Exploring the Interiors of Icy Planetary Bodies Via Geodynamical Modeling

BY

SEAN O'HARA
B.A., Bowdoin College, 2005
B.S., Northeastern Illinois University, 2013

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Earth and Environmental Sciences
in the Graduate College of the
University of Illinois at Chicago, 2019

Chicago, Illinois

Defense Committee:

Andrew J. Dombard, Chair and Advisor
Roy Plotnick, Earth and Environmental Sciences
Stefany Sit, Earth and Environmental Sciences
Carol Stein, Earth and Environmental Sciences
Mike Sori, Lunar and Planetary Laboratory, University of Arizona

This thesis is dedicated to all the people in my life who treated me with skepticism and inspired me to prove them wrong, as well as all the people who supported me unconditionally and inspired me to live up to their example.

ACKNOWLEDGEMENTS

I would like to thank my thesis committee—Andrew Dombard, Roy Plotnick, Stefany Sit, Carol Stein, and Mike Sori—for their support and assistance in completing this thesis. They, along with the other faculty, staff, and graduate students of the Earth and Environmental Sciences department, have provided invaluable mentoring and support in my research, teaching, personal, and professional life, without which I would not be here. I would also like to acknowledge the faculty of the Earth Sciences department at Northeastern Illinois University, including my undergraduate advisor Dr. Genet Ide Duke, for providing a valuable and professionally supportive undergraduate education in the earth sciences for a returning student.

I would also like to acknowledge the members of the New Horizons team whose discoveries formed the basis of much of the work in this thesis. I would especially like to thank Paul Schenk, Orkan Umurhan, and Kelsi Singer, conversations with whom provided impetus and valuable feedback for the work presented here.

CONTRIBUTION OF AUTHORS

Two chapters in this work are written in the plural voice. These chapters are intended for eventual publication in a scientific journal, and the plural voice reflects that of both the author and the co-author Andrew J. Dombard, the advisor for this thesis. All of the research, analysis, and writing for this thesis was completed solely by the author.

TABLE OF CONTENTS

CHAPTER PAGE

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Exploring and understanding the geologic processes that have shaped the other planets in our solar system is a challenging feat. Our understanding of our own planet's history and ongoing evolution is based on vast sets of data built up from detailed field work, observations of phenomena going back centuries, and archives of physical samples. The enormous technical challenges of space travel mean that we have only a fraction of the same data for our terrestrial neighbors. Only a single trained geologist has stood on the Moon, and while Mars has been host to many capable robotic landers, they are still limited to small areas and constrained by their scientific instrumentation.

The case is yet worse for the icy moons and planets of our outer solar system. The challenge and expense of a dedicated orbiter for one of these bodies, much less a lander, has so far left them in the realm of fly-bys or shared observation during missions dedicated to their gas giant hosts. On top of this, the icy shells of these bodies resist our usual abilities of geologic inference, showing us surface compositions and landforms that are unique to their class and between each other. Lacking the ability to create detailed geologic maps and histories, we must instead make the most of our limited data sets to determine the history of these far-off bodies.

Of the methods used to analyze icy planetary bodies, geophysical modeling has been one of the most consistently useful. By considering the body as a system defined by mathematically related principles- the transfer of heat, the force of gravity, and the conservation of mass and energy- we can build simplified models of these interactions within the observable or inferable constraints of the body. These models may be as simple as a force-balance model and as sophisticated as a coupled series of non-linear equations simulated over a broad parameter space, but they share the basic functionality of attempting to constrain the possible expressions of a complicated set of observable phenomena. In other words, to the extent that it is possible, they must couple a comprehensive understanding of what we *do* know (correct material parameters, prior assumptions, and geologic and geophysical inference) with effective methods (analytical or numerical simulation) to reach a conclusion (a result that can reasonably explain the phenomena under study).

I present the results of three studies that utilize numerical methods to explore the link between the internal geophysical state of icy solar system bodies and their exteriors. While each study is concerned with a different planetary body and utilizes different geophysical approaches, they are linked by the philosophy of geophysical modeling outlined above: the observation of an unusual planetary feature uniquely related to its status as an icy body; the use of a numerical simulation to solve the equations describing a simplified model of the system; and the evaluation of the model results to find a set of solutions that could realistically constrain the formation of the features of interest.

The first study explores the emplacement of the chaotic mountain block ranges found along the eastern rim of Pluto's Sputnik Planitia basin. These mountainous blocks, partly immersed in the giant nitrogen glacier that fills the basin, have a jumbled appearance that does not appear consistent with formation by tectonic activity. I create a force-balance model testing the hypothesis that the chaotic arrangement of these mountains was caused by downhill transport of coherent mountain blocks by basally lubricated sliding along a shallow incline. Conceptually this mechanism is similar to the "sliding stones" of Racetrack Playa (e.g., Lorenz, 2011), except gravitationally driven instead of by wind, and with the lubricating agent being solid or liquid nitrogen instead of water. I also account for the force of convecting nitrogen on the face of the block as it descends into the basin. I find that, depending on the lubrication regime, even the largest mountain blocks can travel over the very shallow slopes of the basin in geologically short periods of time, after which they will be naturally halted by the loss of slope and the viscous resisting force of the solid nitrogen.

The second study is focused on the enigmatic “cantaloupe terrain” of Neptune’s moon Triton. Schenk and Jackson (1993) hypothesized that this terrain, defined by a wide region of interfering curvilinear ridges and depressions, was the surface expression of a compositional diapir canopy composed of solid CO_2 or ammonia dihydrate overlying a water ice crust. However, the method used did not account for the temperature-dependent material response or the thermally insulating properties of the materials, which were poorly known at the time. To provide a more robust test of the compositional diapir hypothesis, I design numerical convection simulations that incorporated more recent understanding of Triton’s internal heat state and the material properties of the putative diapir-forming layers. I do not find any parameter space in which diapir formation occurred. The insulating nature of the surface ices creates high geothermal gradients in the near surface, which due to the significantly different material properties results in viscosity contrasts that prevent upward flow of ice. However, for certain model setups, I do observe high stress localization that may be consistent with a fracture-induced sublimation formation mechanism. I consider these results to be suggestive of a possible formation mechanism, but constraining the time scales and brittle rheological behavior of the overlying layers will require additional studies beyond the scope of this work.

The third study develops a method for understanding the interior structure of a planetary body from observations of its spatially variable gravitational field. Typically in the planetary sciences, this task is done by an inverse modeling method, but in situations where the gravity field is difficult to measure fully, certain assumptions about a planet’s geophysical state must be assumed for inverse modeling to function. I present an alternate, forward-modeling path of determining a planet's spheroidal density structure by utilizing genetic algorithms and a simplified potential field generation model to sort efficiently through the model space in a way that bypasses the assumptions required for the traditional inverse modeling approach. I foresee this code as a community tool for hypothesis testing purposes, allowing researchers to test the feasibility of various density structures and predict the resulting gravity potential fields of the body. The code is currently in a preliminary state and requires additional bug testing and features to be a useful community tool, but I present two benchmark cases: one for Jupiter's moon Europa and one for the dwarf planet Ceres. These models attempt to replicate the reverse-modeling predictions of Anderson et al. (1998) for Europa and of Park et al. (2016) for Ceres.

2. DOWNHILL SLEDDING AT 40 AU: MOBILIZING PLUTO'S CHAOTIC MOUNTAIN BLOCKS

2.1 Abstract

We present a force-balance analysis of the mobilization and emplacement of the large mountain block chains observed on the western rim of Pluto's Sputnik Planitia basin. These mountain blocks are likely disrupted pieces of Pluto's water ice crust. After fracturing from the basin rim, they slid downslope into the basin under their own weight on timescales as short as hundreds to thousands of years, collecting into the observed ranges when the basin slope shallows below a critical value. Solid nitrogen ice played an important role in promoting the fragmentation and lubrication of the blocks as they moved, but buoyant movement and lateral convective forces within the nitrogen did not play a significant role in the mobilization of all but the smallest of the observed blocks.

2.2 Introduction

The western rim of Pluto's informally named Sputnik Planitia, a large (~900,000 km^2) basin near the dwarf planet's equator, is dominated by a north-south trending series of mountain block chains (Fig. 2.1). Each chain in the series is a cluster of blocks of fragmental appearance, with larger blocks oriented or tilted in a chaotic manner and often embedded in what appears to be a matrix of smaller blocks. Individual blocks reach a mode height of ~1 km above the surrounding plains and have a maximum height of ~4 km, while block widths have a mode diameter of 5-10 km and a maximum diameter of ~40 km (Schenk et al., 2018). The size and high relief of the blocks indicate that they are likely composed of water ice, which comprises the

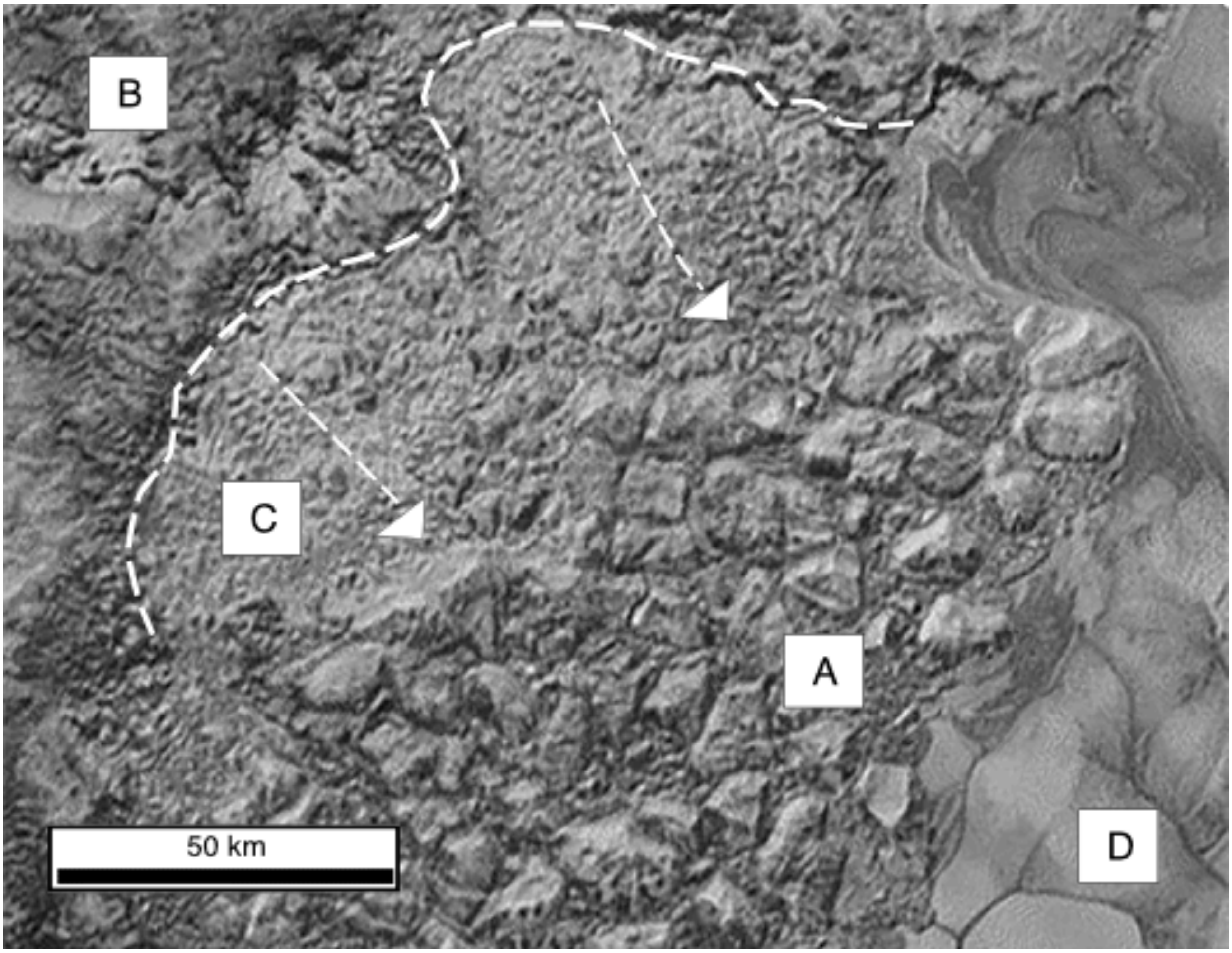

Figure 2.1. New Horizons LORRI image at a resolution of 386 m/pixel of the northern margin of the al-Idrisi Montes, showing the geologic terrains described in this paper. North is at top of image. Area A: The main body of mountain blocks. Block relief is on average 1-2 km. Area B: Disrupted water ice uplands. The detachment point for the mountain blocks is marked by the dotted line. Area C: Hummocky terrain dividing the mountain blocks from the uplands. Arrows indicate proposed direction of ice block transport. Area D: The nitrogen ice sheet of Sputnik Planitia. Image modified from NASA Photojournal image PIA19934, courtesy NASA/Johns Hopkins University Applied Physics Laboratory/Southwest Research Institute.

majority of Pluto’s crust and is very rigid at the average surface temperature of 44 K (Stern et al., 2015). It is thus likely that the mountain blocks represent disrupted pieces of Pluto’s crust (area A in Fig. 2.1).

White et al. (2017) suggested that the mountain blocks are transported material, formed from large mass wasting or buoyant rafting of basin rim crustal material. The authors identified a region to the north of the al-Idrisi Montes that may be undergoing disruption, fragmentation, and subsidence due to nitrogen ice exploiting regional tectonic faults (area B in Fig. 2.1). To the south of these faults, the landscape has a hummocky appearance similar to the inter-block material that embays the mountain blocks (area C in Fig. 2.1). Schenk et al. (2018) additionally identified a north-south trending fault system to the west of Sputnik Planitia (southwest of the area imaged by Fig. 2.1), which is separated from the mountain blocks by inter-block material and a series of ridges mantled by dark material. If the mountains are transported material and these faults are detachment points from which the mountain blocks were separated from the crust and subsequently transported to their current location, then the mantled ridges and hummocky terrains are presumably lag deposits from the transport, possibly analogous to terrestrial landslide deposits.

The basin that encompasses Sputnik Planitia is likely the surface expression of an ancient giant impact crater (Nimmo et al., 2016). The basin serves as a cold trap for nitrogen, which has condensed to fill the basin with a convecting nitrogen ice sheet (area D in Fig. 2.1; McKinnon et al., 2016). The bedrock water ice in the region is likely heavily fractured, both from the impact itself and from subsequent extensional tectonics as Pluto’s ice shell freezes (Nimmo et al., 2016). The basin also would possess a regional inward slope near the basin rim, although the nitrogen ice sheet obscures it.

Nitrogen ice is denser (~1000 kg m^{-3}) than water ice (~920 kg m^{-3}) and flows readily at plutonian surface temperatures of ~40 K. Nitrogen ice can efficiently exploit preexisting fractures in the crust, pressurizing and possibly lubricating fault systems. Failure along these faults would create debris that could then be transported to its current location, either by downslope mass movement or buoyantly, if the nitrogen is deep enough. Based on crater scaling relationships and the melting point of solid nitrogen, the maximum likely depth of Sputnik Planitia is ~6 km, and the rim where the mountain block ranges are found will be considerably shallower, grading toward zero at the edge of the ice sheet. Buoyant transport of the larger observed blocks would take a significantly deeper amount of nitrogen than is currently present or likely, even given evidence of past glacial extent and climate change (White et al., 2017; Bertrand and Forget, 2016). We thus focus our investigation on how these grounded blocks may be moved downslope under their own weight. Because the blocks are at least partially submerged in convecting nitrogen (White et al., 2017), we additionally consider the lateral forces applied to the block by neighboring convection cells.

2.3 **Methodology**

We begin by creating a simplified force-balance model of a cylindrical mountain block sitting on a shallow slope and partially immersed in convecting nitrogen. Figure 2.2 summarizes the forces considered. The weight of the block (F_W) is separated into two components: the slope-normal force (F_N) and the force oriented downslope (F_D). F_D is opposed by the basal friction force (F_F) and modified by any net lateral forces from convection in the nitrogen ice that fills the basin (F_C). If there is a net downhill force (F_D - F_F > 0), it will be balanced by the viscous drag force of the low viscosity nitrogen ice (F_V), effectively a geometrical modified Stokes flow (e.g.,

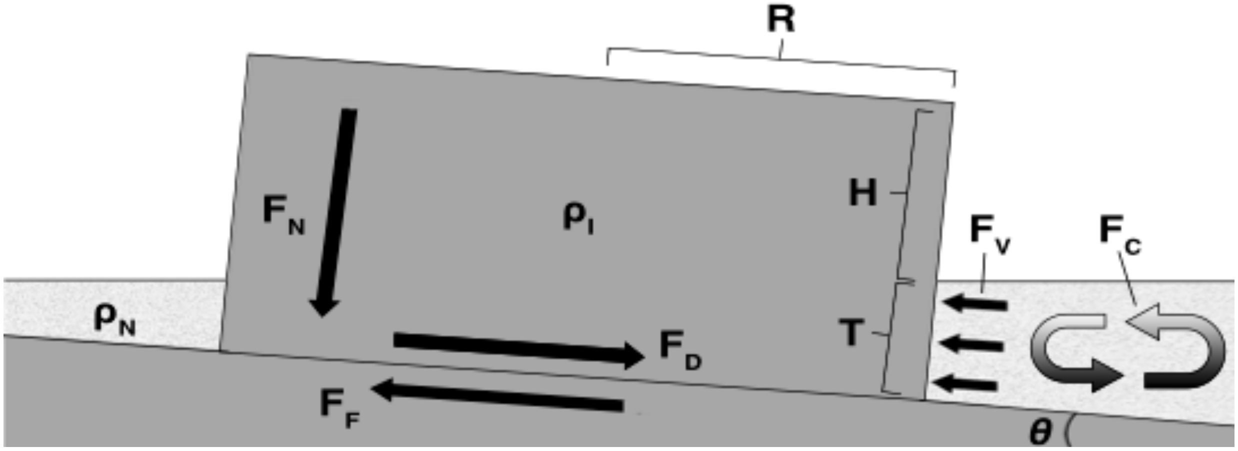

Figure 2.2. Force balance diagram of a cylindrical mountain block with water ice density ρ_I, sitting on a slope of angle θ and partially immersed in nitrogen with density ρ_N. The radius, R, and the height above and below nitrogen, H and T, respectively, represent the dimensions of the block. Directions of the normal force F_N, the force directed downslope F_D, the friction force F_F, the convective force F_C, and the viscous resistance F_V are indicated by arrows. The lengths of the respective arrows are not representative of the relative magnitude of the forces.

Turcotte and Schubert, 2014). To calculate these forces, we require estimates of the coefficient of friction μ and its contribution to the basal friction force F_F; the net lateral forces of convection acting on a block F_C; the contribution of partial buoyancy to the weight of the block F_W; and a Stokes flow derivation of the viscous drag forces F_V.

2.3.1 **Modeling convection**

To estimate the convective forces F_C, we implement a material model for solid nitrogen in the finite element code ASPECT (Advanced Solver for Problems in Earth's ConvecTion [Kronblicher et al., 2012]). Modeling convection requires an estimate of the dynamic viscosity of the material (η), which depends on its material properties and the deformation mechanism active at a given temperature and strain rate. The material properties of solid nitrogen are not well understood - see the discussions in McKinnon et al. (2016) and Umurhan et al. (2017). For our purposes, we utilize a Newtonian, linear viscosity based on work by Eluszkiewicz and Stevenson (1990) and expanded in Umurhan et al. (2017):

$$\eta_{N_2(i)} = \frac{kTd^2}{42D_{0V}\Omega}, \qquad (2.1)$$

where k is the Boltzmann constant, T is temperature in Kelvin, d is the ice grain size, D_{0V} is the empirically assessed self-diffusion rate through N_2 grains ($1.6 \times 10^{-7}e^{-Tv/T}$ m^2/s, where T_v is the activation temperature ≈1030 K), and Ω is the volume of a single molecule of N_2 (4.9×10^{-29} m^3). Note that the only non-constant values in this equation are the temperature and the grain size, which we set as 1 mm. The actual N_2 ice grain size is not well constrained, but this size is consistent with grain sizes reported for convecting mantle rock and glacial ice (McKinnon et al., 2016). This model assumes that grain-boundary sliding (GBS) is the dominant creep mechanism, which is dominant at low stresses near the melting point. Material parameters used in the simulation are given in Table I.

Nitrogen convection is modeled in a wide (~1:20 aspect ratio) 2D box, bounded by immobile water ice, with nitrogen depths varying from 500 m to 2 km. The surface temperature is fixed at 37 K, and the bottom temperature is adjusted until it provides a convective heat flux equal to Pluto's estimated radiogenic production (~3 mW m^{-2}) (Robuchon and Nimmo, 2011). No-slip boundary conditions are applied to the base, representing nitrogen in contact with bedrock ice. The side boundary corresponding to the ice block is also set to no-slip conditions, with a temperature profile fixed to that of thermally conductive water ice.

Convection simulation results

Convective systems within the nitrogen are observed to develop and exhibit traction on the side boundary representing the mountain block. The convection cells have wide aspect ratios,

TABLE I: NITROGEN MATERIAL PARAMETERS

Parameter	Value	Source
Reference temperature	45 K	Yamashita et al. (2010)
Density at reference temperature[a] ρ_0	980 kg m^{-3}	McKinnon et al. (2016)
Viscosity at reference temperature[b] η_0	2 x 10^{10} Pa s	Elusciewicz and Stevenson (1990)
Thermal conductivity k	0.2 W m^{-1} K^{-1}	McKinnon et al. (2016)
Thermal expansion coefficient α	2 x 10^{-3} K^{-1}	McKinnon et al. (2016)
Activation energy[c] E_A	3.5 kJ mol^{-1}	McKinnon et al. (2016)
Specific heat C_p	1606 J kg^{-1} K^{-1}	Scott (1976)
Surface gravity g	0.617 m s^{-2}	Stern et al. (2015)

[a] Based on the formula $\rho_0 = 1000 - 2.14(T - 36\text{K})$ kg m^{-3}.

[b] See equation 1.

[c] For power law exponent $n = 2.2$.

surface velocities of approximately tens of centimeters per year, and surface heat fluxes of ~3 mW m^{-2}. In these respects, they closely resembled the sluggish-lid convection found by McKinnon et al. (2016) for convection within Sputnik Planitia; we thus consider our simulation to be well representative of the current understanding of nitrogen convection on Pluto.

Integrated deviatoric stresses experienced on the no-slip side boundary are determined over a range of nitrogen depths, from 500 m to 2 km, and for a range of block radii (and corresponding block weight F_W) varying from 1 km to 20 km. In all cases, the magnitude of F_C is 3-4 orders of magnitude less than that of F_W or of the other lateral forces considered here. We thus conclude that lateral convective forces are not an important component in the transport of large, grounded mountain blocks, and do not incorporate F_C into subsequent force-balance equations.

In the limit of a buoyantly floating block, F_w approaches zero. In these cases, convective forces would be an effective transport mechanism, and can explain the presence of small isolated hills found in the middle of Sputnik Planitia, particularly in the boundaries between convective cells (White et al., 2017). However, lateral convective forces cannot explain the transport of the majority of the mountain block ranges.

2.3.2 <u>**Modeling force balance**</u>

To model the downhill force on a partially submerged mountain block, we consider a cylindrical block with height above nitrogen H, depth below nitrogen T, and radius R (see Fig. 2). Because the blocks are sitting on the slope of the basin, their basinward edge will be sitting in deeper nitrogen than their uphill edge, which may provide additional buoyant support. However for the shallow (< 3°) slopes expected, we neglect downhill variances in H and T. The total weight of the grounded block is thus

$$F_W = (\rho_I g H - \Delta\rho T) g \pi R^2 , \qquad (2.2)$$

where ρ_I is the water ice density (920 kg m^{-3}), $\Delta\rho$ is the density difference between water ice and solid nitrogen (~80 kg m^{-3}), and g is Pluto's gravity (0.617 m s^{-2}).

From elementary analysis of forces acting on a slope, the total lateral force F_T is the difference between the downslope and frictional forces ($F_D - F_F$):

$$F_T = F_W (\sin\theta - \mu \cos\theta) , \qquad (2.3)$$

where θ is the angle of slope of the basin. We can estimate the coefficient of friction μ required for a given slope by noting that for F_T to be positive,

$$\mu < \tan\theta , \qquad (2.4)$$

since F_W is positive by definition.

We can infer a likely value of θ from rims of other large impact basins in the Solar System. South Polar Aitken on the Moon and the Hellas basin on Mars both have slopes of ~2° near their rims (Rosenburg et al., 2011; Tanaka et al., 2002). Substituting this value into Eq. (2.4) results in a maximum μ for movement of ≈ 0.04. Such values are achieved in full-film regimes (see below).

2.3.3 **Modeling viscous drag**

If there is a net positive force ($F_T > 0$), then the block will slide downslope into the nitrogen glacier. Its downslope velocity will be controlled by the viscous drag of the nitrogen around the block, which balances F_T. We can thus solve for the velocity to find the emplacement time of a block. Stokes flow defines the drag force felt by a sphere descending into a viscous fluid as

$$F_V = 6\pi\eta R v\,, \qquad (2.5)$$

where η is the viscosity and v is the sphere's descent velocity into the fluid (e.g., Turcotte and Schubert, 2014). While this law does not technically apply to a cylinder, we can use Oseen's approximation (Proudman and Pearson, 1957) to argue that the solution only differs by a geometric correction factor of order unity for a low Reynolds number Re. For the slow, laminar flow we are modeling, $Re \approx 10^{-9}$, so Oseen's approximation is valid. Thus, equating F_T to F_V, we can rearrange the equation to estimate the block's downhill velocity v:

$$v = \frac{(\rho_l H - \Delta\rho T)gR}{6\eta} * (\sin\theta - \mu\cos\theta)\,. \qquad (2.6)$$

Figure 2.3 presents a diagram of emplacement velocity as a function of basin slope for a range of plausible coefficients of friction μ. We also scale this velocity to a generalized "emplacement time," the time it would take a block moving at this constant velocity to travel 100 km. The curves truncate with endcaps where slopes become so shallow that the net downhill

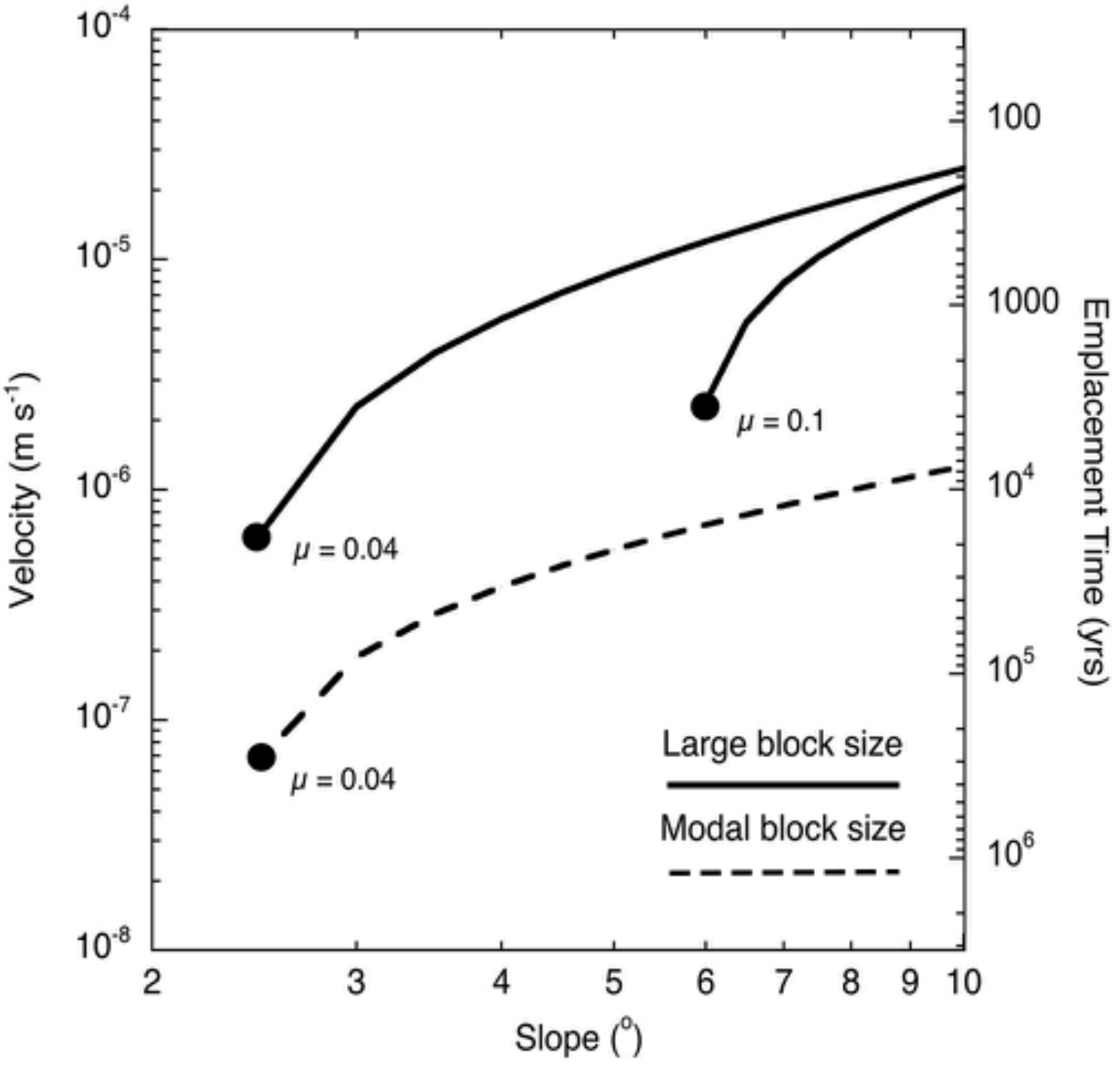

Figure 2.3. Emplacement velocity as a function of basin slope for representative coefficients of friction μ. Curves are scaled to a notional emplacement time, representing the time it would take a block moving at constant velocity to travel 100 km. For a particular value of μ, there is a critical value of slope (indicated by circles) below which friction exceeds the downslope weight of the block and velocity drops to zero. Large (R = 20 km) blocks show higher velocities and shorter emplacement times than modal (R = 10 km) blocks owing to their greater resistance to viscous drag.

force no longer is positive. We observe that in the most favorable case for mobilization ($\mu = 0.04$, slope < 5°, nitrogen viscosity $\eta = 10^{12}$ [as determined from our convection calculations]), a large block could be emplaced on a timescale of hundreds of years. Note that while the viscosity of the nitrogen is an imprecisely known parameter, it is a linear factor in the velocity equation. Higher estimates of the viscosity would change these timescales by the same order of magnitude as the viscosity and would still result in geologically short emplacement times.

2.4 **Discussion**

We find that there is a critical value of slope for a given μ value. Below this value, the friction from the weight of the block will be sufficient to overcome the net downslope force, resulting in no mobilization. For a μ of 0.04, a slope of ~2.5° is required to mobilize a block. Increasing the μ value to 0.1 requires a slope of 6° or more for mobilization. It is geologically unlikely that the rim of the Sputnik Planitia basin can support a slope of 6° or more over the observed emplacement distance (cf. Rosenburg et al., 2011; Tanaka et al., 2002). This observation has two important implications.

First, for the mountain blocks to be moving under their own weight, they must be relatively well lubricated. Water ice-on-water ice has a coefficient of friction between 0.29 to 0.76 below 100 K (Schulson and Fortt, 2012). These values are in the boundary or mixed regimes, where the boundary surfaces are supported at least partially by the asperities on their contact surfaces, and do not allow for mobility in our model. A μ value of 0.04 requires the blocks to be in the full-film or hydrodynamic regime, where the boundary surfaces are completely supported by the lubricant (Schipper and Maathuis, 1995). The most likely lubricant on Pluto is solid or semisolid nitrogen, infiltrated along fractures in the same process that initially separated the blocks (Palasantzas and De Hosson, 2004). Wetted terrestrial glacial beds show a μ between 0.05 and 0.08 (Cohen et al., 2005), suggesting our critical value is plausible for this mechanism.

Second, the strong dependence on slope provides a natural limiting mechanism on movement. As the basin shallows inward, the slope will eventually drop below the critical value required for movement at a particular value of μ (see Fig. 2.3). Once a block reaches this point, it cannot be moved farther inward unless the lubrication regime changes, the nitrogen surrounding it increases in depth, or it is pushed by a sufficiently large force (such as more blocks descending downslope behind it). Thus, while the emplacement times are geologically short, we cannot say that the emplacement was necessarily a recent event.

We further note that the viscous drag force F_V scales with the radius of the block. Larger blocks resist the drag more than smaller blocks, creating a size-filtering effect where larger blocks have higher emplacement velocities than smaller blocks. This is a common finding of Stokes flow (e.g., Turcotte and Schubert, 2014) and can be observed in the way that gravel will settle out of a water column faster than sand. We illustrate this in Fig. 2.3 with a curve calculated for a modal block (R=10 km) with intermediate coefficient of friction μ. This finding may play a role in the radial distribution of mountain blocks. Qualitatively, the largest mountain blocks appear to be located radially inward of the basin, compared to the smallest fragments that are either located radially outward or interstitially to larger blocks (see Fig. 2.1). This finding may be a result of the size-filtering effect imposed on the blocks by the nitrogen during their emplacement.

2.4.1 **Planetary analogues**

The emplacement mechanism we describe here is not without precedent in the Solar System. One terrestrial analogue is in the “sailing stones” of Racetrack Playa. These boulder-sized rocks have been observed to slide along their base during winter, when the playa is inundated with water. In this case, the driving force is provided by the fetch of wind along the thin layer of ice that encases the boulders. Buoyant force provided by the ice and lubrication along the boulders’ bases allow them to slide, leaving their famous tracks in the mud after the water evaporates (Lorenz et al., 2011).

Other analogues are apparent in large, base-lubricated, shallow slope mass movements seen throughout the Solar System, such as the Heart Mountain slide in Wyoming (Malone et al., 2017) or the large runout landslides along the walls of Mars’s Valles Marineris (Watkins et al., 2015). Figure 2.4 shows the collapse of the Larsen B ice shelf in Antarctica, which exhibits a remarkably similar block size distribution to Pluto’s mountain blocks (cf. Fig. 2.1). We posit that this is due to the viscous resistance that the seawater imposes on the icebergs, resulting in a similar size-filtering effect. Comparative analysis between these features and Pluto’s mountain blocks should proceed with caution, however, as they occurred in different dynamical regimes (e.g., long runout landslides are inherently inertial).

2.5 **Conclusions**

Figure 2.5 presents a schematic cross-section across the chaotic mountain blocks that illustrates our suggested formation mechanism. Condensed nitrogen plays an important role in the formation of the ranges, first by promoting fracture of chunks of uplands crust, and then by providing basal lubrication that allows large, grounded blocks to slide down the shallow regional slope over hundreds of kilometers. Blocks small enough to become buoyant can be carried deeper into the basin by convective forces. Larger blocks can travel until the basin slope shallows sufficiently for friction to arrest their movement but are too heavy to be mobilized by the buoyant or convective forces of nitrogen acting on them. Emplacement times are controlled by the viscous resistance of the nitrogen, which is a function of block size, and range from hundreds to tens of thousands of years.

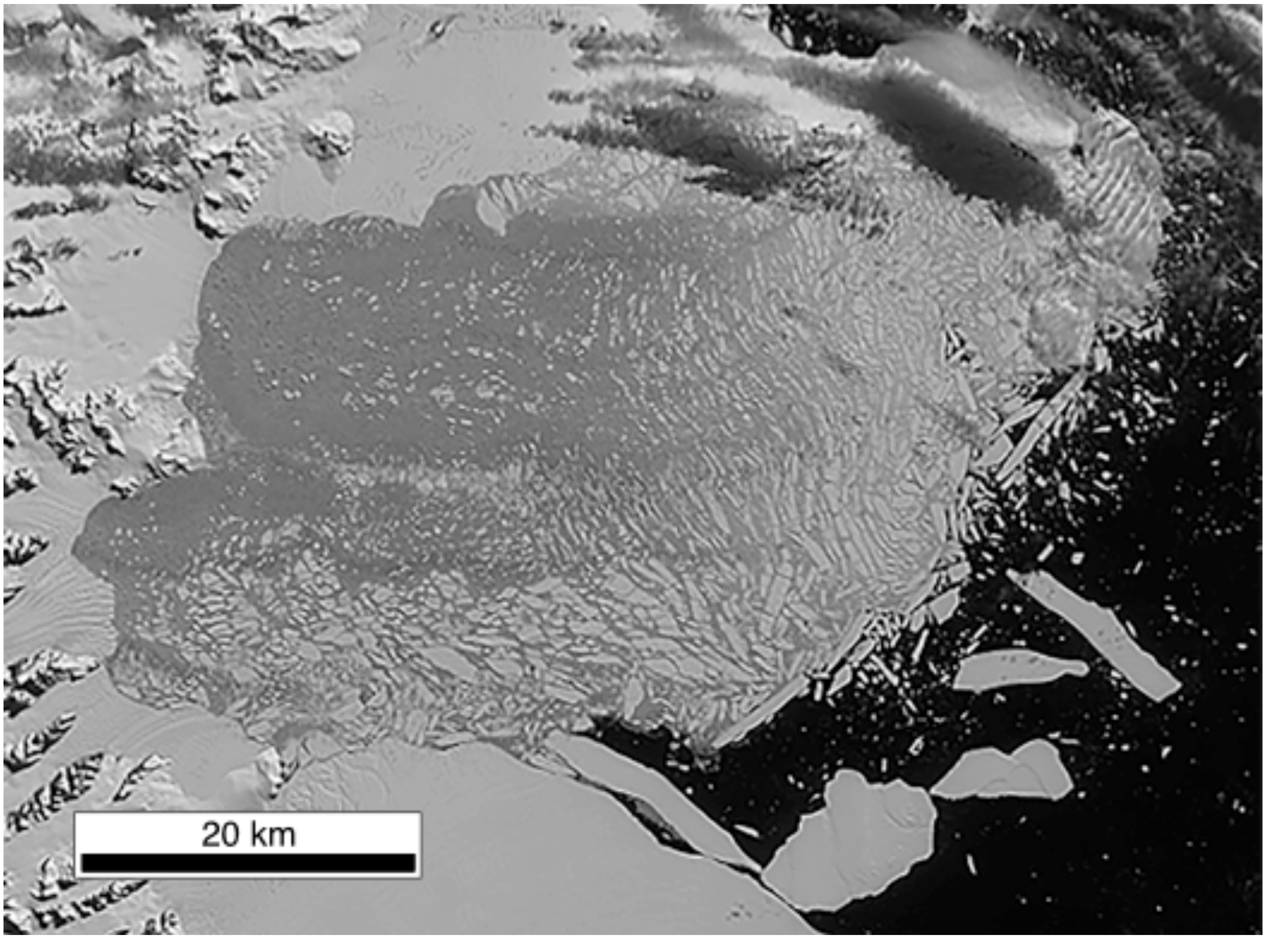

Figure 2.4. Terra MISR image of the northern section of Antarctica's Larsen B ice shelf during its collapse in March 2002. Note the size filtering of the blocks, with larger blocks located more distal to the broad region of smaller blocks between the collapse front and the detachment. Image modified from NASA Photojournal image PIA03702, courtesy of NAA/GSFC/LaRC/JPL, MISR Team.

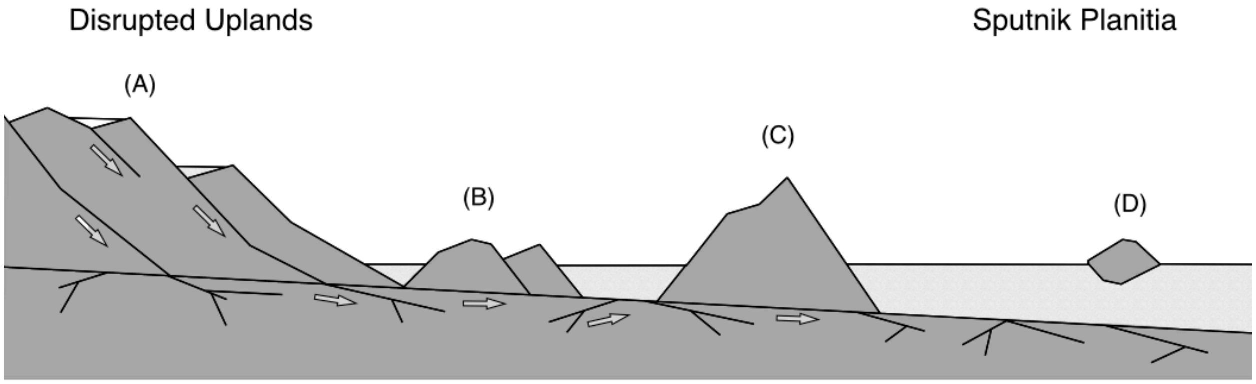

Figure 2.5. Schematic cross-section across Pluto's chaotic mountain blocks. Nitrogen condensing and pooling in the uplands (4a) sinks into the water-ice bedrock, promoting fracture along pre-existing faults (arrows). Calved blocks (4b), lubricated by nitrogen, slide downslope into the basin. Larger blocks (4c) resist the viscous drag of the nitrogen present in the basin, allowing them to slide until the slope descends below a critical value. Smaller blocks and debris (4d) are fully buoyant and can be transported by convection within the nitrogen glacier. Diagram is not to scale.

3. SIMULATING FORMATION OF TRITON'S CANTALOUPE TERRAIN BY COMPOSITIONAL DIAPIRS

3.1 **Abstract**

We present results from numerical simulations designed to test the compositional diapir formation hypothesis for Triton's cantaloupe terrain as originally formulated by Schenk and Jackson (1993). These simulations utilize an updated understanding of Triton's internal heat state from Nimmo and Spencer (2014), as well as more realistic material rheologies, to determine the spatial and temporal extent of deformation within a dense, insulating layer of ice (ammonia dihydrate or CO_2) overlying a water ice crust. We find that density-driven diapiric overturn as suggested by Schenk and Jackson (1993) is inhibited by strong viscosity contrasts in the near surface caused by a high geothermal gradient and temperature-dependent material rheologies. Because the denser ices are markedly weaker than water ice at the same temperature, any model configuration where the water ice was upwardly mobile resulted in a collapse and thinning of the upper layer. We conclude that cantaloupe terrain could not have been formed by compositional diapirs given our current understanding of the likely materials involved.

For a small parameter space (~10 km thick upper layer of CO_2), localized downwelling "sinks" of CO_2 were observed to concentrate stresses within the upper rigid parts of the layer on a similar scale to cantaloupe terrain *cavi*. These stress localizations may result in surface fracture and enhanced sublimation of volatile ices intermingled with the substrate, forming *cavi* by a process of scarp retreat.

The presence of an insulating layer of non-water ice on its surface has a strong effect on the convection regime within the mantle, resulting in more organized convection cells and

localized high temperatures. Time-varying convection results in movement of the cores of individual cells, resurfacing a larger surface area as individual regions are pushed above or below the critical temperature for CO_2 downwellings to form.

3.2 **Introduction**

3.2.1 **Geologic context**

Triton is the only large moon of Neptune and stands as a unique member of the solar system's icy satellites. Its orbit is retrograde and highly inclined, which combined with a distinct lack of other large satellites in the system suggest that it is a captured Kuiper Belt Object (McKinnon and Kirk, 2007). It thus likely shares a formation environment with its slightly smaller cousin Pluto, although the two bodies have experienced very different dynamical environments. The 1989 encounter by *Voyager 2* imaged approximately 40% of Triton's surface and identified many features consistent with a young, geologically active body, including smooth plains deposits associated with cryovolcanism; active plumes in the south polar cap; and a large extent of enigmatic terrain dubbed "cantaloupe terrain" after its dimpled appearance (Fig. 3.1; Croft et al., 1995).

Cantaloupe terrain is characterized by regularly-spaced, semi-circular depressions separated by irregular sub-parallel ridges. The terrain covers ~40% of the imaged surface of the satellite and appears to extend into the far hemisphere not visible to the Voyager 2 spacecraft (Fig. 3.2). Individual depressions, or *cavi*, average between 25 and 35 km in length and appear to have a complex interior morphology consisting of a rough center, a smooth inner annulus, and pitted ridges surrounding the depression (Croft et al., 1995). While detailed elevation data is difficult to interpret due to image quality and the complex (Kay et al., 2019), the ridges appear to

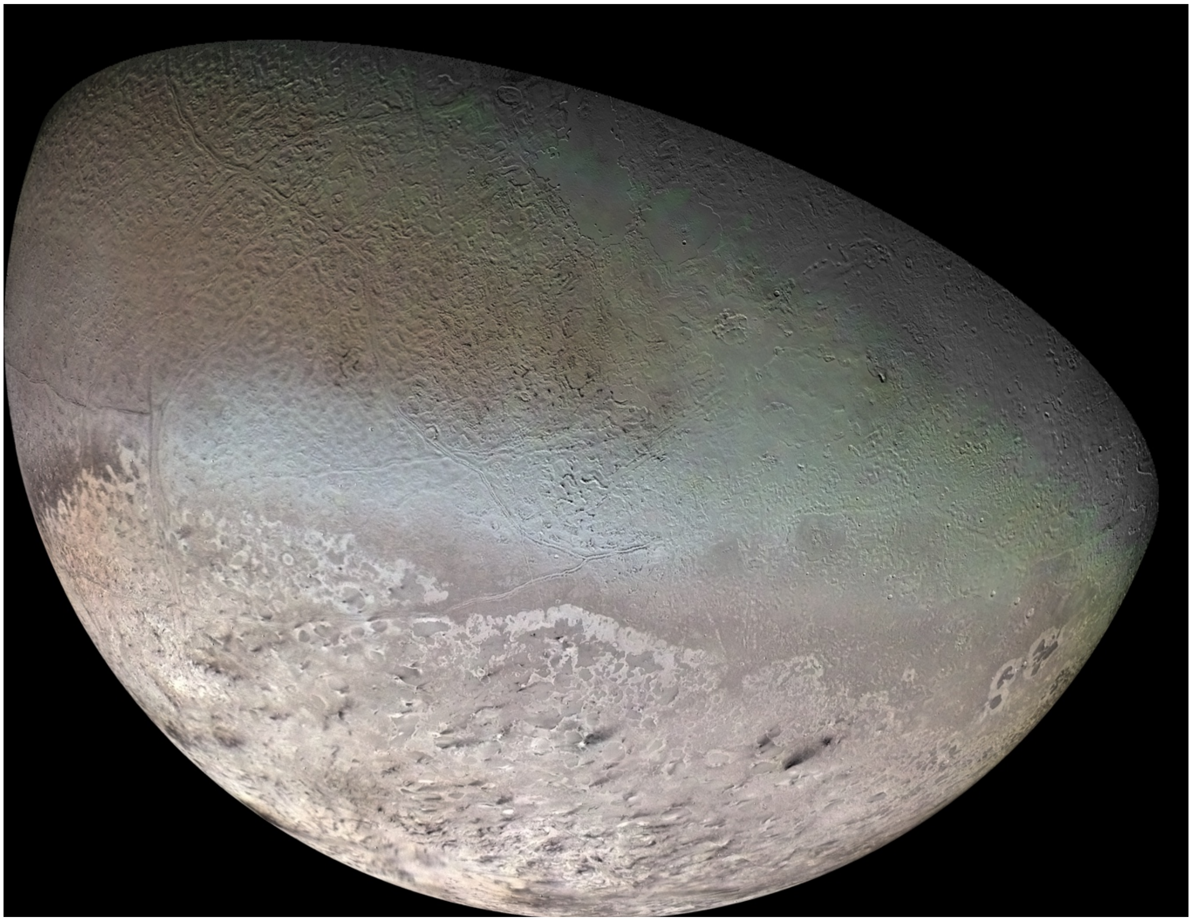

Figure 3.1. Voyager 2 global color mosaic of Triton. Cantaloupe terrain is the dark, pitted region in the northwest of the imaged hemisphere. The difference in terrain morphology between cantaloupe terrain and the potentially cryovolcanic smooth plains deposits in the northeast is distinct. The bright southern cap is likely nitrogen ice and shows the dark blotches of plume activity. Global fracture features known as *sulci* crosscut through all imaged terrains. At this scale Triton's lack of impact craters, and thus young surface, is apparent. Image courtesy NASA/JPL.

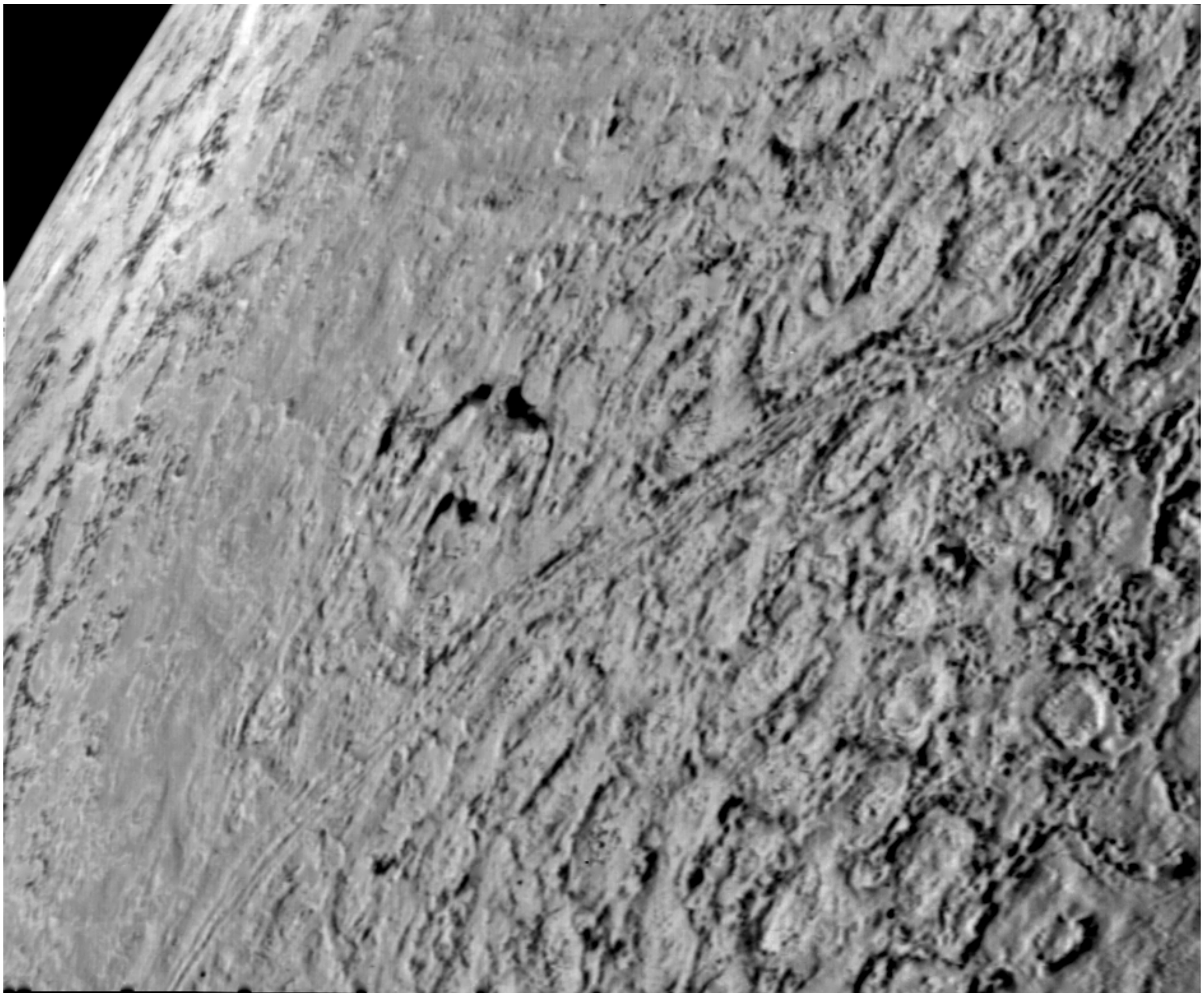

Figure 3.2. Voyager 2 image of cantaloupe terrain at closest approach. The image was taken from a distance of 40,000 km and is approximately 220 km across. North is to the right of the frame. Effective resolution of surface features is 750 m. The oval features in the bottom right of the image are *cavi*. Each is approximately 25-35 km in diameter. Note the interfering relationships between the pitted ridges separating out individual *cavi*. Image courtesy NASA/JPL.

be of the order of a kilometer in elevation. Cantaloupe terrain is crossed by large extensional faults named *sulci* and is flooded to the east by potentially cryovolcanic smooth plains deposits, indicating that it is the stratigraphically oldest known terrain on Triton. Despite its relative age, almost no confirmed craters have been identified within the terrain, suggesting a youthful absolute age.

Schenk and Jackson (1993) interpreted the complex interior morphology and interfering (rather than cross-cutting) relationship of the *cavi* as similar to the surface expression of terrestrial salt diapirs. Terrestrial diapirs form when a layer of salt (which is relatively incompressible and flows readily under low strain rates) becomes buried under higher density sedimentary rock. This configuration is gravitationally unstable, and a Raleigh-Taylor instability can develop if the layer is perturbed by differential loading (Turcotte and Schubert, 2014). The underlying material will rise in an ascending plume, characteristically (although not always) forming a thin stalk with a large, pancake-shaped plume head that displaces the overlying material. When several diapirs form in close proximity, the plume heads form a composite diapir canopy, which expresses on the surface as a series of interfering elliptical features with internal cellular structure related to the original stratigraphy (Jackson, 1990). The spacing of the plumes within a canopy is related to the original thickness h of the layer by (Turcotte and Schubert, 2014):

$$\lambda \approx 2.6\, h \qquad (3.1)$$

For the hypothesized diapir canopies on Triton, Schenk and Jackson (1993) estimated a plume separation wavelength of 54 km and a corresponding overlying layer thickness of ~20 km. The first-order structure and evolution model implied by these estimates is of a layered structure with the less dense material (presumably water ice, the major component of Triton's ice shell) on the bottom, overlain by a ~20 km thick overburden of two distinct, denser ice phases (see Fig. 3.3).

The compositions of the dense ice phases participating in the hypothesized diapiric overturn are uncertain, but the most likely components (and the ones investigated by Schenk and Jackson [1993]) are water ice (H_2O), carbon dioxide (CO_2), and ammonia dihydrate (NH_3-$2H_2O$; hereafter abbreviated as ADH). These ices have either been telescopically identified on Triton's surface (e.g., Quirico et al., 1999) or inferred from geochemical modeling (e.g., Shock et al., 1993), and maintain enough rheological strength at Triton surface conditions to participate in geologic activity. The other ices identified on Triton - methane (CH_4), nitrogen (N_2), and carbon monoxide (CO) - are relatively volatile at Triton pressure/temperature conditions. They are also weak Van der Waals bonded solids that would not be able to support the observed topography over geologic time (Yamashita et al., 2010).

Based on an inferred surface age of ~500 Myr, Schenk and Jackson (1993) estimated a conservative ascent time for these putative diapirs as ~1 Gyr. Depending on the materials involved, this ascent time gives a mean viscosity of the overturning layers between 10^{21}-10^{22} Pa s, down to as low as 10^{20} Pa s if the formation time is as low as 10 Myr.

Thermal convection is one method of potentially enhancing diapirism (Talbot, 1978). While Schenk and Jackson (1993) did consider the possible effects of thermal convection in enhancing diapir formation, they concluded that density contrasts from the differing materials would be the dominating force in a Triton diapir due to the low (<10 K/km) geothermal gradient. This conclusion was based on the assumption at the time that Triton's interior was frozen and warmed only by relict radiogenic heat flux.

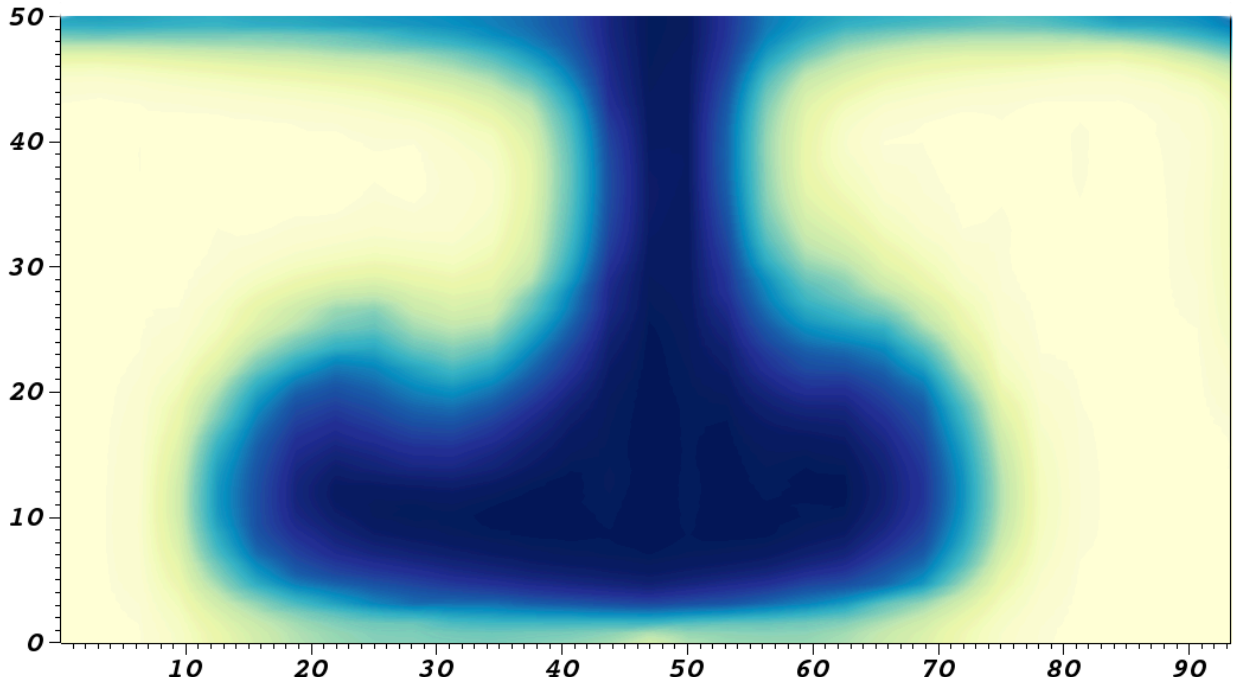

Figure 3.3. ASPECT simulation output for a model re-creating the compositional diapir model as formulated by Schenk and Jackson (1993).[1] A dense ($\Delta\rho = 640$ kg/m^3) overlying layer, originally 20 km thick, sinks, and the less dense underlying material rises in a diapir. Material rheologies are uniform and have a viscosity of 3x10^{22} Pa *s, equivalent to an overturn time of ~1 Gyr.

3.2.2 **Challenges to the diapir hypothesis**

More recent observations and modeling have raised challenges with the diapir hypothesis outlined above. Updated material data for the ices involved show that their temperature-dependent rheologies may not be effectively captured by such a simple Rayleigh-Taylor instability analysis (e.g. Yamashita 1997; Durham et al., 2010; Durham et al., 1999; Cross et al., 2019). Recent studies (Nimmo and Spencer, 2014; Ruiz, 2003) show that Triton's subsurface heat flow may have considerable non-radiogenic components, and the thermal conductivities of CO_2 and ADH are significantly lower than that of water ice, potentially increasing the effective subsurface temperature (e.g., Schurmeier and Dombard, 2018). The surface age of Triton has also been revised downward, to as low as 10 Myr, challenging the relative age of formation for the putative diapirs (Schenk and Zahnle, 2007).

While advantageous in that it simplifies the considerable uncertainties involved, Rayleigh-Taylor instability analysis is a parametric method that specifies a constant bulk viscosity between the components and across the model space. In reality, the effective viscosity of the ices involved is highly temperature dependent and variable between the components (e.g. Durham et al. 2010; Durham et al., 1993; Cross et al., 2019). For these materials to participate in near-surface viscous flow (as implied if cantaloupe terrain is the surface expressions of diapirs), the subsurface temperature must be warm enough to lower the viscosity sufficiently to permit movement. If one component becomes too stiff to move viscously in the near surface, then it is difficult to imagine a scenario where subsurface diapirs are exposed at the planetary scale of the cantaloupe terrain.

Raising the internal heat flow of Triton sufficiently to permit near-surface deformation is a challenge in the absence of tidal heating. Depending on estimates of its chondritic composition, Triton's present-day heating rate varies between 1 and 6 mW m^{-2}, not nearly enough to power geologic activity (Robuchon and Nimmo 2011; Gaeman et al., 2012; Brown et al., 1990). Triton is likely to be a thermally processed body due to the circularization of its orbit after capture by Neptune, which produced enough heat to melt completely the satellite (Ross and Schubert, 1990). However, Triton's small size means that it is unlikely to have retained this heat over the age of the Solar System (Nimmo and Spencer, 2014).

Triton has a negligible eccentricity, so eccentricity tidal heating is not an effective heating mechanism. However, recent analysis by Nimmo and Spencer (2014) showed that ongoing heat from ocean tidal dissipation of Triton's orbital obliquity may be sufficient to maintain an internal ocean of ~240 K, assuming the presence of ammonia to serve as an antifreeze. They show that the resulting surface heat flux of between 7 and 18 mW m^{-2} would be sufficient to drive convective motion in the ice shell, and consequently bring lower viscosity ices closer to the surface. This effect could be enhanced by the insulating effect of the surface ices. ADH has a thermal conductivity of 1.5 W m^{-1} K^{-1} at Triton surface conditions, half that of water ice, while CO_2 is a full order of magnitude lower (Ross and Kargel, 1998). These lower conductivities would reduce the amount of heat required to mobilize the diapir components.

In addition, Schenk and Zahnle (2007) reexamined the cratering record of Triton and concluded that the majority of its craters were from planetocentric debris, not heliocentric impactors. Accordingly, they estimated the average surface age as ~10 Myr old. Such a young surface age presents challenges for the diapir hypothesis as stated because diapirism is not a recurring process. If we assume that the Schenk and Jackson (1993) model is correct and the diapirs took ~1 Gyr to form, then it would be fortuitous that they were exposed in the last 10 Myr. Conversely, if the viscosities of the ices are radically different from those specified in Schenk and Jackson (1993) and they formed faster, then what process is responsible for creating the required density inversion within the last 10 Myrs, and has it been active previously in Triton's history?

Emplacement of ADH layers on the surface via effusive cryovolcanism is one potential way for multiple generations of diapirs to form. Regenerating layers of CO_2 is a less straightforward process, as its most likely source in recent geologic time is from atmospheric photochemical production (Krasnopolsky, 2012; Shock and McKinnon, 1993). In either case, it is not clear how the timescales of these processes line up with the formation time of the hypothesized diapirs. Better constraints on the likely formation time of the diapirs would help determine to what extent we need to appeal to these resurfacing mechanisms.

The goal of this work is to re-address the diapir formation hypothesis for cantaloupe terrain accounting for these new developments in our understanding of Triton. To do so, we go beyond the Rayleigh-Taylor scaling arguments of Schenk and Jackson (1993) and utilize numerical simulations to model the interior heat of Triton throughout its ice shell. We then use those boundary conditions to create a compositional model replicating the first order stratigraphy of the Schenk and Jackson (1993) diapir hypothesis, updated with the temperature-dependent rheologies of the materials involved.

3.3 **Methodology**

To model numerically the flow of material within Triton's near-surface, we implement a material model for ice in the finite element code ASPECT (Advanced Solver for Problems in Earth's ConvecTion; Kronbichler et al., 2012). ASPECT solves the Navier-Stokes equations describing thermally driven convection using the Boussinesq (incompressible) approximation and accounting for the conservation of mass and energy. ASPECT utilizes modern numerical methods such as adaptive mesh refinement, nonlinear solvers, and the advection of compositional fields with differing physical properties from the background material.

We model the effective viscosity of the ices under consideration as that of Newtonian (non-stress dependent), temperature-dependent fluids using a formulation based on that of Nimmo and Spencer (2014):

$$\eta = \frac{\eta_0 E_A}{R} * \left(\frac{1}{T} - \frac{1}{T_0}\right), \qquad (3.2)$$

where η_0 is the viscosity at the reference temperature T_0, E_A is the activation temperature, and R is the gas constant. This relationship utilizes the fact that the various creep mechanisms are dominated by thermally activated processes at the low stresses of icy planetary crusts (Durham et al., 2010; Tobie et al., 2003). We neglect the effects of partial melting on viscosity; see below for a discussion on possible melting behavior. Table II contains the relevant material parameters for the ices used in this study.

To incorporate our updated understanding of Triton's internal heat flow, we model the entirety of Triton's ice shell based on the work of Nimmo and Spencer (2014): a 300 km thick convecting ice shell underlain by an ammonia-water ocean at a temperature of 240 K. Above this ice shell we model a thin layer of insulating, putatively diapir-forming ice (ADH or CO_2), as described in Schenk and Jackson (1993). The convecting ice mantle will transfer heat to this thin insulating layer, affecting its viscosity and possibly causing diapir formation or other deformation.

A problem with simulating this domain is the small scale of the expected diapirs versus that of the ice shell itself. To resolve small scale temperature and material changes in the upper layers of the ice shell, we would require a very fine mesh that would significantly increase simulation time. To avoid this problem, we run two suites of simulations: one "regional" (on the scale of cantaloupe terrain's global extent) and one "local" (on the scale of individual diapir fields). The regional simulation then serves to set temperature boundary conditions for the local simulation. The outputs of interest in the regional simulations are the horizontal scale of the upwellings and the resulting temperature structure in the upper 50 km (equivalent to the bottom of the local simulation) of the ice shell. The local simulation is used to investigate the effects of local scale deformation or diapir formation.

TABLE II. MATERIAL PARAMETERS FOR ICES RELEVANT TO TRITON

Parameter	$H_2O_{(s)}$	$NH_3*H_2O_{(s)}$	$CO_{2(s)}$	Sources
Reference temperature T_0 (K)	273	180	176	Nimmo and Spencer (2014); Kargel (1991); Yamashita and Kato (1997)
Density at reference temperature ρ_0 (kg m^{-3})	930	965	1560	Barr and Pappalardo (2004); Croft et al. (1988); Shenck and Jackson (1993)
Viscosity at reference temperature η_0 (Pa s)[a]	1x10^{14}	1x10^{16}	7x10^{9}[b]	Nimmo and Spencer (2014); Durham et al. (1998); Yamashita and Kato (1997)
Thermal conductivity k (W m K^{-1})	3	1.2	0.3	Klinger (1980); Desch et al. (2009); Ross and Kargel (1998)
Thermal expansion coefficient α (K^{-1})	5.5x10^{-5}	2.81x10^{-5}	2x10^{-5}	Butkovich (1959); Fortes et al. (2003); Jinjin et al. (2015)
Activation energy E_A (kJ mol^{-1})	60	107.5	80	Nimmo and Spencer (2014); Durham et al. (1998); Durham et al. (1998)
Specific heat Cp (J kg^{-1} K^{-1})	2110	1971	1240	Desch et al. (2009); Durham et al. (1998); Giaque and Egan (1937)

[a] See equation 3.2.

[b] For numerical stability purposes, the lowest effective viscosity was limited to 1x10^{14} Pa s.

3.3.1 **Regional simulations**

The regional-scale simulation domain is a Cartesian box model 300 km deep by 1500 km wide (effective cell resolution 5 km x 20 km). Following the work of Nimmo and Spencer (2014), we assume that heating from obliquity tides is sufficient to maintain an ammonia-rich ocean at this depth. The bottom boundary is set to 240 K, the average temperature of the proposed ocean, and the top to Triton's average surface temperature of 40 K.

We assume that the majority of Triton's crust is relatively pure water ice, and that the bulk of its ammonia has been either concentrated in the ocean or in layers or pockets of near-surface ADH. This inference is supported by modeling from Hammond et al. (2019), who suggested that while ammonia rich ice pockets may become trapped in the upper 5-10 km of Triton's crust, the majority of the ammonia will concentrate in the ocean and leave the rest of the crust as pure water ice. Nimmo and Spencer (2014) also suggested that Triton's crust thickness will be self-regulated by concentration of ammonia as the ocean freezes or melts. Because of these inferences and simulation results that the temperature never materially exceeds 240 K, we neglect the effect of partial melting or thinning of the ice shell.

To model the warming effect of the surface layers of ADH or CO_2, we implement a thin, thermally insulating compositional field at the top of the domain. This field differs in material parameters from the mantle ice only in its thermal conductivity. We choose this simplification because the resolution of the regional model is not sufficient to observe the small-scale deformation of cantaloupe terrain; at this scale, we are only interested in its effect on the temperature boundary conditions. Comparison with subsequent local simulation results show that deformation does not occur on a larger scale than the cell resolution of the regional simulations, supporting the rationale for this simplification. Thus, the main free parameter for the regional simulations is the composition and thickness of this upper insulating layer.

Several simulations were also conducted to examine the effects of a thinner ice shell (as thin as 100 km). The wavelength and temperature gradient of the resulting convection cells varied as expected by changes in the thickness of a convective layer (Solomatov, 1995). However, the absolute variation was relatively small compared to the uncertainties of the model. We conclude that the ice shell thickness is not a major contributor to the formation of cantaloupe terrain.

3.3.2 **Local simulations**

We conduct the local scale simulations in a Cartesian box model domain 50 km deep by 250 km wide (effective cell resolution 2 km x 4 km). The upper portion of the domain is designated a compositional field of variable thickness with the full rheologic properties of the material composing the dense upper layer (Table II). The bottom boundary of the domain is set to a range of average temperatures observed from the regional simulations. To promote the formation of a Rayleigh-Taylor instability, we perturb the boundary between the two materials by periodically varying its elevation according to a sine function. Variations in the amplitude and wavelength of the layer boundary function (to enhance the effect of differential loading) did not noticeably affect the wavelength of any resulting instabilities and was neglected in further study. The free parameters in each simulation are thus the upper layer composition, its thickness, and the basal temperature.

Two cases are tested: ADH overlying water ice (Schenk and Jackson [1993] case 1); and CO_2 overlying water ice (Schenk and Jackson [1993] case 2). Preliminary results show that more complicated stratigraphies that combined CO_2 and ADH were largely unstable, as the presence of even a thin layer of CO_2 insulated the system sufficiently to exceed the 176 K melting point of ADH.

3.4 **Results**

3.4.1 **Regional simulation results**

We observe significant variations in the vigor and relative organization of convection related to the thickness and thermal conductivity of the overlying volatile ice layer. To summarize, the presence of even a thin insulating layer results in more organized convection cells, with high temperatures concentrated at the center of the cells. Layers of 10 km thickness and thinner show time-varying convection, with the center of individual cells migrating over ten million-year timescales. Thicker layers showed more pronounced and stable convection cells.

With no insulating layers, the convection is disorganized (Fig. 3.4). Subsurface temperatures are consistent and relatively high (~230 K at 50 km depth). The rigid lithosphere is approximately 20 km thick, and high convective stresses (in the MPa range) are concentrated at its base. This model setup is consistent with the calculations of the brittle-ductile transition depth of Triton by Ruiz (2003), and also of Nimmo and Spencer's (2014) model for mobile lid convection. Our model thus successfully reproduces two previous model predictions, with the caveat that neither of these prior works was addressing cantaloupe terrain or the diapir hypothesis specifically.

Applying a 20 km layer of pure ADH ice (thermal conductivity of 1.5 W m^{-1} K^{-1}; thickness equivalent to the Schenk and Jackson (1993) diapir hypothesis) to the crust results in a considerably more organized convection regime than the baseline non-insulated case (Fig. 3.5). Rather than a disorganized system under a rigid lid, the convection occurs in broad (~500

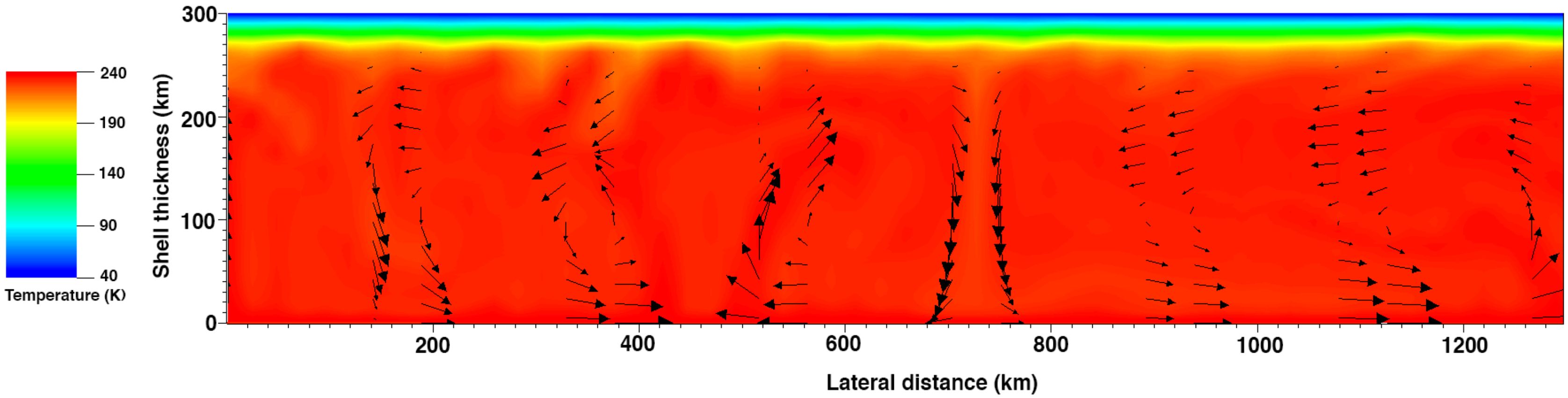

Figure 3.4. ASPECT regional simulation of Triton's crust with no insulating surface layers present. Temperature at the bottom of the domain is 240 K. Convection is disorganized and time-varying. The mechanical lithosphere is approximately 20 km thick.

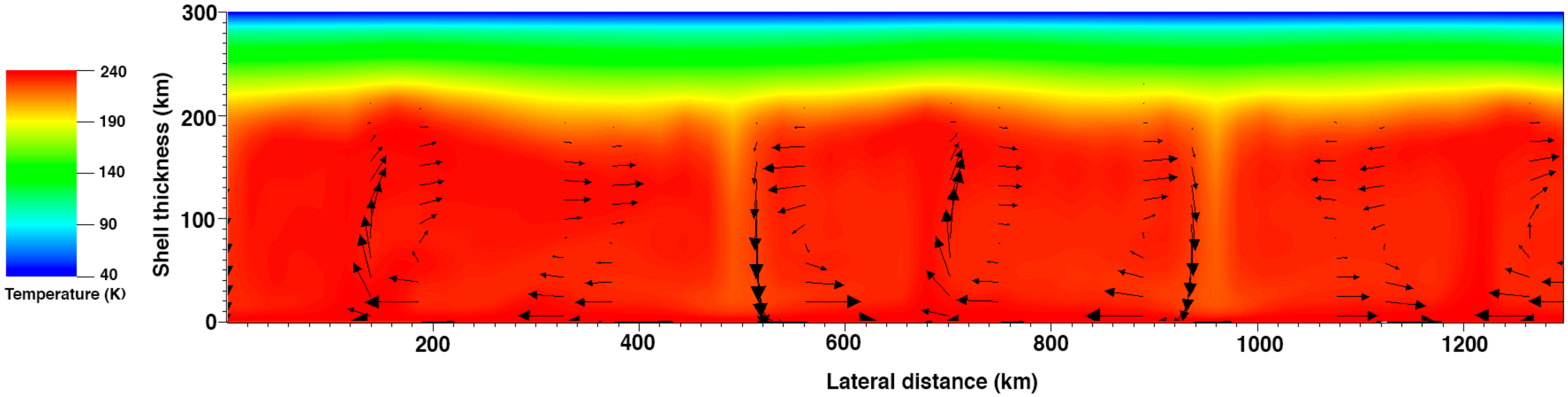

Figure 3.5. ASPECT regional simulation of Triton's crust. The top 20 km of the domain is a compositional field with the thermal conductivity of ammonia dihydrate (NH_3*H_2O, abbreviated ADH). The increased insulation results in a change in the convective regime, leading to more organized convection and lower temperatures in the top ~100 km of the crust.

km width) upwellings and downwellings. The temperature observed in the cores of upwellings at a depth of 50 km is ~160 K. At the same depth in the downwellings, the temperature is 150 K. This finding implies that the addition of the insulating layer changes the convection from a relatively inefficient regime to a more organized regime that is more efficient at passing heat through the ice shell, resulting in lower absolute temperatures despite the presence of an insulating layer.

At the same depth, 20 km of CO_2 ice (thermal conductivity of 0.3 W m^{-1} K^{-1}) allows temperatures to reach a maximum of ~230 K in the cores of upwellings and a minimum of 180 K in downwellings (Fig. 3.6). As in the ADH case, the convection cells are well organized and broad (~700 km width). Note that this minimum temperature is above the eutectic melting point for ADH, and the maximum potentially crosses the phase space of liquid CO_2, indicating that cryovolcanic processes may be active if the crust is this warm.

Progressively thinner insulating layers reduce the vigor and organization of the convection cells. Below ~15 km thickness, the convection becomes time-varying, and the center of the cells begin to oscillate around a central point, varying by ~100 km every 10 Myr (Fig 3.7). This variation, and the corresponding spatial movement of the maximum temperature distribution, has important implications for our hypothesis of the formation of cantaloupe terrain.

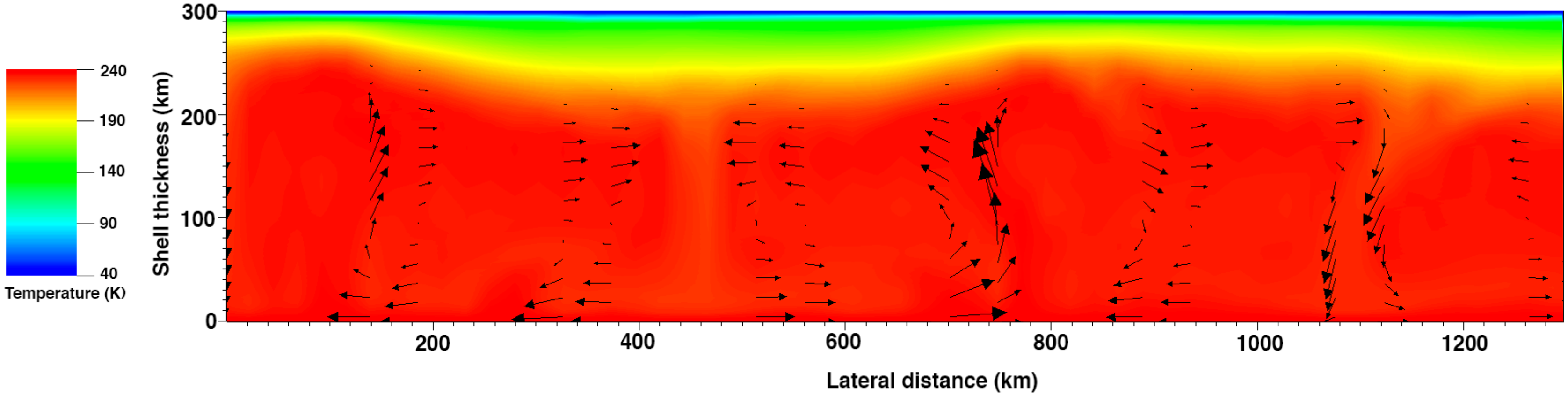

Figure 3.6. ASPECT regional simulation of Triton's crust. The initial conditions are the same as for Figure 3.5 (20 km thick insulating layer), but the compositional field has the thermal conductivity of CO_2. The greater insulating properties results in a similar change in convective regime but higher subsurface temperatures.

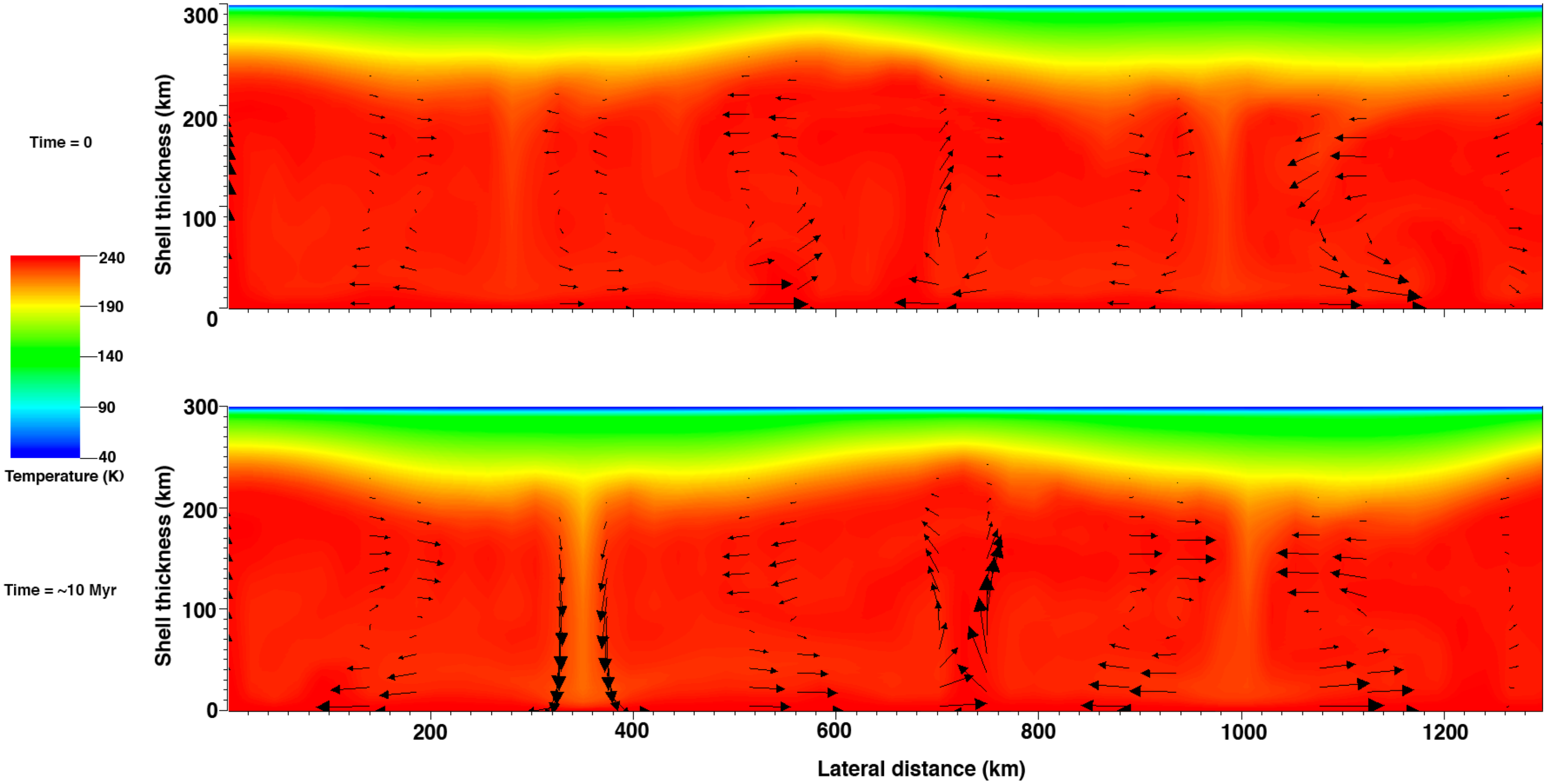

Figure 3.7. Time series of an ASPECT regional simulation with a surface layer compositional field 10 km in thickness and the thermal conductivity of CO_2. Variations in the vigor of convection result in movement of the center of convective upwellings and downwellings, shifting the relative position of the maximum temperatures at depth by ~100 km over 10 Myr.

3.4.2 **Local simulation results**

Three major regimes of deformation are observed to develop, depending on the relationship between the bottom temperature, the composition of the upper layer, and its thickness.

In the first regime, the boundary between the upper and lower layer is stable over the age of the solar system. This regime occurs when the basal temperature is relatively low (< 180 K), the upper, thermally insulating layer is thin (< 5 km), or both. Based on the results of the regional simulation, no model setup involving ADH exceed basal temperatures of 180 K. Accordingly, all simulations involving ADH as the overlying layer fall into this regime, and so we conclude that a layer of pure ADH is unlikely to overturn or deform within the assumptions of our model (Fig. 3.8).

In the second regime, the boundary between the layers diffuses. Fine-scale downwellings of denser, overlying material begin to develop, but no diapiric upwellings or signs of overturn occur. Downwelling material concentrates into narrow (~25 km width) sinks, below which material is slowly mixed with the underlying layer (Fig. 3.9). Vertically oriented stresses above these sinks average ~1 MPa and can exceed 3-4 MPa. Once formed, a sink is stable over the age of the solar system. This regime occurs when the bottom temperature exceeds 180 K, and when the overlying layer is composed of CO_2 between 5 and 15 km in thickness.

In the third regime, the bottom portion of the overlying layer rapidly delaminates, sinks, and mixes with the material below on a timescale of 10-100 kyr. The top 5-10 km of the thinned layer remains rigid, and no stable diapirs develop. This regime develops only when bottom temperatures are high (> 200 K) and the overlying CO_2 layer is thick (20 km or more). It is important to note here that, with the range of temperatures observed from the regional

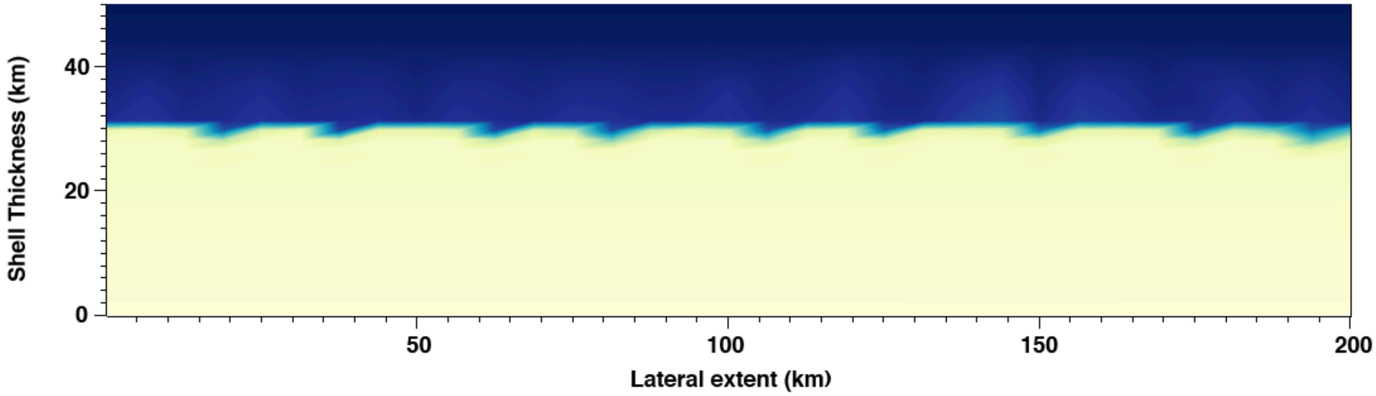

Figure 3.8. ASPECT local scale simulation of a two-layer density model. The top, blue layer is 20 km of ADH; the lower yellow layer is water ice. The bottom temperature of the simulation is 180 K, the maximum temperature observed at this depth and surface composition in the regional simulations. Despite a small amount of diffusion, the simulation's initial conditions remain largely static over the age of the solar system.

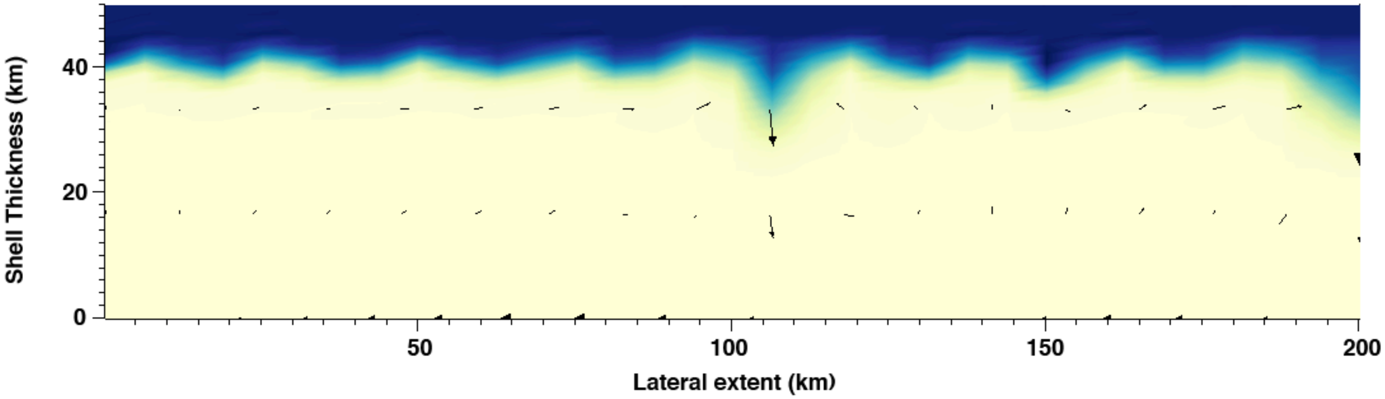

Figure 3.9. ASPECT local scale simulation of a two-layer density model, where the upper layer is 10 km of CO_2. The bottom temperature is 210 K. Two downwellings of CO_2 have developed, and the initial boundary conditions have diffused. There is, however, no upward motion of water ice that would result in diapir formation. This configuration is stable over the age of the solar system.

simulations, 20 km of CO_2 will always exhibit this delamination behavior (Fig. 3.10). This finding indicates that the 20 km layer thickness required for a Rayleigh-Taylor instability to reproduce the observed feature wavelength is not compatible with the stable formation of a diapir, as the layer will collapse and thin on a very short timescale compared to that of the hypothesized diapir formation.

Examining the temperature and related viscosity structure of each of these regimes shows that their behavior is controlled by large viscosity contrasts along the layer boundaries. Fig. 3.11 is a representative example. The geothermal gradient below ~15 km is approximately 5 K/km. The ice below this depth is low viscosity and deformable (viscosities between 10^{17} and 10^{20} Pa s). Water ice along the layer boundary, however, quickly increases in viscosity as temperatures drop below 160 K, up to 10^{24} Pa s - effectively rendering it undeformable. The CO_2 ice at the same depth is deformable, but the increasing slope of the geothermal gradient renders the top 5 km as rigid as the water ice.

3.5 **Discussion**

None of the model setups successfully recreate diapiric upwellings on the scale of cantaloupe terrain as predicted by Schenk and Jackson (1993). Despite the relatively high temperatures produced by mantle convection and thermally insulating upper layers, Triton's extremely cold surface temperature (~40 K) creates a strong geothermal gradient. The effective viscosities within the upper 20 km of lithosphere varied by as many as 5 orders of magnitude due to these thermal effects. In such an environment, the isoviscous flow required by a true Rayleigh-Taylor instability cannot function because upward-flowing material becomes too rigid to move. The compositional differences in rheology effectively prevent the two layers from deforming in a

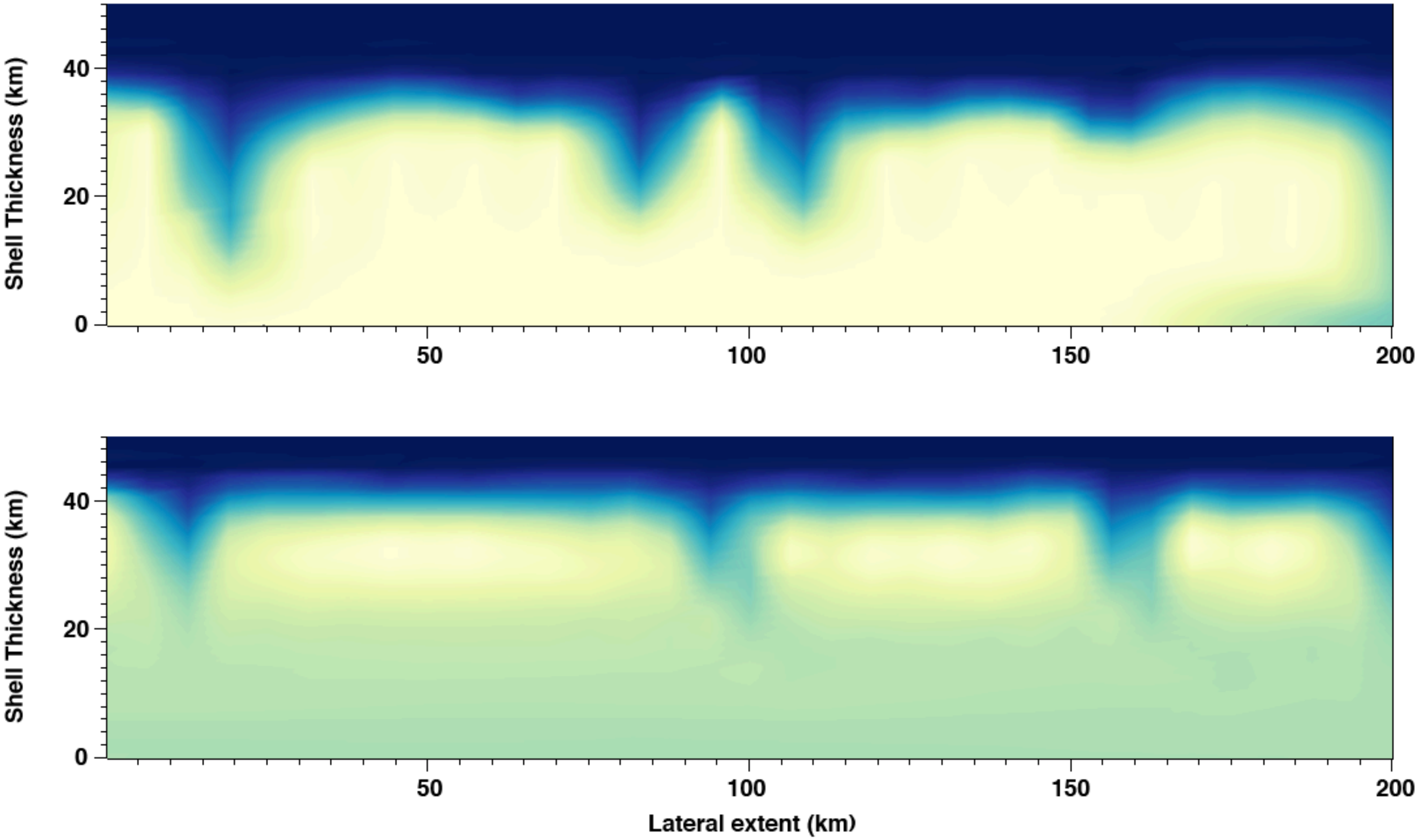

Figure 3.10. Time series of an ASPECT local scale simulation with a two-layer density model. The original top layer is 20 km of CO_2; the bottom temperature is 210 K, the minimum temperature observed at this depth and surface composition in the regional simulations. Within approximately 1 Myr, the surface layer overturns, mixes with the water ice below, and thins. Water ice does not rise buoyantly beyond a depth of 10 km.

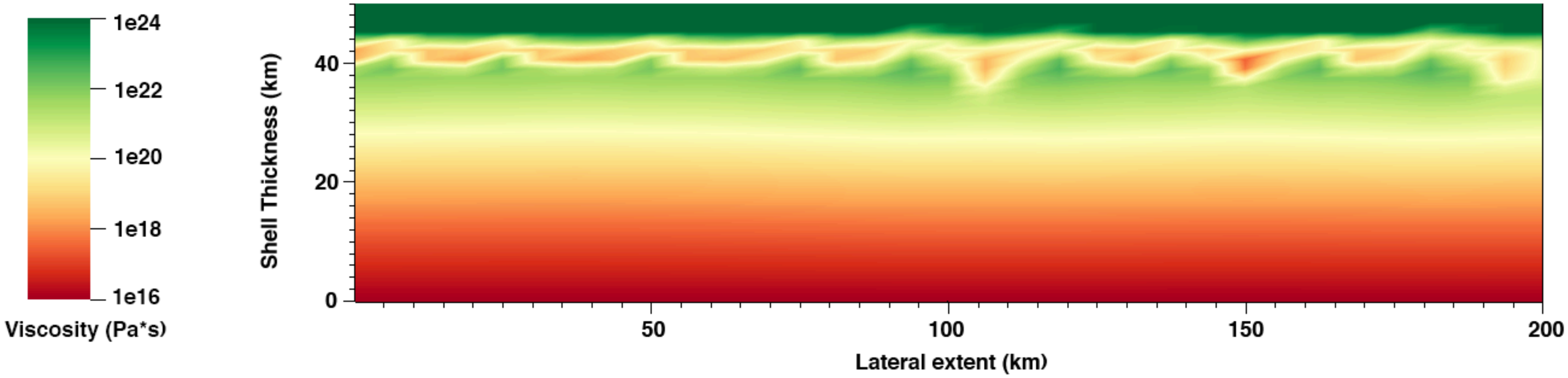

Figure 3.11. Viscosity output for an ASPECT local scale simulation with 10 km of CO_2 and a bottom temperature of 210 K. The periodic low-viscosity zones in the upper 10 km of the domain indicate the bottom of the CO_2 layer. The downwelling “sink” apparent in the density structure of Figure 12 appears here as the somewhat angular low-viscosity zone in the middle of the domain. Water ice above a depth of 15 km is too viscous to deform at the same rate as the very soft CO_2 immediately above it.

co-equal fashion; if the stronger layer is weak enough to deform, the weaker layer is so weak as to collapse and mix with the material below.

We note that although no diapirs breached the surface, the formation of incipient downwellings on the material boundaries is associated with concentrated stresses on the surface of up to ~3-4 MPa (Fig. 3.12). This is well in excess of the yield strength of these ices, implying that even incomplete overturn may substantially fracture the surface at periodic intervals. These fracture zones may subsequently be prone to increased sublimation of volatile ices (CH_4, CO, and N_2) within the substrate, as suggested by Croft et al. (1995) The depressions and ridges of the cavi thus may have formed from advanced sublimation and associated scarp retreat.

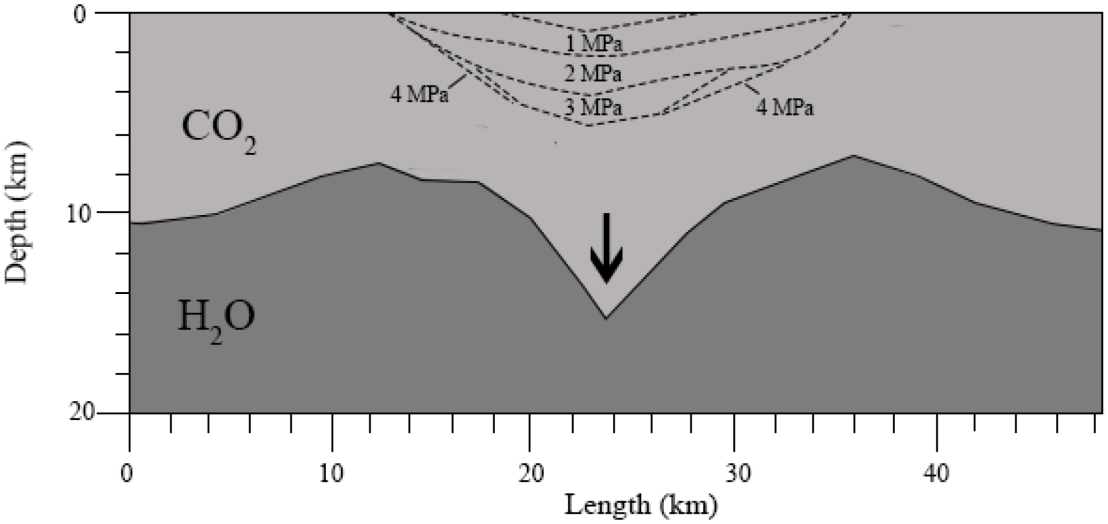

Figure 3.12: Schematic diagram of the stress field overlying the downwelling “sink” from Figures 3.9 and 3.11. Shear stresses up to 4 MPa are concentrated in the mechanically strong portion of the upper CO_2 layer. The lateral area of the stress field (~23 km) is comparable to that of individual *cavi* (25-35 km).

The most favorable case for this formation mechanism is a ~10 km-thick layer of CO_2, which provides the best fit of subsurface temperatures (200-220 K at 50 km depth over a horizontal scale of ~300 km) and rapid localization of stress without total delamination of the layer. A layer of this thickness would also result in time-varying convection deeper in the ice shell, allowing the center of the convection cells to migrate over time and potentially resurface a larger surface area.

We summarize the proposed formation mechanism as follows:

- Solid CO_2 collects on Triton's surface, either condensing from the atmosphere or from endogenic production. This forms a deposit that is likely intermingled with volatiles (CH_4, CO, and N_2) in the form of "ground ice" (Croft, 1995).
- The insulating nature of the ice results in a change in the convection regime deeper in the mantle, centering the core of a mantle convection cell under the deposit. High mantle temperatures become spatially focused beneath the deposit.
- Once the deposit is ~10 km in thickness, temperatures at the bottom of the layer rise sufficiently to the basal viscosity below a critical point. A downwelling sink of CO_2 begins to seep into the deeper mantle, where it mixes with convecting water ice (Fig. 3.9).
- Stress concentration above the downwelling sink fractures the rigid upper layers of the deposit, exposing the intermingled volatile "ground ices" within the upper 1 km to sublimation. Sublimation of these ices results in slope failure and scarp retreat of the CO_2 deposit, forming *cavi*.
- The core of the convection cell shifts away due to time-varying convection, and subsurface temperatures drop below the critical point for the downwelling sink to

operate. *Cavi* formation in the original location ceases, and new *cavi* form above the recentered core of the convection cell.

This formation mechanism may provide an explanation for the youthful appearance of cantaloupe terrain. The formation of the downwelling "sinks" and associated stress that may lead to *cavi* formation is dependent on the focusing of high temperatures by mantle plumes. These plumes, in turn, become more organized due to the presence of the CO_2. Both processes are dependent on the thickness of the CO_2 layer, so the onset of cantaloupe terrain formation is linked to the time it takes for the requisite amount of CO_2 to deposit.

The existence of a 5-10 km thick layer of crystalline CO_2 on Triton's surface would clearly make it a unique body in the solar system and have major implications for its geology. While solid CO_2 deposits likely exist at Mars's polar caps and on Uranus's moon Ariel, these are limited in areal extent and only stable seasonally or within cold traps (Kelly et al., 2006; Sori et al., 2017). Pluto, the only other well studied solar system body on which solid CO_2 is stable, does not have significant deposits, suggesting that its significant (Quirico et al., 1999) presence on Triton is unique to the system.

Recent work by Poppe et al. (2019) suggested that Neptune's gravitational influence may increase the interplanetary dust particle flux into Triton's atmosphere to as much as two orders of magnitude over Pluto's. The oxygen-bearing species in this dust (mostly H_2O) would ablate into the atmosphere and contribute to forming oxidized carbon species, which could hypothetically precipitate to form stable CO_2 deposits. Estimating the time scales for this process, and whether they are compatible with these models, is a clear next step for this research. We intend to utilize the atmospheric mixing models of Wong et al. (2017), adapted to Triton, as a first step to estimate whether this is a realistic deposition mechanism.

While the local simulations showed high stress above the downwelling sinks, ASPECT is fundamentally a convection code and is unable to model the viscoelastic or brittle deformation that would result from this stress. Other finite element codes may be required to simulate properly the effects of this stress, and whether the resulting structures (possibly after modification by sublimation) would be consistent with the observed features of cantaloupe terrain.

A consequence of the observation that downwelling sinks (and thus *cavi*) only form over the center of convection cells is that the age of the *cavi* is linked to the movement of the convection cells. As a cell moves, it should leave a track of older *cavi* in its wake, similar to the volcanic centers left by the movement of tectonic plates over a mantle hot spot on Earth. This age relation is not immediately obvious from the available data. Two hypotheses may explain this discrepancy. First, the only high-resolution images from Voyager 2 that captured cantaloupe terrain show an areal extent of about 240,000 km^2 (Schenk and Jackson, 1993). This is approximately the area covered by the center of a mantle plume. The remainder of cantaloupe terrain is covered by lower resolution images in which the degradation state of individual *cavi* is difficult to assess. Without novel image processing or new data, we cannot assess the age relations of the *cavi* by this method.

It is also possible that convection plumes may be interfering with one another, obscuring the track of individual plumes. The simulations here were conducted in a 2-dimensional space which did not account for this. In a 3-dimensional space, it is possible that time-varying convection plumes could interfere with one another and change the areal extent of the proposed resurfacing mechanism. ASPECT is a dimensionally agnostic code, making 3-dimensional simulations a natural next step; however, the additional time and computing resources required were not available for this portion of the project.

3.6 **Conclusions**

Numerical simulations of the diapir model of cantaloupe terrain formation do not successfully recreate overturn as predicted by Schenk and Jackson (1993). A high geothermal gradient in the near-surface, differing material rheologies, and the boundary conditions imposed by convection within the ice shell result in large viscosity contrasts that prevent diapir formation. The presence of a dense, rheologically weak, and thermally insulating ice on the surface may, however, promote localized fracture of the surface. Extensive sublimation of surface materials resulting from this fracture may play a role in resurfacing cantaloupe terrain. Estimating the time scales and required extent for such a process is the next phase of this work.

4. GFGA: GRAVITY FROM GENETIC ALGORITHMS, A FORWARD MODELING TOOL FOR PLANETARY SCIENCE

4.1 **Abstract**

I present Gravity from Genetic Algorithms (GFGA), a C++ code designed to forward model the gravitational potential field and related spherical harmonics of triaxial planetary bodies. The code uses genetic algorithms to sort efficiently through the large parameter space inherent to forward modeling, matching the solution to a set of user-defined constraints that can include either model data such as spherical harmonic coefficients of the potential or observations such as bulk mass, shape, or spacecraft acceleration data. The purpose of this code is to model the density structure of planetary interiors from their gravity while bypassing the assumptions required for more common inverse modeling techniques. The code is currently in a preliminary state, with additional bug fixes and features required to be usable by the scientific community. Two test cases for Europa and Ceres are presented that demonstrate the core functionality of the code.

4.2 **Introduction**

Measuring the gravitational potential field outside of a planetary body is a well-understood method of probing its internal structure. Density variations within a body affect the gravitational acceleration of nearby objects, including spacecraft. This can be measured as minute changes in the Doppler shift of radio communications with the spacecraft, resulting in a model of a body’s gravitational potential field at a series of points above the body’s surface. Expanding these gravity models into spherical harmonic coefficients allows us to infer the internal density distribution of the body, and thus constrain its orbital and geologic history (e.g., Turcotte and Schubert, 2014).

The term that is most often dominant in the spherical harmonic expansion is known as the J_2 term, which is a zonal (latitudinal) term that describes a body's gravitational potential due to its oblate shape. For a tidally deformed body, determining the internal distribution of mass additionally requires the C_{22} coefficient. Ideally, these coefficients should be uniquely determined. This task requires a model of the variation in a body's gravitational potential with both latitude and longitude, which in turn can only be constructed from data collected when the observing spacecraft is in a polar orbit. Combining these data with other constraints such as the bulk mass, density, triaxial shape, and rotation of the object allows one to construct a model of the body's internal density structure by a process of inverse modeling (Wieczorek, 2007).

However, in situations where a polar orbit is not possible due to mission design or architecture, it is still possible to separate J_2 and C_{22} (and thus analyze the density structure) by assuming that the internal shape of the body is in hydrostatic equilibrium. This assumption connects J_2 and C_{22} by:

$$\frac{J_2}{C_{22}} = \frac{10}{3} \qquad (4.1)$$

(Anderson et al., 1998). This assumption allows one to inverse model a density structure, but only by constraining the parameters of the model to the hydrostatic assumption. Any spherical harmonic solution for a particular potential field is not unique, so if a non-hydrostatic component is present or if the density structure is unusual, then the inferred interior structure may be inaccurate.

An alternative approach would be to forward model the density structure. This procedure would generate a series of notional density structures, determine the gravitational potential and spherical harmonic coefficients that each would produce, and find the model that best matches the observations (line-of-sight accelerations). This allows unique determination of the J_2 and C_{22} coefficients while bypassing the hydrostatic assumption. However, forward modeling presents a significant challenge due to the non-unique nature of gravity analysis. Many different interior structures may be reasonable fits for the data. There is thus demand for a tool that can efficiently sort through a large parameter space.

One method of exploring a large parameter space is to approach it as an optimization problem. Optimization problems use iterative evaluations of a function, starting from an initial guess, in an attempt to minimize or maximize some value (usually the error) of the function relative to a particular set of criteria. The optimization algorithm proceeds stepwise, with the set of parameters that best approximate the search criteria forming the initial guess of the next step. Many different optimization schema are available, differing in their computational complexity, dimensionality, and response to local vs. global maxima/minima in the function. The choice of which schema to use is based on the relative complexity of the problem, as well as the probability of local minima in the error function (Gershenfeld, 1999).

4.2.1 **Scope of Project**

I have developed an open-source C++ code named GFGA (Gravity from Genetic Algorithms) that can forward model the internal density distribution of a planetary body. GFGA takes as its input a set of shape and gravity observations of the body, as well as a set of user-defined physical parameters. It then generates a population of models with arbitrary density structures and calculates their resulting gravitational potential field and spherical harmonic coefficients. It then utilizes a genetic algorithm (henceforth, GA) to optimize the solution to find the bodies within the population that best match the available observations and defined parameters. The final result is a set of hypothetical interior structures and related spherical harmonic coefficients that could, in principle, bypass an assumption of a hydrostatic internal structure. These results can be used to make predictions for the internal structure of planetary bodies where we lack complete observations of their gravitational potential, as described below in "Applications."

GFGA is currently in a preliminary state and functions as a proof of concept. It can currently reproduce the three-layer structure of Europa described in Anderson et al. (1998) based on the modeled degree-2 coefficients. It also efficiently generates a plausible internal structure and bulk mass for the dwarf planet Ceres based on calculations of its J_2 spherical harmonic coefficient produced from the Dawn mission (Park et al., 2016). Both of these models are described in Preliminary Results. These proof-of-concept case studies utilize spherical harmonic models of Anderson et al. (1998) and Park et al. (2016) to constrain the GA; thus, they implicitly accept the assumptions made by these inverse modeling works. They are presented as preliminary results to showcase the potential of forward modeling by GAs. Future versions of this code will accept direct line-of-sight acceleration data, allowing it to compare forward-modeled potential fields directly to data without *a priori* models of the potential.

4.3 **Methodology**

The current version of GFGA is comprised of three major modules: a gravity potential module, which constructs notional planetary structures and calculates their gravity parameters; the evaluation module, which define the parameters of a successful gravity model; and the GA, which instantiates a population of gravity models, compares each to the results of the evaluation function, and then evolves a new population based on the most fit members of the previous generation. The functioning of each module is described in detail below.

4.3.1 **Gravity potential module**

The gravity potential module is comprised of three sub-modules that run in sequence: a plate model that constructs the notional planetary body, a gravity potential module that calculates the acceleration felt at a series of nodes around that body, and a harmonic function that integrates the potential and expresses the result in terms of its degree-2 spherical harmonic coefficients.

The plate model module is based off the formulation of Kattoum and Dombard (2009). When calculating the gravity potential outside of a mostly spherical body with a radially varying density structure - a good approximation of a planetary body - the classical formulation is to treat each layer of differing density as a hollow shell of some thickness. While logical, these computations are difficult to code numerically. Kattoum and Dombard (2009) varied the method for a uniform body of constant density used by Cheng et al. (2002) by treating the body as a series of overlapping triaxial ellipsoids, each with radii that extend to the center of the body. The effective density of each successively inward body is the density contrast between the actual physical layers. While unphysical in real terms (multiple bodies occupy the same space), this approximation can be shown to produce the same result as the classical formulation. For a hypothetical three-layer body, the potential *U* at a vector position ***x*** outside the body can be given by:

$$U(x) = -\int_0^{V_1} \frac{d^3x' G\rho_1}{|x-x'|} - \int_0^{V_2} \frac{d^3x' G\rho_2}{|x-x'|} - \int_0^{V_3} \frac{d^3x' G\rho_3}{|x-x'|} \,, \qquad (4.2)$$

where V_1, V_2, and V_3 are the volumes of a series of 3 sequentially enclosed bodies, and ρ_1, ρ_2, and ρ_3 represent the density *contrast* between each body and the preceding body that encloses it.

Note that in the case of V_1, this is the contrast between it and the zero-density space outside of the body.

The terms in this equation depend on the density contrast between each successively inward body, not the physical density of the material. I can thus use the triaxial ellipsoidal shape of each overlapping body, and its density contrast with the overlying layer, to calculate the gravitational potential around the entire body. The number of interior bodies can be any arbitrary number, allowing generation of a density gradient at various resolutions (albeit at an increase in computational resources required).

Each ellipsoid is meshed according to a longitude/latitude grid of variable resolution (between 1 and 10 degrees depending on the desired balance of accuracy and solution time). This mesh is then tessellated into a series of triangular plates. Following Cheng et al. (2002), gravity elements are calculated by treating these plates as prisms that extend to the center of the body and integrating along a vector running through the plate's centroid to a point on the reference sphere outside the body. If a center of mass or center of figure offset is present in the body, it is applied as an offset to these vectors relative to the reference sphere. The result is a field representing the gravitational potential at a series of points on the reference sphere. Each point accounts for the gravitational pull on it from every plate-prism within the body.

Mesh points within the polar latitudes ($> 80°$) are excluded from the gravity integration because the small size of individual plates at these latitudes creates degeneracies in the solution over many iterations of the plate model. This is an acceptable approximation as the contribution of these areas to the total potential is small compared to the rest of the body. See the Benchmarking section below for details.

The module then calculates the degree-2 spherical harmonic coefficients based on the generated potential field. The procedure for this portion casts the problem according to Heiskanen and Moritz (1967). The general expression for a spherical harmonic function is:

$$F(\theta,\varphi) = \sum_{l=0}^{\infty}\sum_{m=0}^{l}(C_{lm}P_{lm}(\cos\theta)\cos m\varphi + S_{lm}P_{lm}(\cos\theta)\sin m\varphi), \quad (4.3)$$

where θ is colatitude, φ is longitude, l and m are the degree and order of the spherical harmonic function (indicating the wavelength and orientation on the sphere, respectively), P_{lm} is the respective associated Legendre polynomial as a function of cos θ, and C_{lm} and S_{lm} are the respective spherical harmonic coefficients. Because gravitational potential is a function that varies by colatitude and longitude, I can represent it by a combination of the C_{lm} and S_{lm} coefficients; they are effectively an expression of the density distribution within a body.

In order to derive the C_{lm} and S_{lm} coefficients from my simulated plate model, I must fully normalize the spherical harmonics such that the integrals in their orthogonality relationships equal the area of a unit sphere. If I normalize them like so:

$$\bar{Y}_{l0} = \sqrt{2l+1}\,Y_{lo} \quad (4.4)$$

$$\bar{Y}_{lm} = \sqrt{2(2l+1)\frac{(l-m)!}{(l+m)!}}\,Y_{lm}, \quad (4.5)$$

where

$$Y_{lm} = P_{lm}(\cos\theta)\cos m\varphi \;\text{ or }\; P_{lm}(\cos\theta)\sin m\varphi, \quad (4.6)$$

then I can represent the orthogonality relationship as

$$\iint(\bar{Y}_{lm})^2 d\sigma = 4\pi, \quad (4.7)$$

where integration is around the entire body and

$$d\sigma = \sin\theta\; d\theta\; d\varphi. \quad (4.8)$$

By setting $F(\theta, \varphi)$ equal to my calculated gravitational potential, I can then solve for the fully normalized coefficients like so:

$$\overline{C,S}_{lm} = \frac{1}{4\pi} \iint F(\theta,\varphi)\, \bar{Y}_{lm}(\theta,\varphi)\, d\sigma\ . \tag{4.9}$$

When dealing with rotationally flattened bodies such as planets and larger moons, the degree-2 coefficients dominate the description of the mass and density distribution. Because I am working with triaxial ellipsoids, I only calculate coefficients up to degree-2, although it would be trivial to calculate higher degrees.

4.3.2 **Evaluation function module**

The purpose of the evaluation function is to assign value to individual gravity models based on the constraints of their fixed and derived parameters. The basic standard for a model's value is how close it approaches the observable parameters of the real body. For the full version of the code, this would be line-of-sight spacecraft accelerations, along with bulk mass, density, and shape. For this preliminary version of the code, the most important constraint is the spherical harmonic coefficients derived from previous work.

The default evaluation function prevents the gravity models from assuming unphysical or degenerate configurations. The user can additionally set the model to favor hypothetical configurations, such as possible density inversions or layer configurations. Output from the evaluation function is in the form of a numerical value that is passed to the GA for ranking population members.

There are three basic types of evaluation defined within the module: Boolean functions that add value if true and do not if false; "strict Boolean" functions that add value if true and reset the value to zero if false (effectively removing the model from the viable population); and error functions that add a value defined by the difference between the calculated and true values.

The ellipsoidal parameter functions are all examples of “strict Boolean” evaluation functions. They ensure that each ellipsoid is:

- Internal and wholly contained within the next largest ellipsoid
- The major axis is larger than the minor axis
- The minor axis is not less than 90% of the major axis.

These constraints allow the latitude/longitude mesh to remain consistent and for the Kattoum and Dombard density contrast formulation to remain coherent. If an ellipsoid is larger than its enclosing body, the formulation breaks down, so no model in which this occurs is allowed to have a value greater than 0.

Error function type evaluations are used for observable parameters for the body of interest, such as the degree-2 spherical harmonic coefficients and its bulk mass. These error functions take the form:

$$Value = 1 - |true - calculated|. \qquad (4.10)$$

This result is added to the total value of the model being evaluated. The error function type allows a high degree of discrimination between the value of individual genomes, as a viable solution that is closer to the true value will always be higher valued than solutions that meet the hypothesized criteria but fail to approach (for example) the true value of the mass of the body (Gershenfeld, 1999).

The remainder of the evaluation functions are user-defined to help limit the solutions to those that can physically represent the body of interest. Changing the evaluation functions and the constraints that generate the genome are the two methods by which a user can test different hypotheses of internal structure. For example, a user could define a Boolean evaluation function that adds value to solutions that have increasing density with depth below a defined point. If line-of-sight gravity observations are available, an evaluation function could compare the predicted accelerations at each point to the observed potential. See “Applications,” below, for examples of some of the evaluations defined in my studies of Europa and Ceres.

Definition of the evaluation function is currently only possible by modifying the code itself. Allowing the user to define or add custom evaluation functions is an important part of enabling hypothesis testing and is a high priority for further development of GFGA, as discussed below in “Future Directions.”

4.3.3 **Genetic algorithm module**

The purpose of the GA is to generate and evolve a population of gravity model objects. The members of the population with the highest score from the evaluation function are allowed to continue to the next generation, where portions of their parameter set (henceforth “genome”) are crossed, mutated, and added to a set of newly generated models. The fittest genomes of this new generation are then evaluated, and the process continues until the end conditions are satisfied (usually a certain number of generations; Wall, 1999).

Each genome is a set of parameters unique to that particular gravity model. Each ellipsoid within a body is defined by its major, intermediate, and minor axis, as well as its density (contrast). These four parameters are repeated for each overlapping ellipsoid within the body, the full set of which represents its genome. Each genome within the population has a randomly determined value for these parameters, within limits set by the evaluation function and by its inheritance from previous generations in the run. The derived parameters of interest (e.g., spherical harmonic coefficients, bulk mass and density) are calculated for each individual body and can affect its fitness, but only the dimensions and layer densities are passed on or altered directly by the GA.

When the GA generates a new population, the values of each genome are selected randomly from a set of values within defined ranges. For each planetary body under study, it is necessary to first define the number of ellipsoids (effectively, the number of structural "layers" within the body) and the range of radii and density that each can select from. In the current version, these values are set within the code itself; future versions will be selectable as input parameters.

Because the user must define a conceptual model for the planetary body, the resulting model does not completely bypass *a priori* assumptions. Such *a priori* assumptions are also part of the standard inverse modeling approach. However, preliminary tests have indicated that it is not feasible to attempt to generate a realistic solution in a tractable amount of time without putting limits on the genome parameters themselves. If a population entirely composed of unphysical genomes is generated, the evaluation function would effectively be choosing randomly, and errors would propagate without any improvement in the solution over many generations. Model selection thus plays an important role in evaluating the results of GFGA; the user must design a set of parameters that encompasses the physically possible structures of the planetary body while also confining the results so that a solution can be generated in a reasonable amount of simulation time. I discuss my choices for my Europa and Ceres models in "Preliminary Results" below.

The software for this work utilizes the GAlib genetic algorithm package (Wall, 1999). GAlib is a freely available source code that provides the C++ framework for the construction and evaluation of genetic algorithm codes. GAlib supports overlapping (steady-state) and nonoverlapping (simple) populations, multiple replacement or selection methods, and fully customizable initialization, mutation, and crossover functions, as well as evaluation functions with either population or individual-based evaluation. It forms the basis for both the GA and the evaluation function modules within GFGA.

4.3.4 **Benchmarking**

To verify the core functionality of the gravity potential and spherical harmonic modules, I implement a series of benchmark functions that may be activated to test model output.

The plate model test generates a unit sphere and sets the notional gravity potential to 1. The resulting spherical harmonic functions, integrated over the unit sphere, should be equal to 4π if the plate model is correctly meshing. I find that the accuracy of the resulting solution is dependent on the latitude/longitude mesh resolution. A doubling of resolution from 10x10 degrees to 5x5 degrees resulted in a 10% increase in solution accuracy. I also find that including the polar node points resulted in a decrease in solution accuracy due to degenerate rounding errors due to small plate area. This observation led to my decision to remove the polar node points from the full gravity integration.

The second test investigates the orthogonality relationships of the spherical harmonics (Eq. 4.7). I square the normalized spherical harmonics and integrate around the body based on the area of the unit sphere, ideally equaling 4π. I observe a similar mesh resolution and solution accuracy dependence as in the first test function.

The third test explores the spherical harmonic module’s ability to calculate correctly a potential field. This test function takes as its input a synthetic potential field. I define a specific spherical harmonic and invert for the resulting potential field. I then input a file containing the data points from this field, replacing the regular output of the gravity potential module. The spherical harmonic module then integrates this field and outputs the same harmonic that I originally specified.

The fourth test looks at the gravity potential module by setting a specific density for the body. The resulting potential field is then output, along with the coordinates of a series of arbitrary plate centroids within the body. From these data, it is possible to calculate the potential of that plate as a point mass and compare the result to that of the calculated potential field. This test successfully reproduced the synthetic field input and the associated calculated spherical harmonic coefficient. However, this test was designed to test a single-layer, constant density body. As of the writing, a bug has been discovered related to multi-layer bodies. The significance of this bug is discussed in detail in “Preliminary Results,” below.

4.4 **Applications**

The primary purpose of GFGA is to forward model the internal density structure of a planetary body based on its degree-2 shape and gravity. While currently comparing to spherical harmonic coefficients of the potential model, the approach can be extended to compare directly to the line-of-sight acceleration observations. It can also be utilized to determine if a particular unusual structure such as a density inversion may be present, or to test other hypotheses about its internal degree-2 shape or density structure.

For the first case, the hydrostatic assumption is typically made when gravity observations are unable to separate uniquely the J_2 (zonal) and C_{22} components of its spherical harmonic coefficients, which are required for the inverse modeling approach. This is often the case when a mission is composed of flyby orbits and where a polar orbit of the body is unachievable due to mission architecture. Notably, I too use these gravity coefficients separated by the hydrostatic assumption here, but future versions of the code can bypass this assumption by comparing the predicted gravity models directly to the line-of-sight acceleration observations.

The hydrostatic assumption is required to inverse model the structure of the Galilean satellites of Jupiter, due to the high gravity and challenging radiation environment which has thus far precluded a polar orbit mission architecture (Buffington, 2014). The most interesting of these satellites from a biological perspective is Europa, which has both an ocean underneath its ice shell and a source of tidal heating, thus making it a prime target for astrobiological exploration.

4.4.1 **Europa: context and previous work**

The presence of a subsurface liquid ocean on Europa is attested to by several lines of evidence, including surface features, its bulk density, geochemical modeling, the presence of an induced magnetic field, and its tidal response (Kivelson et al., 2000; Pappalardo et al., 1999; Head et al., 1999; Kargel et al., 2000; Moore and Schubert, 2000). However, the relative thickness of the ice crust to the ocean depth, and the size and composition of Europa's core, can only be weakly constrained by these observations.

Anderson et al. (1998) analyzed Doppler radar measurements from four Galileo flybys of Europa to observe its degree-2 gravitational field, again using the hydrostatic assumption to separate the coefficients. From these coefficients, they estimated that it had a moment of inertia of 0.346 ± 0.005, which combined with its average density and the assumption that its degree 2 gravitational field was dominated by the rotational and tidal forces of its orbit around Jupiter, allowed them to infer its internal density structure. They concluded that Europa was centrally condensed and composed of a shell of ice and/or liquid water 100-200 km thick, a silicate mantle, and a metallic core. Their models were not able to distinguish between solid and liquid water on the basis of density, and the core radius and density was strongly dependent on the composition of the silicate mantle.

In the years since the Galileo encounter, Europa has only increased as a target of scientific interest. The growth in importance of astrobiological research and planetary habitability have led to studies of the interactions between its subsurface ocean, its surface tectonics, its silicate mantle (e.g., Zolotov and Shock, 2003, 2004; Vance et al. 2016; Hand et al., 2007) that may lead to habitable conditions within its ocean.

Key among the remaining questions regarding Europa is the thickness of its ice shell. Transfer of materials between the surface of the ice shell (where surface materials are oxidized by Jupiter's radiation) and its interior (where the processed material could support biochemistry) is a critical parameter in its habitability. If the ice shell is thick, the feasibility of this transfer supporting life may be low.

NASA's planned Europa Clipper mission will dramatically improve the quality of data available to us, including multiple flybys for gravitational observations. However, its planned elliptical encounter profile (Buffington, 2014) will still force analyses to rely on the hydrostatic assumption to calculate Europa's degree-2 gravity coefficients. In addition, the recent descoping of the planned ICEMAG (Interior Characterization of Europa Using Magnetometry) instrument jeopardizes the mission's ability to characterize Europa's internal ocean and thus the thickness of the ice shell (Pappalardo, 2019).

To provide an alternate method of addressing Europa's internal density structure and ice shell thickness, I create a set of model constraints for GFGA and allow it to evolve towards a solution. As a preliminary effort, I attempt to recreate the three-layer model of Anderson et al. (1998), to see if my code is able to recreate their internal structure. My approach can also be extended with the possibility of discriminating between solid and liquid water (and thus ice shell thickness), exploring the existence of a silicate crust on top of a silicate mantle, or the relationship between the core and mantle compositions; however, for this initial work, I focus on simply recreating the previous model.

The constraints applied were the J_2 and C_{22} coefficients modeled by Anderson et al. (1998), Europa's bulk density and mass, and the radius of its major and minor axes (Nimmo et al., 2007). The outer shell and first internal ellipsoid are allowed to have a density between 900-1100 kg m^{-3}, representing various compositions of water ice with a maximum depth of 300 km. The next ellipsoid had a density equivalent to silicate rock (2000-3800 kg m^{-3}) and could extend from 100 km to 1000 km in depth. The third layer varied between dense silicate and a pure iron core (3800-8000 kg m^{-3}) and could extend as far out as 1000 km (.65 R_{Europa}, the hypothetical maximum suggested by Anderson et al. [1998]). The simulation is allowed to run for 1000 generations, with a population of 500 individuals within each generation, for a total of 5×10^5 model evaluations.

4.4.2 **Ceres: context and previous work**

The dwarf planet Ceres is the largest body in the asteroid belt. Prior to its exploration via NASA's Dawn spacecraft, it was known that Ceres was a low-density icy and rocky body that was likely differentiated, with the ice covered by a layer of likely hydrated silicates (McCord and Sotin, 2005), but the relative proportion of ice to silicates was not well constrained. The relatively high average surface temperature of Ceres (150-160 K at the equator) leaves pure water ice very soft and not capable of supporting topography. This factor led Bland (2013) to predict that if Ceres' crust was primarily ice, it would show evidence of extensive viscous crater relaxation. This relaxation would be most effective at the equator, where craters less than 4 km in diameter could be totally erased. Mid-latitude craters older than 10 Myr or larger than ~16 km in diameter would also be almost totally relaxed (Bland, 2013).

By contrast, Ceres' surface appears heavily cratered. Several large (> 80 km diameter) craters do not appear to be significantly relaxed. In addition, no latitudinal dependence on relaxation was observed; craters in the polar regions had the same relative amount of relaxation as equatorial craters. Bland et al. (2016) estimated that for Ceres to maintain its observed crater topography, the upper ~10 km of the surface would need to be composed of ~60% non-ice material (silicate rock, clays, or carbonates). Conversely, the global-scale shape of Ceres approaches hydrostatic equilibrium, suggesting a decrease of viscosity with depth that preserves the crater topography but permits the long-wavelength shape to relax (Fu et al., 2017).

However, there is abundant evidence that ice is prevalent in Ceres' near subsurface. Bland et al. (2016) did note the appearance of several anomalously relaxed craters that they related to lateral variations in subsurface ice content. Schmidt et al. (2017) noted that the craters show similar scaling relationships to those on icy instead of terrestrial planets, and that there was abundant spectral evidence for hydrated phases and morphological evidence of ice-rich surface flows. They estimated that between 10-50% of Ceres' surface material was ice by volume, and that the distribution of surface flows followed a latitudinal pattern consistent with greater ice volumes at high latitudes.

The Dawn gravity investigation provided high resolution observations of Ceres' gravitational field, up to spherical harmonic degree 10. Park et al. (2016) used these data to construct a two-layer model of Ceres' interior consisting of a chondritic core density (2500-2900 kg m^{-3}) and an outer shell 70-190 km thick composed of mixed volatiles, silicates, and salts (1680-1950 kg m^{-3}).

A major outstanding question for understanding Ceres' evolution is how water ice can remain in its upper layers. Detection of exposed water ice in Occator crater (Buczkowski et al., 2016), along with the evidence for ice rich flows from Schmidt et al. (2017), shows that ice is certainly present in Ceres' subsurface, but ice is thermodynamically unstable on Ceres' surface. Sublimation of surface ice would result in a decrease in relative ice to silicate content within the outer crust. In this model, the ice-rich or putative cryovolcanic landforms identified in Schmidt et al. (2017) and Buczkowski et al. (2016) would be localized sources of water ice recharge to the surface, not a global crustal component. Ceres' outer crust would then be a higher density than the underlying water-rich layers, providing the crustal strength to preserve the observed crater topography of Bland et al. (2016) but allowing the decrease in viscosity with depth (Fu et al., 2017).

Because of the high quality of Ceres' gravity data and the unresolved scientific questions of its density structure, I consider it an excellent benchmark case for testing my code. I construct a notional test model for Ceres using the two-layer model of Park et al. (2016). The model constraints are Ceres' ellipsoidal shape (major and minor axes), its bulk density, and its degree-2 gravity coefficients as measured by Dawn (Park et al., 2018). I allow the internal layers to vary between pure ice and pure silicate (900-3000 kg m^{-3}) above 200 km depth, and between mixed silicate and chondritic density (up to 4000 kg m^{-3}) deeper than that. To test the hypothesis that the upper layers of Ceres are dehydrated and thus denser than the ice-rich mantle, I have made an additional model run in an additional layer was added. The evaluation function for this model run is set to value the outer shell as having a higher density than the layer immediately below it.

4.5 **Preliminary results**

Initial testing of my models for Europa and Ceres show results that are broadly consistent with the results of Anderson et al. (1998) and Park et al. (2016), respectively. This demonstrates that the core functionality of GFGA is sound: given the same inputs and sets of assumptions as two previously published inverse modeling studies, it is able to reproduce results that are within the same parameter space using a forward modeling technique. Some persistent glitches prevent me from lending a broad interpretation to these results, so the following is not intended to advance any particular hypothesis about the internal state of Europa or Ceres. Rather, they are presented as evidence that GFGA can be used to replicate more common inverse modeling results, while also maintaining the future capability of bypassing the assumption of hydrostatic equilibrium that constrains inverse modeling of Europa.

4.5.1 **Europa: results**

The GFGA modeled output for Europa, compared to the previously observed or modeled constraints, is presented in Table III. The broad structure of the body is an ice shell with a thickness of ~30 km overlying a liquid water layer that extends to a depth of ~300 km. Below this is a layer of silicates ~250 km thick, followed by a large and dense iron core of R_{Europa} = 0.63.

This structure is within the parameter space described by Anderson et al. (1998). The authors considered it an outlier example, more consistent with their most extreme modeled value for C_{22}. They note that such a large iron core would mean that Europa’s formation was more relatively enriched in iron than Io’s, which would be counterintuitive to models of Galilean moon formation. However, given the paucity of data, it is a plausible alternative.

TABLE III: EUROPA RESULTS

Parameter	GFGA	Anderson et al. (1998)[a]
Average radius (km)	1561	1561
Layer 1 thickness (km)	319	~300
Layer 2 thickness (km)	244	~250
Layer 3 thickness (km)	998	~1015
Outer shell density (kg m^{-3})	940	N/A[b]
Layer 1 density (kg m^{-3})	1025	~1100
Layer 2 density (kg m^{-3})	2505	~3000
Layer 3 density (kg m^{-3})	7915	8000
J_2 [c]	-155×10^{-3}	$435.5 \pm 8.2 \times 10^{-6}$
C_{22} [c]	55.1×10^{-3}	$131.0 \pm 2.5 \times 10^{-6}$
Bulk mass (kg)	4.77×10^{22}	$4.79 \times 10^{22} \pm 6.2 \times 10^{-4}$
Bulk density (kg m^{-3})	3003	2989 ± 46

[a] If core radius = 0.65 R_{Europa}

[b] Anderson et al. (1998) calculated the density of the outer shell and first layer together.

[c] See text for discussion of errors in the spherical harmonic calculations.

Of concern to me, and the reason I hesitate to extend my analysis based on this result, is the significant variance of the C_{20} (equivalent to $-J_2$) and C_{22} coefficients from the Anderson et al. (1998) model. Attempts at bug fixing reveal that the value of the potential field and related spherical harmonics seem to be very sensitive to changes in unit input format (e.g., km vs. m) or mesh resolution parameters when applied to multi-layer models such as Europa. These bugs are not apparent in the benchmarking functions, which mostly deal with single-layer or constant density bodies with verifiable analytical solutions. This suggests that there is a bug in the plate model formulation module when multiple layers are applied.

Nevertheless, the GA and evaluation module are able to reach a solution that is within the possible parameter space for Europa. It is likely that this solution is based on the constraints that are not determined by the plate model: the body's total mass, triaxial shape, and other limits specified in the evaluation function such as density increase with depth. This demonstrates that, with a properly defined system, the GA and evaluation function working together are able to evolve towards a viable solution even in the presence of poor constraint inputs.

4.5.2 **Ceres: results**

The GFGA modeled output for Ceres is presented in Table IV. Both models are within the parameter space defined by Park et al. (2016), with a core approximately 400 km in radius with mean density slightly under CI chondrite (2460 kg m^{-3}), overlain by a shell of mixed ice and silicates with mean density ~1500 kg m^{-3}. Comparing the two models (increasing density vs. a density inversion in the top layer) shows that the GA did not strongly favor a large density inversion, maintaining an upper shell of only ~1 km thickness with a ~9 kg m^{-3} density differential.

While the simulated C_{20} of these models is significantly closer to the value modeled by Park et al. (2016) compared to that of the Europa simulations, the C_{22} error remains high. I thus reiterate my previous caveats regarding the interpretation of these results except to note that they are broadly consistent with the parameter space established by Park et al. (2016), and thus verify that the core functionality of the code is valid despite the glitches apparent in the plate model and related potential functions.

TABLE IV: CERES RESULTS

Parameter	GFGA: increasing density	GFGA: density inversion	Park et al. (2016)
Average radius (km)	484	484	484
Layer 1 thickness (km)	8	1	N/A[a]
Layer 2 thickness (km)	76	33	70-190
Layer 3 thickness (km)	400	450	294-414
Outer shell density (kg m^{-3})	943	1133	N/A[a]
Layer 1 density (kg m^{-3})	1105	1124	N/A[a]
Layer 2 density (kg m^{-3})	1568	1132	1680-1950
Layer 3 density (kg m^{-3})	2335	2250	2500-2900
J_2 [c]	11.9 x 10^{-3}	18.5 x 10^{-3}	11.8 x 10^{-3}
C_{22} [c]	12.3 x 10^{-2}	14.2 x 10^{-2}	1.59 x 10^{-4}
Bulk mass (kg)	9.39 x 1020	9.38 x 1020	9.39 x 1020
Bulk density (kg m^{-3})	2163	2161	2162 ± 8

[a] Park et al. (2016) calculated the density of the outer shell and first layer together.

[c] See text for discussion of errors in the spherical harmonic calculations.

4.6 **Future directions**

At the time of writing, the recently determined, major glitch in the plate model layering function has yet to be resolved. Continued bug fixing and verification efforts will continue between now and the final submission of this thesis for consideration.

For the code to be a useful tool to the planetary science community, the following functions remain to be implemented:

- Ability to accept as input parameters either spherical harmonic coefficients or line-of-sight gravity acceleration residuals. Currently, the code only accepts spherical harmonic coefficients as input. This will allow the code to function in its intended purpose of bypassing the hydrostatic assumption required for inverse modeling in situations such as planned missions such as Europa Clipper.

- Enable the user to set initial condition or evaluation function physical parameters (such as plausible density limits) to limit the solution and test hypotheses. Currently, these limits are set within the code itself.
- Implement a robust series of test and benchmarking functions with set inputs and outputs so that users can be confident in the products of the code.
- Host the project on an open source repository such as GitHub, with a plan of management, so that users can modify or improve the code.

5. SUMMARY

I summarize the major findings of the three chapters of this thesis below. The first chapter is considered complete and is in preparation for publishing. The second chapter has resulted in a reevaluation of the primary hypothesis for the formation of cantaloupe terrain on Triton and suggested an alternative method. However, additional research is required to validate this new formation mechanism. The third chapter presents preliminary results that validate the core functionality of the designed code, however, significant work remains to be done to remove bugs and render the code usable to the public.

5.1 Chaotic mountain blocks of Pluto

The chaotic mountain block ranges of Pluto's Sputnik Planitia were interpreted by White et al. (2017) as transported blocks of fractured water ice crust, but the mechanism of their transport was unclear. I present research that shows that lubricated basal sliding is a plausible emplacement mechanism. The presence of large nitrogen ice glaciers would serve to enhance the crustal fracture and calving of large blocks of water ice. Once fractured, the nitrogen would serve to basally lubricate these blocks. They would then slide down the low (~2°) regional slope of the Sputnik Planitia basin under their own weight. Their emplacement velocity would be rate-limited by the viscosity of the nitrogen and the amount of basal lubrication, but emplacement times as low as hundreds to thousands of years are plausible. The current position of the blocks is likely determined by the decrease of the regional slope below a critical value, and to a second order may also be controlled by the resistance to viscous drag exhibited by larger blocks.

5.2 The formation of cantaloupe terrain

I set out to test the diapir formation hypothesis for cantaloupe terrain originally formulated by Schenk and Jackson (1993). I incorporated modern numerical simulation techniques, models of Triton's internal heat budget, and temperature-dependent rheologies to expand on the simplified Rayleigh-Taylor analysis presented in that paper.

I find that diapir formation is inhibited by a combination of factors. The insulating properties of the surface ice have a marked effect on the convection regime of Triton's water ice mantle, changing it from a disordered and high-temperature convection regime to a more efficient regime where high temperatures are focused in mantle upwellings. This lower average subsurface temperature prevents ammonia dihydrate (NH_3*H_2O, abbreviated ADH), one of the suggested diapir components, from deforming readily at the surface.

By contrast, the higher relative insulation and weak rheology of surface CO_2 layers permitted deformation, but the co-viscous behavior expected for diapir formation by Rayleigh-Taylor instability did not occur. Instead, CO_2 that was warm enough to deform was confined by the rigid behavior of water ice at the same temperature. Layer thicknesses of 20 km (consistent with Schenk and Jackson [1993]) resulted in delamination and thinning of the CO_2 layer.

A parameter space was observed for CO_2 layer thicknesses ~10 km where downwelling "sinks" of CO_2 began to seep into the water ice mantle below. These sinks concentrated stresses in the rigid material above them in excess of their yield strength, indicating substantial surface fracture on a similar scale to the *cavi* that form cantaloupe terrain. The conditions for optimal downwelling sink formation were coincident with the center of mantle convection cells. These cells were observed to migrate laterally over time, suggesting a continuing resurfacing mechanism. Since the timing and plausibility of this mechanism is reliant on several factors outside the scope of this study (e.g. viscoelastic behavior of surface ice, rates of sublimation terrain formation, and rates of CO_2 ice deposition), I plan to expand this work further in the future.

5.3 GFGA: Gravity from Genetic Algorithms

I present a preliminary version of a C++ code designed to forward model the interior density structure of planetary bodies by the use of genetic algorithms. The purpose of this code is to provide an alternative way to model the gravity potential, spherical harmonics, and resulting structure of a planet in a similar way to more common inverse modeling techniques but which bypasses assumptions of hydrostatic shape that are required by those models. This is especially useful in situations such as Europa, where the full latitudinal and longitudinal variation of its gravity field (and thus the ability to determine higher order spherical harmonic coefficients without making the hydrostatic assumption) is not currently available. The full version of this code will be able to compare line-of-sight accelerations of a spacecraft directly to a series of models, and then utilize a genetic algorithm to find the best fit to the data directly.

The core functionality of this code is demonstrated: its ability to generate a degree-2 model of a planetary body, to calculate the gravity potential and resulting spherical harmonics of the body, and then to compare a population of these simulated objects against a standard and evolve towards the best fit. I present test cases where we attempt to recreate the three-layer model of Europa presented in Anderson et al. (1998) and the two-layer model of Ceres presented in Park et al. (2016), based on available degree-2 spherical harmonic models and bulk parameters.

As of the time of this writing, a major bug has been identified that prevents the code from correctly calculating the gravity potential (and thus spherical harmonics) of a multi-layered body. This prevents me from making any meaningful statements on the accuracy of the test case models presented here. However, even with this drawback, GFGA was able to evolve forward models that were within the parameter spaces presented by Anderson et al. (1998) and Park et al. (2016), based on the bulk parameters (mass and shape) and basic physical constraints of the body. Benchmark tests are also presented that demonstrate that the fundamentals of how GFGA calculates gravity potential and spherical harmonics are sound despite the current error.

The future work for this project will be focused on fixing the current bugs in the calculations, as well as adding features that will make it a useful open-source tool for the planetary science community.

CITED LITERATURE

Anderson, J. D. *et al.* Europa's differentiated internal structure: inferences from four Galileo encounters. *Science* **281,** 2019–22 (1998).

Bertrand, T. & Forget, F. Observed glacier and volatile distribution on Pluto from atmosphere–topography processes. *Nature* **540,** 86–89 (2016).

Bland, M. T. Predicted crater morphologies on Ceres: Probing internal structure and evolution. *Icarus* **226,** 510–521 (2013).

Bland, M. T. *et al.* Composition and structure of the shallow subsurface d of Ceres revealed by crater morphology. *Nat. Geosci.* **9,** 538–542 (2016).

Brown, R. H., Kirk, R. L., Johnson, T. V & Soderblom, L. A. Energy Sources for Triton's Geyser-Like Plumes. *Science.* **250,** 431–435 (1990).

Buczkowski, D. L. *et al.* The geomorphology of Ceres. *Science* **353,** aaf432 (2016).

Buffington, B. Trajectory Design Concept for the Proposed Europa Clipper Mission (Invited). in *AIAA/AAS Astrodynamics Specialist Conference* (American Institute of Aeronautics and Astronautics, 2014). doi:10.2514/6.2014-4105

Cheng, A. F. *et al.* Small-Scale Topography of 433 Eros from Laser Altimetry and Imaging. *Icarus* **155,** 51–74 (2002).

Cohen, D. *et al.* Debris-bed friction of hard-bedded glaciers. *J. Geophys. Res.* **110,** F02007 (2005).

Croft, S. K. *et al.* The geology of Triton. *Neptune Trit.* **1,** 879–947 (1995).

Cross, A. J., Hager, T. F., Smith, I. B. & Goldsby, D. L. Rheological Behavior of CO2 Ice with Application to Glacial Flow on Mars. in *50th Lunar and Planetary Science Conference* (2019).

Durham, W. B., Prieto-Ballesteros, O., Goldsby, D. L. & Kargel, J. S. Rheological and Thermal Properties of Icy Materials. *Space Sci. Rev.* **153,** 273–298 (2010).

Durham, W. B., Kirby, S. H. & Stern, L. A. Steady-state flow of solid CO_2 : Preliminary results. *Geophys. Res. Lett.* **26,** 3493–3496 (1999).

Eluszkiewicz, J. & Stevenson, D. J. Rheology of solid methane and nitrogen: applications to Triton. *Geophysical Research Letters* (1990).

Fu, R. R. *et al.* The interior structure of Ceres as revealed by surface topography. *Earth Planet. Sci. Lett.* **476,** 153–164 (2017).

Gaeman, J., Hier-Majumder, S. & Roberts, J. Sustainability of a subsurface ocean within Triton's interior. *Icarus* (2012).

Gershenfeld, N. *The nature of mathematical modeling*. Cambridge university press (1999).

Goren, L., Aharonov, E. & Anders, M. H. The long runout of the Heart Mountain landslide: Heating, pressurization, and carbonate decomposition. *J. Geophys. Res.* **115,** B10210 (2010).

Hammond, N.P., Parmentier, E.M., Barr, A. Melt Transport through Ammonia-Rich Ice Shells: Implications for the Evolution of Triton. in *AGU 2018* (2018).

Hand, K. P., Carlson, R. W. & Chyba, C. F. Energy, Chemical Disequilibrium, and Geological Constraints on Europa. *Astrobiology* **7,** 1006–1022 (2007).

Head, J. W. & Pappalardo, R. T. Brine mobilization during lithospheric heating on Europa: Implications for formation of chaos terrain, lenticula texture, and color variations. *J. Geophys. Res. Planets* **104,** 27143–27155 (1999).

Heiskanen, W. A. & Moritz, H. Physical geodesy. *Bull. géodésique* **86,** 491–492 (1967).

Jackson, M. P. A. *et al.* Salt Diapirs of the Great Kavir, Central Iran. 1–150 (1990). doi:10.1130/MEM177-p1

Kargel, J. S. *et al.* Europa's Crust and Ocean: Origin, Composition, and the Prospects for Life. *Icarus* **148,** 226–265 (2000).

Kattoum, Y. N. & Dombard, A. J. Calculating the topography of a differentiated Vesta. *Geophys. Res. Lett.* **36,** L24201 (2009).

Kay, J., Schenk, P. & Prockter, L. Triton, Europa, Enceladus, and Pluto, Oh my!: Topography of Acive Icy Ocean Worlds. in *50th Lunar and Planetary Science Conference* 2975 (2019).

Kelly, N. J. *et al.* Seasonal polar carbon dioxide frost on Mars: CO_2 mass and columnar thickness distribution. *J. Geophys. Res. Planets* **111,** (2006).

Kivelson, M. G. *et al.* Europa's Magnetic Signature: Report from Galileo's Pass on 19December 1996. *Science* **276,** 1239–1241 (1997).

Krasnopolsky, V. A. Titan's photochemical model: Further update, oxygen species, and comparison with Triton and Pluto. *Planet. Space Sci.* **73,** 318–326 (2012).

Kronbichler, M., Heister, T. & Bangerth, W. High accuracy mantle convection simulation through modern numerical methods. *Geophys. J. Int.* **191,** 12–29 (2012).

Lorenz, R. D., Jackson, B. K., Barnes, J. W., Spitale, J. & Keller, J. M. Ice rafts not sails: Floating the rocks at Racetrack Playa. *Am. J. Phys.* **79,** 37–42 (2011).

Malone, D.H., *et al.* Volcanic Initiation of the Eocene Heart Mountain Slide, Wyoming, USA. *The Journal of Geology* **125**, 439-457 (2017).

McKinnon, W. B. & Kirk, R. L. Triton. *Encycl. Sol. Syst.* 861–881 (2014). doi:10.1016/B978-0-12-415845-0.00040-2

McKinnon, W. B. *et al.* Convection in a volatile nitrogen-ice-rich layer drives Pluto's geological vigour. *Nature* **534,** 82–85 (2016).

Moore, W. B. & Schubert, G. The Tidal Response of Europa. *Icarus* **147,** 317–319 (2000).

Nimmo, F. *et al.* Reorientation of Sputnik Planitia implies a subsurface ocean on Pluto. *Nature* **540,** 94–96 (2016).

Nimmo, F. & Spencer, J. R. Powering Triton's recent geological activity by obliquity tides: Implications for Pluto geology. *Icarus* (2014). doi:10.1016/j.icarus.2014.01.044

Nimmo, F., Thomas, P. C., Pappalardo, R. T. & Moore, W. B. The global shape of Europa: Constraints on lateral shell thickness variations. *Icarus* **191,** 183–192 (2007).

Palasantzas, G. & De Hosson, J. T. M. Effects of self-affine surface roughness on the friction coefficient of rubbers in the presence of a liquid interlayer. *J. Appl. Phys.* **95,** 389–392 (2004).

Pappalardo, R. T. *et al.* Does Europa have a subsurface ocean? Evaluation of the geological evidence. *J. Geophys. Res. Planets* **104,** 24015–24055 (1999).

Park, R. S. *et al.* A partially differentiated interior for (1) Ceres deduced from its gravity field and shape. *Nature* **537**, (2016). doi:10.1038/nature18955

Park, R. S., Konopliv, A. S., Asmar, S. W. & Buccino, D. R. Dawn Ceres Derived Gravity Data. *NASA Planetary Data System* DAWN-A-RSS-5-CEGR-V3.0 (2018).

Poppe, A. R. & M. Horányi. Interplanetary Dust Delivery of Water to the Atmospheres of Pluto and Triton. In *50th Lunar and Planetary Science Conference* (2019).

Proudman, I. & Pearson, J. R. A. Expansions at small Reynolds numbers for the flow past a sphere and a circular cylinder. *J. Fluid Mech.* **2,** 237 (1957).

Quirico, E. *et al.* Composition, physical state, and distribution of ices at the surface of Triton. *Icarus* **139,** 159–178 (1999).

Robuchon, G. & Nimmo, F. Thermal evolution of Pluto and implications for surface tectonics and a subsurface ocean. *Icarus* **216,** 426–439 (2011).

Rosenburg, M. A. *et al.* Global surface slopes and roughness of the Moon from the Lunar Orbiter Laser Altimeter. *J. Geophys. Res.* **116,** E02001 (2011).

Ross, R. G. & Kargel, J. S. Thermal Conductivity of Solar System Ices, with Special Reference to Martian Polar Caps. In *Solar system ices* (Springer, Dordrecht, 1998). doi:10.1007/978-94-011-5252-5_2

Ruiz, J. Heat flow and depth to a possible internal ocean on Triton. *Icarus* **166,** 436–439 (2003).

Schenk, P. & Jackson, M. Diapirism on Triton: A record of crustal layering and instability. *Geology* **21,** 299–302 (1993).

Schenk, P. M. & Zahnle, K. On the negligible surface age of Triton. *Icarus* **192,** 135–149 (2007).

Schenk, P. M. *et al.* Basins, fractures and volcanoes: Global cartography and topography of Pluto from New Horizons. *Icarus* **314,** 400–433 (2018).

Schipper, D. J. & Maathuis, O. Friction Regimes in the Lubricants Solid-State Regime. *Tribol. Ser.* **30,** 45–51 (1995).

Schmidt, B. E. *et al.* Geomorphological evidence for ground ice on dwarf planet Ceres. *Nat. Geosci.* **10,** 338–343 (2017).

Schulson, E. M. & Fortt, A. L. Friction of ice on ice. *J. Geophys. Res. Solid Earth* **117,** (2012).

Schurmeier, L. R. & Dombard, A. J. Crater relaxation on Titan aided by low thermal conductivity sand infill. *Icarus* **305,** 314–323 (2018).

Shock, E. L. & McKinnon, W. B. Hydrothermal Processing of Cometary Volatiles—Applications to Triton. *Icarus* **106,** 464–477 (1993).

Singer, K. N., McKinnon, W. B., Schenk, P. M. & Moore, J. M. Massive ice avalanches on Iapetus mobilized by friction reduction during flash heating. *Nat. Geosci.* **5,** 574–578 (2012).

Solomatov, V. S. Scaling of temperature- and stress-dependent viscosity convection. *Phys. Fluids* **7,** 266–274 (1995).

Sori, M. M., Bapst, J., Bramson, A. M., Byrne, S. & Landis, M. E. A Wunda-full world? carbon dioxide ice deposits on Umbriel and other Uranian moons. *Icarus* **290,** 1–13 (2017).

Stern, S. A. *et al.* The Pluto system: Initial results from its exploration by New Horizons. *Science* **350,** aad1815 (2015).

Talbot, C. J. Halokinesis and thermal convection. *Nature* **273,** 739–741 (1978).

Tanaka, K. L., Kargel, J. S., MacKinnon, D. J., Hare, T. M. & Hoffman, N. Catastrophic erosion of Hellas basin rim on Mars induced by magmatic intrusion into volatile-rich rocks. *Geophys. Res. Lett.* **29,** 37-1-37–4 (2002).

Tobie, G., Choblet, G. & Sotin, C. Tidally heated convection: Constraints on Europa's ice shell thickness. *J. Geophys. Res.* **108,** 5124 (2003).

Turcotte, D. L. & Schubert, G. *Geodynamics*. (2014).

Umurhan, O. M. *et al.* Modeling glacial flow on and onto Pluto's Sputnik Planitia. *Icarus* **287,** 301–319 (2017).

Vance, S. D., Hand, K. P. & Pappalardo, R. T. Geophysical controls of chemical disequilibria in Europa. *Geophys. Res. Lett.* **43,** 4871–4879 (2016).

Wall, M. GAlib: A C++ Library of Genetic Algorithm Components. *Mechanical Engineering Department, Massachusetts Institute of Technology, 87* (1999).

Watkins, J. A., Ehlmann, B. L. & Yin, A. Long-runout landslides and the long-lasting effects of early water activity on Mars. *Geology* **43,** 107–110 (2015).

White, O. L. *et al.* Geological mapping of Sputnik Planitia on Pluto. *Icarus* **287,** 261–286 (2017).

Wieczorek, M. A. The gravity and topography of the terrestrial planets. in *Treatise on Geophysics* 165–206 (2007).

Wong, M. L. *et al.* The photochemistry of Pluto's atmosphere as illuminated by New Horizons. *Icarus* **287,** 110–115 (2017).

Yamashita, Y. & Kato, M. Viscoelastic properties of polycrystalline solid methane and carbon dioxide. *Geophys. Res. Lett.* **24,** 1327–1330 (1997).

Yamashita, Y., Kato, M. & Arakawa, M. Experimental study on the rheological properties of polycrystalline solid nitrogen and methane: Implications for tectonic processes on Triton. *Icarus* **207,** 972–977 (2010).

Zolotov, M. Y. & Shock, E. L. Energy for biologic sulfate reduction in a hydrothermally formed ocean on Europa. *J. Geophys. Res.* **108,** 5022 (2003).

Zolotov, M. Y. & Shock, E. L. A model for low-temperature biogeochemistry of sulfur, carbon, and iron on Europa. *J. Geophys. Res.* **109,** E06003 (2004).

APPENDICES

APPENDIX A: ASPECT MATERIAL MODEL: VOLATILE ICES

```
/*
  Copyright (C) 2011 - 2017 by the authors of the ASPECT code.

  This file is part of ASPECT.

  ASPECT is free software; you can redistribute it and/or modify
  it under the terms of the GNU General Public License as published by
  the Free Software Foundation; either version 2, or (at your option)
  any later version.

  ASPECT is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public License
  along with ASPECT; see the file LICENSE.  If not see
  <http://www.gnu.org/licenses/>.
*/

/*
 This is a material model for temperature-dependent ices using a
volatile ice rheology as defined in Umurhan et al. (2017). It was
designed by Sean O'Hara as a plugin for ASPECT. It is a modification
of the "compositional field" material model included as a cookbook
with the distribution of ASPECT.
 */

#include <aspect/material_model/ice_volatile.h>


namespace aspect
{
  namespace MaterialModel
  {
    template <int dim>
    void
    ice_volatile<dim>::
    evaluate(const MaterialModel::MaterialModelInputs<dim> &in,
             MaterialModel::MaterialModelOutputs<dim> &out) const
    {
      for (unsigned int i=0; i < in.position.size(); ++i)
        {
```

```
      const double c = (in.composition[i].size()>0)
        ?
        std::max(0.0, in.composition[i][0])
        :
        0.0;


        //additions to hook in additional dependencies based on the
compositional fields.
        const double T0 = (reference_T + (compositional_delta_T * c));
        const double Ea = (activation_energy + (compositional_delta_Ea
* c));

        double temperature_dependence =
std::exp(Ea/(0.008314*in.temperature[i]));

            //calculating the viscosity assuming a Newtonian rheology
and a grain size of 1mm, as per Umurhan et al. (2017) eq. 7.
            out.viscosities[i] =
1.84e16*pow(.001,2)*(in.temperature[i]/45)*exp((1030/in.temperature[i]
)-(1030/45));


          out.densities[i] = reference_rho * (1 - thermal_alpha *
(in.temperature[i] - T0)) + compositional_delta_rho * c;
          out.thermal_expansion_coefficients[i] = thermal_alpha +
(compositional_delta_alpha * c);
          out.specific_heat[i] = reference_specific_heat +
(compositional_delta_cp);
          out.thermal_conductivities[i] = k_value +
(compositional_delta_k);
          out.compressibilities[i] = 0.0;
          // Pressure derivative of entropy at the given positions.
          out.entropy_derivative_pressure[i] = 0.0;
          // Temperature derivative of entropy at the given positions.
          out.entropy_derivative_temperature[i] = 0.0;
          // Change in composition due to chemical reactions at the
          // given positions. The term reaction_terms[i][c] is the
          // change in compositional field c at point i.
          for (unsigned int c=0; c<in.composition[i].size(); ++c)
            out.reaction_terms[i][c] = 0.0;
        }
    }
```

APPENDIX A (CONTINUED)

```
template <int dim>
double

ice_volatile<dim>::
reference_viscosity () const
{
  return eta;
}



template <int dim>
bool
ice_volatile<dim>::
is_compressible () const
{
  return false;
}



template <int dim>
void
ice_volatile<dim>::declare_parameters (ParameterHandler &prm)
{
  prm.enter_subsection("Material model");
  {
    prm.enter_subsection("Ice model");
    {
      prm.declare_entry ("Reference density", "3300",
                         Patterns::Double (0));
      prm.declare_entry ("Reference temperature", "293",
                         Patterns::Double (0));
      prm.declare_entry ("Reference Viscosity", "5e24",
                         Patterns::Double (0));
      prm.declare_entry ("Composition viscosity prefactor", "1.0",
                         Patterns::Double (0));
      prm.declare_entry ("Thermal viscosity exponent", "0.0",
                         Patterns::Double (0));
      prm.declare_entry("Maximum thermal prefactor","1.0e2",
                        Patterns::Double (0));
      prm.declare_entry("Minimum thermal prefactor","1.0e-2",
                        Patterns::Double (0)):;
      prm.declare_entry ("Thermal conductivity", "4.7",
                         Patterns::Double (0));
      prm.declare_entry ("Reference specific heat", "1250",
                         Patterns::Double (0));
```

APPENDIX A (CONTINUED)

```
         prm.declare_entry ("Thermal expansion coefficient", "2e-5",
                            Patterns::Double (0));
         prm.declare_entry ("Density differential for compositional 
field 1", "0",
                            Patterns::Double());

          ///entries below express other dependencies of material 
properties based on the compositional field.

          prm.declare_entry ("Power-law exponent", "1.0",
                             Patterns::Double (0),
                             "The power law exponent n in the 
Arrhenius viscosity formulation. Varies with the compositional field 
and the deformation mechanism.");

          prm.declare_entry ("Activation energy", "0",
                             Patterns::Double(0),
                             "The activation energy of the 
background material. Varies with the deformation mechanism.");

          prm.declare_entry ("Reference viscosity differential for 
compositional field 1", "0",
                             Patterns::Double(0),
                             "The dependence of the reference 
viscosity of a material on the compositional field, expressed as a 
differential.");

          prm.declare_entry ("Reference temperature differential for 
compositional field 1", "0",
                             Patterns::Double(0),
                             "The dependence of the reference 
temperature of a material on the compositional field, expressed as a 
differential.");
           prm.declare_entry ("Activation energy differential for 
compositional field 1", "0",
                            Patterns::Double(0),
                            "The dependence of the activation energy 
of a material on the compositional field, expressed as a 
differential.");
           prm.declare_entry ("Specific heat differential for 
compositional field 1", "0",
                            Patterns::Double(0),
                            "The dependence of the specific heat of 
a material on the compositional field, expressed as a differential.");
           prm.declare_entry ("Thermal conductivity differential for 
compositional field 1", "0",
```

APPENDIX A (CONTINUED)

```
                              Patterns::Double(0),
                              "The dependence of the thermal
conductivity of a material on the compositional field, expressed as a
differential.");
          prm.declare_entry ("Thermal expansion coefficient
differential for compositional field 1", "0",
                              Patterns::Double(0),
                              "The dependence of the thermal expansion
coefficient of a material on the compositional field, expressed as a
differential.");
        }
        prm.leave_subsection();
      }
      prm.leave_subsection();
    }



    template <int dim>
    void
    ice_volatile<dim>::parse_parameters (ParameterHandler &prm)
    {
      prm.enter_subsection("Material model");
      {
        prm.enter_subsection("Ice model");
        {
          reference_rho = prm.get_double ("Reference density");
          reference_T = prm.get_double ("Reference temperature");
          eta   = prm.get_double ("Reference Viscosity");
          composition_viscosity_prefactor = prm.get_double
("Composition viscosity prefactor");
          thermal_viscosity_exponent = prm.get_double ("Thermal
viscosity exponent");
          maximum_thermal_prefactor = prm.get_double ("Maximum thermal
prefactor");
          minimum_thermal_prefactor = prm.get_double ("Minimum thermal
prefactor");
          if ( maximum_thermal_prefactor == 0.0 )
maximum_thermal_prefactor = std::numeric_limits<double>::max();
          if ( minimum_thermal_prefactor == 0.0 )
minimum_thermal_prefactor = std::numeric_limits<double>::min();

          k_value = prm.get_double ("Thermal conductivity");
          reference_specific_heat = prm.get_double ("Reference
specific heat");
```

```
          thermal_alpha = prm.get_double ("Thermal expansion
coefficient");
          compositional_delta_rho = prm.get_double ("Density
differential for compositional field 1");

//new parameters
            exponent = prm.get_double ("Power-law exponent");
            activation_energy = prm.get_double ("Activation energy");
            compositional_delta_eta  = prm.get_double ("Reference
viscosity differential for compositional field 1");
            compositional_delta_T = prm.get_double ("Reference
temperature differential for compositional field 1");
            compositional_delta_Ea = prm.get_double ("Activation
energy differential for compositional field 1");
            compositional_delta_k = prm.get_double ("Thermal
conductivity differential for compositional field 1");
            compositional_delta_cp = prm.get_double ("Specific heat
differential for compositional field 1");
            compositional_delta_alpha = prm.get_double ("Thermal
expansion coefficient differential for compositional field 1");
          if (thermal_viscosity_exponent!=0.0 && reference_T == 0.0)
            AssertThrow(false, ExcMessage("Error: Material model
simple with Thermal viscosity exponent can not have reference_T=0."));
        }
        prm.leave_subsection();
      }
      prm.leave_subsection();

      // Declare dependencies on solution variables
      this->model_dependence.compressibility =
NonlinearDependence::none;
      this->model_dependence.specific_heat =
NonlinearDependence::compositional_fields;
      this->model_dependence.thermal_conductivity =
NonlinearDependence::compositional_fields;

        //added nonlinear dependence on T and composition. -S

        this->model_dependence.viscosity =
NonlinearDependence::compositional_fields |
NonlinearDependence::temperature;
      this->model_dependence.density =
NonlinearDependence::compositional_fields |
NonlinearDependence::temperature;

      if (thermal_viscosity_exponent != 0)
```

APPENDIX A (CONTINUED)

```
        this->model_dependence.viscosity |=
NonlinearDependence::temperature;
      if (composition_viscosity_prefactor != 1.0)
        this->model_dependence.viscosity |=
NonlinearDependence::compositional_fields;

      if (thermal_alpha != 0)
        this->model_dependence.density
|=NonlinearDependence::temperature |
NonlinearDependence::compositional_fields;
      if (compositional_delta_rho != 0)
        this->model_dependence.density
|=NonlinearDependence::compositional_fields;
    }
  }
}

// explicit instantiations
namespace aspect
{
  namespace MaterialModel
  {
    ASPECT_REGISTER_MATERIAL_MODEL(ice_volatile,
                                   "ice_volatile",
                                   "
                                   This is a material model for
temperature-dependent ices using a volatile ice rheology as defined in
Umurhan et al. (2017). It was designed by Sean O'Hara as a plugin for
ASPECT. It is a modification of the "compositional field" material
model included as a cookbook with the distribution of ASPECT.
                                   ")
  }
}
```

APPENDIX B: ASPECT MATERIAL MODEL: ARRHENIUS RELATIONSHIP VISCOSITIES

```
/*
  Copyright (C) 2011 - 2017 by the authors of the ASPECT code.

  This file is part of ASPECT.

  ASPECT is free software; you can redistribute it and/or modify
  it under the terms of the GNU General Public License as published by
  the Free Software Foundation; either version 2, or (at your option)
  any later version.

  ASPECT is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public License
  along with ASPECT; see the file LICENSE.  If not see
  <http://www.gnu.org/licenses/>.
*/

/*
 This is a material model for temperature-dependent ices using an
Arrhenius type viscosity. It was designed by Sean O'Hara as a plugin
for ASPECT. It is a modification of the "compositional field" material
model included as a cookbook with the distribution of ASPECT.
 */


#include <aspect/material_model/ice_arrhenius.h>


namespace aspect
{
  namespace MaterialModel
  {
    template <int dim>
    void
    ice_arrhenius<dim>::
    evaluate(const MaterialModel::MaterialModelInputs<dim> &in,
             MaterialModel::MaterialModelOutputs<dim> &out) const
```

APPENDIX B (CONTINUED)

```
    {
      for (unsigned int i=0; i < in.position.size(); ++i)
        {

 const double c = (in.composition[i].size()>0)
        ?
        std::max(0.0, in.composition[i][0])
        :
        0.0;


        //additions to hook in additional dependencies based on the
compositional fields.
        const double T0 = (reference_T + (-1 * compositional_delta_T *
c));
        const double Ea = (activation_energy + (-1 *
compositional_delta_Ea * c));

            double temperature_dependence =
std::min(std::exp((Ea/0.008314)*((1/in.temperature[i])-
(1/reference_T))),1e10);


            //calculating the viscosity as per Nimmo et al. (2014) eq.
3.
            out.viscosities[i] = eta * temperature_dependence;


          out.densities[i] = reference_rho * (1 - thermal_alpha *
(in.temperature[i] - T0)) + compositional_delta_rho * c;
          out.thermal_expansion_coefficients[i] = thermal_alpha + (-1
* compositional_delta_alpha * c);
          out.specific_heat[i] = reference_specific_heat +
(compositional_delta_cp * c);
          out.thermal_conductivities[i] = k_value + ( (-1 *
compositional_delta_k) * c);
            //The -1 factor here is a kludge around Aspect not easily
letting you input a negative double value.
          out.compressibilities[i] = 0.0;
          // Pressure derivative of entropy at the given positions.
          out.entropy_derivative_pressure[i] = 0.0;
          // Temperature derivative of entropy at the given positions.
          out.entropy_derivative_temperature[i] = 0.0;
          // Change in composition due to chemical reactions at the
          // given positions. The term reaction_terms[i][c] is the
          // change in compositional field c at point i.
```

APPENDIX B (CONTINUED)

```
     for (unsigned int c=0; c<in.composition[i].size(); ++c)
        out.reaction_terms[i][c] = 0.0;
    }
}


template <int dim>
double
ice_arrhenius<dim>::
reference_viscosity () const
{
  return eta;
}



template <int dim>
bool
ice_arrhenius<dim>::
is_compressible () const
{
  return false;
}



template <int dim>
void
ice_arrhenius<dim>::declare_parameters (ParameterHandler &prm)
{
  prm.enter_subsection("Material model");
  {
    prm.enter_subsection("Ice model");
    {
      prm.declare_entry ("Reference density", "3300",
                         Patterns::Double (0));
      prm.declare_entry ("Reference temperature", "293",
                         Patterns::Double (0));
      prm.declare_entry ("Reference viscosity", "5e24",
                         Patterns::Double (0));
      prm.declare_entry ("Composition viscosity prefactor", "1.0",
                         Patterns::Double (0));
      prm.declare_entry ("Thermal viscosity exponent", "0.0",
                         Patterns::Double (0));
      prm.declare_entry("Maximum thermal prefactor","1.0e2",
                        Patterns::Double (0));
```

```
       prm.declare_entry("Minimum thermal prefactor","1.0e-2",
                            Patterns::Double (0));
          prm.declare_entry ("Thermal conductivity", "4.7",
                             Patterns::Double (0));
          prm.declare_entry ("Reference specific heat", "1250",
                             Patterns::Double (0));
          prm.declare_entry ("Thermal expansion coefficient", "2e-5",
                             Patterns::Double (0));
          prm.declare_entry ("Density differential for compositional
field 1", "0",
                             Patterns::Double());

            ///new entries that express other dependencies of material
properties based on the compositional field.

            prm.declare_entry ("Power-law exponent", "1.0",
                               Patterns::Double (),
                               "The power law exponent n in the
Arrhenius viscosity formulation. Varies with the compositional field
and the deformation mechanism.");

            prm.declare_entry ("Activation energy", "0",
                               Patterns::Double(),
                               "The activation energy of the
background material. Varies with the deformation mechanism.");

            prm.declare_entry ("Reference viscosity differential for
compositional field 1", "0",
                               Patterns::Double(),
                               "The dependence of the reference
viscosity of a material on the compositional field, expressed as a
differential.");

            prm.declare_entry ("Reference temperature differential for
compositional field 1", "0",
                               Patterns::Double(),
                               "The dependence of the reference
temperature of a material on the compositional field, expressed as a
differential.");
           prm.declare_entry ("Activation energy differential for
compositional field 1", "0",
                              Patterns::Double(),
                              "The dependence of the activation energy
of a material on the compositional field, expressed as a
differential.");
```

APPENDIX B (CONTINUED)

```
          prm.declare_entry ("Specific heat differential for
compositional field 1", "0",
                             Patterns::Double(),
                             "The dependence of the specific heat of
a material on the compositional field, expressed as a differential.");
          prm.declare_entry ("Thermal conductivity differential for
compositional field 1", "0",
                             Patterns::Double(),
                             "The dependence of the thermal
conductivity of a material on the compositional field, expressed as a
differential.");
          prm.declare_entry ("Thermal expansion coefficient
differential for compositional field 1", "0",
                             Patterns::Double(),
                             "The dependence of the thermal expansion
coefficient of a material on the compositional field, expressed as a
differential.");
        }
        prm.leave_subsection();
      }
      prm.leave_subsection();
    }



    template <int dim>
    void
    ice_arrhenius<dim>::parse_parameters (ParameterHandler &prm)
    {
      prm.enter_subsection("Material model");
      {
        prm.enter_subsection("Ice model");
        {
          reference_rho = prm.get_double ("Reference density");
          reference_T = prm.get_double ("Reference temperature");
          eta = prm.get_double ("Reference viscosity");
          composition_viscosity_prefactor = prm.get_double
("Composition viscosity prefactor");
          thermal_viscosity_exponent = prm.get_double ("Thermal
viscosity exponent");
          maximum_thermal_prefactor = prm.get_double ("Maximum thermal
prefactor");
          minimum_thermal_prefactor = prm.get_double ("Minimum thermal
prefactor");
          if ( maximum_thermal_prefactor == 0.0 )
maximum_thermal_prefactor = std::numeric_limits<double>::max();
```

APPENDIX B (CONTINUED)

```
          if ( minimum_thermal_prefactor == 0.0 )
minimum_thermal_prefactor = std::numeric_limits<double>::min();

          k_value = prm.get_double ("Thermal conductivity");
          reference_specific_heat = prm.get_double ("Reference
specific heat");
          thermal_alpha = prm.get_double ("Thermal expansion
coefficient");
          compositional_delta_rho = prm.get_double ("Density
differential for compositional field 1");

//new parameters

            activation_energy = prm.get_double ("Activation energy");
            compositional_delta_eta  = prm.get_double ("Reference
viscosity differential for compositional field 1");
            compositional_delta_T = prm.get_double ("Reference
temperature differential for compositional field 1");
            compositional_delta_Ea = prm.get_double ("Activation
energy differential for compositional field 1");
            compositional_delta_k = prm.get_double ("Thermal
conductivity differential for compositional field 1");
            compositional_delta_cp   = prm.get_double ("Specific heat
differential for compositional field 1");
            compositional_delta_alpha = prm.get_double ("Thermal
expansion coefficient differential for compositional field 1");

          if (thermal_viscosity_exponent!=0.0 && reference_T == 0.0)
            AssertThrow(false, ExcMessage("Error: Material model
simple with Thermal viscosity exponent can not have reference_T=0."));
        }
        prm.leave_subsection();
      }
      prm.leave_subsection();

      // Declare dependencies on solution variables
      this->model_dependence.compressibility =
NonlinearDependence::none;
      this->model_dependence.specific_heat =
NonlinearDependence::compositional_fields;
      this->model_dependence.thermal_conductivity =
NonlinearDependence::compositional_fields;

        //added nonlinear dependence on T and composition. -S
```

APPENDIX B (CONTINUED)

```
        this->model_dependence.viscosity =
NonlinearDependence::compositional_fields |
NonlinearDependence::temperature;
      this->model_dependence.density =
NonlinearDependence::compositional_fields |
NonlinearDependence::temperature;

      if (thermal_viscosity_exponent != 0)
        this->model_dependence.viscosity |=
NonlinearDependence::temperature;
      if (composition_viscosity_prefactor != 1.0)
        this->model_dependence.viscosity |=
NonlinearDependence::compositional_fields;

      if (thermal_alpha != 0)
        this->model_dependence.density
|=NonlinearDependence::temperature |
NonlinearDependence::compositional_fields;
      if (compositional_delta_rho != 0)
        this->model_dependence.density
|=NonlinearDependence::compositional_fields;
    }
  }
}

// explicit instantiations
namespace aspect
{
  namespace MaterialModel
  {
    ASPECT_REGISTER_MATERIAL_MODEL(ice_arrhenius,
                                   "ice_arrhenius",
                                   " This is a material model for
temperature-dependent ices using an Arrenhius type viscosity. It was
designed by Sean O'Hara as a plugin for ASPECT. It is a modification
of the "compositional field" material model included as a cookbook
with the distribution of ASPECT.")
  }
}
```

APPENDIX C: GRAVITY FROM GENETIC ALGORITHMS RAW CODE

```
//
//  GFGA (Gravity from Genetic Algorithms)
//
//
//  Begun by Sean O'Hara on 7/2016. Last update 6/2019.
//
//

#include <stdio.h>
#include <cmath>
#include <math.h>
#include <ga/ga.h>
#include <ga/gaconfig.h>
#include <ga/std_stream.h>
#include <ga/GARealGenome.h>
#include <ga/GARealGenome.C>
#include <ga/GAAllele.h>

#define INSTANTIATE_REAL_GENOME
#define GALIB_USE_STREAMS


using namespace std;

int Parameters(int layers, double a, double b, double c, double
trueMass, double trueC20, double trueC21, double trueC22, double
trueS21, double trueS22);


double Platemodel(double a, double b, double c, int v, int offset);
double PotentialCalc(GAGenome &);
double SphericalHarmonics(int a, GARealAlleleSetArray &);


float ShapeObjective(GAGenome &);

double factorial (double);

int spacing;
double majr;
double inter;
double minr;
double density;
double R0;
//Reference radius for Europa = 1565 km (Anderson et al., 1998)
```

APPENDIX C (CONTINUED)

```
int spacing_lat=5;
int spacing_long=5;
//spacing is currently set in the code definition despite ideally
being a settable parameter.

int N_lat = (180/spacing_lat) + 1;
int N_long = (360/spacing_long) + 1;
int N_nodes = N_lat * N_long;
int N_plates = 2 * (N_long - 1) * (N_lat - 2);

int i;
int s;
int v;
int p;
int layers;
const double pi = acos(-1);
double trueMass;
double trueC20;
double trueC21;
double trueC22;
double trueS21;
double trueS22;
double offsetX;
double offsetY;
double offsetZ;
double TestValue;

double V;
double Rn;

double C2[3] = {0.0,0.0,0.0};
double S2[2] = {0.0,0.0};
//spherical harmonic coefficients

double Yc[3] = {0.0};
double Ys[2] = {0.0};
double Ybar[3] = {0.0};
//Terms for calculating associated Legendre polynomials

double Vsurf[65341][3];
//plate model vertices in Cartesian coordinates. See Cheng 2002
Appendix.

double Potential[128880];
double Cen[20][128880][3];
double Cen_position[20][128880][3];
```

APPENDIX C (CONTINUED)

```
//Enough array slots for a 1x1 degree spacing plate model.
double Ref_Cen[128880][3];
double Ref_Cen_position[128880][3];
double Ref_area[128880];
double N[20][128880][3];
double Plate_area[20][128880];
double Unit_area[128880];

double ShellMass;
double PointMass;
double BodyMass;

double Xcm[20][3];



double factorial (double number)
{
    int temp = 0;
    if (number <= 1.0) return 1.0;

    temp = number * factorial(number-1.0);
    return temp;
}


int Parameters()
{
    //The code reads in a parameter file and gives an error if not
found. It looks for a .txt file with the following data, in order,
separated by hard returns: The latitude/longitude grid spacing in
degrees; the major, intermediate, and minor axes of the body in km;
the reference radius of the body in km; the number of internal
ellipsoids to generate; the calculated C20, C21, C22, S21, and S22
coefficients for the body; and the center of mass/center of figure
offset of the body in kilometers.

    ifstream
in("/Users/seanohara/Dropbox/Research/GA/Input/europa_parameter_test.t
xt");

    if (in.is_open())
        {
            cout << "Parameter file opened successfully." << endl;
        }
    Else
```

```
   {
            cout << "Could not find parameter file. Make sure it is in 
the Input directory." << endl;
            throw std::exception();
        }

    in >> spacing >> majr >> inter >> minr >> R0  >> layers >> 
trueMass >> trueC20 >> trueC21 >> trueC22 >> trueS21 >> trueS22 >> 
offsetX >> offsetY >> offsetZ >> TestValue;

    if (TestValue == 0)
        {
            cout << "Seed parameters:" << "\n" ;
            cout << "Generating a " << spacing << "x" << spacing << " 
degree plate model grid." << "\n";
            cout << "Body major axis a: " << majr << "\n" ;
            cout << "Body intermediate axis b: " << inter << "\n" ;
            cout << "Body minor axis c: " << minr << "\n" ;
            cout << "Reference radius: " << R0 << "\n";
            cout << "COM/COF offset in x (km): " << offsetX << "\n";
            cout << "COM/COF offset in y (km): " << offsetY << "\n";
            cout << "COM/COF offset in z (km): " << offsetZ << "\n";
            cout << "Generating shell + " << layers-1 << " internal 
layers. \n";
        }
    else
        {
            cout << "Error: Test function not currently enabled." << 
"\n" ;
            throw std::exception();
        }

    return(0);

}

//  Plate model function by Sean O'Hara on 7/5/16. Modification of 
triangular plate model code from Kattoum and Dombard 2009.


double Platemodel(double a, double b, double c, int v, int offset)
{
    double J1 = 0.0;
    double J2 = 0.0;
    double R1 = 0.0;
    double R2 = 0.0;
```

APPENDIX C (CONTINUED)

```
 double R3 = 0.0;

    int spacing_lat = spacing;
    int spacing_long = spacing;

    int latitude = -90;
    int longitude = 0;

    int lat[N_lat*N_long];
    int lon[N_lat*N_long];
    int colat[N_lat*N_long];

    double Vsurf[N_nodes][3];
//plate model vertices in Cartesian coordinates. See Cheng 2002
Appendix.

    double Node_position[N_nodes][3];
//Spherical coordinates of each vertex.

    for (int s = 0; s < N_nodes; s++)
    {

    if (longitude < 360)
            {
    longitude = longitude + spacing_long;
            }

    else if (latitude < 180)
            {
    longitude = 0;
    latitude = latitude + spacing_lat;
            }
    else if (longitude == 360 && latitude == 180)
            {
    longitude = 0;
    latitude = -90;
            }



    lat[s] = latitude;
    lon[s] = longitude;

    colat[s] = (90-lat[s]);
```

APPENDIX C (CONTINUED)

```
    J1 = lon[s] * pi/180.0;
    J2 = colat[s] * pi/180.0;
//convert latitude to colatitude, and both lon and colat to radians


    double* R = new double[N_nodes];

    R1 = ((1.0 + cos(2 * J1)) * (1.0 - cos(2 * J2))) / (2.0 * (a * a)
);
    R2 = ((1.0 - cos(2 * J1)) * (1.0 - cos(2 * J2))) / (2.0 * (b * b)
);
    R3 = (1.0 + cos(2 * J2)) / (2.0 * (c * c) );

    R[s] = sqrt(1.0/(R1+R2+R3));
//calculates radius of ellipsoid at each mesh point in spherical polar
coordinates.

        if (offset == 0)
//parameter switch to indicate whether to include the COM/COF offset
in calculations. Set to 0 in the reference sphere or unit sphere
calculations and 1 otherwise.
    {
    Vsurf[s][0] = a * sin(J2) * cos(J1);
    Vsurf[s][1] = b * sin(J2) * sin(J1);
    Vsurf[s][2] = c * cos(J2);
    }
//calculates Cartesian position of each individual vertex of the plate
model.


    else if (offset != 0)
    {
        Vsurf[s][0] = a * sin(J2) * cos(J1) + offsetX;
        Vsurf[s][1] = b * sin(J2) * sin(J1) + offsetY ;
        Vsurf[s][2] = c * cos(J2) + offsetZ;
    }
//calculates Cartesian position of each individual vertex of the plate
model.

    Node_position[s][0] = R[s];
    Node_position[s][1] = J1 * (180.0/pi);
    Node_position[s][2] = J2 * (180.0/pi);

    delete[] R;

    }
```

APPENDIX C (CONTINUED)

```
            //NOTE SIGN CONVENTION FOR VECTOR ARRAYS: [0] = r/x, [1] =
lon/y, [2] = colat/z


    int p[N_plates][3];


        for (int i=0; i < (N_long - 2) + 1; i++)      //Defines node
points for generating the triangular plate model.
    {
            int j = N_plates - N_long + 1;
            p[i][0] = i;
            p[i][1] = i + N_long + 1;
            p[i][2] = i + N_long;
            p[i+j][0] = i + N_nodes - 2 * N_long;
            p[i+j][1] = i + N_nodes - 2 * N_long + 1;
            p[i+j][2] = i + N_nodes - N_long + 1;
    }

        for (int i=1; i < (N_lat - 3) + 1; i++)
    {
            int j = (N_long - 1) * (2 * i - 1);
            for (int k = 0; k < (N_long - 2) + 1; k++)
            {
                p[j+k][0] = i * N_long + k;
                p[j+k][1] = i * N_long + k + 1;
                p[j+k][2] = i * N_long + k + 1 + N_long;
                p[j+k + N_long - 1][0] = i * N_long + k;
                p[j+k + N_long - 1][1] = i * N_long + k + N_long + 1;
                p[j+k + N_long - 1][2] = i * N_long + k + N_long;
            }
    }

    int k;
    int l;
    int m;
    double Plate_normal[layers][N_plates];

    for (int i=0; i<N_plates; i++)
    {

        k = p[i][0];
        l = p[i][1];
        m = p[i][2];
```

APPENDIX C (CONTINUED)

```
   //Definition of outward normal vector of plates - see Cheng 2002
appendix. Represented as x,y,z [j] coordinates of endpoint.

        double U1;
        double U2;
        double U3;
        double V1;
        double V2;
        double V3;

        U1 = Vsurf[l][0] - Vsurf[k][0];
        U2 = Vsurf[l][1] - Vsurf[k][1];
        U3 = Vsurf[l][2] - Vsurf[k][2];
        V1 = Vsurf[m][0] - Vsurf[k][0];
        V2 = Vsurf[m][1] - Vsurf[k][1];
        V3 = Vsurf[m][2] - Vsurf[k][2];

        N[v][i][0] = (U2 * V3) - (U3 * V2);
        N[v][i][1] = (U3 * V1) - (U1 * V3);
        N[v][i][2] = (U1 * V2) - (U2 * V1);

        Plate_normal[v][i] = sqrt((N[v][i][0] * N[v][i][0]) +
(N[v][i][1] * N[v][i][1]) + (N[v][i][2] * N[v][i][2]));

        Plate_area[v][i] = Plate_normal[v][i]/2.0;

    //Plate centroid definition for each plate [i] of layer [v].

        Cen[v][i][0] = (Vsurf[k][0] + Vsurf[l][0] + Vsurf[m][0])/3.0;
        Cen[v][i][1] = (Vsurf[k][1] + Vsurf[l][1] + Vsurf[m][1])/3.0;
        Cen[v][i][2] = (Vsurf[k][2] + Vsurf[l][2] + Vsurf[m][2])/3.0;

    //We find the spherical coordinates of the plate centroid in
radians, for use in calculating spherical harmonics.

        Cen_position[v][i][0] = sqrt( ( Cen[v][i][0] * Cen[v][i][0] )
+ ( Cen[v][i][1] * Cen[v][i][1] ) + ( Cen[v][i][2] * Cen[v][i][2] ) );
        Cen_position[v][i][1] = atan2( Cen[v][i][1], Cen[v][i][0] );
        Cen_position[v][i][2] = acos( Cen[v][i][2] /
Cen_position[v][i][0] );

    }

    //Volume of shell and center of mass coordinates, for use in
testing. Formulas from Cheng et al.
```

APPENDIX C (CONTINUED)

```
   V = 0;

    for (i = 0; i < N_plates; i++)
    {
        V = V + ((Cen[v][i][0] * N[v][i][0]) + (Cen[v][i][1] *
N[v][i][1]) + (Cen[v][i][2] * N[v][i][2]))/6.0;

        Xcm[v][0] = (N[v][i][0] * (Cen[v][i][0] * Cen[v][i][0])) +
Xcm[v][0];
        Xcm[v][1] = (N[v][i][1] * (Cen[v][i][1] * Cen[v][i][1])) +
Xcm[v][1];
        Xcm[v][2] = (N[v][i][2] * (Cen[v][i][2] * Cen[v][i][2])) +
Xcm[v][2];
    }


    Xcm[v][0] = Xcm[v][0] / (4*V);
    Xcm[v][1] = Xcm[v][1] / (4*V);
    Xcm[v][2] = Xcm[v][2] / (4*V);


  //  cout << sqrt( ( Xcm[v][0] * Xcm[v][0] ) + ( Xcm[v][1] *
Xcm[v][1] ) + ( Xcm[v][2] * Xcm[v][2] ) ) <<  "\n";

return (0);

}



double PotentialCalc(int test, GAGenome&d)
{

    GARealGenome& ShapeGenome = (GARealGenome&)d;

    double U[N_plates]; //Gravitational potential
    double A[3] = {0};
    double B = 0;
    double G = (6.67e-11)/4; //Gravitational constant (why divide by
4? - see Cheng appendix)
    double Density[layers];

//We now calculate the gravitational potential at each centroid of the
reference sphere plate model (x) in relation to each plate centroid of
the internal layers. This is set up such that it runs all the
```

APPENDIX C (CONTINUED)

```
calculations at one point on the reference sphere for all layers and
points, records it as U[plate], and then continues to the next
reference centroid and repeats.

for (v=0; v<layers; v++)
{

    if (test == 0)
    {
        Density[0] = ShapeGenome.gene(2);              //If
intermediate axis is included, should be ShapeGenome.gene(3).

        if (v > 0)
        {
        Density[v] = (ShapeGenome.gene(((v+1)*3)-1)) -
(ShapeGenome.gene((v*3)-1));         //If intermediate axis is
included, step factor should be 4, not 3.
        }
    }

    //note that Density[v] is calculated as a contrast between layers.

    if (test == 1)
    {
        Density[0] = 1000;
        Density[v+1] = 100;
    }
//test function for arbitrary increasing density.


    if (test == 2)
    {
        Density[v] = 2161;
    }
//test function for uniform bulk density.



}


for (i = 0; i < N_plates; i++)
{
    Potential[i] = 0;
```

APPENDIX C (CONTINUED)

```
    for (v=0; v<layers; v++)
     {
         U[i] = 0;

         for (s = 0; s < N_plates; s++)
         {
             A[0] = (Ref_Cen[i][0] - Cen[v][s][0]);
             A[1] = (Ref_Cen[i][1] - Cen[v][s][1]);
             A[2] = (Ref_Cen[i][2] - Cen[v][s][2]);

             B = sqrt( (A[0] * A[0]) + (A[1] * A[1]) + (A[2] * A[2]) );

             double X, Y, Z;

             X = (A[0] * N[v][s][0]);
             Y = (A[1] * N[v][s][1]);
             Z = (A[2] * N[v][s][2]);

             U[i] = U[i] + (( X + Y + Z ) / B);
//sums up gravitational potential contribution of the plate. Sqrt is
to get absolute value of A.

         }

     Potential[i] = Potential[i] + (U[i] * G * Density[v]);
//Gravitational potential of individual plate

     Potential[i] = Potential[i]*-1;

     }

}

  /*  double Om = 952.1532; // Om^2 used to calculate centrifugal
potential - Park et al 2016
    double rot;
    double q;
    double GM;
    double J2;
    double f;

    Om = Om / 86164;
    Om = Om * (pi/180);     //converting Om from Park et al 2016 from
degree/day to rad/sec
```

```
    rot = Om*Om*(483.1*481.0*445.9);

    GM = (6.67e-11 * V * 2161);      //GM of Ceres

    q = rot/GM;     //see Murray and Dermott eq. 4-102
//J2 from the rotational parameters.

    J2 = -q/2;

    cout << "Analytical J2 (homogenous body): " << J2 << "\n";
 */

    return(0);
}



        //This section calculates the spherical harmonic coefficients
from the gravitational potential. Problems are cast according to
calculations from Heiskanen and Moritz (1967)section 1-14, adapted by
Dombard. Coefficients are calculated up to degree 2.

double SphericalHarmonics(int test)
{

    C2[0] = 0;
    C2[1] = 0;
    C2[2] = 0;
    S2[0] = 0;
    S2[1] = 0;

    double P2[3];

    if (test == 3)

    {
        int position = 0;
        ifstream in("/Users/seanohara/Desktop/J2test.txt");

        while (!in.eof() && position < N_plates)
        {
            in >> Potential[position];
            position++;
        }
    }
// test function - input generated from a synthetic potential field.
```

APPENDIX C (CONTINUED)

```
for (int i=71; i<N_plates-143; i++)
  //  Excludes polar node points due to degeneracy issues.
    {

// Calculate associated Legendre polynomials for each degree and
order. This hard-codes for degree 2 only.

        P2[0] = 0.5 * ( 3.0 * pow( cos( Ref_Cen_position[i][2] ) ,2 )
- 1 );
        P2[1] = -3.0 * sin( Ref_Cen_position[i][2] ) * cos(
Ref_Cen_position[i][2] );
        P2[2] = 3.0 * pow( sin( Ref_Cen_position[i][2] ) ,2 ) ;

        Yc[0] = P2[0];
        Yc[1] = P2[1] * cos( Ref_Cen_position[i][1] );
        Yc[2] = P2[2] * cos ( 2.0 * Ref_Cen_position[i][1] );

        Ys[0] = P2[1] * sin ( Ref_Cen_position[i][1] );
        Ys[1] = P2[2] * sin ( 2.0 * Ref_Cen_position[i][1] );

    if (test == 0 || test == 3 )
        {
        C2[0] = C2[0] + Potential[i] * (Ybar[0] * Yc[0]) *
Unit_area[i];
        C2[1] = C2[1] + Potential[i] * (Ybar[1] * Yc[1]) *
Unit_area[i];
        C2[2] = C2[2] + Potential[i] * (Ybar[2] * Yc[2]) *
Unit_area[i];
        S2[0] = S2[0] + Potential[i] * (Ybar[1] * Ys[0]) *
Unit_area[i];
        S2[1] = S2[1] + Potential[i] * (Ybar[2] * Ys[1]) *
Unit_area[i];
        }

    if (test == 1)
        {
         Potential[i] = 1;   // test function
         C2[0] = C2[0] + Potential[i] * Unit_area[i];
         C2[1] = C2[1] + Potential[i] * Unit_area[i];
         C2[2] = C2[2] + Potential[i] * Unit_area[i];
         S2[0] = S2[0] + Potential[i] * Unit_area[i];
         S2[1] = S2[1] + Potential[i] * Unit_area[i];
        }
//test functions for unit sphere -- total when Potential = 1 is 4pi.
```

```
    if (test == 2)
        {
        C2[0] = C2[0] + ((Ybar[0] * Yc[0]) * (Ybar[0] * Yc[0])) *
Unit_area[i];
        C2[1] = C2[1] + ((Ybar[1] * Yc[1]) * (Ybar[1] * Yc[1])) *
Unit_area[i];
        C2[2] = C2[2] + (((Ybar[2] * Yc[2]) * (Ybar[2] * Yc[2])) *
Unit_area[i]);
        S2[0] = S2[0] + ((Ybar[1] * Ys[0]) * (Ybar[1] * Ys[0])) *
Unit_area[i];
        S2[1] = S2[1] + ((Ybar[2] * Ys[1]) * (Ybar[2] * Ys[1])) *
Unit_area[i];
        }
//test functions for Yc/Ys terms -- should equal 4pi.
    }


    if (test == 0 || test==3)
    {
        C2[0] = C2[0] * 0.25 * pi;
        C2[1] = C2[1] * 0.25 * pi;
        C2[2] = C2[2] * 0.25 * pi;

        S2[0] = S2[0] * 0.25 * pi;
        S2[1] = S2[1] * 0.25 * pi;
    }
//once coefficients have been integrated for every plate, multiply the
result by 1/4pi.


for (i=0; i<N_plates; i++)
    {
    Potential[i] = 0;
    }
//resets the gravity potential for the next plate model run.


    return (0);
}


int main(int argc, char** argv)
{
    // Here we define a random seed value for the GA, for testing
purposes. Inputting a defined seed value is not supported as of this
time (1/2019).
```

APPENDIX C (CONTINUED)

```
    cout.precision(8);
    int n = 0;


    unsigned int seed = 0;
    for(int ii=1; ii<argc; ii++)
    {
        if(strcmp(argv[ii++],"seed") == 0)
        {
            seed = atoi(argv[ii]);
        }
    }

    Parameters();
    GARandomSeed(seed);

        //Here we calculate various important constants and fix them
in memory. Future calls of PotentialCalc() and SphericalHarmonics()
will use these values.

    Platemodel(1.0, 1.0, 1.0, 1, 0);
//Generates the unit sphere for normalizing the spherical harmonics.

    for (int i=0; i<N_plates; i++)
        {
            Unit_area[i] = Plate_area[1][i];
        }
// Calculates unit sphere plate area.


    Platemodel(R0, R0, R0, 1, 0);
// Generates the reference sphere.

    for (int i=0; i<N_plates; i++)
        {
            Ref_Cen[i][0] = Cen[1][i][0];
            Ref_Cen[i][1] = Cen[1][i][1];
            Ref_Cen[i][2] = Cen[1][i][2];
            Ref_Cen_position[i][0] = Cen_position[1][i][0];
            Ref_Cen_position[i][1] = Cen_position[1][i][1];
            Ref_Cen_position[i][2] = Cen_position[1][i][2];
            Ref_area[i] = Plate_area[1][i];
        }

    Ybar[0] = sqrt( (4.0 + 1.0) );
    Ybar[1] = sqrt(( 2.0 * (4.0 + 1.0) * (factorial(1.0) /
factorial(3.0))));
    Ybar[2] = sqrt(( 2.0 * (4.0 + 1.0) * (factorial(0.0) /
factorial(4.0))));

// Normalizes the associated Legendre polynomials.


    //Test(TestValue);       //Input a command line value here to run
a test of the calculation modules. If no input, skips to main body of
function. Not implemented as of 1/2019.


        // Here we define the parameters of the shape model that the
genetic algorithm will vary. The alleles are part of an array that
will populate the genome and then be varied accordingly during the
algorithm's evolution. See GAlib documentation for more information on
the relationship between alleles, the genome, and the genetic
algorithm.

        // The first step is to define the alleles. The first four are
the axes of the body in km and a possible range of bulk densities in
kg/m3.

GARealAlleleSetArray ShapeAlleles;

    ShapeAlleles.add(1561e3, 1561e3);
 //   ShapeAlleles.add(inter, inter);      //  Random determination of
intermediate axis removed from the current J2 version of the code.
Still required input due to plate model, but needs to be same as major
axis in parameter file to work.
    ShapeAlleles.add(1557e3, 1557e3);
    ShapeAlleles.add(900, 1100, 10);

        // Outer shell parameters. Each internal layer must be within
the overall body axes. The size varies in increments of a kilometer
from a minimum of 10 km.


    ShapeAlleles.add(1200e3, 1560e3, 1e3);
   ShapeAlleles.add(1200e3, 1556e3, 1e3);
    ShapeAlleles.add(900, 1100, 10);     //first internal volume
(minimum water thickness; water only for 3 layer model)

    ShapeAlleles.add(500e3, 1500e3, 1e3);
    ShapeAlleles.add(500e3, 1500e3, 1e3);
```

APPENDIX C (CONTINUED)

```
  ShapeAlleles.add(2500, 4000, 10);     //second internal volume

    ShapeAlleles.add(150e3, 1000e3, 1e3);
    ShapeAlleles.add(150e3, 1000e3, 1e3);
    ShapeAlleles.add(3800, 8000, 10);     //third internal volume


        //Here we set the parameters for the genome (which fleshes out
the population). See GAlib documentation for description of these and
the following section's parameters.

GARealGenome ShapeGenome(ShapeAlleles, ShapeObjective);
    GAParameterList ShapeParams;
    GASteadyStateGA::registerDefaultParameters(ShapeParams);
    ShapeParams.set(gaNnGenerations, 1000);
    ShapeParams.set(gaNpopulationSize, 1);
    ShapeParams.set(gaNscoreFrequency, 10);
    ShapeParams.set(gaNflushFrequency, 50);
    ShapeParams.set(gaNselectScores, (int)GAStatistics::AllScores);
    ShapeParams.parse(argc, argv, gaFalse);

        //Here we set the parameters for the genetic algorithm that
will evolve the system. We use a steady state algorithm with
overlapping populations.

GASteadyStateGA shape(ShapeGenome);
    GAGenome::SexualCrossover sexual;
    shape.parameters(ShapeParams);
    shape.scaling(GANoScaling());
    shape.pReplacement(0.5);
    shape.pCrossover(0.9);
    shape.pMutation(0.2);
  //  shape.terminator(GADCrowdingGA::TerminateUponConvergence);

        //Here we manually step the GA through the evolution process
using the initialize() and step() functions. We do this stepwise
(rather than using the evolve() function) so that at each step the
best member of the shape model population is re-run through the plate
model, gravity potential, and spherical harmonic modules and
displayed. This is to help the user verify that the evolution is
occurring. Once the code is in a more secure state, consider re-
writing this section using the evolve() function and then doing the
module calculations at the end before flushScores() in order to save
memory/processing time.
```

APPENDIX C (CONTINUED)

```
        //The fitness assessment of each population member and its
associated gravity and spherical harmonic values are handled by
ShapeObjective() which the algorithm reference each time it calls
step(). The genome is only varying the axes and density of each
overlapping ellipsoid.

    shape.initialize();
    cout << "Genome initialized" << "\n";

    while (shape.done() != true)
        {
            double a[layers];
            double c[layers];
            double densitycontrast[layers];
            double ellipsoidvolume[layers];
            double ellipsoidmass[layers];
            double bodymass = 0;


            shape.step();

            ShapeGenome = shape.population().best();

            int i = 0;

            for (v = 0; v < layers; v++)
                {
                    Platemodel(ShapeGenome.gene(i),
ShapeGenome.gene(i), ShapeGenome.gene(i+1), v, 1);
                    i = i+3;
                }
//If intermediate axis needs to be calculated, second input should be
i+1 and i should increment by 4.


            PotentialCalc(0, ShapeGenome);
            SphericalHarmonics(0);

            bodymass = 0;
            i=0;

            for (v=0; v < layers ; v++)
            {
                a[v] = ShapeGenome.gene(i);
                c[v] = ShapeGenome.gene(i+1);
```

APPENDIX C (CONTINUED)

```
                densitycontrast[v] = (ShapeGenome.gene(i+5) -
ShapeGenome.gene(i+2));
                i = i+3;
            }
            //generate the ellipsoid mass and density contrast for the
internal layers. Note that densitycontrast[v] represents the contrast
between the *previous* layer; when calculating, you must use
density[v-1] to correctly assign the contrast to the associated
ellipsoid.

            ellipsoidvolume[0] = 1.333333 * pi *a[0]*a[0]*c[0];

            ellipsoidmass[0] = ellipsoidvolume[0] *
(ShapeGenome.gene(2));

            //generate the ellipsoid mass for the outer layer (no
density contrast)

            bodymass = ellipsoidmass[0];


            for (v=1; v < layers; v++)
            {
                ellipsoidvolume[v] = (1.33333*pi*a[v]*a[v]*c[v]);

                ellipsoidmass[v] = ellipsoidvolume[v] *
densitycontrast[v-1];

                bodymass = bodymass + ellipsoidmass[v];
            }
            //total up the body's mass

            double bulkdensity;
            double bulkvolume;

            bulkvolume = ellipsoidvolume[0];
            bulkdensity = (bodymass / bulkvolume);

            cout << "Best of generation " << n << ":" "\n";
            cout << "C20: " << C2[0] << "\n";
            cout << "C22: " << C2[2] << "\n";
            cout << "Mass: " << bodymass << "\n";
            cout << "Bulk Density: " << bulkdensity << "\n";
```

APPENDIX C (CONTINUED)

```
            cout << "Shape Parameters: " << shape.population().best()
<< "\n";

               n++;


        }

        //Output the result of the GA and clear the memory. Right now
the genome spits out the axes and density of each shell basically
unformatted into the terminal window. In the future, upgrade this to a
formatted output file.

    shape.flushScores();

    return(0);

}




float
ShapeObjective(GAGenome & d)
{

    GARealGenome& ShapeGenome = (GARealGenome&)d;

    float ShapeValue = 0.0;
    double C22Error = 0.0;
    double C20Error = 0.0;

    int i = 0;

    for (v = 0; v <= layers; v++)
    {
        Platemodel(ShapeGenome.gene(i), ShapeGenome.gene(i),
ShapeGenome.gene(i+1), v, 1);
        i = i+3;
    }
//If intermediate axis needs to be calculated, second input should be
i+1 and i should increment by 4.


    PotentialCalc(0, ShapeGenome);
```

APPENDIX C (CONTINUED)

```
    SphericalHarmonics(0);


    i = 0;
    int score = 0;

    C22Error = trueC22 - C2[2];
    ShapeValue += 1-abs(C22Error);
//increases the value of a solution by a value based on its precision
compared to the actual C22 value.

    C20Error = trueC20 - C2[0];
    ShapeValue += 1-abs(C20Error);
//increases the value of a solution by a value based on its precision
compared to the modeled C20 value. C20 = -J2.

    //This section calculates the volume of each ellipsoid and its
density based on the density contrast, then sums up the total mass of
the body and compares it to the actual. The value of the solution
increases based on its precision compared to the actual mass value.

    double a[layers];
    double c[layers];
    double densitycontrast[layers];
    double ellipsoidvolume[layers];
    double ellipsoidmass[layers];
    double bodymass = 0;
    double MassError;
    i=0;

    bodymass = 0;
    i=0;

    for (v=0; v < layers ; v++)
    {
        a[v] = ShapeGenome.gene(i);
        c[v] = ShapeGenome.gene(i+1);
        densitycontrast[v] = (ShapeGenome.gene(i+5) -
ShapeGenome.gene(i+2));
        i = i+3;
    }
    //generate the ellipsoid mass and density contrast for the
internal layers. Note that densitycontrast[v] represents the contrast
between the *previous* layer; when calculating, you must use
```

APPENDIX C (CONTINUED)

```
density[v-1] to correctly assign the contrast to the associated
ellipsoid.

    ellipsoidvolume[0] = 1.333333 * pi *a[0]*a[0]*c[0];

    ellipsoidmass[0] = ellipsoidvolume[0] * (ShapeGenome.gene(2));

    //generate the ellipsoid mass for the outer layer (no density
contrast)

    bodymass = ellipsoidmass[0];


    for (v=1; v < layers; v++)
    {
        ellipsoidvolume[v] = (1.33333*pi*a[v]*a[v]*c[v]);

        ellipsoidmass[v] = ellipsoidvolume[v] * densitycontrast[v-1];

        bodymass = bodymass + ellipsoidmass[v];
    }

    MassError = (bodymass - trueMass) / trueMass;
    ShapeValue += 1-abs(MassError);
//increase the value of the solution if it is near the actual body
mass

    if ( (ShapeGenome.gene(i+2) > ShapeGenome.gene(i+5)))
//increases the value of a solution if density increases with depth.
        {ShapeValue++;};

    i = 0;
    score = 0;

    for (v=0; v< layers; v++)
    {
          if(     (ShapeGenome.gene(i) > ShapeGenome.gene(i+3))
                    &&
//increases the value of a solution *only* if each ellipsoid is
smaller than the preceding.
                    (ShapeGenome.gene(i+1) > ShapeGenome.gene(i+4))
                       &&
                        (ShapeGenome.gene(i) > ShapeGenome.gene(i+1))
//increases the value of a solution *only* if the major axis is larger
than the minor axis.
```

APPENDIX C (CONTINUED)

```
                        &&  (ShapeGenome.gene(i+1) >=
(ShapeGenome.gene(i)*.95))
//increases the value of a solution *only* if the a minor axis is at
least 95% of the major axis.
             )
                        {score++;}

        i = i+3;
    }

    if (score >= layers)
    {ShapeValue++;}
    else ShapeValue = 0;
//if none of the above conditions are met, the value of the solution
is 0, since it would be unphysical or would throw off the gravity
solution. Note that the increase of density with depth is only for
this model of Europa.

    if (ShapeValue < 0)
    {ShapeValue = 0;};

    return (ShapeValue);
}
```

VITA

Education

University of Illinois at Chicago, Chicago IL

Fall 2013-Present

Doctoral Candidate in Planetary Science

- Advisor: Andrew Dombard

Northeastern Illinois University, Chicago IL

B. S., May 2013

B.S. Earth Science, Minor: Chemistry, Magna Cum Laude

- Advisor: Genet Ide Duke

Bowdoin College, Brunswick ME

B. A., May 2005

Major: East Asian Studies

- Junior year abroad in Kyoto, Japan (Associated Kyoto Program)

Research Experience

Lunar and Planetary Institute Center for Lunar Science and Exploration

Exploration Science Intern

Summer 2015

- Principal Investigator: David Kring
- Analyzed proposed geologic traverses for robotic sample return from Schrödinger basin on lunar farside

Stable isotopes of carbon and oxygen in carbonatites from Proterozoic to Recent

2011-2012

- Principal Investigator: Genet Ide Duke
- Performed stable isotope analysis and mineral content characterization on samples of carbonatite, in an effort to determine if there has been a systematic change in fractionation over geologic time

Publications

A.J. Dombard, S. O'Hara. *Planetary science: Pluto's polygons explained.* Nature 534, 40-41. News and Views article.

Presentations

S. O'Hara and A. Dombard. *Simulating formation of Triton's cantaloupe terrain by compositional diapirs.* Poster presentation. Lunar and Planetary Institute, Houston TX, 2019.

S. O'Hara and A. Dombard. *Downhill sledding at 40 AU: mobilizing Pluto's chaotic mountain blocks.* Oral presentation. Lunar and Planetary Institute, Houston TX, 2018.

A. Calzada Diaz et al. *Analyses of robotic traverses and sample sites for the HERACLES human-assisted sample return mission concept.* Oral presentation. Lunar and Planetary Institute, Houston TX, 2015.

S. O'Hara, Y. Herrera, G.I. Duke. *Stable isotopes in carbonatites and associated alkali igneous rocks: constraints on evolution of deep mantle carbon.* Oral presentation. NEIU Student Center for Science Engagement Summer Symposium, Chicago IL, 2012.

Employment

University of Illinois at Chicago, Chicago IL

2013-Present

- Teaching Assistant
    - Andrew Malone (Global Environmental Change, Spring 2019)
    - Stefany Sit (Geophysics, Fall 2015; Earth, Energy, and the Environment, Spring 2017 & Fall 2018)
    - Sarah Cadieux (Earth Materials, Fall 2016)
    - Steve Guggenheim (Mineralogy, Spring 2016)
    - Andrew Dombard (Current Topics in Earth and Environmental Science, Fall 2015)
    - Fabien Kenig (Field Work in Missouri, Spring 2015)
    - Carol Stein (Structural Geology and Tectonics, Fall 2014)
    - Peter Doran (Field Work in Missouri, Spring 2014)
    - D'arcy Meyer-Dombard (Global Environmental Change, Fall 2013)
- Research Assistant
    - Advisor: Andrew Dombard

Employment (continued)

Northeastern Illinois University, Chicago IL

2011-2013

- Tutor in Earth Science

Awards and Honors

Graduate

- Outer Planets Assessment Group Early Career Participant travel grant
- Illinois Space Grant Consortium Graduate Fellowship (2016, 2017)
- Outstanding Teaching Assistant Award (Spring 2015)

Undergraduate

- National Science Foundation STEM Scholarship in Math and Physical Sciences
- College of Liberal Arts and Sciences PhD Student Travel Award
- Geological Society of America North Central Regional Conference 2012 travel grant
- Astrobiology Science Convention 2012 travel grant
- SACNAS National Conference 2011 and 2012 travel grants
- Outstanding Student in Earth Sciences award

Meetings and Workshops Attended

- Outer Planets Assessment Group Spring 2019 Meeting
- PSD ROSES review panel secretary
- Lunar and Planetary Science Conference 2015, 2017, 2018
- Workshop on the Study of the Ice Giant Planets, Johns Hopkins Applied Physics Laboratory, 2014
- Geological Society of America North Central Regional Conference, 2012
- Astrobiology Science Convention, 2012
- SACNAS National Conference, 2011 and 2012