A NUMERICAL INVESTIGATION OF A SOLIDIFICATION MODEL FOR SNOW MICRO-STRUCTURE METAMORPHOSIS

by

Micah Johnson

A thesis

submitted in partial fulfillment

of the requirements for the degree of

Master of Science in Mechanical Engineering

Boise State University

August 2015

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the thesis submitted by

Micah Johnson

Thesis Title: A Numerical Investigation of a Solidification Model for Snow Micro-Structure Metamorphosis

Date of Final Oral Examination: 29 June 2015

The following individuals read and discussed the thesis submitted by student Micah Johnson, and they evaluated his presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Inanc Senocak, Ph.D.	Chair, Supervisory Committee
Andrew Slaughter, Ph.D.	Member, Supervisory Committee
Donna Calhoun, Ph.D.	Member, Supervisory Committee

The final reading approval of the thesis was granted by Inanc Senocak, Ph.D., Chair of the Supervisory Committee. The thesis was approved for the Graduate College by John R. Pelton, Ph.D., Dean of the Graduate College.

Dedicated to my wife who has seen very little of me in the last two years.

ACKNOWLEDGMENTS

I would like to thank several people who have enabled me through every phase of this thesis. First, I would like to thank Dr. Inanc Senocak for helping me pursue an education in computational fluid dynamics and guiding my thesis. Second, I would like to thank Dr. Andrew Slaughter for inspiring me to pursue snow science and mentoring me through several topics that will be invaluable as I move forward in my career. I would also like to thank Eric Whiting and the MOOSE development team for providing me the opportunity to create a MOOSE-based application. Edward Adams and David Walters have generously shared μ-ct images of snow micro-structure that they produced at the Subzero Science and Engineering Research Facility for which I am very grateful. I would like to acknowledge Mathis Plapp, who was willing to answer my questions regarding his model during the summer of 2014.

This research made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy under Contract No. DEAC0705ID14517.

This material is also based upon work supported by the National Science Foundation under Grant No. (1229709). The Kestrel HPC Cluster were used in some of the simulations.

ABSTRACT

Understanding the way snow changes during environmental events has wide spread benefits ranging from avalanche prediction to water conservation. Snow on a micro-structural level constantly changes from the moment it forms. Snow metamorphosis is driven by the transport of water vapor. Thus far it has only been investigated as a pure diffusion process in dry snow, despite observations that suggest that natural convection may have a role in the heat and mass transport in snow packs. This thesis research numerically explores the role of transport processes in the context of snow metamorphism. An existing solidification model is reformulated to simulate the effects of natural convection on the evolution of snow micro-structure. The finite-element based Multi-physics Object Oriented Simulation Environment (MOOSE) is adopted for numerical computations. The solidification model is based on the phase field equation, which captures an evolving interface without a direct interface tracking method. This equation is coupled with the Navier-Stokes equations to simulate fluid flow over complex solid boundaries, formed from micro-tomographic images of snow micro-structure. The numerical framework is then applied to investigate the role of diffusion and convection on the snow micro-structure. The results demonstrate diffusion-based metamorphism on real snow. And natural convection is shown to develop in snow micro-structure under realistic conditions prior to coupling with phase change effects.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FEM – Finite Element Method

PDE – Partial Differential Equation

MOOSE – Multi-physics Object Oriented Simulation Environment

INL – Idaho National Laboratory

ISSW – International Snow Science Workshop

SLF – Translated from Swiss as the ”Institute for Snow and Avalanche Research

EM – Equi-temperature Metamorphism

TGDM – Temperature Gradient Driven Metamorphism

Re – Reynolds Number

Ra – Rayleigh Number

μ**-ct** – Micro-Computed Tomography

CFD – Computational Fluid Dynamics

HPC – High Performance Computer

API – Application Program Interface

C++ – Preferred programming language

PJFNK – Preconditioned Jacobian Free Newton Krylov

SMP – Single Matrix Preconditioning

CHAPTER 1

INTRODUCTION

Avalanches are unique in the fact that they present a large problem for both infrastructure and recreationists. They are deceiving natural disasters because snow is rarely perceived as dangerous. Most interactions with snow for recreationists is either in a flat environment or in a highly regulated mountain area like a ski resort. Unfortunately, some of the most sought after terrain by winter sport enthusiasts is in places where avalanches are common. In 30 years (1978-2007), 329 people were directly killed by avalanches in Canada [9]. The problem is even worse in the U.S., which is demonstrated by the upward trend of avalanche caused fatalities seen in Figure 1.1. While a majority of avalanche related fatalities in North America are recreation related, in Asia they are typically unsuspecting villages that are found in avalanche terrain. This is confirmed by the most recent avalanches that have happened in Southern Asia. In Afghanistan, the Panjshir province experienced avalanche that struck a village and killed approximately 150 people [1]. During the same year, Nepal suffered a large earthquake, which triggered an avalanche and buried more than 300 people [38]. In terms of infrastructure, governments everywhere contend with minimizing the economic losses that can be caused by avalanches. The 2013 Alaska State Hazard Mitigation plan recalls the 1999-2000 winter season, which had a large number of snow slides that caused a total of 11 million dollars in damage to

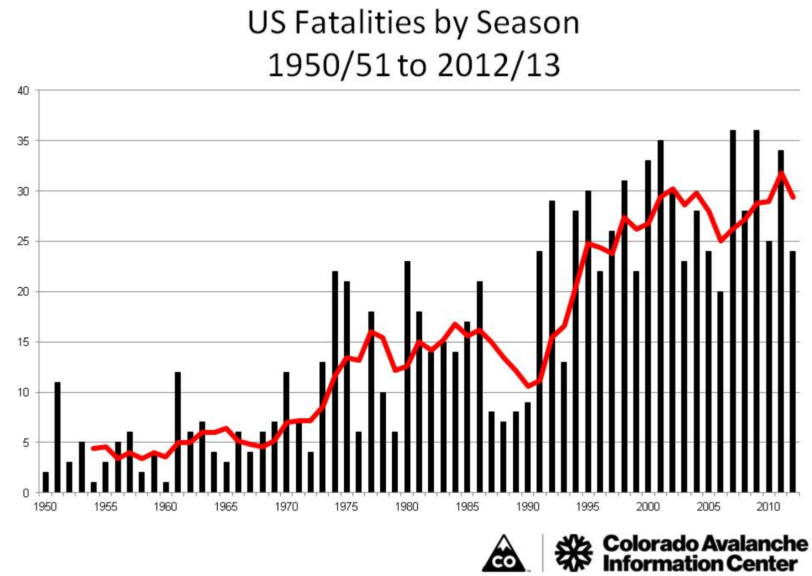


Figure 1.1: Avalanche caused fatalities in the U.S. [5] Graphic Source: Colorado Avalanche Information Center.

various communities around the state. Outside of North America, avalanches pose as a costly threat to many roads and railways in Europe. The Swiss government spends around 400,000 U.S. dollars per acre on preemptive structures to mitigate avalanche risk [20]. While this works, it is clearly expensive. And many avalanche mitigation programs world wide face the same issue of expense. This is why Switzerland and others have whole research institutions devoted to understanding avalanches and ultimately developing better mitigation and prediction tools.

Accurate avalanche prediction has the potential to save millions of dollars in building costs and hundreds of lives annually. Unfortunately, how a snow flake becomes part of an avalanche is a complex process. A large portion of the snow science literature is focused on how avalanche conditions develop and is the basis for avalanche forecasting. For an avalanche forecaster, one of the goals is to reduce the unknowns associated with how the snow pack's properties change with time and space [30]. The variability that exists in snow pack's thermal and mechanical properties are heavily dependent on the micro-structure. As a result, how these conditions develop for avalanches are largely observed and studied at the micro-structure level [36].

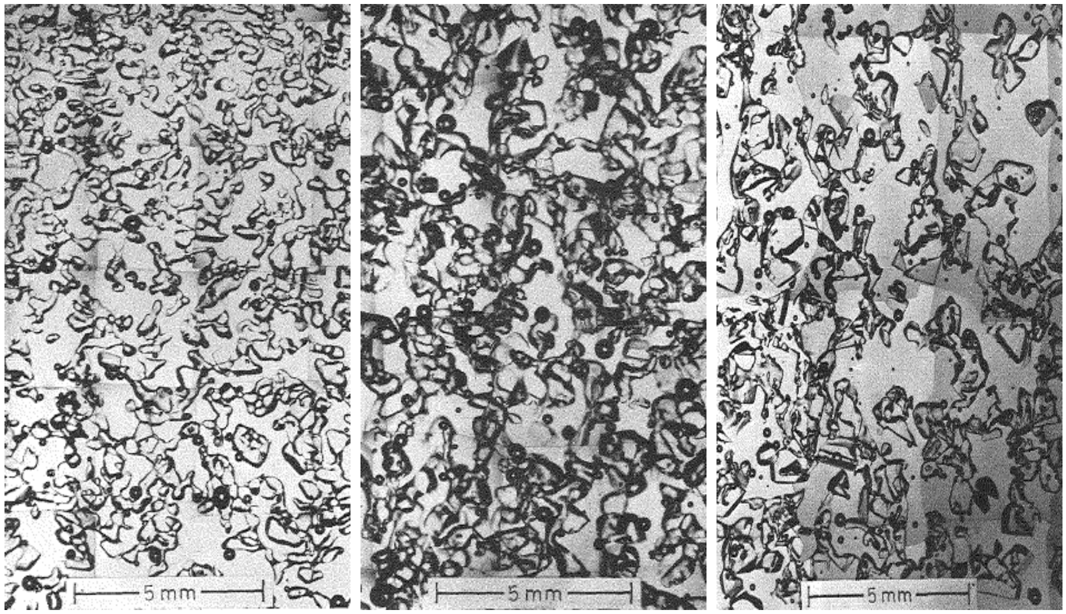
In dry snow (where little to no liquid water is present in the snow pack), slab avalanches are often triggered by humans and are responsible for most recreation related fatalities [37]. A slab avalanche is described best as a large section of multi-layer snow that releases all at once. Currently, the accepted view on the mode of failure in dry snow slabs is the collapse of a weak layer [35]. McClung and Schaerer points out that for dry avalanche conditions to exist, a weak layer must be present [31]. These weak layers can be briefly described as a layer of weakly bonded snow grains and ultimately presents itself as shear plane for the release of a slab. As the layers above a weak layer are continuously loaded, they are in constant state of deforming and bonding themselves together. The net bonds that destroyed or generated between snow grains determines the stability of the slope. At some point, a loaded weak layer will collapse either through artificial triggering (e.g., skiers) or through a significant weather event. In Figure 1.2 is an observation made in the field with a partially collapsed weak layer. These weak layers can form *in situ* due to environmental conditions, or can form on the surface and become buried. Investigating weak layer formation and evolution is a large part of understanding dry avalanche conditions and has been the focus of many snow scientists since the inception of the field.

Many elements have been studied as mechanisms for weak layer development and most can be classified under snow metamorphism. Snow metamorphosis occurs either through equi-temperature metamorphosis (EM) and temperature gradient driven metamorphosis (TGDM). EM is the rounding and breaking up process of a snow flake and is best described as the minimization of free energy. As the snow flake

Figure 1.2: A partially collapsed weak layer, Graphic Source: [36]

breaks up, it forms bonds with other existing snow grains. This process is commonly referred to as sintering. Ultimately, EM will form stronger snow and is not known to cause avalanches. TGDM describes the change of the micro-structure as a result of the water vapor being given up, transported, and deposited. It is also considered the main developer of the types of snow crystals that comprise a weak layer [13]. Birkeland [8] described their formation as the deposition of water vapor as it travels along temperature gradients that exist in the air. To better understand the transition of fresh snow to the snow that exists in a weak layer, Akitaya [2] took a sequence of photos, which are shown in Figure 1.3. The images show how these types of crystals can be developed under certain temperature gradients, which were found by Akitaya [2] to be necessary to form them in situ. While EM will form rounded snow grains, TGDM forms highly angular snow grains as demonstrated in Figure 1.3. This is why grains formed through this process make up weak layers. The high angularity of the snow grains creates stress concentrations where the snow bonds to other grains.

How a weak layer forms is the result of several factors but, ultimately the under-



(a) Initial Compact Snow (b) One Week (c) Six Weeks

Figure 1.3: Snow metamorphism over the course of six weeks with a temperature gradient of 19°C/m and a mean temperature of -12 ° C, Graphics Source: [2]

lying mechanism is the transportation of water vapor and how it interacts with the surrounding snow. While it is generally accepted that vapor diffusion is the main mode of mass transport, the role that natural convection has in the evolution of the micro-structure of snow has been debated [3, 4, 12, 34, 41]. Colbeck [12] discussed that the role of natural convection or temperature driven flow is not fully understood and suggested more experiments are needed to investigate the claims made about its effects on the micro-structure. Those who have pointed out the existence of thermal convection in the snow pack have also called for further investigation. Thus, it is the goal of this work to initiate a detailed numerical investigation of natural convection in the snow pack.

1.1 Works Published

Works published during the course of this study:

- A. Slaughter, M. Johnson, M. Tonks, D. Gaston, C. Permann, J. Peterson, D. Andrs, J. Miller, E. Adams, D. Walters, "MOOSE: A Framework to Enable Rapid Advances and Collaboration in Modeling Snow and Avalanches." International Snow Science Workshop Proceedings, October 2014

CHAPTER 2

TECHNICAL BACKGROUND

Investigating the dynamics of snow micro-structure numerically provides valuable insight as to what impacts certain environmental conditions have on the metamorphic processes of snow. As discussed in Chapter 1, the role that natural convection has in the micro-structural evolution is not well understood. Powers [34] investigated natural convection through a series of lab experiments and concluded that natural convection was indeed possible in snow and greatly enhanced heat and mass transfer. Sturm and Johnson [41] explored the role of natural convection in the snow pack and found there to be non-homogeneities in the temperature distribution, which could not be explained by classical diffusion models. They continued to discuss that even though there are some numerical models that exist, they were unable to capture the importance of convection due to simplifying assumptions.

2.1 Snow Metamorphism Models

Through the last 30 years, numerical models for snow metamorphism have been developed for a wide range of purposes. There have been many attempts at modeling how the snow micro-structure changes under various conditions. This section is focused on the models that have been developed for snow metamorphism with the intent of providing brief insight to their complexity, incorporated physics, and ultimately their suitability for the present study.

The Swiss Federal Institute for Snow and Avalanche Research (SLF) developed one of the more popular models for snow metamorphism. It is a one dimensional finite element formulation encompassing conductive heat transfer and estimated source terms describing latent, sensible heat, and radiation. The vapor is transported using a binary gas diffusion equation, which is driven by the temperature and has source terms for melt outflow, wind scour, and sublimation. The SLF's [6] model is able to capture all three phases of water by implementing two conservation of mass equations for liquid water and vapor. Their model is comprehensive, even accounting for stress-strain equations in the ice phase. Since the model is exclusively one dimensional, it is of limited use in the discussion surrounding a problem like natural convection, which is inherently multidimensional without making additional generalizations.

Miller et al. [32] proposed that a generalized approach was needed for modeling the many aspects of snow metamorphism. They provided a model using heat conduction coupled with a vapor diffusion equation that used simple geometry to model snow metamorphism in two dimensions. Using round elements, they were able to model phase change under various temperature conditions. While their model provides valuable insight to vapor diffusion effects, it assumes that natural convection is negligible. The use of round elements is potentially a reasonable approximation to snow micro-structure, however to continue a discussion on the effects that natural convection will have on the metamorphosis process, the geometry modeled should represent snow as best as possible.

The numerical models that incorporate better representations of snow do so through micro-computed tomographic (μ-ct) images of snow micro-structure. Flin et al. [19] demonstrated the role that curvature has on the EM of snow using small three dimensional μ-ct images. Calonne et al. [11] produced a numerical approach for estimating the effective thermal conductivity using three dimensional scans of real snow. Kaempfer and Plapp [23] presented a phase field model to better understand how the micro-structure is changing under imposed temperature gradients. Their model included a phase field equation coupled with heat and mass transport equations. The mass equation included vapor diffusion and a simple source term representing sublimation. These three studies are able to study temperature effects on the micro-structure but ultimately, all of them assume natural convection to be a negligible effect.

Though the aforementioned models cover a wide degree of complexity in geometry and incorporated physics, all of them assume that natural convection is not a major source of mass transfer. There is a limited number of works that have explored this phenomena in snow. Klever [24] explored the role of natural convection numerically using a stream function for the velocity field and convected heat and mass equations. The model is simplified by using spatially average material properties including a generalized porosity term that is representative of the types of snow. The study was focused on parameter sensitivity through viable perturbations to the Rayleigh number and porosity. It was found that while no convection can occur in small rounded snow, it is always occurring in new snow. Klever [24] was able to demonstrate that weak layers are capable of temperature driven vapor flow, albeit sensitive to environmental changes. The experiment was exclusively a numerical exploration of the possible variations of the conditions that would occur in a snow environment. A conclusion from the study was that natural convection is more prevalent than previously thought.

Another study interested in the role natural convection has in the snow pack was done by Powers [34]. This study was focused on the role that phase change has on the heat transfer and what Rayleigh numbers are relevant in sloped layers. He drew similar conclusions as Klever [24]. Both studies examine natural convection in the snow context, however since both models used spatially averaged values they are not able to distinguish effects that this phenomena has on the evolution of the micro-structure.

In other fields such as material science, the role of natural convection in multi-phase media has been studied in more detail. These studies are often solidification problems observing the formation of metal alloys. Under the same definitions, snow metamorphism falls into the same fundamental category of solidification physics. Despite the differences in material properties, solidification models present similar theory used for modeling snow. It is through the fundamental theory of solidification that the importance of convection can be further demonstrated. Beckermann et al. [7] developed a model for simulating binary alloy melt using the phase field equation and incorporated heat and mass transport equations. Like the snow science community, the role that natural convection has in the micro-structure of binary alloys was not well understood and was the source of motivation for their work. The study showed their model's capacity to grow dendrites and presents several benchmark problems. They successfully replicated Stokes flows despite the use of a diffuse interface to separate phases and concluded additional studies were needed to fully investigate the underlying physics. Tan and Zabaras [42] also showed the capacity for dendritic growth of metal alloys in three dimensions using the level-set method and included natural convection. The model solved the mass, momentum, heat, species, and level set equations using the finite element method. Validation was performed in part by comparing tip velocities of dendrites with previously developed phase field models.

Several demonstrations of the model's complexity were shown, including the ability to capture flow between dendrites. They were able to conclude that by incorporating natural convection the dendrites grew 22 percent faster.

The complexity of structures simulated in the solidification of metal alloys is unrivaled by the snow science's contributions to micro-structural metamorphism even though dendrites are observed in snow frequently. Various elements of a complex, generalized, snow metamorphism model exist in the literature. Existing models incorporate a wide variety of physics, but have different goals as seen in the SLF's [6] model, which was to provide an efficient tool capable modeling many physics, but only in one dimension. Other models incorporate less physics, but use more complex geometry through the use of μ-ct scans similar to the models presented by Flin et al. [19] or Calonne et al. [11]. Additionally, there are a limited number of studies investigating natural convection through numerical modeling of snow metamorphism. Most models are similar to the numerical investigation posed by Klever [24], which are too simplified to evaluate the effects that natural convection has on snow metamorphism.

Kaempfer and Plapp's [23] model appears to be the most advanced model available in terms of incorporated physics and the use of complex geometry for modeling micro-structural evolution. Their phase field approach is novel to the snow science community and allows for material properties to be phase dependent through interpolating equations, which eliminates the need to spatially averaged values. Many other snow metamorphism models assume that vapor diffusion through the pore space is the only mode of mass transport and neglect convected terms altogether [6, 23, 32].

2.2 Thesis Statement

Currently, the snow science community lacks a metamorphosis model capable of natural convection at a micro-structural level. Other models have been presented in the past, but proved to be too simplistic and limited due to assumptions. In other fields, modeling micro-structure evolution in material solidification and the inclusion of natural convection have been explored with varying degrees of success and complexity. Literature regarding snow science shows that there is limited focus in modeling this effect in snow, despite the observations made in the field that indicate its role in heat and mass transfer. It is clear that modeling the formation of avalanche conditions requires a comprehensive snow micro-structure model in which certain environmental conditions can be explored numerically. To this end, a new model is needed that should be able to demonstrate this phenomena within μ-ct scans of snow micro-structure.

There is a need for a snow metamorphism model that is capable of investigating the role of natural convection at the micro-structure scale. The goal of this research is to layout in detail the components necessary to investigate the role of natural convection in snow micro-structural evolution. Thus, a numerical investigation of diffusion and natural convection in snow is initiated in this work through a generalized set of equations representing solidification of a pure substance. The specific objectives of the present thesis research are as follows:

1. Investigate the limits for simulating fluid flow.

2. Determine the suitability of the phase field equation for modeling flow past solid surfaces.

3. Provide a phase change formulation for a dry snow context.

4. Demonstrate the role of natural convection in a passive snowpack

The first objective in this work is achieved by investigating the limits of the implemented numerical scheme by replicating common benchmarks problems. There are three cases simulated: the lid driven cavity, flow over a cylinder, and natural convection in a vertically heated enclosure. The goal of these tests is to validate the fluid flow physics before coupling them to a solidification model. And they are also motivated by the fact that one of the end goals is to simulate flow through snow micro-structure, which covers a whole host of complex geometry. The lid driven cavity is a well known benchmark problem in the computational fluid dynamics (CFD) community. Ghia et al. [22] presented a benchmark quality solution of the problem shown. The problem, in brief, is a square enclosure in which the top is driving the flow. The second simulation investigates the feasibility and accuracy of modeling flow over the complex shapes that are seen in snow micro-structure. This is done through another common CFD benchmark problem. The problem is uniform flow over an infinitely long cylinder that is perpendicular to the flow. Several authors have investigated this problem in depth, providing metrics for comparison such as the re-circulation length behind the cylinder. The last test for validating the fluid physics simulates natural convection in a vertically heated enclosure. This problem is a square box that has a quiescent fluid. One of the vertical walls is instantly heated and due to buoyancy effects the warmer fluid rises. Using these problems, the scope and capacity of the fluid flow equations are assessed.

The second thesis statement is achieved by re-simulating the aforementioned problems but coupling the fluid flow physics to the phase field equation. Since the end goal is to model changing micro-structure, a method is required to represent the ice and vapor as a function of time and space. The phase field equation is used to model dynamic and static interfaces. This means that the interface between ice and vapor is resolved and not directly tracked. The domain of the fluid flow problems can then be redefined by specifying ice for all the solid surfaces. By modeling fluid flow using the phase field equation to represent a stationary surface, the fluid flow physics can be validated again while coupled with the phase.

The third statement is achieved by expanding upon Kaempfer and Plapp's [23] model so that it is capable of capturing natural convection. An experiment is used to validate the model's capacity for simulating the evolution of the snow micro-structure to ensure proper implementation. The problem in brief is a migrating air bubble trapped in ice. As the bubble experiences a temperature gradient, it begins to move. This happens because one side of the bubble is warmer than the other and begins to give up water vapor, which is then deposited on the opposite, cooler side. This experiment was also replicated by Kaempfer and Plapp [23] and the simulation parameters are shown in Figure 2.1.

The final objective is achieved by simulating natural convection in a passive snowpack under realistic conditions. Passive in this case is defined such that no phase changes will occur. This is accomplished by simulating a snow micro-structure undergoing a temperature gradient while on a slope.

These statements ultimately will provide two things. The first is how well suited the numerical framework is for modeling natural convection in snow. The second is that these problems will uncover any necessary requirements for moving forward and simulating a fully coupled problem.

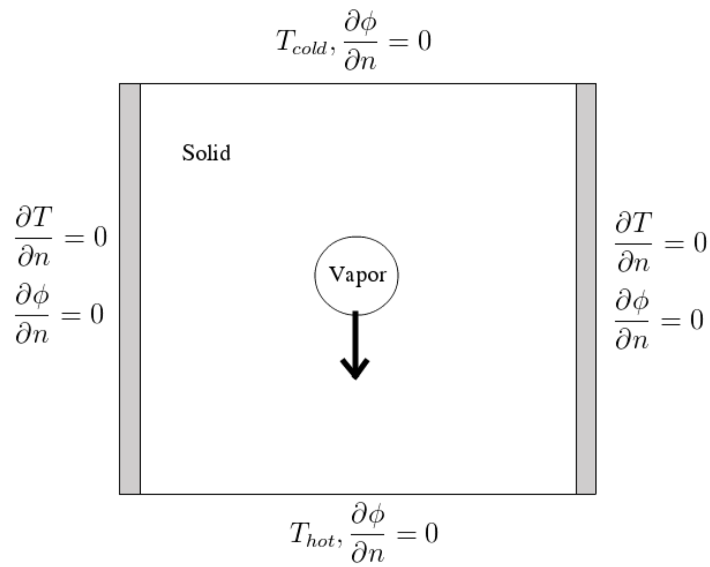


Figure 2.1: Simulation conditions for bubble migration through a block of ice

2.3 Governing Equations for Snow Metamorphism

The model developed here will extend the work of Kaempfer and Plapp [23]. As discussed in Section 2.1, they model solidification of snow by taking advantage of the phase field equation. In brief, the phase field method is a common treatment for multi-phase problems where physics and material properties are phase dependent. The method introduces an extra variable that represents the phase of the media, which is denoted ϕ. This technique is useful because material properties can be defined as continuous functions of ϕ which ranges from -1 (water vapor) to 1 (ice). Additionally, this approach is not a direct interface tracking method and resolves the interfaces between the two phases continuously, making it ideal for modeling phase change. The two phases are separated by a diffuse interface of some assumed thickness (W). While this section aims to build upon the work done by Kaempfer and Plapp [23], the model presented here will solve the equations using the finite element method as opposed to the finite difference method that was actually used in the original work.

2.3.1 Equations for Dry Snow Metamorphism by Diffusion Transport

Kaempfer and Plapp's [23] model is summarized in this section. Their model incorporates a single source of vapor through sublimation and thus only models dry snow. The micro-structural evolution can be modeled by three equations governing the phase evolution (2.1), heat diffusion (2.2), and vapor diffusion (2.3). The unknown variables are the phase (ϕ), temperature (T), and a dimensionless water vapor concentration (X).

$$\tau \frac{\partial \phi}{\partial t} = W^2 \nabla^2 \phi + (\phi - \phi^3) + \lambda [X - X_{eq}](1 - \phi^2)^2, \tag{2.1}$$

$$C(\phi) \frac{\partial T}{\partial t} = \nabla \cdot [K(\phi) \nabla T] + \frac{L_{sg}}{2} \frac{\partial \phi}{\partial t}, \tag{2.2}$$

$$\frac{\partial X}{\partial t} = \nabla \cdot [D(\phi) \nabla X] - \frac{1}{2} \frac{\partial \phi}{\partial t}, \tag{2.3}$$

where X is defined as the difference between vapor density and the saturated vapor density normalized by the density of ice as shown in Equation 2.4.

$$X = \frac{\rho_{vapor} - \rho_{v.s.}(T_{ref})}{\rho_{ice}}. \tag{2.4}$$

Equation 2.1 is a classic phase field equation that represents the bulk phase with the first two terms on the right hand side. The third term drives the phase change based on the availability of vapor or X. τ is a relaxation time coefficient and λ is a coupling constant. Ultimately, together they control the rate of interface migration.

Both coefficients are formulated in terms of the capillary length and the interface kinetic coefficient, which are common phase field terms. The derivation of these terms can be seen in Kaempfer and Plapp's [23] work. The other two equations are diffusion equations with phase dependent source terms. In accordance with phase field techniques, the material properties in Equation 2.2, and 2.3 are linearly interpolated in Equations 2.5. Notice that Equation 2.5c tends towards zero as $\phi \rightarrow 1$, thus the water vapor does not diffuse through ice. This is known as a one-sided model in the phase field community.

$$C(\phi) = C_{ice}\frac{1+\phi}{2} + C_{air}\frac{1-\phi}{2} \tag{2.5a}$$

$$K(\phi) = K_{ice}\frac{1+\phi}{2} + K_{air}\frac{1-\phi}{2} \tag{2.5b}$$

$$D(\phi) = D_v\frac{1-\phi}{2} \tag{2.5c}$$

In modeling ice evolution and water vapor, large disparities are present in time scales. Kaempfer and Plapp [23] recommend treating the evolution as a quasi-steady problem. Using this assumption, the ice density, diffusion terms, and latent heat were scaled with little impact on the final solution to enable larger time steps. While this technique worked, it is unclear how the proposed adaptations (presented in Section 5.1) of these equations should be scaled. An implicit time stepping scheme is used, which allows for larger time steps than in the explicit time stepping performed in the original work. However, for validation purposes, the results shown in Section 5.1 were produced using the same equations implemented by Kaempfer and Plapp [23]. And for clarity, the following is a brief description of the temporal scaling process.

The velocity of the phase interface is an important metric in the phase field method, thus the scaling is applied such that the interface velocity is preserved. This is accomplished by applying the scaling only to the Equations 2.1, 2.2, and 2.3 and not the property definitions. The scaling process can be seen in Equation 2.6 where $5 \times 10^{-5} < \xi < 1$.

$$D(\phi) \rightarrow D(\phi)\xi, \tag{2.6a}$$

$$K(\phi) \rightarrow K(\phi)\xi, \tag{2.6b}$$

$$L_{sg} \rightarrow L_{sg}\xi, \tag{2.6c}$$

$$X \rightarrow \frac{X}{\xi}, \tag{2.6d}$$

$$X_{eq} \rightarrow \frac{X_{eq}}{\xi}, \tag{2.6e}$$

$$\lambda \rightarrow \lambda\xi. \tag{2.6f}$$

Note that Equation 2.1 is unaffected by the scaling since it was applied to X, X_{eq}, and λ in the source term, which cancels the effect. The vapor diffusion equation's scaling is manipulated to resemble the scaling on the heat diffusion equation. Initially the scaling thats applied to Equation 2.3 is shown in Equation 2.7.

$$\frac{1}{\xi}\frac{\partial X}{\partial t} = \nabla \cdot [\xi D(\phi)\nabla \frac{1}{\xi} X] - \frac{1}{2}\frac{\partial \phi}{\partial t} \tag{2.7}$$

Since ξ is a constant, the scaling on the diffusion term is canceled. By multiplying Equation 2.7 by ξ, both of the transport equations are now scaled exactly the same as seen in Equations 2.9 and 2.8.

$$C(\phi)\frac{\partial T}{\partial t} = \xi\nabla \cdot [K(\phi)\nabla T] + \frac{\xi L_{sg}}{2}\frac{\partial \phi}{\partial t} \tag{2.8}$$

$$\frac{\partial X}{\partial t} = \xi\nabla \cdot [D(\phi)\nabla X] - \frac{\xi}{2}\frac{\partial \phi}{\partial t} \tag{2.9}$$

The model makes several assumptions, but there are two important ones that should be known. The first is that the gas in the snow is near saturation with water. The second is that the metamorphosis process is isotropic, and thus all growth will happen proportional to the available water vapor. For a more in depth derivation of these equations and assumptions, please refer to Kaempfer and Plapp [23].

2.3.2 Equations for Incorporating Fluid Flow through Snow Micro-structure

To explore natural convection, a set of equations representing the fluid flow in three dimensions is presented for the proposed application. Thus, Equation 2.10 is the Navier-Stokes equation

$$\rho\Big(\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla \vec{V}\Big) = \nabla \cdot \sigma + \rho \vec{b}, \tag{2.10}$$

where μ is the dynamic viscosity, and ρ is the fluid's density. By assuming the flow is incompressible, the continuity equation for the Equation 2.10 becomes

$$\nabla \cdot \vec{V} = 0. \tag{2.11}$$

Though this equation will be used to model the conservation of mass of saturated air, it is expected that the flow will be considerably slower than a Mach number of 0.3, which is considered the limited of this assumption [33]. Additionally, if the fluid is considered Newtonian, Stokes Law is defined such that,

$$\sigma = -pI + \mu \nabla \vec{V}, \tag{2.12}$$

where I is the identity tensor. By using the Boussinesq approximation, the Navier-Stokes equations are able to account for density variations without violating the classic continuity equations for incompressible flows. This is accomplished by incorporating an appropriate forcing term representing buoyancy. The Boussinesq assumption implies that the change in density is unaffected by pressure and velocity. In the case of snow, thermal variations are driving the velocity field and potentially transporting vapor. The forcing term $\vec{b}$ in Equation 2.10 becomes a function of temperature and is multiplied by the gravity vector as shown in 2.13

$$\vec{b} = [1 - \alpha(T - T_{ref})]\vec{g}, \tag{2.13}$$

where α is the coefficient of thermal expansion and $\vec{g}$ is the gravitational vector. Since the formulation is based on the phase field equation, the solid/vapor interface will actually be resolved and will not have a boundary condition applied, leaving out the no-slip condition. To mitigate this problem, another forcing term can be added to the momentum equation to penalize the flow in the solid phase. Combining techniques shown by Beckermann et al. [7] and Shyy [39], the forcing term is applied in a one sided fashion in solid phase only, demonstrated in Equation 2.14

$$M_d = \mu h \frac{1+\phi}{2W^2}\vec{V}. \tag{2.14}$$

This is similar to a Darcy flow forcing term, which states that the resistance to the flow varies linearly to the velocity. h is a dimensionless coefficient that was set to be large enough to best align with one of the benchmark problems, which is demonstrated in Chapter 4. Coupling this equation with Equations 2.8 and 2.9 results in an added term that represents the quantities being advected as in 2.15 and 2.16. Since vapor diffusion is the main mode of vapor transport, it is expected that the advected terms will be at most on the same time scale. Thus, the temporal scaling is applied to the advected terms as well.

$$C(\phi)\frac{\partial T}{\partial t} = \nabla \cdot [K(\phi)\nabla T] - C(\phi)\vec{V} \cdot \nabla T + \frac{L_{sg}}{2}\frac{\partial \phi}{\partial t} \tag{2.15}$$

$$\frac{\partial X}{\partial t} = \nabla \cdot [D(\phi)\nabla X] - \vec{V} \cdot \nabla X - \frac{1}{2}\frac{\partial \phi}{\partial t} \tag{2.16}$$

The final form of Equation 2.10 is

$$\rho_{air}\left(\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla \vec{V}\right) = \nabla \cdot \sigma + \rho_{air}\vec{b} - M_d. \tag{2.17}$$

Equations 2.1, 2.11, 2.15, 2.16, and 2.17 are the equations proposed for a generalized approach to model snow metamorphism capable of natural convection.

2.4 Numerical Methods

2.4.1 Finite Element Method

The Multi-physics Object Oriented Simulation Environment (MOOSE) is a finite element method (FEM) framework developed for solving coupled non-linear equations [26]. This is the numerical framework that is used to solve Equations 2.1, 2.11, 2.15, 2.16, and 2.17. Since the FEM is being utilized, the aforementioned partial differential equations (PDE) must be formulated in their weak form prior to being implemented. This process is demonstrated on Equation 2.17. The first step requires that all terms are one sided such that the entire equation is equal to zero as is shown in Equation

2.18, where

$$\rho_{air}\Big(\frac{\partial\vec{V}}{\partial t} + \vec{V}\cdot\nabla\vec{V}\Big) - \nabla\cdot\sigma - \rho_{air}\vec{b} + M_d = 0. \tag{2.18}$$

Equation 2.18 is then multiplied by weight function and integrated across the entire domain

$$\psi\rho_{air}\Big(\frac{\partial\vec{V}}{\partial t} + \psi\vec{V}\cdot\nabla\vec{V}\Big) - \psi\nabla\cdot\sigma - \psi\rho_{air}\vec{b} + \psi M_d = 0, \tag{2.19}$$

where ψ is the weight function. By integrating Equation 2.19 across the domain Ω, the first version of the residual that would be acceptable for inputting into MOOSE is

$$\int_\Omega \Big[\psi\rho_{air}\Big(\frac{\partial\vec{V}}{\partial t} + \psi\vec{V}\cdot\nabla\vec{V}\Big) - \psi\nabla\cdot\sigma - \psi\rho_{air}\vec{b} + \psi M_d\Big] d\Omega = 0. \tag{2.20}$$

Fortunately the order of Equation 2.20 can be reduced by applying Green's theorem, which states

$$\int_\Omega \psi\nabla\cdot\sigma d\Omega = \int_\Gamma \psi\sigma\cdot\vec{n}d\Gamma - \int_\Omega \nabla\psi\cdot\sigma d\Omega, \tag{2.21}$$

where Γ is a boundary and $\vec{n}$ is the outward normal. This means that any diffusion terms can be reduced by implementing a corresponding boundary condition. When Green's theorem and equation 2.12 are applied to Equation 2.22, the stress term becomes

$$\int_\Gamma \psi(-pI + \mu\nabla\vec{V})\cdot\vec{n}d\Gamma - \int_\Omega \nabla\psi(-pI + \mu\nabla\vec{V})d\Omega. \tag{2.22}$$

Inserting Equation 2.22 back into Equation 2.20 results in the final equation implemented into the MOOSE framework. This process of developing the weak form with reduced order was applied to Equations 2.1, 2.11, 2.15, 2.16, and 2.17, which are shown in their final forms in Appendix A

The MOOSE framework is used to solve each equation using the Galerkin Formulation by default. This means that the same functions will be used for the weight and the trial functions. This has been found in the past to be problematic when advection becomes stronger in advection-diffusion equations similar to the ones being solved in the present work. It is demonstrated in Chapter 4 that this can be a challenging issue but this is not an issue in the case in fulfilling the objectives of this research. The last component of solving these equations is selecting what type of shape function to solve. Fortunately MOOSE has a myriad of shape functions that can be independently assigned to variables. Thus, for the variables representing the temperature (T), vapor concentration (X), pressure (p), and phase (ϕ) will be solved using first order Lagrange trial functions. And only the velocity components (u, v, w) will use second order Lagrange trial functions. At this point, MOOSE assembles the matrices to be solved.

2.4.2 Solver

While MOOSE offers several solvers, for non-linear, coupled PDEs, the developers recommend using the preconditioned Jacobian-Free Newton-Krylov solver (PJFNK). In brief, this is two solving techniques nested within another. PJFNK relies on traditional Newton solver techniques to achieve super linear convergence. Unfortunately, Newton's method relies on the Jacobian to be calculated exactly, which can be exceedingly difficult. To avoid having to store and write a large Jacobian, the derivatives are estimated numerically based on some small perturbation. The use of an approximated Jacobian results in a loss in convergence, thus Krylov subspace linear iterations are implemented to make up for an imperfect Jacobian. In a nested loop, the Generalized Minimization of the Residual (GMRES) is the Krylov method of choice and is used to provide the initial guess for the Newton solver. The Jacobian Free Newton-Krylov Method is summarized in detail by Knoll and Keyes [25]. An unfortunate side effect of using Krylov methods is the necessary use of preconditioning. MOOSE also offers several selections for preconditioning, but Single Matrix Preconditioning (SMP) is again recommended by the developers. By default, MOOSE will disregard the off diagonal terms of the Jacobian unless a preconditioner is selected. SMP allows the user to place off diagonal terms in the Jacobian if they are defined. A summary of the features available in MOOSE is available on their website [26]. To improve the convergence of the method, most of the terms of the Jacobian are formulated and utilized.

2.4.3 Jacobian Formulation

The promise of improved convergence is enough to pursue the formulation of a nearly full Jacobian matrix. Here the Jacobian terms implemented for a single direction of the Navier-Stokes equations. The Jacobian in this context is defined as

$$\mathbf{J}_{i,j} = \frac{\partial R_i}{\partial u_j}, \tag{2.23}$$

where R_i is the residual formed by each weak form implemented and u_j are the variables associated with the problem. However, in the FEM, the solution that is determined is a function called the trial solution. This is defined as

$$u \approx u^h = \sum_{e=1}^{N_{elem}} N^e u^e, \tag{2.24}$$

where N^e is the shape function, u^e is the nodal values, and e and h denote element and global approximations, respectively. Based on this definition, the derivative of anything with respect to the variable has to have the chain rule applied. Thus, the

Jacobian is redefined as

$$\mathbf{J}_{i,j} \approx \frac{\partial R_i}{\partial u_j}\frac{\partial u_j}{\partial u_j^e}, \tag{2.25}$$

Considering the definition in Equation 2.24, it can be observed that

$$\frac{\partial u}{\partial u^e} = \sum_{e=1}^{N_{elem}} N^e, \tag{2.26}$$

$$\frac{\partial \nabla u}{\partial u^e} = \sum_{e=1}^{N_{elem}} \nabla N^e. \tag{2.27}$$

Given the above information, forming the Jacobian entries is demonstrated here. Consider that all integrals are evaluated numerically, the process is briefly demonstrated on the convective acceleration term in Equation 2.20, the Jacobian of that becomes

$$\frac{\partial}{\partial u}\left[\psi\rho_a(\vec{V}\cdot\nabla u)\right] = \psi\rho_a(\vec{V}\cdot\nabla N + N\frac{\partial u}{\partial x}). \tag{2.28}$$

Note, that since $\vec{V}$ has the u component velocity, the product rule has to be applied also to form this section of the Jacobian. The Jacobians for each equation that were implemented are shown in Appendix B.

CHAPTER 3

NUMERICAL SIMULATION OF INCOMPRESSIBLE FLUIDS WITH A FINITE ELEMENT MODEL

A solidification model was defined in Chapter 2 incorporating natural convection for an ice and water vapor matrix that was capable of changing its structure due to the availability of water vapor. Before investigating the role of thermal convection in snow micro-structural evolution, various portions of the physics are validated prior to progressing forward. Specifically, this chapter presents a form of validation for each portion of the physics associated with the problem. The physics presented in Chapter 2 are grouped in three groups that are fluid momentum, natural convection, and ice metamorphosis based on vapor diffusion.

Non-dimensional numbers like the Reynolds number provide a basis to quantify the limits of the current finite element formulation. Fortunately, Navier-Stokes equations have already been solved with accuracy at lower Reynolds numbers using the MOOSE framework. The FEM formulation of Navier-Stokes equation are tested in two ways. The first is through lid driven cavity flow, which is a standard benchmark problem in the CFD community and provides a basis for validation. The second is through another benchmark problem that simulates flow over a cylinder, which provides insight as to how effective the numerical methods are at resolving complex geometries. These two tests are solely governed by the Reynolds number, and thus provides valuable insight to the limits of the numerical methods.

3.1 Lid Driven Cavity Flow

It is known that the finite element method can develop numerical instabilities when advection effects dominate over viscous diffusion [18]. Specifically, this can occur when solving the Navier-Stokes equations using the FEM without any additional stabilization methods. Thus, it is necessary to know what are the limits of the unstabilized form currently implemented in the MOOSE framework. Using the Reynolds number (Re), which describes the ratio of inertial forces to viscous forces [33], equations 2.10 and 2.11 can be tested. The equation for the Reynolds number is shown in Equation 3.1

$$Re = \frac{\rho V L}{\mu}, \tag{3.1}$$

where V is a characteristic velocity, L is a characteristic length, ρ and μ are the fluid's density and dynamic viscosity, respectively. In the case of lid driven cavity flow, V is the lid velocity and L is the side length of the cavity. This case is a square enclosure containing a fluid that has a moving surface that interacts with the fluid. The fluid velocity if specified to be zero on the all the non-moving walls, which is known as the no-slip condition. The top or the lid has a boundary condition that is set to a predetermined constant velocity. Using these boundary conditions, the lid driven cavity case was simulated and compared with the data presented by Ghia et al. [22]. While it is traditional in the fluids community to begin with Re=1000 case, it is expected that the actual Reynolds number is much lower in the thermally driven convection in a snow pack [34]. Thus, the first case simulated was using the lowest Reynolds number that Ghia et al. [22] presented, which was Re=400. As seen

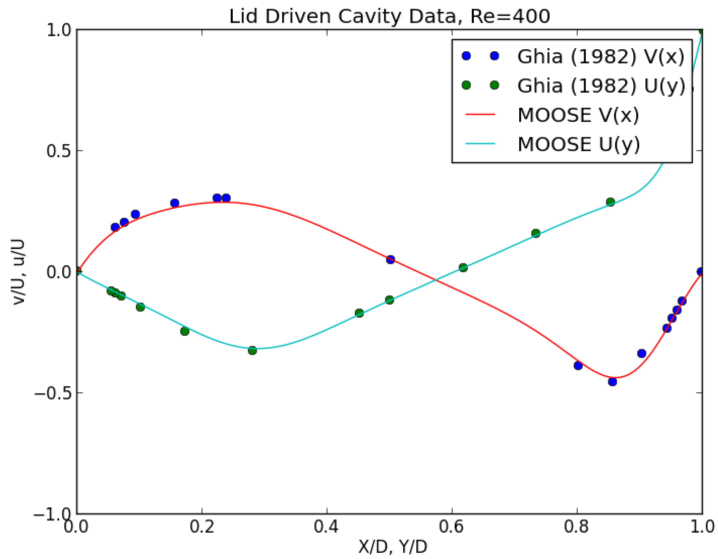


Figure 3.1: Velocity profiles compared between a FEM formulation of the incompressible Navier-Stokes equations simulating a lid driven cavity and compared with data from Ghia et al. [22] at Re=400

in Figure 3.1, the formulation produces values that are in good agreement with the benchmark data. The Re=1000 case was simulated to see if the FEM would produce the correct solution. However, the results proved too erroneous, which suggests the numerical limit was surpassed. A mesh refinement study can be seen in Figures 3.2 and 3.3, which shows that the solution was producing poor results regardless of the mesh size. This demonstrates the current formulation is accurately replicating flows below Re=400. This Reynolds number serves as the upper limit for the rest of work. In the context of snow, this likely means that all the snow simulated will have to be considered deep in the snowpack and unaffected by external wind over the snow.

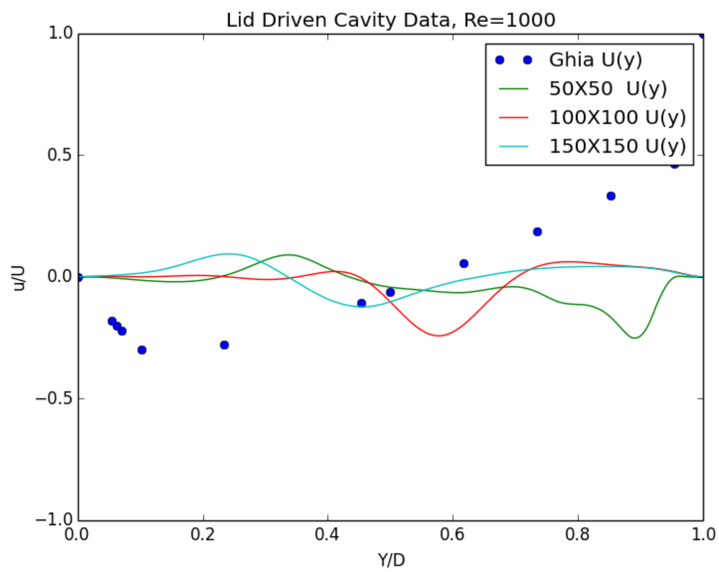


Figure 3.2: X velocity component as a function of Y for the lid driven cavity case at RE=1000

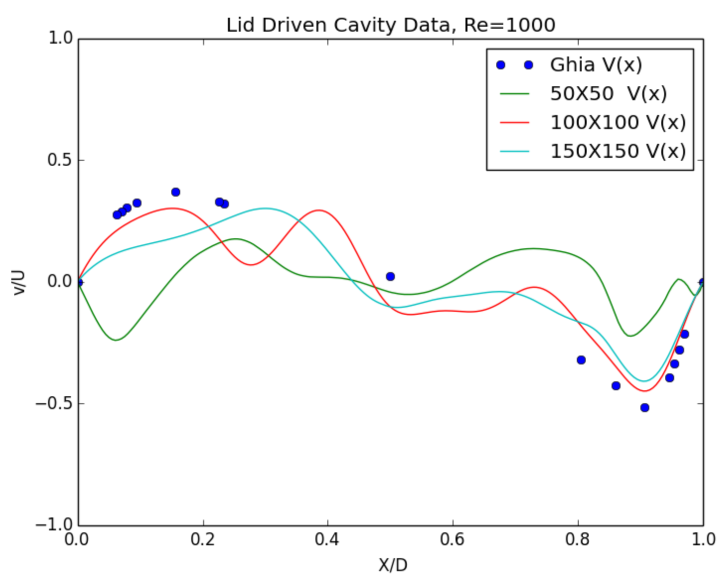


Figure 3.3: Y velocity component as a function of X for the lid driven cavity case at RE=1000

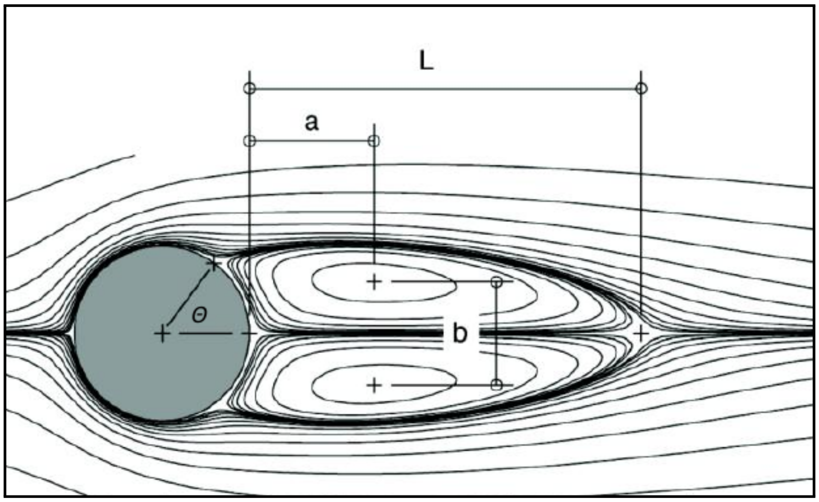


Figure 3.4: Commonly compared features of the flow over a cylinder case. Graphic Source: [15]

3.2 Flow Over a Cylinder

Since one of the identified needs of the snow science community is to present a model using complex geometry, the Navier-Stokes is further validated using an infinite cylinder in cross flow. The outlet has a pressure specified to zero, and the walls parallel to the flow have natural or Neumann boundary conditions equal to zero for the x direction of the velocity component. The vertical velocity component is specified to zero at the walls. On the cylinder surface, the entire velocity vector is equal to zero. The inlet then has a velocity specified at a constant value. In computational fluid dynamics literature, traditional starting points are at a Re=20 and Re=40 for steady flow over a cylinder, where the Reynolds is calculated with V is the inlet velocity, and L is the diameter of the cylinder. Certain features from the Figures 3.5 and 3.6 are compared with other published results to assess the accuracy of the current simulation. Using the same nomenclature in Figure 3.4, Table 3.1 and 3.2 shows a summarized comparison of the features of interest. There is excellent

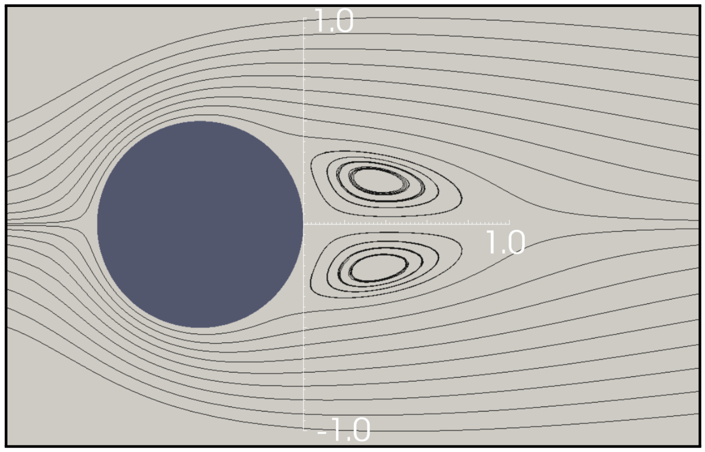


Figure 3.5: Streamlines of the flow over a cylinder case at Re=20

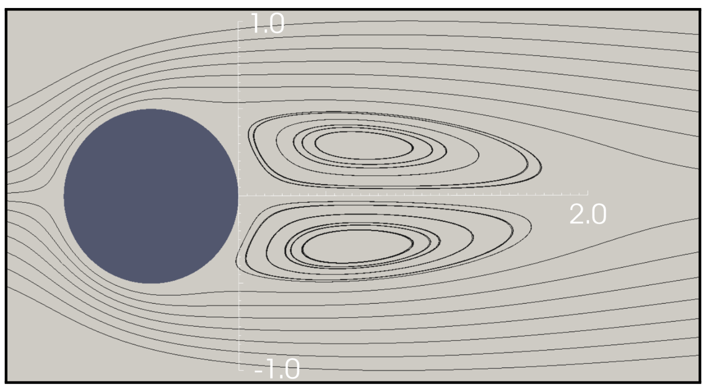


Figure 3.6: Streamlines of the flow over a cylinder case at Re=40

Table 3.1: Summary of the features compared for the flow over a cylinder case at Re=20

Works	L	a	b	θ
Fornberg [21]	0.91			45.7
Dennis and Cheng [17]	0.94			43.7
Coutanceau and Bouard [14]	0.93	0.33	0.46	45.0
Linnick and Fasel [27]	0.93	0.36	0.43	43.5
Brehm and Fasel [10]	0.94	0.36	0.43	43.5
Present Work	0.93	0.36	0.42	44.6

Table 3.2: Summary of the features compared for the flow over a cylinder case at Re=40

Works	L	a	b	θ
Fornberg [21]	2.24			55.6
Dennis and Cheng [17]	2.35			53.8
Coutanceau and Bouard [14]	2.13	0.76	0.59	55.8
Linnick and Fasel [27]	2.28	0.72	0.60	53.6
Brehm and Fasel [10]	2.29	0.72	0.60	52.4
Present Study	2.00	0.70	0.59	54.5

agreement for the Re=20 case. For the Re=40 case, there is similarity with the other data with the exception of the re-circulation length or L of Figure 3.4, which deviates noticeably from the other works shown. As seen in the lid driven cavity case, this is potentially a sign of the upper capacity of the formulation is being used in MOOSE, which potentially points to the under estimation in the re-circulation length. These findings further restrict the scope of the proposed work such that the formulation is exclusively adequate for low Reynolds number flow with limited re-circulation. Fortunately, the Reynolds number in snow is expected to be much lower than 20 and more on the order of Stokes flow Reynolds numbers [34]. Because the flow is slow in the snowpack, re-circulation behind snow grains will not occur. Thus, the formulation is still valid for the current problem.

3.3 Vertically Heated Square Enclosure

The heat transport equation and Navier-Stokes equations are applied to a natural convection benchmark problem to demonstrate a capacity for buoyancy driven flows. The problem is a two dimensional square cavity with the left wall heated, the right wall is cooled and all walls have a no-slip condition applied. This scenario produces

Table 3.3: Maximum u of the mid (x=0.5) cavity velocity profile at varying Rayleigh numbers

Rayleigh Number	10^3	10^4	10^5	10^6
De Vahl Davis [16]	3.634	16.2	34.8	65.33
Manzari [28]	3.68	16.1	34.0	65.4
Mayne et al. [29]	3.6493	16.7198	34.7741	64.6912
Wan et al. (FEM) [43]	3.489	16.122	33.39	65.40
Present Study	3.65	16.12	33.41	

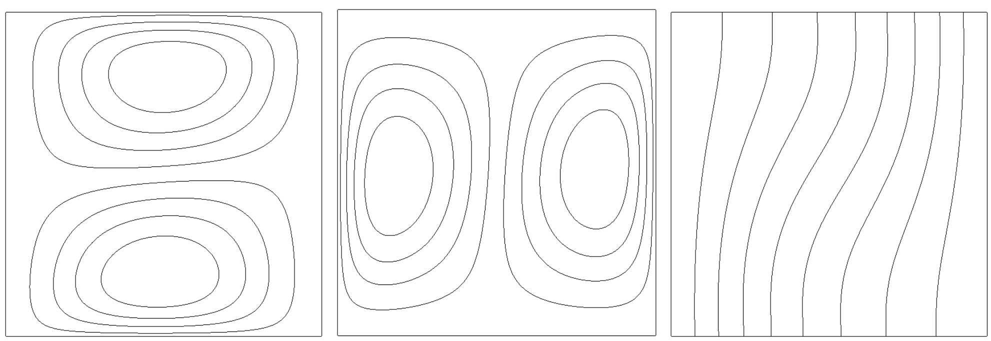
Table 3.4: Maximum v of the mid (y=0.5) cavity velocity profile at varying Rayleigh numbers

Rayleigh Number	10^3	10^4	10^5	10^6
Davis et al. [16]	3.679	19.51	68.22	216.75
Manzari [28]	3.73	19.9	70.0	228
Mayne et al. [29]	3.6962	19.6177	68.6920	220.8331
Wan et al. (FEM) [43]	3.686	19.79	70.63	227.11
Present Study	3.69	19.77	70.53	

clockwise flow and specific flow structures form as a function the Rayleigh number (Ra), which is calculated

$$Ra = \frac{\rho^2 C_p g \alpha L^3 \Delta T}{\mu k}, \tag{3.2}$$

where α is the coefficient of thermal expansion, ρ is the density of the fluid, ΔT is the temperature difference, μ is the dynamic viscosity of the fluid, g is the gravitational constant, C_p is the specific heat, and L is also the side length of the cavity. Figure 3.7 shows the iso-contours and more specifically the flow structure that has developed at Ra=10^3. The structures shown align well with iso-contours presented in benchmark data sets [43]. The data shown in Table 3.3 and 3.4 show that there is excellent agreement with the data provided by several publications.



(a) u velocity component contours (b) v velocity component contours (c) Temperature contours

Figure 3.7: Iso-contours for various quantities of a heated cavity at a Ra=10^3

CHAPTER 4

PHASE FIELD MODELING OF FLOW AROUND SOLID BOUNDARIES

In the previous chapter, it was shown that the MOOSE framework is capable of capturing lower Reynolds and Rayleigh number flows. In simulating the test cases, numerical limits were established and used to limit the scope of the formulation. It has been demonstrated that the implemented form of the FEM is capable of simulating classic fluid problems within the realistic parameters seen in snow. This chapter focuses on results that represent the coupling of the momentum equations with the phase field equation.

4.1 Continuous Application of the No-Slip Condition

The lid driven cavity case is replicated again here but instead of solving 2.10, Equation 2.17 is used to simulate the same case. This provides some validation for the coupling of the momentum equation with the phase field equation. As it was discussed in Chapter 2, the use of the phase field equation results in a solid interface that is resolved continuously, which requires the formulation to have the extra forcing term seen in Equation 2.14. The selection of h from Equation 2.14 is pertinent to penalizing the flow to zero in the solid phase with accuracy. To compare the effects of h, the lid driven cavity case was simulated at a Reynolds number of 400 while varying h. The case was exactly the same as it was in Figure 3.1 except the no-slip boundary conditions are replaced by setting the phase equal to one (solid) and the velocity now has a natural boundary condition of zero on the three non-moving walls. Only one no-slip condition was specified on the lid for the vertical component of the velocity. Additionally, an initial condition is used to demonstrate the no-slip condition is applied with being effected by the velocities boundary conditions. Using a solid phase that was $10W$ thick on the three non-moving walls, the continuous application of the no-slip condition could be observed with little impact from the boundary conditions of the velocity field. Since the top boundary now spans solid and gas, the original Dirichlet boundary condition is modified to be a function of the phase. Thus, the lid boundary condition is applied such that

$$\frac{(1-\phi)}{2}\vec{V}. \tag{4.1}$$

This eliminates the possibility of specifying a velocity in the solid region that is non-zero.

The goal of this case was to determine if accurate results could be produced even though the boundary conditions were different. The selection of h was then determined by qualitatively comparing the velocity profiles shown in Figure 4.1. The value of h determines how accurately a solid boundary is represented. Ideally, the value is chosen to be as high as possible. However, this selection of h can cause numerical instability to occur if chosen to be too large. While not shown, a value of h=1000 was simulated but the results do not improve. Thus, based on Figure 4.1, h= 100 is suitable for the needs of lower Reynolds number flows that are observed in snow.

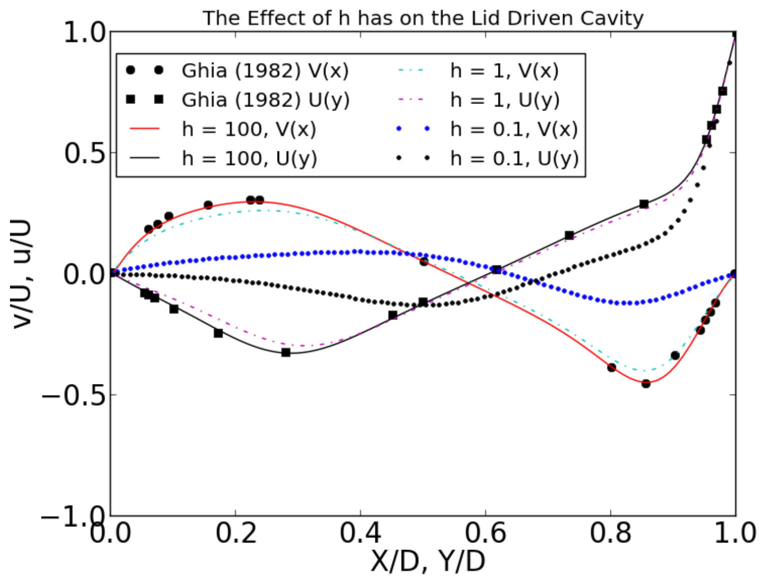
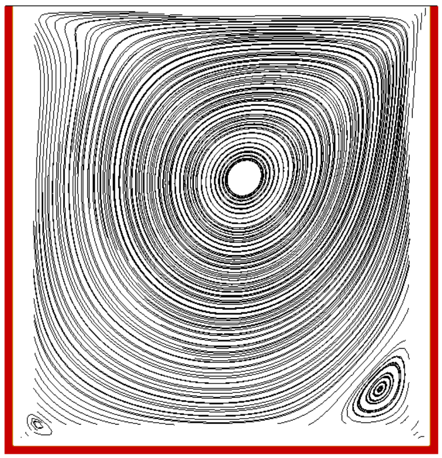
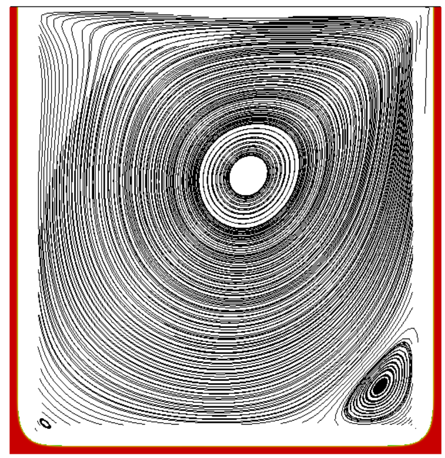


Figure 4.1: The effect h from Equation 2.14 has on the lid driven cavity case at Re = 400 where each wall is specified to be ice, demonstrating the capacity to accurately model flow past resolved solid interfaces.

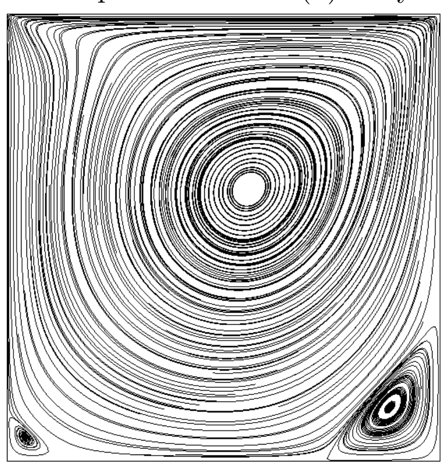
It is also important to point out the existence of the secondary eddies that form in the bottom corners and the effect the current formulation can have on them. Since the geometry bounding the fluid flow is defined via the phase field variable, two simulations are run. The first is to relax the phase field initial condition by exclusively solving with the phase field equation to some point in time. This ensures that ϕ is stable for simulating fluid flow. The second simulation is performed with the fluid flow physics coupled with the phase field equation, using the results from the first simulation as the initial condition for ϕ. Unfortunately, a decision has to be made for the first simulation that can affect the second, which is how far in time is the phase relaxed enough. For a steady solve of the phase field equation, the corners in the lid driven cavity case become significantly round and are no long representing a square cavity. The effect that this has is demonstrated using the streamlines of the lid driven cavity case in the Figure 4.2. It is apparent in Figure 4.2a and 4.2b



(a) Partially relaxed phase



(b) Fully relaxed phase



(c) No phase included

Figure 4.2: The effect the initial condition of ϕ has on the streamlines of lid driven cavity at a Re=400, where $\phi = 1$ is shown in red that borders the domain.

how the relaxation of the initial condition can affect the size of the corner eddies. In general, the phase field coupling has a dampening effect on the flow structures despite the excellent agreement in the lid driven cavity velocity profiles show in Figure 4.1. While this is not ideal, the effect is noted and further restricts the applicability of the current formulation to lower Reynolds numbers. The input file for the lid driven cavity simulation for $h = 100$ can be found in Appendix C.

4.2 Flow Over Complex Geometry

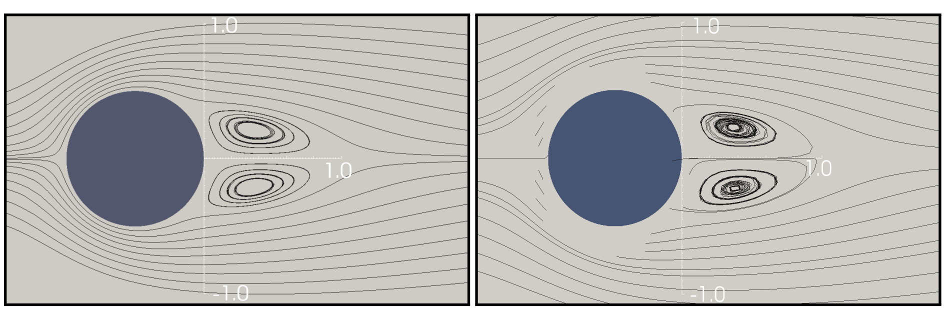
As was shown in the previous chapter, the flow over cylinder case presents a excellent benchmark for a formulation's ability to replicate flow around complex shapes that make up snow grains. In describing the need for the present model, it was pointed out that some of the previous works were limited to simplified geometry. Thus, this section examines the flow over cylinder case in an attempt to further validate the coupling of the Navier-Stokes equation with the phase field equation.

The flow over cylinder case is replicated again here at a Reynolds number of 20. Instead of using a mesh fitted to the cylinder, an initial condition was provided to the flow solver. This represented a cylinder of ice surrounded by the gas phase. The cylinder has a diameter of 1 millimeter. The flow over a cylinder has similar results when compared to the body-fitted solution provided in Chapter 3, but ultimately is falling short of the benchmark. One hypothesis for this that choice of W is related to how the flow is penalized and has great impact on this case. The width of the interface that separates the two phases is determined by the sharp interface limit, which determines the range at which the physics ares still valid. Kaempfer and Plapp [23] demonstrated that the sharp interface limit is ideal but implement a so-called thin interface limit with success. They recommended using interface thicknesses between 1x10^{-6} and 1x10^{-4} meters. As a result, the thickness has been defaulted to 1x10^{-5}meters. Thus, this case was re-simulated to see the impact that the choice of the interface thickness would have on the the re-circulation metrics. Unfortunately, the solution has notable drawbacks in replicating the previous body-fitted mesh, which

Table 4.1: The flow over a cylinder case at Re=20 using various diffuse interface thicknesses (W)

Mesh Type	L	a	b	θ
Body Fitted	0.93	0.36	0.42	44.6
Phase Field, W =1x10^{-4}	1.06	0.40	0.46	43.4*
Phase Field, W =1x10^{-5}	1.02	0.40	0.45	39.1*

is clearly demonstrated in Table 4.1. It should be noted that the separation angles are especially difficult to determine due to the fact that some of the streamlines actually entered into the solid region. These angles are the best approximation but it is subject to significant uncertainty. This also might explain the extended re-circulation length. Extra flow that is passing through the cylinder instead of being full diverted around the cylinder would lead to greater flow in the x direction right behind the cylinder. It is clear that the choice of the interface thickness is critical in replicating flows of high re-circulation. This is shown in Figure 4.3, which shows a side-by-side comparison of the flow over cylinder case at Re=20. Figure 4.3a shows the streamlines from the original simulation of the case shown in Chapter 3. Figure 4.3b shows the streamlines from the phase field representation of the same flow. Notice how the length of the re-circulation bubble behind the cylinder is longer than the original solution. Based

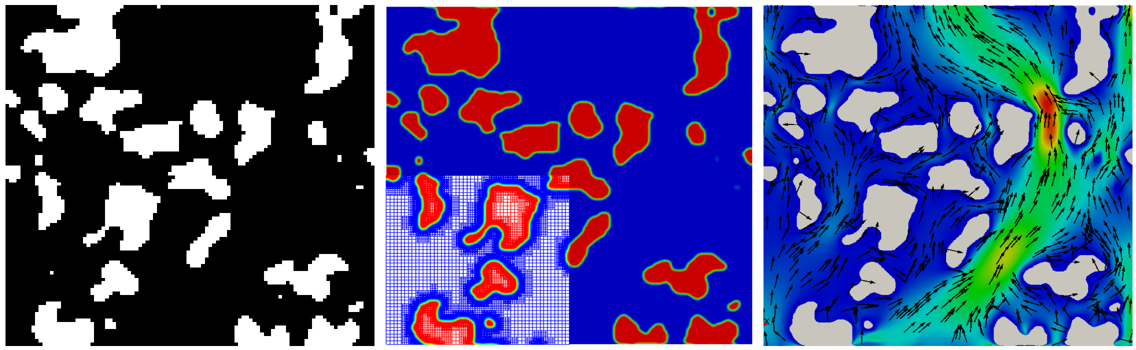


(a) Body fit mesh (b) Phase field representation.

Figure 4.3: A comparison of a body fit mesh versus a phase field representation of flow over cylinder at a Re=20, emphasizing re-circulation lengths

on these results, the current formulation is struggling to replicate the Re=20 case and further results produced should be exclusively used for even lower Reynolds numbers. Fortunately, Powers [34] estimated that the Reynolds number in the snow micro-structure would be on the order of 0.01. The input files for this case at Re=20 and W = $1\text{x}10^{-5}$M can be found in Appendix C.

One of the major weaknesses of the previous models was that they were complex in some capacity but implemented simple geometry. It is demonstrated qualitatively that a real μ-ct image of snow micro-structure can be meshed and used in fluid flow simulations. Figure 4.4 shows how this is accomplished. First, an original μ-ct image of the snow that is roughly 25 square millimeters is used to generate the geometry. Second, the image is converted to a continuous variable, which in this study is ϕ or the phase field variable. A rule of thumb for the mesh and the phase field equation is that at least 3 cells should span the interface thickness in any given place to ensure stability of the phase field around curved surfaces. As in the previous section, two simulations are run and the result is passed as an initial condition to the flow simulation. Third, Figure 4.4c shows the final result, qualitatively demonstrating forced convection through the snow, where the vectors and coloring represent the direction and magnitude of the flow respectively with red being the highest velocity. These simulations also suffer from a similar issue as was shown in the lid driven cavity case bound by the solid phase. This is the length at which the initial condition is simulated. The difference here is that small features can be lost from the original image if the initial condition is simulated to steady state. Here the solution is to only simulate to 1000 seconds, which is relatively small on the interface migration time scale. Even so, the smallest features are distorted and even lost; this is best seen when comparing the right side of Figures 4.4a and 4.4b.



(a) Original μ-ct image of snow (b) Image shown as a relaxed phase field (c) Qualitative flow through snow

Figure 4.4: Qualitative demonstration of simulating fluid flow through a complex geometry produced from real snow images

4.3 Natural Convection Bound by Ice

The vertically heated enclosure case is replicated here to demonstrate the ability to simulate a natural convection flow scenario in geometry represented by the phase field. The case shown with Ra=1000 is the only case examined. The temperature is specified in a 5 mm square box that is encased in ice. Using 1000 seconds to initialize the phase field initial condition, the square corners are preserved. The problem is formulated to snow micro-structure conditions, which means that the properties used here are the actual properties seen in snow. Thus, to provide a reasonable comparison between velocities, the velocity is scaled according to Wan et al. [43], which says that

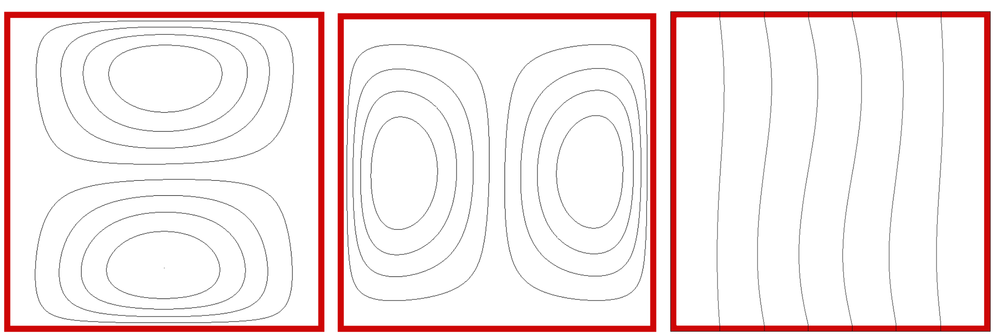
$$u \to \frac{C(\phi = -1) u L_{ch}}{K(\phi = -1)}, \tag{4.2}$$

where $C(\phi = -1)$ and $K(\phi = -1)$ are the vapor's heat capacity and thermal conductivity respectively, and L_{ch} is the characteristic length. The results from the simulation differ significantly from the original solution. Unfortunately, it would

Table 4.2: Maximum velocity components of the mid cavity velocity profile at Rayleigh number of 1000 with and without the phase field included

Work	Max u	Max v
W/o Phase Field	3.65	3.69
W/ Phase Field	1.91	1.91

seem that this formulation is ill-suited for flows driven by a Ra=1000. This is clearly demonstrated by the maximum of velocity components taken along the mid cavity profile (i.e., $u|_{x=0.5}$) shown in Table 4.2. The lower velocities coincides with what was observed in the lid driven cavity case, which was a general dampening effect. Since the velocities are much smaller, this dampening effect is exaggerated. More research in this area is definitely needed to circumvent this issue. The input files for this case are provided for review in Appendix C.



(a) u velocity component contours (b) v velocity component contours (c) Temperature contours

Figure 4.5: Iso-contours for various quantities of a heated cavity formed by solid boundaries generated from the phase field at a Ra=10^3.

The dampening effect is really noticed when comparing T iso-contours in Figures 3.7c and 4.5c. This does of course make sense when considering the velocity was nearly half that of the original solution and thus the advection effect it would have would also be reduced. This might be due to the simulation parameters. Since the box is surrounded in ice, the walls may be acting as a heat reservoir. Recall that the thermal properties are defined as functions of the phase. The thermal properties of the walls are considerably greater than that of the fluid flowing inside. The temperature contours reflect this to a degree as they are slightly “hooked” in to the ice, which only qualitatively validates this hypothesis. Additionally, the scaling process may be contributing to this error. Wan et al. used a scaling process for single phase materials and it may be an incorrect way of scaling this scenario. While this presents a new problem, results here are left to be pursued in the future. The research moves forward because no firm conclusions can be made due to the unknowns of the scaling process and how comparable this problem is to the original.

CHAPTER 5

THERMAL CONVECTION IN A SNOWPACK

5.1 Diffusion Based Snow Metamorphism

The Equations 2.1, 2.8, and 2.9 have been solved, which represents the original diffusion-based model. Following Kaempfer and Plapp [23], an experiment presented by Stehle [40] was used for validation. The experiment was an investigation of temperature induced migration of air bubbles in ice. In brief, the experiment consisted of a block of single crystal ice that was 2.5 cm by 2.0 cm by 2.0 cm block with a small hole in the middle. A temperature gradient was applied through copper plates on opposite sides. Stehle [40] provides interface velocities of the bubble, which were used to validate this portion of the model. The problem parameters can be seen in Figure 5.1. Note that both this work and Kaempfer and Plapp's [23] work simulate this problem in 5mm x 5mm domain, not the whole 2cm x 2cm block of ice used in the experiment. Table 5.1 shows the resulting migration velocities with comparison values. This simulation was able to produced migration velocities on the same order of magnitude as Stehle's [40] experiment. Kaempfer and Plapp [23] had similar results and explained that better agreement is hardly expected considering that Stehle's [40] experiment developed frost in the hole and Equations 2.1, 2.8, and 2.9 only account for vapor diffusion through sublimation, not frost growth. Stehle's [40] result also showed that the hole's shape was distorted unevenly, losing an aspect ratio of unity.

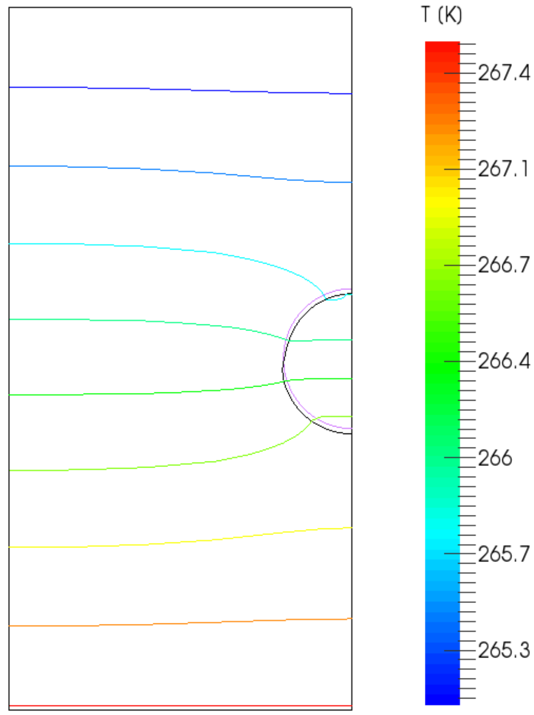


Figure 5.1: Stehle's [40] experiment replicated numerically showing temperature contours. The purple and black circles indicate the location of the bubble (defined as $\phi = 0$) before and after 4 hours of exposure to a 543 K/M temperature gradient, respectively.

Table 5.1: Air bubble migration velocity through ice during a 543K/M temperature gradient

Author	**Approximate Velocity (m/s)**	**Method**
Stehle [40]	5x10^{-9}	Experiment
Kaempfer and Plapp [23]	4x10^{-9}	Finite Difference
Present work	6x10^{-9}	Finite Element

This outcome is not documented well and is rather qualitatively pointed out as a unexpected result. Thus, no quantitative comparisons can be made to the shape of the hole. Additionally, the result shown in Figure 5.1 and in Kaempfer and Plapp's [23] work did not replicate this distortion. The results produced are not completely uniform. It is not completely uniform because when observing Figure 5.1 closely, there exists some slight anisotropic evolution which is introduced in the present work. This isotropic metamorphism, in part, is assumed through the phase-field time relaxation τ and interface kinetic coefficient constants λ. As stated before, τ and λ are both functions of capillary length and the interface kinetic coefficient, which are both functions of temperature. The higher velocity produced and the slight anisotropic growth in this work in part due to a slight difference in the definition of the aforementioned coefficients between the present work and that which was presented in Kaempfer and Plapp [23]. They recommended using a constant reference temperature to evaluate these terms due to the minimal impact a variable temperature would have on the solution. In the present research, τ and λ were kept as functions of temperature. The input files for the validation case can be seen in Appendix C.

The last demonstration of the diffusion processes is the capacity to model real snow. A 400 K/M temperature gradient is imposed for 6.3 hours using a μ-ct scan as the initial condition. The migration of the micro-structure is significant but is best demonstrated by Figure 5.2. In this simulation, the horizontal walls have specified temperatures to enforce the temperature gradients. The temperature on the walls is specified using a natural boundary condition of zero. And finally, the phase field is periodic in the vertical direction.

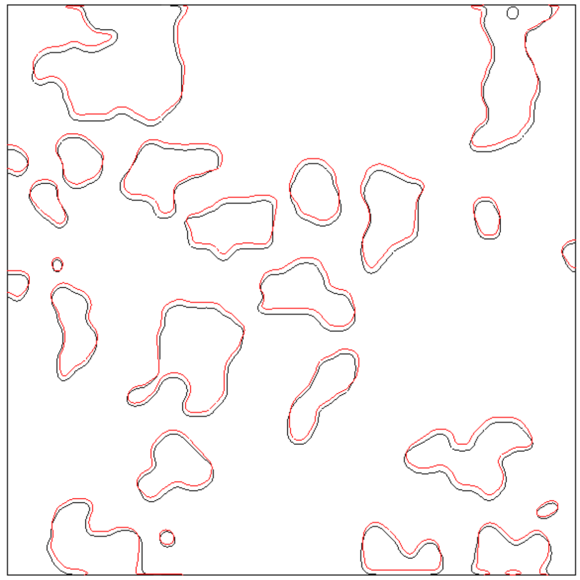


Figure 5.2: Simulation of real snow undergoing metamorphosis by diffusion only. The migration is induced by 6.3 hours of exposure to a 400 K/M temperature gradient. Black represents the original structure and red is the final location of the boundaries.

5.2 Passive Natural Convection in Snow Micro-structure

In order to investigate the relevance of natural convection, the phenomena is simulated with realistic conditions to examine the problem prior to coupling in phase change effects. Simulated here is a small section of snow micro-structure in the snowpack that surrounded by other snow. A temperature gradient of $500K/M$ is applied in the vertical direction, with the hottest on the bottom. The boundary conditions are periodic in the horizontal direction for the velocity and free slip on the top and bottom. The phase is periodic in the vertical direction, and temperature is periodic in the horizontal direction. The entire simulation is done on a slope of 30 degrees. To produce the following results, the momentum equations coupled with the phase field and temperature equations are solved. This represents buoyancy driven flow in the snow pack but is only acting passively. Here passively is defined

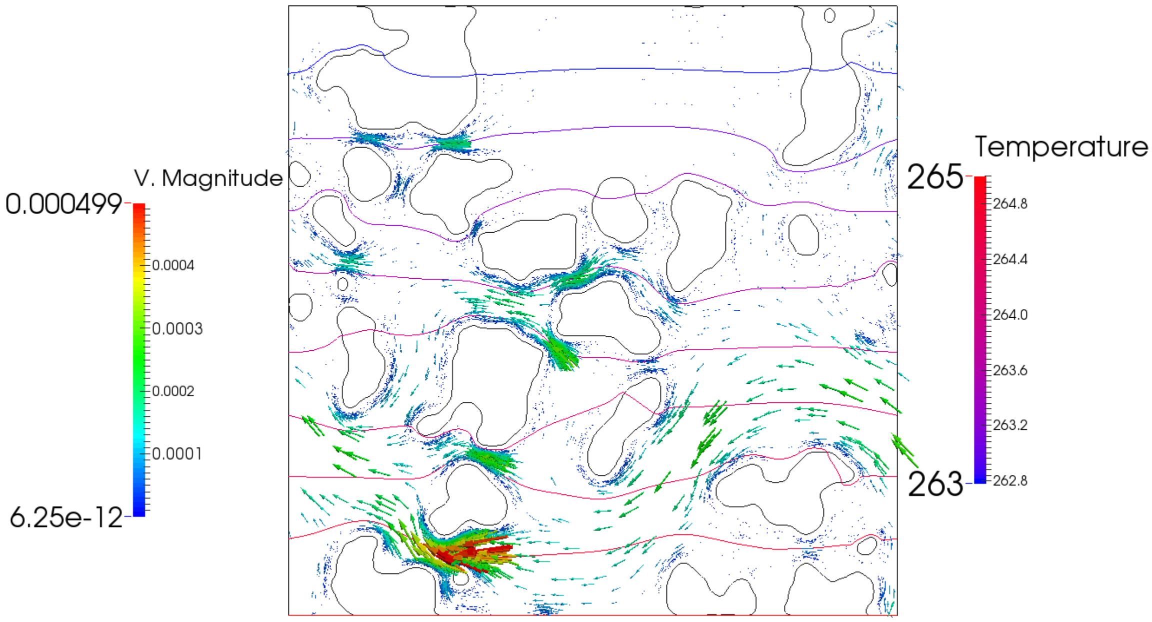


Figure 5.3: Simulation of natural convection developed by a $500K/M$ temperature gradient in real snow micro-structure on a 30 degree slope. The vectors indicate flow direction and are colored by velocity magnitude. The contours represent the temperature in which red represents the warmer temperature.

as not affecting the interface and thus the equations solved are not inclusive of the vapor concentration field. Figure 5.3 represents actual natural convection that would develop in snow on a slope of 30 degrees and a temperature gradient of $500K/M$. Another important feature is that several of the temperature contours follow the path of the flow, which means that the temperature field is noticeably being advected. While the temperature field is not the vapor concentration field, they are coupled and both transport equations are coupled to the velocity field. An analogy can then be drawn between the two variables, which indicates that the vapor field would also be affected and furthermore alter the rate and direction of the metamorphosis. This demonstrates that convection can happen in snow on a micro-structural level, further suggesting that it is inappropriately ignored in many models.

5.3 Considerations for a Fully Coupled Problem

Attempts were made at solving the fully coupled problem, but these simulations did not converge. This suggests that a numerical instability developed when all the equations were coupled. Considering the fact that each equation was simulated to some capacity, there is only one piece of the proposed model that is not represented here by results. This is the advection term in the vapor transport equation. To explore this, it was attempted to solve the equation in the natural convection in square enclosure problem, where it was only coupled with the velocity field with the intention to advect the scalar field X. As was mentioned earlier, the Galerkin formulation for the FEM is known to suffer from numerical instabilities when simulating convective dominant problems. This fact leads to a comparison between the two scalar fields where one (temperature) was capable of being advected and the other (vapor concentration field) has convergence issues. The difference here is in the material properties. Recall the vapor concentration equation's diffusion coefficient is one sided, meaning that it only diffuses in the vapor phase. The advection term, however, was not one sided. Even though the flow is theoretically supposed to be zero in the solid phase, a residual of any size other than zero will result in some small amount of flow. This would make the ratio of advection to diffusion problematic. This too was explored by making the advection term also one sided as shown in Equation 5.2.

$$\Big(\frac{1-\phi}{2}\Big)\vec{V}\cdot\nabla X \tag{5.1}$$

Unfortunately, the result was the same and convergence was never achieved. A final and unexplored theory is that the use of the conservative form of the advection terms might mitigate the convergence issue. Traditionally, fields that are conserved are advected using the conservative form, which in this case would be defined as

$$\nabla \cdot \big(\vec{V}X\big). \tag{5.2}$$

While this is a hypothesis, a thorough investigation is needed to identify the source of the convergence issue for the fully coupled problem.

CHAPTER 6

SUMMARY

Snow micro-structure constantly evolves under environmental conditions. Currently the snow science community lacks a metamorphism model capable of modeling natural convection on the micro-structural scale despite some observations made that suggest its presence. This thesis research initiates a work to address this need. A model was presented by adapting the model presented by Kaempfer and Plapp [23], which originally neglected natural convection. In an attempt to validate the fluid flow physics that were added to the model, three test cases were investigated. It was demonstrated that the current formulation of the FEM was capable of simulating lid driven cavity, flow over cylinder, and Natural convection in a closed cavity cases, up to certain limits of the advection effects. In all three cases, the numerical formulation was shown to accurately replicate lower Reynolds and Rayleigh number flows. At higher Reynolds flow, the Galerkin formulation of the FEM is well known to have instabilities in convection dominated problems, which was observed by simulating the lid driven cavity at Re=1000.

The three cases were then simulated again but the solid walls were modeled using a phase-field approach. In each simulation, the phase field equation is used to represent the geometry that defines the boundaries of the original computational domain. The momentum equations are coupled with the phase field equation by adding a Darcy-like term that is a function of W,ϕ, μ, aptly named the interfacial stress term. This term is formulated to attain zero velocity in the solid phase. In doing so, complex geometry was simulated with limited accuracy, demonstrating some capacity for the phase field equation to be used as an immersed boundary technique in low Reynolds number flows. Unfortunately, the formulation completely failed to replicate the vertically heated cavity case. While this may be seen as a loss, it has raised some questions. The first question is how should the velocity be normalized in a multi-phase domain with disparities between thermal properties? The second question is what role does having thermal properties that are greater at the walls affect the problem? At the end of the problem, no conclusions could be made without knowing the answers to these questions.

Kaempfer and Plapp's [23] work was then replicated and validated using the same experiment they used. Their diffusion based model with temporal scaling was simulated using MOOSE and show reasonable accuracy for simulating migrating vapor/ice interfaces. The model is then used to simulate the metamorphism of real snow through the use of μ-ct images of snow micro-structure. While under a 400 K/M temperature gradient, the pores of the snow micro-structure migrate toward the warmer side of the snow via sublimation.

The last demonstration showed passive natural convection simulation through snow micro-structure on slope. The simulation parameters represented real conditions that are observed in snow. This problem only includes solving the Navier-Stokes equations coupled with the phase field and heat transport equations. The results shown indicate advection is occurring, which in turn would affect the metamorphic processes. Natural convection in the snowpack should be further studied to arrive at firm conclusions.

6.1 Future Work

While the research here represents a starting point for investigating the role of natural convection in snow metamorphosis, future work is needed. First, the issues demonstrated in the vertically heated cavity need to be resolved prior to moving forward. This is necessary to validate the equations proposed here for the use of modeling natural convection in the snow pack.

Second, a numerical stability limit is potentially being violated with the inclusion of the advected term in the vapor transport equation. Convection dominated problems in the FEM community have been solved with a wide variety of stabilization techniques. A recommended method would be some variant of the Streamlined Upstream Petrov-Galerkin (SUPG) formulation. This stabilized technique essentially weights the functions in the direction of flow and has been successful in simulating fluid flow problems.

A third recommendation for future research surrounding this problem calls for an experiment to be able to replicate numerically. As it is now, there is not an experiment that physically matches the problem surrounding convection in the snow context. The problem could be an ice cylinder in a humid cross flow of air. Under the right conditions the ice would deform according to the available to water vapor. This could be monitored and measurements of the aspect ratio would provide excellent data with which to compare. The cylinder may be subject to frost like in Stehle's [40] bubble experiment and would render the results flawed. Thus, careful considerations would have to be made to avoid this if the experiment is undertaken.

Finally, the current formulation should be used to draw conclusions about the role of natural convection in the snowpack. This should be approached in three ways. First, the research should address how snow metamorphosis is affected by the inclusion of the fluid flow physics. The second is geometry based. It is well known that layers of varying density are present in the snowpack at any time. Thus, natural convection should be examined in multiple types of snow using μ-ct scans. And thirdly, what effect does the slope have on the metamorphosis within the context of natural convection? Specifically, the slopes should encompass those that are associated with high avalanche frequency. The direction of metamorphosis should be examined and may reveal if the natural convection is a component of the formation of avalanche conditions.

REFERENCES

[1] Afghanistan avalanche kill 150 people, February 2015. [Online; posted 26-February-2015].

[2] E. Akitaya. Some experiments on the growth of depth hoar. *Physics of Snow and Ice: proceeding*, 1(2):713–723, 1967.

[3] M. Albert and J. Hardy. Ventilation experiments in seasonal snow cover. *IAHS Publications-Series of Proceedings and Reports-Intern Assoc Hydrological Sciences*, 228:41–50, 1995.

[4] M. Albert and W. McGilvary. Thermal effects due to air flow and vapor transport in dry snow. *J. Glaciol*, 38(129):273–281, 1992.

[5] Colorado Avalanche Association. Statistics and reporting, 2014.

[6] P. Bartelt and M. Lehning. A physical snowpack model for the swiss avalanche warning: Part i: numerical model. *Cold Regions Science and Technology*, 35(3):123–145, 2002.

[7] C. Beckermann, H. Diepers, I. Steinbach, A. Karma, and X. Tong. Modeling melt convection in phase-field simulations of solidification. *Journal of Computational Physics*, 154(2):468–496, 1999.

[8] K. Birkeland. Terminology and predominant processes associated with the formation of weak layers of near-surface faceted crystals in the mountain snowpack. *Arctic and Alpine Research*, pages 193–199, 1998.

[9] J. Boyd, P. Haegeli, R. Abu-Laban, M. Shuster, and J. Butt. Patterns of death among avalanche fatalities: a 21-year review. *Canadian Medical Association Journal*, 180(5):507–512, 2009.

[10] C. Brehm and H. Fasel. A non-staggered immersed interface method for solving the incompressible navier-stokes equations. In *40th Fluid Dynamics Conference and Exhibit, AIAA*, volume 4433, page 2010, 2010.

[11] N. Calonne, F. Flin, S. Morin, B. Lesaffre, S. du Roscoat, and C. Geindreau. Numerical and experimental investigations of the effective thermal conductivity of snow. *Geophysical Research Letters*, 38(23):n/a–n/a, 2011.

[12] S. Colbeck. A review of the metamorphism and classification of seasonal snow cover crystals. *IAHS Publication*, 162:3–24, 1987.

[13] S. Colbeck. *Dynamics of snow and ice masses*. Elsevier, 2012.

[14] M. Coutanceau and R. Bouard. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. part 1. steady flow. *Journal of Fluid Mechanics*, 79(02):231–256, 1977.

[15] P. De Palma, M. De Tullio, G. Pascazio, and M. Napolitano. An immersed-boundary method for compressible viscous flows. *Computers & fluids*, 35(7):693–702, 2006.

[16] G. De Vahl Davis. Natural convection of air in a square cavity: a bench mark numerical solution. *International Journal for Numerical Methods in Fluids*, 3(3):249–264, 1983.

[17] S. Dennis and G. Chang. Numerical solutions for steady flow past a circular cylinder at reynolds numbers up to 100. *Journal of Fluid Mechanics*, 42(03):471–489, 1970.

[18] J. Fish and T. Belytschko. *A First Course in Finite Elements*. John Wiley and Sons, 1 edition, 2007.

[19] F. Flin, J. Brzoska, B. Lesaffre, C. Coléou, and R. Pieritz. Full three-dimensional modelling of curvature-dependent snow metamorphism: first results and comparison with experimental tomographic data. *Journal of Physics D: Applied Physics*, 36(10A):A49, 2003.

[20] P. Föhn and W. Ammann. Snow avalanches. In *SUB-FORUM ON SCIENCE AND TECHNOLOGY IN SUPPORT OF NATURAL DISASTER REDUCTION*, page 111, 1999.

[21] B. Fornberg. A numerical study of steady viscous flow past a circular cylinder. *Journal of Fluid Mechanics*, 98(04):819–855, 1980.

[22] U. Ghia, K. Ghia, and C. Shin. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of computational physics*, 48(3):387–411, 1982.

[23] T Kaempfer and M. Plapp. Phase-field modeling of dry snow metamorphism. *Phys. Rev. E*, 79:031502, Mar 2009.

[24] N. Klever. Air and water-vapour convection in snow. *Ann. Glaciol*, 6:39–42, 1985.

[25] D. Knoll and D. Keyes. Jacobian-free newton–krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

[26] Idaho National Lab. Moose framework, 2015.

[27] M. Linnick and H. Fasel. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics*, 204(1):157–192, 2005.

[28] M. Manzari. An explicit finite element algorithm for convection heat transfer problems. *International Journal of Numerical Methods for Heat & Fluid Flow*, 9(8):860–877, 1999.

[29] D. Mayne, A. Usmani, and M. Crapper. h-adaptive finite element solution of high rayleigh number thermally driven cavity problem. *International Journal of Numerical Methods for Heat & Fluid Flow*, 10(6):598–615, 2000.

[30] D. McClung. The elements of applied avalanche forecasting, part i: The human issues. *Natural Hazards*, 26(2):111–129, 2002.

[31] D. McClung and P. Schaerer. *Tha Avalanche Handbook*. The Mountaineers Books, 2006.

[32] D. Miller, E. Adams, and R. Brown. A microstructural approach to predict dry snow metamorphism in generalized thermal conditions. *Cold regions science and technology*, 37(3):213–226, 2003.

[33] B. Munson, D. Young, and T. Okiishi. *Fundamentals of fluid mechanics*. New York, 2006.

[34] D. Powers, S. Colbeck, and K. O'Neill. Experiments on thermal convection in snow. *Ann. Glaciol*, 6:43–47, 1985.

[35] J. Schweizer. Review of dry snow slab avalanche release. *Cold Regions Science and Technology*, 30(1-3):43–57, 1999.

[36] J. Schweizer, B. Jamieson, and M. Schneebeli. Snow avalanche formation. *Review of Geophysics*, 41(4):1016, 2003.

[37] J. Schweizer and M. Lütschg. Characteristics of human-triggered avalanches. *Cold Regions Science and Technology*, 33(2):147–162, 2001.

[38] Gopal Sharma. Hundreds of bodies may be buried in nepal avalanche, official says, May 2015. [Online; posted 7-May-2015].

[39] W. Shyy. *Computational modeling for fluid flow and interfacial transport.* Courier Corporation, 2014.

[40] N. Stehle. Migration of bubbles in ice under a temperature gradient. Technical report, DTIC Document, 1965.

[41] M. Sturm and J. Johnson. Natural convection in the subarctic snow cover. *Journal of Geophysical Research*, 96:11,657–11,671, July 1991.

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

[43] C. Wan, B. Patnaik, and G. Wei. A new benchmark quality solution for the buoyancy-driven cavity by discrete singular convolution. *Numerical Heat Transfer: Part B: Fundamentals*, 40(3):199–228, 2001.

APPENDIX A

WEAK FORMULATIONS/ RESIDUALS

A.1 Weak Formulation/ Residual for the Navier-Stokes Equations and the Conservation of Mass Equation

Weak Formulation of the Momentum Equation in the X Direction

$$\begin{aligned}\int_\Omega \psi\rho_a\frac{\partial u}{\partial t} + \int_\Omega \psi\rho_a\xi\vec{V}\cdot\nabla u d\Omega - \int_\Gamma \psi\xi(-pI + \mu\nabla u)\cdot\vec{n}d\Gamma ...\\ - \int_\Omega \frac{\partial\psi}{\partial x}\xi p d\Omega + \int_\Omega \nabla\psi\xi\cdot\mu\nabla u)d\Omega + \int_\Omega \psi\xi\mu h\frac{1+\phi}{2W^2}ud\Omega\end{aligned} \tag{A.1}$$

Weak Formulation of the Momentum Equation in the Y Direction

$$\begin{aligned}\int_\Omega \psi\rho_a\frac{\partial v}{\partial t} + \int_\Omega \psi\rho_a\xi\vec{V}\cdot\nabla v d\Omega - \int_\Gamma \psi\xi(-pI + \mu\nabla v)\cdot\vec{n}d\Gamma - \int_\Omega \frac{\partial\psi}{\partial y}\xi p d\Omega ...\\ + \int_\Omega \nabla\psi\xi\cdot\mu\nabla v)d\Omega - \int_\Omega \psi[1-\alpha(T-T_{ref})]g + \int_\Omega \psi\rho_a\xi\mu h\frac{1+\phi}{2W^2}vd\Omega\end{aligned} \tag{A.2}$$

Weak Formulation of the Momentum Equation in the Z Direction

$$\int_{\Omega}\psi\rho_a\frac{\partial w}{\partial t}+\int_{\Omega}\psi\rho_a\xi\vec{V}\cdot\nabla w d\Omega-\int_{\Gamma}\psi\xi(-pI+\mu\nabla w)\cdot\vec{n}d\Gamma...$$
$$-\int_{\Omega}\frac{\partial\psi}{\partial z}\xi p d\Omega+\int_{\Omega}\nabla\psi\xi\cdot\mu\nabla w)d\Omega+\int_{\Omega}\psi\xi\mu h\frac{1+\phi}{2W^2}w d\Omega \tag{A.3}$$

Weak Formulation of the Conservation of Mass Equation

$$\int_{\Omega}\psi(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z})d\Omega \tag{A.4}$$

A.2 Weak Formulation of the Heat Transport Equation

$$\int_{\Omega}\psi C(\phi)\frac{\partial T}{\partial t}d\Omega+\int_{\Omega}\psi\xi C(\phi)(\vec{V}\cdot\nabla T)d\Omega-\int_{\Gamma}\psi\xi K(\phi)\nabla T\cdot\vec{n}d\Gamma...$$
$$+\int_{\Omega}(\nabla\psi\xi\cdot K(\phi)\nabla T)d\Omega-\int_{\Omega}\psi\frac{\xi L_{sg}}{2}\frac{\partial\phi}{\partial t}d\Omega \tag{A.5}$$

A.3 Weak Formulation of the Vapor Potential Transport Equation

$$\int_{\Omega}\psi\frac{\partial X}{\partial t}d\Omega+\int_{\Omega}\psi\xi(\vec{V}\cdot\nabla X)d\Omega-\int_{\Gamma}\psi\xi D(\phi)\nabla X\cdot\vec{n}d\Gamma...$$
$$+\int_{\Omega}(\nabla\psi\xi\cdot D(\phi)\nabla X)d\Omega+\int_{\Omega}\psi\frac{\xi}{2}\frac{\partial\phi}{\partial t}d\Omega \tag{A.6}$$

A.4 Weak Formulation of the Phase Field Equation

$$\int_\Omega \psi\tau\frac{\partial\phi}{\partial t}d\Omega - \int_\Gamma \psi W^2\nabla\phi\cdot\vec{n}d\Gamma + \int_\Omega(\nabla\psi\cdot W^2\nabla\phi)d\Omega...$$
$$-\int_\Omega \psi(\phi-\phi^3)d\Omega - \int_\Omega \psi\lambda\;[X-X_{eq}](1-\phi^2)^2 d\Omega \tag{A.7}$$

APPENDIX B

JACOBIAN FORMULATION

B.1 Jacobian Matrix

The Jacobian provided to the PJFNK solver is defined such that

$$\mathbf{J}_{ij} = \begin{bmatrix} \frac{\partial R_u}{\partial u} & \frac{\partial R_u}{\partial v} & \frac{\partial R_u}{\partial w} & \frac{\partial R_u}{\partial p} & \frac{\partial R_u}{\partial \phi} & \frac{\partial R_u}{\partial X} & \frac{\partial R_u}{\partial T} \\ \frac{\partial R_v}{\partial u} & \frac{\partial R_v}{\partial v} & \frac{\partial R_v}{\partial w} & \frac{\partial R_v}{\partial p} & \frac{\partial R_v}{\partial \phi} & \frac{\partial R_v}{\partial X} & \frac{\partial R_v}{\partial T} \\ \frac{\partial R_w}{\partial u} & \frac{\partial R_w}{\partial v} & \frac{\partial R_w}{\partial w} & \frac{\partial R_w}{\partial p} & \frac{\partial R_w}{\partial \phi} & \frac{\partial R_w}{\partial X} & \frac{\partial R_w}{\partial T} \\ \frac{\partial R_p}{\partial u} & \frac{\partial R_p}{\partial v} & \frac{\partial R_p}{\partial w} & \frac{\partial R_p}{\partial p} & \frac{\partial R_p}{\partial \phi} & \frac{\partial R_p}{\partial X} & \frac{\partial R_p}{\partial T} \\ \frac{\partial R_\phi}{\partial u} & \frac{\partial R_\phi}{\partial v} & \frac{\partial R_\phi}{\partial w} & \frac{\partial R_\phi}{\partial p} & \frac{\partial R_\phi}{\partial \phi} & \frac{\partial R_\phi}{\partial X} & \frac{\partial R_\phi}{\partial T} \\ \frac{\partial R_X}{\partial u} & \frac{\partial R_X}{\partial v} & \frac{\partial R_X}{\partial w} & \frac{\partial R_X}{\partial p} & \frac{\partial R_X}{\partial \phi} & \frac{\partial R_X}{\partial X} & \frac{\partial R_X}{\partial T} \\ \frac{\partial R_T}{\partial u} & \frac{\partial R_T}{\partial v} & \frac{\partial R_T}{\partial w} & \frac{\partial R_T}{\partial p} & \frac{\partial R_T}{\partial \phi} & \frac{\partial R_T}{\partial X} & \frac{\partial R_T}{\partial T} \end{bmatrix}$$

where it is assumed that

$$R_u = f(\phi, u, v, w, p), \tag{B.1a}$$

$$R_v = f(\phi, u, v, w, p, T), \tag{B.1b}$$

$$R_w = f(\phi, u, v, w, p), \tag{B.1c}$$

$$R_p = f(u, v, w), \tag{B.1d}$$

$$R_\phi = f(\phi, X), \tag{B.1e}$$

$$R_T = f(\phi, u, v, w, T), \tag{B.1f}$$

$$R_X = f(\phi, u, v, w, X), \tag{B.1g}$$

which leads to the Jacobian being redefined as

$$\mathbf{J}_{ij} = \begin{bmatrix} \frac{\partial R_u}{\partial u} & \frac{\partial R_u}{\partial v} & \frac{\partial R_u}{\partial w} & \frac{\partial R_u}{\partial p} & \frac{\partial R_u}{\partial \phi} & 0 & 0 \\ \frac{\partial R_v}{\partial u} & \frac{\partial R_v}{\partial v} & \frac{\partial R_v}{\partial w} & \frac{\partial R_v}{\partial p} & \frac{\partial R_v}{\partial \phi} & 0 & \frac{\partial R_v}{\partial T} \\ \frac{\partial R_w}{\partial u} & \frac{\partial R_w}{\partial v} & \frac{\partial R_w}{\partial w} & \frac{\partial R_w}{\partial p} & \frac{\partial R_w}{\partial \phi} & \frac{\partial R_w}{\partial X} & 0 \\ \frac{\partial R_p}{\partial u} & \frac{\partial R_p}{\partial v} & \frac{\partial R_p}{\partial w} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\partial R_\phi}{\partial \phi} & \frac{\partial R_\phi}{\partial X} & 0 \\ \frac{\partial R_X}{\partial u} & \frac{\partial R_X}{\partial v} & \frac{\partial R_X}{\partial w} & 0 & \frac{\partial R_X}{\partial \phi} & \frac{\partial R_X}{\partial X} & 0 \\ \frac{\partial R_T}{\partial u} & \frac{\partial R_T}{\partial v} & \frac{\partial R_T}{\partial w} & 0 & \frac{\partial R_T}{\partial \phi} & 0 & \frac{\partial R_T}{\partial T} \end{bmatrix}$$

At this point the formulation of non-zero Jacobian values are presented by section according to their respective equations.

B.1.1 Jacobians for the X Direction of the Momentum Equation

$$\frac{\partial R_u}{\partial \phi} = \int_\Omega \psi \xi \mu h \frac{N}{2W^2} u d\Omega \tag{B.2}$$

$$\frac{\partial R_u}{\partial u} = \int_\Omega \psi\rho_a \frac{\partial u_k}{\partial t} N + \int_\Omega \psi\rho_a \xi\big(\vec{V}\cdot\nabla N + N\frac{\partial u}{\partial x}\big) d\Omega ... \\ + \int_\Omega \nabla\psi\xi\cdot\mu\nabla N) d\Omega + \int_\Omega \psi\xi\mu h \frac{1+\phi}{2W^2} N d\Omega \tag{B.3}$$

$$\frac{\partial R_u}{\partial v} = \int_\Omega \psi\rho_a \xi\big(N\frac{\partial u}{\partial y}\big) d\Omega \tag{B.4}$$

$$\frac{\partial R_u}{\partial w} = \int_\Omega \psi\rho_a \xi\big(N\frac{\partial u}{\partial z}\big) d\Omega \tag{B.5}$$

$$\frac{\partial R_u}{\partial p} = -\int_\Omega \frac{\partial \psi}{\partial x} \xi N d\Omega \tag{B.6}$$

B.1.2 Jacobians for the Y Direction of the Momentum Equation

$$\frac{\partial R_v}{\partial \phi} = \int_\Omega \psi\xi\mu h \frac{N}{2W^2} v d\Omega \tag{B.7}$$

$$\frac{\partial R_v}{\partial v} = \int_\Omega \psi\rho_a \frac{\partial v_k}{\partial t} N + \int_\Omega \psi\rho_a \xi\big(\vec{V}\cdot\nabla N + N\frac{\partial v}{\partial y}\big) d\Omega ... \\ + \int_\Omega \nabla\psi\xi\cdot\mu\nabla N) d\Omega + \int_\Omega \psi\xi\mu h \frac{1+\phi}{2W^2} N d\Omega \tag{B.8}$$

$$\frac{\partial R_v}{\partial u} = \int_\Omega \psi\rho_a \xi\big(N\frac{\partial v}{\partial x}\big) d\Omega \tag{B.9}$$

$$\frac{\partial R_v}{\partial w} = \int_\Omega \psi\rho_a \xi\big(N\frac{\partial v}{\partial z}\big) d\Omega \tag{B.10}$$

$$\frac{\partial R_v}{\partial p} = -\int_{\Omega} \frac{\partial \psi}{\partial y} \xi N d\Omega \tag{B.11}$$

$$\frac{\partial R_v}{\partial T} = \int_{\Omega} \psi \alpha N g \tag{B.12}$$

B.1.3 Jacobians for the Z Direction of the Momentum Equation

$$\frac{\partial R_w}{\partial \phi} = \int_{\Omega} \psi \xi \mu h \frac{N}{2W^2} w d\Omega \tag{B.13}$$

$$\begin{aligned}\frac{\partial R_w}{\partial w} = \int_{\Omega} \psi \rho_a \frac{\partial w_k}{\partial t} N + \int_{\Omega} \psi \rho_a \xi \big(\vec{V} \cdot \nabla N + N \frac{\partial w}{\partial z}\big) d\Omega ... \\ + \int_{\Omega} \nabla \psi \xi \cdot \mu \nabla N) d\Omega + \int_{\Omega} \psi \xi \mu h \frac{1+\phi}{2W^2} N d\Omega\end{aligned} \tag{B.14}$$

$$\frac{\partial R_w}{\partial u} = \int_{\Omega} \psi \rho_a \xi \big(N \frac{\partial w}{\partial x}\big) d\Omega \tag{B.15}$$

$$\frac{\partial R_w}{\partial v} = \int_{\Omega} \psi \rho_a \xi \big(N \frac{\partial w}{\partial y}\big) d\Omega \tag{B.16}$$

$$\frac{\partial R_w}{\partial p} = -\int_{\Omega} \frac{\partial \psi}{\partial z} \xi N d\Omega \tag{B.17}$$

B.1.4 Jacobians for the Conservation of Mass

$$\frac{\partial R_p}{\partial u} = \int_{\Omega} \psi \big(\frac{\partial N}{\partial x}\big) d\Omega \tag{B.18}$$

$$\frac{\partial R_p}{\partial v} = \int_\Omega \psi (\frac{\partial N}{\partial y}) d\Omega \tag{B.19}$$

$$\frac{\partial R_p}{\partial w} = \int_\Omega \psi (\frac{\partial N}{\partial z}) d\Omega \tag{B.20}$$

B.1.5 Jacobians for the Phase Field Equation

$$\begin{aligned} \frac{\partial R_\phi}{\partial \phi} = \int_\Omega \psi \tau \frac{\partial \phi_k}{\partial t} N d\Omega + \int_\Omega (\nabla \psi \cdot W^2 \nabla N) d\Omega ... \\ - \int_\Omega \psi (1 - 3\phi^2) N d\Omega - \int_\Omega \psi \lambda \ [X - X_{eq}](4\phi^3 - 4) N d\Omega \end{aligned} \tag{B.21}$$

$$\frac{\partial R_\phi}{\partial X} = - \int_\Omega \psi \lambda \ N(1 - \phi^2)^2 d\Omega \tag{B.22}$$

B.1.6 Jacobians for the Vapor Potential Transport Equation

$$\frac{\partial R_X}{\partial \phi} = \int_\Omega \psi \frac{\xi}{2} \frac{\partial \phi_k}{\partial t} N d\Omega \tag{B.23}$$

$$\frac{\partial R_X}{\partial u} = \int_\Omega \psi \xi N \frac{\partial X}{\partial x} d\Omega \tag{B.24}$$

$$\frac{\partial R_X}{\partial v} = \int_\Omega \psi \xi N \frac{\partial X}{\partial y} d\Omega \tag{B.25}$$

$$\frac{\partial R_X}{\partial w} = \int_\Omega \psi \xi N \frac{\partial X}{\partial z} d\Omega \tag{B.26}$$

$$\frac{\partial R_X}{\partial X} = \int_\Omega \psi \frac{\partial X_k}{\partial t} N d\Omega + \int_\Omega \psi\xi(\vec{V}\cdot\nabla N)d\Omega + \int_\Omega (\nabla\psi\xi\cdot D(\phi)\nabla N)d\Omega \tag{B.27}$$

Note that though the diffusion coefficient, $(D(\phi)$, for the vapor potential transport equation is defined as a function of ϕ, the convenience of the PJFNK allows for a close approximation of the full Jacobian, thus it is assumed that $\frac{\partial}{\partial\phi}(D(\phi)\nabla X) \approx 0$.

B.1.7 Jacobians for the Heat Transport Equation

$$\frac{\partial R_T}{\partial \phi} = \int_\Omega \psi L_{sg} \frac{\xi}{2} \frac{\partial \phi_k}{\partial t} N d\Omega \tag{B.28}$$

$$\frac{\partial R_T}{\partial u} = \int_\Omega \psi\xi N \frac{\partial T}{\partial x} d\Omega \tag{B.29}$$

$$\frac{\partial R_T}{\partial v} = \int_\Omega \psi\xi N \frac{\partial T}{\partial y} d\Omega \tag{B.30}$$

$$\frac{\partial R_T}{\partial w} = \int_\Omega \psi\xi N \frac{\partial T}{\partial z} d\Omega \tag{B.31}$$

$$\frac{\partial R_T}{\partial T} = \int_\Omega \psi C(\phi) \frac{\partial T_k}{\partial t} N d\Omega + \int_\Omega \psi C(\phi)\xi(\vec{V}\cdot\nabla N)d\Omega + \int_\Omega (\nabla\psi\xi\cdot K(\phi)\nabla N)d\Omega \tag{B.32}$$

Note that though the heat capacity, $C(\phi)$, and thermal conductivity, $K(\phi)$, coefficient for the heat transport equation is defined as a function of ϕ, the convenience of the PJFNK method allows for a close approximation of the full Jacobian, thus it is assumed that $\frac{\partial}{\partial\phi}(C(\phi)\vec{V}\nabla T)$ and $\frac{\partial}{\partial\phi}(K(\phi)\nabla T) \approx 0$.

APPENDIX C

INPUT FILES

C.1 Lid Driven Cavity with Solid Walls Initialization

```
[Mesh]
  type = GeneratedMesh
  dim = 2
  nx = 10
  ny = 10
  xmin = -1e-4
  ymin = -1e-4
  xmax = .0051
  ymax = .0050
  elem_type = QUAD9
[]

[Variables]
  [./phi]
  [../]
[]

[AuxVariables]
  [./u]
  [../]
  [./phi_aux]
  [../]
[]

[Kernels]
active = 'phase_time phase_diffusion phase_double_well'
  [./phase_time]
    type = PikaTimeDerivative
    variable = phi
    property = relaxation_time
  [../]
  [./phase_diffusion]
    type = PikaDiffusion
    variable = phi
    property = interface_thickness_squared
  [../]
  [./phase_double_well]
    type = DoubleWellPotential
    variable = phi
    mob_name = mobility
  [../]
[]
```

```

[AuxKernels]
  [./phi_aux]
    type = PikaPhaseInitializeAux
    variable = phi_aux
    phase = phi
  [../]
[]

[BCs]
  [./solid]
    type = DirichletBC
    variable = phi
    boundary = 'left bottom right'
    value = 1
  [../]
[]

[Executioner]
  # Preconditioned JFNK (default)
  type = Transient
  dt = 1
  end_time = 1000
  nl_max_its = 20
  solve_type = PJFNK
  petsc_options_iname = '-ksp_gmres_restart -pc_type -pc_hypre_type'
  petsc_options_value = '50 hypre boomeramg'
  nl_rel_tol = 1e-07
  nl_abs_tol = 1e-12
  l_tol = 1e-4
  l_abs_step_tol = 1e-13
  [./TimeStepper]
    type = SolutionTimeAdaptiveDT
    dt = 1
    percent_change = 10
 [../]
[]
[Adaptivity]
  max_h_level = 8
  initial_steps = 8
  marker = phi_marker
  initial_marker = phi_marker
  [./Indicators]
    [./phi_grad_indicator]
      type = GradientJumpIndicator
      variable = phi
    [../]
  [../]
  [./Markers]
    [./phi_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-7
      indicator = phi_grad_indicator
      refine = 1e-5
    [../]
  [../]
[]

[Outputs]
  print_linear_residuals = true
  print_perf_log = true
  [./out]
```

```
      output_final = true
      type = Exodus
      file_base = phi_initial_out
      output_final = true
      output_initial = true
    [../]
[]

[ICs]
    [./phi_box_IC]
      y2 = 0.0051
      y1 = 0
      inside = -1
      x2 = 0.005
      outside = 1
      variable = phi
      x1 = 0
      type = BoundingBoxIC
    [../]
[]

[PikaMaterials]
   temperature = 263.15
   interface_thickness = 1e-5
   phase = phi
   temporal_scaling = 1
[]
```

C.2 Lid Driven Cavity with Solid Walls, Re=400

```
[Mesh]
  type = GeneratedMesh
  dim = 2
  nx = 50
  ny = 50
  xmin = -1e-4
  ymin = -1e-4
  xmax = .0051
  ymax = .0050
  elem_type = QUAD9
[]
[MeshModifiers]
  [./pin]
    type = AddExtraNodeset
    new_boundary = 99
    coord = '0 0'
    tolerance = 1e-04
  [../]
[]
[Variables]
  [./v_x]
    order = SECOND
  [../]
  [./v_y]
    order = SECOND
  [../]
  [./p]
  [../]
  [./phi]
  [../]
[]
[Functions]
  [./phi_func]
    type = SolutionFunction
    from_variable = phi
    solution = uo_initial
  [../]
[]
[Kernels]
  [./x_momentum]
    type = PikaMomentum
    variable = v_x
    vel_y = v_y
    vel_x = v_x
    component = 0
    p = p
  [../]
  [./x_no_slip]
    type = PhaseNoSlipForcing
    variable = v_x
    phase = phi
    h = 1000

  [../]
  [./y_momentum]
    type = PikaMomentum
    variable = v_y
    vel_y = v_y
    vel_x = v_x
```

```
    component = 1
    p = p
  [../]
  [./y_no_slip]
    type = PhaseNoSlipForcing
    variable = v_y
    phase = phi
    h = 1000

  [../]
  [./mass_conservation]
    type = INSMass
    variable = p
    v = v_y
    u = v_x
    p = p
  [../]
  [./phase_time]
    type = PikaTimeDerivative
    variable = phi
    property = relaxation_time
    use_temporal_scaling = false
  [../]

  [./phase_diffusion]
    type = PikaDiffusion
    variable = phi
    property = interface_thickness_squared
    use_temporal_scaling = false
  [../]
  [./phase_double_well]
    type = DoubleWellPotential
    variable = phi
    mob_name = mobility
  [../]
[]
[BCs]
#Based on a RE=400 where L=0.005m
  [./lid]
    type = DirichletBC
    variable = v_x
    boundary = top
    value = 0.9539149888
  [../]
  [./y_no_slip_top]
    type = DirichletBC
    variable = v_y
    boundary = top
    value = 0.0
  [../]
  [./solid_phase_wall]
    type = DirichletBC
    variable = phi
    boundary = 'left right bottom'
    value = 1
  [../]
[]

[UserObjects]
  [./uo_initial]
    type = SolutionUserObject
    execute_on = initial
```

```
    mesh = phi_initial_out.e-s004
    timestep = 1
  [../]
[]

[VectorPostprocessors]
  [./horizontal]
    type = LineValueSampler
    variable = v_y
    num_points = 100
    end_point = '0.0051 0.0025 0'
    sort_by = x
    execute_on = timestep_end
    start_point = '-1e-4 0.0025 0'
  [../]
  [./vertical]
    type = LineValueSampler
    variable = v_x
    num_points = 100
    start_point = '0.0025 -1e-4 0'
    end_point = '0.0025 0.005 0'
    sort_by = y
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
  [../]
[]

[Executioner]
  type = Transient
  dt = 0.01
  end_time = 0.1
  solve_type = PJFNK
  petsc_options_iname = '-ksp_gmres_restart '
  petsc_options_value = '100 '
  l_max_its = 100
  nl_max_its = 150
  nl_rel_tol = 1e-08
  l_tol = 1e-08
  line_search = none

[]
[Adaptivity]
  max_h_level = 5
  initial_steps = 5
  steps = 0
  marker = phi_marker
  initial_marker = phi_marker
  [./Indicators]
    [./phi_grad_indicator]
      type = GradientJumpIndicator
      variable = phi
    [../]
  [../]
  [./Markers]
    [./phi_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-7
```

```
      indicator = phi_grad_indicator
      refine = 1e-5
    [../]
  [../]
[]
[Outputs]
  [./console]
    type = Console
    output_linear = true
    output_nonlinear = true
  [../]
  [./exodus]
    file_base = phase_LDC_h_100
    type = Exodus
    output_final = true
    output_initial = true
  [../]
  [./csv]
    file_base = phase_LDC_h_100
    type = CSV
  [../]
[]


[PikaMaterials]
  phase = phi
  temperature = 263.15
  interface_thickness = 1e-05
  temporal_scaling = 1
[]

[ICs]
  [./phase_ic]
    variable = phi
    type = FunctionIC
    function = phi_func
  [../]
[]
```

C.3 Flow over Cylinder Initialization

```
[Mesh]
  type = GeneratedMesh
  dim = 2
  nx = 400
  ny = 200
  xmin = 0
  xmax = 0.04
  ymin = 0
  ymax = 0.02
  elem_type = QUAD9
[]

[MeshModifiers]
  [./pin]
    type = AddExtraNodeset
    coord = '0.005 0.01'
    tolerance = 1e-4
    new_boundary = 99
  [../]
[]

[Variables]
  [./phi]
  [../]
[]

[Kernels]
  [./phi_time_derivative]
    type = PikaTimeDerivative
    variable = phi
    property = relaxation_time
  [../]
  [./phi_diffusion]
    type = PikaDiffusion
    variable = phi
    property = interface_thickness_squared
    temporal_scaling = false
  [../]
  [./phi_double_well]
    type = DoubleWellPotential
    variable = phi
    mob_name = mobility
  [../]
[]
[BCs]
  [./vapor_walls]
    type = DirichletBC
    value = -1
    variable = phi
    boundary = 'top left right bottom'
  [../]

  [./solid_pin]
    type = DirichletBC
    value = 1
    variable = phi
    boundary = 99
  [../]

```

```
[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
  [../]
[]

[Executioner]
  # Preconditioned JFNK (default)
  type = Transient
  dt = 100
  end_time = 1000
  nl_max_its = 20
  solve_type = PJFNK
  petsc_options_iname = '-ksp_gmres_restart -pc_type -pc_hypre_type'
  petsc_options_value = '50 hypre boomeramg'
  nl_rel_tol = 1e-07
  nl_abs_tol = 1e-12
  l_tol = 1e-4
  l_abs_step_tol = 1e-13
[]
[Adaptivity]
  max_h_level = 5
  initial_steps = 5
  steps = 4
  marker = phi_marker
  initial_marker = phi_marker
  [./Indicators]
    [./phi_grad_indicator]
      type = GradientJumpIndicator
      variable = phi
    [../]
  [../]
  [./Markers]
    [./phi_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-7
      indicator = phi_grad_indicator
      refine = 1e-5
    [../]
  [../]
[]

[Outputs]
  print_linear_residuals = true
  print_perf_log = true
  [./out]
    type = Exodus
    file_base = re_20_initial_out
    output_final = true
    output_initial = true
  [../]
[]
[PikaMaterials]
  phase = phi
  temperature = 263.15
  interface_thickness = 1e-05
  temporal_scaling = 1 # 1e-05
  gravity = '0 -9.81 0'
[]

[ICs]
```

```
active = 'phase_ic'
  [./phase_ic]
    y1 = 0.01
    variable = phi
    x1 = 0.005
    type = SmoothCircleIC
    int_width = 1e-5
    radius = 0.0005
    outvalue = -1
    invalue = 1
    3D_spheres = false
  [../]
[]
```

C.4 Flow over Cylinder , Re= 20

```
[Mesh]
  type = GeneratedMesh
  dim = 2
  nx = 400
  ny = 200
  xmin = 0
  xmax = 0.04
  ymin = 0
  ymax = 0.02
  elem_type = QUAD9
[]
[MeshModifiers]
  [./pin]
    type = AddExtraNodeset
    new_boundary = 99
    coord = '0.005 0.01'
    tolerance = 1e-04
  [../]
[]
[Variables]
  [./v_x]
    order = SECOND
  [../]
  [./v_y]
    order = SECOND
  [../]
  [./p]
  [../]
  [./phi]
  [../]
[]
[Functions]
#   [./phi_func]
#     type = SolutionFunction
#     from_variable = phi
#     solution = uo_initial
#   [../]
  [./phi_func]
    type = SolutionFunction
    from_variable = phi
    solution = uo_restart
  [../]
  [./p_func]
    type = SolutionFunction
    from_variable = p
    solution = uo_restart
  [../]
  [./v_x_func]
    type = SolutionFunction
    from_variable = v_x
    solution = uo_restart
  [../]
  [./v_y_func]
    type = SolutionFunction
    from_variable = v_y
    solution = uo_restart
  [../]
[]
[Kernels]
```

```
  [./x_momentum_time]
    type = PikaTimeDerivative
    variable = v_x
    coefficient = 1.341
    use_temporal_scaling = false
  [../]

  [./x_momentum]
    type = PikaMomentum
    variable = v_x
    vel_y = v_y
    vel_x = v_x
    component = 0
    p = p
  [../]
  [./x_no_slip]
    type = PhaseNoSlipForcing
    variable = v_x
    phase = phi
    h = 100
  [../]

  [./y_momentum_time]
    type = PikaTimeDerivative
    variable = v_y
    coefficient = 1.341
    use_temporal_scaling = false
  [../]

  [./y_momentum]
    type = PikaMomentum
    variable = v_y
    vel_y = v_y
    vel_x = v_x
    component = 1
    p = p
  [../]
  [./y_no_slip]
    type = PhaseNoSlipForcing
    variable = v_y
    phase = phi
    h = 100
  [../]
  [./mass_conservation]
    type = INSMass
    variable = p
    v = v_y
    u = v_x
    p = p
  [../]
  [./phase_time]
    type = PikaTimeDerivative
    variable = phi
    property = relaxation_time
    use_temporal_scaling = false
  [../]

  [./phase_diffusion]
    type = PikaDiffusion
    variable = phi
    property = interface_thickness_squared
    use_temporal_scaling = false
```

```
  [../]
  [./phase_double_well]
    type = DoubleWellPotential
    variable = phi
    mob_name = mobility
  [../]
[]
[BCs]
  [./inlet]
    type = DirichletBC
    variable = v_x
    boundary = left
    value = 0.2384787472
  [../]

  [./y_no_slip_top]
    type = DirichletBC
    variable = v_y
    boundary = 'top bottom'
    value = 0.0
  [../]
 [./pressure_out]
   type = DirichletBC
   variable = p
   boundary = right
   value = 0
 [../]
[]

[UserObjects]
#  [./uo_initial]
#    type = SolutionUserObject
#    execute_on = initial
#    mesh = re_20_initial_out.e-s010
#    timestep = 1
#  [../]
  [./uo_restart]
    type = SolutionUserObject
    execute_on = initial
    mesh = ../1e6_001/re_20_out.e-s004
    timestep = 1
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
  [../]
[]

[Executioner]
  type = Transient
  dt = 0.01
  start_time = 0.045
  end_time = 0.12
  solve_type = PJFNK
  petsc_options_iname = '-ksp_gmres_restart '
  petsc_options_value = '100 '
  l_max_its = 100
  nl_max_its = 150
  nl_rel_tol = 1e-08
```

```
  l_tol = 1e-08
  line_search = none
  scheme = 'crank-nicolson'


[]
[Adaptivity]
  max_h_level = 7
  initial_steps =7
  steps = 0
  marker = combo_marker
  initial_marker = phi_marker
  [./Indicators]
    [./phi_grad_indicator]
      type = GradientJumpIndicator
      variable = phi
    [../]
    [./v_x_grad_indicator]
      type = GradientJumpIndicator
      variable = v_x
    [../]
    [./v_y_grad_indicator]
      type = GradientJumpIndicator
      variable = v_y
    [../]

  [../]
  [./Markers]
    [./phi_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-7
      indicator = phi_grad_indicator
      refine = 1e-5
    [../]
    [./v_x_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-7
      indicator = v_x_grad_indicator
      refine = 1e-5
    [../]
    [./v_y_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-7
      indicator = v_y_grad_indicator
      refine = 1e-5
    [../]
    [./combo_marker]
      type = ComboMarker
      markers = ' phi_marker v_x_marker v_y_marker'
    [../]
  [../]
[]
[Outputs]
  [./exodus]
    file_base = re_20_out
    type = Exodus
    output_final = true
    output_initial = true
    interval = 1
  [../]
[]
```

```

[PikaMaterials]
  phase = phi
  temperature = 263.15
  interface_thickness = 1e-6
  temporal_scaling = 1
[]

[ICs]
  [./phase_ic]
    variable = phi
    type = FunctionIC
    function = phi_func
  [../]
  [./p_ic]
    variable = p
    type = FunctionIC
    function = p_func
  [../]
  [./v_x_ic]
    variable = v_x
    type = FunctionIC
    function = v_x_func
  [../]
  [./v_y_ic]
    variable = v_y
    type = FunctionIC
    function = v_y_func
  [../]
[]
```

C.5 Natural Convection in a Square Ice Enclosure Initialization

```
[Mesh]
  type = GeneratedMesh
  dim = 2
  nx = 50
  ny = 50
  xmin = -1e-4
  ymin = -1e-4
  xmax = .0051
  ymax = .0051
  elem_type = QUAD9
[]

[Variables]
  [./phi]
  [../]
[]

[AuxVariables]
  [./u]
  [../]
  [./phi_aux]
  [../]
[]

[Kernels]
  [./phase_time]
    type = PikaTimeDerivative
    variable = phi
    property = relaxation_time
  [../]
  [./phase_diffusion]
    type = PikaDiffusion
    variable = phi
    property = interface_thickness_squared
  [../]
  [./phase_double_well]
    type = DoubleWellPotential
    variable = phi
    mob_name = mobility
  [../]
[]

[AuxKernels]
  [./phi_aux]
    type = PikaPhaseInitializeAux
    variable = phi_aux
    phase = phi
  [../]
[]

[BCs]
  [./solid]
    type = DirichletBC
    variable = phi
    boundary = 'top left bottom right'
    value = 1
  [../]
```

```
[]
[Executioner]
  type = Transient
  dt = 100
  end_time = 1000
  nl_max_its = 20
  solve_type = PJFNK
  petsc_options_iname = '-ksp_gmres_restart -pc_type -pc_hypre_type'
  petsc_options_value = '50 hypre boomeramg'
  nl_rel_tol = 1e-08
  nl_abs_tol = 1e-12
  l_tol = 1e-8
  l_abs_step_tol = 1e-13
[]
[Adaptivity]
  max_h_level =5
  initial_steps = 5
  steps = 5
  marker = phi_marker
  initial_marker = phi_marker
  [./Indicators]
    [./phi_grad_indicator]
      type = GradientJumpIndicator
      variable = phi
    [../]
  [../]
  [./Markers]
    [./phi_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-7
      indicator = phi_grad_indicator
      refine = 1e-5
    [../]
  [../]
[]

[Outputs]
  print_linear_residuals = true
  print_perf_log = true
  [./out]
    type = Exodus
    file_base = phi_initial_out
    output_final = true
    output_initial = true
  [../]
[]

[ICs]
active = 'phi_full_box_IC'
  [./phi_full_box_IC]
    y2 = 0.005
    y1 = 0
    inside = -1
    x2 = 0.005
    outside = 1
    variable = phi
    x1 = 0
    type = BoundingBoxIC
  [../]
  [./phi_small_box_IC]
    y2 = 0.005
    y1 = 0
```

```
      inside = -1
      x2 = 0.0001
      outside = 1
      variable = phi
      x1 = 0
      type = BoundingBoxIC
    [../]

[]

[PikaMaterials]
  temperature = 263.15
  interface_thickness = 1e-5
  phase = phi
  temporal_scaling = 1
[]
```

C.6 Natural Convection in a Square Ice Enclosure, Ra=1000

```
[Mesh]
  type = GeneratedMesh
  dim = 2
  nx = 50
  ny = 50
  xmin = -1e-4
  ymin = -1e-4
  xmax = .0051
  ymax = .0051
  elem_type = QUAD9
[]
[MeshModifiers]
  [./pin]
    type = AddExtraNodeset
    new_boundary = 99
    coord = '0 0 '
    tolerance = 1e-04
  [../]
[]
[Variables]
  [./v_x]
    order = SECOND
  [../]
  [./v_y]
    order = SECOND
  [../]
  [./p]
  [../]
  [./phi]
  [../]
  [./T]
  [../]
[]
[Functions]
  [./phi_func]
    type = SolutionFunction
    from_variable = phi
    solution = uo_initial
  [../]
[]
[Kernels]
  [./x_momentum_time]
    type = PikaTimeDerivative
    variable = v_x
    coefficient = 1.341
    use_temporal_scaling = false
  [../]
  [./x_momentum]
    type = PikaMomentum
    variable = v_x
    vel_y = v_y
    vel_x = v_x
    component = 0
    p = p
  [../]
  [./x_no_slip]
    type = PhaseNoSlipForcing
    variable = v_x
    phase = phi
```

```
      h = 100
    [../]
    [./x_boussinesq]
      type = Boussinesq
      component = 0
      variable = v_x
      T = T
  [../]

  [./y_momentum_time]
    type = PikaTimeDerivative
    variable = v_y
    coefficient = 1.341
    use_temporal_scaling = false
  [../]
    [./y_momentum]
      type = PikaMomentum
      variable = v_y
      vel_y = v_y
      vel_x = v_x
      component = 1
      p = p
    [../]
    [./y_no_slip]
      type = PhaseNoSlipForcing
      variable = v_y
      phase = phi
      h = 100
    [../]
    [./y_boussinesq]
      type = Boussinesq
      component = 1
      variable = v_y
      T = T
  [../]

    [./mass_conservation]
      type = INSMass
      variable = p
      v = v_y
      u = v_x
      p = p
    [../]

    [./phase_time]
      type = PikaTimeDerivative
      variable = phi
      property = relaxation_time
      use_temporal_scaling = false
    [../]

    [./phase_diffusion]
      type = PikaDiffusion
      variable = phi
      property = interface_thickness_squared
      use_temporal_scaling = false
    [../]
    [./phase_double_well]
      type = DoubleWellPotential
      variable = phi
      mob_name = mobility
    [../]
```

```

 [./heat_time]
    type = PikaTimeDerivative
    variable = T
    property = heat_capacity
    use_temporal_scaling = false
  [../]
  [./heat_convection]
    type = PikaConvection
    property = heat_capacity
    use_temporal_scaling = false
    variable = T
    vel_x = v_x
    vel_y = v_y
  [../]
  [./heat_diffusion]
    type = PikaDiffusion
    property = conductivity
    use_temporal_scaling = true
    variable = T
  [../]

[]
[BCs]
  [./solid_phase_wall]
    type = DirichletBC
    variable = phi
    boundary = 'left right top bottom'
    value = 1
  [../]
  [./pressure_pin]
    type = DirichletBC
    variable = p
    boundary = 99
    value = 0
 [../]
 [./T_hot]
    type = DirichletBC
    variable = T
    boundary = right
    value = 301.7369208944
 [../]
 [./T_cold]
    type = DirichletBC
    variable = T
    boundary = left
    value = 263.15
 [../]
[]

[UserObjects]
  [./uo_initial]
    type = SolutionUserObject
    execute_on = initial
    mesh = phi_initial_out.e-s002
    timestep = 1
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
```

```
    full = true
  [../]
[]

[Executioner]
  type = Transient
  dt = 0.01
  start_time = 0
  end_time = 0.01
  solve_type = PJFNK
  petsc_options_iname = '-ksp_gmres_restart '
  petsc_options_value = '100 '
  l_max_its = 100
  nl_max_its = 150
  nl_rel_tol = 1e-08
  l_tol = 1e-08
  line_search = none

[]
[Adaptivity]
  max_h_level = 5
  initial_steps = 5
  steps = 0
  marker = phi_marker
  initial_marker = phi_marker
  [./Indicators]
    [./phi_grad_indicator]
      type = GradientJumpIndicator
      variable = phi
    [../]
  [../]
  [./Markers]
    [./phi_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-7
      indicator = phi_grad_indicator
      refine = 1e-5
    [../]
  [../]
[]
[Outputs]
  [./console]
    type = Console
    output_linear = false
    output_nonlinear = true
  [../]
  [./exodus]
    file_base = phase_convection_out
    type = Exodus
    output_final = true
    output_initial = true
  [../]
  [./csv]
    file_base = phase_conv
    type = CSV
  [../]
[]


[PikaMaterials]
  phase = phi
  temperature = 263.15
```

```
  interface_thickness = 1e-05
  temporal_scaling = 1
  gravity = '0 -9.81 0'
[]

[ICs]
  [./phase_ic]
    variable = phi
    type = FunctionIC
    function = phi_func
  [../]
  [./T_ic]
    variable = T
    type = FunctionIC
    function =7717.3841788774*x+263.15
  [../]
[]
```

C.7 Stehle's Migrating Bubble Initialization

```
[Mesh]
  type = GeneratedMesh
  dim = 2
  nx = 50
  ny = 50
#  xmax = 0.0025
  xmax = 0.005
  ymax = 0.005
  elem_type = QUAD9
  []

[Variables]
  [./phi]
  [../]
[]

[AuxVariables]
  [./u]
  [../]
  [./phi_aux]
  [../]
[]

[Kernels]
  [./phase_time]
    type = PikaTimeDerivative
    variable = phi
    property = relaxation_time
  [../]
  [./phase_diffusion]
    type = PikaDiffusion
    variable = phi
    property = interface_thickness_squared
  [../]
  [./phase_double_well]
    type = DoubleWellPotential
    variable = phi
    mob_name = mobility
  [../]
[]

[AuxKernels]
  [./phi_aux]
    type = PikaPhaseInitializeAux
    variable = phi_aux
    phase = phi
  [../]
[]

[Executioner]
  # Preconditioned JFNK (default)
  type = Transient
  dt = 10
  solve_type = PJFNK
  petsc_options_iname = '-ksp_gmres_restart -pc_type -pc_hypre_type'
  petsc_options_value = '50 hypre boomeramg'
  nl_rel_tol = 1e-07
  nl_abs_tol = 1e-12
  l_tol = 1e-4
```

```
  [./TimeStepper]
    type = IterationAdaptiveDT
    dt = 1
    growth_factor = 3
  [../]
  num_steps = 10
[]

[Adaptivity]
  max_h_level = 4
  initial_steps = 4
  marker = phi_marker
  initial_marker = phi_marker
  [./Indicators]
    [./phi_grad_indicator]
      type = GradientJumpIndicator
      variable = phi
    [../]
  [../]
  [./Markers]
    [./phi_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-7
      indicator = phi_grad_indicator
      refine = 1e-5
    [../]
  [../]
[]

[Outputs]
  output_initial = true
  print_linear_residuals = true
  print_perf_log = true
  [./out]
    output_final = true
    type = Exodus
    interval = 1
  [../]
[]

[ICs]
  [./phase_ic]
    int_width = 1e-5
    x1 = 0.0025
    y1 = 0.0025
    radius = 0.0005
    outvalue = 1
    variable = phi
    invalue = -1
    type = SmoothCircleIC
  [../]
[]

[PikaMaterials]
  temperature = 258.2
  interface_thickness = 1e-5
  phase = phi
  temporal_scaling = 1e-04
[]
```

C.8 Stehle's Migrating Bubble

```
[Mesh]
  type = GeneratedMesh
  dim = 2
  nx = 50
  ny = 50
  xmax = 0.0025
  ymax = 0.005
  elem_type = QUAD9
[]

[MeshModifiers]
  [./pin]
    type = AddExtraNodeset
    new_boundary = 99
    coord = '0.0 0.0 '
  [../]
[]
[Variables]
  [./v_x]
    order = SECOND
  [../]
  [./v_y]
    order = SECOND
  [../]
  [./p]
  [../]
  [./phi]
  [../]
  [./T]
  [../]
  [./X]
  [../]
[]

[AuxVariables]
  [./phi_aux]
  [../]
[]

[Functions]
  [./T_func]
    type = ParsedFunction
    value = -543*y+267.515
  [../]
  [./phi_func]
    type = SolutionFunction
    from_variable = phi
    solution = phi_initial
  [../]
[]

[Kernels]
  [./x_momentum_time]
    type = PikaTimeDerivative
    variable = v_x
    coefficient = 1.341
    use_temporal_scaling = false
  [../]
  [./x_momentum]
```

```
      type = PikaMomentum
      variable = v_x
      vel_y = v_y
      vel_x = v_x
      component = 0
      p = p
    [../]
    [./x_no_slip]
      type = PhaseNoSlipForcing
      variable = v_x
      phase = phi
      h = 100
    [../]
    [./x_momentum_boussinesq]
      type = Boussinesq
      variable = v_x
      component = 0
      T = T
    [../]

  [./y_momentum_time]
    type = PikaTimeDerivative
    variable = v_y
    coefficient = 1.341
    use_temporal_scaling = false
  [../]

    [./y_momentum]
      type = PikaMomentum
      variable = v_y
      vel_y = v_y
      vel_x = v_x
      component = 1
      p = p
    [../]
    [./y_no_slip]
      type = PhaseNoSlipForcing
      variable = v_y
      phase = phi
      h = 100
    [../]
    [./y_momentum_boussinesq]
      type = Boussinesq
      variable = v_y
      component = 1
      T = T
    [../]
    [./mass_conservation]
      type = INSMass
      variable = p
      u = v_y
      v = v_x
      p = p
    [../]

    [./phi_time]
      type = PikaTimeDerivative
      variable = phi
      property = relaxation_time
      use_temporal_scaling = false
    [../]
    [./phi_diffusion]
```

```
      type = PikaDiffusion
      variable = phi
      property = interface_thickness_squared
      use_temporal_scaling = false
    [../]
    [./phi_double_well]
      type = DoubleWellPotential
      variable = phi
      mob_name = mobility
    [../]
    [./phi_transition]
      type = PhaseForcing
      variable = phi
      chemical_potential = X
      property = phase_field_coupling_constant
      use_temporal_scaling = false
    [../]

  [./Heat_time]
    type = PikaTimeDerivative
    variable = T
    property = heat_capacity
  [../]
    [./Heat_convection]
      type = PikaConvection
      variable = T
      vel_x = v_x
      use_temporal_scaling = false
      property = heat_capacity
      vel_y = v_y
    [../]
    [./Heat_diffusion]
      type = PikaDiffusion
      variable = T
      use_temporal_scaling = true
      property = conductivity
    [../]
    [./Heat_phi_time]
      type = PikaCoupledTimeDerivative
      variable = T
      use_temporal_scaling = true
      property = latent_heat
      coupled_variable = phi
      scale = -0.5
    [../]
  [./Vapor_time]
    type = PikaTimeDerivative
    variable = X
    coefficient = 1.0
    use_temporal_scaling = false
  [../]
    [./Vapor_convection]
      type = PikaPhaseConvection
      variable = X
      vel_x = v_x
      use_temporal_scaling = false
      phase = phi
      coefficient = 1.0
      vel_y = v_y
    [../]
    [./Vapor_diffusion]
      type = PikaDiffusion
```

```
      variable = X
      use_temporal_scaling = true
      property = diffusion_coefficient
    [../]
    [./Vapor_phi_time]
      type = PikaCoupledTimeDerivative
      variable = X
      use_temporal_scaling = true
      coupled_variable = phi
      coefficient = 1
      scale = 0.5
    [../]
[]
[AuxKernels]
    [./phi_aux_kernel]
      type = PikaPhaseInitializeAux
      variable = phi_aux
      phase = phi
    [../]
[]
[BCs]
    [./T_hot]
      type = DirichletBC
      variable = T
      boundary = bottom
      value = 267.515
    [../]
    [./T_cold]
      type = DirichletBC
      variable = T
      boundary = top
      value = 264.8
    [../]

[]

[Postprocessors]
[]

[UserObjects]
    [./phi_initial]
      type = SolutionUserObject
      mesh = phi_initial_1e5_out.e-s009
      system_variables = phi
    [../]
[]

[Executioner]
  type = Transient
  dt = 0.01
  end_time = 1000
  solve_type = PJFNK
  petsc_options_iname = '-ksp_gmres_restart '
  petsc_options_value = '100 '
  l_max_its = 100
  nl_max_its = 150
  nl_rel_tol = 1e-08
  l_tol = 1e-08
  line_search = none
  scheme = 'crank-nicolson'
[]
```

```
[Adaptivity]
  max_h_level = 5
  marker = combo_marker
  initial_steps = 5
  initial_marker = combo_marker
  [./Indicators]
    [./phi_grad_indicator]
      type = GradientJumpIndicator
      variable = phi
    [../]
    [./X_grad_indicator]
      type = GradientJumpIndicator
      variable = X
    [../]
  [../]
  [./Markers]
    [./combo_marker]
      type = ComboMarker
      markers = 'phi_grad_marker X_grad_marker '
    [../]
    [./X_grad_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-10
      indicator = X_grad_indicator
      refine = 1e-8
    [../]
    [./phi_grad_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-7
      indicator = phi_grad_indicator
      refine = 1e-5
    [../]
  [../]
[]

[Outputs]
  output_initial = true
  exodus = true
  csv = true
  print_linear_residuals = true
  print_perf_log = true
[]

[ICs]
  [./phase_ic]
    variable = phi
    type = FunctionIC
    function = phi_func
  [../]
  [./temperature_ic]
    variable = T
    type = FunctionIC
    function = T_func
  [../]
  [./vapor_ic]
    variable = X
    type = PikaChemicalPotentialIC
    block = 0
    phase_variable = phi
    temperature = T
  [../]
[]
```

```
[PikaMaterials]
  temperature = T
  interface_thickness = 1e-5
  temporal_scaling = 1
  condensation_coefficient = .01
  phase = phi
  gravity = '0 -9.81 0'
[]

[PikaCriteriaOutput]
  air_criteria = false
  velocity_criteria = false
  time_criteria = false
  vapor_criteria = false
  chemical_potential = X
  phase = phi
  use_temporal_scaling = true
  ice_criteria = false
  super_saturation = false
  interface_velocity_postprocessors = max
  temperature = T
[]
```

C.9 Natural Convection in Sloped Snow Initialization

```
[Mesh]
  # uniform_refine = 6
  type = GeneratedMesh
  dim = 2
  nx = 50
  ny = 50
  xmax = .005
  ymax = .005
[]

[MeshModifiers]
[]

[Variables]
  [./phi]
  [../]
[]

[AuxVariables]
  [./u]
  [../]
  [./phi_aux]
  [../]
[]

[Functions]
  [./snow_ct]
    type = ImageFunction
    upper_value = 1
    lower_value = -1
    file = snow_small.png
    threshold = 128
  [../]
[]

[Kernels]
  [./phase_time]
    type = PikaTimeDerivative
    variable = phi
    property = relaxation_time
  [../]
  [./phase_diffusion]
    type = PikaDiffusion
    variable = phi
    property = interface_thickness_squared
  [../]
  [./phase_double_well]
    type = DoubleWellPotential
    variable = phi
    mob_name = mobility
  [../]
[]

[AuxKernels]
  [./phi_aux]
    type = PikaPhaseInitializeAux
    variable = phi_aux
    phase = phi
  [../]
```

```
[]

[BCs]
  [./Periodic]
    [./phi_periodic]
      variable = phi
      auto_direction = 'x y'
    [../]
  [../]
[]

[Adaptivity]
  max_h_level = 3
  initial_steps = 3
  marker = phi_marker
  initial_marker = phi_marker
  [./Indicators]
    [./phi_grad_indicator]
      type = GradientJumpIndicator
      variable = phi
    [../]
  [../]
  [./Markers]
    [./phi_marker]
      type = ErrorToleranceMarker
      coarsen = 1e-7
      indicator = phi_grad_indicator
      refine = 1e-5
    [../]
  [../]
[]

[Executioner]
  # Preconditioned JFNK (default)
  type = Transient
  dt = 10
  solve_type = PJFNK
  petsc_options_iname = '-ksp_gmres_restart -pc_type -pc_hypre_type'
  petsc_options_value = '50 hypre boomeramg'
  nl_rel_tol = 1e-07
  nl_abs_tol = 1e-12
  l_tol = 1e-4
  num_steps = 10
  [./TimeStepper]
    type = IterationAdaptiveDT
    dt = 1
    growth_factor = 3
  [../]
[]

[Outputs]
  output_initial = true
  console = false
  print_linear_residuals = true
  print_perf_log = true
  [./out]
    output_final = true
    type = Exodus
  [../]
[]

[ICs]
```

```
  [./phase_ic]
    variable = phi
    type = FunctionIC
    function = snow_ct
  [../]
[]

[PikaMaterials]
  temperature = 263.15
  interface_thickness = 1e-5
  phase = phi
  temporal_scaling = 1e-04
  condensation_coefficient = .1
[]
```

C.10 Natural Convection in Sloped Snow, 500K/M

```

[Mesh]
  type = GeneratedMesh
  dim = 2
  nx = 50
  ny = 50
  xmax = 0.005
  ymax = 0.005
  elem_type = QUAD9
[]

[MeshModifiers]
  [./pin]
    type = AddExtraNodeset
    coord = '0 0'
    new_boundary = 99
  [../]
[]

[Variables]
  [./v_x]
    order = SECOND
  [../]
  [./v_y]
    order = SECOND
  [../]
  [./p]
  [../]
  [./phi]
  [../]
  [./T]
  [../]
[]

[Functions]
  [./phi_func]
    type = SolutionFunction
    from_variable = phi
    solution = uo_initial
  [../]
  [./T_func]
    type = ParsedFunction
    value = -500*y+265.65
  [../]
[]

[Kernels]
  [./x_momentum_time]
    type = PikaTimeDerivative
    variable = v_x
    coefficient = 1.341
    use_temporal_scaling = false
  [../]
  [./x_momentum]
    type = PikaMomentum
    variable = v_x
    vel_y = v_y
    vel_x = v_x
    component = 0
```

```
    p = p
  [../]
  [./x_boussinesq]
    type = Boussinesq
    variable = v_x
    T = T
    component = 0
  [../]
  [./x_no_slip]
    type = PhaseNoSlipForcing
    variable = v_x
    phase = phi
    h = 100
  [../]


  [./y_momentum_time]
    type = PikaTimeDerivative
    variable = v_y
    coefficient = 1.341
    use_temporal_scaling = false
  [../]
  [./y_momentum]
    type = PikaMomentum
    variable = v_y
    vel_y = v_y
    vel_x = v_x
    component = 1
    p = p
  [../]
  [./y_boussinesq]
    type = Boussinesq
    variable = v_y
    T = T
    component = 1
  [../]
  [./y_no_slip]
    type = PhaseNoSlipForcing
    variable = v_y
    phase = phi
    h = 100
  [../]


  [./mass_conservation]
    type = INSMass
    variable = p
    v = v_y
    u = v_x
    p = p
  [../]


  [./phase_time]
    type = PikaTimeDerivative
    variable = phi
    property = relaxation_time
    use_temporal_scaling = false
  [../]
  [./phase_diffusion]
    type = PikaDiffusion
    variable = phi
```

```
    property = interface_thickness_squared
    use_temporal_scaling = false
  [../]
  [./phase_double_well]
    type = DoubleWellPotential
    variable = phi
    mob_name = mobility
  [../]


  [./heat_time]
    type = PikaTimeDerivative
    variable = T
    property = heat_capacity
    scale = 1.0
  [../]
  [./heat_convection]
    type = PikaConvection
    variable = T
    use_temporal_scaling = true
    property = heat_capacity
    vel_x = v_x
    vel_y = v_y
  [../]
  [./heat_diffusion]
    type = PikaDiffusion
    variable = T
    use_temporal_scaling = true
    property = conductivity
  [../]
  [./heat_phi_time]
    type = PikaCoupledTimeDerivative
    variable = T
    property = latent_heat
    scale = -0.5
    use_temporal_scaling = true
    coupled_variable = phi
  [../]

[]
[BCs]
  [./Periodic]
    [./periodic_v_x]
      variable = v_x
      auto_direction = 'x y'
    [../]
    [./periodic_v_y]
      variable = v_y
      auto_direction = 'x y'
    [../]
    [./periodic_phi]
      variable = phi
      auto_direction = ' x y'
    [../]
    [./periodic_T]
      variable = T
      auto_direction = 'x'
    [../]
  [../]
#  [./pressure]
#    type = DirichletBC
#    variable = p
```

```
#     boundary = 99
#     value = 0
#   [../]
  [./T_cold]
    type = DirichletBC
    variable = T
    boundary = top
    value = 263.15
  [../]
  [./T_hot]
    type = DirichletBC
    variable = T
    boundary = bottom
    value = 265.65
  [../]
#   [./free_slip]
#     type = DirichletBC
#    variable = v_y
#    boundary = 'top bottom'
#    value = 0
#  [../]
[]

[UserObjects]
  [./uo_initial]
    type = SolutionUserObject
    execute_on = initial
    mesh = phi_initial_small_out.e-s009
    timestep = 1
  [../]
[]

[Preconditioning]
  [./SMP_PJFNK]
    type = SMP
    full = true
  [../]
[]

[Executioner]
  type = Transient
  dt = 0.01
  end_time = 0.5
  solve_type = PJFNK
  petsc_options_iname = '-ksp_gmres_restart '
  petsc_options_value = '100 '
  l_max_its = 100
  nl_max_its = 150
  nl_rel_tol = 1e-08
  l_tol = 1e-08
  line_search = none
[]

[Adaptivity]
  max_h_level = 4
  marker = phi_marker
  initial_steps = 4
  initial_marker = phi_marker
  [./Indicators]
    [./phi_grad_indicator]
      type = GradientJumpIndicator
      variable = phi
```

```
      [../]
    [../]
    [./Markers]
      [./phi_marker]
        type = ErrorToleranceMarker
        coarsen = 1e-4
        indicator = phi_grad_indicator
        refine = 1e-3
      [../]
    [../]
[]

[Outputs]
  output_initial = true
  output_final = true
  exodus = true
#  console = true
  interval = 1
#  print_linear_residuals = true
#  print_perf_log = true
[]

[PikaMaterials]
  temperature = T
  interface_thickness = 1e-05
  temporal_scaling = 1
  condensation_coefficient = .01
  phase = phi
#Slope of 30 degrees
  gravity = '4.905 -8.49571 0'
[]

[ICs]
  [./phase_ic]
    variable = phi
    type = FunctionIC
    function = phi_func
  [../]
  [./temperature_ic]
    variable = T
    type = FunctionIC
    function = T_func
  [../]
[]
```