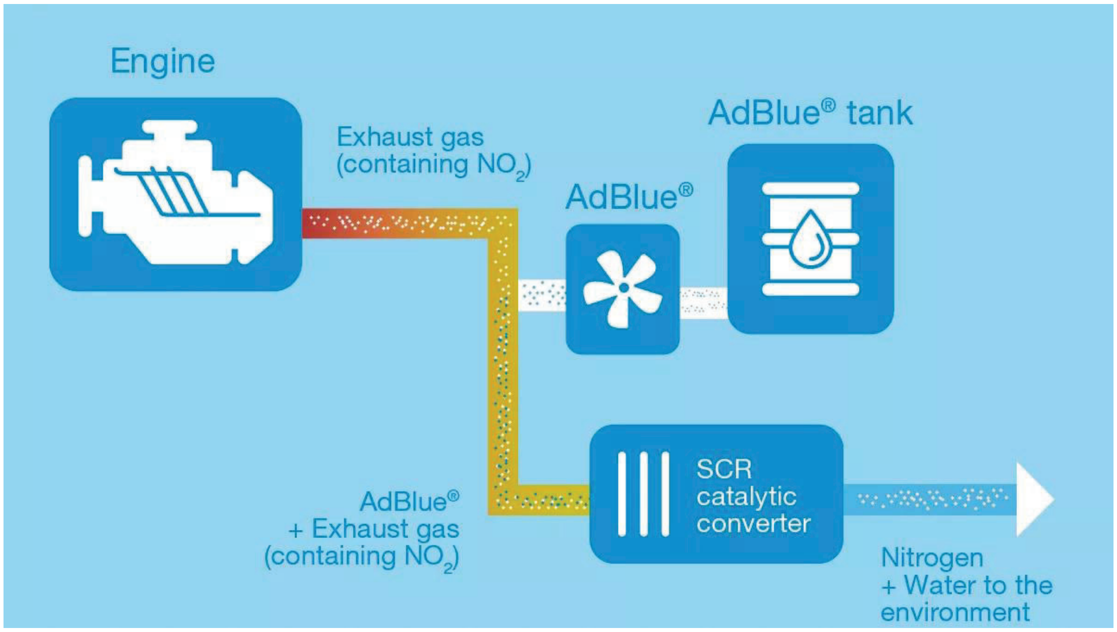
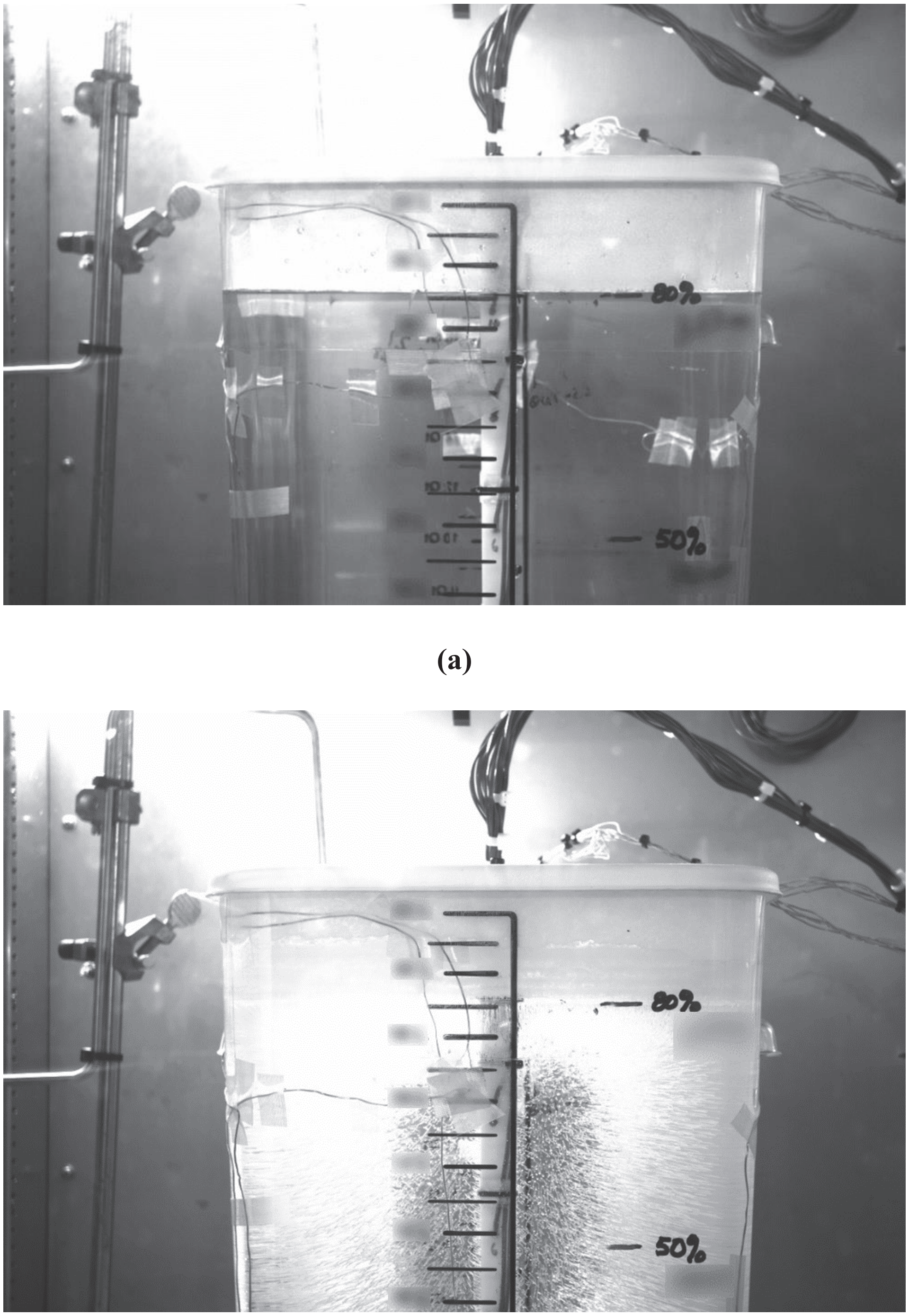

Engine
Exhaust gas (containing $NO_2$)
AdBlue® tank
AdBlue®
SCR catalytic converter
AdBlue® + Exhaust gas (containing $NO_2$)
Nitrogen + Water to the environment

**Figure 1.1: Schematic of SCR system with AdBlue® injection system[3].**

AdBlue® is required within the tank to be able to inject a spray of the liquid solution into the exhaust stream during operation [7]. This had led to considerable interest in developing efficient heater designs and mitigation strategies. For example, in an experimental study performed by Choi *et al.* [8], a heating system that could prepare a sufficient amount of liquid AdBlue® solution from frozen AdBlue® within a urea-SCR system was investigated. The heating system employed an engine coolant heat exchanger and an electric heater simultaneously to melt the frozen AdBlue®. Parameters such as the coolant flow rate and temperature, shape of the heat exchanger, and power of the electric heater were varied to study their effect on melt rate. In another study, Choi *et al.* [9] conducted a numerical investigation to identify the best shape of a heating pipe to melt frozen AdBlue® within a storage tank. The study considered four different heating pipe shapes and evaluated which design resulted in the maximum amount of liquid solution produced after 1000s of heating. Beeck *et al.* [10] designed a new heating technology using a resistive wire distributed in a tank's volume. This technology allowed maximum heating system coverage of the tank's volume irrespective of tank geometry. The effectiveness of this system was subsequently demonstrated by experimentally testing it during winter conditions under various driving scenarios.

As evident from the above discussion, designing heating systems for proper storage of AdBlue® and functioning of SCR systems has been of significant interest. In the design process of such systems, numerical modeling is an oft-leveraged tool. This is because experimental measurement of the solidification front propagation is not always a trivial task. For example, if the tank used for experimentation purposes is a closed tank or is one without an easily removable lid, it is not possible to take direct measurements of the final ice block. Alternate techniques such as taking photographs of the solidification front over the course of a freezing experiment are only possible if the tank walls are transparent. Even in this were the case, further problems are encountered. In an experimental study conducted by engineers at the Ford Motor Company, the sponsors of the project the current work is part of, excessive frosting was observed to form on the walls of the tank during the freezing process, as shown in Fig. 1.2. Figure 1.2(a) shows a picture of a transparent tank filled with water at the start of a freezing experiment. In this picture, while there is some moisture on the walls of the tank, the air-water interface is nonetheless clearly visible. However, at the end of freezing, the picture Fig. 1.2(b) shows that the moisture seen earlier has frozen into a layer of frosting. This frosting completely occludes the air-ice interface within the tank, thereby preventing measurement of any useful information about the location of the ice-air interface from being obtained from the photographs. Finally, the liquid-solid phase interface is almost always trapped deep inside the tank and cannot be observed using any known experimental technique. These experimental challenges also restrict the validation studies of numerical models. While significant studies have been performed, modeling the solidification/melting of substances and the scenarios faced by AdBlue® storage tanks pose some unique challenges, as explained in the following sections.



(a)

(b)

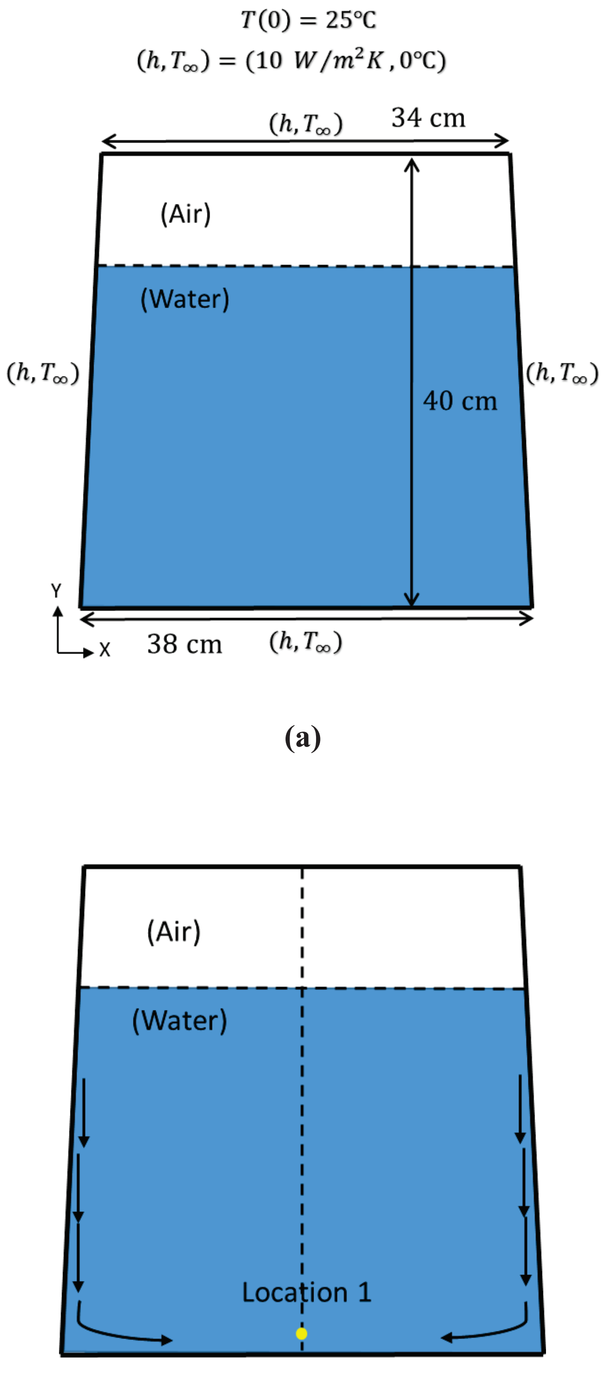
**Figure 1.2: Pictures of a transparent tank filled with water at (a) the start of freezing (b) the end of freezing.**

## 1.2 An Overview of the Freezing Process in Partially Filled Tanks

Prior to discussing the numerical modeling of the solidification/melting process, an understanding of the various aspects relevant to the freezing process would help better appreciate the requirements and challenges posed in the modeling process. While the primary focus is the freezing of AdBlue®, a useful preliminary step is to study the freezing process of pure water, as the thermodynamic and solidification processes of freezing water are much better documented, while being very similar to that of Adblue. It, therefore, provides a solid foundation to develop the necessary physical models and address any numerical issues that arise during such development. Most of the conclusions drawn for pure water will be directly applicable to AdBlue® as well. Interestingly, liquid AdBlue® does not show any density inversion at any temperature unlike water at 4°C, which makes its modeling slightly easier than pure water.

As shown in the studies by Wiesche *et al.* [6] and Choi *et al.* [8], a typical cooling scenario of a tank containing a fluid involves cooling the tank from an initial condition where the entirety of the liquid is at a temperature above the freezing point of the liquid. The cooling then proceeds until the entire liquid within the tank solidifies and the solid reaches a steady-state temperature equal to an ambient temperature below the freezing point of the liquid. This freezing process can be divided into three regimes; a pre-solidification regime prior to the onset of solidification, a solidification regime where both liquid and solid states of the fluid exist simultaneously, and finally a post-solidification regime where the entirety of the liquid has solidified. Different physics dominate during each of these regimes leading to differences in complexity.

During the pre-solidification regime, heat transfer occurs primarily through conduction and buoyancy-driven natural convection. The heat transfer due to natural convection has a significant effect on the temperature distribution within the tank and cannot be neglected during modeling even though density change of liquid water with temperature is not nearly as dramatic as in the gas phase. This is because the length scales (tank sizes) in such problems are large. To illustrate this, Ramesh [11] performed a study where water is cooled in a simple hypothetical 2D tank (refer Figure 3(a)). When cooling begins, the water adjacent to the side walls begins to cool down causing its density to increase relative to the water further interior of the tank (density of water increases as it cools down to the density inversion point at 4°C). This causes the water adjacent to the walls to descend and leads to the formation of buoyancy driven fluid flow, which constitutes natural convection. To evaluate the effect of natural convection on the heat transfer rate in such a system, the temperature at a point near the bottom of the tank (Location 1 as seen in Fig. 1.3(b)) is monitored. Figure 1.4 shows the temperature versus time data at this location with and without natural convection for this system. From the figure, it is evident that the presence of natural convection in water dramatically affects the temperature profile as opposed to the presence of conduction alone. In this situation, the natural convection driven flow enhances heat transfer at Location 1.



(a)

(b)

**Figure 1.3: Schematic of a hypothetical water tank in two dimensions: (a) geometry and boundary conditions, (b) natural convection pattern from the side walls.**

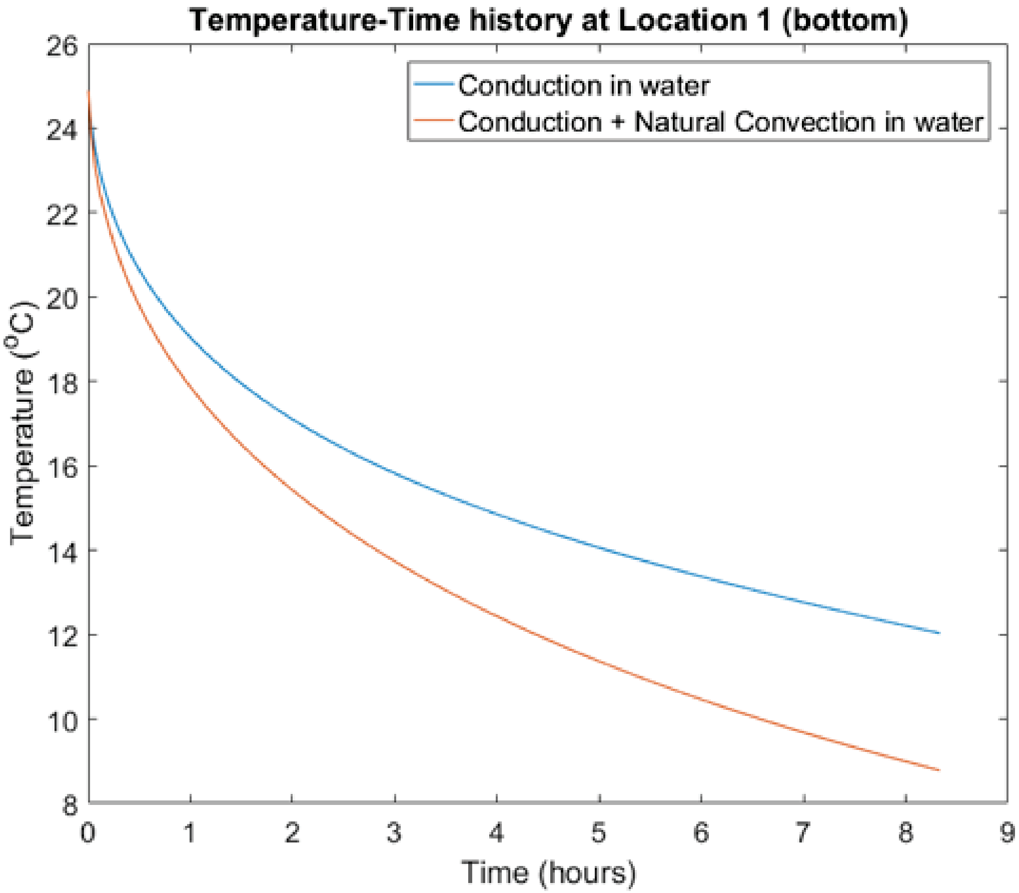


**Figure 1.4: Temperature versus time data at Location 1 with and without natural convection, from Ramesh [11].**

The solidification regime consists of the most complex physics of the three regimes. The phase change process from liquid to solid creates a solid-liquid interface. Many physical properties and quantities, such as density and thermal conductivity, abruptly change across this interface due to the phase change. Additionally, the phase change process from liquid to solid results in the release of latent heat due to solidification at the solid-liquid interface. Therefore, predicting and tracking the location of this interface is an important aspect when modeling the heat transfer within the tank.

Both water and AdBlue® show a significant decrease in density (~10%) when freezing from liquid to solid. This creates a corresponding increase in volume as per conservation of mass. In partially filled tanks, this increase in volume manifests itself as the movement of the gas-(solid/liquid) interface at the free surface of the solid/liquid volume. Figure 1.5 shows a sketch depicting this phenomenon. As the expanding ice can come into contact with any tank components that were previously above the free surface, it is often of interest to track the movement of this additional interface. Additionally, as reported by Akyurt *et al.* [55], extremely high pressures (~4000 psi or 270 atm) can be generated in volumes of water confined by ice. Therefore, tracking the location of the shrinking liquid bubble is of importance in order to avoid placing any delicate tank components at this location. Consequently, study of the solidification regime requires the capabilities to track two distinct interfaces, a solid-liquid interface within the volume of the fluid and a gas-(solid/liquid) interface at the free surface of the fluid volume.

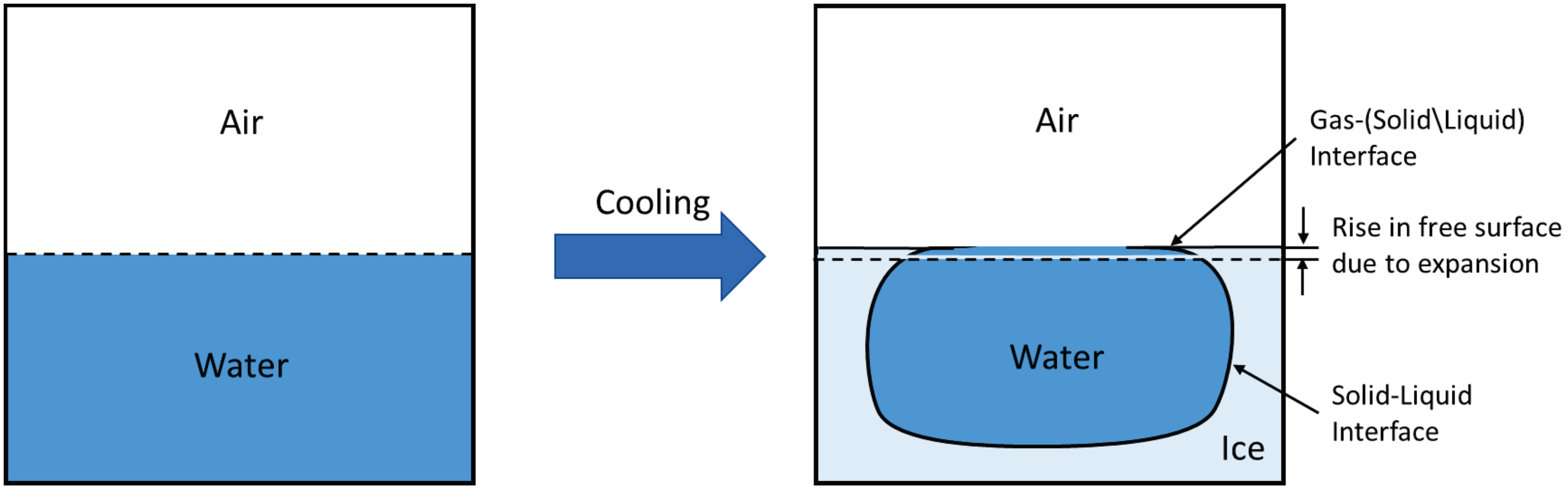


Figure 1.5: Sketch depicting presence of two distinct interfaces and rise of free surface during freezing.

The post-solidification regime is the simplest of the three regimes from a physical standpoint. As there is no remaining liquid, natural convection cannot take place and heat transfer occurs solely via conduction within the solid. Additionally, the lack of a solid-liquid interface only the gas-solid interface to be tracked.

### 1.3 Modeling the Freezing Process in Partially Filled Tanks

Modeling of the general solidification/melting process has been an area of active interest since the late 20th century, particularly in the context of metallurgical processes. In one such investigation, Minaie *et al.* [12] analyzed the flow patterns and associated solidification phenomena during the die casting of a square cavity. An important feature in this work was the inclusion of the residual flow field in the solidification analysis, as opposed to assuming an initially motionless molten metal in a cavity. This residual flow field within the molten metal is a consequence of the filling stage, where the flow field created filling does not immediately disappear once the cavity is filled. In another analysis, Im *et al.* [13] numerically analyzed the simultaneous filling and solidification of a square cavity. For this study, both natural convection flow and residual flow were considered to investigate the coupled effects of filling and natural convection on solidification during casting. Other numerical studies of solidification/melting have been performed in the context of energy storage applications. For example, applications involving latent heat storage require the prediction of the overall freezing/melting rate. These studies highlight the importance of modeling solidification/melting in the thermal performance and design optimization of such systems. In one study, Tay *et al.* [14] developed and validated a computational fluid dynamics (CFD) model for tubes filled with water in a phase change thermal energy storage system. Elsewhere, Trp [15] performed a numerical investigation of the heat transfer during the melting/solidification of paraffin in a shell-and-tube latent thermal energy storage unit. Al-abidi *et al.* [16] surveyed studies which used commercial CFD software and self-developed programs to simulate the heat transfer in phase change materials (PCMs) in various applications including electronic cooling technology, building thermal storage and HVAC.

Modeling of the freezing process in partially filled tanks requires tracking of the different phase boundaries present in the system and incorporating the effect they have on the heat transfer process. These moving phase boundary problems are called Stefan problems, named after J. Stefan who first introduced these problems in 1889 when studying the formation of ice in the polar seas [17],[18]. Stefan problems are highly non-linear and analytical solutions are known only for simple geometries. For example, Hoffman [19] provides the exact solution to the classical one-phase Stefan problem – a one-dimensional melting problem where the temperature is assumed to be varying in only one spatial dimension. A wide range of numerical methods applied to Stefan problems have been reported in Crank's book on moving boundary problems [20]. Elsewhere, Javierre *et al.* [21] provide a comparative study of the numerical models for one-dimensional Stefan problems. However, exact solutions for problems involving more complex geometries and multiple dimensions appear to be unobtainable. Therefore, numerical approaches towards such problems are much more common.

Numerical models to solve moving boundary problems typically fall under two classes: (i) fixed-grid formulations (ii) variable-grid formulations. Variable-grid (or moving grid) formulations solve two different sets of governing equations for each phase and move the phase change interface based on these equations. The grid is then recalculated in each phase. As these methods typically result in sharply defined interfaces and accurately calculate properties around the interface, there has been some development of models of this category. For example, Jana *et al.* [22] developed a grid-adaptive numerical method and demonstrated its effectiveness using benchmark solidification problems available in the literature. Elsewhere, Zhang et al. [23] developed a combined multizone adaptive grid-generation technique with a curvilinear finite volume scheme, to model solidification in a rectangular enclosure. However, as moving grid methods rely on complex and computationally expensive adaptive grid generation, simpler fixed-grid methods are instead the preferred choice when modeling solidification/melting problems.

A special class of the fixed grid approaches is based on the “Enthalpy method” first proposed by Voller *et al.* [24]. The enthalpy method was originally developed for problems where the material underwent phase change over a narrow range of temperatures. As a result, this method found widespread use in several industrial applications, particularly involving alloys and semitransparent materials. For example, Lapka et al. [25] utilized the enthalpy method to numerically investigate the solidification process of a semitransparent medium in a square cavity. However, as demonstrated by Voller et al. [26], the enthalpy method produces non-physical features when the melting/solidification temperature is sharply defined. Consequently, Voller et al. [27] proposed an extension to the conventional enthalpy method that eliminated this problem. This modified enthalpy method is general and can handle situations where the phase change occurs at a distinct temperature or over a temperature range. This method, termed the “Enthalpy-porosity method”, is one of the most widely used techniques for modeling solidification/melting problems today.

Unlike the solid-liquid interface, the gas-(solid/liquid) interface does not necessarily move as a direct consequence of the phase change process. Instead, it typically moves either due to an external flow being enforced (such as in pouring and filling problems) or expansion/contraction occurring due to volume (density) changes. As these are mechanical processes as opposed to thermal processes, the enthalpy-porosity method is unsuitable to track the movement of this interface and other fixed-grid methods such as the “Volume of Fluid” (VOF) method [28], the “Level Set” method [29] or the “Phase Field” method [30]-[32] are instead employed. In both the VOF and the Level Set method, a function is defined to identify the surface. Mass and energy jump conditions are implemented at the interface and the entire computational domain, including both phases, is solved simultaneously. In contrast, the Phase Field method does not explicitly track the interface and instead obtains a phase field equation from thermodynamic considerations. While the current work borrows ideas from the VOF method, both the Level Set and Phase Field methods have been utilized in various studies. For example, Tan *et al.* [33] modeled microstructure evolution in the solidification of multi-component alloys using a level set method. Elsewhere, Boettinger *et al.* [34] provide an overview of the phase field method and demonstrate its application to various problems involving dendritic growth in pure materials; dendritic, eutectic, and peritectic growth in alloys; and solute trapping during rapid solidification.

The VOF method is chosen for the current work due to its prolific use for problems that track free surfaces with a fixed grid. The VOF method is also compatible with the enthalpy-porosity method and often coupled with the enthalpy-porosity method to tackle problems involving multiple phase interfaces, such as during simultaneous filling and solidification of cavities. In the vast majority of computations reported for solidification/melting problems, the time-step sizes used are small, as dictated by the Courant-Friedrichs-Lewy (CFL) criterion [35], which states that the numerical solution may become unstable if the phase boundary crosses more than a grid cell during a single time-step. The VOF method is inherently an explicit time-marching method and the CFL criterion must be obeyed in this method to keep solutions stable. In an early study, Yeoh et al. [36] investigated the effects of natural convection during freezing of water in a three-dimensional (3D) cavity using a finite-volume formulation on a body-fitted mesh [37]. The non-dimensional time-step size used was $5 \times 10^{-5}$, while the physical size of the cavity was only 32 mm. More recently, Bourdillon et al. [38] developed an enthalpy-porosity based solidification model to study the freezing phenomenon in two geometries: a two-dimensional (2D) square cavity and a 2D cylindrical enclosure. A time-step size of 0.05 seconds was used to ensure stable solutions. The physical time scales (total solidification time) for the two cases were only 100 seconds and 5000 seconds for the square cavity and cylindrical enclosure, respectively. The maximum size (global length scale) in the two test cases was only about 83 mm. In another recent study, Sharma et al. [39] conducted a numerical study of the solidification of copper-water nanofluid and pure water in a 2D trapezoidal cavity. In this study, a time-step size of 0.5 seconds was used. The time and length scales for the problem were also fairly small—approximately 3000 seconds and 10 mm, respectively.

Due to the constraint on small time-step sizes, computational results for solidification/melting problems utilizing the enthalpy-porosity method have only been reported either for simplified 2D geometries and/or for problems where the overall time/length scales are small enough to complete the computations within a reasonable timeframe. Consequently, solidification/melting simulations involving tank sizes of interest for automotive applications are extremely challenging when conventional methodologies in CFD are used and have witnessed little to no progress. The primary reason for this extreme challenge is that the total duration for freezing in such "large" tanks can be almost a day, as seen in experiments conducted by aus der Wiesche [6]. Similar experiments were conducted by Choi et al. [8], where the researchers performed a freezing experiment of AdBlue® inside a tank filled to 80% of its volume. Figure 1.6 shows the resulting temperature vs. time data at various thermocouple locations within the tank. It can be seen from this data that the time taken for the tank to reach an ambient temperature of -30°C was approximately 72 hours (3 days), which is almost two orders of magnitude longer than the solidification times in the computational studies conducted by Bourdillon et al. [38] and Sharma et al. [39]. Furthermore, the maximum length scale for the AdBlue® tank was approximately 330 mm, which is significantly larger than the length scales involved in the aforementioned modeling studies.

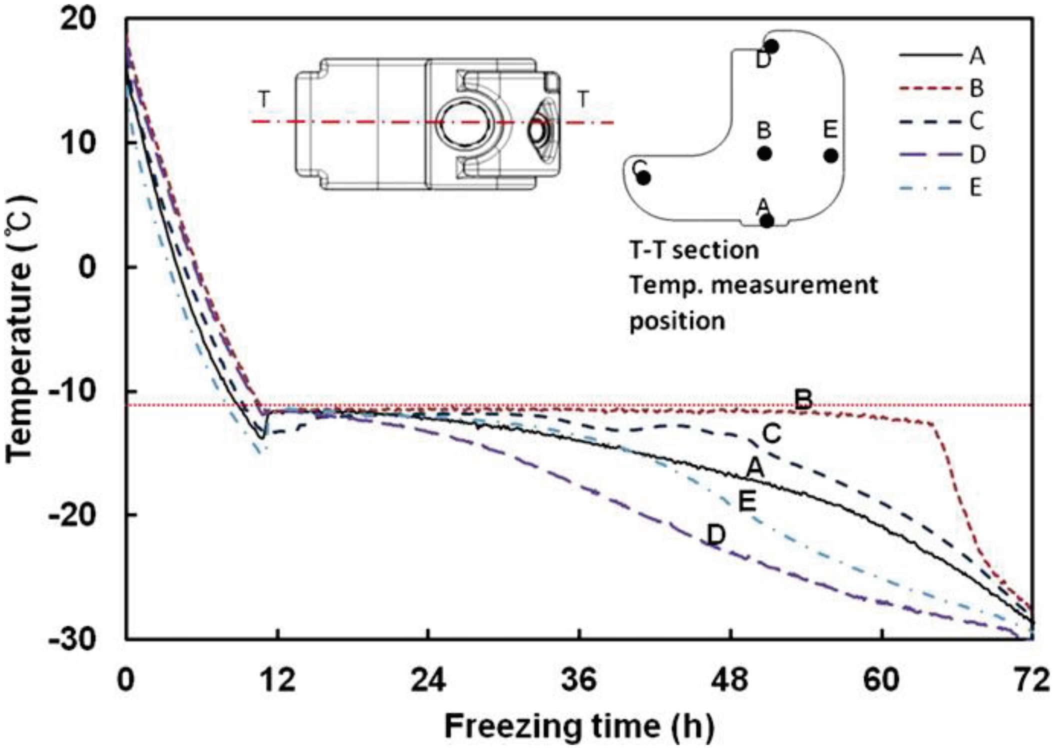


**Figure 1.6: Temperature versus time data at various thermocouple locations during the experimental freezing process of an AdBlue® tank carried out by Choi et al. [4].**

### 1.4 Dissertation Scope and Objectives

As described in Section 1.3, the different regimes of the freezing process require the modeling of two different phase interfaces. The studies discussed in the preceding section highlight the fact that the use of conventional CFD methods for tackling such problems limits them to either simple geometries and one or two spatial dimensions, or to problems where the overall time scales of the solidification process are shorter than a couple of hours. Although accurate numerical models for solidification exist in the literature, none of them have been used to thoroughly investigate the freezing of AdBlue® tanks for actual SCR applications. Moreover, owing to the small time-step sizes needed to model solidification (as low as a few milliseconds), performing a CFD analysis of the freezing process in tanks with physical time scales of several days appears to be prohibitively expensive. In fact, this constitutes the first objective of this dissertation - to identify the limitations, if any, of modeling the solidification process in such AdBlue® tanks for SCR applications. The ultimate goal of this work is to develop a numerical solution strategy for the efficient modeling of the solidification process (including the expansion of ice).

The specific objectives of this dissertation are as follows:

1. To perform a systematic study to investigate the capabilities of the in-built physical models of ANSYS Fluent™ pertaining to freezing/thawing and to establish the limits of feasibility.
2. To develop an efficient heat transfer model to simulate the freezing process of water/AdBlue® in tanks for SCR applications. Typically, such processes require a total simulation time of ~10 hours.

3. To develop an efficient interface-tracking model (compatible with the prior reduced heat transfer model) to simulate the expansion of ice during the freezing process and couple it to the afore-mentioned heat transfer model.
4. To perform validation studies of the proposed model by comparing numerical results with experimental observations for freezing of both pure water and AdBlue®.
5. To demonstrate the proposed model for simulation of freezing of water/Adblue® in production-sized DEF tanks.

### 1.5 Dissertation Outline

This dissertation is organized in the following manner: the following chapter, i.e., Chapter 2 investigates the existing modeling capabilities offered by the commercial CFD software ANSYS Fluent$^{TM}$, with particular focus on computational efficiency and scalability. Consequently, Chapter 3 describes a new heat transfer model for efficient simulation of such processes and outlines the relevant mathematical equations. Chapter 4 covers some preliminary results generated from implementing this model in ANSYS Fluent$^{TM}$ using User-Defined Functions (UDFs). Here, ANSYS Fluent$^{TM}$ is only used to solve the energy equation, while providing a platform to implement the proposed model without sacrificing the quality-of-life features offered by commercial CFD codes, such as the ability to handle complex geometries and meshes, GUI support, streamlined post-processing, etc. Chapter 5 covers a volume transport model developed to predict the expansion of ice and the subsequent movement of the gas-(solid/liquid) interface during the solidification process. It discusses the formulation of the relevant governing equations and their implementation.

Chapter 6 then covers results generated for various scenarios from implementing this volume transport model in conjunction with the reduced heat transfer model from Chapter 3. Chapter 7 summarizes the work conducted as part of this dissertation and recommends some future work.

# CHAPTER 2. SOLVING MELTING/SOLIDIFICATION PROBLEMS WITH COMMERCIAL SOFTWARE

Modelling of the freezing process of partially filled tanks requires tackling a three-phase problem where gas, liquid and solid phases co-exist simultaneously. As covered in Chapter 1, this involves solving two moving boundary problems: (1) the solidification (phase change) boundary and (2) the free surface boundary. Commercial CFD software often provide inbuilt models to tackle such problems. In the current work, ANSYS Fluent$^{\text{TM}}$ was chosen as the CFD software as it provided modelling capabilities for both moving boundary problems. For the three-phase problem at hand, it uses the enthalpy-porosity method as the solidification/melting model. Consequently, it offers only the volume-of-fluid method as a compatible free surface tracking model. Each method introduces a new scalar to describe the corresponding interface; (i) the solid volume fraction, $s_v$, to describe the solidification front and (ii) the gas volume fraction, $f_g$, to describe the free surface. Each scalar represents the ratio between the volume of a particular phase with respect to the total volume of the containing cell, respectively. Specifically,

$$s_v = \frac{V_s}{V} \tag{2.1}$$

where $V_s$ is the volume of the solid in a particular cell and $V$ is the total volume of that cell [63]. Similarly,

$$f_g = \frac{V_g}{V} \tag{2.2}$$

where $V_g$ is the volume of the gas phase in a particular cell. These scalars are two additional unknowns which are to be computed. A brief description of these two models and their implementation in ANSYS Fluent$^{\mathrm{TM}}$ is provided in the subsequent sections, followed by an analysis into their viability and limitations when tackling freezing of large partially filled tanks.

## 2.1 The Governing Equations

Before discussing the aforementioned methods, it is useful to consider the governing equations for the conservation of mass, momentum and energy in a three-phase system to better understand the modifications these methods make. These conservation equations are respectively given by [27],[28],[40],[41],

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \tag{2.3}$$

$$\frac{\partial}{\partial t}(\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U}\mathbf{U}) = -\nabla p + \nabla \cdot (\mu \nabla \mathbf{U}) + \rho \mathbf{g} + S_p \tag{2.4}$$

$$\frac{\partial}{\partial t}(\rho h_{tot}) + \nabla \cdot (\rho \mathbf{U} \mathrm{h}_{tot}) = \nabla \cdot (k \nabla T) + S_{\boldsymbol{e}} \tag{2.5}$$

where $\mathbf{U}$ is the phase-averaged velocity vector, $p$ is the pressure, $T$ and $h_{tot}$ denote the phase-averaged temperature and total enthalpy, respectively. $S_p$ is a momentum sink term used to account for the effect of the mushy zone on the fluid flow. $S_e$ is the energy source term used to account for latent heat transfer due to phase change. The phase averaged velocity is expressed in terms of the cell velocity as,

$$\mathbf{U} = \varepsilon \mathbf{U}_{\text{cell}} \tag{2.6}$$

where $\varepsilon$ is the porosity, defined as the volume of pores (fluid flow passages) to the total volume. In the enthalpy-porosity method, the porosity is determined from the solid fraction as follows,

$$\varepsilon = 1 - s_v \tag{2.7}$$

In pure air regions, the porosity is set to 1. $\rho$, $\mu$ and $k$ denote the volume averaged density, dynamic viscosity, and thermal conductivity respectively. These properties are calculated using the volume fraction $f_g$ and porosity $\varepsilon$ as follows [63],

$$\rho = f_g\rho_g + (\varepsilon - f_g)\rho_l + (1 - \varepsilon)\rho_s \tag{2.8}$$

$$\mu = f_g\mu_g + (\varepsilon - f_g)\mu_l \tag{2.9}$$

$$k = f_g k_g + (\varepsilon - f_g)k_l + (1 - \varepsilon)k_s \tag{2.10}$$

where the subscripts $g$, $l$ and $s$ denote the gas, liquid and solid phases respectively. The VOF method also necessitates use of an advection equation governing the volume fraction $f_g$. This is given by

$$\frac{\partial}{\partial t}(f_g\rho_g) + \nabla \cdot (f_g\rho_g\mathbf{U}) = 0 \tag{2.11}$$

In summary, Eq. (2.3), (2.4), (2.5) and (2.11) represent the governing equations that are to be solved simultaneously, while Eq. (2.8), (2.9) and (2.10) list the constitutive relations for density, viscosity, and thermal conductivity in a three-phase system. The solid fraction, $s$, is calculated from temperature as proposed by Voller *et al.* [27] and will be discussed in the following section.

## 2.2 The Enthalpy-Porosity Method

In the literature survey covered in Chapter 1, several numerical techniques for modeling solidification/melting problems were summarized, from simplified fixed grid methods to more complex moving mesh formulations. Of these methods, the enthalpy-porosity method is a commonly used fixed grid method and is ideal in scenarios where the material undergoes phase change either at a distinct temperature (a pure substance) or over a temperature range (an alloy). Like many other solidification/melting models, it introduces an unknown scalar in the calculation: the solid fraction, $s$. The solid fraction, $s$, is related to the solid volume fraction, $s_v$, as follows [63]

$$\rho_{ls}s = \rho_s s_v \tag{2.12}$$

where

$$\rho_{ls} = \rho_s s_v + \rho_l(1 - s_v) \tag{2.13}$$

The solid fraction, $s$, is meant to represent the fraction of the cell's mass that has solidified and is, therefore, a function of temperature. Therefore, Voller *et al.* [27] proposed the following relation between solid fraction and temperature.

$$s(T) = \begin{cases} 0; & T \geq T_l \\ \dfrac{T - T_l}{T_s - T_l}; & T_s < T < T_l \\ 1; & T \leq T_l \end{cases} \tag{2.14}$$

where $T_l$ and $T_s$ are the liquidus and solidus temperatures of the material. The liquidus temperature is defined as the temperature above which the material is a pure liquid and conversely the solidus temperature is defined as the temperature below which the material is a pure solid. Eq. (2.14) is valid for materials which undergo phase change over the temperature range $[T_s, T_l]$. For a pure substance (like water), where the solidus and liquidus temperatures are equal (to the melting temperature, $T_m$), Eq. (2.14) reduces to the form,

$$s(T) = \begin{cases} 0; & T \geq T_m \\ 1; & T \leq T_m \end{cases} \tag{2.15}$$

Voller *et al.* [27] also expressed the relationship between the solid fraction and the latent heat of the material, resulting in an expression for the source term, $S_e$, in the energy equation, which is discussed next.

### 2.2.1 The Energy Equation Source Term, $\boldsymbol{S_e}$

The total enthalpy of the material, $h_{tot}$, is the sum of sensible (static) enthalpy, $h_{sen}$, and the latent heat, $h_L$,

$$h_{tot} = h_{sen} + h_L \tag{2.16}$$

where the sensible enthalpy is given by,

$$h_{sen} = h_{ref} + \int_{T_{ref}}^{T} c_p \, dT \tag{2.17}$$

Here, $h_{ref}$ and $T_{ref}$ are the reference enthalpy and reference temperature respectively. The latent heat varies between zero for the solid phase to $L$ (the latent heat of fusion) for the liquid phase. Then, for a material undergoing phase change over the temperature range $[T_s, T_l]$, the functional form of the latent heat is expressed as,

$$h_L = \begin{cases} L; & T \geq T_l, f_g \neq 1 \\ L(1-s); & T_s < T < T_l, f_g \neq 1 \\ 0; & T \leq T_l, f_g \neq 1 \\ 0; & f_g = 1 \end{cases} \tag{2.18}$$

For a pure substance, the latent heat's variation is instead given as,

$$h_L = \begin{cases} L; & T \geq T_m, f_g \neq 1 \\ 0; & T \leq T_m, f_g \neq 1 \\ 0; & f_g = 1 \end{cases} \tag{2.19}$$

Eq. (2.14), (2.16) and (2.18) together describe the relationships between the latent heat, solid fraction and the total enthalpy of the material undergoing phase change over the temperature range $[T_s, T_l]$. For a pure substance, the relationships are described by Eq. (2.15), (2.16) and (2.19) instead. Next, substituting Eq. (2.16) into Eq. (2.5) and rearranging, we have,

$$\frac{\partial}{\partial t}(\rho h_{sen}) + \nabla \cdot (\rho \mathbf{U} h_{sen}) = \nabla \cdot (k \nabla T) - \underbrace{\left[\frac{\partial}{\partial t}(\rho h_L) + \nabla \cdot (\rho \mathbf{U} \mathrm{h}_L)\right]}_{S_e} \tag{2.20}$$

In this manner, the contribution of the solidification process to the energy equation manifests as a volumetric source term proportional to the latent heat, which is in turn related to the solid fraction. Thus, in the mushy region where solidification occurs, the source term assumes a non-zero value based on the solid fraction, whereas in purely liquid or solid regions, the source term drops to zero. On the other hand, the source term is always zero in the air since no phase change occurs here.

### 2.2.2 The Modified Momentum Equations

In the enthalpy-porosity method, Voller *et al.* [27] proposed a mechanism to halt the velocities to zero when the cell has fully solidified. This was done by defining the volumetric source term $S_p$ as,

$$S_p = -A\mathbf{U} \tag{2.21}$$

where $A$ is defined such that it increases from zero when the cell is liquid to a very large value when the cell is completely solid. This source term is closely related to the Darcy source term of porous media flows [40], which represents the viscous drag within the pores. In purely liquid cells, the source term assumes a value of zero. In mushy region cells, the value of the source term increases such that Eq. (2.4) approximately models the momentum equation for a porous media flow. Finally, when the cell is completely solid, the source term assumes a very large value which dominates all other terms in Eq. (2.4). This forces the momentum equations to lead to trivial solutions, thereby setting the cell velocities to zero.

To achieve the intended behavior of $A$, the value of $A$ is set to be dependent on the solid fraction of the cell. Voller *et al.* [27] assumed a form for $A$ as,

$$A = -C_{mushy} \frac{(1 - \varepsilon)^2}{\varepsilon^3 + q} \tag{2.22}$$

where $C_{mushy}$ is a constant called the mushy zone constant and $q$ is a value introduced to avoid division by zero (usually set to 0.001).

In this manner, the enthalpy-porosity method can model the movement of the solidification (phase change) boundary in a three-phase system. However, as described earlier, this only constitutes one of two moving boundaries in such a system. In the following section, an outline of the VOF method used to track the gas-(liquid/solid) surface (free surface) is provided.

### 2.3 The Volume-Of-Fluid (VOF) Method

In Chapter 1, it was highlighted that in the present problem of a partially filled tank, a separate method is required to track the movement of the gas-(liquid/solid) interface since expansion during the freezing of water causes this interface to move. The enthalpy-porosity method is unsuitable to track this interface as its movement is the consequence of mechanical (expansion) processes. To model flows with a moving material interface or a free surface, two basic approaches exist: (1) interface tracking methods, and (2) interface capturing methods. In interface tracking methods, the interface is treated as a region of steep gradient in some quantity (for example, density) and the mesh is updated as the flow evolves. The marker-and-cell (MAC) and front-tracking methods are examples of interface tracking methods. Further discussions of the methods are presented by McKee *et al*. [57] and Tryggvason *et al*. [58], respectively. On the other hand, in interface capturing methods an interface function (piecewise polynomials, level-set functions, cell variables, etc.) is used to represent the location of the interface. These methods are based on fixed grids/meshes where the interface function is computed to "capture" the interface. Then, reconstruction algorithms are employed to construct the location of the interface. While interface tracking methods yield more accurate representations of the interface, they involve significant complexity by virtue of requiring costly mesh updates. Therefore, interface capturing methods are used as a cheaper and simpler alternative, as they are compatible with fixed meshes. In this section, a brief description is provided for one of the most popular interface capturing methods applied to multiphase flows – the Volume-Of-Fluid (VOF) method.

The essential feature of the VOF method is that it tracks the volume of each phase in cells that contain the interface. For example, consider the air-water interface in a partially filled water tank as shown in Fig. 2.1, and the goal is to track the air-water interface. As seen in the figure, the air-water interface is contained within several computational cells in the grid. In the VOF method, the exact location of this interface is translated to a set of

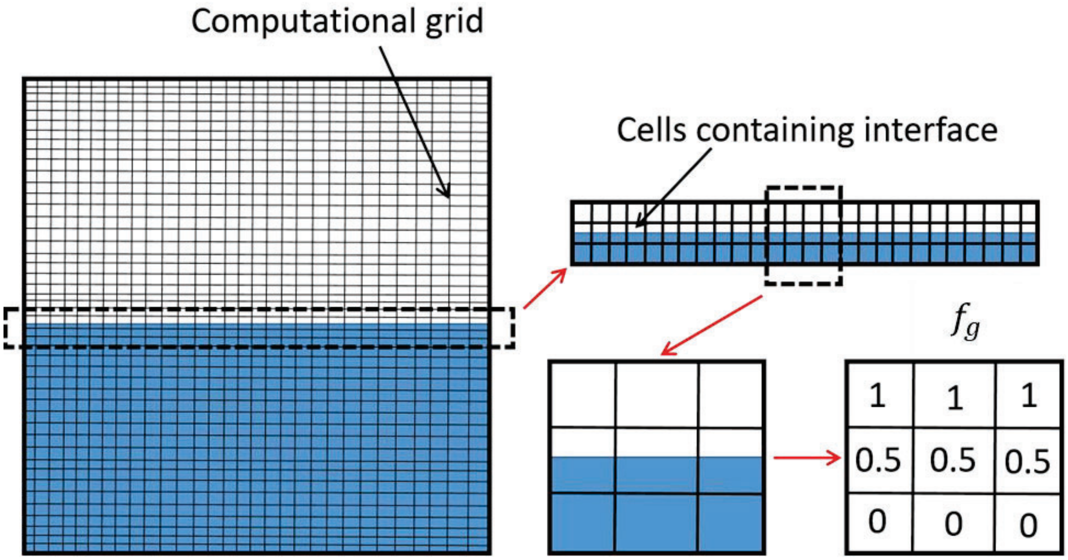


**Figure 2.1: Schematic of a partially filled tank showing a stencil around the air-water interface.**

discrete volume data by means of computing the volume fraction for each grid cell. For the air phase, the volume fraction is defined as shown in Eq. (2.2). It is apparent that for cells that contain the interface, the volume fraction is between 0 and 1. The evolution of the interface with time is then obtained by following the movement of fluid volumes computed as dictated by the advection equation shown in Eq. (2.11).

It can be noted in this example that the specified interface geometry guarantees a unique distribution of volume data. However, the converse is not true. That is, unique interface information cannot be extracted from a given volume fraction distribution. Instead, interface geometry must be inferred based on the local volume and the assumption of a particular *reconstruction* algorithm. The VOF method offers several interface reconstruction algorithms to determine the approximate location of the interface from a given field of volume fractions. Some of the popular algorithms are described in the following section and the algorithm relevant to the present study is highlighted.

### 2.3.1 Reconstruction of the Interface

Since their development in the 1970s, most volume tracking methods published to date utilized one of three algorithms for interface reconstruction: (1) Piecewise Linear Interface Calculation (PLIC) scheme, (2) Simple Line Interface Calculation (SLIC) scheme, and (3) Piecewise constant/"stair-stepped" scheme. In all of these schemes, line segments (or planes in three dimensions) are used to represent the reconstructed interface within each cell. However, the differences between the methods arise in the assumed interface geometry. Of these, the PLIC scheme, pioneered by DeBar [43] and Youngs [44] in 1974, was the first to be proposed, and notably, is also the most accurate of the three. In this method, the interface between the two phases (and the resulting line segment in the reconstruction) is assumed to have a linear slope within each cell. This allows for the interface reconstruction to closely approximate the actual interface. However, the complexity of the implementation of this method prompted the development of simpler schemes. Subsequently, Noh and Woodward's SLIC scheme [42], as the name suggests, is a simplification of the PLIC scheme, where the interface within each cell is instead assumed to be a line segment aligned with one of the mesh coordinates. Fig. 2.2 shows a simple stencil outlining the major differences between the SLIC and PLIC schemes. As seen in the figure, the piecewise constant approximation in the SLIC scheme forces the reconstructed interface to align with the Cartesian coordinates, whereas the piecewise linear approximation in PLIC allows the interface to be oriented with the interface's shape. Lastly, the piecewise constant or stair-stepped scheme can be considered to be an extension of the SLIC scheme, where the line representing the interface within a cell is allowed to "stair-step" i.e., allowed to align with more than one mesh coordinate. This scheme was first proposed and employed in the original description of the VOF method by Hirt and Nichols [28]. An in-depth historical review of volume tracking methods can be found in Ref. [56].

The PLIC scheme is typically the preferred scheme in modern interface tracking algorithms. Incidentally, the commercial CFD code ANSYS Fluent™ uses the PLIC scheme in its VOF formulation, where it is termed as the "Geometric Reconstruction Scheme". However, the SLIC and piecewise constant/stair-step methods are still in widespread use due to their relative ease of implementation.

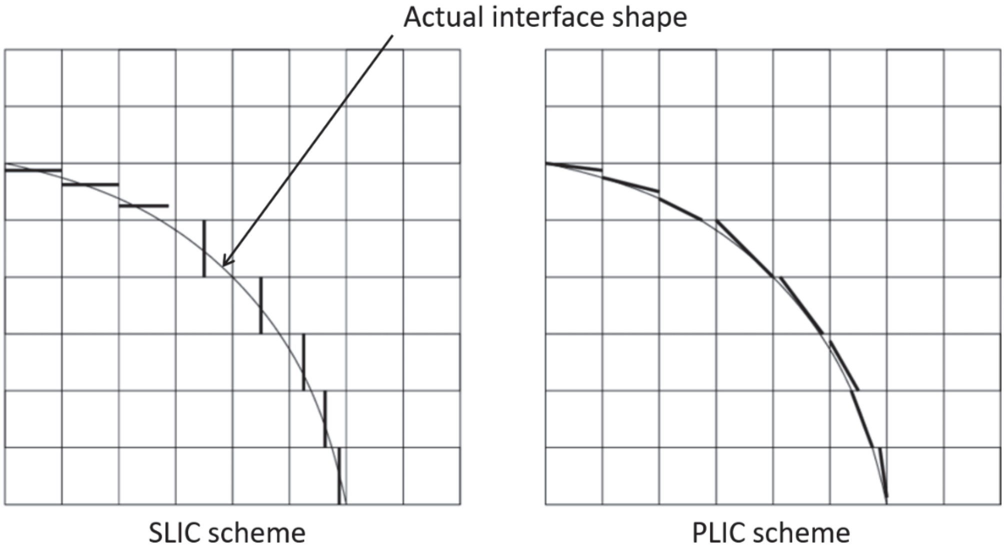


**Figure 2.2: A two-dimensional stencil showing the approximate reconstruction of a given interface using the SLIC and PLIC schemes.**

## 2.4 CFD Simulations of solidification of water using a commercial CFD code

The goal of this study was to test and assess the capabilities of the high-fidelity, in-built physical models provided in commercial CFD codes for solidification of water. At the behest of the Ford Motor Company, the sponsors of the current work, ANSYS Fluent™ was the chosen CFD software. A simple 2D planar tank was used in this study, and Fig. 2.3(a) shows a schematic of this geometry. The dimensions of the tank were chosen to be approximately 1/4$^{th}$ the size of the real tank used in the experiments (to be discussed later in Chapter 4). These decisions were motivated by the fact that testing the in-built models could potentially involve many test runs while requiring significant computational resources and time to reach solidification in full-size 3D simulations. A smaller sized tank would solidify quicker and would therefore require less computational time. The water level in the tank was chosen to be at the 50% height level to allow for natural convection to occur in both air and water.

A structured mesh with 10000 quadrilateral cells was used in the simulation and is shown in Fig. 2.3(b). Coarser meshes comprised of 3000, 5000 and 7000 cells were also used initially for the purpose of testing grid independence, and it was found that temperature predictions and resulting velocity contours did not show significant change beyond 7000 cells. Therefore, the final mesh size was chosen to be 10000 cells. As can be seen in Fig. 2.3(b), the mesh density was made uniform everywhere to allow sufficient resolution of the moving interface and natural convection patterns within both phases.

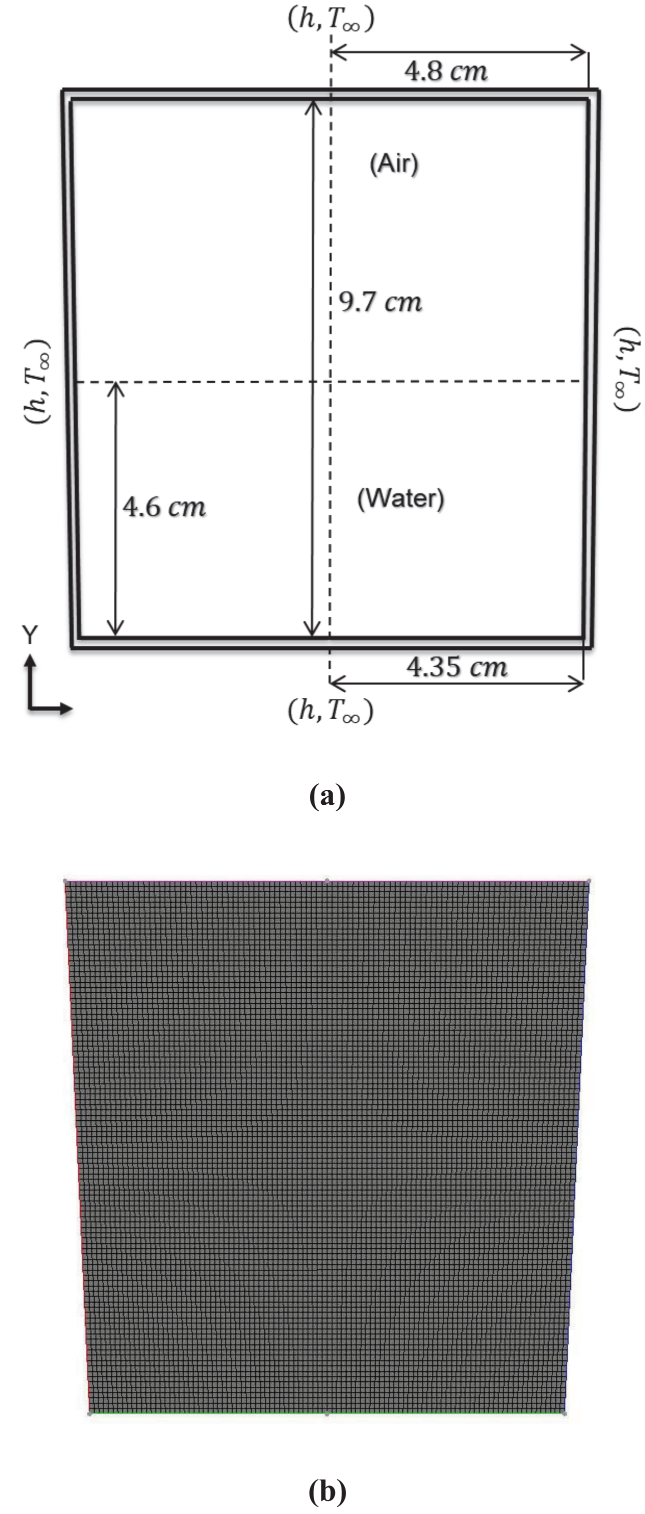


**Figure 2.3: (a) Schematic of the 2D tank showing the geometry and water fill level (b) Mesh used for the CFD computations.**

### 2.4.1 CFD Model Parameters

Some of the important considerations in the CFD model pertaining to material properties are provided below:

- In order to capture the natural convection patterns in water, the density of water was considered to be temperature dependent, utilizing the functional form provided in Ref. [33]. Fig. 2.4 below shows a plot of this density variation of water with temperature. Crucially, the plot shows that this density profile captures the density inversion at 4°C. This density inversion is expected to play an important role in the computed results and its subsequent interpretation, as will be demonstrated later.
- Air was assumed to behave as an ideal gas.
- Other thermophysical properties of water were assumed to be constant. Table 2.1 summarizes the properties used in the analysis.
- The thermophysical properties of the solid tank wall (HDPE) were set to be temperature independent (properties referenced in Table 2.1).
- For ice, the density was considered to be temperature dependent and its functional form was utilized from Ref. [34]. The solidus and liquidus temperatures were assumed to be $T_s$ = 0°C and $T_l$ = 0.05°C while the latent heat of fusion was $L$ = 333,550 J/kg.

For thermal boundary conditions, all four boundary walls were set as convective heat transfer boundary conditions with heat transfer coefficients of $h$ = 100 W/m$^2$K and constant ambient temperature of $T_\infty$ = -10°C on all sides. These values were chosen to facilitate faster heat transfer and achieve quick solidification during the simulation. Lastly,

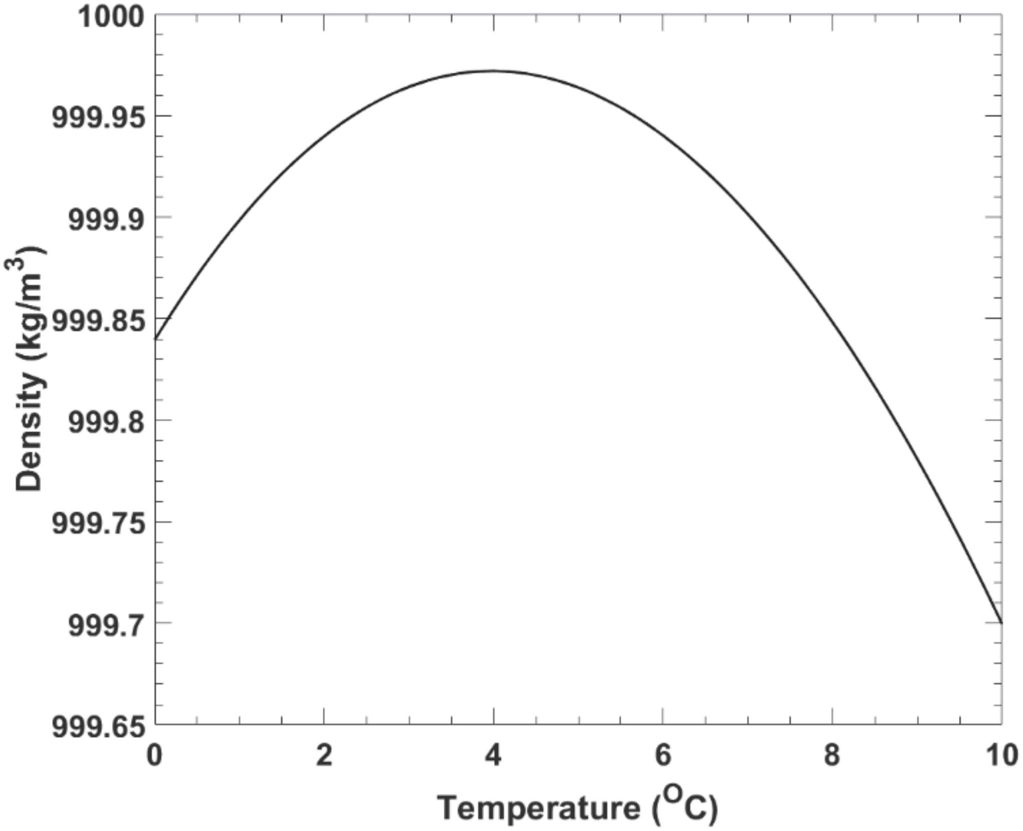


**Figure 2.4: Density variation of water with temperature, showing the density inversion at 4°C.**

**Table 2.1: Thermo-physical properties for the solid wall and two phases used in the CFD calculations.**

| Material | Density (kg/m$^3$) | Specific Heat (J/kg-K) | Thermal Conductivity (W/m-K) | Dynamic Viscosity (kg/m-s) |
|---|---|---|---|---|
| Plastic | 932 | 1900 | 0.5 | N/A |
| Air | Ideal gas law | 1006.43 | 0.024 | $1.789\times10^{-5}$ |
| Water | Ref. [33] | 4182 | 0.6 | $1.003\times10^{-3}$ |

the simulation was initialized with a uniform temperature distribution of 5°C everywhere.

Based on these model parameters, the Rayleigh number for each of the phases was calculated using the following relation [45]

$$\mathrm{Ra} = \frac{g\Delta\rho L_C^3}{\alpha\mu} \tag{2.23}$$

where $g$ is the acceleration due to gravity, $\alpha$ and $\mu$ are the thermal diffusivity and dynamic viscosity of the material, respectively. $\Delta\rho$ denotes the local density difference and $L_C$ is the characteristic length scale for the problem. In this case, $L_c$ is the height of the respective phases. Consequently, the Rayleigh number for the air and water phases were estimated to be $3\times10^5$ and $2.5\times10^4$, respectively. These are both well below the critical Rayleigh number for turbulence, which is approximately equal to $10^9$ [52]. Therefore, the flow in these problems can be considered to be laminar and any turbulence effects can be neglected.

### 2.4.2 Results

In this section, the results for the CFD simulation of the test tank are presented. A constant time-step size of 0.5 milliseconds was used in all computations. Even though an implicit time-formulation was used, this time-step size was required for stability of the explicit VOF algorithm, as time-step sizes larger than this value led to divergence. Figure 2.5 shows a contour plot of the solid fraction, *s*, at time 4000 seconds. As seen in the figure, solidification begins from the side and bottom walls and progresses inwards, as expected when cooling occurs at the walls. Furthermore, heat transfer through the air by natural convection results in the formation of a very thin layer of ice ($0.1< s <0.2$) at the air-water interface. As described earlier in this study, this is typical of the freezing phenomena in a partially-filled tank—analogous to the freezing of water in a lake which causes the surface to freeze prior to the water underneath.

Another important observation from Fig. 2.5 is the apparent lack of smoothness and symmetry of the solidification front, particularly near the bottom wall. This is due to the fact that the bottom of the tank is a region of unstable stratification. This can be easily seen in the velocity contours at the same time instant shown in Fig. 2.6(b), where the velocities near the bottom interface are strongly unsteady. It must be noted that the regions with no arrows near the bottom and side walls represent regions of pure solid ice, and therefore zero velocity. The unstable stratification near the bottom of the tank is a consequence of

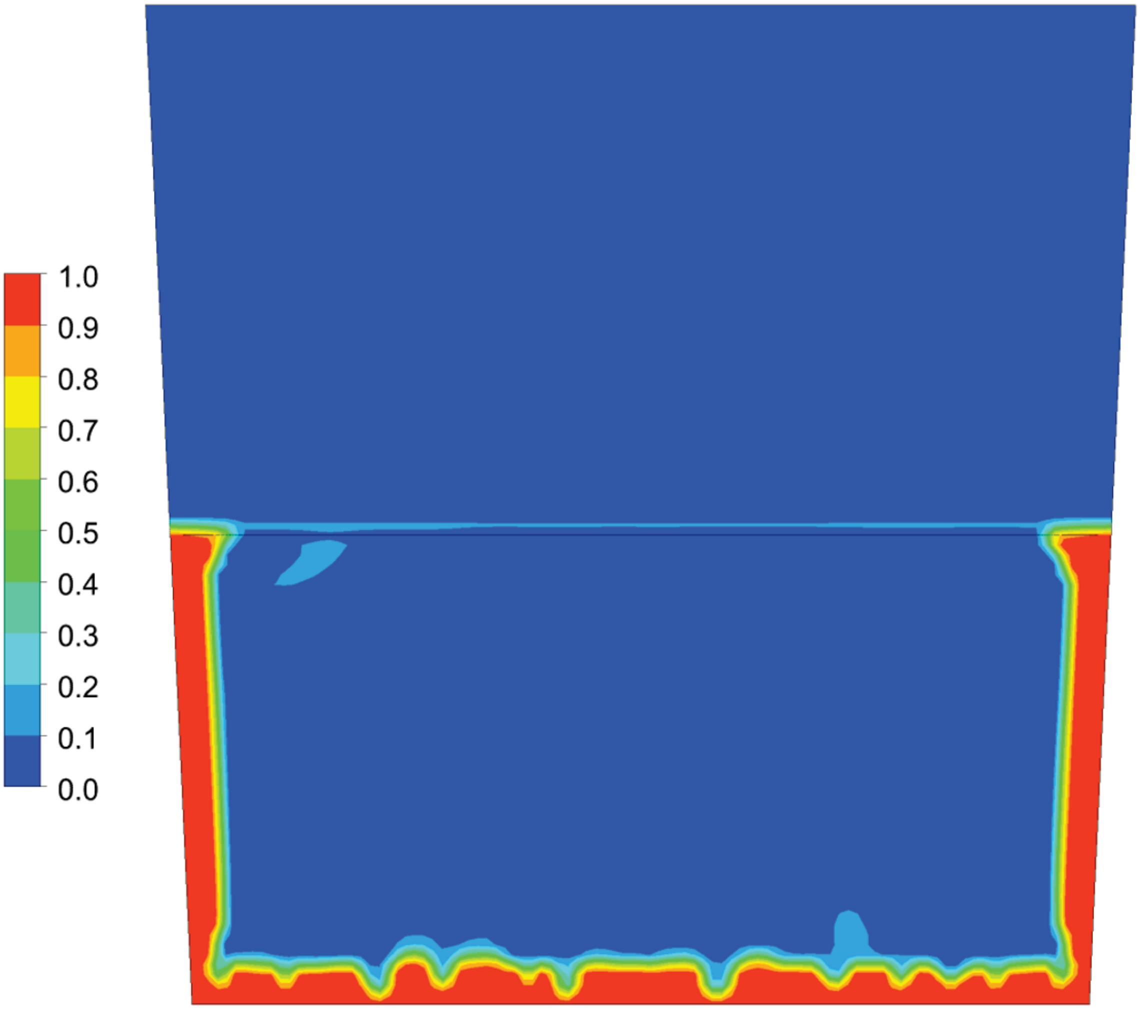
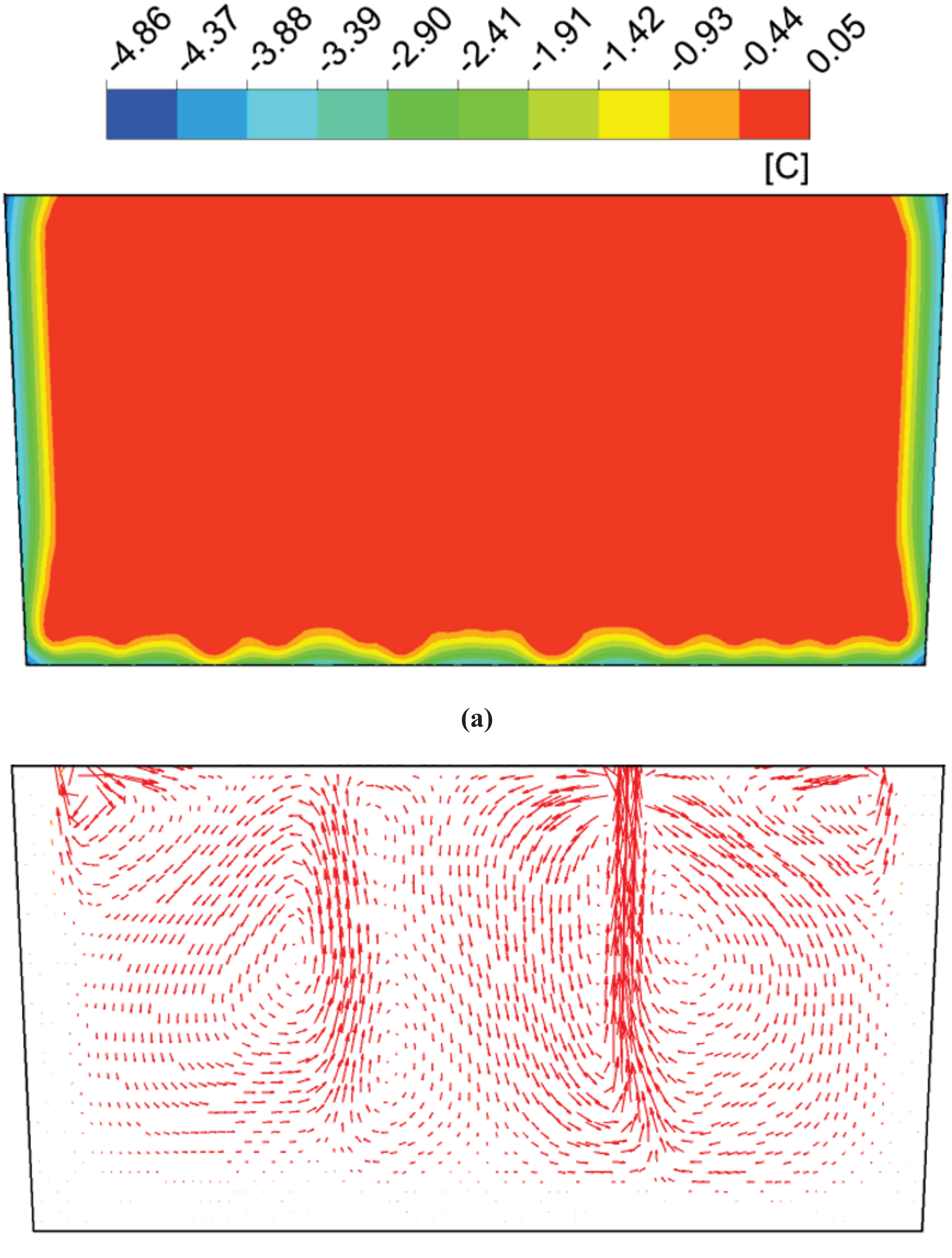


**Figure 2.5: Solid fraction in the tank after 4000 seconds.**



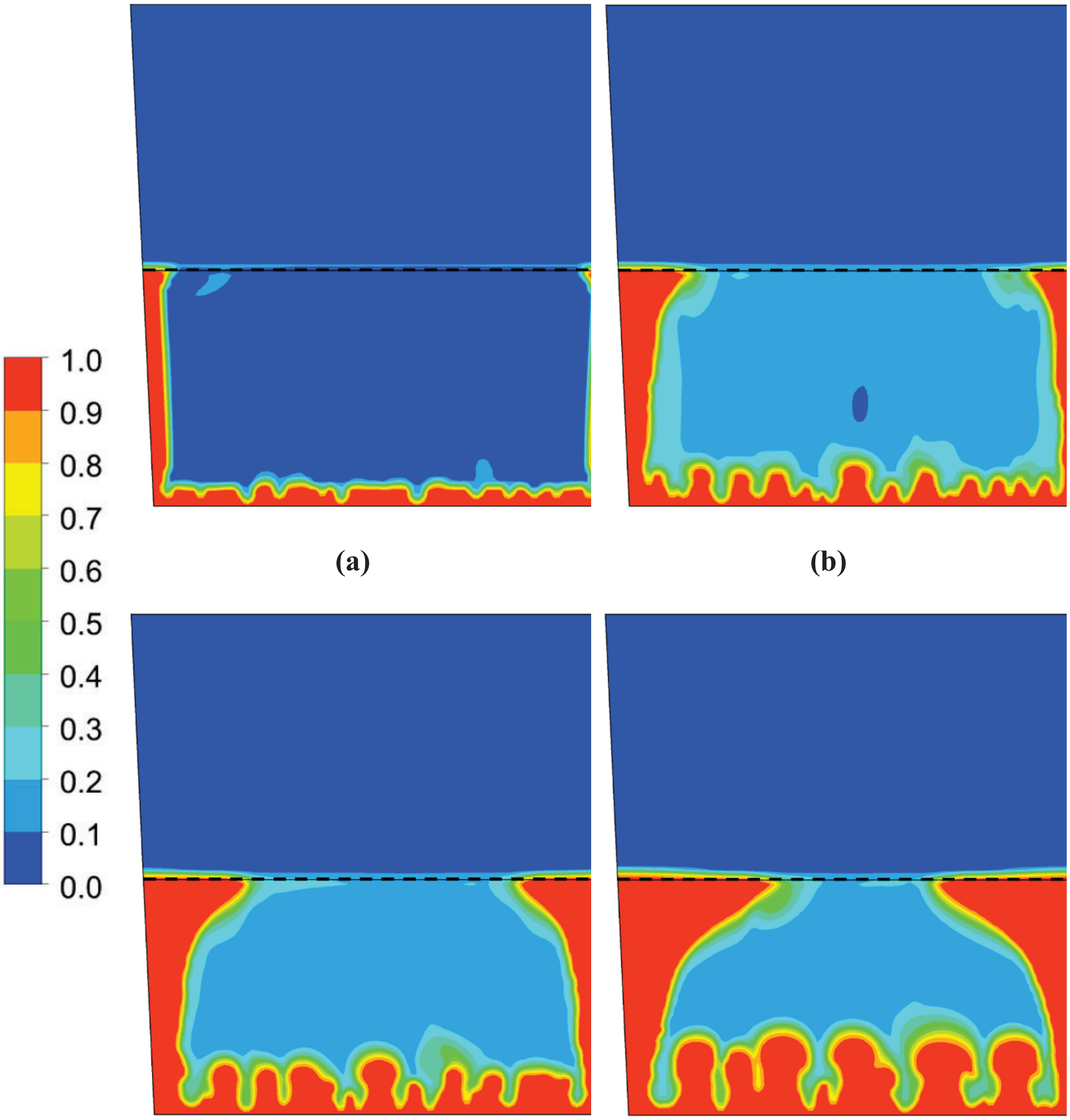
(a)

(b)

**Figure 2.6: (a) Temperature distribution, (b) Flow pattern in the water domain after 4000 seconds.**

the density inversion of water at 4°C. As water cools below 4°C, its density decreases, causing it to flow upward before completely solidifying. The velocity contours in Fig. 2.6(b) also showcase the asymmetric nature of natural convection patterns in the bulk of the water, especially near the center and the bottom. This further promotes the unevenness of the solidification front which amplifies the asymmetry of the system. Fig. 2.7 shows a sequence of contours of the solid fraction at four different time instances of 4000, 8000, 10000 and 14000 seconds.

The initial water level is denoted by the black dashed line seen in all four contours. Comparing the different contours, it can be seen that the water level slowly rises as solidification progresses. This is due to the increased volume occupied by solid ice compared to liquid water, thereby pushing the water in from the bottom and sides. As seen in the figure, the formation of ice at the air-water interface has progressed further inward, forming an overhang structure. The unevenness initially seen at the bottom solidification front has evolved into mushroom-like dendritic structures. The total wall-clock time to simulate 14000 seconds (28 million time steps) of physical time for this problem was found to be approximately 2 months on a single processor. Parallelizing the simulations using ANSYS Fluent's™ parallel solver yielded no improvements to computational speed. It was found that since the cell count for this problem was small (10000 quadrilateral cells), distributing an even smaller number of cells between processors resulted in signification communication overheads between the processors, thus resulting in negative parallel computation efficiency. Hence, using the in-built VOF and solidification models in



(a)

(b)

(c)

(d)

**Figure 2.7: Solid fraction in the tank at: (a) 4000 seconds (b) 8000 seconds (c) 10000 seconds (d) 14000 seconds.**

ANSYS Fluent™, modeling this three-phase system even in two dimensions was found to be prohibitively expensive.

### 2.5 Summary

To summarize this study, the in-built VOF and solidification models in the commercial CFD code ANSYS Fluent™ were utilized to simulate the freezing of water in a partially filled 2D water tank. The size of the tank was chosen to be about 1/4$^{th}$ the size of the real tank used in the experimental studies (as will be demonstrated in the next chapter). Numerical test results revealed that extremely small time-step sizes (0.5 milliseconds) were needed to attain convergence. With this time-step size, the wall clock time required to simulate 3 hours of real time was approximately 2 months. Attempts to improve computation efficiency by parallelizing the simulation were unsuccessful, since the cell count for the current 2D problem is too small leading to significant communication overheads.

As shown in Fig. 1.5, the total experimental time required to freeze real-sized tanks is between 24 to 72 hours. Further, at the time of this study, the maximum computational capability to run a simulation in parallel was 64 processors (due to licensing restrictions). Thus, extrapolating the above wall clock time obtained in this study to simulate 24 hours (~86000 seconds) of solidification in three dimensions, and assuming perfect parallel scaling from 1 processor (10000 cells in 2D) to 64 processors (640000 cells in 3D), the wall clock time is estimated to be 12 months. This is an unfeasibly long time to conduct freeze/thaw simulations for tanks of practical length scales. The in-built VOF and solidification models are the primary bottlenecks, requiring the use of very small time-step sizes as seen in this study. Furthermore, the models require the solution to two additional equations besides the standard flow and energy equations, further slowing down computations at each time step. This, together with large cell counts for 3D tanks, results in unrealistic computation times as estimated above. These findings also indicate the reason why most studies involving CFD simulations of such problems are limited to small time and/or length scales. Consequently, these observations prompted the development of two alternative models, one for each moving boundary, which are detailed in the following chapters.

# CHAPTER 3. REDUCED NATURAL CONVECTION MODEL - THEORY

## 3.1 Introduction

As discussed in the preceding chapter, modeling the freezing process of water in a partially filled tank using the in-built physical models of ANSYS Fluent™ is prohibitively expensive. Extremely small time-step sizes (0.5 milliseconds) were needed to model the solidification process, and the resulting wall-clock times were estimated to be approximately 12 months for simulation of the entire freezing process in an actual large three-dimensional tank. Thus, utilizing the in-built models of ANSYS Fluent™ to model solidification/melting in tanks of practical interest was judged to be practically unrealistic, especially for situations requiring multiple runs (parametric studies) or repeated computations. As an alternative, a reduced model is proposed to address these difficulties. The current chapter focuses on the motivation, development, and validation of one part of this model, namely a reduced natural convection model for heat transfer during the freezing/thawing process (or simply *reduced model* for short). In this part of the development, the expansion of ice and, consequently, the movement of the air-(solid/liquid) interface is neglected. The details of the other part of the model, which handles the movement of air-(solid/liquid) interface, are provided in Chapter 5.

As explained in the study in Chapter 2, one of the major computational difficulties associated with using Fluent's in-built models is the need to model natural convection driven flow in the system. Furthermore, the time scales associated with natural convection can often be very small. Resolving these small natural convection time scales and obtaining a stable solution to the Navier-Stokes equations for multi-phase fluid flow using small time steps leads to prohibitively large computational times. To circumvent the difficulties of utilizing such small time-steps to model flow, the proposed reduced model completely bypasses solving for flow and instead models the entire natural convection driven heat transfer process as a pure conduction (diffusion) process. In theory, this would allow for the use of much larger time step sizes in addition to eliminating the need to solve the Navier-Stokes equation. The proposed reduced model achieves the goal of capturing heat transfer due to natural convection by introducing an artificial thermal conductivity such that the convective heat flux can be represented by an equivalent conductive heat flux. This artificial thermal conductivity is derived from physical laws governing natural convection. The formulation strategy is analogous to the treatment of turbulence; wherein turbulent transport is modeled using the gradient diffusion hypothesis and an effective eddy diffusivity that is computed using the physical laws that govern turbulent transport. In this manner, even though the energy equation would reduce to a pure conduction equation, the effects of natural convection on overall heat transfer rates are still considered. The net benefit is that computational efficiency can be significantly improved.

The effect of the phase change process on heat transfer also needs to be modeled. For this purpose, our model introduces a volumetric source term in the energy equation. Like the enthalpy-porosity method described in the preceding chapter, the proposed volumetric source will automatically account for the latent heat release when the material is solidifying. In other words, in regions where solidification is occurring, the source term assumes a non-zero value proportional to the latent heat of the material while in non-solidifying regions, the volumetric source term drops out.

To summarize, the proposed natural convection model results in the following energy conservation equation:

$$\rho c_p \frac{\partial T}{\partial t} = \nabla . \left[ k_{eff} \nabla T \right] + \dot{q}_s \tag{3.1}$$

where $k_{eff}$ is the effective thermal conductivity and $\dot{q}_s$ is heat generation rate per unit volume due to phase change. Assuming no turbulence effects (refer Section 2.4.1), the effective thermal conductivity is, in turn, given by

$$k_{eff} = k_m + k_{NC} \tag{3.2}$$

where $k_{NC}$ is the thermal conductivity due to natural convection (*artificial thermal conductivity*) and $k_m$ is the molecular thermal conductivity of the material (fluid, solid, or mixture). Combining Eqns. (3.1) and (3.2) gives the final form of the energy equation to be solved as

$$\rho c_P \frac{\partial T}{\partial t} = \nabla . [(k_m + k_{NC}) \nabla T] + \dot{q}_s \tag{3.3}$$

The procedure to determine $k_{NC}$ and $\dot{q}_s$ is detailed in the following sections, along with the procedure to determine thermophysical properties such as $k_m$, $\rho$ and $c_P$.

### 3.2 Model for Artificial Thermal Conductivity

Buoyancy-driven natural convection plays a critical role in the heat transfer process within a partially filled tank of water. The primary goal of the reduced model is to be able to account for the effect of natural convection in heat transfer within a pure conduction framework. As mentioned earlier, the fundamental idea of how to introduce natural convection in such a form comes from turbulence modeling. In turbulent heat transfer, a common approach is to use a new conductivity – the eddy conductivity – to calculate the effect of enhanced heat transfer due to turbulent mixing. Analogously, we define an artificial thermal conductivity due to natural convection, referred to as $k_{NC}$, such that the effective, enhanced thermal conductivity, $k_{eff}$, can written as shown in Eq. (3.2). As the motivation behind defining this artificial thermal conductivity is to make the conductive heat flux equivalent to the convective heat flux, to achieve this it must follow that

$$q_{NC} = h_{NC}\Delta T = k_{NC}\frac{\Delta T}{l} \tag{3.4}$$

where $q_{NC}$ is the heat flux due to natural convection, and $h_{NC}$ is the corresponding heat transfer coefficient due to natural convection. The quantity, $l$ , the so-called local length scale, is defined as a length scale over which the temperature difference (or gradient in three-dimensional scenarios), $\Delta T$, is established. This local length scale is likely to be significantly smaller than the global length scale ($L_c$) for the problem at hand. For example, in the case of a natural convection driven boundary layer next to a hot vertical plate, the length scale over which the temperature difference ΔT is established is the thermal boundary layer thickness, which would be the local length scale in accordance with the aforementioned definition. In contrast, the global length scale is the length of the plate which is significantly larger than the local length scale in accordance with the definition of a boundary layer. Similarly, Fig. 3.1 shows the distinction between the two length scales for a partially filled tank. As $l$ is the length scale over which the temperature difference is

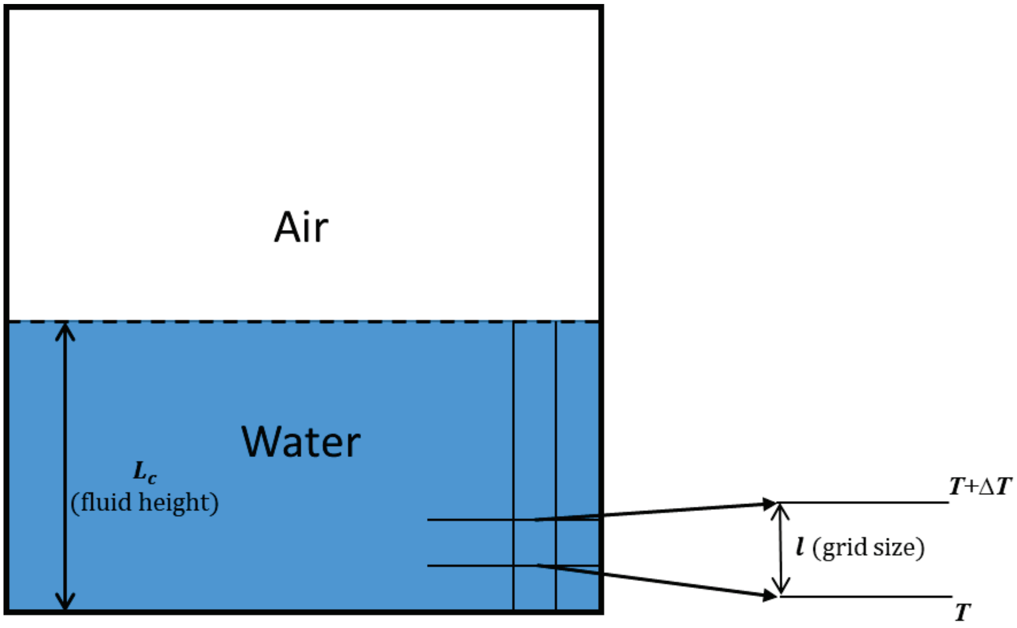


**Figure 3.1: Schematic of a partially filled tank of water illustrating the distinction between local and global length scales.**

established, it makes logical sense to choose $l$ such that it represents the size of the computational cell is some manner. It is therefore proposed to calculate the local length scale using the following formulation,

$$l = \frac{V}{A_s} \tag{3.5}$$

where $V$ and $A_s$ are the volume and surface area of a typical control volume (or cell), respectively.

Next, nondimensional analysis provides a general relationship where the convective heat transfer coefficient is given by [45],

$$h_{NC} = \frac{k_m}{L_C}\mathrm{Nu} = \frac{k_m}{L_C}c\mathrm{Ra}^a\mathrm{Pr}^b \tag{3.6}$$

where Nu is the Nusselt number, $L_c$ denotes the characteristic global length scale while Ra and Pr are the Rayleigh and Prandtl numbers, respectively. $a$, $b$ and $c$ are unknown constants that depend on the type of flow (laminar or turbulent), the geometry of the system, and the boundary conditions, and are generally derived from scaling or order-of-magnitude analysis. Combining Eqns. (3.5) and (3.6) gives

$$k_{NC} = lh_{NC} = l\frac{k_m}{L_C}c\mathrm{Ra}^a\mathrm{Pr}^b \tag{3.7}$$

Substituting the expression for Rayleigh number given by Eq. (2.23) into Eq. (3.7) gives,

$$k_{NC} = l\frac{k_m}{L_C}c\left(\frac{g\frac{\Delta\rho}{\rho_0}L_c^3}{\alpha\nu}\right)^a \mathrm{Pr}^b \tag{3.8}$$

By combining some constants, Eq. (3.8) can be simplified to

$$k_{NC} = C_{NC} l L_c^{3a-1} (\alpha \nu)^{-a} \left(\frac{\Delta\rho}{\rho_0}\right)^a \mathrm{Pr}^b k_m \tag{3.9}$$

where $C_{NC} = cg^a$. In theory, $a$ and $b$ are determined using scaling or order-of-magnitude analysis. However, in the current study involving partially filled tanks, the shape of the tanks is not always known. Even if the tank shapes were known, the corresponding flow patterns can be quite complex. Therefore, scaling analysis can prove to be difficult, if not outright impossible. Instead, an engineering approach is adopted to estimate the exponents $a$ and $b$. In the partially filled tanks of interest, the fluid is completely enclosed between a combination of horizontal and vertical walls. Thus, the tank geometries associated with the flow can be regarded as vertical plates, horizontal plates or an enclosure. For laminar flow, the empirical correlations from the literature [45] for horizontal and vertical plates are purely a function of Ra, while for an enclosure the Nusselt number is a function of both Ra and Pr. These correlations are shown in Table 3.1.

**Table 3.1: Empirical correlations from literature [45] for exponent $a$ and $b$ for various geometries.**

| **Geometry Type** | $\boldsymbol{Value\ of\ a}$ | $\boldsymbol{Value\ of\ b}$ |
|---|---|---|
| Horizontal surface | 0.25 | 0 |
| Vertical surface | 0.33 | 0 |
| Enclosure | 0.33 | 0.05 |

As seen in Table 3.1, the exponent $a$ of the Rayleigh number is bounded between 0.25 or 0.33. A simple average of these bounds is chosen for the value for $a$. The value of $b$ is assumed to be zero since the dependence on the Prandtl number is already included in the Rayleigh number. In short,

$$a = 0.275; \;\; b = 0 \tag{3.10}$$

Substituting the above values of $a$ and $b$ into Eq. (3.9), it can be written as

$$k_{NC} = C_{NC} l L_c^{-0.175} (\alpha \nu)^{-0.275} \left(\frac{\Delta \rho}{\rho_0}\right)^{0.275} k_m \tag{3.11}$$

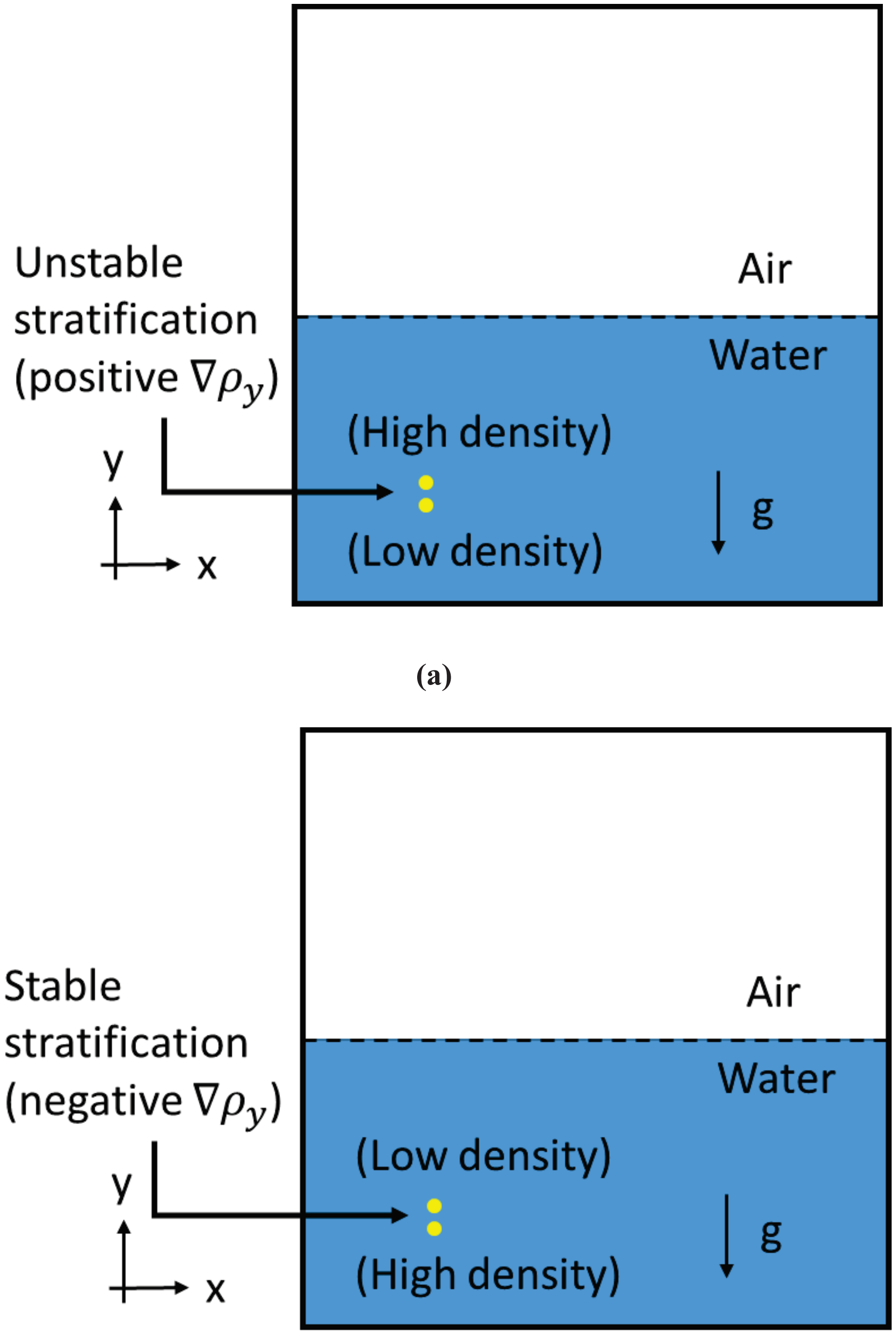
For three-dimensional calculations, the density variation can be expressed as

$$\Delta \rho = l |\nabla \rho| \tag{3.12}$$

where the density gradient $\nabla \rho$ can be further expanded as

$$|\nabla \rho| = \left[(\nabla \rho_x)^2 + \left(\nabla \rho_y\right)^2 + (\nabla \rho_z)^2\right]^{1/2} \tag{3.13}$$

Further details on the calculation of the density gradient are provided in Section 3.4. However, once the density gradient has been calculated, it must be taken into account that natural convection does not occur in the presence of any arbitrary density gradient profile. To illustrate this, consider the scenarios shown in Fig. 3.2. Here, the gravity vector is assumed to be pointing in the negative y-direction. Fundamentally, for natural convection to occur, buoyancy-driven flow must be present. As seen in Fig. 3.2(a), for such a flow to occur, unstable stratification must be present, i.e., a region of high density must be "above" a region of lower density ($\nabla \rho_y > 0$). If instead, stable stratification ($\nabla \rho_y < 0$) is present (Fig.3.2(b)), no buoyancy-driven flow is setup, and thus no natural convection takes place. More generally, it can be stated that natural convection occurs only if the



(a)

(b)

**Figure 3.2: Schematics of a partially filled tank showing (a) an unstable stratification configuration and, (b) a stable stratification configuration.**

component of density gradient aligned with the gravity vector is positive. Thus, in order to account for the unstable stratification effect, Eq. (3.13) is modified to

$$|\nabla\rho| = \left[(\nabla\rho_x)^2 + \left(\mathrm{MAX}(0, \nabla\rho_y)\right)^2 + (\nabla\rho_z)^2\right]^{1/2} \tag{3.14}$$

Generally, the x- and z-components of the density gradient should not cause heat transfer via natural convection. However, the situation can be somewhat different adjacent to walls. For example, consider a natural convection boundary layer next to a vertical plate. There is strong upward or downward flow (depending on whether the plate is heated or cooled) adjacent to the wall even though the gradient in density is normal to the plate, i.e., in the x- or z-directions. Since the current problem is likely to involve such scenarios, the gradients in the x- and z-directions are still retained in Eq. (3.14) in order to be able to capture such behavior.

Next, substituting the modified expression for $|\nabla\rho|$ from Eq. (3.14) into Eq. (3.11) yields

$$\begin{aligned} k_{NC} = C_{NC}(\rho_0\alpha\nu)^{-0.275}L_c^{-0.175}l^{1.275}\Big[(\nabla\rho_x)^2 &+ \left(\mathrm{MAX}(0, \nabla\rho_y)\right)^2 \\ &+ (\nabla\rho_z)^2\Big]^{0.1375} k_m \end{aligned} \tag{3.15}$$

Equation (3.15) clearly shows that the local thermal conductivity due to natural convection in a given fluid, which is a measure of the local strength of natural convection, will depend on four factors: the local density gradient, the length scale over which the gradient is established, the global length scale, and the thermophysical properties of the fluid. Of these, the density gradient is dependent on the temperature and phase distribution, causing regions with large density gradients to have high enhancements of *k*. This is most apparent in the solidification region, since the density change along the solidification front is significant and causes strong natural convection currents in the solid-liquid mixture.

The expression for $k_{NC}$ in Eq. (3.15) can be substituted into Eq. (3.2) to obtain the following final expression for effective thermal conductivity for the reduced natural convection model.

$$k_{eff} = \left[1 + C_{NC}(\rho_0 \alpha \nu)^{-0.275} L_c^{-0.175} l^{1.275} \left[(\nabla\rho_x)^2 + \left(\mathrm{MAX}(0, \nabla\rho_y)\right)^2 + (\nabla\rho_z)^2\right]^{0.1375}\right] k_m \tag{3.16}$$

Much like in a turbulence model, the model proposed in Eq. (3.16) has one undetermined constant, $C_{NC}$. This constant needs to be calibrated by employing experimental data, as will be discussed in Chapter 4. A final important point to note, particularly during the implementation of this reduced model, is to perform this thermal conductivity enhancement solely in fluid (liquid and/or gas) regions. As natural convection can only take place in fluids, purely solid regions must not undergo any form of thermal conductivity enhancement. In a similar vein, as natural convection process differs in liquids and gases, they will each experience differing levels of thermal conductivity enhancement due to their differing thermophysical properties and corresponding length scales.

### 3.3 Model for Phase Change

As discussed earlier, the reduced model introduces a volumetric source term in the energy equation in order to capture the effect of the phase change process on heat transfer. Therefore, this source term must represent the heat generation rate per unit volume due to phase change. To obtain an appropriate expression for such a source term, consider the scenario shown in Fig. 3.3, where a control volume consisting of a mixture of solid and liquid phases is undergoing phase change. Let the mass of the solid phase inside the control volume at a time $t$ be denoted as $m_s$. The solid fraction, $s$, is defined as the ratio of the mass of the solid and the total mass within the control volume, i.e.,

$$s = \frac{m_s}{m} \tag{3.17}$$

If time $dt$ elapses, the amount of solid mass will change within the control volume due to phase change. Let this change be $dm_s$. The heat transfer that must occur to cause this phase change may be written as

$$\dot{q}_s V dt = L_s dm_s \tag{3.18}$$

where $L_s$ is the latent heat of solidification (or fusion). This equation may be rearranged to yield

$$\dot{q}_s = \frac{L_s}{V} \frac{dm_s}{dt} \tag{3.19}$$

To obtain an expression for the derivative in the equation above, Eq. (3.17) can be rewritten as shown in Eq. (3.20) below.

$$m_s = sm = s\rho V \tag{3.20}$$

This equation can then be differentiated (with the assumption of a fixed control volume) to obtain

$$\frac{dm_s}{dt} = V \frac{d(\rho s)}{dt} \tag{3.21}$$

Substituting Eq. (3.21) into Eq. (3.19) then yields

$$\dot{q}_s = L_s \frac{d(\rho s)}{dt} \tag{3.22}$$

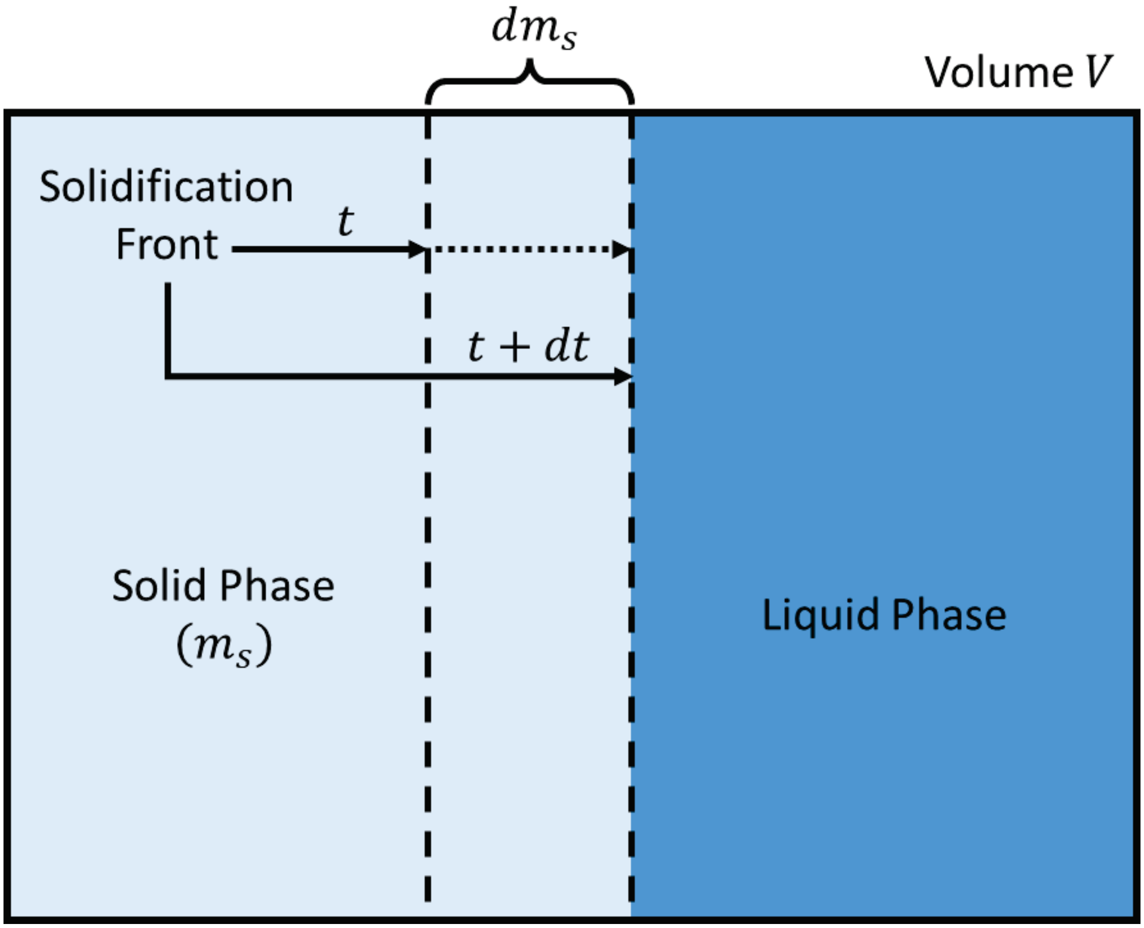


**Figure 3.3: Schematic of a control volume undergoing phase change from liquid to solid over a time period $t$ to $t + \Delta t$.**

Equation (3.22) is the required expression for a volumetric source term representing the heat generation rate per unit volume due to phase change and is similar to the source term used in the enthalpy-porosity method (Eq. (2.20)). Additionally, as the expression is dependent on the rate of change of solid fraction, it has a non-zero value only in regions where solidification is occurring. Elsewhere, it automatically drops out and does not require any additional modifications.

### 3.4 Calculation of thermophysical properties and density gradient

When solving the energy equation, the values of various thermophysical properties such as density, thermal conductivity and specific heat capacity are required. In solidification/melting problems, determining these properties is complicated by the fact that cells are often simultaneously occupied by both solid and liquid phases. Therefore, models are required to appropriately evaluate these properties in all possible configurations.

It follows logically that the thermophysical properties for cells containing both phases (henceforth referred to as *solid-liquid cells,* represented with subscript *sl*) should be dependent on the properties of the individual pure phases and the relative quantities of each phase present in the cell. As detailed in the previous section, the quantity, solid fraction, is defined as the ratio between the mass of the solid and the total mass within the cell (Eq. (3.17)) and therefore, thermophysical properties for solid-liquid cells are calculated using appropriate mixing laws based on their solid fraction.

The solid fraction for each cell is calculated based on the temperature of the cell as shown below [26].

$$s(T) = \begin{cases} 0; & T \geq T_l, \\ \dfrac{T - T_l}{T_s - T_l}; & T_s < T < T_l \\ 1; & T \leq T_s \end{cases} \tag{3.23}$$

where $T_l$ and $T_s$ are the liquidus and solidus temperatures of the material undergoing phase change, respectively.

The densities for the pure liquid and solid phases are determined by the density versus temperature relationships provided in literature ([46],[47]). The density for solid-liquid cells is consequently calculated using a solid fraction-based inverse mixing law as given below.

$$\frac{1}{\rho_{sl}} = \left(\frac{1}{\rho_{s,T_s}}\right) s + \left(\frac{1}{\rho_{l,T_l}}\right)(1 - s) \tag{3.24}$$

where $\rho_{s,T_s}$ and $\rho_{l,T_l}$ are the densities of the solid and liquid phases at the solidus and liquidus temperatures, respectively. The above expression is obtained from mass and volume conservation. This can be seen by substituting Eq. (3.17) alongside the appropriate expressions for $\rho_{s,T_s}$ and $\rho_{l,T_l}$ in the right-hand side of Eq. (3.24), which will result in the correct expression $\rho_{sl}$.

$$\begin{aligned} \left(\frac{1}{\rho_{s,T_s}}\right) s + \left(\frac{1}{\rho_{l,T_l}}\right)(1 - s) &= \left(\frac{V_s}{m_s}\right)\frac{m_s}{m} + \left(\frac{V_l}{m_l}\right)\left(1 - \frac{m_s}{m}\right) \\ &= \left(\frac{V_s}{m_s}\right)\frac{m_s}{m} + \left(\frac{V_l}{m_l}\right)\left(\frac{m_l}{m}\right) = \frac{V_s + V_l}{m} = \frac{1}{\rho_{sl}} \end{aligned} \tag{3.25}$$

where $m_l$ and $m_s$ are the mass of the liquid and solid regions, respectively, and $m$ is the total mass of the cell. They are related as $m = m_l + m_s$. Similarly, $V_l$ and $V_s$ are the volumes of the liquid and solid regions, respectively.

The specific heat capacity and molecular thermal conductivity are instead calculated by conventional solid fraction-weighted mixing laws [48]. The relevant equations are given by,

$$c_{p,sl} = c_{p,T_s}s + c_{p,T_l}(1 - s) \tag{3.26}$$

$$k_{m,sl} = k_{m,T_s}s + k_{m,T_l}(1 - s) \tag{3.27}$$

where $c_{p,T_s}$ and $c_{p,T_l}$ are the specific heat capacities of the solid and liquid phases at the solidus and liquidus temperatures, respectively. Likewise, $k_{m,T_s}$ and $k_{m,T_l}$ are the molecular thermal conductivities of the solid and liquid phases at the solidus and liquidus temperatures, respectively. For solid-liquid cells, the expression for enhanced thermal conductivity in Eq. (3.16) utilizes the averaged molecular thermal conductivity obtained in Eq. (3.27).

As part of the expression for enhanced thermal conductivity in Eq. (3.16), the local density gradient (and its components) must be obtained for each cell. While there are several possible ways to calculate the density gradient, the method described shortly is chosen for numerical implementation reasons that will be explained later in Section 3.5. As mentioned previously in this section, the functional forms of the density versus temperature relationships for the pure liquid and solid phases have already been obtained from literature ([46],[47]). As per these relationships, the density of the pure phases are solely functions of temperature. Consequently, based on the chain rule of differentiation, the density gradient for these phases is expressed in terms of the temperature gradient as follows

$$\nabla\rho = \frac{\partial\rho}{\partial T}\nabla T \tag{3.28}$$

Therefore, it is a simple matter of differentiating the density versus temperature relationships for the pure phases with respect to temperature to obtain the required expressions for ${\partial\rho}/{\partial T}$. In this manner, density gradients for the pure phases can be easily calculated from temperature gradients and then be used in the expression for $k_{NC}$.

Coming to the density gradient of the solid-liquid cells, based on Eq. (3.24) it initially appears that the density of solid-liquid cells is dependent on solid fraction and the expression for density gradient in Eq. (3.28) is therefore invalid for these cells. However, it must be recalled that solid fraction is itself calculated from temperature as shown in Eq. (3.23), thereby making Eq. (3.24) a function of temperature. This can be clearly seen by substituting Eq. (3.23) in Eq. (3.24), giving

$$\frac{1}{\rho_{sl}} = \left(\frac{1}{\rho_{s,T_s}}\right)\left(\frac{T - T_l}{T_s - T_l}\right) + \left(\frac{1}{\rho_{l,T_l}}\right)\left(1 - \frac{T - T_l}{T_s - T_l}\right) \tag{3.29}$$

where $\rho_{s,T_s}$ , $\rho_{l,T_l}$ , $T_s$ and $T_l$ are all constants as previously defined. The expression in Eq. (3.29) can be differentiated with respect to temperature to give the following expression for ${\partial\rho_{sl}}/{\partial T}$.

$$\frac{\partial\rho_{sl}}{\partial T} = \left(\frac{\rho_{s,T_s} - \rho_{l,T_l}}{T_s - T_l}\right)\left(\frac{\rho_{sl}^2}{\rho_{s,T_s}\,\rho_{l,T_l}}\right) \tag{3.30}$$

This expression can then be used in Eq. (3.28) to obtain the density gradient for solid-liquid cells as well, ensuring that density gradient can be obtained regardless of the phase(s) occupying a cell.

### 3.5 Solver choice and numerical implementation details

An important aspect of the reduced model is the fact that its implementation only requires the modification of the thermal conductivity and energy source term as dictated by Eqns. (3.2), (3.16) and (3.22). Any numerical solver which can solve the energy equation in Eq. (3.1) while allowing for specification of custom thermophysical properties and energy source term is compatible with the reduced model. In this manner, the reduced model is solver-agnostic in nature, i.e., it does not rely on any features or capabilities specific to a particular solver. However, the Ford Motor Company, who are the sponsors of the current work, required the use of a commercially available CFD software due to the various quality-of-life features offered by such software, such as the ability to handle complex geometries and meshes, GUI support, streamlined post-processing, etc. As ANSYS-Fluent$^{TM}$ was one such CFD software available at both Ford and OSU HPC facilities, while also offering the required functionalities, it was chosen for the purposes of solving the energy equation and implementing the reduced model.

The backward Euler (or implicit) method [37] was used for time advancement of Eq. (3.1), while the finite-volume method with a central difference scheme [37] was used for spatial discretization. Equations (3.16) (for natural convection effects) and (3.22) (for phase change effects) were incorporated using user-defined functions (UDFs) in ANSYS-Fluent$^{TM}$. The mixture thermophysical properties, as discussed in Section 3.4, were also computed using UDFs. The calculation of the density gradients, as given by Eq. (3.28), was simplified by the fact that ANSYS-Fluent$^{TM}$ provides a function that returns the values of the temperature gradient components. This was the rationale behind calculating density gradients from temperature gradients (detailed previously in Section 3.4) as this is a significantly less complex strategy compared to alternatives, such as utilizing the Gauss-divergence theorem to evaluate gradients. The solution algorithm that was employed is as follows:

1) The temperature, $T$, is initialized in all cells. This is denoted by $T_i^n$, where $i$ is the cell number, while $n$ is the time index, such that the real time is given by $t = n\Delta t$. The density and all other thermophysical properties are calculated at this temperature as described in Section 3.4 and stored.
2) The solid fraction in each cell $i$ and at time index $n$ is computed from the temperature using the relationship given by Eq. (3.23).
3) The temperature at the next time-step (time index $n + 1$) is guessed, and the solid fraction at that same time instant, $s_i^{n+1}$, is calculated. The density and all other thermophysical properties are also calculated.
4) The heat source due to phase change is computed in each cell using a time-discretized form of Eq. (3.22), namely

$$\dot{q}_{s_i} = L_s \frac{(\rho s)_i^{\ n+1} - (\rho s)_i^{\ n}}{dt} \tag{3.31}$$

   As mentioned previously, this source term is zero in regions of pure liquid or solid, and no computations are required for such cells.
5) The effective thermal conductivity is computed in each cell using Eq. (3.16). This requires computation of density gradients, which are computed using Eq. (3.28).

6) The discretized form of Eq. (3.1) is solved using an iterative linear algebraic equation solver. This task is performed by ANSYS-Fluent$^{TM}$ and yields a solution of the temperature field at the current time index, i.e., $T_i^{n+1}$ . The solid fraction and other thermophysical properties are updated using $T_i^{n+1}$.

7) The initial condition set in Step 1 is replaced by the new value, and solution proceeds to the next time-step, i.e., Steps 1 through 6 are repeated.

Equations (3.1), (3.16) and (3.22) together constitute the backbone of the proposed reduced natural convection model to account for heat transfer during solidification/melting in a partially filled tank. However, before this model can be implemented and validated, the constant $C_{NC}$ in Eq. (3.16) must be calibrated using experimental data. The following chapter first covers the details of the calibration process, which is performed using temperature data collected from a freezing experiment of a partially filled tank. Subsequently, the model is validated using temperature data from 6 different freezing experiments, comprised of 3 different fill levels and 2 different fluids, distinct from the experimental data used in the calibration process. Lastly, the chapter also details a numerical verification study of the reduced model using the in-built models provided by ANSYS Fluent$^{TM}$ for a small-scale problem involving natural convection in a differentially-heated cavity.

# CHAPTER 4. REDUCED NATURAL CONVECTION MODEL - RESULTS

## 4.1 Introduction

In the previous chapter, details were provided for the formulation and implementation of a reduced natural convection model for heat transfer during the freezing/thawing process. As a pending final step, the value of the constant $C_{NC}$ in the expression for the enhanced thermal conductivity (Eq. (3.17)) needed calibration. The current chapter first covers the freezing experiments conducted by engineers at the Ford Motor Company, where multiple sets of experimental data were collected. While these data are not the work of the current author, they are processed by the author and presented here to supplement the modelling studies, by using these data for calibration and subsequent validation of simulations employing the reduced natural convection model. Finally, the chapter also details a verification study in which solutions generated by the reduced natural convection model were compared against an exact numerical solution, for a benchmark problem involving natural convection in a differentially heated cavity.

## 4.2 Experimental Study

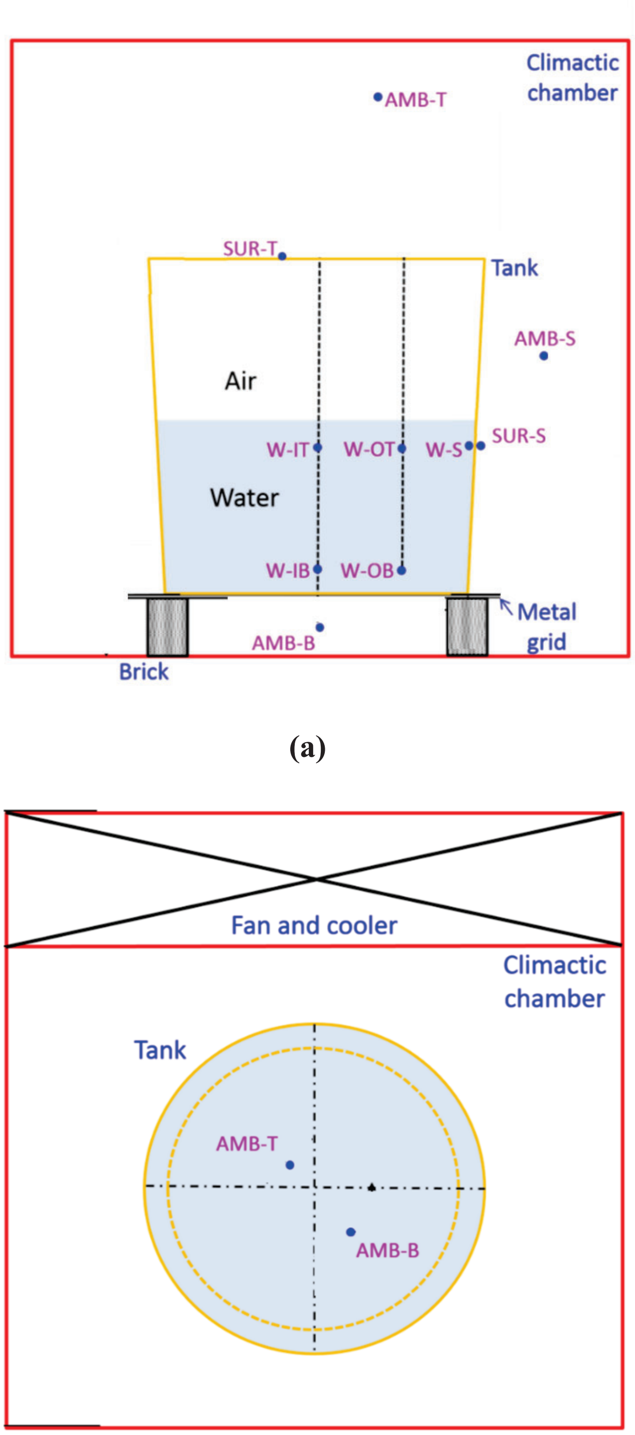
The primary goal of the experimental study was to record temperature vs. time data at various locations inside a large tank during freezing. The experiments were carried out at one of three different fill levels (by height) of the liquid: 25%, 50% and 80% fill levels, and used either water or AdBlue® as the working liquid, resulting in six distinct data sets.

Additionally, an independent set of data was collected for calibration purposes by performing an experiment using a 50% filled tank of water. This data set was used solely for calibration purposes, while data from the other experiments were used for validation studies. In total, the experimental study collected seven data sets—one data set for calibration and six data sets for experimental validation of the model. Table 4.1 summarizes all the experiments that were performed.

As a similar procedure was employed for all experiment runs, for the sake of brevity, a detailed procedure is provided only for the Experiment Run 1 (calibration set). Figures 4.1(a) and (b) show schematics of the front and top views of the experimental setup for the 50% fill level of water with the tank placed inside a climactic chamber. A near-cylindrical tank with a slightly larger lid diameter was used and was placed on a simple stand consisting of a metal grid supported by bricks. The exact same setup (*i.e.*, tank and chamber dimensions) was used for all experimental runs, with the only differences being the locations of the thermocouples inside the tank based on the fill level. Schematics for the other experimental runs are provided later in Section 4.3. As seen in Figures 4.1(a) and (b), the climactic chamber was cooled with the help of a fan/cooler on one of its side walls, generating unsteady forced convection patterns within the chamber. For this experimental run, five thermocouples, designated W-OB (water outer bottom), W-OT (water outer top), W-IB (water inner bottom), W-IT (water inner top) and W-S (water side) were placed at various points throughout the water domain, collectively referred to as the water thermocouples. All of these thermocouples except the W-S thermocouple were attached to plastic rods suspended from the lid of the tank. The W-S thermocouple was directly

**Table 4.1: List of all experimental runs performed with different combinations of fill level and working liquid.**

| Experiment Run # | Liquid | Fill level | Purpose |
|---|---|---|---|
| **1** | Water | 50% | Calibration |
| **2** | Water | 25% | Validation |
| **3** | Water | 50% | Validation<br>(different from Run #1) |
| **4** | Water | 80% | Validation |
| **5** | AdBlue® | 25% | Validation |
| **6** | AdBlue® | 50% | Validation |
| **7** | AdBlue® | 80% | Validation |



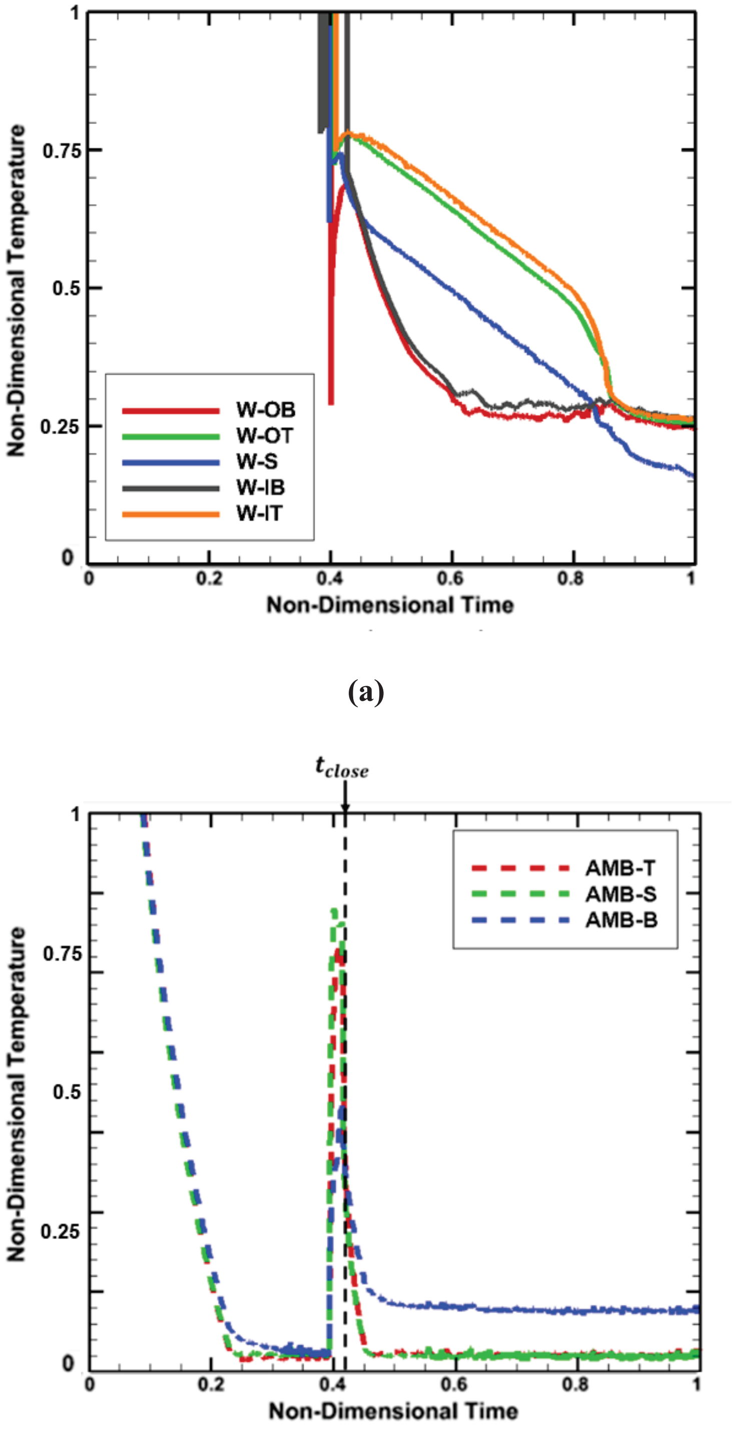
(a)

(b)

**Figure 4.1: Schematic of (a) front view, and (b) top view of the experimental setup for the 50% filled tank.**

attached to the inner surface of the tank. To monitor ambient temperatures in the chamber, thermocouples AMB-T (ambient top), AMB-S (ambient side) and AMB-B (ambient bottom) were placed at locations above, to the side, and below the tank, respectively. Finally, thermocouples SUR-T (surface top) and SUR-S (surface side) were added to monitor changes in surface temperature on the top and outer side surfaces of the tank.

The experiment was started by first pre-cooling the empty climactic chamber (with only the stand in place) to a temperature of $T_{amb} = -38$°C, while the water-filled tank was kept at ∼21°C outside the chamber. All thermocouples (K-type) were fitted to the tank before it was brought inside the chamber. The tank was then placed on the stand and the thermocouple leads were quickly connected to a Data Acquisition (DAQ) system. This process required about 4-5 minutes, and upon conclusion, marked the start of the experiment. Temperatures at all thermocouple locations were then recorded every 6 seconds until the freezing of the entire tank was complete, which, in some cases, required more than 24 hours. Figure 4.2(a) shows a plot of the temperature-time curves for the water thermocouples from when the tank is placed within the chamber, while Figure 4.2(b) plots the temperature-time curves for the ambient thermocouples, including the portion of time when chamber was pre-cooled. As seen in Figure 4.2(b), ambient temperatures were seen to suddenly rise around 4000 seconds, which represents the moment when the chamber's door was opened to place the tank inside. Around 4300 seconds, ambient temperatures were seen to drop, indicating the time at which the chamber's door was closed. This time instant (shown as $t_{close}$ in Figure 4.2(b)) marked the beginning of the run.



(a)

(b)

**Figure 4.2: Temperature vs. time data for freezing experiment run 1 involving a 50% filled tank of water: (a) water thermocouples, (b) ambient thermocouples.**

The exact same procedure was applied to all other experimental runs. Furthermore, each experiment was conducted twice to ensure repeatability of data. To illustrate, Figure 4.3 shows the temperature-time curves for the AMB-B and AMB-S thermocouples from both runs of this experiment. From the figure, the stabilized ambient temperature values from both runs were seen to be within a 2°C margin, lending credibility to repeatability of the data.

Figure 4.4(a) shows the experimental temperature vs. time data for the five water thermocouples, while Figure 4.4(b) shows the temperature vs. time data for the ambient thermocouples. The data in these figures is plotted from the instant when the chamber's door was closed *i.e.*, $t_{close}$ from Figure 4.2(b). As seen in Figure 4.4(a), the total time for the tank to reach steady state was approximately 23 hours. From a validation perspective, the most important information obtained from this plot was the freezing sequence of the thermocouples. This was because one the major goals of the validation study was for the simulation results to correctly replicate the freezing sequence observed in the experiments. Completion of freezing is marked by the temperature at a thermocouple dropping below the freezing point, 0°C in the case of water. For this particular case, the freezing sequence was, therefore, noted as: (1) W-S → (2) W-OB (3) W-IB → (4) W-OT → (5) W-IT. The sequence follows logical sense for the experimental run depicted in Fig. 4.1(a), where the thermocouple on the surface of the tank wall was the first to freeze followed shortly by the ones near the bottom wall. The ones near the water surface freeze after this, with the one near the center of the tank being the last to freeze. While the other experimental runs

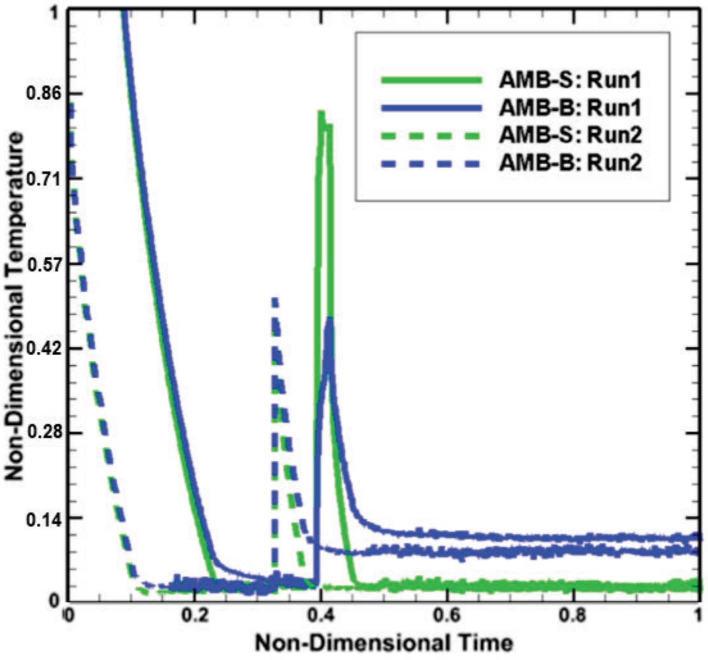
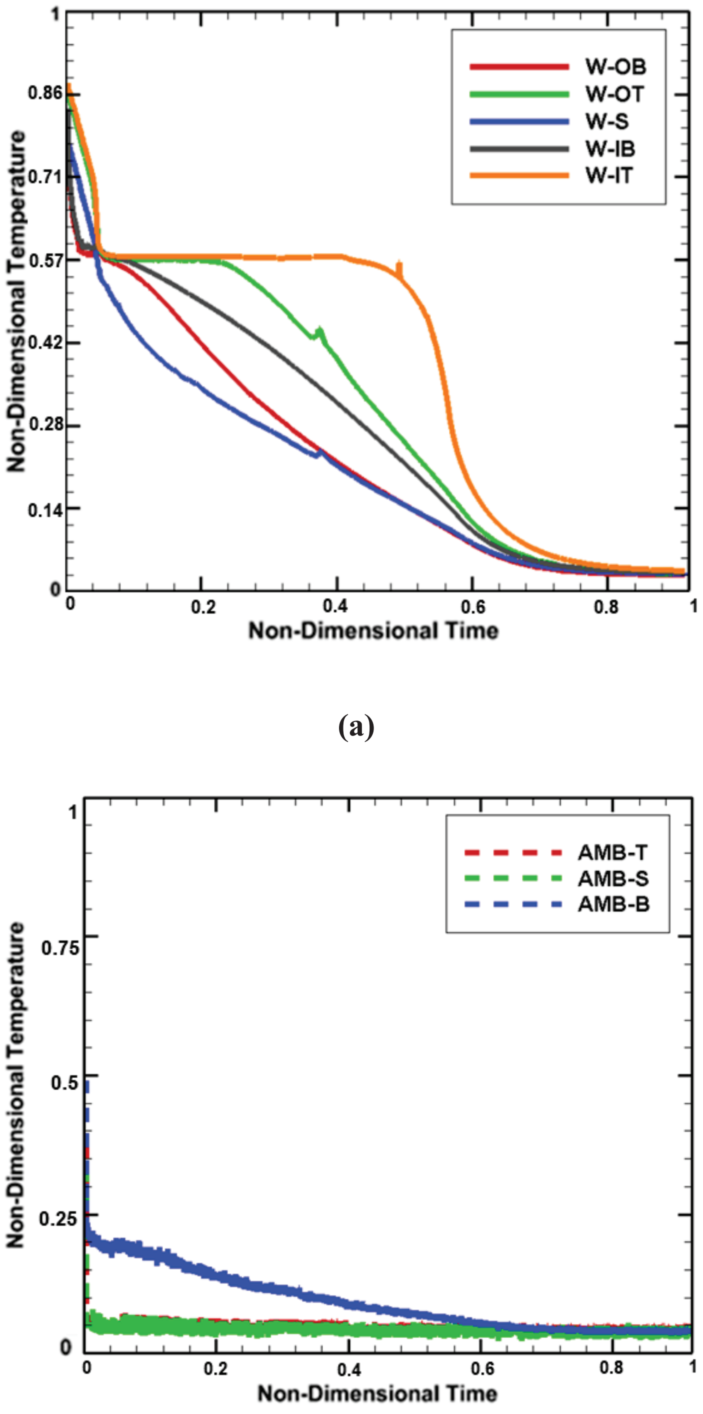


**Figure 4.3: Temperature vs. time data for two ambient thermocouples from two different runs of the experiment**



**(a)**

**(b)**

**Figure 4.4: Temperature vs. time curves from the freezing experiment of a 50% filled tank of water: (a) water thermocouples, (b) ambient thermocouples.**

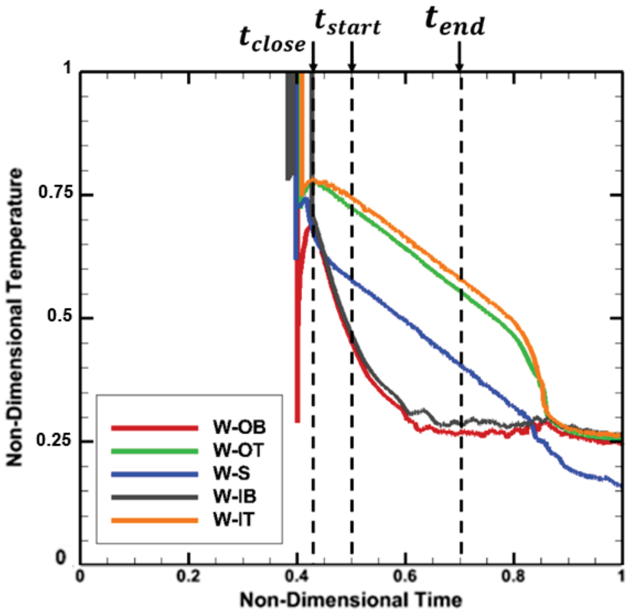
resulted in similar freezing sequences, details of these runs are provided in Section 4.7, during the discussion covering the validation of the simulation results.

Prior to performing validation studies, it was necessary to determine an appropriate initial condition for each simulation, as well as external heat transfer coefficients, which are required for the convective boundary condition at the walls. These two tasks were performed by Ramesh, *et al.* [11],[49] during a previous phase of the project that the current work continued from. While a detailed description of the process involved in these tasks can be found in the listed references, a summary is provided in Sections 4.3 and 4.4 as it provides the necessary context for the remaining discussion in this chapter.

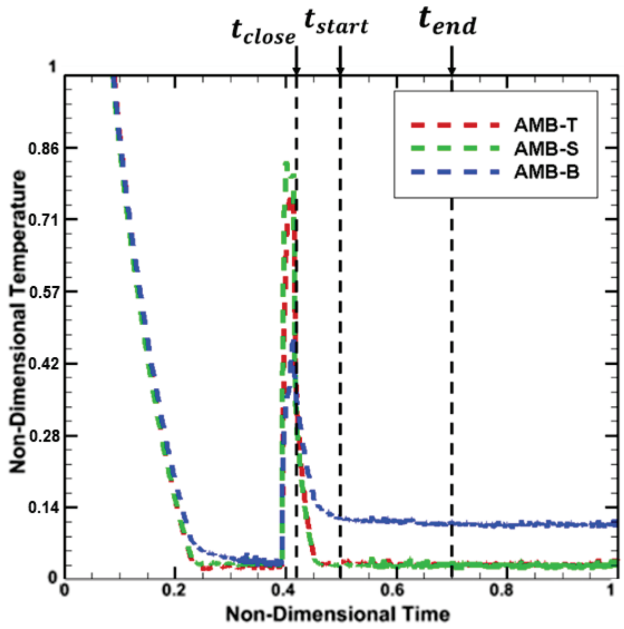
### 4.3 Calculation of Initial Condition Profile

The setting of an appropriate initial condition for the simulations was complicated by the fact that the climatic chamber's ambient conditions were perturbed when the tank was placed in the chamber during the experimental procedure. This can be seen in Fig. 4.5(b), where the closing of the climactic chamber is marked as $t_{close}$. For a short period of time after the chamber is closed, the ambient temperature within the chamber keeps decreasing. This meant that a convective boundary condition would not be applicable at the walls of the tank during this period due to the varying ambient temperature (Note: the definition of the heat transfer coefficient stipulates that the ambient temperature is a constant). Therefore, when simulating this experiment, the initial condition for temperature within the tank had to be prescribed at a time instance past $t_{close}$ when the ambient temperature had settled down to a near-constant value. In Fig. 4.9(b), this time instance is denoted as $t_{start}$ and is $\sim$700 seconds past $t_{close}$. However, during the time taken for the ambient temperature to return to a constant value, the tank would start cooling and develop a temperature distribution. This can be seen in Fig. 4.5(a), where during the time between $t_{close}$ and $t_{start}$, the temperature at the water thermocouples started dropping and, more importantly, were non-uniform. While these thermocouples do provide temperature data at several discrete locations within the tank, this data is insufficient for the initialization of a three-dimensional temperature distribution withing the tank. Therefore, an alternative strategy had to be utilized.

To circumvent this issue, a simulation was setup for the time period between $t_{close}$ and $t_{start}$. In this simulation, the natural convection in both water and air inside the tank was modeled using the original governing equations of fluid flow and heat transfer (i.e., full-blown CFD/CHT calculations) rather than the reduced natural convection model discussed in Chapter 3. The ambient temperature at the walls was fixed to a constant value and three fictitious external heat transfer coefficient values were used for top, side, and bottom wall ($h_{top}, h_{side}, h_{bot}$). These fictitious $h$ values were adjusted until the simulated temperatures at the water thermocouples were within 2°C of the experimental equivalents. Once this was achieved, the temperature profile within the tank was saved to be then used as an initial temperature distribution that closely matched experimental data.



(a)



(b)

**Figure 4.5: Temperature vs. time curves upon commencement of the freezing experiment of a 50% filled tank of water: (a) water thermocouples, (b) ambient thermocouples.**

### 4.4 Calibration of external heat transfer coefficients

As mentioned in the previous section, the climactic chamber enforced a convective boundary condition at the walls of tank, which required appropriate external heat transfer coefficient ($h$) values to be set at the walls. However, the conditions inside the climatic chamber are quite complex due to combined natural and forced convection, and using heat transfer correlations from the literature to estimate $h$ was, therefore, deemed unreliable. Instead, it was decided to calibrate the heat transfer coefficients. It was assumed that only three constant heat transfer coefficient values would be sufficient to replicate the experimental data: $h_{top}, h_{side}$ and $h_{bot}$, representing heat transfer coefficients on the top, side and bottom surfaces of the tank, respectively. The calibration was performed by simulating the time period between $t_{start}$ and $t_{end}$ in Fig. 4.5(b), a period of approximately 2000s. The ambient temperatures at each surface were assumed to be constant during this time, with the values obtained from the temperature data in Fig. 4.5(b). During this calibration phase, the natural convection in both water and air inside the tank was once again modeled using the original governing equations of fluid flow and heat transfer rather than the reduced model. The final outcome of this study were the following values: ($h_{top}, h_{side}$ and $h_{bot}$ ) = (4, 35, 30 W/m$^2$-K). These $h$ values were used in all subsequent simulations.

### 4.5 Simulation setup details

The calibration of the reduced model constant (to be covered in Section 4.6) and the subsequent validation studies (Section 4.7) were all performed using 3D simulations set up in ANSYS-Fluent$^{TM}$. However, unlike the simulations performed in Sections 4.3 and 4.4, these simulations utilized the reduced natural convection model. This meant that the simulations only used ANSYS-Fluent$^{TM}$'s energy solver and did not solve for flow, i.e., they were pure conduction simulations.

Importantly, the simulations covered in this chapter do not consider the movement of the gas-(solid/liquid) interface due to volume expansion. Consequently, the location of the gas-(solid/liquid) interface is fixed. To model the interface, a thermally coupled thin-wall boundary condition was used with the wall thickness set to zero. Simulations capable of tracking the movement of this interface are covered in Chapter 6.

The details pertaining to the thermophysical properties of the materials involved in the various simulations are detailed below. All of the material thermophysical properties were implemented using custom User-Defined Functions (UDFs) in ANSYS-Fluent$^{TM}$. Following this, Section 4.5.5 covers the boundary conditions employed for the simulations. Section 4.5.6 details how the solver was set up in ANSYS-Fluent$^{TM}$. Finally, the meshing of the tank and a corresponding grid independence study is detailed in Section 4.5.7.

### 4.5.1 Gas - Air Properties

The details covering the methods and assumptions involved in setting the thermophysical properties of air, as used in the current simulations, are listed below. The numerical values of these properties, alongside the properties of the other materials used in these simulations, are listed in Table 4.2.

- Air inside the tank is treated as an ideal gas. Its density is described by the ideal gas law at atmospheric pressure.

**Table 4.2: Thermophysical properties of materials used in the study.**

| Material | Density ($kg/m^3$) | Specific Heat Capacity (J/kg-K) | Molecular Thermal Conductivity (W/m-K) | Enhanced Thermal Conductivity (W/m-K) |
|---|---|---|---|---|
| **Air** | Ideal gas law | 1006.43 | 0.02 | Eq. (3.17) |
| **Water - Liquid** | Ref. [46] | 4182 | 0.6 | Eq. (3.17) |
| **Water - Solid** | Ref. [47] | 2092.68 | 2.4 | N/A |
| **AdBlue® - Liquid** | Ref. [50] | 3405.77 | 0.57 | Eq. (3.17) |
| **AdBlue® - Solid** | 1010 | 1600 | 2.09 | N/A |
| **Solid-Liquid Cells** | Eq. (3.25) | Eq. (3.27) | N/A | Eq. (3.28) |
| **Plastic (HDPE)** | 932 | 1900 | 0.5 | N/A |

- The specific heat capacity of air was assumed to be constant, i.e., independent of temperature.
- The molecular thermal conductivity of air was also assumed to be independent of temperature.
- The effective thermal conductivity of air was obtained using the reduced natural convection model, i.e., Eq. (3.17).

### 4.5.2 Working Liquid I – Water Properties

The details covering the methods and assumptions involved in setting the thermophysical and solidification properties of liquid water and solid ice, as used in the current simulations, are summarized below. The thermophysical properties of water are listed in Table 4.2, while the solidification properties are listed in Table 4.3.

- The densities of both water and ice were considered to be temperature dependent. The functional forms for these relationships were obtained from literature, specifically Ref. [46] and [47]. Importantly, the relation provided for the density of liquid water does capture the density inversion behavior at 4°C, as seen in Fig. 2.5.
- The density for mixed solid-liquid cells was calculated using Eq. (3.25).
- The specific heat capacities of both liquid water and solid ice were assumed to be independent of temperature. For mixed solid-liquid cells, the specific heat capacity was determined using Eq. (3.27), a solid fraction-weighted mixing law.
- The thermal conductivities of both liquid water and solid ice were also assumed to be independent of temperature.

**Table 4.3: Phase change properties of materials used in the study.**

| Material | Solidus Temperature (°C) | Liquidus Temperature (°C) | Latent Heat of Solidification (J/kg) |
|---|---|---|---|
| **Water** | 0 | 0.05 | 333,550 |
| **AdBlue®** | -11.2 | -11.15 | 270,000 |

- The effective thermal conductivity for pure liquid water was obtained using Eq. (3.17) of the reduced natural convection model. As there is no natural convection in pure solids, there was no thermal conductivity enhancement in solid ice.
- The effective thermal conductivity for solid-liquid cells was obtained using Eq. (3.28), a solid fraction-weighted mixing law.
- The latent heat release during the solidification process was modelled using a volumetric source term, as given by Eq. (3.23).

### 4.5.3 Working Liquid II – AdBlue® Properties

While water is an extensively studied material across various fields with its properties well-documented, there are limited sources detailing the properties of AdBlue®. Most of the properties listed below were obtained from a technical leaflet for AdBlue® provided by BASF [50]. Once again, the thermophysical properties of AdBlue® are listed in Table 4.2, while the phase change properties are listed in Table 4.3.

- The density of liquid AdBlue® was given to be temperature dependent and the expression for the same was obtained from Ref. [50]. The same source also specified the density of solid AdBlue® as a constant with respect to temperature. It is important to note that AdBlue® does not display the density inversion behavior of water. However, the density of solid AdBlue® is lower than that of liquid AdBlue®, just as for water.
- The specific heat capacities of both solid and liquid phases were assumed to be temperature independent.

- The thermal conductivity of liquid AdBlue® was assumed to be constant with respect to temperature.
- The thermal conductivity for solid AdBlue® was not available in literature. Instead, as it is known that AdBlue® is a 32.5% urea solution (by weight) [2], the thermal conductivity of the solid phase was estimated as per a mass-weighted mixing law using the conductivities of the individual components of AdBlue®. From literature, it was obtained that $k_{m,solid\ urea}$ = 1.46 W/m-K [51], and $k_{m,ice}$ = 2.4 W/m-K. Subsequently, the thermal conductivity of solid AdBlue® was estimated using the following relation.

  $$k_{m,solid\ AdBlue} = k_{m,solid\ urea} m_{solid\ urea} + k_{m,ice} m_{ice} \tag{4.32}$$

  where $m_{solid\ urea}$ and $m_{ice}$ are the mass fractions of pure urea and ice in AdBlue®, respectively. The resulting value for thermal conductivity can be found in Table 4.2.
- The density, specific heat capacity and thermal conductivity for mixed solid-liquid cells were calculated using Eqns. (3.25), (3.27) and (3.28) respectively, just as was done in the case of water.
- The latent heat release during the solidification process was also modelled in the same manner as was done for water, using Eq. (3.23).

### 4.5.4 Tank Wall Properties

The tank walls were made of high density polyethylene (HDPE). The material properties of HDPE were considered to be temperature independent and are listed in Table 4.2. All tank walls were assumed to be perfectly smooth (no roughness).

### 4.5.5 Boundary Conditions

As discussed in Section 4.4, a convective heat transfer boundary condition is employed at the top, side and bottom walls. To this end, heat transfer coefficients were calibrated at each of these walls: ($h_{top}, h_{side}$ and $h_{bot}$ ) = (4, 35, 30 W/m$^2$-K). These heat transfer coefficients remained unchanged between all the various simulations. The ambient temperature at each of the walls is obtained from a time-averaged value of measured ambient temperature data over the entire experimental duration. While the exact values vary between experimental runs, the values were found to be approximately -38°C for the top and side walls, and -36°C for the bottom wall.

### 4.5.6 Solver Setup

As mentioned at the start of Section 4.5, only the energy equation is solved in these simulations, with the reduced natural convection model employed to account for the contribution of natural convection in the heat transfer process. The energy equation is solved in all simulations using the energy equation solver provided by ANSYS-Fluent$^{TM}$. As the energy equation is non-linear (due to temperature dependent properties such as density), it often requires the use of under-relaxation factors. For the current simulations, both energy and density used under-relaxation factors of 0.5. Spatial discretization of the energy equation was performed using the finite volume method with a central difference scheme, while a first-order accurate implicit (backward Euler) scheme was used for the time advancement. A relative convergence criterion of 0.01 and an absolute convergence criterion of $10^{-9}$ were used to monitor convergence of the energy equation. A time-step size of 1 second was used in all simulations. While the experiment study collected temperature data every 6 seconds, this lower time-step size was chosen to ensure time-step independent results. The simulations were run using 28 processors in parallel on a Dell Intel Xeon E5-2680 v4 cluster at the Ohio Supercomputer Center (OSC).

### 4.5.7 Meshing and Grid Independence

The 3D tank was meshed using the mesh generation package ICEM CFD$^{TM}$. As part of grid independence study, three different hexahedral meshes with cell counts of approximately 0.5 million, 1 million, and 2 million cells were generated for a 50% fill level case. Figure 4.6 shows a comparison of the temperature vs. time history at the W-IB and W-OB thermocouple locations for the 50% fill level case with water.

As demonstrated by the plot, the results obtained when using the two finer meshes are quite similar. The RMS error between the 0.5 million mesh and the 2 million mesh was found to be 0.95°C and 0.78°C for W-IB and W-OB, respectively. In contrast, the RMS error between the 1 million mesh and the 2 million mesh was found to be 0.47°C and 0.39°C for W-IB and W-OB, respectively. Considering these results, the decision was made to use a 2 million hexahedral cell mesh for all studies. The mesh for the 50% fill level cases is shown in Fig. 4.7.

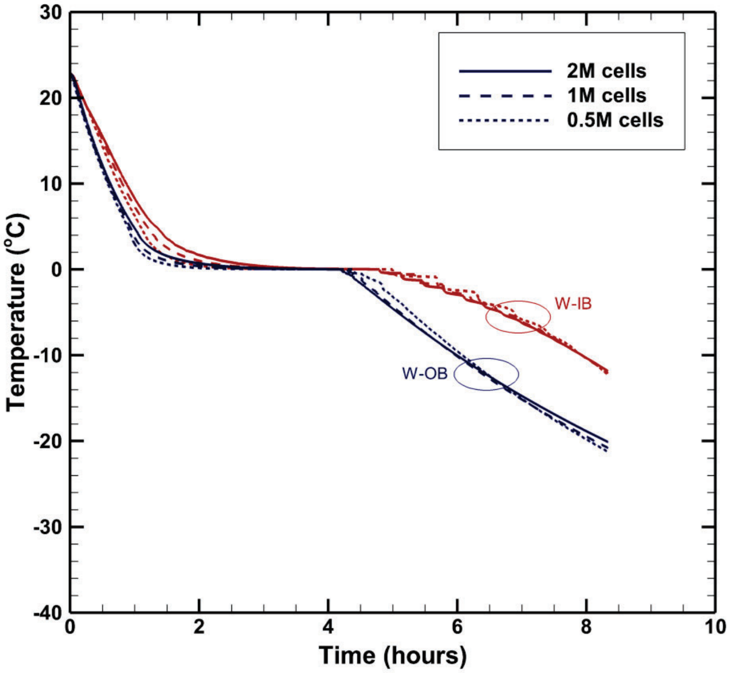


**Figure 4.6: Effect of mesh size on computed temperature vs. time data for the 50% fill level case of water at two different thermocouple locations.**

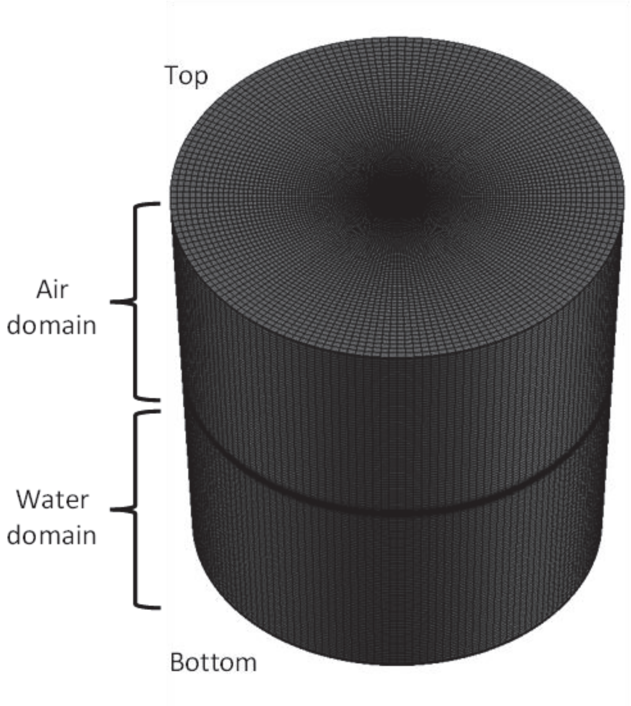


**Figure 4.7: 2 million hexahedral cell mesh used for 50% fill level cases.**

### 4.6 Calibration of reduced natural convection model constant

As discussed previously at the end of Section 3.2, the final expression for the effective thermal conductivity in the reduced natural convection model (Eq. 3.17) had an undetermined constant, $C_{NC}$, which had to be calibrated using experimental data. Rather than calibrating $C_{NC}$ directly, a new constant defined as $C_{NC}(\alpha\nu)^{-0.275}$ was calibrated instead. As this new constant is dependent on thermophysical properties, it has different values for different materials. Thus, these new constants are given by

$$c_w = C_{NC}(\alpha\nu)_{\mathrm{w}}^{-0.275} \tag{4.1}$$

$$c_a = C_{NC}(\alpha\nu)_{a}^{-0.275} \tag{4.2}$$

$$c_u = C_{NC}(\alpha\nu)_{u}^{-0.275} \tag{4.3}$$

where $c_w$, $c_a$ and $c_u$ are the calibration constants for water, air and AdBlue® (urea solution) respectively.

During the calibration process, only $c_w$ was calibrated as the other two constants can be obtained from $c_w$ in the following manner using Eqns. (4.1) – (4.3).

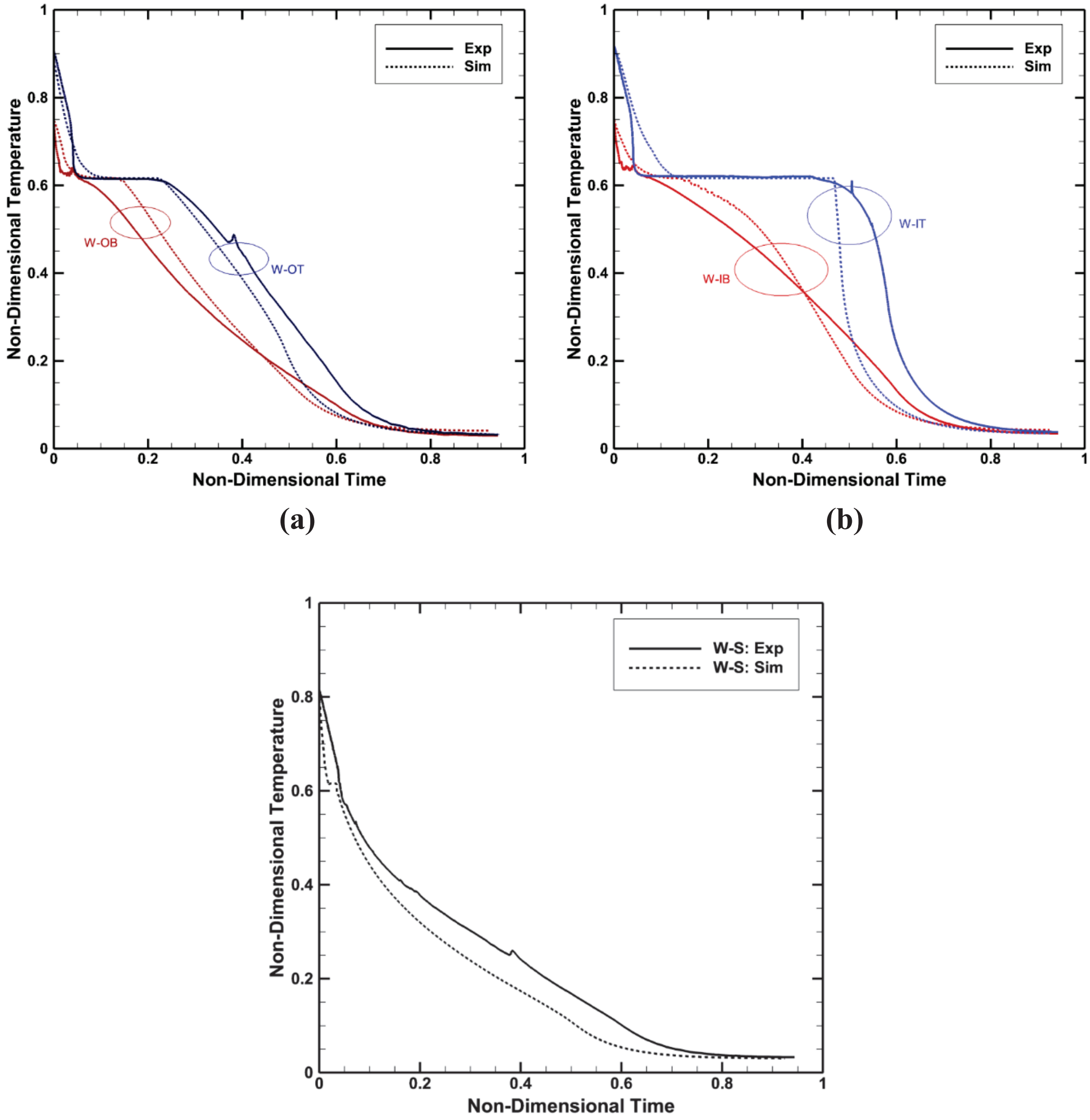
$$c_a = c_w \left[\frac{(\alpha\nu)_a}{(\alpha\nu)_w}\right]^{0.275} \tag{4.4}$$

$$c_u = c_w \left[\frac{(\alpha\nu)_u}{(\alpha\nu)_w}\right]^{0.275} \tag{4.5}$$

For the calibration process, a simulation utilizing the reduced model was set up in ANSYS-Fluent$^{\mathrm{TM}}$ as detailed in Section 4.5. As highlighted in Table 4.1, the calibration process was conducted using the data from experimental run 1, which used water as the working liquid at a 50% fill level. The calibration process began by setting $c_w$ = 500 and then increasing it by one order of magnitude in subsequent runs. The accuracy of the calibration was assessed by comparing the root-mean-square (RMS) error between the simulated and measured temperature data at all water thermocouple locations over the entire duration of the experimental run. The expression for the RMS error is given by

$$Error_{RMS} = \sqrt{\frac{1}{N}\left(\sum_{n=1}^{N}(T_n^s - T_n^e)^2\right)} \tag{4.6}$$

where $T_n^s$ and $T_n^e$ denote the temperatures of the simulation and experiment at the $n$th time step, respectively, and $N$ denotes the total number of time steps. The value of $c_w$ was adjusted until this error was minimized. Figure 4.8 shows a comparison between the experimental data and the simulated results for the five water thermocouples at a $c_w$ value of 500,000. Given the complexity of the problem, the agreement between the two data sets is quite reasonable. For some locations, the predicted onset of freezing is slightly earlier, while for others it is slightly later. Nonetheless, the sequence of freezing, as observed in the experiments, is correctly replicated: W-S → W-OB → W-IB → W-OT → W-IT. The RMS errors are shown in Table 4.4. The largest error observed is 7.44°C for the inner top thermocouple (W-IT), which is the last location in the freezing sequence. Simulation results predict that this thermocouple freezes earlier than the experimental freezing time, which ultimately leads to the large RMS error. Efforts to further reduce this error by fine-tuning the value of $c_w$ further were unsuccessful, as it was found that errors at some thermocouple locations increased while others decreased. Therefore, $c_w$ = 500,000 was considered the optimum value for the model constant and was used in all subsequent studies. This value of $c_w$ resulted in values of $c_a$ = 118,580 and $c_u$ = 460,778, as per Eqns. (4.4) and (4.5), respectively.



(a)

(b)

(c)

**Figure 4.8: Temperature vs. time data using $c_w$ = 500,000 and $c_a$ = 118,580 during calibration: (a) outer thermocouples, (b) inner thermocouples, (c) side wall thermocouple.**

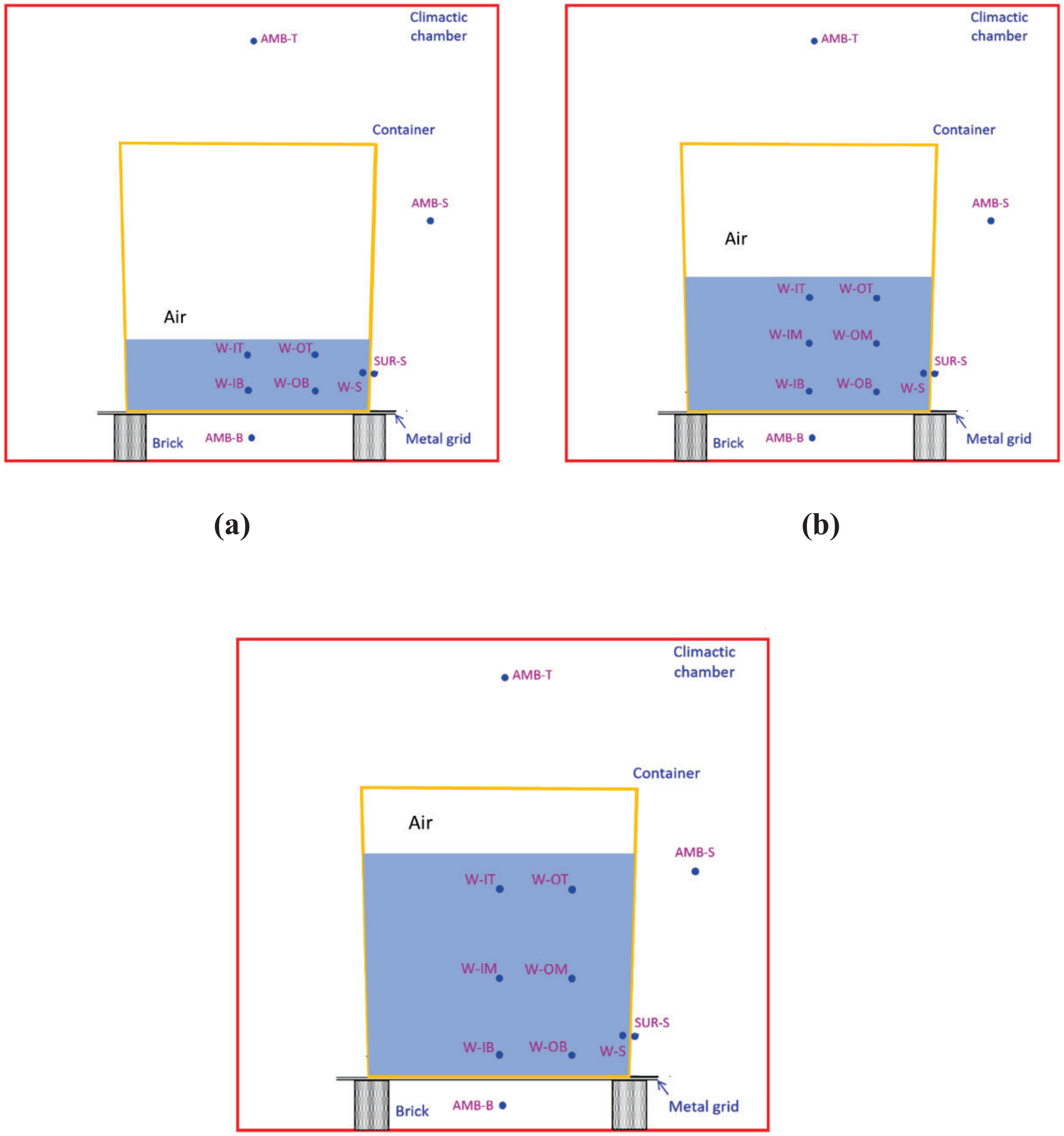
**Table 4.4: Average (RMS) error between the simulation and experiment for all thermocouple locations during reduced model calibration.**

| Thermocouple | RMS error (°C) |
|---|---|
| W-OB | 2.24 |
| W-OT | 2.88 |
| W-S | 3.14 |
| W-IB | 2.47 |
| W-IT | 7.44 |

The computational time required for this simulation was of particular interest. As discussed earlier, reduction in run times for freezing simulations of this nature was the primary motivation behind the development of the reduced natural convection model. The simulations were performed on a Dell Intel Xeon E5-2680 v4 cluster at the Ohio Supercomputer Center (OSC). The wall-clock time to simulate 23 hours of the experimental time was approximately 130 hours using 28 processors. This was a significant improvement from an efficiency standpoint when compared to the inbuilt physical models of ANSYS Fluent$^{TM}$, as the computational time for a simulation using the inbuilt models was estimated to be more than a year.

### 4.7 Validation Study

In the experimental studies detailed earlier in Section 4.2, six experiment runs were conducted for the purposes of validation of the reduced natural convection model (experiment runs 2-6 in Table 4.1). A combination of three different fill levels: 25%, 50% and 80% fill levels, along with two working liquids: water and AdBlue®, were used to perform the freezing experiments. Figure 4.9 shows the schematics of the experimental setups for the 25%, 50% and 80% fill levels. For the 50% and 80% fill level validation runs, two additional thermocouples designated W-OM (water outer middle) and W-IM (water inner middle) were added to the working fluid domain. The heights of the bottom thermocouples, W-OB and W-IB, were consistent across the three fill levels. In a similar fashion, thermocouples W-OT and W-IT were maintained at the same vertical distances from the free surface for all three setups. The procedure for each experiment was identical to that performed for the experimental run involving a 50% filled tank of water, as



(a)

(b)

(c)

**Figure 4.9: Schematic of the front view of the experimental setup of the tank at (a) 25% fill level (b) 50% fill level (c) 80% fill level.**

described earlier in Section 4.2. While the designation of the thermocouples references “water”, the exact same thermocouple positions are used for both water and AdBlue® setups. The “water” designation of the thermocouples is simply intended to convey that the thermocouples are placed within the volume of the working liquid.

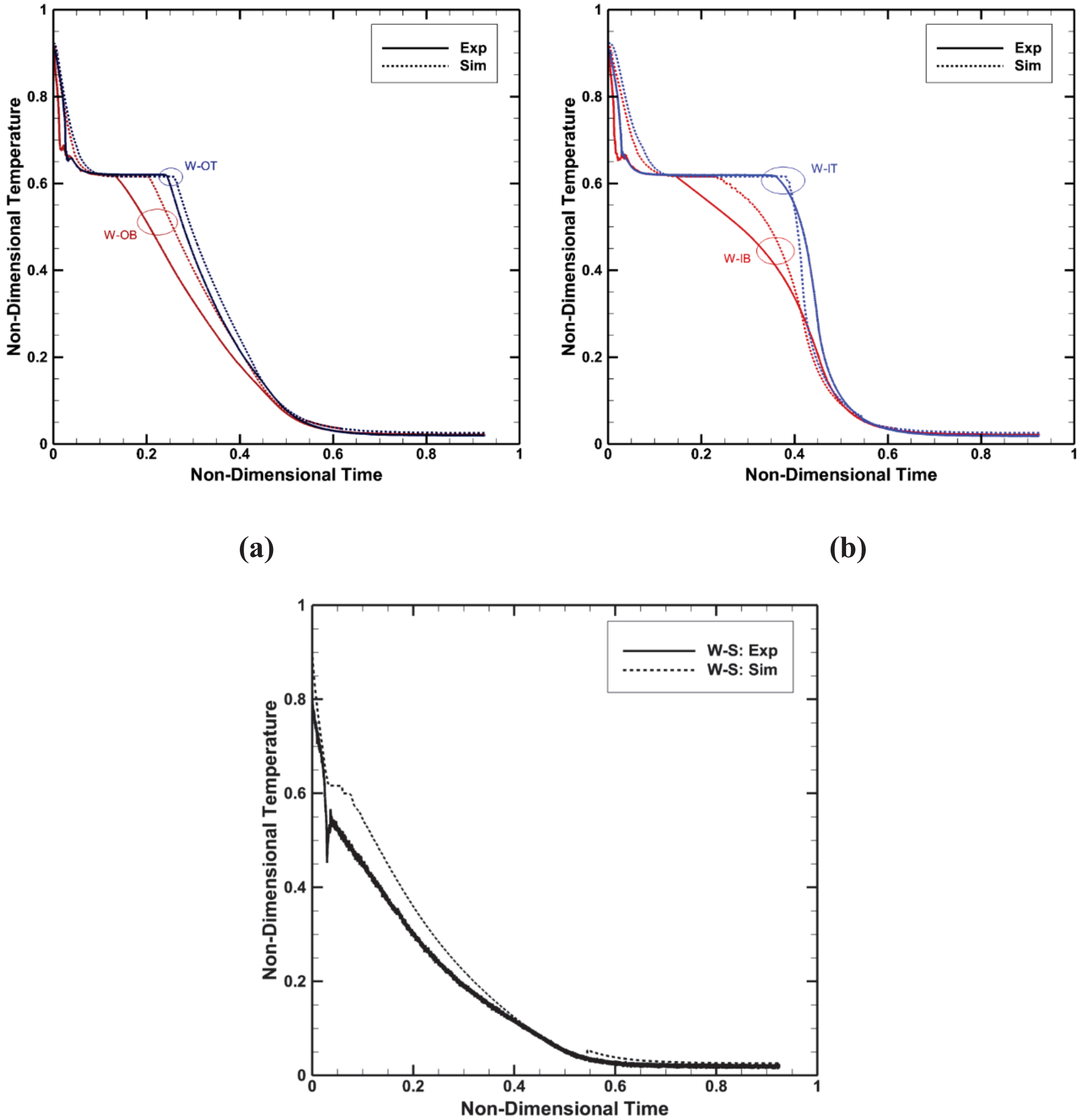
### 4.7.1 Results with Water as Working Liquid

All simulation results for water were generated using the previously calibrated $c$ values of $c_w$ = 500,000 and $c_a$ = 118,580, as detailed in Section 4.6. Calibration was performed only once, using the experimental data set designated for calibration. No further calibration was performed for the validation cases, including those of differing fill levels and working liquid. The RMS errors for the thermocouples are summarized in Table 4.5.

Figure 4.10 shows a comparison between the simulated and experimental temperature vs. time curves at the five water thermocouples for the 25% fill level case. From the temperature data, it can be seen that the time taken for the system to reach steady state was approximately 13 hours. The trends for all thermocouples are fairly consistent with the experiment. In particular, the W-OT and W-IT thermocouples show an excellent match with the experimental data. Furthermore, the agreement between predicted and measured results is found to be better than even the calibrated case, with the maximum RMS error being only 3.00°C. This is explained by the fact that natural convection is weakest in the 25% fill level case due to the lower global length scales (height) involved. Therefore, any limitations of the reduced model do not significantly affect the results. The freezing sequence for the thermocouples predicted by the simulation, noted as: W-S → W-OB → W-IB → W-OT → W-IT, is also consistent with that seen in the experiment. The

**Table 4.5: Average (RMS) error (°C) between the simulation and experiment for all three fill levels of water.**

| Thermocouple | RMS error | | |
|---|---|---|---|
| | 25% fill level | 50% fill level | 80% fill level |
| W-OB | 2.87 | 4.89 | 2.47 |
| W-OM | - | 2.25 | 2.77 |
| W-OT | 1.50 | 2.25 | 5.62 |
| W-S | 3.00 | 3.48 | 3.25 |
| W-IB | 2.87 | 3.45 | 2.25 |
| W-IM | - | 3.39 | 6.78 |
| W-IT | 2.76 | 8.21 | 10.45 |



**(a)**

**(b)**

**(c)**

**Figure 4.10: Temperature vs. time curves for the 25% filled tank of water: (a) outer thermocouples, (b) inner thermocouples, (c) side wall thermocouple.**

wall-clock time for this stimulation was approximately 120 hours.

The propagation of the solidification front for the 25% fill level case is depicted in Fig. 4.11 using snapshots of the solid fraction contours along a central plane of the tank at various instances of time. The solidification starts from the sides and bottom of the tank. With time, the surface of the water starts to freeze, with a sharp freezing line penetrating inward from the sides – henceforth, referred to as *notching*. Similar behavior was also observed in the CFD/CHT results (Fig. 2.7) in the study undertaken in Chapter 2. The water bubble has an oblong shape, where the diameter of the bubble is larger than its height for most of the freezing process. which is similar to the shape of the original volume of the liquid. This suggests that most of the heat transfer (loss) from the system occurs at the bottom wall, rather than the side wall, due to its relatively larger surface area.

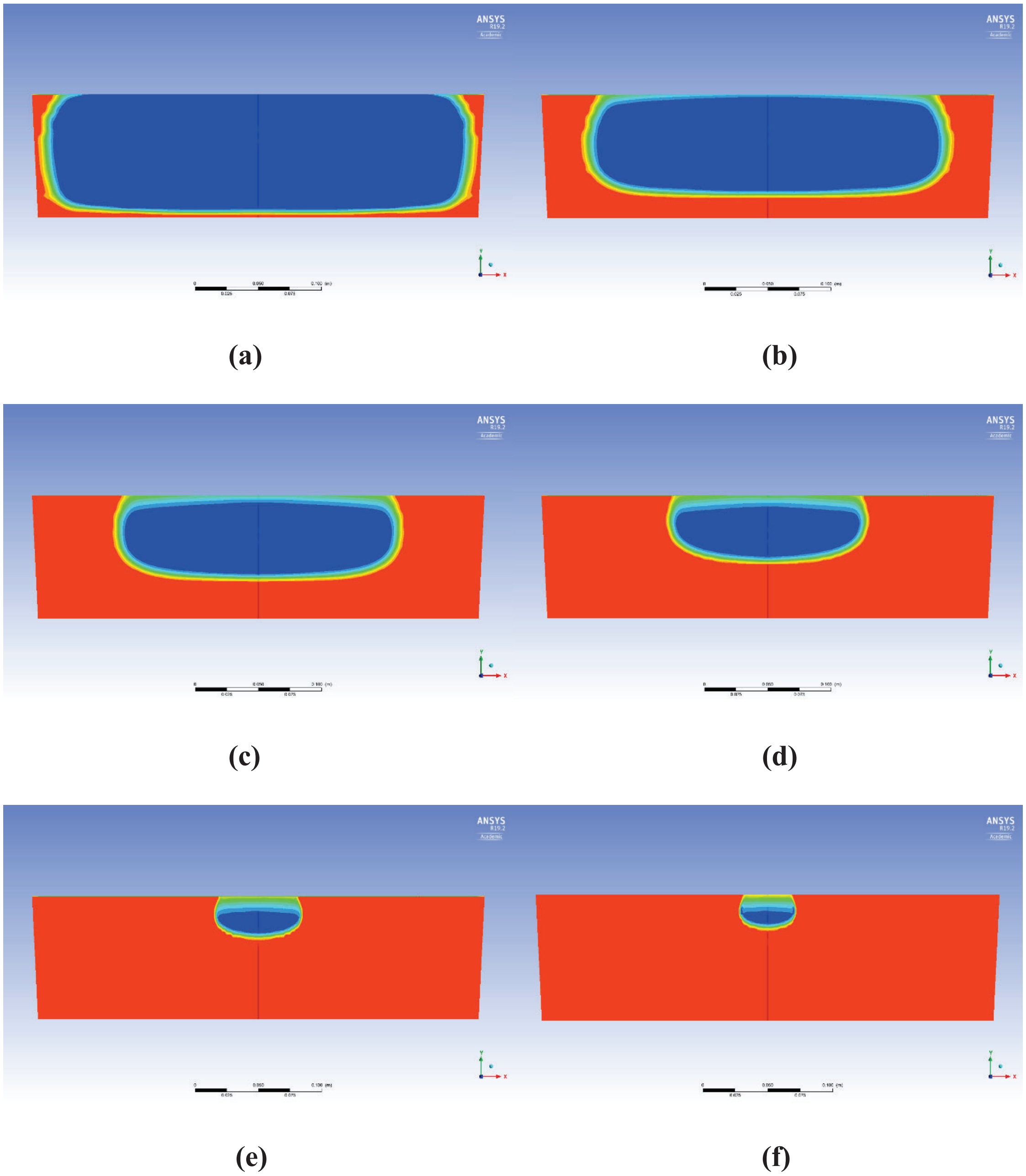
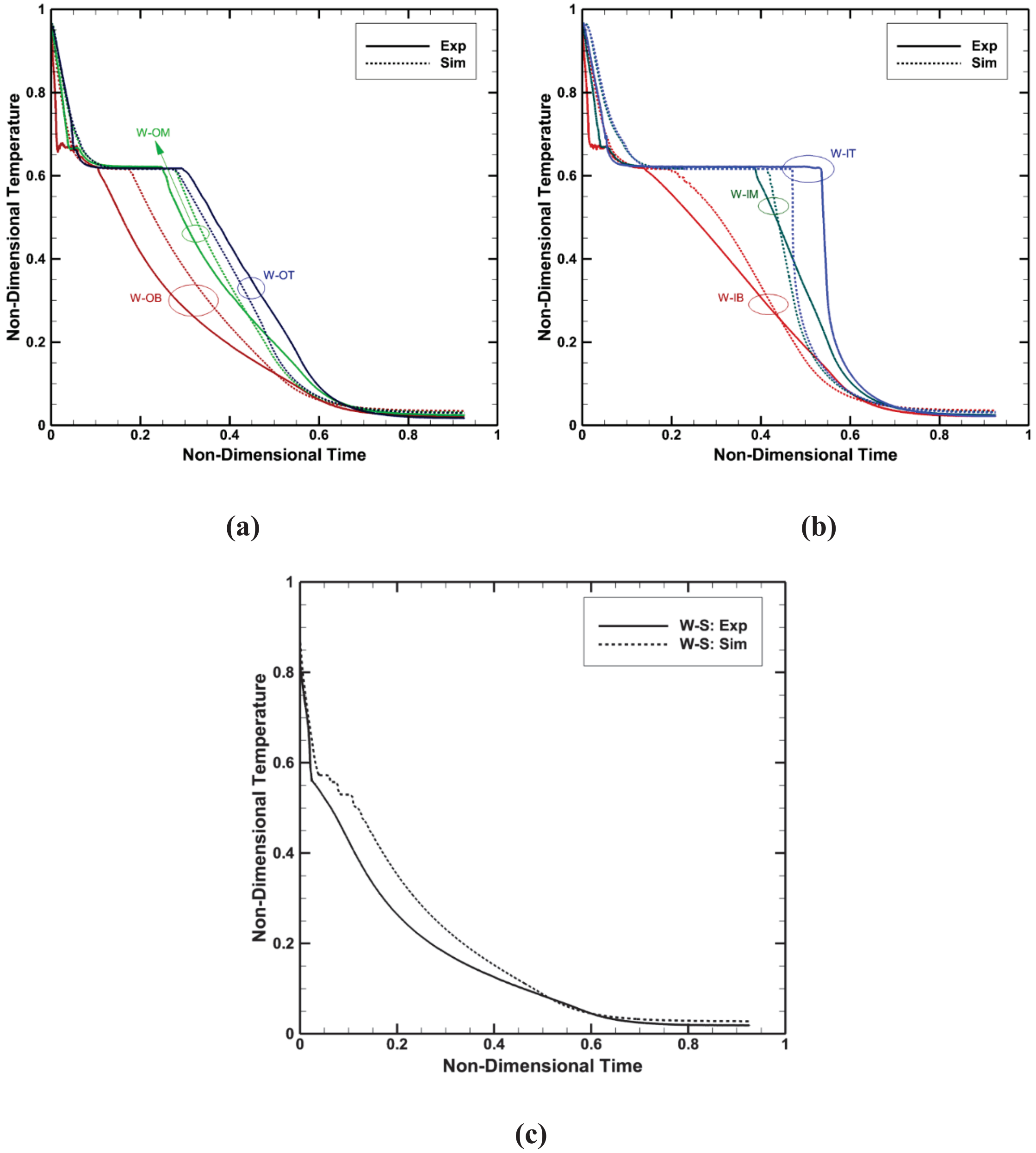


**Figure 4.11: Predicted solid fraction contours for 25% filled tank of water after: (a) 1.4 hours, (b) 2.8 hours, (c) 4.2 hours, (d) 5.6 hours, (e) 6.9 hours (f) 7.2 hours.**

Figure 4.12 shows the temperature vs. time curves for the seven water thermocouples for the 50% fill level validation case. The total time taken for the system to reach steady state is approximately 21 hours which, as expected, is longer than the time taken by the 25% fill level case. The errors in this case are very similar to those obtained during the calibration process. The largest RMS error is 8.21°C for the W-IT thermocouple. The RMS errors for the two new thermocouples, namely W-IM and W-OM, are 3.39°C and 2.25°C, respectively, indicative of the fact that the temperature is predicted with reasonable accuracy not just at locations that were used for calibration, but also at other locations within the water. The freezing sequence in the predicted results is also found to match with experimental observations: W-S → W-OB → W-IB → W-OM → W-OT → W-IM → W-IT was consistent with the experiment. Similar to the calibration run, the wall-clock time for this case was noted to be about 130 hours using 28 processors. The RMS errors for the thermocouples are summarized in Table 4.5.



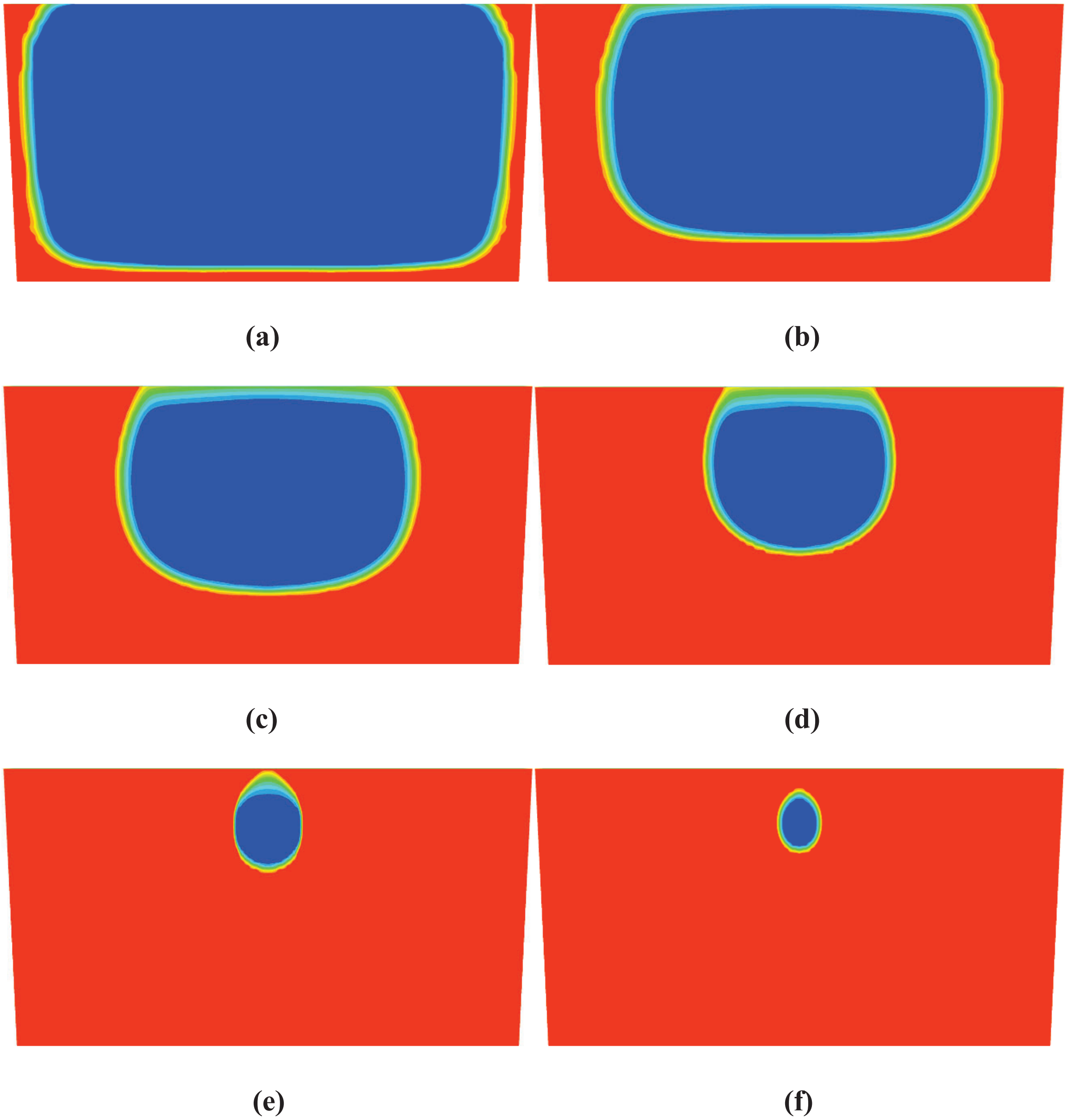
(a)

(b)

(c)

**Figure 4.12: Temperature vs. time curves for the 50% filled tank of water: (a) outer thermocouples, (b) inner thermocouples, (c) side wall thermocouple.**

The propagation of the solidification front for the 50% fill level case is depicted in Fig. 4.13. The initial formation of the solidification front is similar in this case as is in the 25% fill level case. However, the shape of the resulting trapped water bubble is fairly different. Due to the larger surface area, the higher rate of cooling from the side walls results in a much more circular water bubble. Also, near the end of the solidification process, the water bubble is clearly detached from the free surface in this case, while in the case of the 25% fill level case it was adjacent to the free surface. Nonetheless, the observation that the water bubble is closer to the free surface away from the center of the tank implies that cooling rate at the free surface is significantly lower than the cooling at the walls.



(a)

(b)

(c)

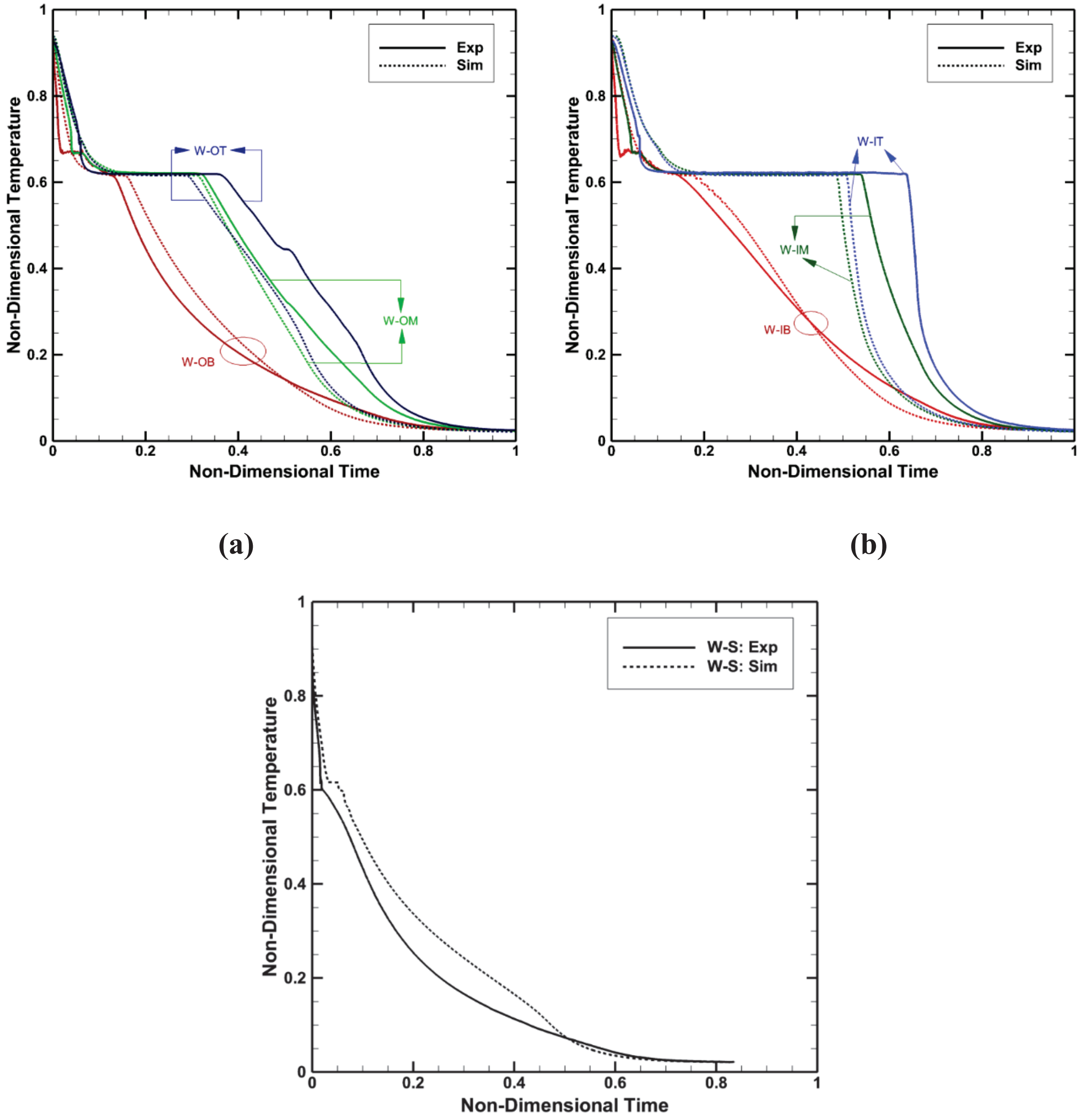
(d)

(e)

(f)

**Figure 4.13: Predicted solid fraction contours for the 50% filled tank of water after: (a) 2.2 hours, (b) 4.4 hours, (c) 6.7 hours, (d) 8.9 hours, (e) 10.8 hours (f) 11.1 hours.**

Figures 4.14 outlines the temperature vs. time curves for the seven water thermocouples for the 80% fill level case. This case required approximately 25 hours to achieve steady state. This is the expected outcome, with higher fill levels requiring longer time periods to complete the solidification process. The natural convection effects are also strongest for this case due to the high fill level. Consequently, the reduced model for natural convection proposed here is put to an ultimate test. As shown in the plots in Fig. 4.16, the simulated results agree reasonably well with the experimental data. The largest RMS error is 10.44°C at the W-IT thermocouple. The increased significance of natural convection effects in this case is likely to explain the slightly larger RMS errors observed in this case relative to the two previous cases. The results are consistent with those of the other two fill levels and the experimental data in all regards except one. The freezing sequence is reversed for two of the thermocouples. In the experiments, the outer top (W-OT) thermocouple freezes after the outer middle (W-OM) thermocouple, as shown in Fig. 4.16(b). In the simulated results, this trend is reversed. W-OT freezes slightly earlier than W-OM. This may be a manifestation of the fact that the rise of the interface and formation of the ice dome was neglected in the model, which acts as an insulating layer for the liquid water underneath. This leads to the model predicting higher heat loss from the top surface of the water than in reality. It may also be a consequence of the assumption that the heat transfer coefficient on the side surface is uniform. This case required approximately 140 hours of computational time to complete the calculations using 28 processors.



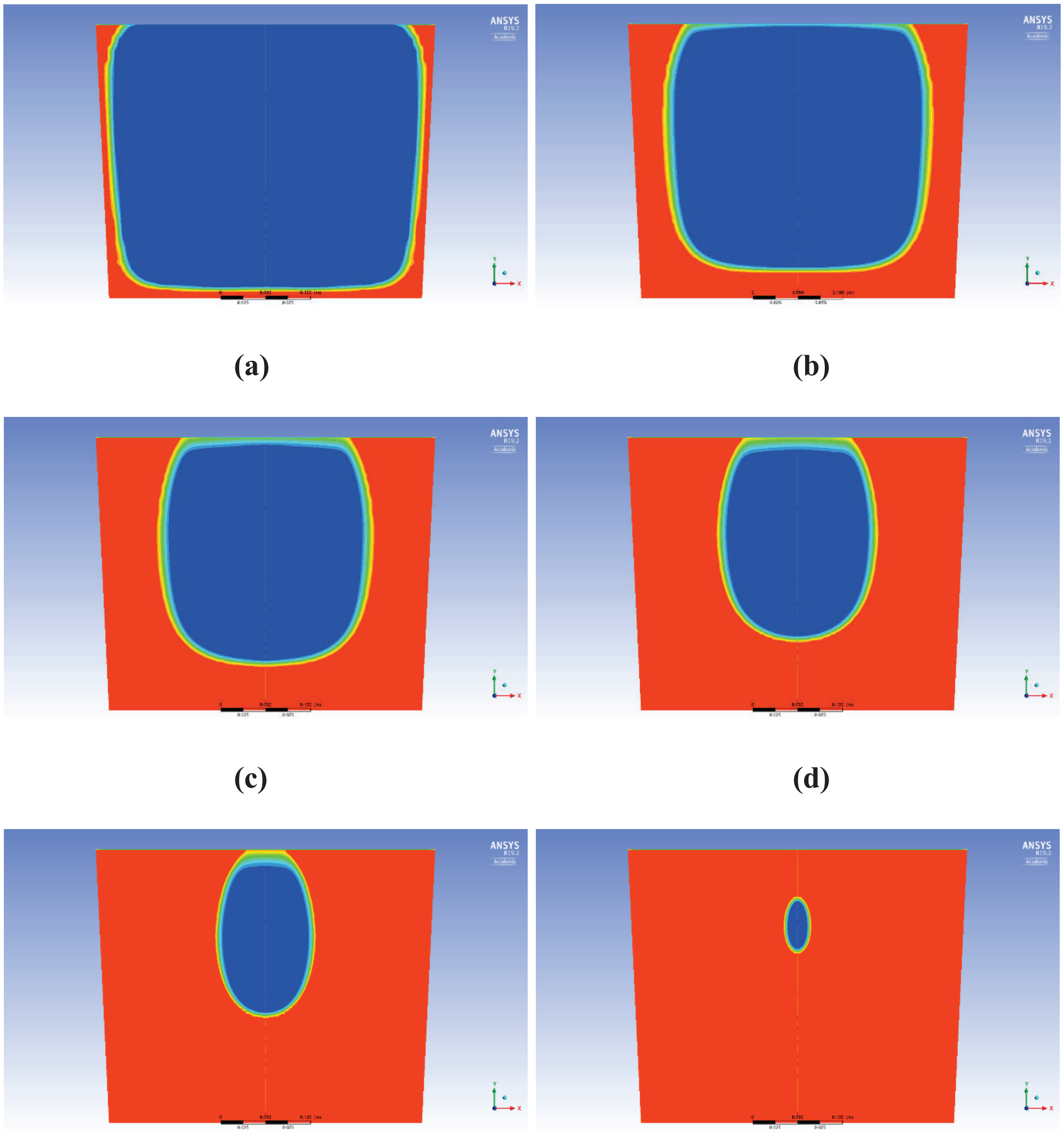
(a)

(b)

(c)

**Figure 4.14: Temperature vs. time curves for the 80% filled tank of water: (a) outer thermocouples, (b) inner thermocouples, (c) side wall thermocouple.**

The propagation of the solidification front for the 80% fill level case is depicted in Fig. 4.15. The initial formation of the solidification front is once again similar in this case to the two previous cases. However, the shape of the water bubble is unique to this fill level. For this case, the shape of the water bubble is opposite to what was observed in the 25% fill case, with the diameter of the bubble smaller than its height. Overall, the variation of the bubble shape between the three fill levels demonstrates the effect increasing the area of side wall has on the cooling rate. Near the end of the solidification process, the water bubble in this case is well below the free surface. This is because the longer solidification time experienced in this case allows the solidification front at the free surface to propagate downwards into the liquid below.



(a)

(b)

(c)

(d)

(e)

(f)

**Figure 4.15: Predicted solid fraction contours for the 80% filled tank of water after: (a) 2.2 hours, (b) 4.4 hours, (c) 6.7 hours, (d) 8.9 hours, (e) 11.1 hours (f) 13.1 hours.**

### 4.7.2 Results with AdBlue® as Working Liquid

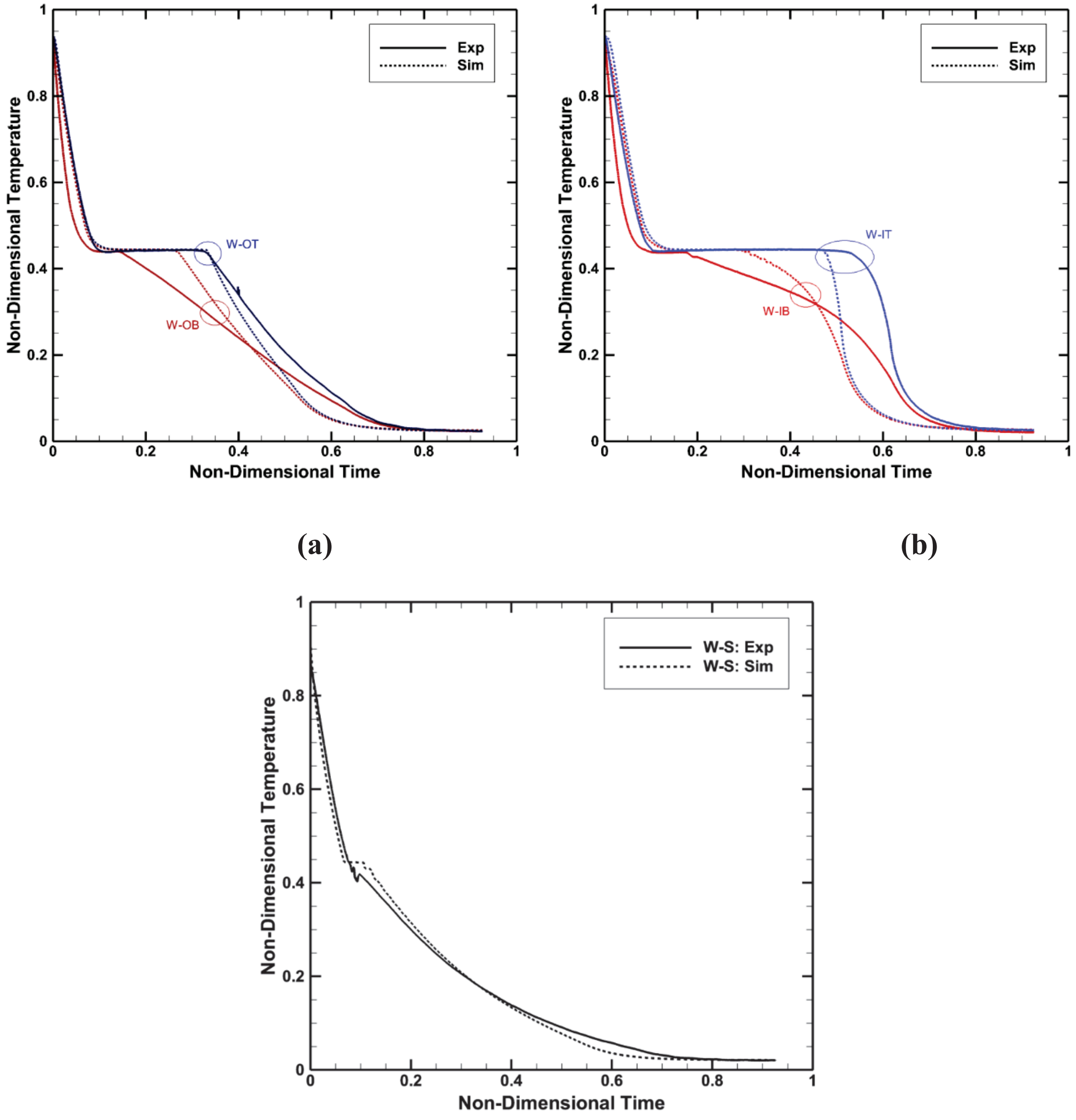
The results for all simulations using AdBlue® at all fill levels are provided in this section. As mentioned previously, the setup for these simulations is nearly identical to the water simulations of the respective fill levels. The only differences are those pertaining to the thermophysical properties of AdBlue® and water, and their respective reduced model constants ($c_u$ vs. $c_w$).

Figure 4.16 outlines the temperature vs. time curves at the thermocouple locations for the 25% fill level. The time taken for this case to reach steady state is approximately 16 hours, which is longer than the time taken by the equivalent water case. This is expected due to the lower freezing point of AdBlue®. The maximum RMS error in this case is 6.48°C at the W-IT thermocouple, which is higher than what was observed in the 25% fill level case for water. However, the freezing sequence of the thermocouples is still correctly replicated by the simulation for this case. The wall-clock time for this stimulation was approximately 125 hours. The RMS errors for the thermocouples are summarized in Table 4.6.

The propagation of the solidification front for this case is depicted in Fig. 4.17. The progress of the solidification front in this case is nearly identical to what was observed in the case with water. This suggests that the propagation of the solidification front is more dependent on geometric factors and boundary conditions of the system than the thermophysical properties of the material.

Table 4.6: Average (RMS) error (°C) between the simulation and experiment for all three fill levels of AdBlue®.

| Thermocouple | RMS error | | |
|---|---|---|---|
| | 25% fill level | 50% fill level | 80% fill level |
| W-OB | 2.73 | 2.72 | 3.22 |
| W-OM | - | 2.51 | 4.04 |
| W-OT | 1.97 | 4.47 | 7.03 |
| W-S | 1.03 | 1.55 | 2.37 |
| W-IB | 4.12 | 3.44 | 3.79 |
| W-IM | - | 4.79 | 8.31 |
| W-IT | 6.48 | 8.28 | 11.48 |

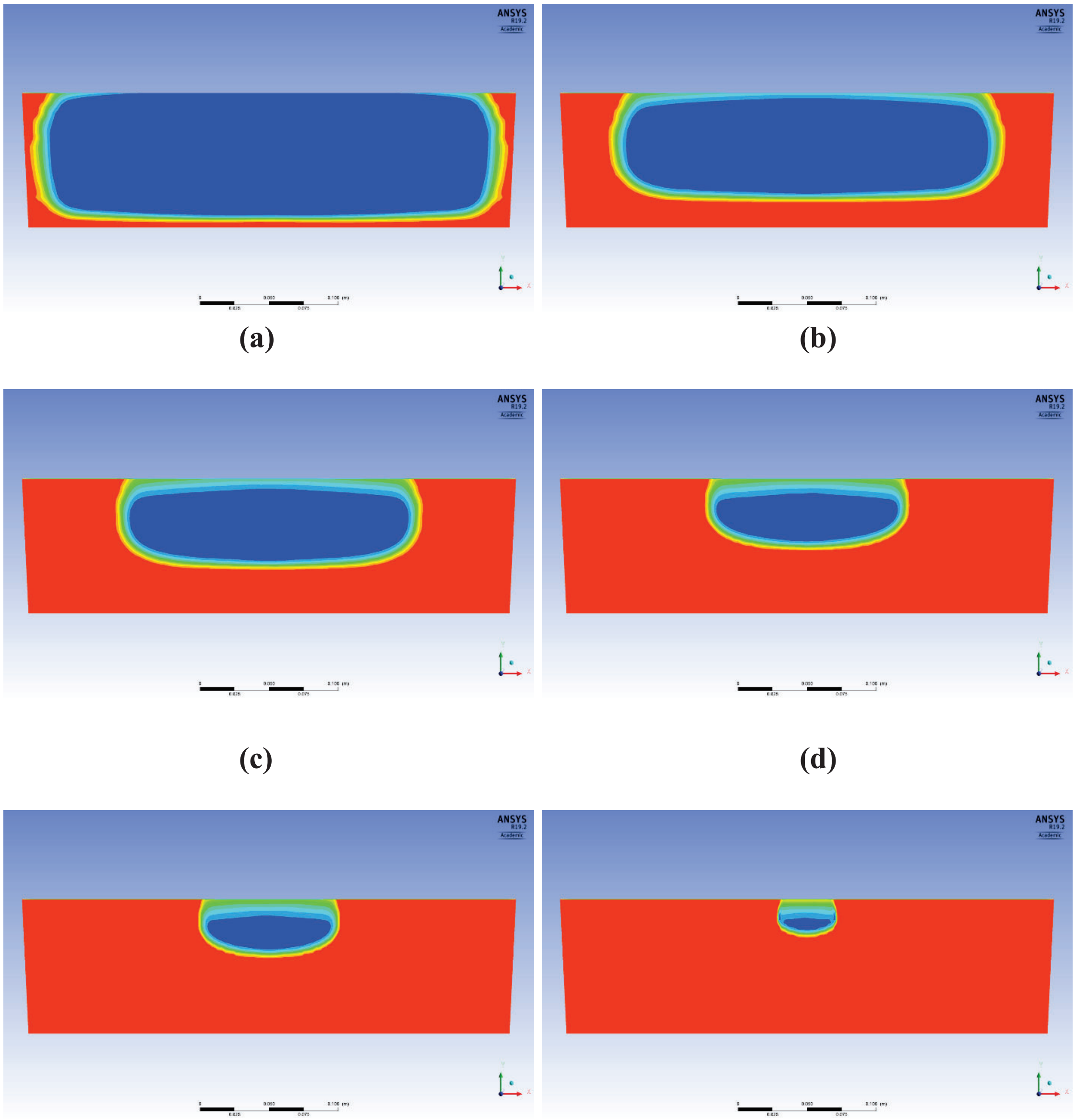


(a)

(b)

(c)

**Figure 4.16: Temperature vs. time curves for the 25% filled tank of AdBlue®: (a) outer thermocouples, (b) inner thermocouples, (c) side wall thermocouple.**



(a) (b)

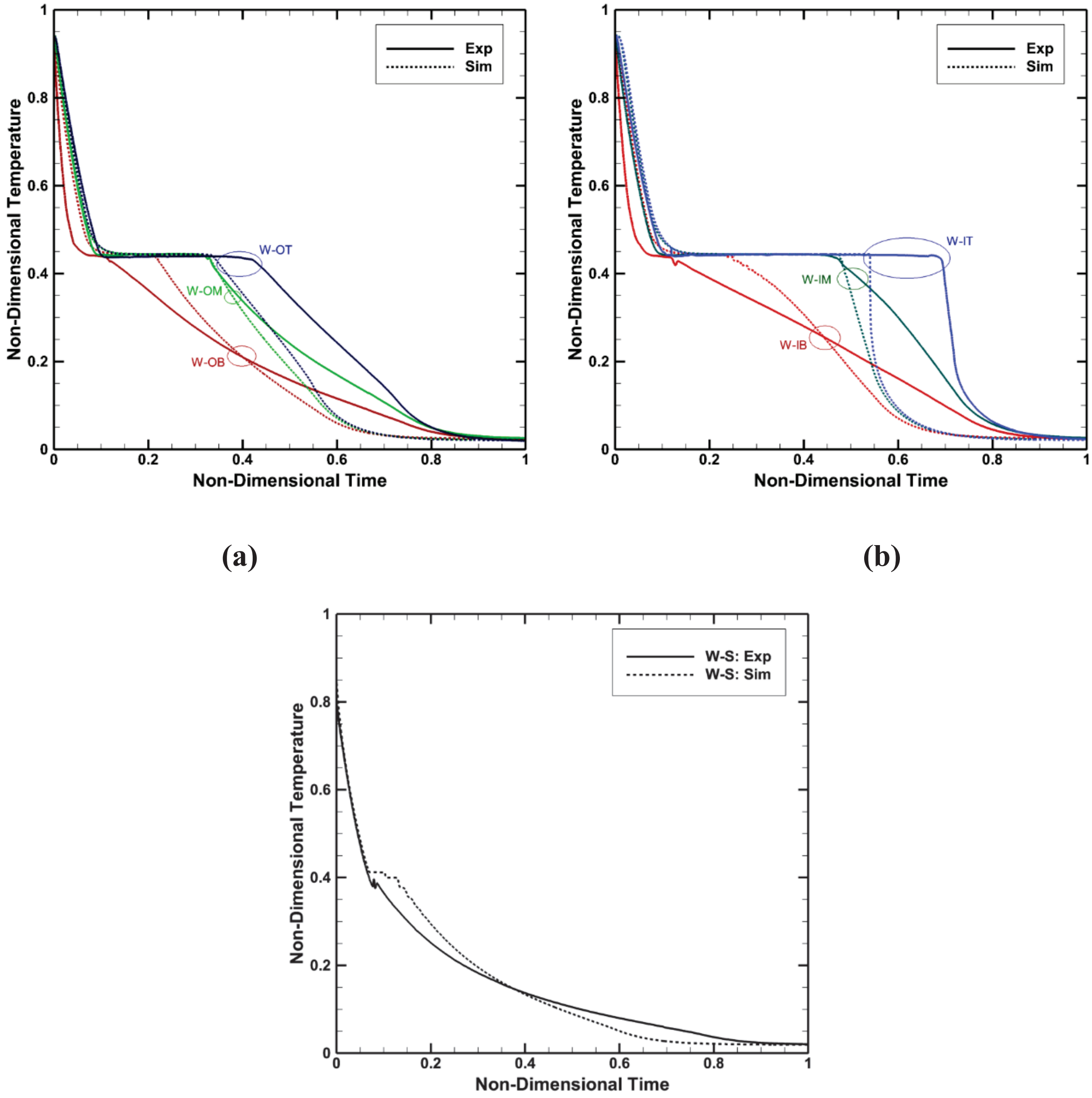
(c) (d)

(e) (f)

**Figure 4.17: Predicted solid fraction contours for 25% filled tank of AdBlue® after: (a) 2.2 hours, (b) 3.9 hours, (c) 5.6 hours, (d) 7.2 hours, (e) 8.1 hours (f) 8.9 hours.**

Figure 4.18 shows the temperature vs. time curves for the 50% fill level case with AdBlue®. The system takes approximately 25 hours to reach steady state. The largest RMS error is 8.28°C for the W-IT thermocouple. This error is larger than the maximum in both the 25% fill level AdBlue® and 50% fill level water cases. The freezing sequence in the predicted results is also found to match with experimental observations: W-S → W-OB → W-IB → W-OM → W-OT → W-IM → W-IT is consistent with the experiment. The wall-clock time for this case was noted to be about 170 hours using 28 processors.

The propagation of the solidification front for the 50% fill level case is depicted in Fig. 4.19, which is very similar to the behavior seen in the water case in Fig. 4.13. The difference in shape of the water bubble for the 50% fill level case with respect to the 25% fill level case is also replicated with AdBlue®.

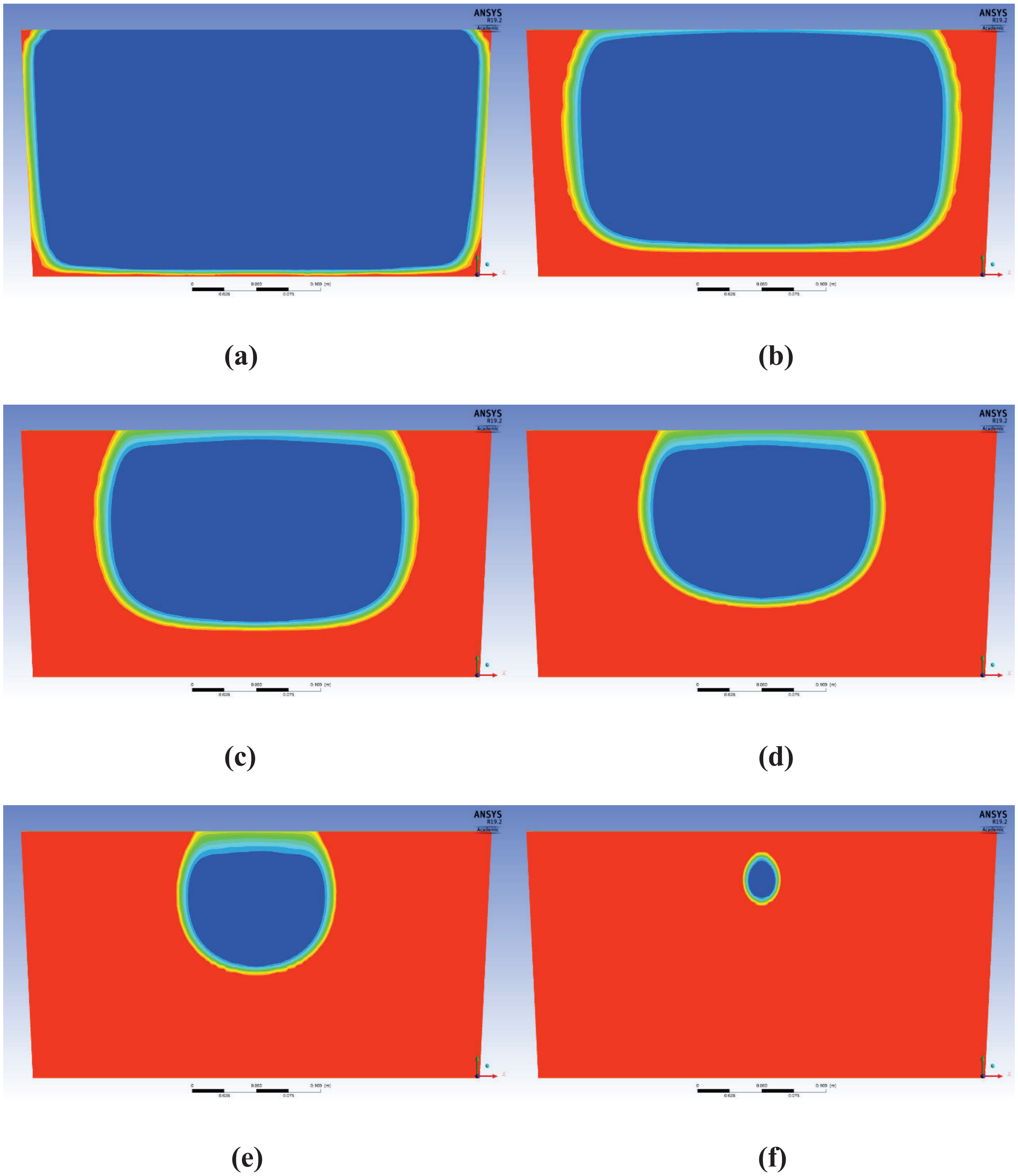


(a)

(b)

(c)

**Figure 4.18: Temperature vs. time curves obtained for the 50% filled tank of AdBlue®: (a) outer thermocouples, (b) inner thermocouples, (c) side wall thermocouple.**



(a)

(b)

(c)

(d)

(e)

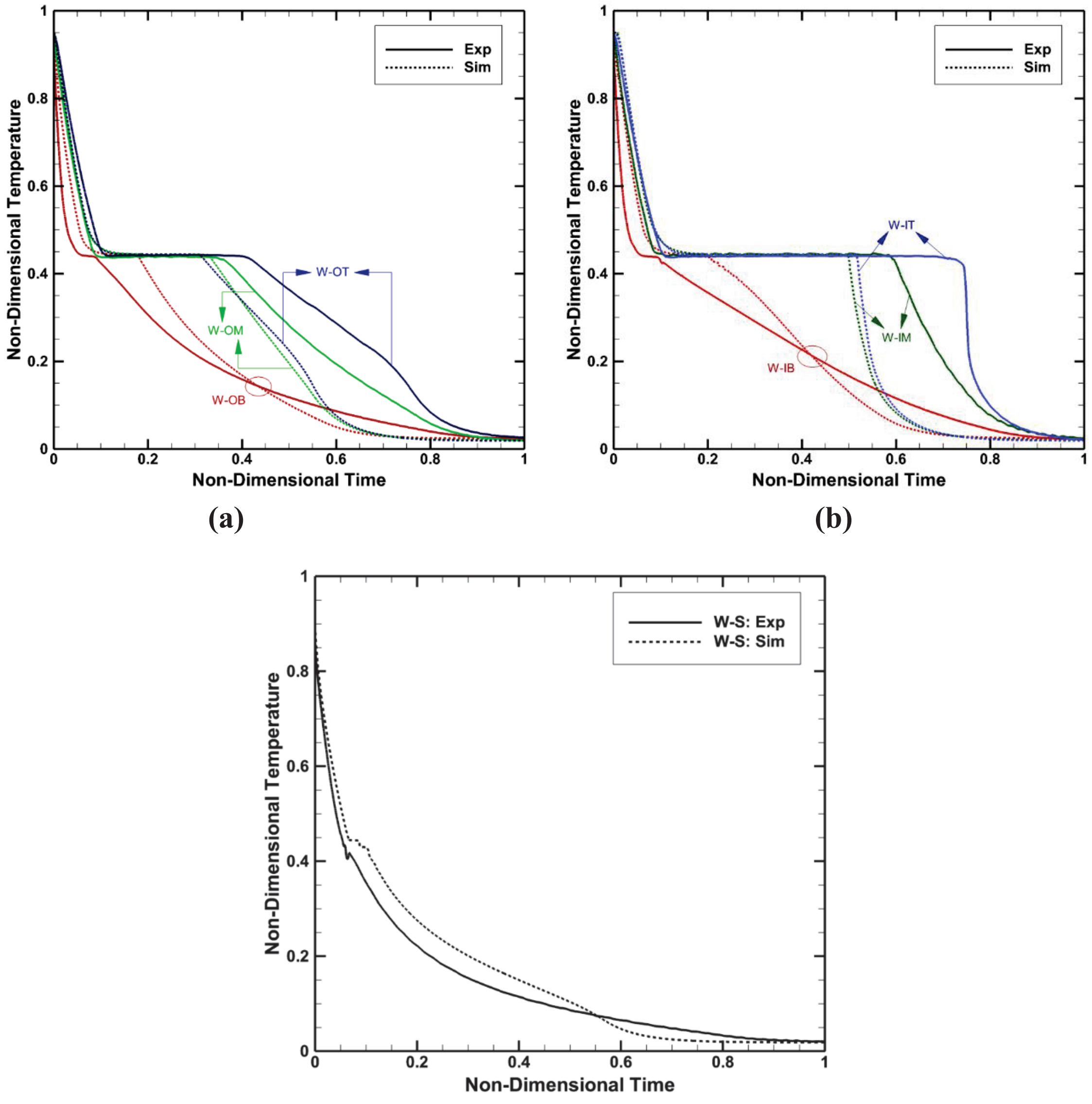
(f)

**Figure 4.19: Predicted solid fraction contours for the 50% filled tank of AdBlue® after: (a) 2.2 hours, (b) 4.4 hours, (c) 6.7 hours, (d) 8.9 hours, (e) 10.8 hours (f) 11.1 hours.**

Lastly, Fig. 4.20 outlines the resulting temperature vs. time curves for all thermocouples for the 80% fill level case. This case took the longest time to reach steady state of all 6 cases - approximately 32 hours. Consequently, the wall-clock times for this case are also the longest, requiring around 150 hours to simulate the experiment. The RMS errors are also the largest for this case, with the maximum RMS error of 11.48°C occurring at the W-IT thermocouple location. All three AdBlue® cases have had their maximum RMS error at this location. The flipped freezing sequence for the W-OM and W-OT thermocouples observed in the water case is also seen in this case.

The propagation of the solidification front for the 80% fill level case is depicted in Fig. 4.21 As with the other fill level cases, the behavior of the solidification front in this case is very similar to the 80% fill level water case.

Table 4.6 lists the RMS errors at all thermocouple locations for the three AdBlue® cases. Compared to the RMS errors reported for the corresponding water cases, the errors are slightly higher for almost all thermocouple locations (with a few exceptions). However, considering that the reduced model constant and external heat transfer coefficients were calibrated only once, and using a 50% fill level case with water, it is impressive that the model is able to provide reasonably accurate predictions. It is likely that if recalibration was performed with an AdBlue® calibration data set, the agreement would improve for the validation data sets.

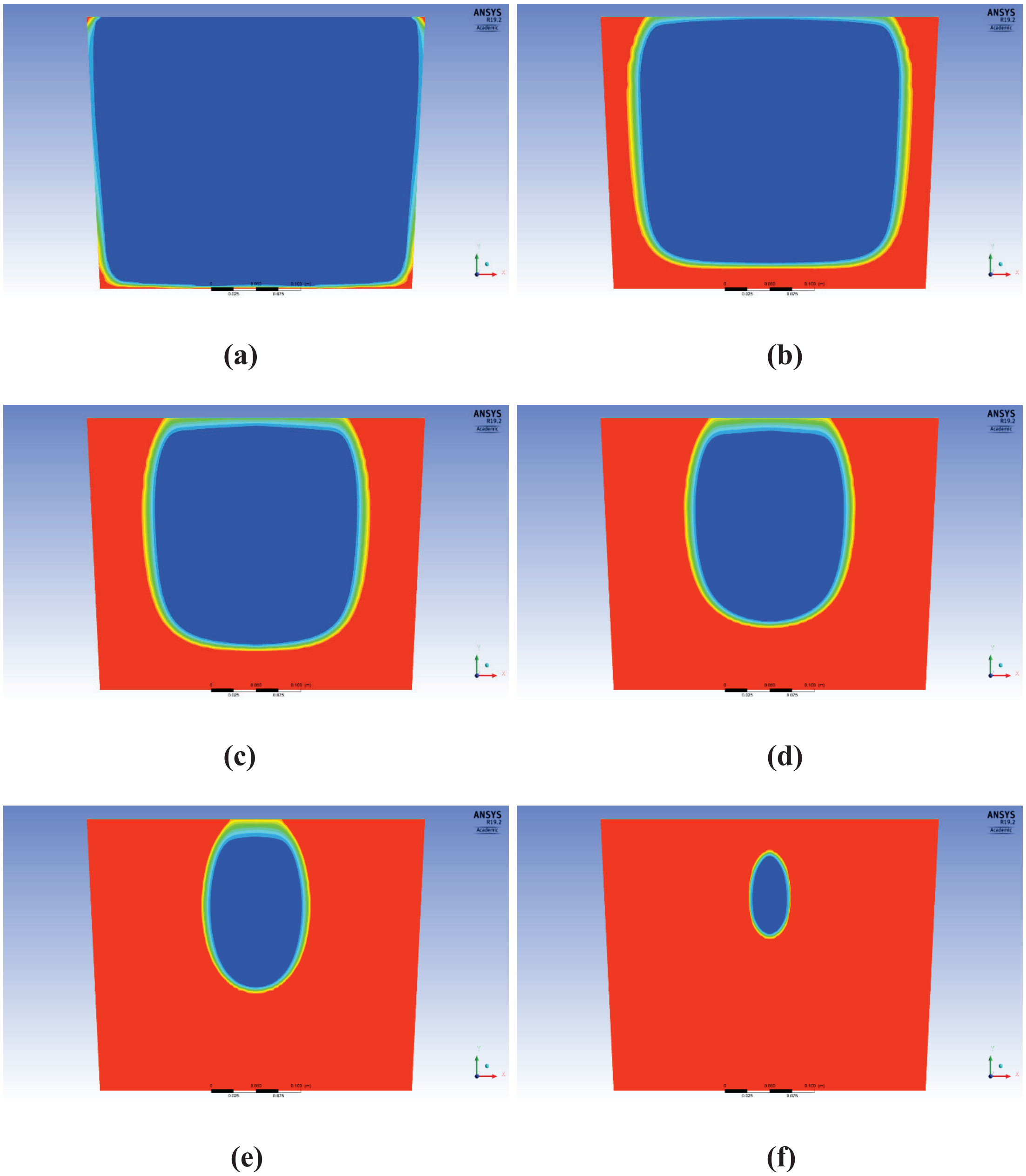


(a)

(b)

(c)

**Figure 4.20: Temperature vs. time curves for the 80% filled tank of AdBlue®: (a) outer thermocouples, (b) inner thermocouples, (c) side wall thermocouple.**



(a)

(b)

(c)

(d)

(e)

(f)

**Figure 4.21: Predicted solid fraction contours for the 80% filled tank of AdBlue® after: (a) 2.2 hours, (b) 5 hours, (c) 7.8 hours, (d) 10.6 hours, (e) 13.3 hours, (f) 15.6 hours.**

## 4.8 Verification Study: Natural Convection in Differentially Heated Cavity

In order to highlight the generality of the reduced natural convection model, a verification study for a benchmark heat transfer problem was undertaken: natural convection in a differentially-heated enclosure.

A differentially-heated enclosure (or cavity) typically involves a rectangular enclosure with one of the walls designated as a hot wall at a high temperature and the opposite wall designated as a cold wall at a lower temperature. These two walls constitute isothermal boundary conditions for the enclosure, while the other two walls are set to be adiabatic (insulated). The enclosure is completely occupied by a fluid, with an initial temperature equal to that of the cold wall. If the hot and cold walls are the vertical walls of the enclosure, the problem can be further classified as a vertical cavity with sidewall heating [52]. A simple schematic of such a problem is shown in Fig. 4.22.

The natural convection problem in differentially-heated enclosure is a well-studied problem in heat transfer, with an enormous volume of literature reviewing experimental, theoretical and numerical simulation results for this problem (Refs. [52]-[54] and the work cited therein). Therefore, this problem was chosen for the current verification study. Additionally, this problem is a more natural convection-focused problem, compared to the solidification problems tackled in the validation studies. Using this problem will therefore test the reduced natural convection model in a problem it was not originally intended for, while also allowing a more targeted analysis of the prediction capabilities of the reduced model during this study.

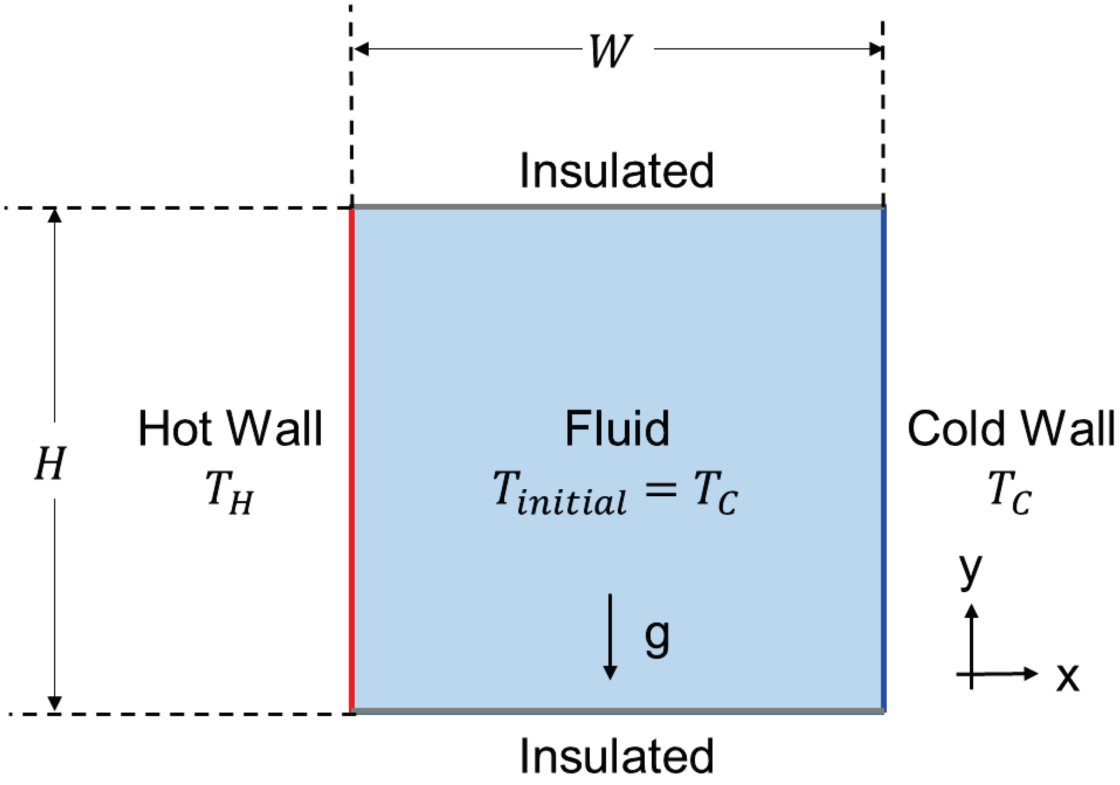


**Figure 4.22: Schematic of a typical differentially-heated enclosure.**

In the current study, a square cavity ($H/W$=1) is considered for the problem. Air is chosen as the enclosed fluid, as the reduced model constant, $c_a$, was already calibrated during the validation studies. It is important to note that the same calibrated $c_a$ value (=118,580) is used in this study without any recalibration procedures. Two different Rayleigh numbers were considered: Ra = ($10^4$, $10^5$, $10^6$). These Rayleigh numbers are above the minimum threshold for buoyancy-driven flow in enclosures ($10^3$) and well below the critical Rayleigh number for turbulence ($10^9$) [52]. The temperature of the cold wall ($T_C$) was set to be 300 K. The temperature of the hot wall ($T_H$) and size of the cavity (H) were computed from the prescribed Rayleigh numbers by inverting the following relationship [52],

$$\mathrm{Ra} = \frac{g\beta(T_H - T_C)H^3}{\alpha\nu} \tag{4.7}$$

where $g$ is the acceleration due to gravity, $\beta$ is the volumetric thermal expansion coefficient of air, $\alpha$ and $\nu$ are the thermal diffusivity and kinematic viscosity of air, respectively. The values used for these properties, and other thermophysical properties of air as used in this study are listed in Table 4.7. The corresponding hot wall temperatures and cavity dimensions were subsequently obtained as $T_H$ = (301, 306, 307) K and $H$ = (0.05, 0.05, 0.1) m, respectively.

Table 4.7: Thermophysical properties of air as used in the current study.

| Parameter | Value/Definition |
|---|---|
| **Density** | Ideal gas law |
| **Thermal Conductivity** | 0.024 W/m-K |
| **Specific Heat Capacity** | 1006.43 J/kg-K |
| **Dynamic Viscosity** | 1.7894×10$^{-5}$ kg/m-s |
| **Vol. Thermal Exp. Coeff., $\boldsymbol{\beta}$** | 3.38×10$^{-3}$ kg/m-s |
| **Prandtl number** | 0.707 |

As stated, the goal of this verification study is to compare the predictions of simulations to exact numerical solutions of the original governing equations. For the sake of convenience, ANSYS Fluent$^{TM}$ was used to generate these exact numerical solutions as the reduced model was already successfully implemented in Fluent, as discussed in Section 3.5. Simulations were set up using the high-fidelity models and the proposed reduced natural convection separately for each of the prescribed Rayleigh numbers, resulting in a total of six cases. The major difference between these simulations is the fact that the high-fidelity simulations solved for both energy and flow (with gravity turned on), while the other (reduced model) simulations only solved for energy while enhancing the thermal conductivity of air as per Eq. (3.17) of the reduced model. Other simulation setup details such as thermophysical properties, meshing and solver setup were the same for both simulations. These details are summarized below.

- The simulations used Fluent's 2D, pressure-based (in the case of flow), transient solver.
- The cavity was meshed using a simple 2D 50x50 uniform structured mesh comprised of solely quadrilateral cells.
- In the case of the simulation using Fluent's inbuilt models, the SIMPLE scheme was used for pressure-velocity coupling. The flow field and energy equations were discretized using the second order upwind scheme.
- The first order implicit scheme (backward Euler) was used for the transient formulation in both simulations.

- The air in the cavity was assumed to be at rest and at 1 atm pressure at the start of the simulation. It was initialized to a uniform temperature of 300 K, which was equal to the temperature of the cold wall.

A time step size of 1 second was used for all simulations. The simulations were run until the system reached steady state. Steady state was monitored by comparing the total heat fluxes at the hot and cold walls until the two fluxes were equal in magnitude.

### 4.8.1 Results and Observations

To study the predictions of the two models, the predicted temperatures at five locations within the enclosure at steady state are compared. These locations at which these temperatures are measured are designated left-bottom (LB), right-bottom (RB), middle (M), left-top (LT) and right-top (RT), and are marked in green in the schematic of the enclosure shown in Fig. 4.23. The predicted temperatures at these locations are listed in Table 4.8. In this table, the columns RM and HF represent the temperatures predicted by the reduced model and the high-fidelity model, respectively.

From the results in Table 4.8, it can be seen that the predicted temperature at location M is equal for both models across all three Rayleigh numbers. This is expected, as this location at the center of the tank represents a point at which the natural convection effects are balanced out, and therefore the weakest. Next, it is observed that the difference between the predicted temperatures is maximum at the LB and RT locations. However, there is a certain symmetry to this difference, with the reduced model over-estimating the temperature at the LB location, but then under-estimating the temperature at the RT location by a similar magnitude. Once again, this behavior is consistent across all three

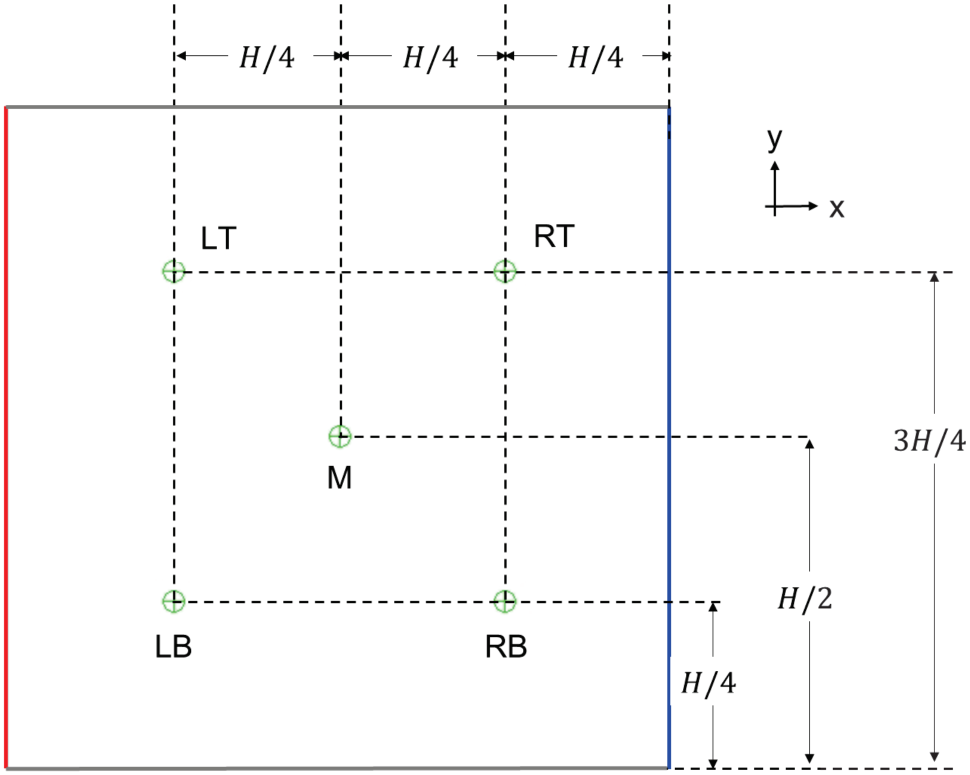


**Figure 4.23: Schematic showing temperature measurement locations in the differentially-heated enclosure.**

**Table 4.8: Temperatures predicted by the reduced and high-fidelity models at five locations within the differentially-heated enclosure.**

| Location | Predicted Temperatures (K) | | | | | |
|---|---|---|---|---|---|---|
| | $10^6$ | | $10^5$ | | $10^4$ | |
| | RM | HF | RM | HF | RM | HF |
| **LB** | 305.24 | 301.94 | 304.50 | 301.74 | 300.36 | 300.75 |
| **RB** | 301.74 | 302.07 | 301.50 | 301.98 | 300.25 | 300.25 |
| **M** | 303.49 | 303.49 | 302.99 | 303.00 | 300.50 | 300.50 |
| **LT** | 305.24 | 304.89 | 304.50 | 304.01 | 300.75 | 300.75 |
| **RT** | 301.74 | 305.05 | 301.50 | 304.25 | 300.64 | 300.25 |

Rayleigh numbers. The other two locations, RB and LT, show a similar behavior but at a lower magnitude. This behavior is due to the comparison of the 2D temperature flow generated by the recirculating flow in the high-fidelity model's results, to the 1D temperature flow that is obtained from the reduced model (by virtue of it reducing the problem to a conduction problem).

The symmetry in the differences in predicted temperatures suggests that, in spite of the local differences, the reduced model provides a good overall estimate of the heat transfer. To verify this hypothesis, the results predicted by both models are compared by calculating the average Nusselt number, $\overline{\mathrm{Nu}}$, at the hot wall for each of the cases, which is given by[52],

$$\overline{\mathrm{Nu}} = \frac{\bar{h}H}{k_{air}} = \frac{1}{k_{air}} \int_{0}^{H} h_y \, dy \tag{4.8}$$

where $\bar{h}$ and $h_y$ are the average and local heat transfer coefficients, respectively, and $k_{air}$ is the thermal conductivity of air. $h_y$ is then calculated from the local wall heat flux, $(q''_{wall})_{\mathrm{y}}$, as

$$h_y = \frac{(q''_{wall})_{\mathrm{y}} y}{(T_H - T_C)} \tag{4.9}$$

Additionally, these Nusselt number predictions were also compared to empirical correlations. For the current problem, the relevant average Nusselt number correlation is given by[52],

$$\overline{\mathrm{Nu}} = 0.18 \left( \frac{\mathrm{Pr}}{0.2 + \mathrm{Pr}} \mathrm{Ra} \right)^{0.29} \tag{4.10}$$

where Pr is the Prandtl number of air.

Table 4.9 lists the average Nusselt numbers at the hot wall for each of the covered cases. From these results, it can be seen that there is good agreement between the high-fidelity and reduced models for all three Rayleigh numbers. Interestingly, the theoretical correlations for Nusselt number are consistently in-between the predictions of the two models. However, it appears that the reduced model over-estimates the effect of natural convection in the Ra = $10^4$ and $10^5$ cases, while slightly under-estimating it in the Ra = $10^6$ case.

These results highlight the fact that the reduced model is very good at providing an overall estimate of the heat transfer occurring within a system while providing significantly higher computational efficiency, as seen from the run times for the simulations that were part of the validation study. However, this efficiency comes at the cost of the finer details of the heat transfer process in the system.

**Table 4.9: Comparison of average Nusselt number at hot wall.**

| Rayleigh Number, Ra | Average Nusselt Number, $\overline{\mathrm{Nu}}$ | | |
|---|---|---|---|
| | **Nu Correlation [52]** | **High-fidelity Model** | **Reduced Model** |
| $10^4$ | 2.85 | 2.48 | 3.43 |
| $10^5$ | 4.79 | 4.27 | 4.95 |
| $10^6$ | 8.36 | 8.42 | 8.30 |

### 4.9 Summary

This chapter covered the details of the experimental study performed for the calibration and validation of the reduced natural convection model. The validation study revealed that there was reasonably good agreement between measured and simulated temperature data for all cases when using the reduced model while also significantly lowering computational times when compared to Fluent's inbuilt models. The reduced model was also able to accurately predict the freezing sequence of the thermocouples in most cases, except for the W-OT and W-OM thermocouples in both 80% fill level cases.  A verification study of the reduced model was then discussed separately for a benchmark problem different from the application at hand. This study showed that the reduced model was able to accurately predict the overall heat transfer rate in the system but may not accurately predict temperature within the system.

The simulations run as part of the validation study in this chapter did not account for the expansion and rise of an ice dome at the free surface. As suggested in the validation study, the absence of an ice dome is a possible explanation for the incorrect freezing sequence observed in the 80% fill level cases, so it is important to model the formation of the ice dome. However, for reasons discussed in the previous chapter, VOF-based interface tracking models are not compatible with the reduced natural convection model. Therefore, the next chapter details the modelling of a new free surface tracking model used to capture the expansion and rise of the ice dome.

# CHAPTER 5. VOLUME TRANSPORT MODEL – THEORY

## 5.1 Introduction

Chapters 3 and 4 covered the modelling and validation of a reduced natural convection model that tracked the solid-liquid solidification front. However, as discussed in Section 1.2 of Chapter 1, the freezing process in partially filled tanks involves a second moving boundary/front: the gas-(solid/liquid) free surface (interface). This interface describes the ice dome that forms due to expansion of ice during solidification. As explained in Section 4.7, the presence (or absence) of this ice dome is expected to have some effect on the heat transport occurring at the top of the gas-(solid/liquid) free interface. The reduced natural convection model, described in Chapters 3 and 4, does not account for expansion of ice and the consequent movement of the gas-(solid/liquid) free interface and, therefore, a separate model is required for this purpose. The current chapter covers the modelling details for such a model as well as its coupling to the overall energy equation.

One of the most prolific models for the tracking of free surfaces is the Volume-of-Fluid (VOF) method [28], already surveyed in Chapter 1. The model proposed here uses the fundamental idea of the VOF method. A defining feature of the reduced natural convection model is that it bypasses solving for fluid flow by modelling the heat transport due to natural convection as a diffusive process via an artificial enhanced thermal conductivity (Section 3.2). However, as the Volume-Of-Fluid (VOF) method [28] uses advection fluxes to propagate the gas-(solid/liquid) interface, it means that the reduced natural convection model and the conventional VOF method are incompatible with each other. This led to the development of a new diffusion-based form of the VOF method (henceforth referred to as the *volume transport model*) that would be able to track the gas-(solid/liquid) interface and could be used alongside the reduced natural convection model. The following sections provide a description of the volume transport model.

## 5.2 Volume Transport Model Formulation

As per the conservation of mass, if the density of a liquid decreases during freezing, the freezing process results in the creation of extra volume. This extra volume may either fill any empty volume available within the control volume or may spill out of the control volume if there is insufficient available volume. The spilled (or excess) amount must then be redistributed (transported) to other adjacent control volumes (or cells). When this process is executed for all cells initially containing liquid or a solid/liquid mixture, it will result in the movement of the gas-(solid/liquid) interface, as shown in Fig. 5.1.

As in the VOF method, the proposed volume transport model describes the gas-(solid/liquid) interface using a scalar called the volume fraction, which is defined as

$$f = \frac{V_{ls}}{V} \tag{5.1}$$

where $f$ is the volume fraction of liquid plus solid, $V_{ls}$ is the volume occupied by liquid plus solid and $V$ is the total volume occupied by all three phases combined. As discussed in the previous section, due to the fact that advective fluxes cannot be used to transport volume (as done in the VOF method), a diffusive transport process is instead proposed and the equation governing this process is obtained from the classical diffusion

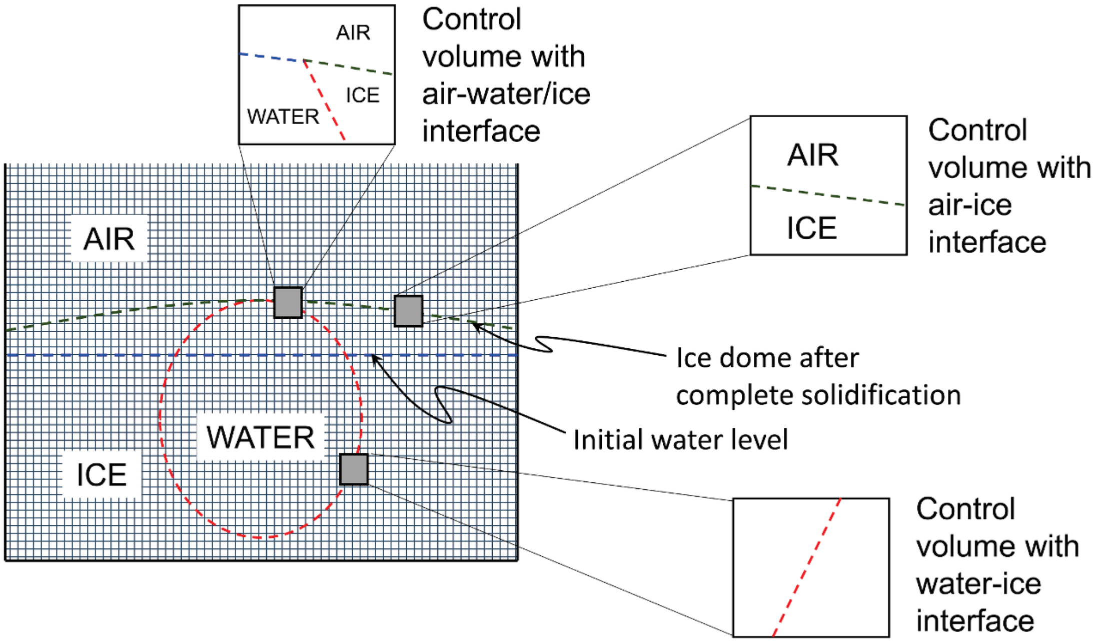

AIR
ICE
WATER
Control volume with air-water/ice interface
AIR
ICE
Control volume with air-ice interface
AIR
Ice dome after complete solidification
Initial water level
WATER
ICE
Control volume with water-ice interface

**Figure 5.1: Schematic representation of freezing of water in a partially filled tank with the various phase boundaries and regions.**

equation. However, before this governing equation for the proposed model can be stated, a quantity derived from volume fraction, called the *excess* volume fraction, $f_e$, must be defined:

$$f_e = \begin{cases} 0 & \text{if } f \leq f_{max} \\ f - f_{max} & \text{if } f > f_{max} \end{cases} \tag{5.2}$$

where $f_{max}$ is the maximum allowable volume fraction, typically equal to 1. However, during the solidification process, imperfections in the form of air pockets are often observed [61][62]. These air pockets prevent the complete filling of cells and have the effect of limiting the maximum allowable volume fraction. To account for this porous nature of the solid, a model for porosity was implemented in the volume transport model. This is achieved by limiting the maximum allowable volume fraction based on the required porosity. Mathematically, this is given by

$$f_{max} = 1 - p \tag{5.3}$$

where $p$ is the prescribed porosity of the solid. A discussion covering appropriate choices for porosity values is provided in Section 6.5, using experimentally measured ice dome height data. With this, the governing equation for the proposed model can be stated as

$$\frac{\partial f_e}{\partial t} = -\nabla . \mathbf{J}_f + \dot{S_f} \tag{5.4}$$

where $\mathbf{J}_f$ is the flux of $f_e$ and $\dot{S_f}$ is a source term for volume, representative of the source of excess volume due to expansion during freezing. The treatment of this source is detailed in Section 5.4.2. $\mathbf{J}_f$ is further defined according to the gradient diffusion hypothesis as

$$\mathbf{J}_f = -\Gamma_f \nabla f_e \tag{5.5}$$

where $\Gamma_f$ is the diffusion coefficient for excess volume. The negative sign in Eq. (5.5) implies that volume transport will occur from high values of $f_e$ (cells with more excess volume) to low values of $f_e$ (cells with less excess volume), which is consistent with physical intuition. In turn, $\Gamma_f$ is analogous to thermal conductivity in the heat conduction equation, and determines the rate of volume transport. Consequently, due to the structural rigidity and near impermeable nature of solids as compared to liquids, it follows that the diffusion coefficients for excess volume in solids must be significantly lower than that in liquids. The selection of appropriate values for the diffusion coefficients is covered in further detail in Section 5.4.1.

Lastly, the proposed volume transport model makes one significant assumption: the effect of gravity is neglected, i.e., there is no “settling” effect.

## 5.3 Calculation of thermophysical properties

While the calculation of thermophysical properties of most cells remains the same as was discussed during the modelling of the reduced natural convection model in Section 3.4, the movement of the gas-(solid/liquid) interface with the volume transport model gives rise to cells filled with a combination of all three phases, as can be seen in Fig. 5.1 previously. Therefore, the relations used to determine the relevant thermophysical properties must be updated to account for this new behavior.

Once again, it follows logically that the thermophysical properties for cells containing multiple phases (henceforth referred to as *mixed cells,* represented with subscript *mix*) must be dependent on the properties of the individual pure phases and the relative quantities of each phase present in the cell. It is important at this point to observe that solid fraction, $s$, as per its definition in Eq. (3.18), represents the ratio between the *mass of the solid* and the *mass of solid plus liquid phase* in the cell. This is in contrast to the definition of volume fraction in Eq. (5.1), which represents the ratio between the *volume of the liquid plus solid phase* and the *total volume* of the cell.

The densities for the pure liquid and solid phases are determined by the density versus temperature relationships provided in literature ([46],[47]), while the density of pure gas (air) is obtained using the ideal gas law. The density for mixed cells is in turn calculated using a volume fraction-based mixing law as shown below.

$$\rho_{mix} = \rho_g(1 - f) + \rho_{sl} f \tag{5.6}$$

where $\rho_g$ is the density of gas and $\rho_{sl}$ is the density of the combined liquid plus solid phase, as given by Eq. (3.25). Unlike Eq. (3.25), Eq. (5.6) does not use an inverse mixing law as $f$ is a volume fraction as opposed to $s$ which is a mass fraction. For the same reason, when calculating the specific heat capacity of mixed cells, the volume fraction must be converted to the corresponding mass fraction before using the appropriate mixing law. This gives the following relation.

$$c_{p,mix} = c_{p,g}\left(1 - f\frac{\rho_{sl}}{\rho_{mix}}\right) + c_{p,sl}\left(f\frac{\rho_{sl}}{\rho_{mix}}\right) \tag{5.7}$$

where $c_{p,g}$ is the specific heat capacity of gas and $c_{p,sl}$ is the specific heat capacity of the combined liquid plus solid phase, as given by Eq. (3.27).

Lastly, the thermal conductivity was calculated using the following expression.

$$k_{mix} = k_{eff,g}(1 - f^2) + k_{eff,sl} f^2 \tag{5.8}$$

where $k_{eff,g}$ is the effective thermal conductivity of gas as obtained using Eq. (3.17) of the reduced natural convection model, and $k_{eff,sl}$ is the effective thermal conductivity of the combined liquid plus solid phase, as given by Eq. (3.28).

### 5.4 Numerical implementation of the Volume Transport model

The densities for the pure liquid and solid phases are determined by the density versus temperature relationships. One of the primary motivating factors behind the development of the proposed volume transport model is its compatibility with the reduced natural convection model proposed previously in Chapter 3. As the reduced natural convection model was implemented in ANSYS-Fluent$^{TM}$ (for reasons explained in Section 3.5), the volume transport model also had to be implemented in the same software. As the reduced natural convection model primarily requires the modification of thermal conductivity, Fluent's in-built numerical solver could be used to solve the energy equation. However, the volume transport model requires the solution to Eq. (5.4) which is not a standard equation solved by Fluent. Therefore, Fluent's in-built numerical solver could not be utilized to solve this equation. Consequently, a parallel, unstructured conjugate gradient squared (CGS) solver with Jacobi pre-conditioning was programmed within the framework of Fluent's UDFs and this custom solver was instead used to solve the volume transport equation. The governing equation was discretized using the unstructured finite-volume procedure (FVM) [37] and backward Euler (or implicit) time discretization. The central difference scheme [37], in conjunction with flux limiters (to be discussed in Section 5.4.3), was used for spatial discretization. Further details of the discretization scheme used, and solver development are provided in Sections 5.5 and 5.6, respectively.

### 5.4.1 Choice of Diffusion Coefficient

As discussed earlier, one of the critical unknown parameters in the volume transport model (Eq. (5.4)) is the diffusion coefficient, $\Gamma_f$, which essentially dictates how quickly the excess volume is redistributed. In the present model, it is assumed that the volume gets redistributed instantaneously. This assumption is supported by the fact that both water and AdBlue are fairly incompressible in both phases, and any expansion (displacement) will be sensed almost instantaneously in the rest of the domain. Instantaneous redistribution implies the use of a very high value of $\Gamma_f$. However, for numerical stability purposes, an excessively large value can be detrimental. Therefore, it was decided to set the diffusion coefficient to be proportional to (some factor of) the thermal diffusivity to ensure that the time scales of the two transport processes involved (volume and energy) are comparable. Additionally, it is important to have a higher diffusion coefficient for the liquid phase and lower diffusion coefficient for the solid phase in order to account for the relative impermeable nature of the solid phase. For the sake of simplicity, a global diffusion coefficient for the liquid phase is defined with respect to the maximum thermal diffusivity in the domain, as shown below.

$$\Gamma_{f,l} = C_f \max(\alpha) \tag{5.9}$$

where $\Gamma_{f,l}$ is the diffusion coefficient for the liquid phase, $\alpha$ is the thermal diffusivity and $C_f$ is a constant of proportionality. Three different values for $C_f$: 2, 10 and 20 were tested for a simple 2D problem. It was found that the value of $C_f$ = 10 provided the best compromise between numerical stability and computational efficiency and was therefore used for all future calculations. Equation (5.4) is then time advanced using a sub-time-stepping procedure to steady-state within each time-step of the energy equation. This ensures that within each time-step of the energy equation, the volume redistribution has reached steady-state. Steady-state for Eq. (5.4) is monitored by calculating the volume-weighted average of $\frac{\partial f}{\partial t}$ over all cells in the computational domain at the end of each sub-time-step and continuing to march forward in time until it decreases below a prescribed threshold.

While Eq. (5.9) is used to specify the diffusion coefficient for the liquid phase, an appropriate value is still required for the solid phase. As it is required for the diffusion coefficient of the solid phase, $\Gamma_{f,s}$, to be lower than that of the liquid phase, it was decided to set $\Gamma_{f,s}$ to be a fractional value of $\Gamma_{f,l}$. This value was decided based on experimentally measured ice dome height data and is detailed in Section 6.4 of Chapter 6. For combined solid-liquid cells, the diffusion coefficient was obtained using a solid-fraction weighted average as shown below.

$$\Gamma_{f,sl} = \Gamma_{f,s}s + \Gamma_{f,l}(1-s) \tag{5.10}$$

### 5.4.2 Treatment of Source Term

The source term in the volume transport equation (Eq. (5.4)) represents the increase in volume caused by the decrease in density during the freezing process. However, the density of the materials is dependent on temperature which, in turn, is affected by the volume fraction distribution. This coupled nature results in a non-linear source term. Attempts to solve Eq. (5.4) after linearizing the source term were unsuccessful due to issues with numerical stability. Therefore, as an alternative numerical implementation strategy, the transport and source operators in Eq. (5.4) are split, with the source term treated first. From mass conservation between two successive time-steps, it can be stated that

$$(\rho_{ls})_i^{n-1}(V_{ls})_i^{n-1} = (\rho_{ls})_i^n(V_{ls})_i^n \tag{5.11}$$

where $i$ is the cell index, $n$ is the time index. Dividing the above equation by the volume of the cell, using the definition of volume fraction as given by Eq. (5.1), and rearranging, the following relation may be obtained.

$$(f)_i^n = \frac{(\rho_{ls})_i^{n-1}}{(\rho_{ls})_i^n}(f)_i^{n-1} \tag{5.12}$$

Using Eq. (5.2), along with Eq. (5.12), provides the initial condition to Eq. (5.4) without the source term. Equation (5.4), without the source term, is then marched forward in time by using sub-time-steps (equal to $\Delta t_s$, and counted using a counter $n_s$), which is typically much smaller than the time-step, $\Delta t$, used to advance the solution to the energy equation.

### 5.4.3 Flux Limiters

If Eq. (5.4) were to be solved "as is" using a central difference scheme, the resulting solution would be smeared and unphysical. For instance, the liquid plus solid would find itself in all parts of the computational domain, including all gas cells, since the governing equation is a diffusion equation. In reality, the liquid plus solid can only fill cells adjacent to the original air-water/ice interface. It cannot appear in the top corners (see Fig. 5.1) of the computational domain. To ensure that the proper physical picture is replicated by a diffusion-based model, the diffusion fluxes must be constrained or limited. To accomplish this, the diffusion flux in Eq. (5.5) is first rewritten as

$$\mathbf{J}_f = -\beta_f \Gamma_f \nabla f_e \tag{5.13}$$

where $\beta_f$ is the so-called *flux limiter*. Here, the following flux limiter is used:

$$\beta_f = \begin{cases} 0 & \text{if } f_1^{n_s} \leq 1 \text{ and } f_2^{n_s} \leq 1 \\ 1 & \text{otherwise} \end{cases} \tag{5.14}$$

where $f_1^{n_s}$ and $f_2^{n_s}$ are volume fractions in two adjacent cells at the previous-sub time-step. The flux limiter is evaluated after the contribution from the source term is considered (as per Eq. (5.12)) Equation (5.14) suggests that there is no exchange of liquid plus solid between two cells that are partially filled. Exchange occurs when only one of them is completely full. In other words, a cell in question must first fill itself before it can donate to adjacent cells. Another important point to note is that the flux limiter uses the volume fraction field from a previous state rather than the current state. The fact that the flux limiter is a function of $f$ itself, makes Eq. (5.4) non-linear. By using previous state information, the equation is linearized in some sense, and small sub-time-steps are necessary to adhere to the linear approximation so that the algorithm does not become unstable.

Based on the preceding discussion, the governing equation for the volume transport model is modified from Eq. (5.4) to instead be given by

$$\frac{\partial f_e}{\partial t} = \nabla.\left(\beta_f \Gamma_f \nabla f_e\right) \tag{5.15}$$

As a final note, it is important to observe that solving Eq. (5.15) provides a solution for $f_e$, the excess volume fraction and not $f$, which is quantity that can be used to track the gas-(solid/liquid) interface. Therefore, an inverse relationship of Eq. (5.2) is required to obtain an appropriate value for $(f)_i^{n_s+1}$ based on $(f_e)_i^{n_s+1}$ and $(f)_i^n$. This relation is given by

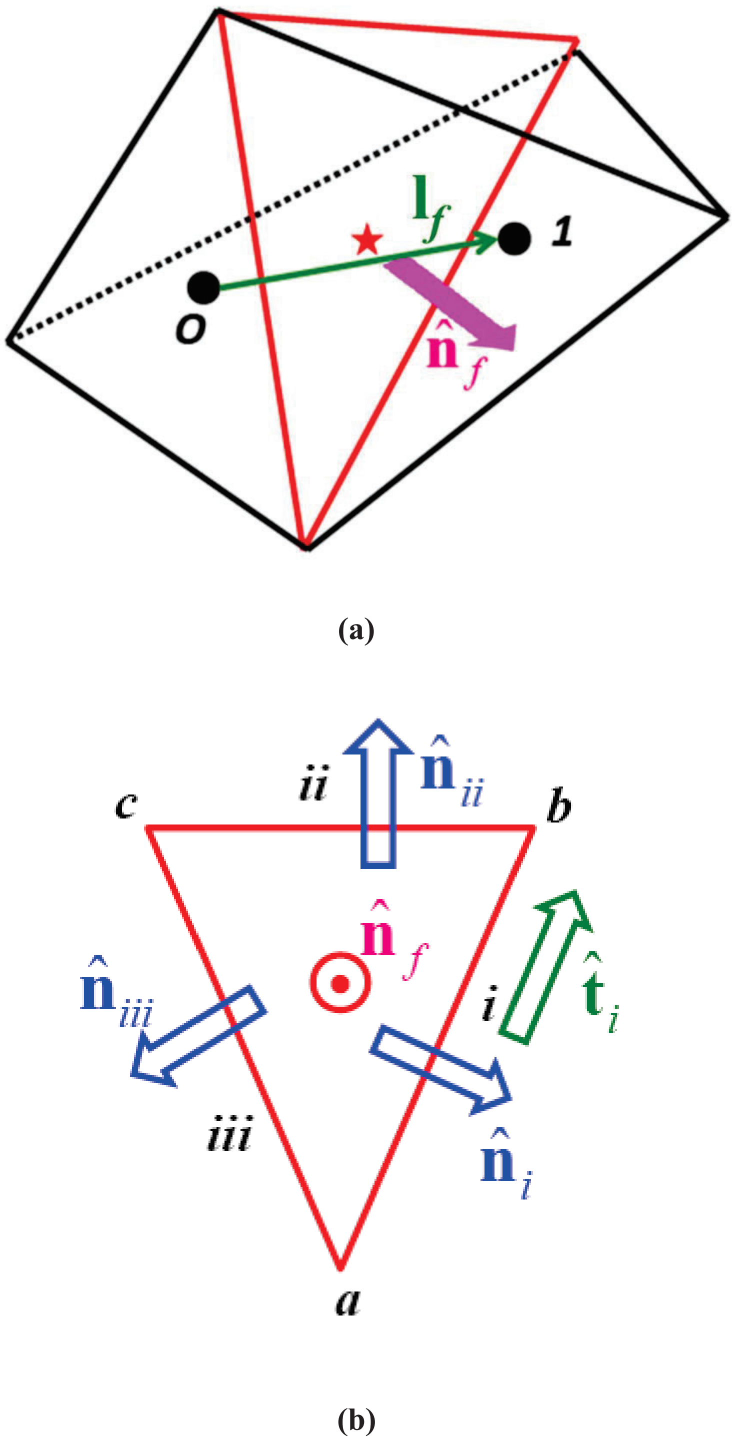
$$(f)_i^{n_s+1} = \begin{cases} (f)_i^{n_s} + (f_e)_i^{n_s+1} & \text{if } (f)_i^{n_s} \leq\ f_{max} \\ f_{max} + (f_e)_i^{n_s+1} & \text{if } f > f_{max} \end{cases} \tag{5.16}$$

## 5.5 Discretization using the Finite Volume Method for unstructured meshes

As mentioned at the beginning of Section 5.4, the unstructured finite volume method was used to discretize the governing equation, with backward Euler (or implicit) method used for time discretization and the central difference scheme used for spatial discretization. A detailed description of these methods can be found in Ref. [37]. Therefore, the derivation of the discretized form of the governing equation (Eq. (5.15)) is omitted in the current discussion and only the final discretized equation yielded by these methods (in 3D) is covered. For ease of discussion and representation, rather than discuss the discretization of the specific governing equation given by Eq. (5.15), the general form of the transient diffusion equation is used, namely

$$\frac{\partial \phi}{\partial t} = \nabla.(\Gamma \nabla \phi) \tag{5.17}$$

where $\Gamma$ is a transport coefficient and $\phi$ is the scalar unknown variable. Consider the 3D stencil shown in Fig. 5.2(a), consisting of two adjacent tetrahedral cells with a common triangular face (marked in red). The two cells are designated Cell O and Cell 1, with $\mathbf{l}_f$ the vector from the cell center from Cell O to the cell center of Cell 1. $\hat{\mathbf{n}}_f$ is the surface normal at the common face pointing outward from Cell O towards Cell 1. Figure 5.2(b) shows a planar projection of the triangular face, such that $\hat{\mathbf{n}}_f$ is pointed out of the plane of the paper. This figure shows the three vertices $a$,$b$,$c$ and the three edges $i$,$ii$,$iii$ with the respective outward pointing edge normals $\hat{\mathbf{n}}_i$, $\hat{\mathbf{n}}_{ii}$, $\hat{\mathbf{n}}_{iii}$. $\boldsymbol{t}_i$ is the edge tangent along edge $i$.



**(a)**

**(b)**

**Figure 5.2: 3D stencil showing (a) two adjacent tetrahedral cells (O and 1) with a common triangular face (b) planar projection of the common triangular face.**

Now, performing unstructured FVM discretization of Eq. (5.17) with the backward Euler method used for time discretization and central difference scheme used for spatial discretization gives to following discretized equation [37]

$$\frac{\phi_O^n - \phi_O^{n-1}}{dt} = \sum_{face=1}^{N_{face,O}} \Gamma_{face} \left( \frac{\phi_{nb(face)}^n - \phi_O^n}{\delta_{face}} \right. \\ \left. - \frac{1}{\delta_{face}} \left( \frac{1}{A_{face}} \sum_{ed=1}^{N_{ed(face)}} \frac{\phi_{a(ed)}^{n-1} + \phi_{b(ed)}^{n-1}}{2} \mathbf{t}_e \right) . \left[ \hat{\mathbf{n}}_f \times \mathbf{l}_f \right] \right) A_{face} \tag{5.18}$$

where $n$ is the time index representing the current and previous time steps and $dt$ is the time step size. To generalize the expression for an arbitrary polyhedral cell, the summations over faces and edges are extended beyond the four faces of Cell O and three edges of the common face as shown in the stencil of Fig. 5.2 to an arbitrary number of faces $N_{face,O}$ for Cell O with each face having $N_{e(face)}$ edges. $\phi_O$ is the cell center value of $\phi$ at Cell O while $\phi_{nb(face)}^n$ is the cell center value of $\phi$ at the cell neighbor of Cell O corresponding to the respective face. $\Gamma_{face}$ is the transport coefficient at the face interpolated from the cell center values and $\delta_{face}$ is defined to be given by $\hat{\mathbf{n}}_f . \mathbf{l}_f$. $\phi_{a(e)}$ and $\phi_{b(e)}$ denote the vertex values of $\phi$ at the two vertices of edge $e$ and are treated explicitly. $\mathbf{t}_e$ is the edge tangent from vertex $a(e)$ to vertex $b(e)$. $A_{face}$ is the area of the current face.

The interpolation of $\Gamma_{face}$ from cell center values is performed using an inverse distance-weighted scheme as shown below.

$$\Gamma_{face} = \frac{\dfrac{\Gamma_O}{d_O} + \dfrac{\Gamma_{nb(face)}}{d_{nb(face)}}}{\dfrac{1}{d_O} + \dfrac{1}{d_{nb(face)}}} \tag{5.19}$$

where $d_O$ and $d_{nb(face)}$ are the respective distances from the cell centroid to the face centroid. For Eq. (5.15), it is important to note that the flux limiter $\beta_f$ is calculated for each face directly and is not, or rather, can not be interpolated from cell center values. Only the diffusion coefficient $\Gamma_f$ is interpolated from cell center values.

As mentioned, the vertex values of $\phi$ are treated explicitly and can therefore be interpolated from cell center values of the immediate neighbors of the vertex. If for a vertex $v$ there are $N_{nb(v)}$ cell neighbors, then the cell-to-vertex interpolation function, $w_{v,i}$, is given by

$$w_{v,i} = \frac{1/d_i}{\sum_{i=1}^{N_{nb(v)}} 1/d_i} \tag{5.20}$$

where $d_i$ is the distance between the vertex $v$ and cell centroid of cell $i$. Once this interpolation function is calculated and stored for each vertex-neighbor cell pair, the vertex value of $\phi$ can be calculated as

$$\phi_v = \sum_{i=1}^{N_{nb(v)}} w_{v,i}\,\phi_i \tag{5.21}$$

The various geometric quantities in Eq. (5.18), such as the normal, tangents and area, must also be calculated. However, at this juncture, it is important to point out the fact that all of the calculations covered thus far must be carried out within the framework provided by Fluent and its UDF functionality. While this does make the calculation of certain geometric quantities, such as the normals, trivial due to the existence of specific macros and functions that calculate and return these quantities, other operations are significantly complicated. For example, there are various quantities in Eq. (5.18) that must be calculated and then stored for each cell, face or node in the computational domain, as appropriate. While Fluent and its UDFs do provide data structures that allow for cell-based and node(vertex)-based storage, there does not exist such a data structure for interior faces. This greatly complicates the storage of quantities such as $\Gamma_{face}$, the transport coefficient at cell faces, or $\beta_f$, the flux limiters at cell faces. In the current work, this limitation was addressed by using the local face indices for each cell and defining multiple cell-based variables offset by the face indices. While this solution was inefficient from a memory perspective due to duplication of saved data, a better alternative was unavailable.

Pending the treatment of the source term in Eq. (5.4) as discussed in Section 5.4.2, and the treatment of the flux limiter, $\beta_f$, using sub-time-stepping (Section 5.4.3), the governing equation to be solved (Eq. (5.15)) becomes quasi-linear. Therefore, upon discretization, the resulting discrete algebraic equations (Eq. (5.18)) will also be linear, which allows for the use of a linear algebraic solver. However, before a solver can be used to obtain a solution, the system of linear algebraic equations must be assembled in the general matrix form:

$$[A][\phi] = [B] \tag{5.22}$$

where $[A]$ is the so-called coefficient matrix, with a size $N \times N$ for a system with $N$ unknowns. $[\phi]$ is a column matrix of length $N$ comprised of the independent variable $\phi$ ($f_e$ for Eq. (5.15)) and $[B]$ is the right-hand side vector. For a particular $\phi_O$, the corresponding equation in the system of linear equations given by Eq. (5.22) can then be represented as

$$A_O\phi_O + \sum_{face=1}^{N_{face,O}} A_{nb(face)}\phi_{nb(face)} = B_O \tag{5.23}$$

where $A_O$ is the diagonal term in the coefficient matrix $[A]$ corresponding to $\phi_O$, $A_{nb(face)}$ represents the off-diagonal terms in $[A]$ corresponding to the neighbors of O, and $B_O$ is the term in $[B]$ corresponding to $\phi_O$. The subsequent goal is to rearrange Eq. (5.18) into the form shown in Eq. (5.23). By performing this rearrangement, and making appropriate substitutions to represent Eq. (5.15), the following relations are obtained.

$$A_{nb(face)} = -\frac{\beta_{f,face}\Gamma_{f,face}A_{face}}{\delta_{face}} \tag{5.24}$$

$$A_O = \frac{1}{dt_s} - \sum_{face=1}^{N_{face,O}} A_{nb(face)} \tag{5.25}$$

$$\mathbf{J}_{tang(face)} = \frac{\beta_{f,face}\Gamma_{f,face}}{\delta_{face}}\left(\sum_{ed=1}^{N_{ed(face)}} \frac{(f_e)_{a(ed)}^{n_s-1} + (f_e)_{b(ed)}^{n_s-1}}{2}\mathbf{t}_e\right).\left[\hat{\mathbf{n}}_f \times \mathbf{l}_f\right] \tag{5.26}$$

$$B_O = \frac{f_e^{n_s-1}}{dt_s} - \sum_{face=1}^{N_{face,O}} \mathbf{J}_{tang(face)} \tag{5.27}$$

where $\mathbf{J}_{tang(face)}$ is the tangential flux term at the cell face, $dt_s$ is the sub-time-step size and $n_s$ is the sub-time index. These equations represent the calculations required to setup the coefficient matrix $[A]$ and the right-hand side vector $[B]$ in preparation for a linear algebraic solver. The details of the linear algebraic solver used in the current work are provided in the following section.

The boundary conditions for the volume transport equation are zero flux boundary conditions, as the walls of the tanks do not allow for any flow through them. This is easily implemented by setting the diffusion coefficient at boundary faces equal to zero.

### 5.6 Conjugate Gradient Squared (CGS) Solver – Parallel Implementation

As mentioned earlier in Section 5.4, Fluent's in-built numerical solver could not be used to solve the governing equation for the volume transport model. Therefore, a custom linear algebraic solver was written within Fluent's UDFs. This solver used the CGS method, and the algorithm for this method was taken from Ref. [37]. However, the algorithm provided in this reference can only be used for serial calculations. Therefore, the algorithm was modified to be usable in parallel computations. While there are several possible strategies to parallelize a solver, in the current work the parallel implementation of the CGS method goes into the linear algebraic solver and breaks up the various matrix and vector operations between parallel partitions. For example, consider a vector product that shows up multiple times in the CGS algorithm: $[R]^T[R]$, where $[R]$ is the so-called residual vector. For this example, let $[R] = [r_1 \quad r_2 \quad r_3 \quad r_4]^T$. In serial, the product $[R]^T[R]$ is trivial to evaluate, as shown in Eq. (5.28).

$$[R]^T[R] = [r_1 \quad r_2 \quad r_3 \quad r_4]\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \sum_i r_i^2 \tag{5.28}$$

Now, consider a parallel computation with two partitions such that each parallel partition only has half of the elements of $[R]$, i.e., in partition 1, $[R] = [r_1 \quad r_2 \quad 0 \quad 0]^T$ and in partition 2, $[R] = [0 \quad 0 \quad r_3 \quad r_4]^T$. If the product $[R]^T[R]$ is now evaluated in each partition, the following results are obtained.

$$([R]^T[R])_{P1} = [r_1 \quad r_2 \quad 0 \quad 0]\begin{bmatrix} r_1 \\ r_2 \\ 0 \\ 0 \end{bmatrix} = r_1^2 + r_2^2 \tag{5.29}$$

$$([R]^T[R])_{P2} = [0 \quad 0 \quad r_3 \quad r_4]\begin{bmatrix} 0 \\ 0 \\ r_3 \\ r_4 \end{bmatrix} = r_3^2 + r_4^2 \tag{5.30}$$

where $([R]^T[R])_{P1}$ and $([R]^T[R])_{P2}$ are the product $[R]^T[R]$ as evaluated by partition 1 and partition 2, respectively. It is clear from these results that each partition performs a partial computation of the required product and to obtain the complete product, the results from the computation in each partition must be summed together in a global summation operation over all partitions, called global sum for short, as shown in Eq. (5.31).

$$([R]^T[R])_{global} = ([R]^T[R])_{P1} + ([R]^T[R])_{P2} = \sum_i r_i^2 \tag{5.31}$$

This approach is extended to multiple partitions in a similar manner and is applied to all similar vector operations in the CGS algorithm. When the CGS solver was written, it was important to identify and account for all such operations in the algorithm.

Matrix operations also pose a challenge in the form of data access and communication. To illustrate, consider the product $[\mathrm{A}][\phi] = \mathrm{A}_{\mathrm{O}}\phi_O + \sum_{nb} A_{nb}\phi_{nb}$ that is evaluated at several points in the CGS algorithm. Assume that this product is to be evaluated for the stencil shown in Fig. 5.3, where the cells O, 1 and 2 are in one partition and cell 3 is in a separate partition, with the partition boundary marked in yellow. In this scenario, when this product is evaluated for cell O in the first partition, it does not have

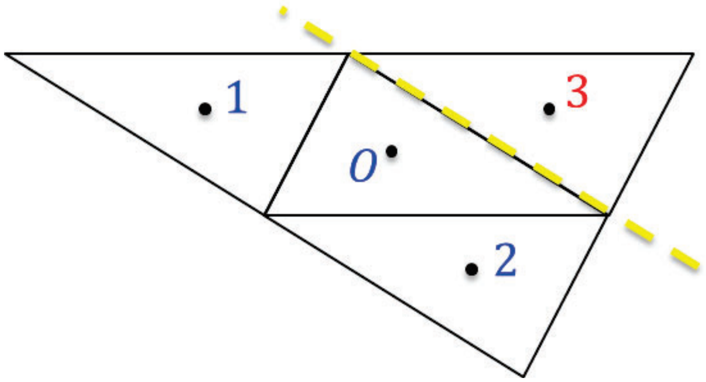


**Figure 5.3: A 2D stencil showing four triangular cells distributed between two partitions, with the partition boundary shown by the dotted yellow line.**

access to the data in cell 3, as it is in a different partition. The value of $\phi_3$, which is stored in the second partition, must be communicated from the second partition to the first partition so that the required product can be evaluated. This is performed using a parallel exchange command (EXCHANGE_UDMI) provided by Fluent's UDFs, which communicates the value of specified variables between all partitions. From a computational efficiency standpoint, it is important to identify exactly which variables need to be communicated between partitions at which points of the solver. These variables must then be synced only as and when needed, as excessive communication between partitions increases the communication overheads. Over the course of the CGS algorithm, there are several such sync points. These are highlighted in the description of the parallel CGS solver algorithm in Section 5.6.1.

Lastly, a backup Gauss-Seidel (GS) solver was also written and implemented. This solver is called when the CGS solver fails to converge and was implemented as a safety net since the CGS algorithm, in general, is erratic during the initial stages of convergence [37]. When the GS solver is called, the solution is reset to the initial condition and the $[A]$ and $[B]$ matrices are reused from when they were set up for the CGS solver.

### 5.6.1 Parallel CGS Solver Algorithm

The algorithm for the parallel CGS solver used to solve the discretized system of equations $[A][\phi] = [B]$ (Eq. (5.22)) is listed below. A Jacobi preconditioner is also used prior to the start of the CGS algorithm. Using this preconditioner entails scaling each of the linear algebraic equations in Eq. (5.22) by its respective diagonal element prior to executing the CGS algorithm, i.e., in Eq. (5.23), $A_{nb(face)} = A_{nb(face)}/A_O$, $B_O = B_O/A_O$ and $A_O = 1$.

As described in the Section 6.5, this algorithm is a modified version of the serial CGS algorithm provided in Ref. [37]. Steps at which variable exchange between parallel partitions (*parallel exchange*) and global summations across parallel partitions (*global sum*) occur are highlighted in the algorithm.

1) Guess the initial value of $\phi$ at all cells. These values are denoted as $[\phi^{(0)}]$, where the superscript denotes the time index. *Parallel exchange* $[\phi^{(0)}]$.
2) Compute the initial residual vector: $[R^{(0)}] = [B] - [A][\phi^{(0)}]$.
3) Initialize the direction vector and the conjugate direction vector, respectively: $[D^{(0)}] = 0$, $[D^{(0)}]^* = 0$. Additionally, initialize $\gamma_0 = 1$ and $[G^{(0)}] = 0$.
4) Compute $\gamma = [R^{(0)}]^T\,[R^{(n)}]$ (involves g*lobal sum*), $\zeta^{(n+1)} = \gamma/\gamma_0$.
5) Update conjugate search direction vector: $[D^{(n+1)}]^* = [R^{(n)}] + \zeta^{(n+1)}[G^{(n)}]$.
6) Update search direction vector: $[D^{(n+1)}] = [D^{(n+1)}]^* + \zeta^{(n+1)}\{[G^{(n)}] + \zeta^{(n+1)}[D^{(n)}]\}$. *Parallel exchange* $[D^{(n+1)}]$.
7) Compute $\xi^{(n+1)} = \frac{\gamma}{[R^{(0)}]^T[A][D^{(n)}]}$. The product $[R^{(0)}]^T[A][D^{(n)}]$ requires a *global sum*.
8) Compute $[G^{(n+1)}] = [D^{(n+1)}]^* - \xi^{(n+1)}[A][D^{(n)}]$.
9) Update solution using $[\phi^{(n+1)}] = [\phi^{(n+1)}] + \alpha^{(n+1)}\{[G^{(n+1)}] + [D^{(n+1)}]^*\}$. *Parallel exchange* $[\phi^{(n+1)}]$.
10) Compute the new residual vector, $[R^{(n+1)}] = [Q] - [A][\phi^{(n+1)}]$.

11) Calculate the L$^2$Norm of the residual vector, $R2^{(n+1)} = \sqrt{[R^{(n+1)}]^T\,[R^{(n+1)}]}$ , and the normalized residual, $(R2^*)^{(n+1)} = R2^{(n+1)}/(R2_{max})$. Calculation of $R2$ involves a g*lobal sum*.

12) Monitor convergence, i.e., check if $R2^{(n+1)} < \varepsilon_{abs}$ or $(R2^*)^{(n+1)} < \varepsilon_{rel}$, where $\varepsilon_{abs}$ and $\varepsilon_{rel}$ are the absolute and relative convergence criteria, respectively. If YES, then stop iterations and proceed to Step 13. If NO, return to Step 4 and start the next iteration.

13) Check if CGS solver has converged correctly, i.e., check if $R2^{(n+1)}/\ R2^{(1)} < 10^2$. If YES, then exit. If NO, reset solution by setting $\left[\phi^{(n+1)}\right] = \left[\phi^{(0)}\right]$ and use backup Gauss-Seidel (GS) solver to solve $[A][\phi] = [B]$.

## 5.7 Volume Transport Equation Solver Algorithm

The algorithm used to obtain the volume fraction distribution by solving the volume transport equation (Eq. (5.4)) is listed below. Once again, steps at which variable exchange between parallel partitions (*parallel exchange*) and global summations across parallel partitions (*global sum*) occur are highlighted in the algorithm. The entirety of this code can be found in Appendix B.

1) The diffusion coefficient, $\Gamma_f$, is calculated for all cells and faces as per Eq. (5.10) and Eq. (5.19), respectively. The diffusion coefficient at boundary faces is set to zero.

2) The minimum and maximum sub-time steps are defined based on the energy equation time-step size. The initial sub-time-step size is set to the minimum value.

3) Equation (5.12) is used to account for the contribution from the source term in Eq. (5.4) and update the initial condition for the volume fraction, $f_i^{n_s}$, such that $f_i^{n_s} = f_i^n$ where $i$ is the cell index, $n_s$ is the sub-time index and $n$ is the time index. The initial condition for the excess volume fraction, $(f_e)_i^{n_s}$, is calculated from $f_i^{n_s}$ using Eq. (5.2). *Parallel exchange* $f_i^{n_s}$ and $(f_e)_i^{n_s}$.

4) The vertex (nodal) values of $f_e^n$ are calculated using the relationships given by Eq. (5.20) and Eq. (5.21).

5) The flux limiters, $\beta_f$, are calculated based on Eq. (5.14) at each face in the computational domain.

6) The coefficient matrix $[A]$ and right-hand side vector $[B]$ for the discretized form of Eq. (5.15) are set up using the relationships given by Eq. (5.24) through Eq. (5.27).

7) The excess volume fraction distribution at the next sub-time-step, $(f_e)_i^{n_s+1}$, is obtained by solving the using the parallel CGS solver described in Section 5.6.1 to solve the system of algebraic equations described in the previous step. *Parallel exchange* $(f_e)_i^{n_s+1}$

8) The volume fraction distribution at the next sub-time-step, $f_i^{n_s+1}$, is obtained using the relation provided by Eq. (5.16). *Parallel exchange* $f_i^{n_s+1}$.

9) Calculate the volume-weighted average of $\left(\frac{\partial f}{\partial t_s}\right)_i = \frac{f_i^{n_s+1} - f_i^{n_s}}{\Delta t_s}$, $\left(\frac{\partial f}{\partial t_s}\right)_{avg}$, over all cells. Calculating $\left(\frac{\partial f}{\partial t_s}\right)_{avg}$ involves *global sum*.

10) Check for steady-state, i.e., is $\left(\frac{\partial f}{\partial t_s}\right)_{avg} < \varepsilon_{SS}$, where $\varepsilon_{SS}$ is the steady-state threshold criterion. If YES, end sub-time-stepping and proceed to next step. If NO, increment sub-time-size and count, and return to Step 5 to start next sub-time-step.

11) Volume fraction distribution at next time-step, $f_i^{n+1}$, is updated as $f_i^{n+1} = f_i^{n_s+1}$.

## 5.8 Overall Solution Algorithm

The modified overall solution algorithm (including the reduced natural convection and volume transport model) that was employed is as follows:

1) The temperature, $T$, and volume fraction, $f$, are initialized in all cells. These are denoted by $T_i^n$ and $f_i^n$, respectively, where $i$ is the cell number, while $n$ is the time index.

2) The solid fraction in each cell $i$ and at time index $n$, $s_i^n$, is computed from the temperature using the relationship given by Eq. (3.24). The density and all other thermophysical properties are calculated at this time instant as described in Section 5.3 and stored.

3) The temperature at the next time-step is determined by solving the discretized form of the energy equation (Eq. 3.1) using the numerical solver provided by ANSYS-Fluent$^{\text{TM}}$, yields a solution of the temperature field at the next time index, i.e., $T_i^{n+1}$.

4) The volume fraction at the next time-step is determined by solving the volume transport equation (Eq. (5.15)) as per the algorithm described in Section 5.7. This yields $f_i^{n+1}$.

5) The solid fraction, $s_i^{n+1}$, and other thermophysical properties are updated.

6) The initial condition set in Step 1 is replaced by the new values, and the solution proceeds to the next time-step, i.e., Steps 1–7 are repeated.

## 5.9 Summary

The volume transport model presented in this chapter follows the fundamental premise of the VOF algorithm but allows for the tracking of the gas-(solid/liquid) interface in the absence of a flow field, as required by the reduced natural convection model proposed earlier in Chapter 3. This is achieved by solving an additional classical diffusion equation for the excess volume fraction, as described by Eq. (5.4). The solution of this equation on an unstructured mesh of arbitrary topology with the inclusion of parallel processing required development of a numerical solver from the ground up and constitutes a major fraction of this work. The following chapter covers the details of the rudimentary ice dome height data collected as part of the experimental study detailed in Chapter 4, and how this ice dome height data is then used to determine appropriate values for the porosity, $p$, of the resulting solid and the solid diffusion coefficient, $\Gamma_{f,s}$. The validation studies presented in Chapter 4 are then repeated with the inclusion of ice expansion and dome formation.

# CHAPTER 6. VOLUME TRANSPORT MODEL - RESULTS

## 6.1 Introduction

In the previous chapter, details were provided for the formulation and implementation of a volume transport model, which tracks the movement of the gas-(solid/liquid) interface while also being compatible with the reduced natural convection model proposed in Chapter 3. However, two parameters of the volume transport model were yet to be determined. The first of these is the ratio between the rate at which the expanded volume redistributes within the solid and liquid, which, in our volume transport model, is characterized by the diffusion coefficient, $\Gamma_f$, while the second is the porosity of the ice dome, $p$. These quantities were estimated using rudimentary experimental ice dome height data collected by engineers at the Ford Motor Company during the experimental study detailed in Chapter 4. Details of the experimental ice dome height data are provided in the current chapter. Subsequently, the current chapter covers the procedure used to determine the appropriate porosity and diffusivity values. Finally, the chapter presents a study on the formation of the ice dome and its effects on the temperature predictions at thermocouple locations within the tank using the same experimental temperature data detailed in Chapter 4.

### 6.2 Experimental Study

As mentioned earlier in Chapter 4, engineers at the Ford Motor Company were responsible for performing the experiments for the current study and the results obtained from these experiments are presented here to supplement the modeling studies. As Chapter 4 focused on validating the reduced natural convection model and did not involve the formation of the ice dome, the details provided at that time in Section 4.2 were restricted to the collection of temperature data at various thermocouple locations. While this same temperature data is reused in the current chapter to study and validate the effects of ice dome formation on temperature predictions, further details of the experimental study pertaining to the measurement of ice dome heights are provided in the current section.

The measurement of ice dome height was complicated due to the nature of the experimental setup. As discussed in Section 4.2, the thermocouples used for temperature measurements within the liquid volume were attached to plastic rods suspended from the lid of the tank. Over the course of the freezing experiment, these rods would slowly become embedded within the freezing liquid. At the conclusion of the experiment, this prevented the complete removal of the rods and consequently, the lid of the tank, prohibiting detailed ice dome shape measurements from the top of the tank. Furthermore, the walls of the tank were opaque, thereby preventing any profile measurements of the ice dome from the side of the tank. Therefore, only rudimentary ice dome height could be collected as part of this experimental study. The height of the final ice dome was measured posteriori at two locations: (1) the center of the dome/tank and (2) the wall of the tank. Figure 6.1 illustrates these measurements for one of the experimental runs. It is important to note that the shape

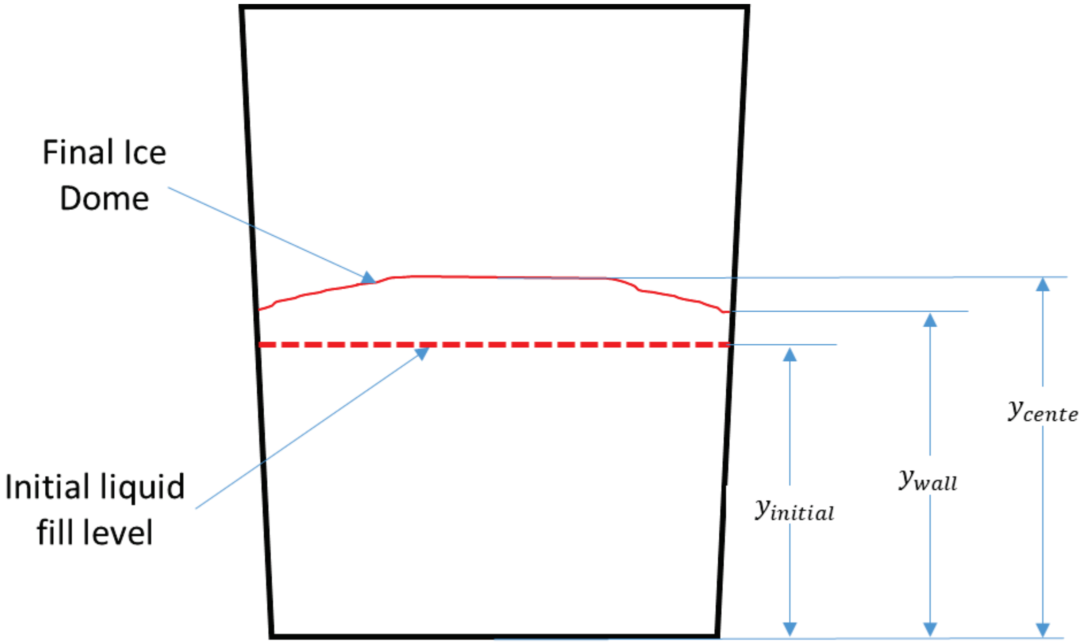


**Figure 6.1: Schematic showing the locations of final ice dome height measurements for a 50% filled tank.**

of the ice dome depicted (in red) in this figure is purely a qualitative sketch done for illustration purposes and does not represent any actual ice dome shape measurements.

While temperature measurements were carried out for seven experimental runs, including one calibration run and six validation runs (refer Table 4.1), ice dome height measurements were carried out only for a single run, specifically, the 50% fill level case of water. As depicted in Fig. 6.1, the height of the ice dome at the walls of the tank is denoted as $y_{wall}$ while the height at the center is denoted as $y_{mid}$. The initial height of the liquid is denoted as $y_{initial}$. While the ice dome height measurements were made with respect to the bottom of the tank, from an analysis perspective it is more useful to consider the percentage increase in the heights of the ice dome as measured with respect to the initial fill level of the liquid. This is because it is only the solid above the initial fill level (the ice dome) that represents the increase in volume due to density decrease. To this end, these adjusted heights are represented by $\Delta y_{mid}$ and $\Delta y_{wall}$, with the percentage increases represented by $\Delta y_{mid}\%$ and $\Delta y_{wall}\%$ and are given by Eqns. (6.1) through (6.4), respectively. Lastly, to represent the shape of the ice dome, it is useful to consider the "steepness" of the ice dome, represented by the ratio of the two newly defined heights. All of these quantities are reported in Table 6.1.

$$\Delta y_{mid} = y_{mid} - y_{initial} \tag{6.1}$$

$$\Delta y_{wall} = y_{wall} - y_{initial} \tag{6.2}$$

$$\Delta y_{mid}\% = \Delta y_{mid}/\ y_{initial} \tag{6.3}$$

$$\Delta y_{wall}\% = \Delta y_{wall}/y_{initial} \tag{6.4}$$

**Table 6.1: Ice dome height measurements for the 50% fill level case of water.**

| **Liquid** | **Fill level** | $\Delta y_{mid}$ **(mm)** | $\Delta y_{wall}$ **(mm)** | $\Delta y_{mid}/\Delta y_{wall}$ |
|---|---|---|---|---|
| **Water** | 50% | 56 | 19 | 2.95 |

### 6.3 Simulation setup details

As was done during the validation studies carried out for the reduced natural convection model in Chapter 4, 3D simulations were set up to study the results predicted but now, with the inclusion of volume transport in conjunction with the reduced natural convection for the various experimental cases. The setup for these simulations with the inclusion of volume transport is nearly identical to the simulation setup details as covered in Section 4.6. The minor difference arising from the inclusion of the volume transport model, and the setup details unique to the model are covered subsequently.

As explained previously in Chapter 5, the reduced natural convection model was implemented in ANSYS-Fluent$^{TM}$ and used the in-built numerical solver to solve the energy equation. However, the governing equation for the volume transport model (Eq. 5.15) was solved using a custom parallel, unstructured conjugate gradient squared (CGS) solver (Section 5.6), which was implemented using a User-Defined Function (UDF) as part of this work. As the volume transport equation requires an appropriate initial condition, as part of the initialization procedure, an extra UDF is used to initialize the volume fraction distribution based on the fill level of each of the cases.

The volume transport equation was solved using a variable sub time-step size starting at 0.02 s and doubling every 1000 sub-time steps up to a maximum size of 0.1 s. These values were chosen after numerical experimentation with various combinations to obtain a balance between computation speed, accuracy, and numerical stability. Additionally, the volume transport equation was only solved once every 100 time-steps of the energy equation, which was also deemed to be appropriate after comparing solutions obtained by varying this number. The tolerance used to determine the steady-state of the volume transport equation was $10^{-8}$. The parallel CGS solver for the volume fraction equation used an absolute tolerance of $10^{-10}$ and a relative tolerance of $10^{-6}$ for terminating iterations. While the previous simulations covered in Section 4.6 were run using the computing facilities offered at the Ohio Supercomputer Center (OSC), the current simulations involving the volume transport model were instead run at the Ford Motor Company's HPC (High Performance Computing) facility. This facility allowed for the use of 192 processors in parallel for the simulations. However, the 50% fill level case of water was repeated at OSC to ensure that a meaningful comparison between the runtimes of the simulations with and without the volume transport model enabled was possible.

While the same material thermophysical properties were used for the current simulations as before, the calculation of these properties for the new mixed three phase cells had to be accounted for. Therefore, the corresponding material property UDFs were updated to use the appropriate equations as described in Section 5.3 (Eqns. (5.6), (5.7) and (5.8)) instead.

To reiterate, the setup for the part of the simulation involving the energy equation and the reduced natural convection model is unchanged from the setup details provided previously in Section 4.5. This includes the values of the calibrated wall heat transfer coefficients (($h_{top}, h_{side}$ and $h_{bot}$ ) = (4, 35, 30 W/m$^2$-K)) used for the convective boundary conditions and the calibrated model constants ($c_w, c_a, c_u$) of the reduced natural convection model. Lastly, the same meshes used in the study in Chapter 4 (example shown in Fig. 4.8) were used for the simulations in the current study.

However, before any simulations utilizing the proposed volume transport model could be set up, appropriate values for the ratio between the diffusivity of the solid and liquid, and the porosity of the ice dome, $p$, were yet to be determined. The following sections detail the process used to obtain these values.

## 6.4 Estimation of Diffusion Coefficient

The values for the volume transport diffusion coefficient, $\Gamma_f$, for the solid and liquid phases represent the rate at which volume can be transported through these phases. Higher values of diffusion coefficient would allow for volume to diffuse faster through the material. As solid ice is rigid and does not "flow" easily, it was concluded that its diffusion coefficient must be lower than that of the liquid phase. As the diffusion coefficient for the liquid phase was defined based on thermal diffusivity as given by Eq. (5.9), the diffusion coefficient for the solid phase was then defined to be a fractional value of the liquid diffusion coefficient. Simulations of the 50% fill level water case were then set up, as described in the previous section, using different diffusion coefficient ratios, $\Gamma_{f,s}/\Gamma_{f,l}$, and the heights of the predicted ice domes were recorded. This specific case was chosen based on two factors: (1) the thermophysical properties of water are well-documented in the literature, and (2) the 50% fill level case is the only case for which reliable ice-dome height data is available. As the porosity of the ice dome was yet to be established at this point, it was assumed that the ice dome was completely non-porous, i.e., $p = 0$. Consequently, it was observed that varying the diffusion coefficients did not have a significant effect on the average heights of the ice dome (defined as the average between $\Delta y_{mid}$ and $\Delta y_{wall}$), but instead affected the shape (height ratio) of the ice dome. These results are summarized in Table 6.2 below. The first row lists the experimentally measured data, while the second column in the table lists the various diffusion coefficient ratios trialed. A diffusion coefficient ratio of 1 was first utilized to establish a baseline value for comparison. As seen in Table 6.2, it results in a nearly flat dome, with a height ratio of only 1.26. This is significantly different from the experimentally observed height ratio of 2.95. Following this, two further diffusion coefficient ratios of 1:10 and 1:5 were tested. These two cases resulted in domes whose average heights were very close to the previous case, but drastically different dome shapes. The 1:10 case resulted in a steeper dome than the experimentally observed dome, while the 1:5 case resulted in a flatter dome. Based on these results, a final case with a diffusion coefficient ratio of 1:6.25 was tested. This case also did not show a significant change in the average height of the ice dome but instead was very close to the experimentally observed ice dome height ratio. Therefore, it was decided to use this diffusion coefficient ratio of 1:6.25 for all future simulations involving the volume transport model (all fill levels and both liquids).

One final observation from the data in Table 6.2 is that the average height predicted in all 4 simulation cases is significantly lower than the height observed in the experiment. This is attributed to the assumption of zero porosity, as will be explained in the following section.

**Table 6.2: List of adjusted ice dome height measurements for experimental runs performed with different combinations of fill level and working liquid.**

| Experiment /Simulation? | $\Gamma_{f,s}$: $\Gamma_{f,l}$ | $\Delta y_{mid}$ (mm) | $\Delta y_{wall}$ (mm) | $\frac{\Delta y_{mid}+\Delta y_{wall}}{2}$ (mm) | $\Delta y_{mid}$ $/\Delta y_{wall}$ |
|---|---|---|---|---|---|
| **Exp** | - | 56 | 19 | 37.5 | 2.95 |
| **Sim** | 1:1 | 17.7 | 14 | 15.85 | 1.26 |
| **Sim** | 1:10 | 28.1 | 7.7 | 17.9 | 3.65 |
| **Sim** | 1:5 | 24.7 | 9.6 | 17.15 | 2.57 |
| **Sim** | 1:6.25 | 25.7 | 9 | 17.35 | 2.86 |

### 6.5 Ice Dome Porosity

Originally, it was assumed that the porosity of the ice dome was zero. When simulations were run using this assumption, it was found that the predicted ice domes were significantly smaller in height than what was experimentally observed. This prompted a simple investigation into ice dome heights obtained in the experiments. Consider the 50% fill level case of water for examination. An average height for the ice block can be calculated by taking the average of the two heights reported in Table 6.1. While there are other ways one could obtain an average ice height (such as assuming a certain shape or profile for the ice surface), each of these methods would involve their own assumptions and approximations. As a rough estimate of the average height will be sufficient for the current analysis, the simplest method available was chosen. This average height can then be used to calculate an approximate final volume for the ice block using the dimensions of the tank and is described next. The radius of the tank, $r_{tank}$, at a distance $y$ from the bottom of the tank is given by Eq. (6.5). This relation can then be used to obtain an expression for a volume within the tank, $V_y$, up to the same height, $y$, as shown in Eq. (6.6)

$$r_{tank}(y) = \frac{r_{top} - r_{bot}}{y_{max}} y + r_{bot} \tag{6.5}$$

$$V_y(y) = \int_0^y \pi (r_{tank})^2 \, dy \tag{6.6}$$

where $r_{top}$ and $r_{bot}$ are the radii of the top and bottom of the tank, and $y_{max}$ is the height of the tank. From the initial fill height and using Eq. (6.6), the initial volume of liquid water, $V_{initial}$, is calculated. Similarly, an approximate final volume for the ice block, $V_{final}$, can be estimated based on the average final height. As the mass of the water within the tank remains constant, the final ice density required to obtain the observed final volume increase can be estimated. Based on the actual dimensions of the tank, the ratio between the initial and final volumes is obtained as

$$\frac{V_{final}}{V_{inital}} = 1.21 \tag{6.7}$$

As the mass of the water in the tank, $m$, remains constant during the freezing process, the left-hand side of Eq. (6.7) can be multiplied and divided by the mass of the water to cast this equation in terms of density instead. This gives

$$\frac{\frac{V_{final}}{m}}{\frac{V_{inital}}{m}} = \frac{\frac{1}{\rho_{final}}}{\frac{1}{\rho_{intial}}} = \frac{\rho_{intial}}{\rho_{final}} = 1.21 \tag{6.8}$$

where $\rho_{intial}$ is the average initial density of the liquid water and $\rho_{final}$ is the average final density of the solid ice block. Rearranging Eq. (6.8) then gives

$$\rho_{final} = \frac{\rho_{intial}}{1.21} \tag{6.9}$$

Based on the measured thermocouple data (refer Fig. 4.5(a)), it can be assumed the water is at an average initial temperature of 15°C. Using the temperature-density relationship obtained from literature [46], the density of water at this temperature can be obtained to be 998.8844 kg/m$^3$, which is assumed to be $\rho_{intial}$. Equation (6.9) then results in a $\rho_{final}$ value of 827.1699 kg/m$^3$. This value of density is significantly lower than the density of ice as reported in literature, which is close to 923 kg/m$^3$ at the steady state temperature of -38°C. While the above calculation for $\rho_{final}$ is an approximate estimate, the assumptions involved are unlikely to account for the nearly 100 kg/m$^3$ difference in densities observed.

A potential explanation for this observed lower density of solid ice is that the ice dome is not completely solid and is instead porous in nature. The presence of tiny air pockets trapped within the ice dome would decrease its apparent density and therefore cause its volume to increase beyond what would be expected from the phase change process alone. Therefore, the volume transport model was adapted to account for potential porosity of the ice dome, as detailed in Section 5.2.

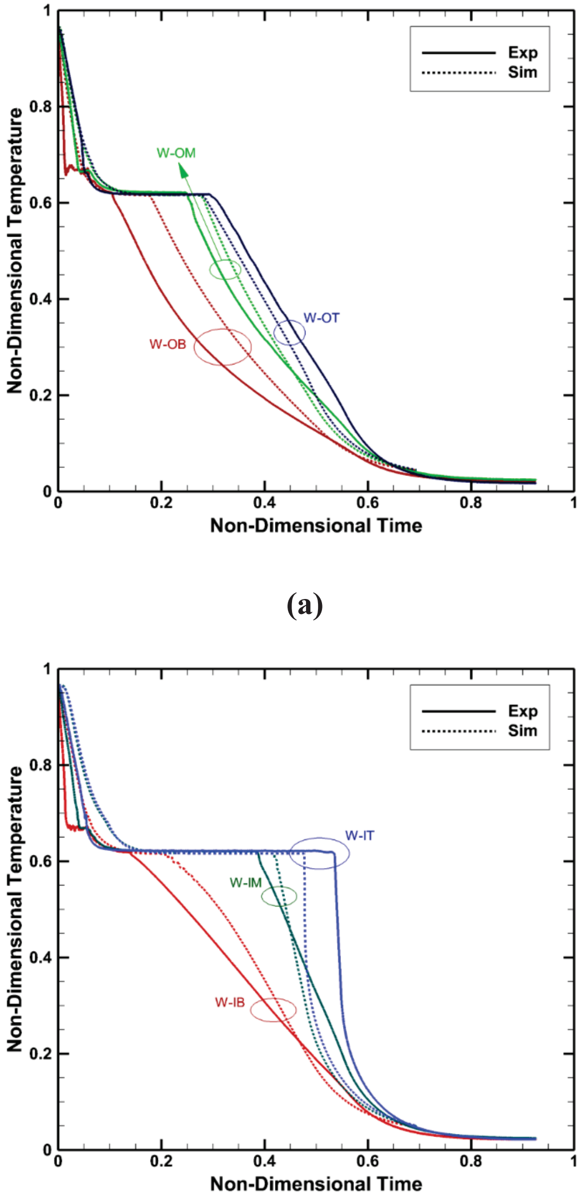
Using a slight variation of the prior calculations, where $V_{final}$ is expressed as the sum of the initial volume and the volume of the ice dome, the porosity of the ice dome can be estimated to be close to 0.4 for this case. However, this is only a very rough approximation for the porosity from a single dataset. For subsequent testing of the model, a porosity value of 0.5 was utilized for all future simulations involving the volume transport model (all fill levels and both liquids), with the hope that future experiments will shed more light on an appropriate value to use.

### 6.6 Simulation Results

As detailed earlier in Section 6.2, six experiment runs (two working liquids at 3 different fill levels) collected temperature data, but ice dome height data was collected for only one case (50% fill level of water). The temperature data from these runs, alongside the simulation results provided during the validation studies of the reduced natural convection model in Section 4.7, are used in the current section to study the effect incorporating the formation of the ice dome has on the predicted temperature results. Therefore, the same setups as described in the validation study of Section 4.7 are used here (refer Fig. 4.9). The predicted ice dome heights in the 50% fill level case of water are also compared to the experimentally observed values.

### 6.6.1 Results with water as working liquid

Figure 6.2 shows the temperature vs. time curves for the six water thermocouples for the 50% fill level case with the volume transport model (ice dome formation) enabled. The RMS errors in this case are very similar to those obtained previously without the volume transport model (refer Section 4.7.1) and are tabulated in Table 6.3. As before, the largest RMS error is for the W-IT thermocouple, with a value of 7.85°C. This is lower than the previous error of 8.21°C obtained when there was no ice dome. Similarly, the W-OT, W-OM and W-IM thermocouples all show a slight decrease in the RMS error at these locations, indicating an improvement in agreement with experimental data. While the two bottom thermocouples show a slight increase in the RMS errors, these are lower in magnitude compared to the improvements for the other four thermocouple locations. This makes physical sense, as the effect of the ice dome on the heat transfer process will be strongest near the ice dome itself at the top of original fill level. Therefore, the largest change in the RMS error is also observed at the W-IT thermocouple, with a decrease of 0.36°C. The freezing sequence of the thermocouples remains the same as before: W-S → W-OB → W-IB → W-OM → W-OT → W-IM → W-IT, which agrees with experimental observations.



**(a)**

**(b)**

**Figure 6.2: Temperature vs. time curves with ice dome formation enabled for the 50% filled tank of water: (a) outer thermocouples, (b) inner thermocouples.**

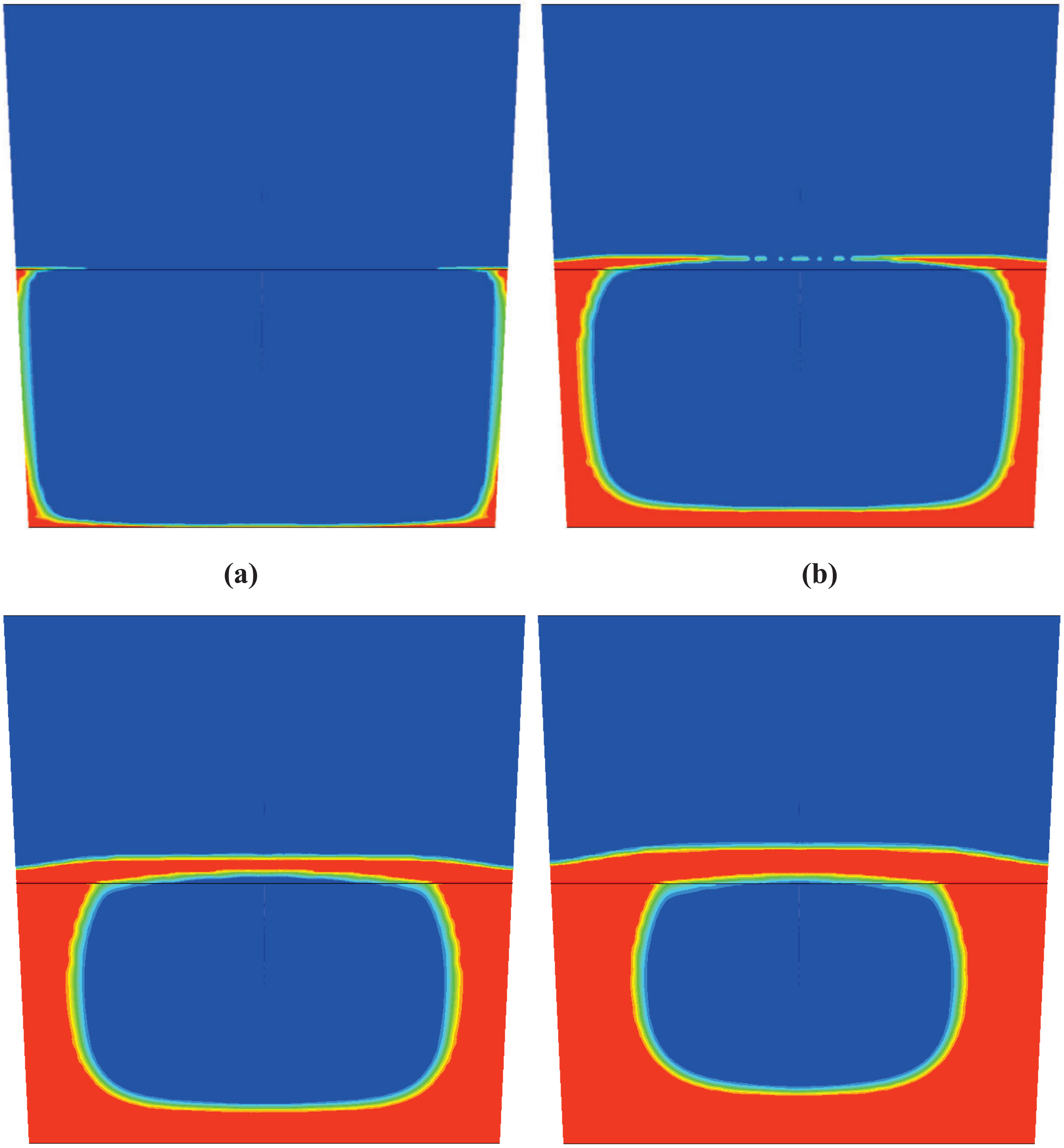
**Table 6.3: Average (RMS) error (°C) between the simulation and experiment for 50% fill level with water.**

| Thermocouple | RMS error | |
|---|---|---|
| | Without Ice Dome | With Ice Dome |
| W-OB | 4.89 | 4.92 |
| W-OM | 2.25 | 2.10 |
| W-OT | 2.25 | 2.11 |
| W-IB | 3.45 | 3.61 |
| W-IM | 3.39 | 3.32 |
| W-IT | 8.21 | 7.85 |

However, the expansion of the ice dome can have other important consequences. Figure 6.3 shows contours depicting the evolution of the water-ice front over the course of the freezing process. While the movement of the front underneath the initial air-(water/ice) interface is similar to what was reported before (refer Fig. 4.15), there are certain important differences. Primarily, the notching seen at the start of the solidification process (Fig. 6.3 (b)) occurs noticeably above the initial liquid fill level. This causes the top of liquid bubble within the solid to start at a point above the initial fill level, and over the course of the freezing process leads to a final liquid bubble that is below this level. This difference in the progression of the freezing front is related to the improvements in temperature predictions at the top thermocouples as reported in Table 6.3.

As these figures demonstrate, the ice dome clearly rises a significant height above the initial fill level. Therefore, tracking the ice dome would be necessary if it is important that it does not come into contact with any delicate components that are located at a short distance above the original fill level. As will be seen in the case of the higher fill level, it is also possible for the ice dome to collide with the lid of the tank and consequently apply pressure on it. This can potentially lead to structural damage of the tank. Therefore, while the presence of the ice dome might not significantly alter temperature predictions, there are scenarios where the rise of the ice dome is of direct importance.

The predicted heights of the ice dome are listed in Table 6.4, alongside the experimentally observed heights for this case. Even though only a rough estimate was used



(a)

(b)

(c)

(d) continued

**Figure 6.3: Predicted solid fraction contours for the 50% filled tank of water after: (a) 1.4 hours, (b) 2.8 hours, (c) 4.2 hours, (d) 5.6 hours, (e) 7.0 hours (f) 8.3 hours (g) 9.7 hours (f) 11.1 hours.**

Figure 6.3 continued

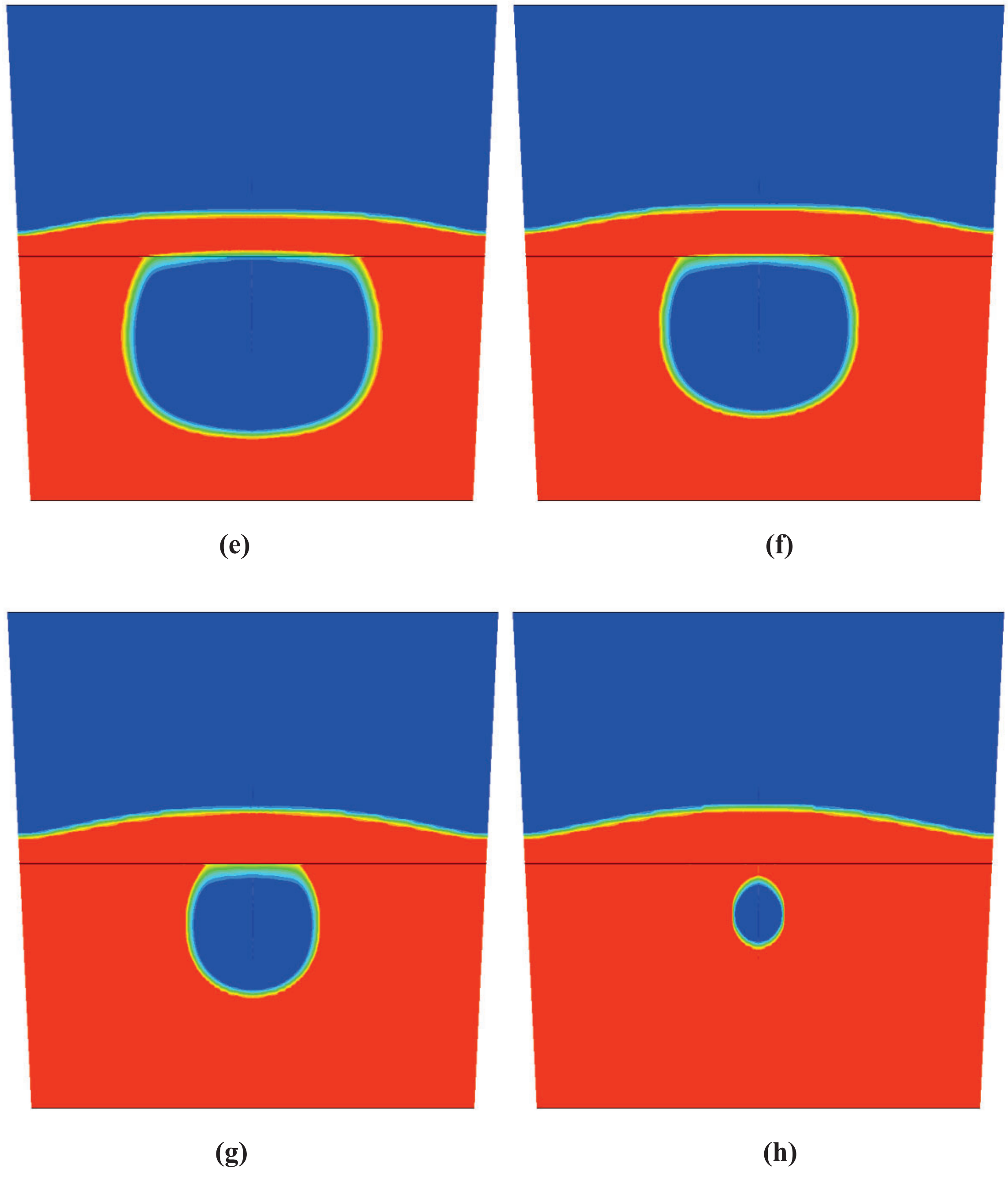


**Table 6.4: Comparison of ice dome heights between simulation and experiment for 50% fill level with water.**

| Experiment<br>/Simulation? | $\Delta y_{mid}$ (mm) | $\Delta y_{wall}$ (mm) | $\Delta y_{mid}/\Delta y_{wall}$ |
|---|---|---|---|
| **Exp** | 56 | 19 | 2.95 |
| **Sim** | 45.8 | 21.4 | 2.14 |

for the value of the porosity of the ice dome, the simulation predictions are reasonably close to the experimentally observed heights, with a difference of 10.2 mm at the center of the dome and 2.4 mm at the walls of the tank. It is possible that better estimates for the diffusion coefficient ratio and the porosity of the ice dome could result in better agreement with experimental data. The experimental data may also have uncertainties.

The wall-clock time to simulate 17 hours of freezing for this case was noted to be about 445 hours using 28 processors at OSC. This is in contrast to the 170 hours required for the simulation without the volume transport model (refer Section 4.7.1). An increase in run time is expected as an extra PDE for volume transport is being solved. When the simulation with volume transport enabled was run at Ford's HPC facility using 192 processors, the simulation required only 80 hours to complete. While it is understood that the wall-clock times observed from running this simulation at these two different facilities cannot be directly compared, it can still provide some insights into the parallel scalability of the simulation. Going from 28 processors to 192 processors is an increase in processor count by a factor of approximately 6.9. Ignoring the difference between the processors of the two computational facilities and assuming ideal parallel scaling, the run time for the simulation would be expected to decrease by the same factor, resulting in an expected run time close to 65 hours. The obtained run time of 80 hours is close to this approximation, which, despite the assumptions involved, indicates that the simulation has significant parallel scalability.

## 6.7 Simulation of Ford Production DEF Tank

So far, freezing simulations utilizing the two proposed models have been carried out for the (near) cylindrical tank shown in Fig. 4.1, as part of validation studies to correlate with experimental data. As a test of the versatility of these models, a freezing simulation was performed using a production DEF (Diesel Exhaust Fluid) tank used by the Ford Motor Company. A 3D CAD (Computer-Aided Design) model of this tank is shown in Fig. 6.4. As can be clearly seen, the geometry of this tank is significantly more complex than that of the cylindrical tank used thus far. Therefore, even though there is no experimental data available for the DEF tank, a simulation of this tank with both the reduced natural convection model and the volume transport model was set up for testing purposes as a worthwhile exercise.

### 6.7.1 Problem Setup

The simulation of this tank used AdBlue as the working liquid. The liquid was initially filled up to the height of the region marked in green in Fig. 6.4. The initial temperature within the tank was set to a uniform profile of 25°C. Convective boundary conditions were enforced at most walls of the tank. The ambient temperature was set to -40°C at all walls. The heat transfer coefficient at the top wall (in pink in Fig. 6.4), side wall (in green) and bottom walls for this tank were set to be the same as the calibrated values for the cylindrical tank, specifically ($h_{top}, h_{side}$ and $h_{bot}$ ) = (4, 35, 30 W/m$^2$-K). However, this tank possessed a urea-delivery module on the bottom wall of the tank. This module can be seen highlighted in green in Fig. 6.5(a). This module was set an adiabatic boundary condition instead. A baffle surrounded this module within the tank (also visible in Fig. 6.5(a)). This

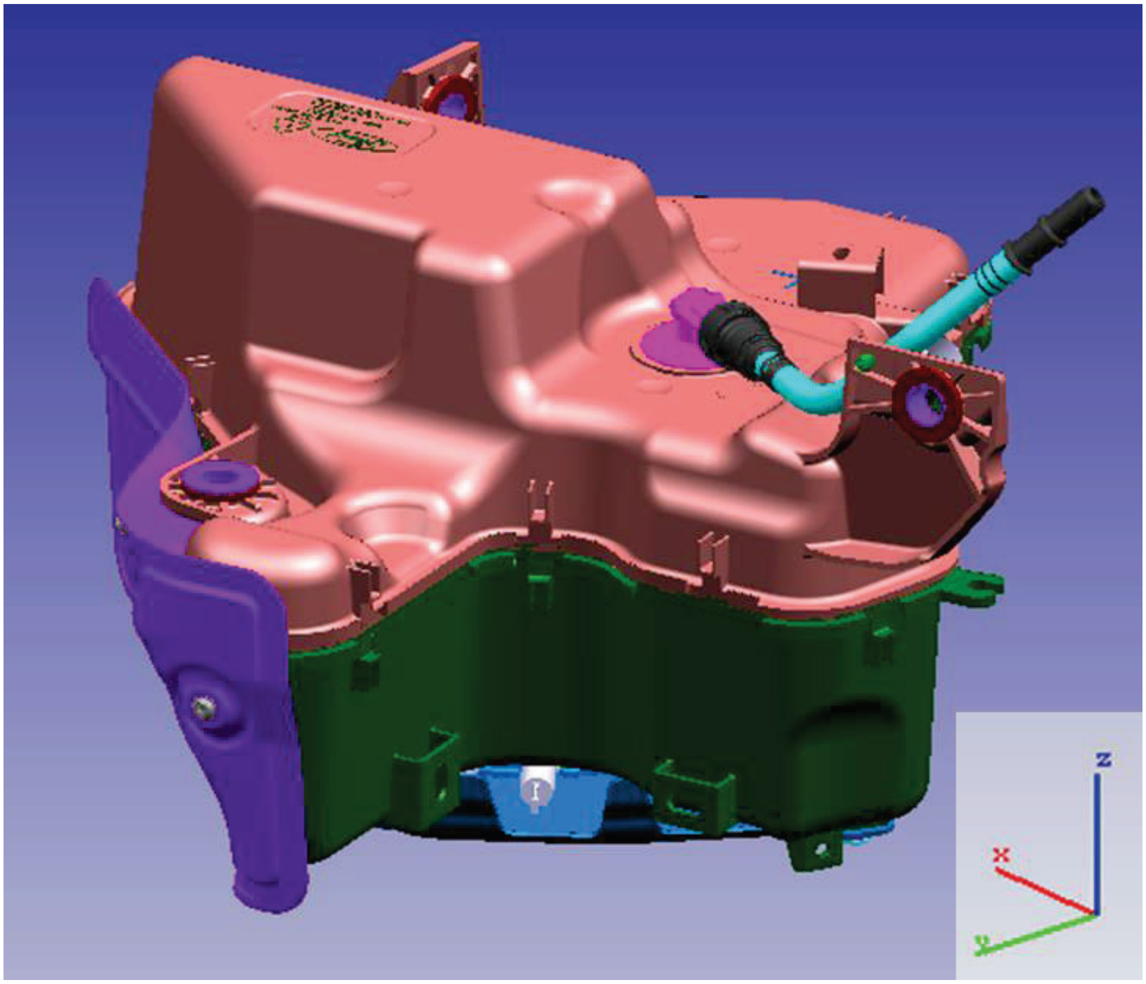
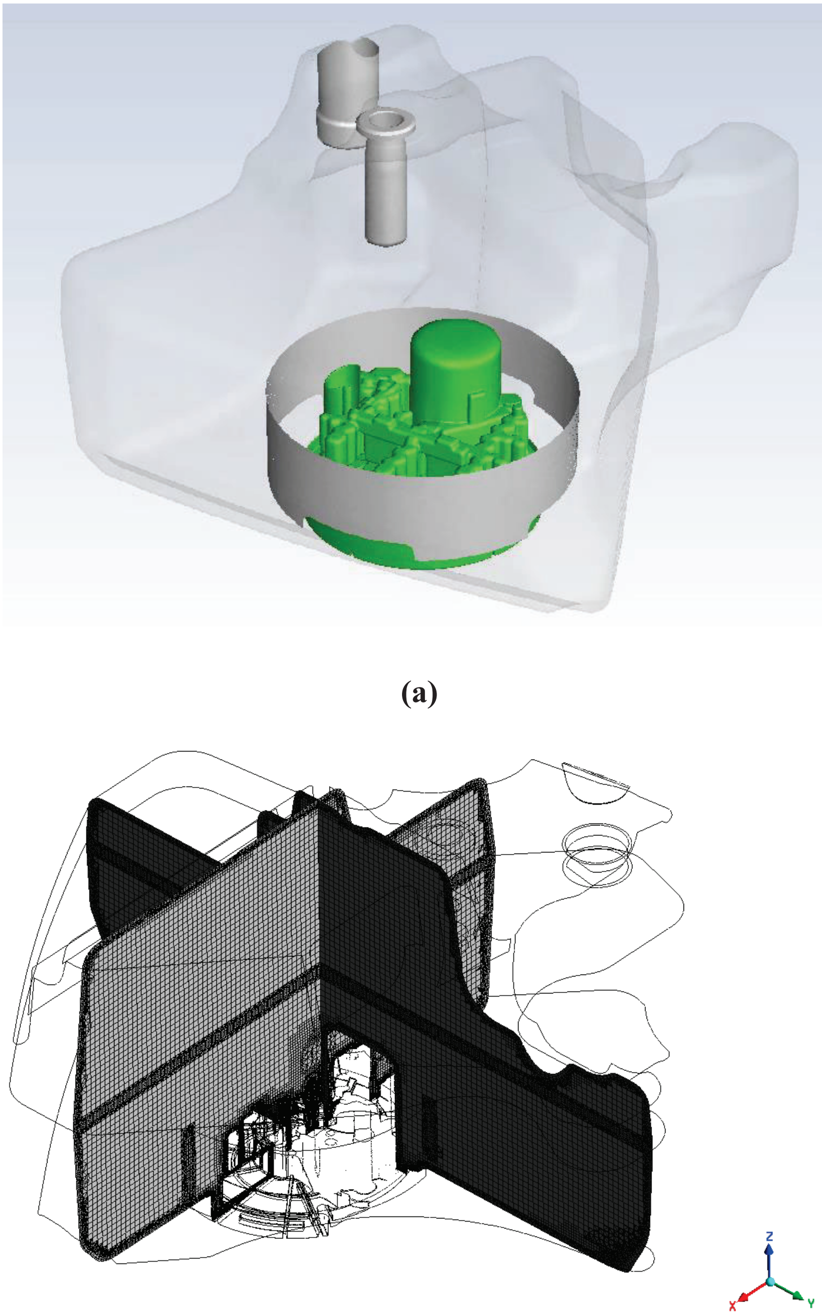


**Figure 6.4: 3D CAD model of a production DEF tank used by the Ford Motor Company.**

baffle was set to a conductive thin wall boundary condition. Additionally, the interior of this tank contains many protrusions and obstructions, unlike in the case of the cylindrical tank. This further complicates the geometry that the proposed models must be able to account for.

The tank was meshed using approximately six million cells of all topology (tets, quads, prisms, and arbitrary polyhedrons); the complex geometry of this tank and the particular mesh generator used by Ford resulted in cells with up to 24 faces. This polyhedral mesh prompted the extension of the volume transport equation solver's capabilities to handle meshes comprised of cells with an arbitrary number of faces, as discussed in Section 5.5. Figure 6.5(b) shows this mesh along two cutting planes of the tank. The rest of the setup for this simulation is the same as detailed previously in Section 6.3 for the cylindrical tank.



(a)

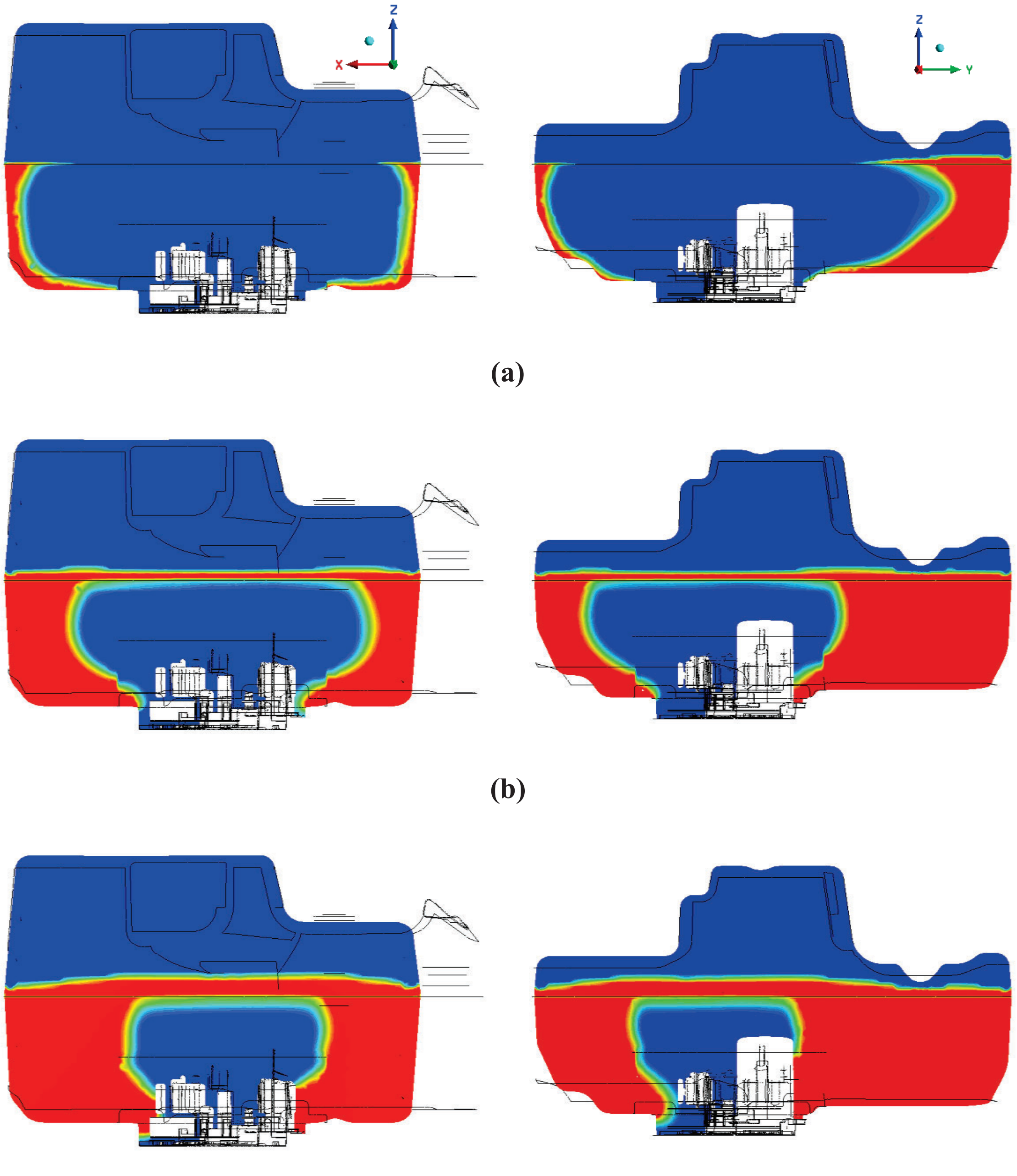
(b)

**Figure 6.5: (a) Simplified production DEF tank geometry used for simulation with urea delivery module and surrounding baffle within the tank (b) 6 million polyhedral cell mesh for the tank along cut planes in the XZ and YZ planes.**

### 6.7.2 Simulation Results

Figure 6.6 shows the progression of the liquid-solid front along the XZ (left) and YZ (right) cutting planes over the course of the freezing process. Due to the adiabatic boundary condition enforced at the urea delivery module on the bottom of the tank, the solidification front develops primarily from the side walls of the tank, with the location of the final liquid bubble located adjacent to this module. As mentioned in Chapter 1, this liquid bubble experiences extremely high pressures, close to 400psi (270 atm) [55]. If the current configuration for the freezing of the tank is a realistic one, it could have led to the potential damage of the module. The apparent discontinuity or kink observed adjacent to the urea delivery module in the contours of Figs. 6.6(b) and (c) is due to presence of the baffle mentioned earlier. Unlike the cylindrical tank, which due to its axisymmetric nature resulted in a symmetric liquid-solid front, the front for the current tank does not display any symmetry. Furthermore, the ice dome can be seen to respond to the local geometry as it expands. This is visible in top right of the contours along the YZ cutting plane. The rise of the ice dome is impeded by the dip in the roof of the tank, and the ice dome instead expands on either side of this dip.

These results highlight the usefulness of the proposed models. In the case of the cylindrical tank, it is possible to obtain an intuitive expectation of the progression of the solid-liquid front form our understanding of the governing laws. However, this is a much more difficult task for the kind of complex geometries showcased in this study. It is in such scenarios that the proposed models provide a potential avenue to obtain, at the very least, a reasonable approximation of the solid-liquid front.



(a)

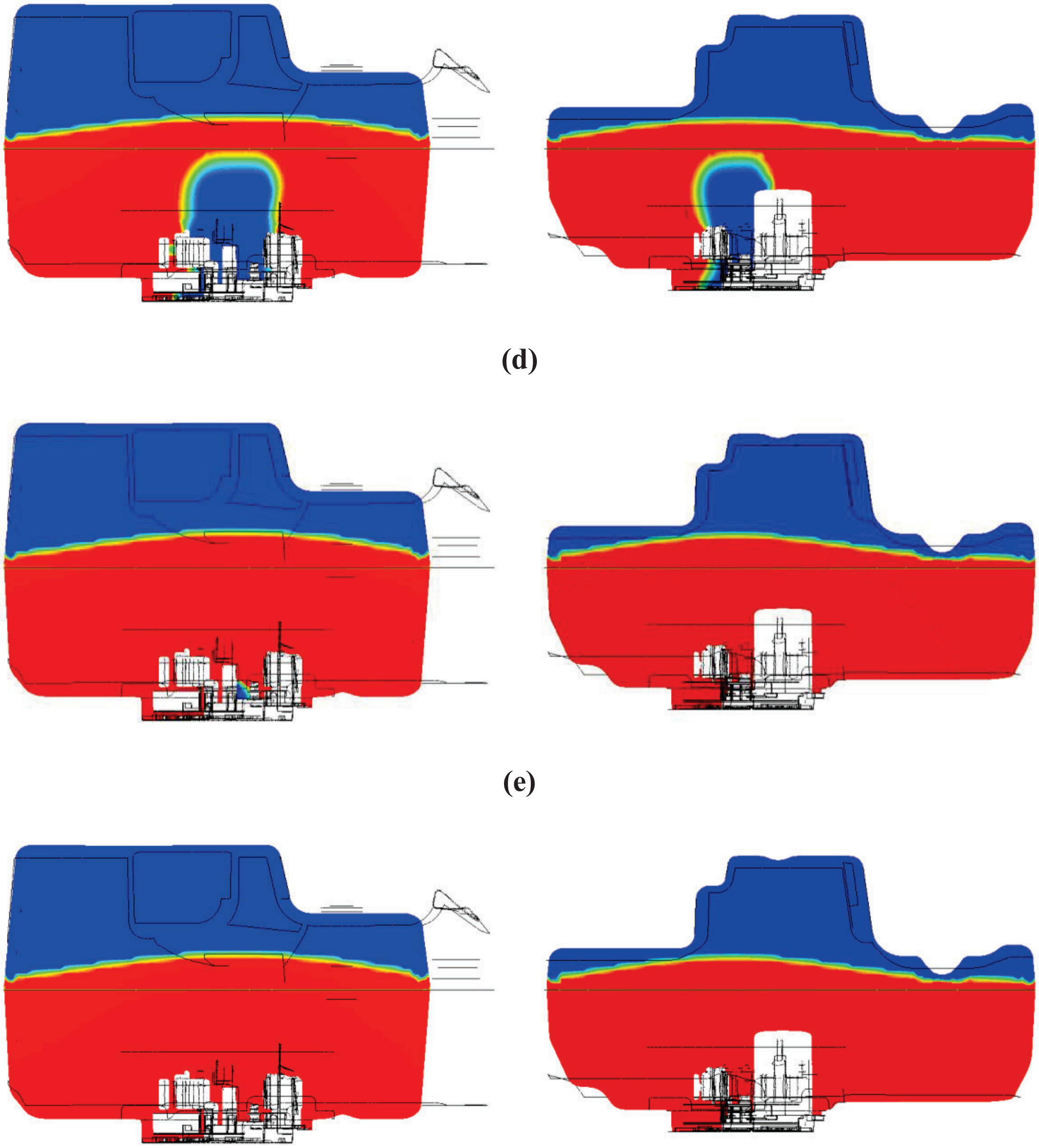
(b)

(c)

continued

**Figure 6.6: Contours of the solid-liquid front for the production DEF of AdBlue after: (a) 2 hours, (b) 4 hours, (c) 6 hours, (d) 8 hours, (e) 10 hours (f) 12 hours.**

Figure 6.6 continued



**(d)**

**(e)**

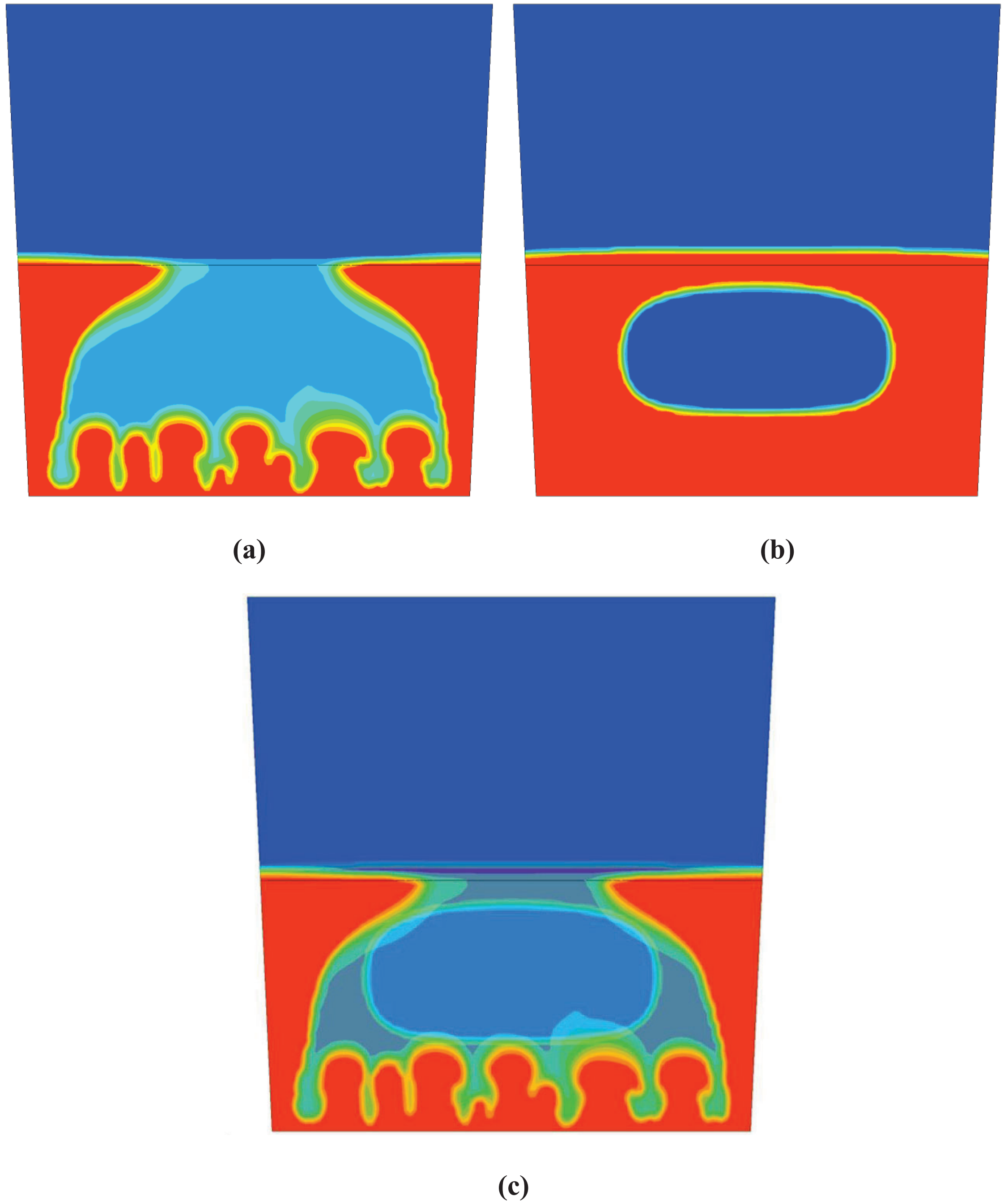
**(f)**

### 6.8 Comparison between high-fidelity models and proposed reduced models

In Chapter 2, a study was undertaken to assess the capabilities of high-fidelity models for freezing problems. A simple 2D freezing problem of a small 10 cm × 10 cm tank of water was set up for this purpose, and it was concluded that these high-fidelity models were unsuitable for even problems of such sizes (refer Section 2.4). At this stage, as both the reduced natural convection model and the volume transport model have been successfully implemented and validated for freezing problems, these reduced models can be used to solve the same freezing problem from Chapter 2. This would allow for a direct comparison between the predictions made by high-fidelity models vs. the reduced models.

The simulation is set up in the same manner as described in Section 2.4, with appropriate modifications as required by the two proposed models. Notably, the equations for flow are not solved and instead only the energy equation is solved. The effect of natural convection on heat transfer is accounted for by using the reduced natural convection model while the expansion of the ice dome is tracked using the volume transport model. The models are set up as was done for the cylindrical tank, and details can be found in Sections 4.6 and 6.3.

Figures 6.7(a) and (b) show the contours of the solid-liquid front after 14000 seconds of freezing as predicted by the high-fidelity models and the reduced models, respectively. For ease of comparison, Fig. 6.7(b) has its transparency increased and is then overlaid on Fig. 6.7(a). This modified figure is shown in Fig. 6.7(c). It is immediately obvious that the solid-liquid front seen in Fig. 6.7(b) lacks the fine details seen in Fig. 6.7(a). While the solidification front predicted by the high-fidelity models displays the



(a) (b)

(c)

**Figure 6.7: Contours of the solid-liquid front in a simple 2D tank after 4 hours of freezing using: (a) Fluent's in-built models, (b) the proposed models (c) overlap of contours in (a) and (b).**

formation of mushroom-like dendritic structures at the bottom of the tank, the solidification front predicted by the reduced models is smooth without any irregularities. Additionally, the liquid water bubble is completely enclosed in Fig. 6.7(b), while the prediction by the high-fidelity models shows that the surface of the liquid is yet to completely freeze over. This is explained by the observation that the high-fidelity models predict an extended mushy zone, represented by a light blue color in the center of the tank in Fig. 6.7(a). This is not captured in the predictions by the reduced model, where the mushy zone is narrow and well-defined resulting in a "tighter", enclosed liquid bubble. As can be seen in Fig. 6.7(c), this results in the fact that the reduced models predict a solidification front that could be considered an "average" of the front predicted by the in-built models, which, at a superficial level appears to be smaller than expected, but in fact makes sense based on the nature of the mushy zone in the two cases. From the definitions of solid fraction and volume fraction, the mass of ice predicted by each model at the shown time can be calculated. Consequently, the mass of ice as predicted by the high-fidelity model is 2.33 kg, while the reduced model predicts a mass of 3.03 kg. Lastly, while the simulation using the high-fidelity models required 2 months of CPU time to simulate 14000 seconds of freezing, the reduced models required only 8 hours. Therefore, these results highlight that in scenarios where the fine details of the solidification front are of lower priority compared to computational speed and efficiency, the proposed models are a suitable alternative to high-fidelity models.

# CHAPTER 7. SUMMARY AND FUTURE WORK

## 7.1 Summary

The overall objective of this study was to simulate the freezing of water and/or Adblue® in “large” partially-filled tanks, typically used to store liquid Adblue® onboard diesel vehicles. Numerical modelling of the solidification/melting process within large partially-filled tanks is a challenging task. In solidification/melting problems, the process of solving the standard equations for flow and energy is complicated by the requirement to account for the multiple phases (solid, liquid and gas) and the transition between the solid and liquid phases. Tracking the three phases is typically accomplished by introducing and solving for two new variables, solid fraction and volume fraction, each representing the relative quantity of the solid and liquid phases, respectively. While commercial CFD codes utilizing pre-existing models (enthalpy-porosity, VOF) have been used to tackle solidification/melting problems, most studies have been limited to simple two-dimensional geometries spanning a few square centimeters in length  and a few hours in time. The tanks of interest in this study typically have a capacity of several liters, and complete solidification has been reported to take over 24 hours. To kick-start the project, a study was first undertaken where the high-fidelity models offered by the commercial CFD solver ANSYS Fluent$^{TM}$ were utilized to study the freezing process in a partially-filled tank. In this study, a simple 2D tank was used, which was about a quarter of the size of the tanks used in subsequent studies. The tank was assumed to be partially-filled with water and was meshed using 10000 quadrilateral cells. The freezing of this multi-phase system was modeled using the solidification model provided by Fluent alongside the Volume-of-Fluid (VOF) method to account for the expansion of ice. Results from the 2D simulations revealed that though Fluent was able to provide detailed predictions of the solidification front, both within the water volume and at the air-water interface, the simulation required an extremely long run time, where simulating 14000 seconds of freezing required approximately 2 months of CPU time. This was primarily due to the extremely small time-step sizes (0.5 milliseconds) that were needed to capture the natural convection effects. Parallelizing the simulation also did not lead to any improvements in run time due to the small count used for the 2D problem. As the eventual goal was to run freezing simulations of significantly larger tanks with freezing time scales close to 24 hours, the in-built models of Fluent were deemed to be unsuitable for such problems. To illustrate, a simple extrapolation of the run times observed in the 2D problem to the planned 3D cases lead to estimates of run times that were over a year, even under the assumption of perfect parallel scaling with 64 compute nodes. Therefore, this led to the development an efficient 3D computational model that could be used to model the solidification process in storage tanks used in Selective Catalytic Reduction (SCR) systems, which is the overall objective of this dissertation. In pursuit of this objective, a reduced model for the heat transfer due to natural convection was first developed and validated. Following this, a diffusion-based volume transport model was developed to account for the expansion of ice during the freezing process. The details of these studies are summarized in this chapter.

During the development of the reduced natural convection model for heat transfer, the expansion of ice and, consequently, the movement of the air-(solid/liquid) interface was neglected. The primary focus was to account for the heat transfer due to natural convection by introducing an artificial thermal conductivity such that the convective heat flux could be represented by an equivalent conductive heat flux. This artificial thermal conductivity was derived from physical laws governing natural convection. Importantly, this approach bypassed solving for flow and reduced the energy equation to a simple heat conduction equation while still accounting for the effects of natural convection. This was expected to provide significant benefits in computational efficiency. Lastly, the latent heat released during solidification was modeled using a volumetric source term.

To validate this reduced model, experimental studies were carried out by engineers at the Ford Moto Company, who are the sponsors of the project. In these experimental studies, the freezing process of a cylindrical tank was observed in various configurations. Three fill levels (25%, 50% and 80%) of the tank were considered and separate experiments were conducted with AdBlue and water as the working liquid. Temperature data at various thermocouple locations within the tank were collected over the course of the experiment for validation purposes. Prior to validation, an unknown adjustable constant embedded within the reduced model was calibrated such that the temperature vs. time predictions at five thermocouple locations matched experimental measurements. A separate calibration data set, distinct from the validation data sets, was used for this purpose. Validation studies revealed that predictions using reduced model generally showed good agreement with the experimental data. The RMS errors between the predicted and measured temperatures were calculated at the thermocouple locations, and were all found to be under 10°C. The errors were the smallest for the lowest fill level and largest for the highest fill level. The errors for the AdBlue cases were slightly higher than their corresponding water cases. The freezing sequences of the thermocouples were correctly predicted for most cases, with only the two 80% fill cases showing a reversal of order for two thermocouples. From a computation efficiency perspective, it was observed that the simulations with the reduced model required a maximum of 150 hours (for the 80% fill cases), which was a significant improvement over the 1-year estimates when utilizing Fluent's in-built models.

Though the development of the reduced model had been motivated with freezing problems as the primary objective, the reduced model could, in principle, be applied to any general natural convection problem. To test this contention, a problem studying the heat transfer process in a differentially heated cavity while using the reduced natural convection model was set up at three different Rayleigh numbers ($10^4$, $10^5$ and $10^6$). As a point of comparison, the problem was also solved using high-fidelity models that solved the full set of conservation equations. The temperatures predicted by high-fidelity models and the reduced model at five different locations were compared, with slight differences at certain locations. The average Nusselt numbers at the wall were also calculated and compared, alongside empirical correlations from literature, showing good agreement. This highlighted that the reduced model was good at providing an overall estimate of the heat transfer occurring within a system while providing significantly higher computational efficiency at the cost of loss of finer details such as local temperatures.

Next, a model was developed to account for the expansion of ice that occurs during the freezing process and the resulting rise of an ice dome at the gas-(solid/liquid) interface. Conventional methods such as the VOF method could not be used here as they use fluid flow to propagate the interface, while the major feature of the reduced model was that it bypassed solving the equation for flow and reduced the problem to a pure diffusion problem. Therefore, a new diffusion-based form of the VOF method was developed that would be able to track the gas-(solid/liquid) interface and could also be used alongside the reduced natural convection model. In this model, a new equation for excess volume fraction, derived from mass conservation, was introduced to track the movement of the gas-(solid/liquid) interface. The model used different diffusion coefficients for the liquid and solid phases in order to capture the near impermeable nature of the solid phase and appropriate flux limiters were implemented to mimic multi-phase flow with sharp interfaces. This equation was solved using a sub-time-stepping procedure, such that any excess volume created by ice expansion was redistributed completely with each time-step of the energy equation.

As this model introduced a new equation that was not part of the standard set of equations solved by Fluent, a significant step in this work involved implementing the volume transport model, i.e., the development of a custom parallel, unstructured conjugate gradient squared (CGS) solver with Jacobi pre-conditioning within the framework of Fluent's UDFs, that was instead used to solve the volume transport equation.

The data from the previously mentioned experimental studies were used once again for validation purposes. Only rudimentary ice dome height data was available. Therefore, validation studies focused primarily on temperature predictions with the effect of the ice dome taken into account. However, prior to the validation studies, rough estimates were made for the porosity of the ice dome and the volume “diffusion coefficients” of the solid and liquid phases using the limited ice dome height data available. Consequently, it was found from the validation studies that the inclusion of the ice dome in the simulation slightly improved the agreement with experimental data at the thermocouples located closer to gas-(solid/liquid) interface while having little to no effect on the thermocouples further away, such as near the bottom wall of the tank. The results from the simulations clearly showed the rise of the ice dome at the free surface over the course of the freezing process. Lastly, it was found that the required computational time doubled when using both the reduced model and the volume transport model as compared to when using the reduced model alone.

To further test the capabilities of the proposed models, a freezing simulation was also performed using a production DEF (Diesel Exhaust Fluid) tank used by the Ford Motor Company. This tank possessed a significantly more complex geometry compared to the axisymmetric cylindrical tank used in studies up to this point. Therefore, meshing this tank required approximately six million cells of all topological shapes (tets, quads, prisms, and arbitrary polyhedrons); the complex geometry of this tank and the particular mesh generator used by Ford resulted in cells with up to 24 faces. The tank was assumed to be partially filled with AdBlue for the purposes of testing. As no experimental data was available for this case, validation studies could not be performed. Nonetheless, this case was used to successfully demonstrate the ability of the reduced models to handle scenarios involving tanks with complex geometries.

As a final comparison, the simple 2D problem previously used to evaluate the capabilities of Fluent's in-built models was solved again using the proposed models instead. The predicted solidification fronts after 14000 seconds of freezing were then compared. These results reinforced the contention that the reduced model was able to provide a good average estimate of the heat transfer occurring within a system. Furthermore, the simulation using the proposed model required only 8 hours of run time to simulate 14000 seconds of freezing, as opposed to the 2 months required by Fluent's models. This highlighted the gains in computational efficiency offered by the proposed models.

## 7.2 Future Work

The work presented in the course of this thesis provides many further avenues of development and experimentation in the interest of the improvement of the proposed models. To illustrate, the following are some recommendations for future work.

- All studies and simulations performed in this work involved freezing of a partially filled tank. As damage to the tank and its components primarily occurs during freezing, thawing simulations are not as critical. However, it would nonetheless be useful to study the thawing process as well. This is because mitigation strategies, such as turning on heaters at prescribed locations and durations, will involve simultaneous freezing and thawing.

- In relation to the previous point, while the reduced natural convection model is equally capable at handling freezing and thawing problems, the volume transport model has been designed exclusively for freezing problems. The extension of this model to thawing scenarios is a challenging prospect, due to the importance of gravity and the resulting settling effect during thawing. In particular, the calculation of excess volume fraction, the flux limiters, and the fluxing scheme itself would require modifications to account for thawing.
- Calibration of the diffusion coefficient and the porosity of the ice dome in the volume transport model were hampered by the limited experimental ice dome height data available. Further experiments with detailed measurements would help further refine this model.
- As the reduced models can predict the location of the final liquid bubble as it shrinks during solidification, they inherently also predict the location of maximum pressure, as these two are one and the same [55]. However, it would be useful to be able to quantify the pressure at this location. In the current implementation of the reduced models, this is impossible to do due to the lack of flow calculations.

# REFERENCES

[1] Koebel, M., Elsener, M., and Kleemann, M., (2000), "Urea-SCR: a promising technique to reduce NOx emissions from automotive diesel engines," *Catalysis today*, Vol. 59(3-4), pp. 335-345.

[2] BASF, *AdBlue® Brochure*, https://www.basf.com/cn/documents/en/products-and-industries/adblue/adblue-brochure-en.pdf

[3] BASF, AdBlue®, Retrieved January 30, 2023, from https://chemicals.basf.com/global/en/Monomers/adblue.html

[4] Birkhold, F., Meingast, U., Wassermann, P., & Deutschmann, O. (2006). Analysis of the injection of urea-water-solution for automotive SCR DeNOx-systems: modeling of two-phase flow and spray/wall-interaction (No. 2006-01-0643). SAE Technical Paper.

[5] Abu-Ramadan, E., Saha, K., & Li, X. (2011). Modeling of the injection and decomposition processes of urea-water-solution spray in automotive SCR systems (No. 2011-01-1317). SAE Technical Paper.

[6] aus der Wiesche, S., (2007), "Numerical heat transfer and thermal engineering of AdBlue (SCR) tanks for combustion engine emission reduction," *Applied Thermal Engineering*, Vol. 27(11-12), pp. 1790-1798.

[7] L.L.C. Chrysler, Thermal analysis of urea tank solution warm-up for selective catalytic reduction (SCR), SAE-paper No. 2010-01-0971.

[8] Choi, B. C., Kim, Y. K., Jhung, W. N., Lee, C. H., & Hwang, C. Y. (2013). Experimental investigation on melting characteristics of frozen urea–water-solutions for a diesel SCR de-NOx-system. Applied Thermal Engineering, 50(1), 1235-1245.

[9] Choi, B., & Woo, S. M. (2015). Numerical analysis of the optimum heating pipe to melt frozen urea-water-solution of a diesel urea-SCR system. Applied Thermal Engineering, 89, 860-870.

[10] de Beeck, J. O., Slusser, K., & Booth, N. (2014). DEF Storage and Delivery System for Operation in Extreme Winter Conditions (No. 2014-01-1530). SAE Technical Paper.

[11] Ramesh, V. (2019). Modeling Freeze/Thaw Behavior in Tanks for Selective Catalytic Reduction (SCR) Applications [Master's thesis, Ohio State University]. OhioLINK Electronic Theses and Dissertations Center. http://rave.ohiolink.edu/etdc/view?acc_num=osu156328011301541

[12] Minaie, B., Stelson, K. A., and Voller, V. R., 1991, "Analysis of flow patterns and solidification phenomena in the die casting process," *Journal of engineering materials and technology*, Vol. 113(3), pp. 296-302.

[13] Im, I. T., Kim, W. S., and Lee, K. S., 2001, "A unified analysis of filling and solidification in casting with natural convection," *International Journal of Heat and Mass Transfer*, Vol. 44(8), pp. 1507-1515.

[14] Tay, N. H. S., Bruno, F., and Belusko, M., (2012), "Experimental validation of a CFD model for tubes in a phase change thermal energy storage system," *International Journal of Heat and Mass Transfer*, Vol. 55(4), pp. 574-585.

[15] Trp, A., (2005), "An experimental and numerical investigation of heat transfer during technical grade paraffin melting and solidification in a shell-and-tube latent thermal energy storage unit," *Solar Energy*, Vol. 79(6), pp. 648-660.

[16] Al-Abidi, A. A., Mat, S. B., Sopian, K., Sulaiman, M. Y., and Mohammed, A. T., (2013), "CFD applications for latent heat thermal energy storage: a review," *Renewable and Sustainable Energy Reviews*, Vol. 20, pp. 353-363.

[17] Stefan, J., (1891), "Über die Theorie der Eisbildung, insbesondere über die Eisbildung im Polarmeere," Annalen der Physik, Vol. 278(2), pp. 269-286.

[18] Vuik, C. (1993). Some historical notes about the Stefan problem.

[19] K.H. Hoffman, Fachbereich Mathematik, Vol. I-III, Berlin Freie Universitat, 1977.

[20] Crank, J. (1987). *Free and moving boundary problems*. Oxford University Press.

[21] Javierre, E., Vuik, C., Vermolen, F. J., & Van der Zwaag, S. (2006). A comparison of numerical models for one-dimensional Stefan problems. *Journal of Computational and Applied Mathematics*, ***192***(2), 445-459.

[22] Jana, S., Ray, S., and Durst, F., 2007, "A numerical method to compute solidification and melting processes." *Applied Mathematical Modelling*, Vol. 31(1), pp. 93-119.

[23] Zhang, H., Prasad, V., and Moallemi, M. K., 1996, "Numerical algorithm using multizone adaptive grid generation for multiphase transport processes with moving and free boundaries," *Numerical Heat Transfer*, Vol. 29(4), pp. 399-421.

[24] Voller, V. R., Cross, M., and Walton, P. G., 1979, *Assessment of weak solution numerical techniques for solving Stefan problems,* Swansea, UK: Pineridge Press, pp. 172-181.

[25] Łapka, P., & Furmański, P. (2008). Numerical modelling of solidification processes of semitransparent materials using the enthalpy and the finite volume methods. Heat and Mass Transfer, 44(8), 937.

[26] Voller, V., and Cross, M., 1981, "Accurate solutions of moving boundary problems using the enthalpy method," *International Journal of Heat and Mass Transfer*, Vol. 24(3), pp. 545-556.

[27] Voller, V. R., and Prakash, C., 1987, "A fixed grid numerical modelling methodology for convection-diffusion mushy region phase-change problems," *International Journal of Heat and Mass Transfer*, Vol. 30(8), pp. 1709-1719.

[28] Hirt, C. W., & Nichols, B. D. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. Journal of computational physics, 39(1), 201-225.

[29] Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of computational physics, 79(1), 12-49.

[30] Caginalp, G. (1984). Applications of field theory to statistical mechanics. number, 216, 216-226.

[31] Langer, J. S. (1986). Directions in condensed matter physics. Models of Pattern Formation in First-order Phase Transition, 165.

[32] Wheeler, A. A., Boettinger, W. J., & McFadden, G. B. (1992). Phase-field model for isothermal phase transitions in binary alloys. Physical Review A, 45(10), 7424.

[33] Tan, L., & Zabaras, N. (2007). A level set simulation of dendritic solidification of multi-component alloys. Journal of Computational Physics, 221(1), 9-40.

[34] Boettinger, W. J., Warren, J. A., Beckermann, C., and Karma, A., 2002, "Phase-field simulation of solidification," Annual Review of Materials Research, Vol. 32(1), pp. 163-194.

[35] Shyy, W., Udaykumar, H. S., Rao, M. M., and Smith, R. W., (2007), *Computational Fluid Dynamics with Moving Boundaries*, Dover Publications.

[36] Yeoh, G. H., Behnia, M., De Vahl Davis, G., and Leonardi, E., 1990, "A numerical study of three-dimensional natural convection during freezing of water," *International Journal for Numerical Methods in Engineering*, Vol. 30, pp. 899-914.

[37] Mazumder, S., 2016, *Numerical Methods of Partial Differential Equations: Finite Difference and Finite Volume Methods*, Academic Press, New York.

[38] Bourdillon, A. C., Verdin, P. G., and Thompson, C. P., 2015, "Numerical simulations of water freezing processes in cavities and cylindrical enclosures," *Applied thermal engineering*, Vol. 75, pp. 839-855.

[39] Sharma, R. K., Ganesan, P., Sahu, J. N., Metselaar, H. S. C., and Mahlia, T. M. I., 2014, "Numerical study for enhancement of solidification of phase change materials using trapezoidal cavity," *Powder technology*, Vol. 268, pp. 38-47.

[40] Mazumder, S. (2007). Modeling full-scale monolithic catalytic converters: challenges and possible solutions. Journal of Heat Transfer, 129(4), 526-535.

[41] Mazumder, S., & Sengupta, D. (2002). Sub-grid scale modeling of heterogeneous chemical reactions and transport in full-scale catalytic converters. Combustion and flame, 131(1-2), 85-97.

[42] Noh, W.F., Woodward, P. (1976). SLIC (Simple Line Interface Calculation). In: van de Vooren, A.I., Zandbergen, P.J. (eds) Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28 – July 2, 1976 Twente University, Enschede. Lecture Notes in Physics, vol 59. Springer, Berlin, Heidelberg.

[43] DeBar, R. (1974). Fundamentals of the KRAKEN code. Technical Report UCIR-760.

[44] Youngs, D. L. (1982). Time-dependent multi-material flow with large fluid distortion. Numerical methods for fluid dynamics.

[45] Whitaker, S., 1977, Fundamental Principles of Heat Transfer, Pergamon Press, New York.

[46] Kell, G. S., 1975, "Density, Thermal Expansivity, and Compressibility of Liquid Water From 0. deg. to 150. deg. Correlations and Tables for Atmospheric Pressure and Saturation Reviewed and Expressed on 1968 Temperature Scale," J. Chem. Eng. Data, 20(1), pp. 97–105.

[47] Hobbs, P. V., 2010, *Ice Physics*, Oxford University Press, New York

[48] Alger, Mark. "L." 2017. Polymer Science Dictionary, by Mark Alger, Springer Netherlands.

[49] Ramesh, V., Mazumder, S., Matharu, G., Vaishnav, D., Ali, S., Lawrence, D., and Desai, J., 2018, "Calibration of External Heat Transfer Coefficients During Cooling of a Partially-Filled Water Tank Using Measured Temperature-Time Data," Proceedings of the IMECE2018, Pittsburgh, PA, Nov. 11–14, Paper Number IMECE2018-86716.

[50] BASF, 2016, *AdBlue® Technical Leaflet,* M 6621 e.

[51] Bayram, Ü., Öztürk, E., Aksöz, S., & Maraşlı, N. (2013). Measurements of Thermal Conductivity Variations with Temperature for the Organic Analog of the Nonmetal–Nonmetal System: Urea–4-Bromo-2-Nitroaniline. *Metallurgical and Materials Transactions A*, 44(9), 4051-4058.

[52] Bergman, T.L., Bergman, T.L., Incropera, F.P., Dewitt, D.P. and Lavine, A.S., 2011. *Fundamentals of heat and mass transfer*. John Wiley & Sons.

[53] Mazumder, S., 2007. On the use of the fully compressible Navier-Stokes equations for the steady-state solution of natural convection problems in closed cavities.

[54] Leong, W.H., Hollands, K.G.T. and Brunger, A.P., 1999. Experimental Nusselt numbers for a cubical-cavity benchmark problem in natural convection. *International Journal of Heat and Mass Transfer, 42*(11), pp.1979-1989.

[55] Akyurt, M., Zaki, G., & Habeebullah, B. (2002). Freezing phenomena in ice–water systems. *Energy conversion and management*, *43*(14), 1773-1789.

[56] Rider, W. J., & Kothe, D. B. (1998). Reconstructing volume tracking. *Journal of computational physics*, *141*(2), 112-152.

[57] McKee, S., Tomé, M. F., Ferreira, V. G., Cuminato, J. A., Castelo, A., Sousa, F. S., & Mangiavacchi, N. (2008). The MAC method. *Computers & Fluids*, *37*(8), 907-930.

[58] Tryggvason, G., Scardovelli, R., & Zaleski, S. (2011). *Direct numerical simulations of gas–liquid multiphase flows*. Cambridge university press.

[59] Mirjalili, S., Jain, S. S., & Dodd, M. (2017). Interface-capturing methods for two-phase flows: An overview and recent developments. *Center for Turbulence Research Annual Research Briefs*, *2017*(117-135), 13.

[60] Tezduyar, T. E. (2006). Interface-tracking and interface-capturing techniques for finite element computation of moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering*, *195*(23-24), 2983-3000.

[61] Timco, G.W. and Frederking, R.M.W., 1996. A review of sea ice density. *Cold regions science and technology*, *24*(1), pp.1-6

[62] Sigworth, G.K. and Wang, C., 1993. Mechanisms of porosity formation during solidification: A theoretical analysis. *Metallurgical Transactions B*, *24*, pp.349-364.

[63] Voller, V. R., Brent, A. D., & Prakash, C. (1989). The modelling of heat, mass and solute transport in solidification systems. *International Journal of Heat and Mass Transfer*, *32*(9), 1719-1731.

[64] Fluent, A. (2021). 2021 R2 User's Guide. *Ansys Inc.*

[65] Fluent, A. (2021). 2021 R2 Customization Manual. *Ansys Inc*.

# APPENDIX A.  UDF FOR CALCULATING ENHANCED THERMAL CONDUCTIVITY

```
#include "udf.h"
#include "stdio.h"
#include "ctype.h"
#include "stdarg.h"

#define MW 28.966
#define RGAS (UNIVERSAL_GAS_CONSTANT/MW)

double Lc_liquid;
double Lc_gas;

double k_liquid;
double k_solid;
double calib_const_liquid;
double calib_const_gas;
double k_gas;
double T_liquidus;
double T_solidus;
double rho_liquid;
double rho_solid;
double rho_ref_liquid;
double rho_ref_gas;

int UDM_max;

DEFINE_EXECUTE_AT_END(UDF_0_Execute_At_End_Update_Thermal_Cond)
{
        #if !RP_HOST /* SERIAL OR NODES ONLY */

        double thermalc_final;

        double k_mol_liquid,k_mol_gas;

        double l_f;
        double avg_llf;
        double avg_glf;

        double drho_dT;
        double cell_temp_c;
        double cell_density_ls;

        double x_comp;
        double y_comp;
        double z_comp;
        double mag_grad;

        double k_g,k_ls;

        double T;

        cell_t c;
        Thread *tc;
```

```
	Domain *d;
	d = Get_Domain(1);

	EXCHANGE_UDMI(d,0,UDM_max);
	thread_loop_c(tc,d)
	{
		begin_c_loop_int(c,tc)
		{
			T = C_T(c,tc);

			drho_dT = -101325.0/(RGAS*pow(T,2.0));

			k_g = k_gas;

			if (C_UDMI(c,tc,18)<1E-10)
			{
				x_comp = drho_dT*C_UDMI(c,tc,7);
				y_comp = drho_dT*C_UDMI(c,tc,8);
				z_comp = drho_dT*C_UDMI(c,tc,9);

				y_comp = (C_UDMI(c,tc,8) < 0)*(y_comp > 0)*y_comp; /* If y comp. of grad T >=0, turn off
contribution *//*If y comp. of grad rho <0, turn off contribution */
				/*z_comp = (C_UDMI(c,tc,9) < 0)*(z_comp > 0)*z_comp; /* If y comp. of grad T >=0, turn off
contribution *//*If y comp. of grad rho <0, turn off contribution */
				mag_grad = sqrt(pow(x_comp,2.0) + pow(y_comp,2.0) + pow(z_comp,2.0));
				C_UDMI(c,tc,5) = mag_grad;

				l_f = C_UDMI(c,tc,12);
				avg_glf = C_UDMI(c,tc,3);
				/* avg_glf = 0.001; */
```

```
                k_mol_gas = k_gas;
                k_g = (1 + calib_const_gas*avg_glf*pow(Lc_gas,-
0.175)*pow((mag_grad*l_f/rho_ref_gas),0.275))*k_mol_gas;
            }

            /*k_g = k_mol_gas;*/

            k_mol_liquid = k_solid*C_UDMI(c,tc,1) + k_liquid*(1-C_UDMI(c,tc,1));
            k_ls = k_mol_liquid; /*If solid*/

            if(C_UDMI(c,tc,1)<1.0 && C_UDMI(c,tc,18)>C_UDMI(c,tc,24))/*If not solid and atleast some water*/
            {
                if(C_UDMI(c,tc,1) == 0.0)/*If pure liquid*/
                {
                    cell_temp_c = T - 273.15;

                    drho_dT = ((1 + 16.879850*1E-3*cell_temp_c)*(16.945176 - 2*7.9870401*1E-3*cell_temp_c -
3*46.170461*1E-6*pow(cell_temp_c,2.0) + 4*105.56302*1E-9*pow(cell_temp_c,3.0) - 5*280.54253*1E-12*pow(cell_temp_c,4.0))
                                - 16.879850*1E-3*(999.83952 + 16.945176*(cell_temp_c) -
7.9870401*1E-3*pow(cell_temp_c,2.0) - 46.170461*1E-6*pow(cell_temp_c,3.0) + 105.56302*1E-9*pow(cell_temp_c,4.0) -
280.54253*1E-12*pow(cell_temp_c,5.0)))
                                /pow((1 + 16.879850*1E-3*cell_temp_c),2.0);
                }
                else
                {
                    cell_density_ls = C_UDMI(c,tc,20);
                    drho_dT = ((rho_solid-rho_liquid)/(T_solidus-
T_liquidus))*cell_density_ls*cell_density_ls/(rho_solid*rho_liquid);          /*For Mushy Zone*/
                }
                x_comp = drho_dT*C_UDMI(c,tc,7);
                y_comp = drho_dT*C_UDMI(c,tc,8);
                z_comp = drho_dT*C_UDMI(c,tc,9);

                y_comp = (C_UDMI(c,tc,8) < 0)*(y_comp > 0)*y_comp; /* If y comp. of grad T >=0, turn off
contribution *//*If y comp. of grad rho <0, turn off contribution */
                /*z_comp = (C_UDMI(c,tc,9) < 0)*(z_comp > 0)*z_comp; /* If y comp. of grad T >=0, turn off
contribution *//*If y comp. of grad rho <0, turn off contribution */
                mag_grad = sqrt(pow(x_comp,2.0) + pow(y_comp,2.0) + pow(z_comp,2.0));
                C_UDMI(c,tc,6) = mag_grad;

                l_f = C_UDMI(c,tc,12);
                avg_llf = C_UDMI(c,tc,4);
                /* avg_llf = 0.001; */

                k_ls = (1 + calib_const_liquid*avg_llf*pow(Lc_liquid,-
0.175)*pow((mag_grad*l_f/rho_ref_liquid),0.275))*k_mol_liquid;

                /*k_ls = k_mol_liquid;*/
            }

            thermalc_final = k_g*(1-C_UDMI(c,tc,18)*C_UDMI(c,tc,18)) + k_ls*C_UDMI(c,tc,18)*C_UDMI(c,tc,18);

            /* thermalc_final = MIN(thermalc_final,100000); */
            thermalc_final = MAX(thermalc_final,0.9*k_gas);

            C_UDMI(c,tc,23) = thermalc_final;
        }
        end_c_loop_int(c,tc)
    }
    EXCHANGE_UDMI(d,23,23);
    #endif /* !RP_HOST */

}
```

# APPENDIX B.  UDF FOR THE VOLUME TRANSPORT EQUATION SOLVER

```
# include "udf.h"
# include "sg.h"
# include "para.h"
# include "math.h"
# include "ModelProperties.h"

int vf_N;
double total_vol;
int f_offset;
int n_offset;
int UDM_max;

DEFINE_EXECUTE_AT_END(UDF_1_Volume_Transport)
{

        #if !RP_HOST /* SERIAL OR NODES ONLY */

        int curr_ts = N_TIME;
        int n,nn,nn_nb;
        int cell_count;
        int flag,BCflag;
        int count;
        int nftotal;

        double alpmax;
        double alpha;
        double C1;
double dvf;
double A_by_es;
double ds;
double vfedge;
double max_vf;
double max_evf;
double skewterm;
double lfdottang;
double d0,d1;

double flux_factor;

double dvfdt2,dvfdt;

double vftol = 1E-8;
double current_time;

double sub_dt,sub_dt_max,sub_dt_min;
double sub_dt_factor;

double cc_centroid[ND_ND];
double f_centroid[ND_ND];
double node0Loc[ND_ND];
double node1Loc[ND_ND];
double lf[ND_ND];
double nfxlf[ND_ND];
double A[ND_ND];
double es[ND_ND];
double dr0[ND_ND];
double dr1[ND_ND];
double te[ND_ND];
double sumtan[ND_ND];
double tangent[ND_ND];

double CellCellFlux,TanFlux;

double L2,L2max,L2maxmax,R2;
double tolabs = 1E-10;
double tolrel = 1E-6;

int iterCGS,totaliters;
int iterCGSmax = 10000;
int sdtmax = 100000;
int iterL2max,sdtL2max;

double alphaCGS,beta,delta,delta0,delta1,dtq;

double rhols_num,rhols_den,rhols_avg;

double total_mass;

cell_t c,c_nb,c0,c1;

Thread *tc;
Thread *tc_nb;
Thread *tc0;
Thread *tc1;

face_t f;
Thread *tf;

Node *node;
Node *node0;
Node *node1;

Domain *d;
d = Get_Domain(1);

int failsafe = 0;
int iterGS,iterGSinner;
int iterGSmax = 10000;
double L2ini;
double sum;



FILE *fplog;
FILE *fplog2;

PRF_GSYNC();
EXCHANGE_UDMI(d,0,UDM_max);

if (curr_ts%vf_N==0)
{

        Message0("Execution of Volume Transport Solver has begun\n\n");

        /* Calculate max thermal diffusivity in domain */
        alpmax = 0.0;
        thread_loop_c(tc,d)
        {
                begin_c_loop_int(c,tc)
                {
```

```
                alpha = C_K_L(c,tc)/(C_R(c,tc)*C_CP(c,tc));

                if (alpha>alpmax)
                {
                    alpmax = alpha;
                }
            }
            end_c_loop_int(c,tc)
        }

        PRF_GSYNC();
        alpmax = PRF_GRHIGH1(alpmax);
        PRF_GSYNC();

        /* Setting C1 */

        /*C1 = 10.0*alpmax;*/
        /*C1 = alpmax + 9*(1-sf)*alpmax; /* New algo for diffusivity */

        thread_loop_c(tc,d)
        {
            begin_c_loop_int(c,tc)
            {
                /*C_UDMI(c,tc,25) = 10.0*alpmax;*/
                C_UDMI(c,tc,25) = 1.6*alpmax + 8.4*(1.0-C_UDMI(c,tc,1))*alpmax;                /*C1 = alpmax + 9*(1-
sf)*alpmax; New algo for diffusivity */
            }
            end_c_loop_int(c,tc)
        }
        PRF_GSYNC();
        EXCHANGE_UDMI(d,25,25);
```

```
                    /* Calculating diffusivity at faces */
                    thread_loop_c(tc,d)
                    {
                       begin_c_loop_int(c,tc)
                       {
                          c_face_loop(c,tc,n)
                              {
                                        f = C_FACE(c,tc,n);
                                        tf = C_FACE_THREAD(c,tc,n);

                              if(BOUNDARY_FACE_THREAD_P(tf))
                              {
                                        C_UDMI(c,tc,35+7*f_offset+n) = 0;
                              }
                              else
                              {
                                        c0 = F_C0(f,tf);
                                                            tc0 = THREAD_T0(tf);

                                                            c1 = F_C1(f,tf);
                                                            tc1 = THREAD_T1(tf);

                                                            INTERIOR_FACE_GEOMETRY(f,tf,A,ds,es,A_by_es,dr0,dr1);

                                                            d0 = NV_MAG(dr0) + 1E-25;
                                                            d1 = NV_MAG(dr1) + 1E-25;

                                                            C_UDMI(c,tc,35+7*f_offset+n) = (C_UDMI(c0,tc0,25)*(1/d0) +
C_UDMI(c1,tc1,25)*(1/d1))/((1/d0)+(1/d1)+1E-25);
                              }
```

```
        }
   }
   end_c_loop_int(c,tc)
}
PRF_GSYNC();
EXCHANGE_UDMI(d,35+7*f_offset,35+8*f_offset-1);


/* Calculating sub time step size limits*/
sub_dt_min = (CURRENT_TIMESTEP)/50;
sub_dt_max = (CURRENT_TIMESTEP)/10;
sub_dt_factor = 1.0007;
sub_dt = sub_dt_min/sub_dt_factor;

/* Adding volume source term contribution pre sub time stepping */
max_vf = 0.0;
max_evf = 0.0;
thread_loop_c(tc,d)
{
        begin_c_loop_int(c,tc)
        {
                /* C_UDMI(c,tc,18) = C_UDMI(c,tc,17)*C_UDMI(c,tc,19)/C_UDMI(c,tc,20); */
                C_UDMI(c,tc,18) = C_UDMI(c,tc,17)*C_UDMI(c,tc,21)/C_UDMI(c,tc,20);
                C_UDMI(c,tc,14) = C_UDMI(c,tc,18) - (1-C_UDMI(c,tc,24));
                C_UDMI(c,tc,14) = (C_UDMI(c,tc,14)>0)*C_UDMI(c,tc,14);
                max_vf = MAX(max_vf,C_UDMI(c,tc,18));
                max_evf = MAX(max_evf,C_UDMI(c,tc,14));
        }
        end_c_loop_int(c,tc)
}
```

```
PRF_GSYNC();
EXCHANGE_UDMI(d,18,18);
EXCHANGE_UDMI(d,14,14);

max_vf = PRF_GRHIGH1(max_vf);
max_evf = PRF_GRHIGH1(max_evf);
PRF_GSYNC();
max_vf = max_vf*1.1;
max_evf = max_evf*1.1;

/* Start of sub time stepping */
dvfdt = INFINITY;
count = 0;

/*reslog = fopen("ResLog.txt","a");*/
L2maxmax = 0.0;
totaliters = 0;

if(I_AM_NODE_ZERO_P)
{
        fplog = fopen("SteadyStateLog.txt","a");
        fplog2 = fopen("DetailedLog.txt","a");
}

while(dvfdt>=vftol&&count<sdtmax)
/* while(dvfdt>=vftol&&count<1) */
{
        count = count + 1;

        if(I_AM_NODE_ZERO_P && (curr_ts==100&&count<=10 ))
        {
```

```
        fprintf(fplog2,"Start of sub dt = %d :\n",count);
}

/* Setting sub dt size for this sub dt */
sub_dt = sub_dt*sub_dt_factor;
sub_dt = MAX(sub_dt_min,sub_dt);
sub_dt = MIN(sub_dt_max,sub_dt);

thread_loop_c(tc,d)
{
        begin_c_loop_int(c,tc)
        {
                C_UDMI(c,tc,16) = C_UDMI(c,tc,18);
                C_UDMI(c,tc,13) = C_UDMI(c,tc,14);
        }
        end_c_loop_int(c,tc)
}

PRF_GSYNC();
EXCHANGE_UDMI(d,16,16);
EXCHANGE_UDMI(d,13,13);

/* Calculating nodal values */
thread_loop_c(tc,d)
{
        begin_c_loop_int(c,tc)
        {
                c_node_loop(c,tc,n)
                {
                        node = C_NODE(c,tc,n);
                        N_UDMI(node,1) = 0; /* Initializing all N_UDMIs to 0 */
                    }
                }
                end_c_loop_int(c,tc)
            }

            thread_loop_c(tc,d)
            {
                begin_c_loop_int(c,tc)
                {
                    c_node_loop(c,tc,n)
                    {
                        node = C_NODE(c,tc,n);

                        N_UDMI(node,1) = N_UDMI(node,1) +
C_UDMI(c,tc,35+8*f_offset+n)*C_UDMI(c,tc,13); /* evf_node = evf_node + weight*evf_cell */
                    }
                }
                end_c_loop_int(c,tc)
            }

            PRF_GSYNC();
            /*Calculating coeff matrices*/
            thread_loop_c(tc,d)
            {
                begin_c_loop_int(c,tc)
                {
                    C_UDMI(c,tc,33) = 1/sub_dt;
                    C_UDMI(c,tc,34) = C_UDMI(c,tc,13)/sub_dt;

                    c_face_loop(c,tc,n)
                {
                        c_nb = C_UDMI(c,tc,35+f_offset+n);
                        tc_nb = Lookup_Thread(d,C_UDMI(c,tc,35+2*f_offset+n));

                        f = C_FACE(c,tc,n);
                        tf = C_FACE_THREAD(c,tc,n);

                        /* Flux factor */
                        flux_factor = 1.0;
                        if (C_UDMI(c,tc,13)<C_UDMI(c_nb,tc_nb,13))
                        {
                                flux_factor = C_UDMI(c_nb,tc_nb,20)/C_UDMI(c,tc,20);
                        }

                        /* Flux flag */
                        flag = (C_UDMI(c,tc,16)>=(1-C_UDMI(c,tc,24)))||(C_UDMI(c_nb,tc_nb,16)>=(1-
C_UDMI(c,tc,24)));

                        /*BC flag */
                        BCflag = C_UDMI(c,tc,35+5*f_offset+n);

                        /*Cell-to-Cell Flux*/
                        C1 = C_UDMI(c,tc,35+7*f_offset+n);
                        /*C1 = 10;*/
                        C_UDMI(c,tc,35+n) = -
flux_factor*BCflag*flag*C1*C_UDMI(c,tc,35+3*f_offset+n)/(C_UDMI(c,tc,35+4*f_offset+n)*C_UDMI(c,tc,11)); /* Area/(deltaf*vol)
*/
                        C_UDMI(c,tc,33) = C_UDMI(c,tc,33) - C_UDMI(c,tc,35+n);
                        CellCellFlux = C_UDMI(c,tc,35+n)*(C_UDMI(c_nb,tc_nb,14)-C_UDMI(c,tc,14));

                        /*Tangential Flux*/
```

```
TanFlux = 0;
dvf = 0.0;

if (ND_ND==3)
{
        if (!BOUNDARY_FACE_THREAD_P(tf))
        {
                INTERIOR_FACE_GEOMETRY(f,tf,A,ds,es,A_by_es,dr0,dr1);

                /*Calculating nf, lf and nf x lf*/
                NV_VV(lf,=,dr0,-,dr1);

                NV_CROSS(nfxlf,es,lf);

                nftotal = F_NNODES(f,tf);

                /*Calculating edge contribution*/
                sumtan[0] = 0;sumtan[1] = 0;sumtan[2] = 0;
                f_node_loop(f,tf,nn)
                {
                        node0 = F_NODE(f,tf,nn);
                        nn_nb = (nn+1)%nftotal;
                        node1 = F_NODE(f,tf,nn_nb);

                        node0Loc[0] = NODE_X(node0);
                        node0Loc[1] = NODE_Y(node0);
                        node0Loc[2] = NODE_Z(node0);

                        node1Loc[0] = NODE_X(node1);
                        node1Loc[1] = NODE_Y(node1);
                        node1Loc[2] = NODE_Z(node1);
                        NV_VV(te,=,node1Loc,-,node0Loc);

                        dvf = MAX(dvf,fabs(N_UDMI(node0,1)-N_UDMI(node1,1)));
                        vfedge = 0.5*(N_UDMI(node0,1)+N_UDMI(node1,1));

                        NV_V_VS(sumtan,=,sumtan,+,te,*,vfedge);
                    }

                    TanFlux = NV_DOT(sumtan,nfxlf);
                }

                TanFlux =
flux_factor*BCflag*flag*C1*C_UDMI(c,tc,35+6*f_offset+n)*TanFlux/(C_UDMI(c,tc,35+4*f_offset+n)*C_UDMI(c,tc,11));
            }
            else
            {
                if (!BOUNDARY_FACE_THREAD_P(tf))
                {
                    /*Calc tangent*/
                    node0 = F_NODE(f,tf,0);
                    node1 = F_NODE(f,tf,1);

                    node0Loc[0] = NODE_X(node0);
                    node0Loc[1] = NODE_Y(node0);

                    node1Loc[0] = NODE_X(node1);
                    node1Loc[1] = NODE_Y(node1);

                    NV_VV(tangent,=,node1Loc,-,node0Loc);
```

```
                                /*Calc lf.t */
                                INTERIOR_FACE_GEOMETRY(f,tf,A,ds,es,A_by_es,dr0,dr1)
                                NV_VV(lf,=,dr0,-,dr1);
                                lfdottang = NV_DOT(lf,tangent);

                                /*Calculating skew term contribution*/
                                dvf = N_UDMI(node1,1)-N_UDMI(node0,1);
                                skewterm = dvf*lfdottang/(ds*pow(A_by_es,2));

                                TanFlux =
flux_factor*BCflag*flag*C1*C_UDMI(c,tc,35+3*f_offset+n)*C_UDMI(c,tc,35+6*f_offset+n)*skewterm/C_UDMI(c,tc,11);
                            }
                        }

                        dvf = MIN(dvf,1);

                        /* Next two lines turn off the flux limiter */
                        CellCellFlux = TanFlux;
                        dvf = 0.0;

                        TanFlux = MIN((1-
dvf)*fabs(CellCellFlux),fabs(TanFlux))*TanFlux/(fabs(TanFlux)+1E-25); /* Tan Flux limiter */

                        C_UDMI(c,tc,34) = C_UDMI(c,tc,34) - TanFlux;
                    }

                    /* Jacobi Preconditioning */
                    c_face_loop(c,tc,n)
                    {
                        C_UDMI(c,tc,35+n) = C_UDMI(c,tc,35+n)/C_UDMI(c,tc,33); /* A_nb = A_nb/AO; */
                    }
```

```
                        C_UDMI(c,tc,34) = C_UDMI(c,tc,34)/C_UDMI(c,tc,33); /* Q = Q/AO */
                        C_UDMI(c,tc,33) = 1.0; /* AO = 1 */
                    }
                    end_c_loop_int(c,tc)
                }

                /* Compute Residual Vector */
                thread_loop_c(tc,d)
                {
                    begin_c_loop_int(c,tc)
                    {
                        C_UDMI(c,tc,26) = C_UDMI(c,tc,34) - C_UDMI(c,tc,33)*C_UDMI(c,tc,14);      /*RR[k] = B[k]
- AO[k]*phi[k];*/

                        c_face_loop(c,tc,n)
                        {
                            c_nb = C_UDMI(c,tc,35+f_offset+n);
                            tc_nb = Lookup_Thread(d,C_UDMI(c,tc,35+2*f_offset+n));

                            C_UDMI(c,tc,26) = C_UDMI(c,tc,26) - C_UDMI(c,tc,35+n)*C_UDMI(c_nb,tc_nb,14);
    /*RR[k] = RR[k] - Anb*phinb;*/
                        }

                        C_UDMI(c,tc,27) = C_UDMI(c,tc,26);   /*RR0[k] = RR[k];*/
                        C_UDMI(c,tc,28) = 0.0;/*ds[k] = 0.0;*/
                        C_UDMI(c,tc,29) = 0.0;/*d[k] = 0.0;*/
                        C_UDMI(c,tc,30) = 0.0;/*g[k] = 0.0;*/
                    }
                    end_c_loop_int(c,tc)
                }
```

```
alphaCGS = 1.0;
delta = 1.0;

R2 = INFINITY;
L2 = INFINITY;
L2max = 0.0;
iterCGS = 0;

while(R2>tolrel && L2>tolabs && iterCGS<iterCGSmax)
/* while(iterCGS<1) */
{
        iterCGS = iterCGS + 1;

        /* Compute Inner Product */
        delta0 = delta;
        delta = 0.0;
        thread_loop_c(tc,d)
        {
                begin_c_loop_int(c,tc)
                {
                        delta = delta + C_UDMI(c,tc,27)*C_UDMI(c,tc,26);      /*delta = delta +
RR0[k]*RR[k];*/

                }
                end_c_loop_int(c,tc)
        }

        PRF_GSYNC();
        delta = PRF_GRSUM1(delta);
        PRF_GSYNC();
        beta = delta/(delta0+1E-25);
```

```
                    thread_loop_c(tc,d)
                    {
                            begin_c_loop_int(c,tc)
                            {
                                    C_UDMI(c,tc,28) = C_UDMI(c,tc,26) + beta*C_UDMI(c,tc,30); /*ds[k] = RR[k] +
beta*g[k];*/

                                    C_UDMI(c,tc,31) = C_UDMI(c,tc,29);   /*d0[k] = d[k];*/
                                    C_UDMI(c,tc,29) = C_UDMI(c,tc,28) + beta*(C_UDMI(c,tc,30) +
beta*C_UDMI(c,tc,31));          /*d[k] = ds[k] + beta*(g[k] + beta*d0[k]);*/
                            }
                            end_c_loop_int(c,tc)
                    }

                    PRF_GSYNC();
                    EXCHANGE_UDMI(d,29,29);


                    /*Compute vector q=[A][D]*/
                    thread_loop_c(tc,d)
                    {
                            begin_c_loop_int(c,tc)
                            {
                                    C_UDMI(c,tc,32) = C_UDMI(c,tc,33)*C_UDMI(c,tc,29);          /*q[k] = AO[k]*d[k];*/

                                    c_face_loop(c,tc,n)
                                    {
                                            c_nb = C_UDMI(c,tc,35+f_offset+n);
                                            tc_nb = Lookup_Thread(d,C_UDMI(c,tc,35+2*f_offset+n));

                                            C_UDMI(c,tc,32) = C_UDMI(c,tc,32) +
C_UDMI(c,tc,35+n)*C_UDMI(c_nb,tc_nb,29);  /*q[k] = q[k] + Anb*dnb;*/
```

```
        }
    }
    end_c_loop_int(c,tc)
}

/*Compute DT*q*/
dtq = 0.0;
thread_loop_c(tc,d)
{
    begin_c_loop_int(c,tc)
    {
        dtq = dtq + C_UDMI(c,tc,27)*C_UDMI(c,tc,32);        /*dtq = dtq + RR0(k)*q(k);*/
    }
    end_c_loop_int(c,tc)
}

PRF_GSYNC();
dtq = PRF_GRSUM1(dtq);
alphaCGS = delta/(dtq+1E-25);

thread_loop_c(tc,d)
{
    begin_c_loop_int(c,tc)
    {
        C_UDMI(c,tc,30) = C_UDMI(c,tc,28) - alphaCGS*C_UDMI(c,tc,32);    /*g(:) = ds(:) -
alpha*q(:);*/
    }
    end_c_loop_int(c,tc)
}
```

```
                    /* Update Solution */
                    thread_loop_c(tc,d)
                    {
                            begin_c_loop_int(c,tc)
                            {
                                    C_UDMI(c,tc,14) = C_UDMI(c,tc,14) +
alphaCGS*(C_UDMI(c,tc,28)+C_UDMI(c,tc,30));             /*SSS(:) = SSS(:) + alpha*(ds(:)+g(:));*/

                            }
                            end_c_loop_int(c,tc)
                    }

                    PRF_GSYNC();
                    EXCHANGE_UDMI(d,14,14);

                    /* Compute Residual Vector*/
                    thread_loop_c(tc,d)
                    {
                            begin_c_loop_int(c,tc)
                            {
                                    C_UDMI(c,tc,26) = C_UDMI(c,tc,34) - C_UDMI(c,tc,33)*C_UDMI(c,tc,14);
    /*RR[k] = B[k] - AO[k]*phi[k];*/

                                    c_face_loop(c,tc,n)
                                    {
                                            c_nb = C_UDMI(c,tc,35+f_offset+n);
                                            tc_nb = Lookup_Thread(d,C_UDMI(c,tc,35+2*f_offset+n));

                                            C_UDMI(c,tc,26) = C_UDMI(c,tc,26) -
C_UDMI(c,tc,35+n)*C_UDMI(c_nb,tc_nb,14);  /*RR[k] = RR[k] - Anb*phinb;*/
                                    }
```

```
		}
		end_c_loop_int(c,tc)
	}

	/*Compute L2NORM*/
	delta1 = 0.0;
	cell_count = 0;
	thread_loop_c(tc,d)
	{
		begin_c_loop_int(c,tc)
		{
			delta1 = delta1 + C_UDMI(c,tc,26)*C_UDMI(c,tc,26);   /*delta1 = delta1 +
RR(k)*RR(k);*/
			cell_count = cell_count + 1;
		}
		end_c_loop_int(c,tc)
	}

	PRF_GSYNC();
	delta1 = PRF_GRSUM1(delta1);
	cell_count = PRF_GRSUM1(cell_count);
	PRF_GSYNC();
	L2 = pow(MAX(0,delta1/cell_count),0.5);
	/* L2 = pow(delta1,0.5); */

	if(iterCGS==1)
	{
		L2ini = L2;
	}

	if(L2>L2max)
```

```
            {
                L2max = L2;
            }

            R2 = L2/(L2max+1E-25);

            if(I_AM_NODE_ZERO_P && (curr_ts==100&&count<=10 ))
            {
                fprintf(fplog2,"iter = %d, L2 = %g, R2 = %g\n",iterCGS,L2,R2);
            }

        }

        if((L2/L2ini) >= 1E2)
        {
            if(I_AM_NODE_ZERO_P)
            {
                current_time = CURRENT_TIME;
                fprintf(fplog,"FAILSAFE triggered at subdt = %d, t = %g s\n",count,current_time);
                fprintf(fplog,"CGS solver info @ end: %d iters, inital L2 = %g, final L2  = %g, final R2 = %g,
max L2 = %g\n",iterCGS,L2ini,L2,R2,L2max);
            }

            /* Reset solution */
            thread_loop_c(tc,d)
            {
                begin_c_loop_int(c,tc)
                {
                    C_UDMI(c,tc,14) = C_UDMI(c,tc,13);
                }
                end_c_loop_int(c,tc)
                }

                /*GS Sweeps*/
                iterGS = 0;
                R2 = INFINITY;
                L2 = INFINITY;
                L2max = 0.0;
                while(R2>tolrel && L2>tolabs && iterGS<iterGSmax)
                {
                        /*GS sweep*/
                        iterGSinner = 0;
                        while(iterGSinner<25)
                        {
                                iterGSinner = iterGSinner + 1;

                                thread_loop_c(tc,d)
                                {
                                        begin_c_loop_int(c,tc)
                                        {
                                                sum = 0.0;
                                                c_face_loop(c,tc,n)
                                                {
                                                        c_nb = C_UDMI(c,tc,35+f_offset+n);
                                                        tc_nb = Lookup_Thread(d,C_UDMI(c,tc,35+2*f_offset+n));

                                                        sum = sum +
(C_UDMI(c,tc,35+n))*(C_UDMI(c_nb,tc_nb,14));/* sum = sum + Anb*phinb */
                                                }

                                                C_UDMI(c,tc,14) = (C_UDMI(c,tc,34) - sum)/C_UDMI(c,tc,33); /*
phiO = (B - sum(Anbi*phi_nbi))/AO */
```

```
                                        }
                                        end_c_loop_int(c,tc)
                                    }
                                }

                                iterGS = iterGS + iterGSinner;

                                PRF_GSYNC();
                                EXCHANGE_UDMI(d,14,14);

                                /*Residual Calc*/
                                L2 = 0.0;
                                thread_loop_c(tc,d)
                                {
                                    begin_c_loop_int(c,tc)
                                    {
                                        C_UDMI(c,tc,26) = C_UDMI(c,tc,33)*C_UDMI(c,tc,14) - C_UDMI(c,tc,34);/*
res = AO*phiO - B */
                                        c_face_loop(c,tc,n)
                                        {
                                            c_nb = C_UDMI(c,tc,35+f_offset+n);
                                            tc_nb = Lookup_Thread(d,C_UDMI(c,tc,35+2*f_offset+n));

                                            C_UDMI(c,tc,26) = C_UDMI(c,tc,26) +
C_UDMI(c,tc,35+n)*C_UDMI(c_nb,tc_nb,14); /* res = res + Anb*phi_nb */
                                        }
                                        L2 = L2 + C_UDMI(c,tc,26)*C_UDMI(c,tc,26); /* res = res*res */
                                    }
                                    end_c_loop_int(c,tc)
                                }
```

```
        L2 = PRF_GRSUM1(L2);
        L2 = pow(L2/cell_count,0.5);

        if(L2>L2max)
        {
            L2max = L2;
        }

        R2 = L2/(L2max+1E-25);

        if(I_AM_NODE_ZERO_P && (curr_ts==100&&count<=10 ))
        {
            fprintf(fplog2,"iter = %d, L2 = %g, R2 = %g\n",iterGS,L2,R2);
        }
    }

    if(I_AM_NODE_ZERO_P)
    {
        fprintf(fplog,"GS Solver converged in %d iters, L2 = %g, R2 = %g\n\n",iterGS,L2,R2);
    }
}

/* Calculating vol. weighted dvf/dt to check for steady state */
dvfdt2 = 0.0;
cell_count = 0;
total_mass = 0.0;
thread_loop_c(tc,d)
{
    begin_c_loop_int(c,tc)
    {
```

```
                    C_UDMI(c,tc,18) =  C_UDMI(c,tc,14) + (1-C_UDMI(c,tc,24)) + (C_UDMI(c,tc,16)<(1-
C_UDMI(c,tc,24)))*(C_UDMI(c,tc,16)-(1-C_UDMI(c,tc,24)));

                    C_UDMI(c,tc,18) = MIN(C_UDMI(c,tc,18),max_vf);
                    C_UDMI(c,tc,18) = MAX(0,C_UDMI(c,tc,18));

                    C_UDMI(c,tc,14) = C_UDMI(c,tc,18) - (1-C_UDMI(c,tc,24));
                    C_UDMI(c,tc,14) = (C_UDMI(c,tc,14)>0)*C_UDMI(c,tc,14);

                    C_UDMI(c,tc,14) = MIN(C_UDMI(c,tc,14),max_evf);
                    C_UDMI(c,tc,14) = MAX(0,C_UDMI(c,tc,14));

                    dvfdt2 = dvfdt2 + pow((C_UDMI(c,tc,18)-C_UDMI(c,tc,16))/sub_dt,2)*C_UDMI(c,tc,11);

                    cell_count = cell_count + 1;

                    total_mass = total_mass + C_UDMI(c,tc,18)*C_UDMI(c,tc,20)*C_UDMI(c,tc,11);
               }
               end_c_loop_int(c,tc)
          }

          PRF_GSYNC();
          dvfdt2 = PRF_GRSUM1(dvfdt2);
          cell_count = PRF_GRSUM1(cell_count);
          total_mass = PRF_GRSUM1(total_mass);
          PRF_GSYNC();

          dvfdt = pow(dvfdt2/total_vol,0.5);
          EXCHANGE_UDMI(d,14,14);
          EXCHANGE_UDMI(d,18,18);
```

```
		if(L2max>L2maxmax)
		{
			L2maxmax = L2max;
			iterL2max = (1-failsafe)*iterCGS + failsafe*iterGS;
			sdtL2max = count;
		}

		/*if(I_AM_NODE_ZERO_P && ( (curr_ts==2700&&(count<=10 || count>=2300)) || curr_ts==2800) )*/
		if(I_AM_NODE_ZERO_P && (curr_ts==100&&count<=10 ))
		{
			fprintf(fplog2,"sub dt = %d, dvfdt = %g\n\n",count,dvfdt);
		}

		totaliters = totaliters + (1-failsafe)*iterCGS + failsafe*iterGS;
		failsafe = 0;
	}

	/*Updating prev time step vol fractions & density*/
	thread_loop_c(tc,d)
	{
		begin_c_loop_int(c,tc)
		{
			C_UDMI(c,tc,17) = C_UDMI(c,tc,18);
			C_UDMI(c,tc,21) = C_UDMI(c,tc,20);
		}
		end_c_loop_int(c,tc)
	}

	PRF_GSYNC();
	EXCHANGE_UDMI(d,17,17);
	EXCHANGE_UDMI(d,21,21);
```

```
                    if(I_AM_NODE_ZERO_P)
                    {
                              if(dvfdt>vftol)
                              {
                                        Message0("Steady state not achieved at current time-step, writing to log file\n\n");
                                        current_time = CURRENT_TIME;
                                        fprintf(fplog,"Convergence FAILED at t = %g s, #sub-timesteps = %d, total # iters = %d,\ndfdt = %g,
max residual = %g @ sub dt = %d, iter = %d\nTotal Mass =
%g\n\n",current_time,count,totaliters,dvfdt,L2maxmax,sdtL2max,iterL2max,total_mass);
                                        /*fprintf(fplog,"Convergence FAILED at t = %g s, #sub-timesteps = %d, total # iters = %d,\ndfdt = %g,
avg. ini residual = %g, avg. end residual = %g\nTotal Mass =
%g\n\n",current_time,count,totaliters,dvfdt,L2iniavg,L2endavg,total_mass);*/
                              }
                              else
                              {
                                        Message0("Converged in %d sub time steps, dfdt = %g\n\n",count,dvfdt);
                                        current_time = CURRENT_TIME;
                                        fprintf(fplog,"Convergence ACHIEVED at t = %g s, #sub-timesteps = %d, total # iters = %d,\ndfdt = %g,
max residual = %g @ sub dt = %d, iter = %d\nTotal Mass =
%g\n\n",current_time,count,totaliters,dvfdt,L2maxmax,sdtL2max,iterL2max,total_mass);
                                        /*fprintf(fplog,"Convergence ACHIEVED at t = %g s, #sub-timesteps = %d, total # iters = %d,\ndfdt =
%g, avg. ini residual = %g, avg. end residual = %g\nTotal Mass =
%g\n\n",current_time,count,totaliters,dvfdt,L2iniavg,L2endavg,total_mass);*/
                              }
                              fclose(fplog);
                              fclose(fplog2);
                    }

          }
```

```
#endif /* !RP_HOST */

PRF_GSYNC();

#if !RP_NODE /* SERIAL OR HOST ONLY */

/*

*/


#endif /* !RP_NODE */
```