Modeling the Thermal and Chemical Evolution of the Martian Lithosphere Over Time

Fiona Clare McGroarty

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
In
Geosciences

Megan S. Duncan, Chair
Mark J. Caddick
Scott D. King
Nancy L. Ross

June 29, 2021
Blacksburg, VA, USA

Keywords: Mars, Geochemistry, Modeling

Modeling the Thermal and Chemical Evolution of the Martian Lithosphere Over Time

Fiona Clare McGroarty

ABSTRACT

Mars is an ideal planet to study planetary evolution and development, as its crust has been preserved over its history, rather than continuously recycled through subduction, as has happened on Earth. In order to attain a more coherent understanding of martian evolution, we focused on the thermal and petrologic history of the martian lithosphere. We developed a model that calculates the thermal state and melt composition of Mars over time. This model provides insight into the planet's history and enables us to describe how the density and seismic properties have evolved over time. We calculated the temperature profile through the lithosphere and then fit an equation to pre-existing experimental data in order to produce a model to predict the composition of melt produced as a function of pressure and temperature. From the melt model, we see a trend from ultramafic to mafic composition over time. We calculated the density and seismic properties of the lithosphere and found that they increase over time, but decrease with depth, which is consistent with the recent observations of NASA's InSight mission.

Modeling the Thermal and Chemical Evolution of the Martian Lithosphere Over Time

Fiona Clare McGroarty

GENERAL AUDIENCE ABSTRACT

Mars is an excellent location to study how planets change over time because its crust has remained intact, rather than being destroyed as segments of the crust move and push against each other, which happens on Earth. In order to understand how Mars has evolved over time, we built a model to show how the top part of the planet has changed over time. The model works by calculating the temperature of the rocks. We calculated these temperatures in the present day and at four, three, two, and one billion years ago. We took the temperatures and used them to calculate the elements that are present in the rocks. Knowing the chemistry of the crust made it possible for us to calculate the minerals present in the crust and upper mantle, which we used to calculate the density of the outer layers of Mars and the speed at which earthquake waves would travel through the layers. We found that the density and earthquake wave speeds decrease over the depth of the top part of the planet. Although usually an object that is denser at the top than bottom will flip over, we believe this will not happen on Mars because the rocks are thick enough to prevent them from flipping.

ACKNOWLEDGEMENTS

I would first and foremost like to thank Carol Franklin and my family, Anne, Patrick, Robert, Liam, and Moira, whose unwavering support and confidence have made this project possible, and to my fiancé, Jason Farry, for his total devotion and belief in me. I thank Dr. Scott Eaton, Dr. Christine May, Ben, and Eli Eaton, and all my other friends and family who believed in me, especially when I did not believe in myself. In addition, I am forever and always grateful to my Australian Terrier, GCHB Ch Shastakin Killara Larrikin RedUproar CAA RATO “Larry,” for his constant love and support. I am forever appreciative of and grateful to my advisor, Dr. Megan S. Duncan, and my committee, Dr. Mark Caddick, Dr. Scott King, and Dr. Nancy Ross, for their mentorship and support. I would also like to thank the Virginia Space Grant Consortium for supporting this research, Dr. Mark Caddick and Jonathan Prouty for their assistance with Perple_X, and Priyanka Bose for her assistance with the areotherm code. I do not thank the creators of the ESS system of Virginia Tech for their terrible website that lost my ETD upon my first submission in July, nor do I thank the people in the university who made the decision to make me choose between having my degree in my professional name or my married name by not allowing me to change my legal name while maintaining my maiden name on my university records.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

Mars (Latin). Ma'adim (Hebrew). Ma'rIS (Klingon). That little red point of light has captivated the human mind and imagination since the dawn of humanity. Today, we find ourselves transfixed anew by the Red Planet, as the National Aeronautics and Space Administration (NASA) continues to launch satellites, landers, and rovers to study the planet and as astronauts train and prepare for the first manned missions to Mars. Mars represents the curiosity of humanity, as well as an exciting and innovative future.

However, Mars is of interest to us for more than its inspiration and beauty. Mars is a unique planet, in many ways both like and unlike the Earth. Mars allows us a special insight into planetary science, both as an alternative perspective and as a comparison to the Earth. One of the most promising features of Mars in terms of planetary science is the long-term stability of its crust. On Earth, the crust, or the outermost layer of solid rock, is constantly recycled and reformed through the process of plate tectonics, in which sections of the Earth's crust (plates) are pulled across the surface by mantle (the layer beneath the crust, made of ductile solid rock) currents. The crust and uppermost layer of the mantle are thermally coupled into the lithosphere, which is distinguished from the mantle by the mode of heat transport; the lithosphere conducts heat while the mantle convects heat. The lithosphere itself is divided compositionally between the crust and the conductive mantle, forming two compositional layers within one thermal layer. The plates collide with each other, deforming the rocks and subducting one plate beneath another, push past each other, causing shear stress, and pull apart, allowing molten rock to flow to the surface and solidify into new crust. While this process has provided many valuable insights regarding the interior of the Earth, it has also destroyed the earliest rocks from the Earth's youngest days. In contrast, Mars has no plate tectonic process destroying and recycling its rocks over time, therefore the crust has remained intact, with new layers of lava erupting while the rocks already there remain preserved. Another notable difference between the lithospheres on Earth and Mars is that Earth's lithosphere has two types: the oceanic and continental lithospheres, whose differences are in composition, thickness, density, seismic velocity, and structure. These distinct lithospheric differences are due to plate tectonic processes. Conversely, Mars does not have these distinct continental and oceanic differences, however, it does have highlands regions in the southern hemisphere and lowlands regions in the northern hemisphere. This yields to different lithospheric thicknesses between the Southern Highlands and the Northern Lowlands; the Southern Highlands have lithospheric thicknesses as thick as ~100 km, while the Northern Lowlands have lithospheric thicknesses as thin as ~3 km (Taylor & McLennan, 2009a). Because the martian surface is made of basalt, similar to Earth's ocean floor, we consider it compositionally most similar to Earth's oceanic lithosphere; however, the thickness of the martian lithosphere is more similar to the thickness of Earth's continental lithosphere.

In order to construct a model of how Mars has evolved over time, we use the ancient rock layers still emplaced within the lithosphere. By knowing how Mars has evolved, we broaden our understanding of planetary evolution, which helps us to understand other planets, including the Earth. Knowing how planets evolve will enable us to compare the Earth's own history to its neighbors in the solar system. It also allows us to predict if a planet may be "dead" inside, that is, incapable of producing any molten material and thus volcanoes. Because volcanoes are a major contributing factor to atmospheric development and maintenance, knowing the potential for active volcanism on a planet will provide considerations should plans to colonize another planet progress. In the nearer future, NASA has invested in landers and rovers to study the martian interior, and our research supports that mission by providing a model for how the martian crust and mantle lithosphere, or top section of the mantle, has evolved into the observable modern interior.

We built our model with the crust and mantle lithosphere which are formed from the mantle beneath it. The mantle melts to produce the crust, while the residual, unmelted material formed the mantle portion of the lithosphere. In our model, we calculated an areotherm, or the temperature of the rocks in the crust and mantle lithosphere as a function of depth. This temperature is vital because it reveals the planet’s capacity to produce magma, here also referred to as melt. We used the adiabat, or the temperature through the conductive mantle, to determine the temperature and corresponding pressure at the base of the lithosphere; where the temperatures of the areotherm and adiabat intersect is the depth and pressure of the base of the lithosphere. We also used the martian solidus, which defines if a material is solid or partially molten, to determine the pressure at which melting begins; if the adiabat cross the solidus, then we know that there is melt being produced. We used those parameters to calculate the thickness of the martian crust over the past four billion years and to calculate the amount of melt that was produced over time. We then calculated the chemical composition of the melt, i.e., how much silica, iron, magnesium, etc., was in the molten material. Knowing those chemical compositions allowed us to determine the minerals that the melt crystallized into, from which we determined the type of rock within the crust and mantle lithosphere and calculate the density and seismic velocity, or the speed that earthquake waves travel through, over the depth of each layer, over the past four billion years. Knowing the densities and seismic velocities helps us understand the structure of the martian lithosphere, as well as how that structure has changed over time.

Chapter 2

A manuscript for JGR Planets

2.1 Introduction

There are currently few constraints on the thermal and geochemical evolution of Mars, which plays a direct role in its tectonic and volcanic history. The thermal evolution of the planet controls the amount and duration of mantle melting, which is directly linked to crust generation (e.g., McSween et al., 2009). Current knowledge of Mars is limited to its surface conditions, i.e., data from orbital scans (spectroscopy), surface measurements (rovers and landers), and meteorite studies, but provide key information about the crust. The crustal parameters are vital because we know that the crust is formed from melt produced within the mantle; the basaltic composition indicates an ultramafic/mantle origin of crust material. Thus, by studying the crust, we can infer some of the properties of the mantle. We also know that the most recent lava flows occurred within about 1 Ma, which indicates the potential for localized melting today, and from crater counting we know that the oldest surfaces are over 4 Gy old (Hartmann, 2005; Horvath et al., 2021; Moitra et al., 2021; Werner, 2009). Because the recent lava flows indicate potential for melt in the present day, the melt production over time and in the present day was the focus of this study.

2.1.1 Thermal Constraints

Previously calculated thermal history includes mantle potential temperatures (T_P) derived from measurements of martian meteorites and surface basalts (at Gusev Crater, Meridiani Planum, and Gale Crater), under the assumption that these were formed by mantle melting. The calculated mantle T_P decreased from ~1723 to ~1648 K over the past 4.5 Gy (Filiberto, 2017), indicating overall mantle cooling with time. A similar study (Baratoux et al., 2011) used orbitally-derived surface compositions, from the GRS instrument on Mars Odyssey, and found a similar decrease in mantle T_P with time. These values range from ~1643–1693 K for the older Southern Highlands to ~1613–1673 K for the younger Tharsis and Elysium volcanics. This study also estimated mantle melt fractions and surface heat flows based on their calculated T_Ps which decreased from ~20 to ~5% from ~38 to ~26 mW/m^2 from old to young surfaces, respectively. Other observational-based constraints (i.e., GRS plus gravity and topography) on the heat flow over time also show a decrease, from about 50 mW/m^2 4.5 Ga to 14–25 mW/m^2 (average 20 mW/m^2) in the present day, with uncertainties in the ±1-6 mW/m^2 range (McGovern et al., 2002, 2004; Parro et al., 2017). Parro et al. (2017) recognized that the average heat flow will vary with location based on the presence or absence of mantle plumes. Geodynamic model estimates that include heat production within the interior of the planet, estimate that the surface heat flow in the present day ranges from 17–50 mW/m^2 (average 25 mW/m^2), indicating a range of surface heat flows across the planet (e.g., Plesa et al., 2016). In Plesa et al. (2016), the largest uncertainty in the heat flow was due to uncertainty in the size of the martian core; previous constraints on the martian interior left uncertainty on the size, phase (solid or liquid), and composition (amount of sulfur) in the core. The current NASA InSight mission used seismic studies and recently determined that the martian core is liquid and predicted the radius to be ~1840 km (Stähler S. C., 2021), similar the estimates of Khan et al. (2018). These results will change the surface heat flows above, but this is outside the scope of this study and we will use the current average values to constrain our models; InSight to date has been unable to measure the heat flow through the crust (Grott et al., 2021).

2.1.2 Geochemical Constraints

A key constraint of surface heat flow calculations is the abundance of heat producing elements (HPEs), which for the average martian crust today are 0.18 ppm uranium, 0.7 ppm thorium, and 3740 ppm potassium (Taylor & McLennan, 2009b). The previous studies mentioned above assumed homogenous distribution of these elements throughout the crustal layer along with a homogenous mantle HPE distribution (Dreibus & Wänke, 1985). The current InSight mission also calculated ranges for crustal enrichment (HPEs) and found that in order to match the melt volume today (i.e., melting only at Tharsis), crustal enrichments (Λ) should vary from 20–22 for the two layer model and 11–14 for the three layer model. This leads to Th concentrations of 1200–1700 ppb or 680–1050 ppb for the two and three layer model, respectively, with the caveat that the two layer model would require an enriched HPE layer in the lower crust (Michaut, 2021). Therefore, the three layer model is preferred. Maximum crustal densities were determined to be 2900 kg/m^3 for the two layer model and 3100 kg/m^3 for the three layer model, with a minimum of 2550 kg/m^3, which are similar estimates based off of previous Moment of Inertia constraints (Sohl et al., 2005). They note that these values are less than the calculated densities of the shergottites, and several previous estimates for the bulk crust (e.g., Goossens et al., 2017), but are not unreasonable considering impart gardening porosity, the presence of widespread clays, and water ice that are likely present for the first few kilometers of Mars' crust (e.g., Baratoux et al., 2014).

In addition to the HPE concentration, the thickness of the crustal layer will control heat distribution, where the average crustal thickness today is ~50 km, with variation from ~3 to >100 km (e.g., Neumann et al., 2004; Wieczorek & Zuber, 2004). In the Noachian, 4 Ga, the average crustal thickness was ~20 km (Norman, 1999). The InSight mission refined this global average range based on seismic observations to 24–40 km if the crust is dominated by two seismically distinct layers, and 35–72 km if three layers. They further detected layer boundaries at 10 km and 22 km marked by sharp increases in seismic velocities (Lognonné et al., 2020). They also determined the lithosphere thickness to be ~500 km at the InSight location that is characterized by a negative slope in V_S and neutral to slightly positive slope in V_P (Khan A., 2021). We compared our model results to InSight's, which allowed us to place constraints on the accuracy of our models over time.

Here we produced a self-consistent geochemical-thermal model that places constraints on the evolution of the martian lithosphere, using the constraints described above. We used a combination of areotherm, melt, and thermodynamic calculations to construct the martian crust and mantle lithosphere through time. The result of this model is a model of heat flow through the martian lithosphere over time, the global average melt produced over time, the composition of the melts, and the mineral composition and corresponding density and seismic properties.

2.2 Methods

Using the previously determined constraints, we built a simplified geochemical model moving toward a self-consistent geochemical-thermal model for Mars. The geochemical model consisted of several parts: l) areotherm calculations, 2) construction of a melt model based on previous experiments, and 3) thermodynamic modeling to determine ρ, V_P, and V_S profiles through the crust and mantle lithosphere. All calculations represent global averages, none of the geochemical or geodynamic thermal profiles are intended to represent any specific location on Mars.

2.2.1 Areotherms

The first stage involved calculating areotherms, or temperature profiles through the conductive crust and mantle lithosphere. The areotherm calculations were built off of well-established parameters, originally designed for the Earth, with the appropriate modifications for Mars. Areotherms were calculated after Rudnick et al. (1998) according to the equation:

$$T(z) = T_0 + {q_i \Delta z_i}/{k_i} - {A_i \Delta {z_i}^2}/{2k_i} \tag{1}$$

where T_0 is the surface temperature (~220 K), q is the heat flow in mW/m^2, Δz is the depth interval (100) in m, k is thermal conductivity in W/mK, and A is the heat production from HPEs (U, Th, and K, in W/m^3). Heat flow was calculated as a function of depth using:

$$q_{i+1}(z) = q_i - A_i \Delta z \tag{2}$$

thermal conductivity of the crustal layers was calculated after Chapman (1986):

$$k(z) = {k_0(1 + 0.0015 z_i)}/{1 + 0.0001(T - 273.15)} \tag{3}$$

and thermal conductivity of the mantle lithosphere was calculated after Schatz and Simmons (1972) using the lattice conductivity (W/mK):

$$k_L = {1}/{7.41 \times 10^{-2} + 5.02 \times 10^{-4} T} \tag{4}$$

or

$$k_{L\,min} = 1.26(1 + z) \tag{5}$$

and the radiative conductivity (W/mK):

$$k_R = 2.3 \times 10^{-3}(T - 500) \tag{6}$$

under the constraints that if $T \leq 500$ K, $k = k_L$; and if $T > 500$ K, k = the larger of k_L or $k_{L\ \mathrm{min}} + k_R$. As k is dependent on temperature, we used an initial estimate for k of 2.5 W/mK in the crust and 4 W/mK in the lithosphere (Kiefer, 2003) to calculate an initial temperature, which was iterated back into the calculation for k in order to produce a new calculation of temperature. We iterated this process in order to achieve consistency between the calculated k and temperature values. The areotherms were combined with the adiabat calculations from Filiberto (2017); the intersection of the areotherm and adiabat marked the base of the thermal lithosphere. These calculations for the areotherm and adiabat were performed at one billion year intervals from 4 Ga to the present day (0 Ga).

First, we constructed a reference model in MATLAB that consisted of a single layer with homogenous HPE distribution (Taylor & McLennan, 2009b), and a 50% depleted mantle lithosphere relative to Dreibus and Wänke (1985). The calculated crust layer grew linearly with time from 20 km to 50 km, and the surface heat flow decreased linearly with time from 50 mW/m^2 (McGovern et al., 2002) to 25 mW/m^2 (Parro et al., 2017; Plesa et al., 2016). Given the overall uncertainty in the thermochemical history of Mars, this model serves as a conservative estimate of lithospheric evolution. All input parameters are listed in Table 2-1.

In order to account for mantle melting through time, we modified the reference model such that at each time step melt fraction (F) was calculated using:

$$F = \left[\left(\frac{dT}{dP_{solidus}} - \frac{dT}{dP_{adiabat}}\right) / \left(\frac{\Delta H_F}{C_P} + \frac{dT}{dF}\right)\right](P_0 - P_F) \tag{7}$$

Table 2-1. Inputs used to produce areotherms.

Input Parameter	Symbol		Value	Units	Reference
Crustal Thickness	z	0 Ga	50	km	Wieczorek and Zuber, 2004
		4 Ga	20	km	Norman, 1999
Surface Heat Flow	q_0	0 Ga	20	mW/m^2	Parro et al., 2017
		4 Ga	50	mW/m^2	Baratoux et al., 2011
Surface Temperature	T_0		220	K	
Thermal conductivity	k	Crust	2.5	W/mK	Kiefer, 2003
		Mantle	4	W/mK	Kiefer, 2003
Density	ρ	Crust	2900	kg/m^3	
		Mantle Lithosphere	3400	kg/m^3	
		Mantle	3700	kg/m^3	
Crust HPEs		U	18	ppb	Taylor and McLennan (2009)
		Th	0.7	ppb	
		K	3740	ppm	
Mantle HPEs		U	16	ppb	Dreibus and Wanke (1985)
		Th	56	ppb	
		K	359	ppm	
Adiabat Gradient			0.18	deg/km	Filiberto (2017)
Planetary Radius	rad		3390	km	

where P_0 is the pressure in GPa where the solidus intersects the adiabat, which indicates the initiation of melt production, P_F is the pressure at the cessation of melt production and is marked by the intersection of the adiabat and the areotherm, $dT/dP_{solidus}$ is the slope of the solidus (~106.15 K/GPa over the pressure range here; Duncan et al., 2018), $dT/dP_{adiabat}$ is the slope of the adiabat (0.18 K/km; Filiberto, 2017), ΔH_F is the enthalpy of fusion (6.4x105 J/kg; Kiefer, 2003), C_P is the heat capacity (1200 J/K kg; Kiefer, 2003), and dT/dF is the change in temperature as a function of melt fraction and was calculated from previous experimental data (Collinet et al., 2015; Matsukage et al., 2013a), described below and ranged from 4.38 to 3.9 K/F depending on pressure.

The resulting melt fraction was then used to determine the distribution of HPEs between the "crust" and "depleted mantle lithosphere" using bulk rock/melt partition coefficients (Beattie, 1993; Davis et al., 2011). This was done working under the assumption that all of the melt became crust and that all of the residue became depleted mantle lithosphere, and was constrained by the present day average crustal thickness of 50 km. We constructed two versions of the melt model, one in which each melt pulse created a new crustal layer at the surface (referred to as "Melt-Layer"), and one where the HPE distribution was averaged throughout the crustal layer (referred to as "Melt-Average"), weighted by the thickness at each time step. The latter scenario is closer to "reality" as not all melt generated in the mantle erupts. Further, given the large unknowns in melt eruptability on Mars (e.g., Lillis et al., 2009), an average is the preferred method of HPE distribution. For the initial set of models, we used a linearly decreasing surface heat flow, as in the reference case.

2.2.2 Melt model

In order to determine the dT/dF needed in Eq. 7, and to determine major oxide melt composition with time (in addition to HPEs), we used previous experimental results for mantle melting in Mars (Collinet et al., 2015; Matsukage et al., 2013a). These studies reported melt composition over a range T, P, and F, to which we used least square regression to fit a model of melt composition as a function of P and F (Fig. 1, Appendix A). We began by using the equation forms for peridotite melting from Duncan et al. (2017) for each oxide. The equation forms were modified on an oxide basis depending on if the experimental data indicated significance of the P terms. Using the above calculations of P_F and F over time, we calculated the composition of the melt at each time interval and used mass balance principles to calculate the composition of the residual un-melted material. We made the simplifying assumption that all melt produced was solidified into crust, while the residual, un-melted material formed the mantle lithosphere. From this, we calculated bulk crustal composition at each timestep using the thermodynamic modeling described below.

Our model is similar in method to the work done by Mckenzie and Bickle (1988), as both parameterized the melt fraction as a function of pressure and parameterized melt composition by melt fraction and pressure. McKenzie and Bickle (1988), however, incorporate the liquidus into their calculations; as Earth's liquidus is known INSERT CITATION, it is possible to build an Earth-based model incorporating the liquidus. However, the martian liquidus is still undetermined, and thus our model is designed to work independently of the liquidus. Because we know the martian solidus from INSERT CITATION, we built our model to rely on the intersection of the solidus with the areotherm to produce partial melt, while remaining unconcerned with the liquidus line of total melt. We based the among of melt in the partial melt on the outputs of the areotherm and based the melt composition equations on pressure and melt fraction, eliminating reliance on the liquidus. Our melt model is thus similar in parameterization to McKenzie and Bickel (1988), but without need to calculate and incoporate a martian liquidus.

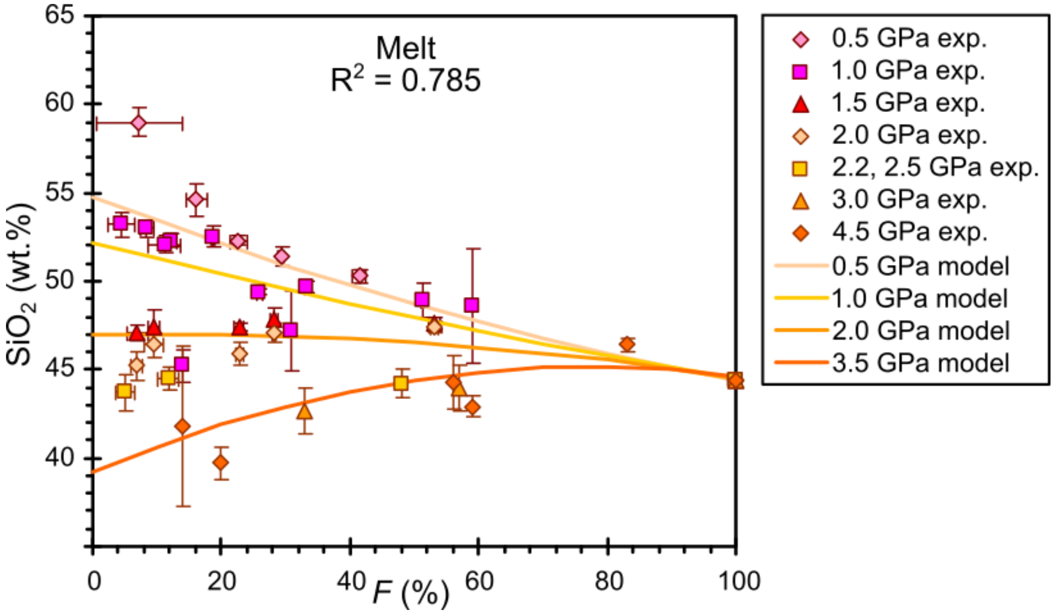

Figure 2-1. Previous experimental data (symbols) with our calculated regressions (lines) for SiO_2. The data and model are denoted by pressure in order to show how the model fits at each experimental pressure range. Percent of mass in melt phase is on the x-axis, while the wt.% of SiO_2 in the melt is on the y-axis. Other oxide plots and fit equations are provided in Appendix A.

2.2.3 Mineralogical calculations

We used the Gibbs Free Energy Minimization Software Perple_X (Connolly, 2009) to calculate density and seismic velocities for the crust and mantle lithosphere at each timestep for both the reference compositions (Dreibus & Wänke, 1985; Taylor & McLennan, 2009b) and our calculated melt and residual compositions. We used the hpha622.ver Holland and Powell (2011) dataset, updated from the Powell et al. (1998) dataset which includes the shear moduli from Abers and Hacker (2016), and corresponding solution models to calculate the mineral modes (olivine, clinopyroxene, orthopyroxene, spinel, and feldspar) and lithospheric properties over the *P-T* range of our areotherms. We selected this dataset due to its inclusion of the pressure ranges of our model, and the updated version specifically in order to more accurately calculate the seismic velocities. Due of the limitations of the Perple_X calculator at the low temperatures of the crust, we calculated crustal mineral modes at a fixed temperature of 800 K; we assumed that the mineral composition was crystalized upon cooling and did not change as it cooled. We then used those mineral modes to calculate the density and seismic velocities of the crust along the areotherm.

To calculate the density, we first calculated the mineral modes in the rock and the endmember modes of each mineral in the rock. To do this, we used the equations 2, 3, 12, and P_{th} from Holland and Powell (2011) and the equation of state parameters in the Perple_X data files to calculate the volume change for each mineral's endmember using least square regression to solve for the pressure of our areotherm. We then calculated the density of the mineral endmember as 1/V. We multiplied the mineral endmember mode by the calculated density to obtain the contribution of the density of the mineral endmember to the mineral. We then multiplied that contribution by the contribution of the mineral to the bulk rock to result in the contribution of the endmember's density to the total density of the crust. After we obtained this value, we then summed the contributions to the rock in order to obtain the total density of the crust.

$$\rho_{bulk\ rock} = \Sigma\left[\left(\frac{MW_{endmember}}{V_{endmember}}\right) * X_{endmember} * X_{mineral}\right] \qquad (8)$$

We calculated the seismic velocities by first calculating K_s and μ, using the thermodynamic data for each specific mineral endmember in the thermodynamic database. We calculated these values according to the equations:

$$K_s = k_0 + k_1(P - P_r) + k_2(T - T_r) \qquad (9)$$

$$\mu = m_0 + m_1(P - P_r) + m_2(T - T_r) \qquad (10)$$

After calculating K_s and μ, we calculated the seismic velocities according to the following equations:

$$V_P = \sqrt{\frac{K_S + \frac{4}{3}\mu}{\rho}} \quad (11)$$

$$V_p = \sqrt{\frac{\mu}{\rho}} \quad (12)$$

$$V_{P_{bulk\ rock}} = \Sigma\left(V_{P_{endmember}} * X_{endmember} * X_{mineral}\right) \quad (13)$$

$$V_{S_{bulk\ rock}} = \Sigma\left(V_{S_{endmember}} * X_{endmember} * X_{mineral}\right) \quad (14)$$

where *X* is the mode of each mineral endmember, e.g., forsterite, in each mineral, e.g., olivine, $V_{endmember}$, $V_{P\ endmember}$, and $V_{S\ endmember}$ are the calculated mineral endmember volumes and seismic velocities at the areotherm temperature.

2.3 Results

2.3.1 Areotherms

In the reference geochemical model (Fig. 2a, Table 2), the crustal thickness growth was controlled from 20 km at 4 Ga to 50 km today, while the depth of the lithosphere increased from 124 km 4 Ga to 242 km in the present day, as constrained by the intersection of the areotherm and adiabat. This leads to a corresponding decrease in *F* over time starting at of 46% across the planet at 4 Ga decreasing to 11.5% at 1 Ga. At 4 Ga, the adiabat crosses the solidus at ~300 km, decreasing through time, until the present day where they do not cross indicating melting is not occurring, on average today.

In the Melt-Layer areotherm model (Fig. 2b, Table 2), the thicknesses of the crust grew as a function of melt fraction from 20 km at 4 Ga to 48 km thick in the present day. The depth of the lithosphere increased from 115 km 4 Ga to 182 km today. The melt production decreased from 49% 4 Ga to 8.9% in the present day. This model assumed that 100% of the melt generated over time was erupted to the surface and that none of the melt remained emplaced in the crust. This assumption is likely less realistic than assuming that the melt was partially emplaced within the crust, leading to the modifications outlined in the Melt-Average model.

In the Melt-Average areotherm model (Fig. 2c, Table 2), the thicknesses of the crust grew from 20 km at 4 Ga to 47 km thick in the present day. The depth of the lithosphere increased from 115 km 4 Ga to 226 km today. The melt production decreased from 49% 4 Ga to 0.18% in the present day. This model was unique in that it assumed that not all melt generated erupted into the surface and instead took into consideration the emplacement of part of the melt produced in the crust.

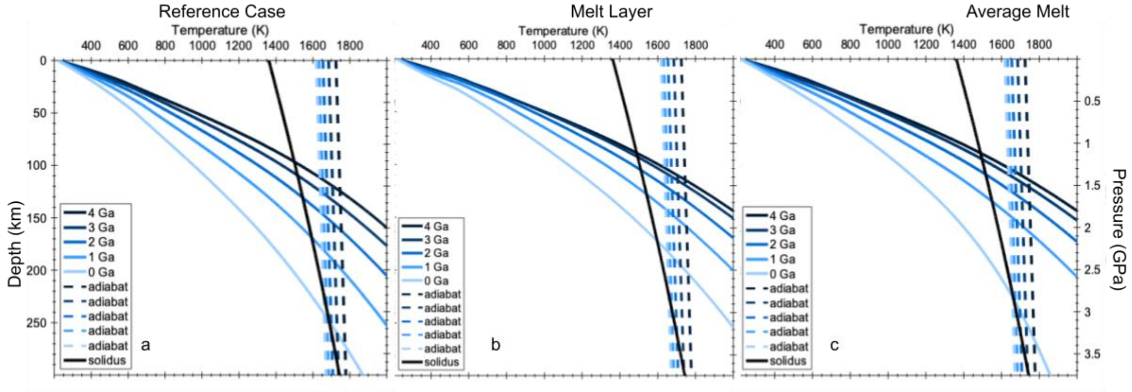

Figure 2-2. Calculated areotherms (solid lines) and adiabats (dashed lines) through time for each model: **a)** Reference, **b)** Melt-Layer, **c)** Melt-Average. Light colors represent thepresent day and the lines darken through time so that the darkest color represents the areotherm/adiabat at 4 Ga. The solidus is the solid black line**.**

Table 2-2. Results of MATLAB areotherm calculations through time.

Age (Ga)	4	3	2	1	0
Reference Model					
Crustal thickness (km)	20.00	27.50	35.00	42.50	50.00
q_0 (mW/m^2)	50.00	43.75	37.50	31.25	25.00
Mantle T_P (K)	1452	1412	1381	1359	1345
P_0 (GPa)	4.27	3.68	3.25	2.96	2.78
P_F (GPa)	1.54	1.64	1.84	2.22	2.96
Lithosphere depth (km)	124.6	133.7	150.8	181.9	241.7
dT/dP_{solidus}	105.2	104.2	102.2	98.5	91.3
dT/dF	4.31	4.24	4.12	3.87	3.40
F (%)	46.14	34.11	23.04	11.50	0.00
Melt-Layer Model					
Crustal thickness (km)	20.00	29.92	37.91	43.96	48.08
q_0 (mW/m^2)	50.00	43.75	37.50	31.25	25.00
Mantle T_P (K)	1452	1412	1381	1359	1345
P_0 (GPa)	4.27	3.68	3.25	2.96	2.78
P_F (GPa)	1.41	1.40	1.51	1.74	2.21
Lithosphere depth (km)	114.6	115.1	125.1	144.7	182.3
dT/dP_{solidus}	106.5	106.6	105.5	103.2	98.6
dT/dF	4.39	4.40	4.33	4.18	3.88
F (%)	49.04	39.20	29.58	20.06	8.95
Melt-Average Model					
Crustal thickness (km)	20.00	29.90	37.76	43.61	47.42
q_0 (mW/m^2)	50.00	43.75	37.50	31.25	25.00
Mantle T_P (K)	1452	1412	1381	1359	1345
P_0 (GPa)	4.27	3.68	3.25	2.96	2.78
P_F (GPa)	1.41	1.42	1.56	1.82	2.77
Lithosphere depth (km)	114.6	117.0	128.6	150.5	225.7
dT/dP_{solidus}	106.4	106.3	105.0	102.4	93.2
dT/dF	4.39	4.38	4.30	4.13	3.52
F (%)	48.93	38.60	28.59	18.58	0.18

2.3.2 Melting

Based on the calculated *F* values, we calculated the oxide composition in the melts over time (Fig. 3, Appendix A), for each of the areotherm models. In each model, the amount of melt produced decreases over time, however, the amount of decrease varies. The reference model produces 11.5% melt 1 Ga, but none in the present day; the solidus never crosses the temperature profile, resulting in 0% melt production. However, the Melt-Layer and Melt-Average models show about 9% and 0.2% melt, respectively, produced in the present day. Likewise, the melt compositions in each model show similar trends. Each becomes changes from ultramafic to mafic with time, with decreasing FeO and MgO contents, coupled with increasing Na_2O, K_2O, CaO, and Al_2O_3 contents. Each model shows similar trends in oxide abundance, although the absolute amount of each oxide calculated for each model varies (Tables A1, A2, A3). The SiO_2 concentrations are similar in each model, but the Average-Melt model is closest to the measured surface composition from (Taylor & McLennan, 2009a), while the Melt-Layer and Reference models are each lower than the measured surface composition. Both the Melt-Layer and Average Melt models produce present-day FeO, CaO, Al_2O_3, MgO, and Na_2O results similar to the measured surface composition, while SiO_2 is significantly greater. Additionally, the Reference and Melt-Layer Models tend to yield similar results, particularly through time, while the Average-Melt result is often slightly higher, such as with FeO and MgO, or slightly lower, such as with Al_2O_3, CaO, K_2O, and Na_2O. However, SiO_2 follows none of the aforementioned patterns and is overall distinguishable as the only oxide for which the Average-Melt values are in between the Reference and Melt-Layer models, and none of the models produce present-day SiO_2 outputs close to the measured surface composition.

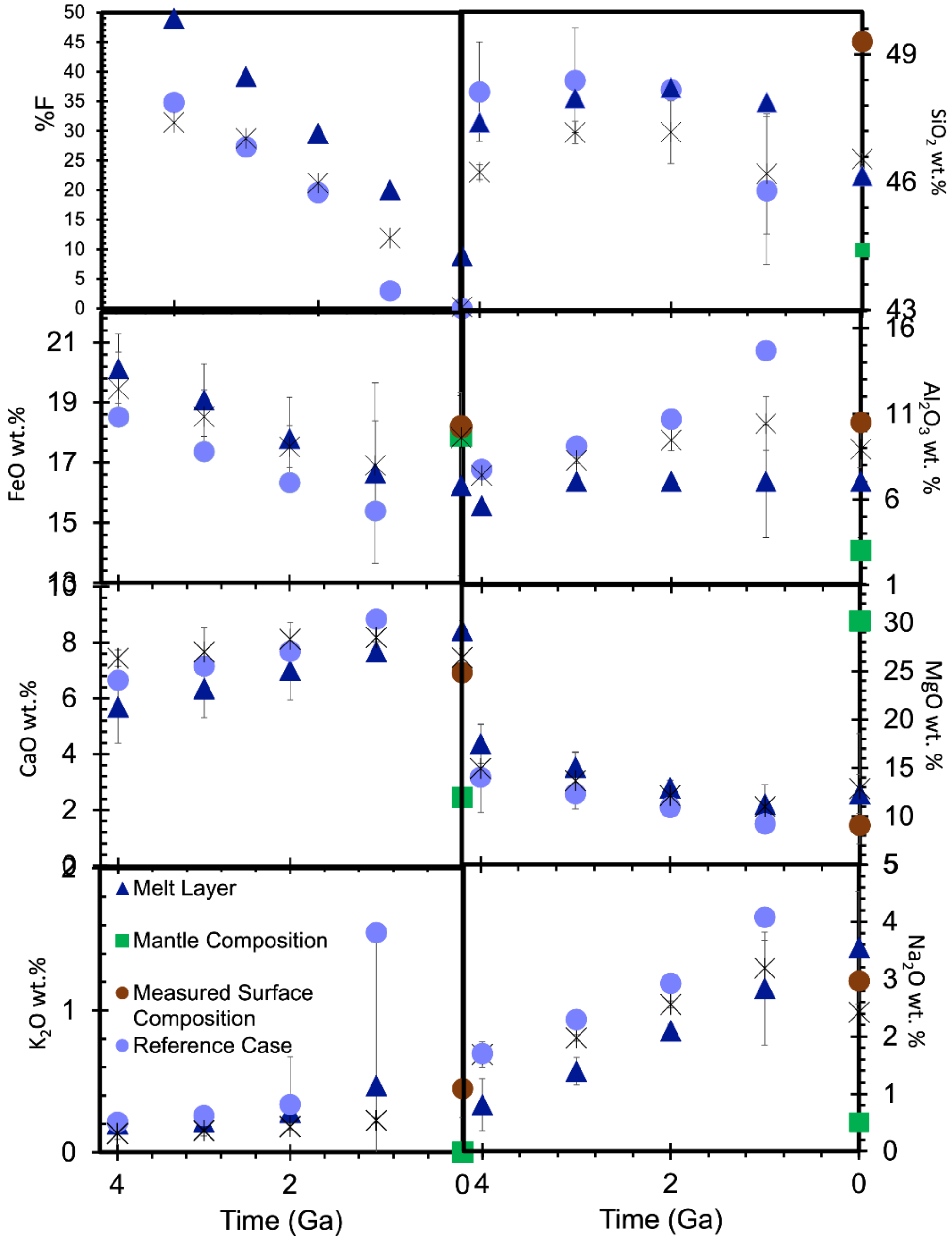

Figure 2-3. Calculated *F* and melt composition over time for each areotherm model: periwinkle circles are the Reference case, blue triangles are the Melt-Layer case, and cobalt stars are the Melt-Average case. The green square represents the undepleted mantle (Dreibus & Wänke, 1985), while the rust circle represents the composition on the planet's surface (Taylor & McLennan, 2009b).

2.3.3 Density

For all areotherm model melt composition results, the density calculated through the crust over time shows a consistent trend of decreasing density with depth. The mantle lithosphere shows the same trend, with the densities of the top of the mantle lithosphere at the mantle/crust boundary being greater than the density calculated at the base of the conductive lithosphere. While the densities of both the crust and the mantle lithosphere decrease over depth, the crustal density is always lower than the mantle lithosphere. Likewise, the density through the mantle lithosphere is lower than the density of the convective mantle beneath it.

After calculating the density for the Melt-Average model compositions (Fig. 4), the density calculated through the crust over time is similar to the previous two models, with some notable differences. At 4 Ga, the density at the top of the crust is ~2.29 g/cm^3, and decreases to 2.01 g/cm^3 at the base of the crust. This trend is observed in the crust at 3, 2, 1, and 0 Ga, with the crustal density decreasing from 2.62 to 2.57, 2.97 to 2.35, 3.13 to 3.11, and 3.09 to 2.01 g/cm^3, respectively. This trend is also observed in the mantle lithosphere; the density of the mantle lithosphere decreases from 3.50 to 3.40, 3.55 to 3.48, 3.59 to 3.53, 3.57 to 3.48, 3.66 to 3.61 g/cm^3 at 4, 3, 2, 1, and 0 Ga, respectively. The underlying mantle density is ~3.5 to 4.4 g/cm^3, increasing over depth (Khan et al., 2021).

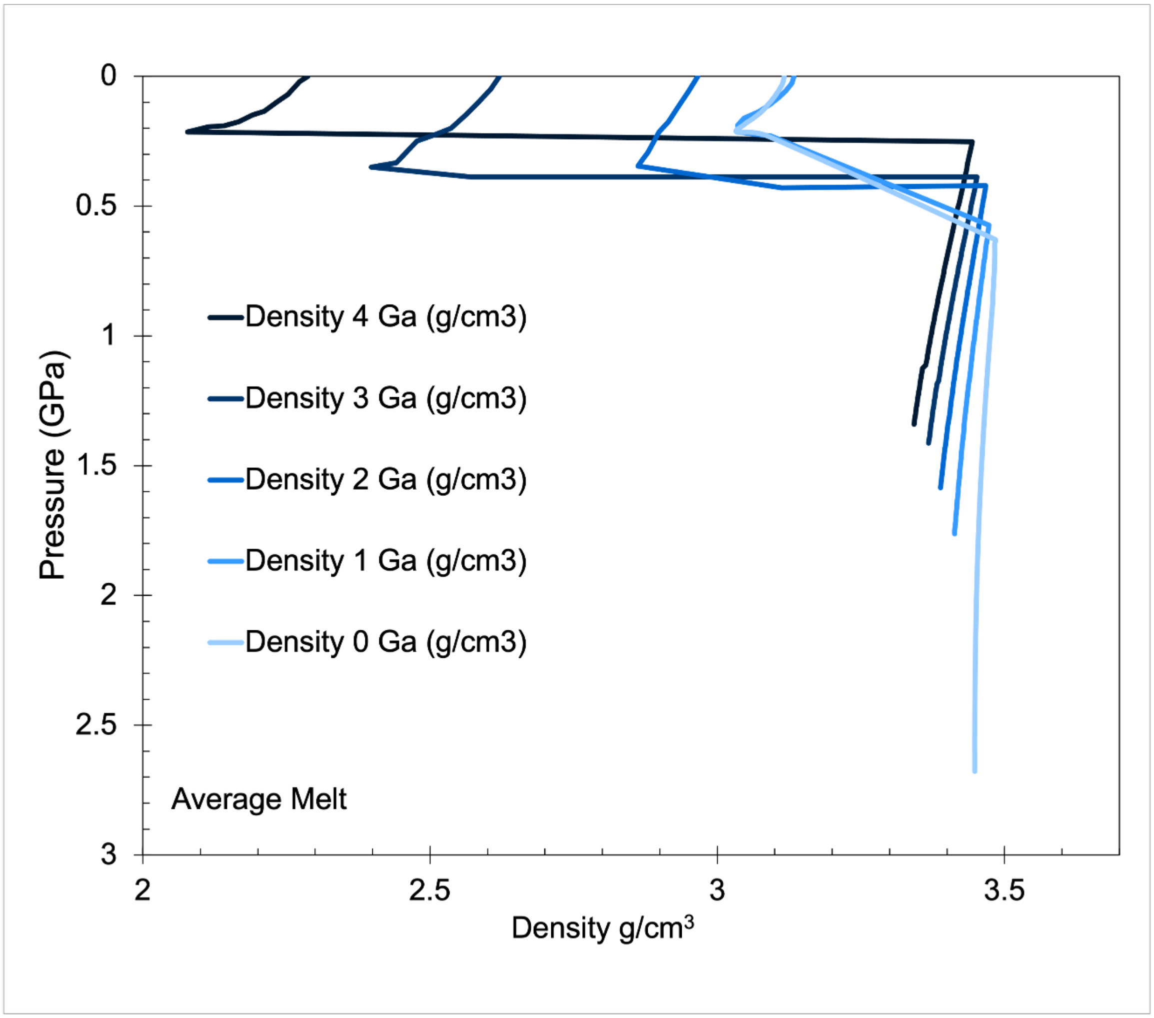

Figure 2-4. Density of the crust and mantle lithosphere (g/cm^3) calculated along the areotherms over time for the Melt-Average case, from 4 Ga (cobalt blue) to the present day (baby blue).

2.3.4 Seismic velocity

For all areotherm models, the velocities of primary and secondary waves generally remain constant through the depth of the crust, while they decrease over the depth of the mantle lithosphere. In the reference model, the seismic velocities decreased slightly over the depth of the crust, then increased across the crust/mantle boundary before decreasing across the depth of the mantle lithosphere, echoing the trend observed in the calculated density.

Using the Melt-Layer areotherms, we calculated a decrease in the seismic velocity of the crust. The values were similar to results of the other areotherm models, and varied primarily in regard to the location of the increase in seismic velocity over the crust/lithosphere boundary, due to the difference in calculated depth of the crust in each model. These increases in seismic velocity occurred at depths corresponding to pressures ranging from ~0.25 GPa at 4 Ga to ~0.5 GPa in the present day.

The results Melt-Average areotherms (Figs. 5,6) show at 4 Ga that V_P in the crust slightly decreased from 7.6 to 7.3 km/s from the top to bottom of the crust. Likewise, V_S decreased from 4.4 to 4.3 km/s with depth. In the mantle lithosphere, V_P and V_S decreased, from 7.7 to 7.6 km/s and 4.4 to 4.3 km/s, respectively, with depth. This pattern continued through time until it stabilized in the crust: V_P at 3, 2, 1, and 0 Ga decreased from 7.7 to 7.6, 7.8 to 7.7.Meanwhile, V_S decreased from 4.5 to 4.4 , 4.6 to 4.5. At 1 Ga, however, Vs remained at 4.5 at the top and bottom of the crust, and in the present d,ay, Vs is calculated at 6.9 km/s at the top and 7.0 km/s at the bottom of the crust.. In the mantle lithosphere, the seismic velocities also decreased over depth. At 3,2,1, and 0 Ga V_P decreased from 7.44 to 6.13, 7.43 to 6.14, 7.40 to 6.20, and 7.44 to 7.12 km/s, while V_S decreased from 4.26 to 2.79, 4.26 to 2.79, 4.24 to 2.81, and 4.26 to 4.01 km/s at 3, 2, and 1 Ga and in the present day. In both the crust and the mantle lithosphere, the results show layers of varying velocity; the difference from the top of bottom of each layer

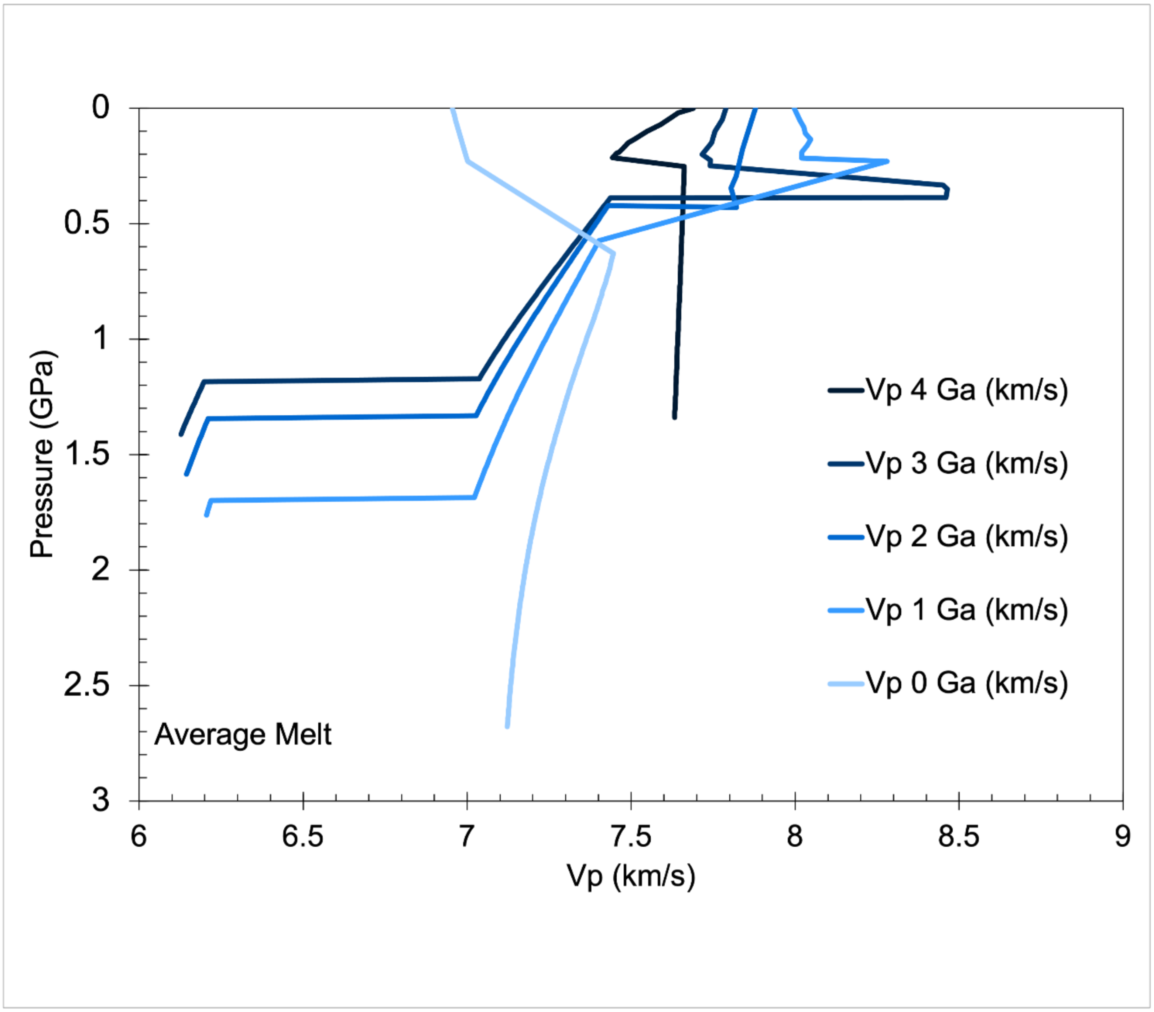

Figure 2-5. Comparison of calculated V_P in the crust over time for the Melt-Average model, from 4 Ga (cobalt blue) to the present (baby blue). The plots indicated that V_P decreases over the depth of the crust and mantle lithosphere each, with an increase along the crust and mantle lithosphere boundary.

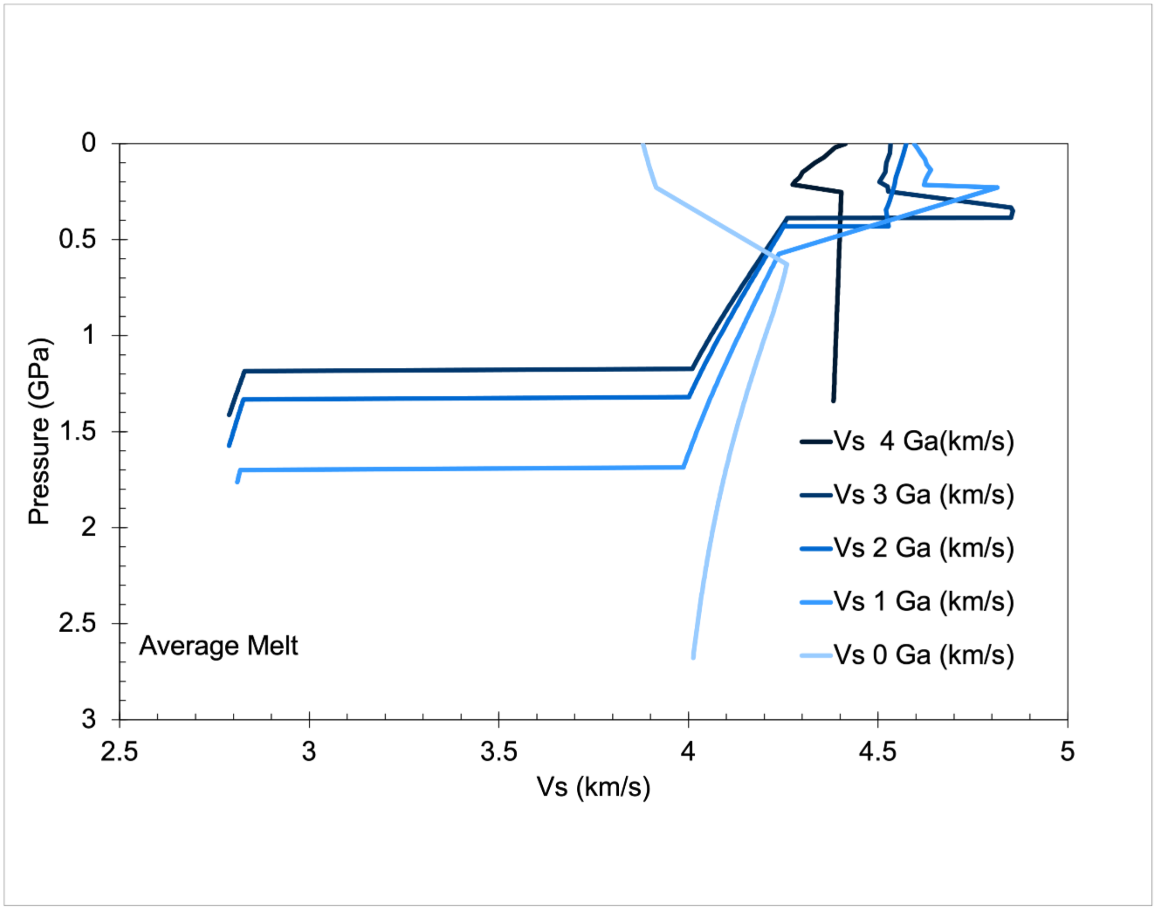

Figure 2-6. Comparison of calculated V_S in the crust over time for the Melt-Average model, from 4 Ga (cobalt blue) to the present (baby blue). The plots indicated that V_S decreases over the depth of the lithosphere.

2.4 Discussion

2.4.1 Areotherms and Melting

Based on the intersection of the solidus and mantle adiabat from Filiberto (2017), we found that there was significantly more melt generated in the past than more recently, with no melt on average being produced today for the Reference case. The Melt-Layer and Melt-Average areotherms indicate that there is about 9% and 0.2% melt generated in the present day, which corresponds to the results from (Horvath et al., 2021; Moitra et al., 2021) indicating that volcanic activity on Mars occurred recently and could feasibly occur again in the future.

One notable difference between the melt model and the modern observed crust is that the melt model prediction does not match what is observed at the surface in the present day (Fig. 2-3); the compositions of the melt through time are slightly different from the observed surface composition. This is most likely due to the partial crystallization process within magma chambers prior to eruption; as the melt sits within the chamber, it cools, partially crystallizes, and changes in composition. We took this into consideration while producing the Melt-Average areotherm model, so that the composition of the crust calculated through time is the composition of the average crust, rather than the composition observed at the surface. This partial crystallization process explains why the ultramafic melt predicted by the model yielded the mafic lava flows observed on the surface. Shortcomings of the model include the lack of experimental data; more data would increase the accuracy of the melt model by providing a larger number of data to use to fit the equations. More experimental work is necessary in order to continue to refine this model.

The layer thicknesses of each new layer decreased over time, from about 9 km of new crust following the melting to about 3 km of crust in the past billion years. The abundances of oxides (Fig. 3, Tables A1-A3) show the change in melt, and subsequently, crust layer, composition over time, from an ultramafic to mafic composition. This was reflected in all models. These compositional differences represent a change in crustal composition with each layer, and also indicate that the 20 km thick crust at 4 Ga input into the model is compositionally different from subsequent layers. This correlates to NASA's InSight mission, which observed a layered crust (Lognonné et al., 2020), which is also apparent in our models.

From these results, we infer that if the crust was thinner or had higher heat flow, there would be more melt production than is currently observed (akin to the conditions in the past). One interesting question to consider is the impact of heat flow versus crustal thickness: which parameter is the driving factor behind melt production? It is most reasonable to predict that the heat flow will have the greater impact; a high temperature crust (enriched in HPEs) is more likely to produce melt even if it is thick, than a thin but cold crust. As the heat flow increases, the temperature within the lithosphere is increased without change in pressure, which naturally generates melt. Likewise, a thinner crust would decrease the pressure while maintaining the same temperatures, which would also generate melt. However, between these two factors, the heat flow would have the greater impact, as the change in temperature at constant pressure would create a greater change within the rock than the pressure decrease along a constant temperature.

2.4.2 Density

The decreasing density over depth of the lithosphere was unexpected, due to the apparent long term stability of the martian crust (i.e., it's average age). Typical density profiles show an increase over the depth of the lithosphere, indicating a stable state, rather than a decrease, which suggests the presence of instability and overturn. However, given the small decrease with depth within a given layer, and the overall increasing density with depth between layers (i.e., from the crust to mantle lithosphere to mantle), it is not necessarily an unstable configuration. The lithosphere, both within the crust and the mantle lithosphere, is thick enough to lock itself into place so that it cannot be broken by the slight instability of the density decrease, which allows it to maintain what would otherwise be an unstable configuration and result in overturn of the lithosphere. Additionally, we did not consider the porosity resulting from the impact-related fracturing that has occurred on the surface and the density of the regolith on the outermost layer of the crust; the density of the top of the crust is in reality lower than we calculated due to these factors.

Due to the observed decreasing density for our models, we reproduced the Perple_X calculations of Semprich and Filiberto (2020), i.e., same input file and solution models, for the Taylor and McLennan (2009b) crustal and Dreibus and Wänke (1985) mantle compositions in conjunction with our present day areotherm. We extracted the densities calculated along our areotherm, which showed that there was a slight increase in density over depth of the crust when done according to the (Semprich & Filiberto, 2020) methods, but that the density across the conductive mantle lithosphere decreased over depth. The density calculated was also significantly lower than according to our model results in the present day; the crust modeled with our Perple_X assumptions had a density of about 3.05 g/cm^3, while the density calculated using the Semprich and Filiberto (2020) assumptions was about 2.45 g/cm^3.

2.4.3 Comparison to InSight's Observations

This research is pertinent to InSight by calculating the seismic profiles through the crust and mantle lithosphere. The heat flow calculations used in the areotherm estimates derived for Elysium Planitia (Plesa et al., 2016), where the InSight lander is located. Since landing, InSight constrained core size: 1830 km radius, and state: entirely liquid (Stähler S. C., 2021). This large radius of the core increases the amount of heat flowing through the mantle and the crust. This is pertinent to us because the amount of heat flowing through the mantle into the lithosphere has a direct impact on the temperature of the lithosphere and the amount of melt being generated, both in the present day and in the past

All of our models show that the seismic velocity decreases over lithosphere depth, following the same trends as calculated from the results of the InSight mission (Khan A., 2021), (Figs. 6,7). Comparing these results to InSight's observations demonstrates the accuracy and consistency of the model in the present day, which in turn shows the accuracy of the model through time. The InSight team also calculated Th concentrations necessary to generate melt producing the recent lava flows observed in the Tharsis region (Michaut, 2021). When compared to their Th concentration estimates of 1200–1700 ppb or 680–1050 ppb for the two and three layer crust models, (Michaut, 2021), the value calculated with our Average-Melt model of 241 ppb Th is significantly lower. Their model calculates the amount necessary to generate melt in Tharsis, while our model calculates the amount present based on melting assumptions of the initial crustal layer. It is possible that the disparity between our model and their results is due to the melt fraction calculation; if our melt fraction was lower than (Michaut, 2021), it would increase the amount of Th in the crust. The Th difference may be in indication that the actual melt fraction is lower than the melt fraction calculated by our model.

Overall, our model shows the same structural layers and seismic velocity profiles as observed by InSight, but with some differences (Th concentrations) between the Tharsis-specific models produced by InSight and the global average models that we used.

2.5 Conclusions

From these models, we have a clearer vision of how the martian lithosphere evolved over time. We see how the crust and mantle lithosphere have thickened over time, with a thickness calculated over Gy increments. We also calculate a decrease in density over the depth of the crust and mantle lithosphere, with the density of the mantle lithosphere greater than the density of the crust. The seismic velocities also remain close to constant in the crust and mantle lithosphere, with only a slight decrease over depth in each. Should additional martian data become available, specifically the surface heat flow, composition of the crust over depth, and observed lithospheric thickness. Experimentally, this model would be improved by incorporating data from more melting experiments of the partitioning of oxides into melt under lithospheric temperature and pressure conditions. Additionally, because this model relies on thermodynamically calculating the partitioning of oxides into minerals from the melt, it would be worthwhile to produce more of that experimental data. Such data could be used to calculate a model showing the partitioning of the oxides from solid into melt and then from melt into minerals. While the data that we used here from (Collinet et al., 2015; Matsukage et al., 2013b) As expected, the melt production decreased in quantity and changed from ultramafic to mafic over time; likely due to fewer Fe- and Mg-rich minerals such as olivine and pyroxene melting and crystalizing out of the upper mantle and into the crust. While the afore-mentioned observational and experimental data are necessary to refine and enhance this model, we can see from the available data and experiments how the lithosphere has changed compositionally over time. These models help us understand how Mars has evolved over time from an initially homogenous mantle composition to the conditions observed on the surface today. We can compare this model to known histories of Earth's evolution to see how each planet has thickened and cooled over time and how the melt composition has changed. Our work here presents a model of planetary lithospheric evolution that can provide a beginning towards comparing each planet's evolution and produce a self-consistent, planetary-wide model.

References

Abers, G. A., & Hacker, B. R. (2016), A MATLAB toolbox and E xcel workbook for calculating the densities, seismic wave speeds, and major element composition of minerals and rocks at pressure and temperature, *Geochemistry, Geophysics, Geosystems*, 17(2), 616-624.

Baratoux, D., Toplis, M. J., Monnereau, M., & Gasnault, O. (2011), Thermal history of Mars inferred from orbital geochemistry of volcanic provinces, *Nature*, 472(7343), 338-341, doi: 10.1038/nature09903.

Baratoux, D., Samuel, H., Michaut, C., Toplis, M. J., Monnereau, M., Wieczorek, M., et al. (2014), Petrological constraints on the density of the martian crust, *Journal of Geophysical Research-Planets*, 119(7), 1707-1727, doi: 10.1002/2014je004642.

Beattie, P. (1993), The generation of uranium series disequilibria by partial melting of spinel peridotite: constraints from partitioning studies, *Earth and Planetary Science Letters*, 117(3-4), 379-391, doi: 10.1016/0012-821X(93)90091-M.

Chapman, D. S. (1986), Thermal gradients in the continental crust, in *The Nature of the Lower Continental Crust*, edited by J. B. Dawson, Carswell, D. A., Hall, J. & Wedepohl, K. H., pp. 63-70, Geological Society Special Publication, London.

Collinet, M., Medard, E., Charlier, B., Vander Auwera, J., & Grove, T. L. (2015), Melting of the primitive martian mantle at 0.5-2.2 GPa and the origin of basalts and alkaline rocks on Mars, *Earth and Planetary Science Letters*, 427, 83-94, doi: 10.1016/j.epsl.2015.06.056.

Connolly, J. A. D. (2009), The geodynamic equation of state: What and how, *Geochemistry Geophysics Geosystems*, 10(10), doi: 10.1029/2009GC002540.

Davis, F. A., Hirschmann, M. M., & Humayun, M. (2011), The composition of the incipient partial melt of garnet peridotite at 3 GPa and the origin of OIB, *Earth and Planetary Science Letters*, 308(3-4), 380-390, doi: 10.1016/j.epsl.2011.06.008.

Dreibus, G., & Wänke, H. (1985), Mars, a volatile rich planet, *Meteoritics*, 20(2), 367-381.

Duncan, M. S., Dasgupta, R., & Tsuno, K. (2017), Experimental determination of CO_2 content at graphite saturation along a natural basalt-peridotite melt join: Implications for the fate of carbon in terrestrial magma oceans, *Earth and Planetary Science Letters*, 466, 115-128, doi: 10.1016/j.epsl.2017.03.008.

Duncan, M. S., Schmerr, N. C., Bertka, C. M., & Fei, Y. W. (2018), Extending the solidus for a model iron-rich martian mantle composition to 25 GPa, *Geophysical Research Letters*, 45(19), 10211-10220, doi: 10.1029/2018gl078182.

Filiberto, J. (2017), Geochemistry of martian basalts with constraints on magma genesis, *Chemical Geology*, 466, 1-14, doi: 10.1016/j.chemgeo.2017.06.009.

Goossens, S., Sabaka, T. J., Genova, A., Mazarico, E., Nicholas, J. B., & Neumann, G. A. (2017), Evidence for a low bulk crustal density for Mars from gravity and topography, *Geophysical Research Letters*, 44(15), 7686-7694, doi: 10.1002/2017gl074172.

Grott, M., Spohn, T., Knollenberg, J., Krause, C., Hudson, T. L., Piqueux, S., et al. (2021), Thermal Conductivity of the Martian Soil at the InSight Landing site from

HP ^{3} Active Heating Experiments, *Earth and Space Science Open Archive ESSOAr*.

Hartmann, W. K. (2005), Martian cratering 8: Isochron refinement and the chronology of Mars, *Icarus*, 174(2), 294-320.

Holland, T. J. B., & Powell, R. (2011), An improved and extended internally consistent thermodynamic dataset for phases of petrological interest, involving a new equation of state for solids, *Journal of Metamorphic Geology*, 29(3), 333-383, doi: 10.1111/j.1525-1314.2010.00923.x.

Horvath, D. G., Moitra, P., Hamilton, C. W., Craddock, R. A., & Andrews-Hanna, J. C. (2021), Evidence for geologically recent explosive volcanism in Elysium Planitia, Mars, *Icarus*, 365, 114499.

Khan, A., Liebske, C., Rozel, A., Rivoldini, A., Nimmo, F., Connolly, J. A. D., et al. (2018), A geophysical perspective on the bulk composition of Mars, *Journal of Geophysical Research-Planets*, 123(2), 575-611, doi: 10.1002/2017JE005371.

Khan A., C. S., van Driel M., Giardini D., Lognonne P. (2021), Constraints on the Martian Upper Mantle from InSight Seismic Data, paper presented at Lunar and Planetary Science Conference, Online.

Khan, D., Liebske, C., & Connolly, J. (2021), An Algorithm for Thermodynamic Parameter Optimization: Application to the Martian Mantle, *Geochemistry, Geophysics, Geosystems*, 22(5), e2020GC009399.

Kiefer, W. S. (2003), Melting in the martian mantle: Shergottite formation and implications for present-day mantle convection on Mars, *Meteoritics & Planetary Science*, 38(12), 1815-1832, doi: 10.1111/j.1945-5100.2003.tb00017.x.

Lillis, R. J., Dufek, J., Bleacher, J. E., & Manga, M. (2009), Demagnetization of crust by magmatic intrusion near the Arsia Mons volcano: Magnetic and thermal implications for the development of the Tharsis province, Mars, *Journal of Volcanology and Geothermal Research*, 185(1-2), 123-138, doi: 10.1016/j.jvolgeores.2008.12.007.

Lognonné, P., Banerdt, W., Pike, W., Giardini, D., Christensen, U., Garcia, R. F., et al. (2020), Constraints on the shallow elastic and anelastic structure of Mars from InSight seismic data, *Nature Geoscience*, 13(3), 213-220.

Matsukage, K. N., Nagayo, Y., Whitaker, M. L., Takahashi, E., & Kawasaki, T. (2013a), Melting of the martian mantle from 1.0 to 4.5 GPa, *Journal of Mineralogical and Petrological Sciences*, 108(4), 201-214, doi: 10.2465/jmps.120820.

Matsukage, K. N., Nagayo, Y., Whitaker, M. L., Takahashi, E., & Kawasaki, T. (2013b), Melting of the Martian mantle from 1.0 to 4.5 GPa, *Journal of Mineralogical and Petrological Sciences*, 120820.

McGovern, P. J., Solomon, S. C., Smith, D. E., Zuber, M. T., Simons, M., Wieczorek, M. A., et al. (2002), Localized gravity/topography admittance and correlation spectra on Mars: Implications for regional and global evolution, *Journal of Geophysical Research-Planets*, 107(E12, 5136), doi: 10.1029/2002je001854.

McGovern, P. J., Solomon, S. C., Smith, D. E., Zuber, M. T., Simons, M., Wieczorek, M. A., et al. (2004), Correction to "Localized gravity/topography admittance and correlation spectra on Mars: Implications for regional and global evolution", *Journal of Geophysical Research-Planets*, 109(E07007), doi: 10.1029/2004je002286.

Mckenzie, D., & Bickle, M. (1988), The volume and composition of melt generated by extension of the lithosphere, *Journal of petrology*, 29(3), 625-679.

McSween, H. Y., Taylor, G. J., & Wyatt, M. B. (2009), Elemental composition of the martian crust, *Science*, 324(5928), 736-739, doi: 10.1126/science.1165871.

Michaut, C., Plesa A.-C., Samuel H., Wieczorek M. A., McLennan S (2021), Crustal Radioactivity on Mars Constrained by InSight Data and Geodynamic Modeling, paper presented at Lunar and Planetary Science Conference.

Moitra, P., Horvath, D. G., & Andrews-Hanna, J. C. (2021), Investigating the roles of magmatic volatiles, ground ice and impact-triggering on a very recent and highly explosive volcanic eruption on Mars, *Earth and Planetary Science Letters*, 567, 116986.

Neumann, G. A., Zuber, M. T., Wieczorek, M. A., McGovern, P. J., Lemoine, F. G., & Smith, D. E. (2004), Crustal structure of Mars from gravity and topography, *Journal of Geophysical Research-Planets*, 109(E08002), doi: 10.1029/2004je002262.

Norman, M. D. (1999), The composition and thickness of the crust of Mars estimated from rare earth elements and neodymium-isotopic compositions of Martian meteorites, *Meteoritics & Planetary Science*, 34(3), 439-449.

Parro, L. M., Jimenez-Diaz, A., Mansilla, F., & Ruiz, J. (2017), Present day heat flow model of Mars, *Scientific Reports*, 7, 45629, doi: 10.1038/srep45629.

Plesa, A. C., Grott, M., Tosi, N., Breuer, D., Spohn, T., & Wieczorek, M. A. (2016), How large are present-day heat flux variations across the surface of Mars?, *Journal of Geophysical Research-Planets*, 121(12), 2386-2403, doi: 10.1002/2016je005126.

Powell, R., Holland, T., & Worley, B. (1998), Calculating phase diagrams involving solid solutions via non-linear equations, with examples using THERMOCALC, *Journal of metamorphic Geology*, 16(4), 577-588.

Rudnick, R. L., McDonough, W. F., & O'Connell, R. J. (1998), Thermal structure, thickness and composition of continental lithosphere, *Chemical Geology*, 145(3-4), 395-411, doi: 10.1016/S0009-2541(97)00151-4.

Schatz, J. F., & Simmons, G. (1972), Thermal conductivity of Earth materials at high temperatures, *Journal of Geophysical Research*, 77(35), 6966-6983, doi: 10.1029/JB077i035p06966.

Semprich, J., & Filiberto, J. (2020), High-pressure metamorphic mineralogy of the Martian crust with implications for density and seismic profiles, *Meteoritics & Planetary Science*, 55(7), 1600-1614.

Sohl, F., Schubert, G., & Spohn, T. (2005), Geophysical constraints on the composition and structure of the martian interior, *Journal of Geophysical Research-Planets,* 110(E12008), doi: 10.1029/2005je002520.

Stähler S. C., C. S., Duran A. C., Garcia R. G., Giardini D. (2021), Seismic Detection of the Martian Core by InSight, paper presented at Lunar and Planetary Science Conference.

Taylor, S. R., & McLennan, S. (2009a), *Planetary crusts: their composition, origin and evolution*, Cambridge University Press.

Taylor, S. R., & McLennan, S. (2009b), Mars: Crustal composition and evolution, in *Planetary Crusts: Their Composition, Origin and Evolution,* edited, pp. 141-180, Cambridge University Press.

Werner, S. C. (2009), The global martian volcanic evolutionary history, *Icarus*, 201(1), 44-68.

Wieczorek, M. A., & Zuber, M. T. (2004), Thickness of the martian crust: Improved constraints from geoid-to-topography ratios, *Journal of Geophysical Research-Planets*, 109(E01009), doi: 10.1029/2003je002153.

Appendix A

Geochemical Data

The equation forms were parameterized by fitting the data by pressure bins to determine correlation to pressure. We then fitted polynomial coefficients to the data by using least square minimization. The resulting equations were used to calculate the oxides based on the melt percent (F) and pressure (P_F).

$SiO_2 = (-4.26*10^{-4}*P + 6.01*10^{-4})*F^2 + (0.09.54*10^{-2}*P – 1.90*10^{-1})*F – (5.14*P + 57.37)$

$TiO_2 = 2.62*F^{-0.58}$

$Al_2O_3 = 1.51*10^{--3}*F^2 – 0.27*F + 15.51$

$Cr_2O_3 = (1.71*10^{-5}*P – 2.53*10^{-4})*F^2 + (-3.126046*10^{-3}*P + 3.71*10^{--2}) + (0.14*P – 0.39)$

$FeO = (9.30*10^{-4}*P – 3.78*10^{-3})*F^2 + (-0.14*P + 0.53) + (5.35*P + 2.71)$

$MnO = 5.73*10^{-5}*F^2 – 1.58*10^{-4}*F – 8.10*10^{-3}$

$MgO = (2.74*10^{-4}*P – 3.15*10^{-4})*F^2 + (6.28*10^{-2}*P + 0.33)*F + (3.61*P + 0.49)$

$CaO = 2.37*10^{-5}*F^2 – 6.97*10^{-2}*F + 9.05$

$Na_2O = (-2.71*10^{-4}*P + 1.28*10^{-3})*F^2 + (3.90*10^{-2}*P – 0.19)*F + (-1.22*P + 7.16)$

$K_2O = 3.74*F^{-0.81}$

$P_2O_5 = 8.90*F^{-0.84}$

Table A-1. Normalized melt compositions calculated with the Reference model.

Age (Ga)	4	3	2	1	0
Melt composition (wt.%)					
SiO_2	47.71	47.89	47.74	46.66	n.a.
TiO_2	0.28	0.33	0.42	0.64	n.a.
Al_2O_3	6.12	7.81	9.40	10.56	n.a.
Cr_2O_3	0.85	0.68	0.49	0.27	n.a.
FeO	20.10	18.84	17.66	17.14	n.a.
MnO	0.56	0.53	0.50	0.51	n.a.
MgO	17.13	14.41	12.37	11.12	n.a.
CaO	5.84	7.22	8.08	8.27	n.a.
Na_2O	0.93	1.66	2.47	3.25	n.a.
K_2O	0.14	0.19	0.26	0.49	n.a.
P_2O_5	0.34	0.44	0.61	1.09	n.a.

Table A-2. Normalized melt compositions calculated with the Melt-Layer model.

Age (Ga)	4	3	2	1	0
Melt composition (wt.%)					
SiO_2	47.80	48.20	48.38	48.15	46.64
TiO_2	0.27	0.31	0.36	0.46	0.74
Al_2O_3	5.80	7.22	8.75	10.15	11.04
Cr_2O_3	0.87	0.75	0.60	0.42	0.20
FeO	20.29	19.20	17.94	16.89	16.67
MnO	0.57	0.54	0.50	0.48	0.50
MgO	17.68	15.21	13.09	11.48	10.59
CaO	5.48	6.67	7.62	8.20	8.17
Na_2O	0.78	1.35	2.05	2.78	3.49
K_2O	0.13	0.16	0.21	0.30	0.61
P_2O_5	0.33	0.39	0.50	0.69	1.34

Table A-3. Normalized melt compositions calculated with the Melt-Average model.

Age (Ga)	4	3	2	1	0
Melt composition (wt.%)					
SiO_2	47.80	48.17	48.31	47.96	28.02
TiO_2	0.27	0.31	0.37	0.48	4.65
Al_2O_3	5.81	7.30	8.86	10.27	6.69
Cr_2O_3	0.87	0.74	0.58	0.39	0.02
FeO	20.28	19.15	17.88	16.87	11.67
MnO	0.57	0.54	0.50	0.48	0.39
MgO	17.66	15.10	12.96	11.37	7.11
CaO	5.49	6.74	7.70	8.25	4.76
Na_2O	0.78	1.39	2.11	2.88	2.36
K_2O	0.13	0.17	0.22	0.32	12.58
P_2O_5	0.33	0.40	0.51	0.73	21.76

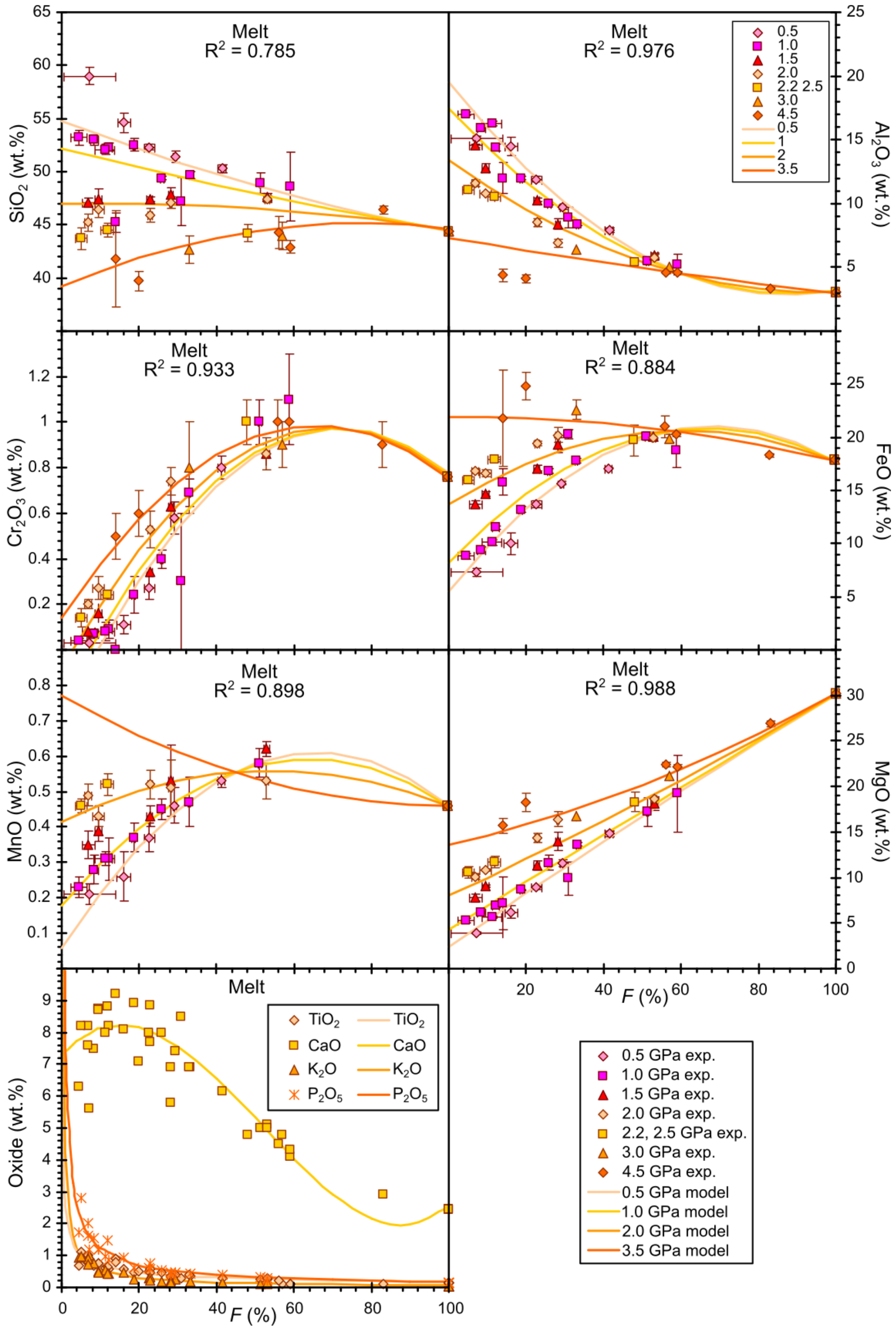

Figure A-1. Model fit to the data. This figure shows the model calculations as solid line curves and the geochemical data from Matsukage, 2013 and Collinet, 2015 as markers. Data is divided into pressure bins to show correlation between pressure and oxide abundance.

Appendix B

Matlab Codes

B.1 Reference Model

```
%---------------------------------------------------------------------
% Project: Areotherm_Reference.m
% Author: Fiona McGroarty
% Date Written: 19 October 2019
% Last Updated: 19 May 2021
% Purpose: This program will calculate the HPE concentration in the
depleted mantle lithosphere and then use those values to calculate the
parameters for an areotherm through time and plot the result. It will
calculate the composition of melt produced at a particular time and the
composition of the residual, with error. It will then repeat the process
using the previous calculation as the starting value for the next timestep
and the previous melt as the most recent layer in the crust.
% Addiitonal files needed: MeltModel.xlsx
%---------------------------------------------------------------------

clear
clc

%% Load the initial values from the Excel file meltmodel.xlsx
% Load values needed later to calculate F (dT/dPsolidus, dT/dPadiabat,
deltaHF, C_P. dT/dF). P0 and Pf will be calculated in a later step
prior to calculating F

% 1. Parameter read in
param = importdata('OxideCoeff.txt');

% 2. Variable read in
SiO2A = param(1,1);
SiO2B = param(1,2);
SiO2C = param(1,3);
SiO2D = param(1,4);
SiO2E = param(1,5);
SiO2F = param(1,6);

TiO2A = param(2,1);
TiO2B = param(2,2);
TiO2C = param(2,3);
TiO2D = param(2,4);
TiO2E = param(2,5);
TiO2F = param(2,6);

Al2O3A = param(3,1);
Al2O3B = param(3,2);
Al2O3C = param(3,3);
Al2O3D = param(3,4);
```

```
Al2O3E = param(3,5);
Al2O3F = param(3,6);

Cr2O3A = param(4,1);
Cr2O3B = param(4,2);
Cr2O3C = param(4,3);
Cr2O3D = param(4,4);
Cr2O3E = param(4,5);
Cr2O3F = param(4,6);

FeOA = param(5,1);
FeOB = param(5,2);
FeOC = param(5,3);
FeOD = param(5,4);
FeOE = param(5,5);
FeOF = param(5,6);

MnOA = param(6,1);
MnOB = param(6,2);
MnOC = param(6,3);
MnOD = param(6,4);
MnOE = param(6,5);
MnOF = param(6,6);

MgOA = param(7,1);
MgOB = param(7,2);
MgOC = param(7,3);
MgOD = param(7,4);
MgOE = param(7,5);
MgOF = param(7,6);

CaOA = param(8,1);
CaOB = param(8,2);
CaOC = param(8,3);
CaOD = param(8,4);
CaOE = param(8,5);
CaOF = param(8,6);

Na2OA = param(9,1);
Na2OB = param(9,2);
Na2OC = param(9,3);
Na2OD = param(9,4);
Na2OE = param(9,5);
Na2OF = param(9,6);

K2OA = param(10,1);
K2OB = param(10,2);
K2OC = param(10,3);
K2OD = param(10,4);
K2OE = param(10,5);
K2OF = param(10,6);

P2O5A = param(11,1);
P2O5B = param(11,2);
P2O5C = param(11,3);
```

```
P2O5D = param(11,4);
P2O5E = param(11,5);
P2O5F = param(11,6);

%% Mars parameters
rhoc = 2900; %kg/m^3 crustal density
rhoml = 3400; %kg/m^3 mantle lithosphere density
rhom = 3700; %kg/m^3 mantle density
T0 = 220; %Temperature at the surface in K
g = 3.72; %m/s^2 gravitational acceleration on Mars
slope = 0.18; %adiabat slope (deg/km)
slope2 = (slope/1000)/(g*rhoml)*1E9; %adiabat slope
deltaHF = 640000; %J/kg (Kiefer, 2003)
C_P = 1200; %J/K*kg (Kiefer, 2003)
rad = 3390; %km radius planet
d = 5000; %The number of rows of the array to hold the values
step = 0.1; %step for depth km

dTdFa = -0.639; %calculated from experimental data
dTdFb = 5.2926; %calculated from experimental data

%Input the oxides in the undepleted mantle (Dreibus and Wanke 1985)
mantleSiO2 = 44.4;
mantleTiO2 = 0.14;
mantleAl2O3 = 3.02;
mantleFeO = 17.9;
mantleCr2O3 = 0.76;
mantleMnO = 0.47;
mantleMgO = 30.2;
mantleCaO = 2.45;
mantleNa2O = 0.50;
mantleK2O = 0.04;
mantleP2O5 = 0.16;

%Initial thermal conductivity guess
kc = 2.5; %crust thermal condutivity(W/mK)
klth = 4.0; %mantle lithosphere thermal conductivity(W/mK)

% Heat Production Constants
U238 = 0.992834; %fraction of U238 in the solar system
U235 = 0.00711; %fraction of U235
U238t = 4.469E9; %half-life in yr
U235t = 0.704E9; %half-life in yr
H238U = 9.464E-5; %rate of heat release 238U (W/kg)
H235U = 5.687E-4; %rate of heat release 235U (W/kg)

Th232 = 0.9998; %fraction of Th232
Th232t = 14.00E9; %half-life in yr
H232Th = 2.638E-5; %rate of heat release 232Th (W/kg)

K40 = 1.193E-4; %fraction of K40
K40t = 1.268E9; %half-life in yr
H40K = 2.97E-5; %rate of heat release 40K (W/kg)

%Bulk peridotite/melt partition coefficients
```

```
DK = 0.003; %Davis et al. 2011 EPSL, garnet peridodite
DU = 1.1e-4; %Beatie 1993 EPSL, spinel peridodite
DTh = 1.7e-4; %Beatie 1993 EPSL, spinel peridodite

%HPE concentrations
Uc = 180/1E9; %Uranium in crust (Taylor and McLennon, 2009)
Thc = 700/1E9; %Thorium in crust (Taylor and McLennon, 2009)
Kc = 3740/1E6; %Potassium in crust (Taylor and McLennon, 2009)

Um = 16/1E9; %Uranium in mantle Dreibus and Wanke
Thm = 56/1E9; %Thorium in mantle Dreibus and Wanke
Km = 315/1E6; %Potassium in mantle Dreibus and Wanke



%% Areotherm 4 Ga
% Input the initial parameters TO BE CHANGED WITH TIME
t = 4*10^9; %time in years, 0 is present
Q04 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad4 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Forig = 0.1; %initial guess for melt fraction for first crust
Uc4 = Um/(Forig+(1-Forig)*DU); %Uranium in crust
Thc4 = Thm/(Forig+(1-Forig)*DTh); %Thorium in crust
Kc4 = Km/(Forig+(1-Forig)*DK); %Potassium in crust
Um4 = (Um-Forig*Uc4)/(1-Forig); %Solve for mantle lithosphere
concentration
Thm4 = (Thm-Forig*Thc4)/(1-Forig); %Solve for mantle lithosphere
concentration
Km4 = (Km-Forig*Kc4)/(1-Forig); %Solve for mantle lithosphere
concentration

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)
A4c = Atotc4;
```

```
A4m = Atotm4;
Am = Atotm;

%Calculate the layer thicknesses
lay1c = 20;
vol1c = (4/3)*pi*(rad^3-(rad-lay1c)^3); %km^3 volume of crustal layer
mass1c = vol1c*(1000^3)*rhoc; %kg mass crustal layer

mass1m = mass1c/Forig-mass1c; %kg mass lithosphere layer
vol1m = mass1m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay1m = (rad-lay1c)-((rad-lay1c)^3-vol1m/((4/3)*pi))^(1/3); %km
thicknes of lithosphere layer

cth4 = lay1c; %total thickness of crust in km
cmth4 = cth4+lay1m;
%% Calculate areotherm at 4 Ga
x = zeros(d,33);
x1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
x(1,1) = 0; %Initial Pressure (GPa)
x(1,2) = 0; %Initial Depth (km)
x(1,3) = Tad4+273;
x(1,4) = (-4.877*x(1,1)^2+120.2*x(1,1)+1088)+273.15; %solidus
x(1,5) = Q04/1000; %This is the surface heat flow (W/m^2)
x(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   x(i,2)=x(i-1,2)+step; %km
   if x(i,2)<cth4
       x(i,1)=(rhoc*g*x(i,2)*1000)/10^9;
   else
       x(i,1)=(rhoc*g*cth4*1000)/10^9+(rhoml*g*((x(i,2)-
cth4)*1000))/10^9;
   end

   %These are the adiabat and solidus
   x(i,3)=slope*x(i,2)+x(1,3);
   x(i,4)=(-4.877*x(i,1)^2+120.2*x(i,1)+1088)+273.15; %solidus

   %This is the q value
   if x(i,2)<cth4
       q=x(i-1,5)-A4c*(x(i,2)-x(i-1,2))*1000;
   elseif x(i,2)>cth4 && x(i,2)<cmth4
       q=x(i-1,5)-A4m*(x(i,2)-x(i-1,2))*1000;
   else
       q=x(i-1,5)-Am*(x(i,2)-x(i-1,2))*1000;
   end
   x(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if x(i,2)<cth4
```

```
        k=kc;
    else
        k=klth;
    end
    x(i,6)=k; %Put this k value into the array

    %Step 2: Calculate the temperature at this z value
    %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
    if x(i,2)<cth4
        T=x(i-1,7)+x(i,5)*((x(i,2)-x(i-1,2))*1000)/x(i,6)-A4c*((x(i,2)-
x(i-1,2))*1000)^2/(2*x(i,6));
    elseif x(i,2)>cth4 && x(i,2)<cmth4
        T=x(i-1,7)+x(i,5)*((x(i,2)-x(i-1,2))*1000)/x(i,6)-A4m*((x(i,2)-
x(i-1,2))*1000)^2/(2*x(i,6));
    else
        T=x(i-1,7)+x(i,5)*((x(i,2)-x(i-1,2))*1000)/x(i,6)-Am*((x(i,2)-
x(i-1,2))*1000)^2/(2*x(i,6));
    end
    x(i,7)=T;

    %Steps 5 and 6 must be iterated 10 times; use a for loop
    for j=8:2:33
        %Step 5: Calculate k using most recent temperature calculation
        if x(i,2)<cth4
            k=kc*(1+0.0015*x(i,2))/(1+0.0001*(x(i,j-1)-273));
        elseif x(i,2)>cth4
            if x(i,j-1)<500
                k=1/(0.0741+0.000502*x(i,j-1));
            else %Take the larger of these values
                k1=1/(0.0741+0.000502*x(i,j-1))+0.0023*(x(i,j-1)-500);
                %or
                k2=1.26*(1+x(i,2)/1000)+0.0023*(x(i,j-1)-500);
                if k1>k2
                    k=k1;
                else
                    k=k2;
                end
            end
        end
        x(i,j)=k; %Save the applicable k in the array

        x(1,j+1)=T0; %set T on first row

        %Step 6: Calculate the temperature using the most recent k
        if x(i,2)<cth4
            T=x(i-1,j+1)+x(i,5)*(x(i,2)-x(i-1,2))*1000/x(i,j)-
A4c*((x(i,2)-x(i-1,2))*1000)^2/(2*x(i,j));
        elseif x(i,2)>cth4 && x(i,2)<cmth4
            T=x(i-1,j+1)+x(i,5)*(x(i,2)-x(i-1,2))*1000/x(i,j)-
A4m*((x(i,2)-x(i-1,2))*1000)^2/(2*x(i,j));
        else
            T=x(i-1,j+1)+x(i,5)*(x(i,2)-x(i-1,2))*1000/x(i,j)-
Am*((x(i,2)-x(i-1,2))*1000)^2/(2*x(i,j));
        end
        x(i,j+1)=T; %Stores the values for T in the array
    end %Ends the Step 5 and 6 loop
```

```
    x1(i,1)=x(i,1)*10000;
    x1(1,2)=x(1,33);
    x1(i,2)=x(i,33);
    A4(i,1)=x(i,33)-x(i,3);
    B4(i,1)=x(i,1);
    C4(i,1)=x(i,2);
    PFp4 = find(A4>0,1,'first');
    PFm4 = find(A4<0,1,'last');
    PF4 = (B4(PFp4)+B4(PFm4))/2; %calulate pressure
    dF4 = (C4(PFp4)+C4(PFm4))/2; %calculate lithosphere+crust thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w=zeros(31,5);
w(1,1)=cth4;
w(2,1)=Q04;
w(3,1)=Tad4;

syms P0
eqn1 = slope2*P0+(Tad4+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P04 = min(vpasolve(eqn1,P0)); %pressure of adiabat and solidus
intersection
P04 = double(P04);
w(4,1)=P04;
w(5,1)=PF4;
w(6,1)=dF4;

dTdPsolidus4 = 2*(-4.877*PF4)+120.2; %solidus slope
dTdF4 = dTdFa*PF4+dTdFb;
w(7,1)=dTdPsolidus4;
w(8,1)=dTdF4;

% Use P0 and PF to calculate F (percentage of melt produced)
F4 = ((dTdPsolidus4-slope2)/((deltaHF/C_P)+dTdF4))*(P04-PF4)*100;
w(9,1)=F4;

% Calculate the composition of the melt and error
SiO24 = (SiO2A*PF4+SiO2B)*F4^2+(SiO2C*PF4+SiO2D)*F4+(SiO2E*PF4+SiO2F);
TiO24 = TiO2A*F4^TiO2B;
Al2O34 =
(Al2O3A*PF4+Al2O3B)*F4^2+(Al2O3C*PF4+Al2O3D)*F4+(Al2O3E*PF4+Al2O3F);
Cr2O34 =
(Cr2O3A*PF4+Cr2O3B)*F4^2+(Cr2O3C*PF4+Cr2O3D)*F4+(Cr2O3E*PF4+Cr2O3F);
FeO4 = (FeOA*PF4+FeOB)*F4^2+(FeOC*PF4+FeOD)*F4+(FeOE*PF4+FeOF);
MnO4 = (MnOA*PF4+MnOB)*F4^2+(MnOC*PF4+MnOD)*F4+(MnOE*PF4+MnOF);
MgO4 = (MgOA*PF4+MgOB)*F4^2+(MgOC*PF4+MgOD)*F4+(MgOE*PF4+MgOF);
CaO4 = CaOA*F4^3+CaOB*F4^2+CaOC*F4+CaOD;
Na2O4 = (Na2OA*PF4+Na2OB)*F4^2+(Na2OC*PF4+Na2OD)*F4+(Na2OE*PF4+Na2OF);
K2O4 = K2OA*F4^K2OB;
P2O54 = P2O5A*F4^P2O5B;
w(10,1)=SiO24;
w(11,1)=TiO24;
w(12,1)=Al2O34;
w(13,1)=Cr2O34;
w(14,1)=FeO4;
w(15,1)=MnO4;
w(16,1)=MgO4;
```

```
w(17,1)=CaO4;
w(18,1)=Na2O4;
w(19,1)=K2O4;
w(20,1)=P2O54;

% Calculate the residual using batch melting- will become mantle lithos
SiO2res4 = (mantleSiO2-((F4/100)*SiO24))/(1-(F4/100));
TiO2res4 = (mantleTiO2-((F4/100)*TiO24))/(1-(F4/100));
Al2O3res4 = (mantleAl2O3-((F4/100)*Al2O34))/(1-(F4/100));
Cr2O3res4 = (mantleCr2O3-((F4/100)*Cr2O34))/(1-(F4/100));
FeOres4 = (mantleFeO-((F4/100)*FeO4))/(1-(F4/100));
MnOres4 = (mantleMnO-((F4/100)*MnO4))/(1-(F4/100));
MgOres4 = (mantleMgO-((F4/100)*MgO4))/(1-(F4/100));
CaOres4 = (mantleCaO-((F4/100)*CaO4))/(1-(F4/100));
Na2Ores4 = (mantleNa2O-((F4/100)*Na2O4))/(1-(F4/100));
K2Ores4 = (mantleK2O-((F4/100)*K2O4))/(1-(F4/100));
P2O5res4 = (mantleP2O5-((F4/100)*P2O54))/(1-(F4/100));
w(21,1)=SiO2res4;
w(22,1)=TiO2res4;
w(23,1)=Al2O3res4;
w(24,1)=Cr2O3res4;
w(25,1)=FeOres4;
w(26,1)=MnOres4;
w(27,1)=MgOres4;
w(28,1)=CaOres4;
w(29,1)=Na2Ores4;
w(30,1)=K2Ores4;
w(31,1)=P2O5res4;
%End of 4 Ga (will become one distinct layer)


%% 3 Ga
% Input the initial parameters TO BE CHANGED WITH TIME
t = 3*10^9; %time in years, 0 is present
Q03 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad3 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc3 = Um/(F4+(1-F4)*DU); %Uranium in crust
Thc3 = Thm/(F4+(1-F4)*DTh); %Thorium in crust
Kc3 = Km/(F4+(1-F4)*DK); %Potassium in crust

Um3 = Um-F4*Uc3/(1-F4); %Uranium in crust
Thm3 = Thm-F4*Thc3/(1-F4); %Thorium in crust
Km3 = Km-F4*Kc3/(1-F4); %Potassium in crust

HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
```

```
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat
production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)

%Calculate the layer thicknesses
lay2c = 20*(F4/100); %km
vol2c = (4/3)*pi*(rad^3-(rad-lay2c)^3); %km^3 volume of crustal layer
mass2c = vol2c*(1000^3)*rhoc; %kg mass crustal layer

lay1c = (rad-lay2c)-((rad-lay2c)^3-vol1c/((4/3)*pi))^(1/3); %km

lay1m = (rad-lay2c-lay1c)-((rad-lay2c-lay1c)^3-vol1m/((4/3)*pi))^(1/3);
%km

mass2m = mass2c/(F4/100)-mass2c; %kg mass lithosphere layer
vol2m = mass2m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay2m = (rad-lay2c-lay1c-lay1m)-((rad-lay2c-lay1c-lay1m)^3-
vol2m/((4/3)*pi))^(1/3); %km

%Calculate base (depth) of each layer
cth3 = lay2c+lay1c; %total thickness of crust in km
cmth3 = cth3+lay1m+lay2m;

A3c = Atotc4*(lay1c/cth3)+Atotc3*(lay2c/cth3);
A3m = Atotm4*(lay1m/(cmth3-cth3))+Atotm3*(lay2m/(cmth3-cth3));
Am = Atotm;
%% Calculate areotherm 3 Ga
y = zeros(d,33);
y1 = zeros(d,2);
```

```
%Fill in the initial row of the array so that the loops can build the
rest
y(1,1) = 0; %Initial Pressure (GPa)
y(1,2) = 0; %Initial Depth (km)
y(1,3) = Tad3+273;
y(1,4) = (-4.877*y(1,1)^2+120.2*y(1,1)+1088)+273.15; %solidus
y(1,5) = Q03/1000; %This is the surface heat flow (W/m^2)
y(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   y(i,2)=y(i-1,2)+step; %km
   if y(i,2)<cth3
       y(i,1)=(rhoc*g*y(i,2)*1000)/10^9;
   else
       y(i,1)=(rhoc*g*cth3*1000)/10^9+(rhoml*g*((y(i,2)-
cth3)*1000))/10^9;
   end

   %These are the adiabat and solidus
   y(i,3)=slope*y(i,2)+y(1,3);
   y(i,4)=(-4.877*y(i,1)^2+120.2*y(i,1)+1088)+273.15; %solidus

   %This is the q value
   if y(i,2)<cth3
       q=y(i-1,5)-A3c*(y(i,2)-y(i-1,2))*1000;
   elseif y(i,2)>cth3 && y(i,2)<cmth3
       q=y(i-1,5)-A3m*(y(i,2)-y(i-1,2))*1000;
   else
       q=y(i-1,5)-Am*(y(i,2)-y(i-1,2))*1000;
   end
   y(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if y(i,2)<cth3
       k=kc;
   else
       k=klth;
   end
   y(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this z value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if y(i,2)<cth3
       T=y(i-1,7)+y(i,5)*((y(i,2)-y(i-1,2))*1000)/y(i,6)-A3c*((y(i,2)-
y(i-1,2))*1000)^2/(2*y(i,6));
   elseif y(i,2)>cth3 && y(i,2)<cmth3
       T=y(i-1,7)+y(i,5)*((y(i,2)-y(i-1,2))*1000)/y(i,6)-A3m*((y(i,2)-
y(i-1,2))*1000)^2/(2*y(i,6));
   else
       T=y(i-1,7)+y(i,5)*((y(i,2)-y(i-1,2))*1000)/y(i,6)-Am*((y(i,2)-
y(i-1,2))*1000)^2/(2*y(i,6));
   end
   y(i,7)=T;
```

```
   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
       %Step 5: Calculate k using most recent temperature calculation
       if y(i,2)<cth3
           k=kc*(1+0.0015*y(i,2))/(1+0.0001*(y(i,j-1)-273));
       elseif y(i,2)>cth3
           if y(i,j-1)<500
               k=1/(0.0741+0.000502*y(i,j-1));
           else %Take the larger of these values
               k1=1/(0.0741+0.000502*y(i,j-1))+0.0023*(y(i,j-1)-500);
               %or
               k2=1.26*(1+y(i,2)/1000)+0.0023*(y(i,j-1)-500);
               if k1>k2
                   k=k1;
               else
                   k=k2;
               end
           end
       end
       y(i,j)=k; %Save the applicable k in the array

       y(1,j+1)=T0; %set T on first row

       %Step 6: Calculate the temperature using the most recent k
       if y(i,2)<cth3
           T=y(i-1,j+1)+y(i,5)*(y(i,2)-y(i-1,2))*1000/y(i,j)-
A3c*((y(i,2)-y(i-1,2))*1000)^2/(2*y(i,j));
       elseif y(i,2)>cth3 && y(i,2)<cmth3
           T=y(i-1,j+1)+y(i,5)*(y(i,2)-y(i-1,2))*1000/y(i,j)-
A3m*((y(i,2)-y(i-1,2))*1000)^2/(2*y(i,j));
       else
           T=y(i-1,j+1)+y(i,5)*(y(i,2)-y(i-1,2))*1000/y(i,j)-
Am*((y(i,2)-y(i-1,2))*1000)^2/(2*y(i,j));
       end
       y(i,j+1)=T; %Stores the values for T in the array
   end %Ends the Step 5 and 6 loop
   y1(i,1)=y(i,1)*10000;
   y1(1,2)=y(1,33);
   y1(i,2)=y(i,33);
   A3(i,1)=y(i,33)-y(i,3);
   B3(i,1)=y(i,1);
   C3(i,1)=y(i,2);
   PFp3 = find(A3>0,1,'first');
   PFm3 = find(A3<0,1,'last');
   PF3 = (B3(PFp3)+B3(PFm3))/2; %calulate pressure
   dF3 = (C3(PFp3)+C3(PFm3))/2; %calculate crust+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,2)=cth3;
w(2,2)=Q03;
w(3,2)=Tad3;

syms P0
eqn1 = slope2*P0+(Tad3+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P03 = min(vpasolve(eqn1,P0)); %pressure of solidus-adiabat intersection
P03 = double(P03);
```

```
w(4,2)=P03;
w(5,2)=PF3;
w(6,2)=dF3;

dTdPsolidus3 = 2*(-4.877*PF3)+120.2; %slope of solidus Duncan et al.
(2018)
dTdF3 = dTdFa*PF3+dTdFb;
w(7,2)=dTdPsolidus3;
w(8,2)=dTdF3;

% Use P0 and PF to calculate F (percentage of melt produced)
F3 = ((dTdPsolidus3-slope2)/((deltaHF/C_P)+dTdF3))*(P03-PF3)*100;
w(9,2)=F3;

% Calculate the composition of the melt and error
SiO23 = (SiO2A*PF3+SiO2B)*F3^2+(SiO2C*PF3+SiO2D)*F3+(SiO2E*PF3+SiO2F);
TiO23 = TiO2A*F3^TiO2B;
Al2O33 =
(Al2O3A*PF3+Al2O3B)*F3^2+(Al2O3C*PF3+Al2O3D)*F3+(Al2O3E*PF3+Al2O3F);
Cr2O33 =
(Cr2O3A*PF3+Cr2O3B)*F3^2+(Cr2O3C*PF3+Cr2O3D)*F3+(Cr2O3E*PF3+Cr2O3F);
FeO3 = (FeOA*PF3+FeOB)*F3^2+(FeOC*PF3+FeOD)*F3+(FeOE*PF3+FeOF);
MnO3 = (MnOA*PF3+MnOB)*F3^2+(MnOC*PF3+MnOD)*F3+(MnOE*PF3+MnOF);
MgO3 = (MgOA*PF3+MgOB)*F3^2+(MgOC*PF3+MgOD)*F3+(MgOE*PF3+MgOF);
CaO3 = CaOA*F3^3+CaOB*F3^2+CaOC*F3+CaOD;
Na2O3 = (Na2OA*PF3+Na2OB)*F3^2+(Na2OC*PF3+Na2OD)*F3+(Na2OE*PF3+Na2OF);
K2O3 = K2OA*F3^K2OB;
P2O53 = P2O5A*F3^P2O5B;
w(10,2)=SiO23;
w(11,2)=TiO23;
w(12,2)=Al2O33;
w(13,2)=Cr2O33;
w(14,2)=FeO3;
w(15,2)=MnO3;
w(16,2)=MgO3;
w(17,2)=CaO3;
w(18,2)=Na2O3;
w(19,2)=K2O3;
w(20,2)=P2O53;

% Calculate the residual using batch melting- will become mantle lithos
SiO2res3 = (mantleSiO2-((F3/100)*SiO23))/(1-(F3/100));
TiO2res3 = (mantleTiO2-((F3/100)*TiO23))/(1-(F3/100));
Al2O3res3 = (mantleAl2O3-((F3/100)*Al2O33))/(1-(F3/100));
Cr2O3res3 = (mantleCr2O3-((F3/100)*Cr2O33))/(1-(F3/100));
FeOres3 = (mantleFeO-((F3/100)*FeO3))/(1-(F3/100));
MnOres3 = (mantleMnO-((F3/100)*MnO3))/(1-(F3/100));
MgOres3 = (mantleMgO-((F3/100)*MgO3))/(1-(F3/100));
CaOres3 = (mantleCaO-((F3/100)*CaO3))/(1-(F3/100));
Na2Ores3 = (mantleNa2O-((F3/100)*Na2O3))/(1-(F3/100));
K2Ores3 = (mantleK2O-((F3/100)*K2O3))/(1-(F3/100));
P2O5res3 = (mantleP2O5-((F3/100)*P2O53))/(1-(F3/100));
w(21,2)=SiO2res3;
w(22,2)=TiO2res3;
w(23,2)=Al2O3res3;
w(24,2)=Cr2O3res3;
```

```
w(25,2)=FeOres3;
w(26,2)=MnOres3;
w(27,2)=MgOres3;
w(28,2)=CaOres3;
w(29,2)=Na2Ores3;
w(30,2)=K2Ores3;
w(31,2)=P2O5res3;
%End of 3 Ga (becomes next distinct layer)


%% 2 Ga
t = 2*10^9; %time in years, 0 is present
Q02 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad2 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc2 = Um/(F3+(1-F3)*DU); %Uranium in crust
Thc2 = Thm/(F3+(1-F3)*DTh); %Thorium in crust
Kc2 = Km/(F3+(1-F3)*DK); %Potassium in crust

Um2 = Um-F3*Uc2/(1-F3); %Uranium in mantle lithosphere
Thm2 = Thm-F3*Thc2/(1-F3); %Thorium in mantle lithosphere
Km2 = Km-F3*Kc2/(1-F3); %Potassium in mantle lithosphere

HU238c2 = Uc2*U238*H238U*exp(t*log(2)/U238t);
HU235c2 = Uc2*U235*H235U*exp(t*log(2)/U235t);
HTh232c2 = Thc2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c2 = Kc2*K40*H40K*exp(t*log(2)/K40t);

HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m2 = Um2*U238*H238U*exp(t*log(2)/U238t);
HU235m2 = Um2*U235*H235U*exp(t*log(2)/U235t);
HTh232m2 = Thm2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m2 = Km2*K40*H40K*exp(t*log(2)/K40t);
```

```
HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc2 = (HU238c2+HU235c2+HTh232c2+HK40c2)*rhoc; %volumetric heat
production crust (W/m3)
Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat
production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm2 = (HU238m2+HU235m2+HTh232m2+HK40m2)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)

%Calculate the layer thicknesses
lay3c = 20*(F3/100); %km
vol3c = (4/3)*pi*(rad^3-(rad-lay3c)^3); %km^3 volume of crustal layer
mass3c = vol3c*(1000^3)*rhoc; %kg mass crustal layer

lay2c = (rad-lay3c)-((rad-lay3c)^3-vol2c/((4/3)*pi))^(1/3); %km

lay1c = (rad-lay3c-lay2c)-((rad-lay3c-lay2c)^3-vol1c/((4/3)*pi))^(1/3);
%km

lay1m = (rad-lay3c-lay2c-lay1c)-((rad-lay3c-lay2c-lay1c)^3-
vol1m/((4/3)*pi))^(1/3); %km

lay2m = (rad-lay3c-lay2c-lay1c-lay1m)-((rad-lay3c-lay2c-lay1c-lay1m)^3-
vol2m/((4/3)*pi))^(1/3); %km

mass3m = mass3c/(F3/100)-mass3c; %kg mass lithosphere layer
vol3m = mass3m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay3m = (rad-lay3c-lay2c-lay1c-lay1m-lay2m)-((rad-lay3c-lay2c-lay1c-
lay1m-lay2m)^3-vol3m/((4/3)*pi))^(1/3); %km

%Calculate base (depth) of each layer
cth2 = lay3c+lay2c+lay1c; %total thickness of crust in km
cmth2 = cth2+lay1m+lay2m+lay3m;

A2c = Atotc4*lay1c/cth2+Atotc3*lay2c/cth2+Atotc2*lay3c/cth2;
A2m = Atotm4*lay1m/(cmth2-cth2)+Atotm3*lay2m/(cmth2-
cth2)+Atotm2*lay3m/(cmth2-cth2);
Am = Atotm;
%% Calculate areotherm 2Ga
z = zeros(d,33);
z1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
```

```
z(1,1) = 0; %Initial Pressure (GPa)
z(1,2) = 0; %Initial Depth (km)
z(1,3) = Tad2+273;
z(1,4) = (-4.877*z(1,1)^2+120.2*z(1,1)+1088)+273.15; %solidus
z(1,5) = Q02/1000; %This is the surface heat flow (W/m^2)
z(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   z(i,2)=z(i-1,2)+step; %km
   if z(i,2)<cth2
       z(i,1)=(rhoc*g*z(i,2)*1000)/10^9;
   else
       z(i,1)=(rhoc*g*cth2*1000)/10^9+(rhoml*g*((z(i,2)-
cth2)*1000))/10^9;
   end

   %These are the adiabat and solidus
   z(i,3)=slope*z(i,2)+z(1,3);
   z(i,4)=(-4.877*z(i,1)^2+120.2*z(i,1)+1088)+273.15; %solidus

   %This is the q value
   if z(i,2)<cth2
       q=z(i-1,5)-A2c*(z(i,2)-z(i-1,2))*1000;
   elseif z(i,2)>cth2 && z(i,2)<cmth2
       q=z(i-1,5)-A2m*(z(i,2)-z(i-1,2))*1000;
   else
       q=z(i-1,5)-Am*(z(i,2)-z(i-1,2))*1000;
   end
   z(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if z(i,2)<cth2
       k=kc;
   else
       k=klth;
   end
   z(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this z value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if z(i,2)<cth2
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-A2c*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   elseif z(i,2)>cth2 && z(i,2)<cmth2
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-A2m*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   else
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-Am*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   end
   z(i,7)=T;

   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
```

```
       %Step 5: Calculate k using most recent temperature calculation
       if z(i,2)<cth2
           k=kc*(1+0.0015*z(i,2))/(1+0.0001*(z(i,j-1)-273));
       elseif z(i,2)>cth2
           if z(i,j-1)<500
               k=1/(0.0741+0.000502*z(i,j-1));
           else %Take the larger of these values
               k1=1/(0.0741+0.000502*z(i,j-1))+0.0023*(z(i,j-1)-500);
               %or
               k2=1.26*(1+z(i,2)/1000)+0.0023*(z(i,j-1)-500);
               if k1>k2
                   k=k1;
               else
                   k=k2;
               end
           end
       end
       z(i,j)=k; %Save the applicable k in the array

       z(1,j+1)=T0; %set T on first row

       %Step 6: Calculate the temperature using the most recent k
       if z(i,2)<cth2
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
A2c*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       elseif z(i,2)>cth2 && z(i,2)<cmth2
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
A2m*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       else
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
Am*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       end
       z(i,j+1)=T; %Stores the values for T in the array
   end %Ends the Step 5 and 6 loop
   z1(i,1)=z(i,1)*10000;
   z1(1,2)=z(1,33);
   z1(i,2)=z(i,33);
   A2(i,1)=z(i,33)-z(i,3);
   B2(i,1)=z(i,1);
   C2(i,1)=z(i,2);
   PFp2 = find(A2>0,1,'first');
   PFm2 = find(A2<0,1,'last');
   PF2 = (B2(PFp2)+B2(PFm2))/2; %calulate pressure
   dF2 = (C2(PFp2)+C2(PFm2))/2; %calculate crusth+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,3)=cth2;
w(2,3)=Q02;
w(3,3)=Tad2;

syms P0
eqn1 = slope2*P0+(Tad2+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P02 = min(vpasolve(eqn1,P0));
P02 = double(P02); %pressure where solidus and adiabat intersect
w(4,3)=P02;
w(5,3)=PF2;
```

```
w(6,3)=dF2;

dTdPsolidus2 = 2*(-4.877*PF2)+120.2; %slope of solidus Duncan et al.
(2018)
dTdF2 = dTdFa*PF2+dTdFb;
w(7,3)=dTdPsolidus2;
w(8,3)=dTdF2;

% Use P0 and PF to calculate F (percentage of melt produced)
F2 = ((dTdPsolidus2-slope2)/((deltaHF/C_P)+dTdF2))*(P02-PF2)*100;
w(9,3)=F2;

% Calculate the composition of the melt and error
SiO22 = (SiO2A*PF2+SiO2B)*F2^2+(SiO2C*PF2+SiO2D)*F2+(SiO2E*PF2+SiO2F);
TiO22 = TiO2A*F2^TiO2B;
Al2O32 =
(Al2O3A*PF2+Al2O3B)*F2^2+(Al2O3C*PF2+Al2O3D)*F2+(Al2O3E*PF2+Al2O3F);
Cr2O32 =
(Cr2O3A*PF2+Cr2O3B)*F2^2+(Cr2O3C*PF2+Cr2O3D)*F2+(Cr2O3E*PF2+Cr2O3F);
FeO2 = (FeOA*PF2+FeOB)*F2^2+(FeOC*PF2+FeOD)*F2+(FeOE*PF2+FeOF);
MnO2 = (MnOA*PF2+MnOB)*F2^2+(MnOC*PF2+MnOD)*F2+(MnOE*PF2+MnOF);
MgO2 = (MgOA*PF2+MgOB)*F2^2+(MgOC*PF2+MgOD)*F2+(MgOE*PF2+MgOF);
CaO2 = CaOA*F2^3+CaOB*F2^2+CaOC*F2+CaOD;
Na2O2 = (Na2OA*PF2+Na2OB)*F2^2+(Na2OC*PF2+Na2OD)*F2+(Na2OE*PF2+Na2OF);
K2O2 = K2OA*F2^K2OB;
P2O52 = P2O5A*F2^P2O5B;
w(10,3)=SiO22;
w(11,3)=TiO22;
w(12,3)=Al2O32;
w(13,3)=Cr2O32;
w(14,3)=FeO2;
w(15,3)=MnO2;
w(16,3)=MgO2;
w(17,3)=CaO2;
w(18,3)=Na2O2;
w(19,3)=K2O2;
w(20,3)=P2O52;

% Calculate the residual using batch melting-will become mantle lithos
SiO2res2 = (mantleSiO2-((F2/100)*SiO22))/(1-(F2/100));
TiO2res2 = (mantleTiO2-((F2/100)*TiO22))/(1-(F2/100));
Al2O3res2 = (mantleAl2O3-((F2/100)*Al2O32))/(1-(F2/100));
Cr2O3res2 = (mantleCr2O3-((F2/100)*Cr2O32))/(1-(F2/100));
FeOres2 = (mantleFeO-((F2/100)*FeO2))/(1-(F2/100));
MnOres2 = (mantleMnO-((F2/100)*MnO2))/(1-(F2/100));
MgOres2 = (mantleMgO-((F2/100)*MgO2))/(1-(F2/100));
CaOres2 = (mantleCaO-((F2/100)*CaO2))/(1-(F2/100));
Na2Ores2 = (mantleNa2O-((F2/100)*Na2O2))/(1-(F2/100));
K2Ores2 = (mantleK2O-((F2/100)*K2O2))/(1-(F2/100));
P2O5res2 = (mantleP2O5-((F2/100)*P2O52))/(1-(F2/100));
w(21,3)=SiO2res2;
w(22,3)=TiO2res2;
w(23,3)=Al2O3res2;
w(24,3)=Cr2O3res2;
w(25,3)=FeOres2;
w(26,3)=MnOres2;
```

```
w(27,3)=MgOres2;
w(28,3)=CaOres2;
w(29,3)=Na2Ores2;
w(30,3)=K2Ores2;
w(31,3)=P2O5res2;
%End of 2 Ga calculations


%% 1 Ga
t = 1*10^9; %time in years, 0 is present
Q01 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad1 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc1 = Um/(F2+(1-F2)*DU); %Uranium in crust
Thc1 = Thm/(F2+(1-F2)*DTh); %Thorium in crust
Kc1 = Km/(F2+(1-F2)*DK); %Potassium in crust

Um1 = Um-F2*Uc1/(1-F2); %Uranium in mantle lithosphere
Thm1 = Thm-F2*Thc1/(1-F2); %Thorium in mantle lithosphere
Km1 = Km-F2*Kc1/(1-F2); %Potassium in mantle lithosphere

HU238c1 = Uc1*U238*H238U*exp(t*log(2)/U238t);
HU235c1 = Uc1*U235*H235U*exp(t*log(2)/U235t);
HTh232c1 = Thc1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c1 = Kc1*K40*H40K*exp(t*log(2)/K40t);

HU238c2 = Uc2*U238*H238U*exp(t*log(2)/U238t);
HU235c2 = Uc2*U235*H235U*exp(t*log(2)/U235t);
HTh232c2 = Thc2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c2 = Kc2*K40*H40K*exp(t*log(2)/K40t);

HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m2 = Um2*U238*H238U*exp(t*log(2)/U238t);
HU235m2 = Um2*U235*H235U*exp(t*log(2)/U235t);
```

```
HTh232m2 = Thm2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m2 = Km2*K40*H40K*exp(t*log(2)/K40t);

HU238m1 = Um1*U238*H238U*exp(t*log(2)/U238t);
HU235m1 = Um1*U235*H235U*exp(t*log(2)/U235t);
HTh232m1 = Thm1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m1 = Km1*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc1 = (HU238c1+HU235c1+HTh232c1+HK40c1)*rhoc; %volumetric heat
production crust (W/m3)
Atotc2 = (HU238c2+HU235c2+HTh232c2+HK40c2)*rhoc; %volumetric heat
production crust (W/m3)
Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat
production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm2 = (HU238m2+HU235m2+HTh232m2+HK40m2)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm1 = (HU238m1+HU235m1+HTh232m1+HK40m1)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)

%Calculate the layer thicknesses
lay4c = 20*(F2/100); %km
vol4c = (4/3)*pi*(rad^3-(rad-lay4c)^3); %km^3 volume of crustal layer
mass4c = vol4c*(1000^3)*rhoc; %kg mass crustal layer

lay3c = (rad-lay4c)-((rad-lay4c)^3-vol3c/((4/3)*pi))^(1/3); %km

lay2c = (rad-lay4c-lay3c)-((rad-lay4c-lay3c)^3-vol2c/((4/3)*pi))^(1/3);
%km

lay1c = (rad-lay4c-lay3c-lay2c)-((rad-lay4c-lay3c-lay2c)^3-
vol1c/((4/3)*pi))^(1/3); %km

lay1m = (rad-lay4c-lay3c-lay2c-lay1c)-((rad-lay4c-lay3c-lay2c-lay1c)^3-
vol1m/((4/3)*pi))^(1/3); %km

lay2m = (rad-lay4c-lay3c-lay2c-lay1c-lay1m)-((rad-lay4c-lay3c-lay2c-
lay1c-lay1m)^3-vol2m/((4/3)*pi))^(1/3); %km

lay3m = (rad-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m)-((rad-lay4c-lay3c-
lay2c-lay1c-lay1m-lay2m)^3-vol3m/((4/3)*pi))^(1/3); %km

mass4m = mass4c/(F2/100)-mass3c; %kg mass lithosphere layer
```

```
vol4m = mass4m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay4m = (rad-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)-((rad-lay4c-
lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)^3-vol4m/((4/3)*pi))^(1/3); %km

%Calculate base (depth) of each layer
cth1 = lay4c+lay3c+lay2c+lay1c; %total thickness of crust in km
cmth1 = cth1+lay1m+lay2m+lay3m+lay4m;

A1c =
Atotc4*lay1c/cth1+Atotc3*lay2c/cth1+Atotc2*lay3c/cth1+Atotc1*lay4c/cth1
;
A1m = Atotm4*lay1m/(cmth1-cth1)+Atotm3*lay2m/(cmth1-
cth1)+Atotm2*lay3m/(cmth1-cth1)+Atotm1*lay4m/(cmth1-cth1);
Am = Atotm;
%% Calculate areotherm 1 Ga
b = zeros(d,33);
b1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
b(1,1) = 0; %Initial Pressure (GPa)
b(1,2) = 0; %Initial Depth (km)
b(1,3) = Tad1+273;
b(1,4) = (-4.877*b(1,1)^2+120.2*b(1,1)+1088)+273.15; %solidus
b(1,5) = Q01/1000; %This is the surface heat flow (W/m^2)
b(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   b(i,2)=b(i-1,2)+step; %km
   if b(i,2)<cth1
       b(i,1)=(rhoc*g*b(i,2)*1000)/10^9;
   else
       b(i,1)=(rhoc*g*cth1*1000)/10^9+(rhoml*g*((b(i,2)-
cth1)*1000))/10^9;
   end

   %These are the adiabat and solidus
   b(i,3)=slope*b(i,2)+b(1,3);
   b(i,4)=(-4.877*b(i,1)^2+120.2*b(i,1)+1088)+273.15; %solidus

   %This is the q value
   if b(i,2)<cth1
       q=b(i-1,5)-A1c*(b(i,2)-b(i-1,2))*1000;
   elseif b(i,2)>cth1 && b(i,2)<cmth1
       q=b(i-1,5)-A1m*(b(i,2)-b(i-1,2))*1000;
   else
       q=b(i-1,5)-Am*(b(i,2)-b(i-1,2))*1000;
   end
   b(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if b(i,2)<cth1
       k=kc;
```

```
    else
        k=klth;
    end
    b(i,6)=k; %Put this k value into the array

    %Step 2: Calculate the temperature at this b value
    %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
    if b(i,2)<cth1
        T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A1c*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
    elseif b(i,2)>cth1 && b(i,2)<cmth1
        T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A1m*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
    else
        T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-Am*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
    end
    b(i,7)=T;

    %Steps 5 and 6 must be iterated 10 times; use a for loop
    for j=8:2:33
        %Step 5: Calculate k using most recent temperature calculation
        if b(i,2)<cth1
            k=kc*(1+0.0015*b(i,2))/(1+0.0001*(b(i,j-1)-273));
        elseif b(i,2)>cth1
            if b(i,j-1)<500
                k=1/(0.0741+0.000502*b(i,j-1));
            else %Take the larger of these values
                k1=1/(0.0741+0.000502*b(i,j-1))+0.0023*(b(i,j-1)-500);
                %or
                k2=1.26*(1+b(i,2)/1000)+0.0023*(b(i,j-1)-500);
                if k1>k2
                    k=k1;
                else
                    k=k2;
                end
            end
        end
        b(i,j)=k; %Save the applicable k in the array

        b(1,j+1)=T0; %set T on first row

        %Step 6: Calculate the temperature using the most recent k
        if b(i,2)<cth1
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A1c*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        elseif b(i,2)>cth1 && b(i,2)<cmth1
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A1m*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        else
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
Am*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        end
        b(i,j+1)=T; %Stores the values for T in the array
    end %Ends the Step 5 and 6 loop
    b1(i,1)=b(i,1)*10000;
```

```
   b1(1,2)=b(1,33);
   b1(i,2)=b(i,33);
   A1(i,1)=b(i,33)-b(i,3);
   B1(i,1)=b(i,1);
   C1(i,1)=b(i,2);
   PFp1 = find(A1>0,1,'first');
   PFm1 = find(A1<0,1,'last');
   PF1 = (B1(PFp1)+B1(PFm1))/2; %calulate pressure
   dF1 = (C1(PFp1)+C1(PFm1))/2; %calculate crusth+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,4)=cth1;
w(2,4)=Q01;
w(3,4)=Tad1;

syms P0
eqn1 = slope2*P0+(Tad1+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P01 = min(vpasolve(eqn1,P0));
P01 = double(P01); %pressure where solidus and adiabat intersect
w(4,4)=P01;
w(5,4)=PF1;
w(6,4)=dF1;

dTdPsolidus1 = 2*(-4.877*PF1)+120.2; %slope of solidus Duncan et al.
(2018)
dTdF1 = dTdFa*PF1+dTdFb;
w(7,4)=dTdPsolidus1;
w(8,4)=dTdF1;

% Use P0 and PF to calculate F (percentage of melt produced)
F1 = ((dTdPsolidus1-slope2)/((deltaHF/C_P)+dTdF1))*(P01-PF1)*100;
w(9,4)=F1;

% Calculate the composition of the melt and error
SiO21 = (SiO2A*PF1+SiO2B)*F1^2+(SiO2C*PF1+SiO2D)*F1+(SiO2E*PF1+SiO2F);
TiO21 = TiO2A*F1^TiO2B;
Al2O31 =
(Al2O3A*PF1+Al2O3B)*F1^2+(Al2O3C*PF1+Al2O3D)*F1+(Al2O3E*PF1+Al2O3F);
Cr2O31 =
(Cr2O3A*PF1+Cr2O3B)*F1^2+(Cr2O3C*PF1+Cr2O3D)*F1+(Cr2O3E*PF1+Cr2O3F);
FeO1 = (FeOA*PF1+FeOB)*F1^2+(FeOC*PF1+FeOD)*F1+(FeOE*PF1+FeOF);
MnO1 = (MnOA*PF1+MnOB)*F1^2+(MnOC*PF1+MnOD)*F1+(MnOE*PF1+MnOF);
MgO1 = (MgOA*PF1+MgOB)*F1^2+(MgOC*PF1+MgOD)*F1+(MgOE*PF1+MgOF);
CaO1 = CaOA*F1^3+CaOB*F1^2+CaOC*F1+CaOD;
Na2O1 = (Na2OA*PF1+Na2OB)*F1^2+(Na2OC*PF1+Na2OD)*F1+(Na2OE*PF1+Na2OF);
K2O1 = K2OA*F1^K2OB;
P2O51 = P2O5A*F1^P2O5B;
w(10,4)=SiO21;
w(11,4)=TiO21;
w(12,4)=Al2O31;
w(13,4)=Cr2O31;
w(14,4)=FeO1;
w(15,4)=MnO1;
w(16,4)=MgO1;
w(17,4)=CaO1;
w(18,4)=Na2O1;
```

```
w(19,4)=K2O1;
w(20,4)=P2O51;

% Calculate the residual using batch melting-will become mantle lithos
SiO2res1 = (mantleSiO2-((F1/100)*SiO21))/(1-(F1/100));
TiO2res1 = (mantleTiO2-((F1/100)*TiO21))/(1-(F1/100));
Al2O3res1 = (mantleAl2O3-((F1/100)*Al2O31))/(1-(F1/100));
Cr2O3res1 = (mantleCr2O3-((F1/100)*Cr2O31))/(1-(F1/100));
FeOres1 = (mantleFeO-((F1/100)*FeO1))/(1-(F1/100));
MnOres1 = (mantleMnO-((F1/100)*MnO1))/(1-(F1/100));
MgOres1 = (mantleMgO-((F1/100)*MgO1))/(1-(F1/100));
CaOres1 = (mantleCaO-((F1/100)*CaO1))/(1-(F1/100));
Na2Ores1 = (mantleNa2O-((F1/100)*Na2O1))/(1-(F1/100));
K2Ores1 = (mantleK2O-((F1/100)*K2O1))/(1-(F1/100));
P2O5res1 = (mantleP2O5-((F1/100)*P2O51))/(1-(F1/100));
w(21,4)=SiO2res1;
w(22,4)=TiO2res1;
w(23,4)=Al2O3res1;
w(24,4)=Cr2O3res1;
w(25,4)=FeOres1;
w(26,4)=MnOres1;
w(27,4)=MgOres1;
w(28,4)=CaOres1;
w(29,4)=Na2Ores1;
w(30,4)=K2Ores1;
w(31,4)=P2O5res1;
%End of 1 Ga calculations


%% Present day
t = 0*10^9; %time in years, 0 is present
Q00 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad0 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc0 = Um/(F1+(1-F1)*DU); %Uranium in crust
Thc0 = Thm/(F1+(1-F1)*DTh); %Thorium in crust
Kc0 = Km/(F1+(1-F1)*DK); %Potassium in crust

Um0 = Um-F1*Uc0/(1-F1); %Uranium in mantle lithosphere
Thm0 = Thm-F1*Thc0/(1-F1); %Thorium in mantle lithosphere
Km0 = Km-F1*Kc0/(1-F1); %Potassium in mantle lithosphere

HU238c0 = Uc0*U238*H238U*exp(t*log(2)/U238t);
HU235c0 = Uc0*U235*H235U*exp(t*log(2)/U235t);
HTh232c0 = Thc0*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c0 = Kc0*K40*H40K*exp(t*log(2)/K40t);

HU238c1 = Uc1*U238*H238U*exp(t*log(2)/U238t);
HU235c1 = Uc1*U235*H235U*exp(t*log(2)/U235t);
HTh232c1 = Thc1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c1 = Kc1*K40*H40K*exp(t*log(2)/K40t);

HU238c2 = Uc2*U238*H238U*exp(t*log(2)/U238t);
```

```
HU235c2 = Uc2*U235*H235U*exp(t*log(2)/U235t);
HTh232c2 = Thc2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c2 = Kc2*K40*H40K*exp(t*log(2)/K40t);

HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m2 = Um2*U238*H238U*exp(t*log(2)/U238t);
HU235m2 = Um2*U235*H235U*exp(t*log(2)/U235t);
HTh232m2 = Thm2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m2 = Km2*K40*H40K*exp(t*log(2)/K40t);

HU238m1 = Um1*U238*H238U*exp(t*log(2)/U238t);
HU235m1 = Um1*U235*H235U*exp(t*log(2)/U235t);
HTh232m1 = Thm1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m1 = Km1*K40*H40K*exp(t*log(2)/K40t);

HU238m0 = Um0*U238*H238U*exp(t*log(2)/U238t);
HU235m0 = Um0*U235*H235U*exp(t*log(2)/U235t);
HTh232m0 = Thm0*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m0 = Km0*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc0 = (HU238c0+HU235c0+HTh232c0+HK40c0)*rhoc; %volumetric heat
production crust (W/m3)
Atotc1 = (HU238c1+HU235c1+HTh232c1+HK40c1)*rhoc; %volumetric heat
production crust (W/m3)
Atotc2 = (HU238c2+HU235c2+HTh232c2+HK40c2)*rhoc; %volumetric heat
production crust (W/m3)
Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat
production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
```

```
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm2 = (HU238m2+HU235m2+HTh232m2+HK40m2)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm1 = (HU238m1+HU235m1+HTh232m1+HK40m1)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm0 = (HU238m0+HU235m0+HTh232m1+HK40m0)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)

%Calculate the layer thicknesses
lay5c = 20*(F1/100); %km
vol5c = (4/3)*pi*(rad^3-(rad-lay5c)^3); %km^3 volume of crustal layer
mass5c = vol5c*(1000^3)*rhoc; %kg mass crustal layer

lay4c = (rad-lay5c)-((rad-lay5c)^3-vol4c/((4/3)*pi))^(1/3); %km

lay3c = (rad-lay5c-lay4c)-((rad-lay5c-lay4c)^3-vol3c/((4/3)*pi))^(1/3);
%km

lay2c = (rad-lay5c-lay4c-lay3c)-((rad-lay5c-lay4c-lay3c)^3-
vol2c/((4/3)*pi))^(1/3); %km

lay1c = (rad-lay5c-lay4c-lay3c-lay2c)-((rad-lay5c-lay4c-lay3c-lay2c)^3-
vol1c/((4/3)*pi))^(1/3); %km

lay1m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c)-((rad-lay5c-lay4c-lay3c-
lay2c-lay1c)^3-vol1m/((4/3)*pi))^(1/3); %km

lay2m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m)-((rad-lay5c-lay4c-
lay3c-lay2c-lay1c-lay1m)^3-vol2m/((4/3)*pi))^(1/3); %km

lay3m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m)-((rad-lay5c-
lay4c-lay3c-lay2c-lay1c-lay1m-lay2m)^3-vol3m/((4/3)*pi))^(1/3); %km

lay4m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)-((rad-
lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)^3-
vol4m/((4/3)*pi))^(1/3); %km

mass5m = mass5c/(F4/100)-mass5c; %kg mass lithosphere layer
vol5m = mass5m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay5m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m-lay4m)-
((rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m-lay4m)^3-
vol5m/((4/3)*pi))^(1/3); %km

%Calculate base (depth) of each layer
cth0 = lay5c+lay4c+lay3c+lay2c+lay1c; %total thickness of crust in km
cmth0 = cth0+lay1m+lay2m+lay3m+lay4m+lay5m;
```

```
A0c =
Atotc4*lay1c/cth0+Atotc3*lay2c/cth0+Atotc2*lay3c/cth0+Atotc1*lay4c/cth0
+Atotc0*lay5c/cth0;
A0m = Atotm4*lay1m/(cmth0-cth0)+Atotm3*lay2m/(cmth0-
cth0)+Atotm2*lay3m/(cmth0-cth0)+Atotm1*lay4m/(cmth0-
cth0)+Atotm0*lay5m/(cmth0-cth0);
Am = Atotm;
%% Calculate areotherm 0 Ga
c = zeros(d,33);
c1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
c(1,1) = 0; %Initial Pressure (GPa)
c(1,2) = 0; %Initial Depth (km)
c(1,3) = Tad0+273;
c(1,4) = (-4.877*c(1,1)^2+120.2*c(1,1)+1088)+273.15; %solidus
c(1,5) = Q00/1000; %This is the surface heat flow (W/m^2)
c(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   c(i,2)=c(i-1,2)+step; %km
   if c(i,2)<cth0
       c(i,1)=(rhoc*g*c(i,2)*1000)/10^9;
   else
       c(i,1)=(rhoc*g*cth0*1000)/10^9+(rhoml*g*((c(i,2)-
cth0)*1000))/10^9;
   end

   %These are the adiabat and solidus
   c(i,3)=slope*c(i,2)+c(1,3);
   c(i,4)=(-4.877*c(i,1)^2+120.2*c(i,1)+1088)+273.15; %solidus

   %This is the q value
   if c(i,2)<cth0
       q=c(i-1,5)-A0c*(c(i,2)-c(i-1,2))*1000;
   elseif c(i,2)>cth0 && c(i,2)<cmth0
       q=c(i-1,5)-A0c*(c(i,2)-c(i-1,2))*1000;
   else
       q=c(i-1,5)-Am*(c(i,2)-c(i-1,2))*1000;
   end
   c(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if c(i,2)<cth0
       k=kc;
   else
       k=klth;
   end
   c(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this z value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if c(i,2)<cth0
```

```
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A0c*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   elseif c(i,2)>cth0 && c(i,2)<cmth0
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A0m*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   else
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-Am*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   end
   c(i,7)=T;

   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
       %Step 5: Calculate k using most recent temperature calculation
       if c(i,2)<cth0
           k=kc*(1+0.0015*c(i,2))/(1+0.0001*(c(i,j-1)-273));
       elseif c(i,2)>cth0
           if c(i,j-1)<500
               k=1/(0.0741+0.000502*c(i,j-1));
           else %Take the larger of these values
               k1=1/(0.0741+0.000502*c(i,j-1))+0.0023*(c(i,j-1)-500);
               %or
               k2=1.26*(1+c(i,2)/1000)+0.0023*(c(i,j-1)-500);
               if k1>k2
                   k=k1;
               else
                   k=k2;
               end
           end
       end
       c(i,j)=k; %Save the applicable k in the array

       c(1,j+1)=T0; %set T on first row

       %Step 6: Calculate the temperature using the most recent k
       if c(i,2)<cth0
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A0c*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       elseif c(i,2)>cth0 && c(i,2)<cmth0
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A0m*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       else
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
Am*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       end
       c(i,j+1)=T; %Stores the values for T in the array
   end %Ends the Step 5 and 6 loop
   c1(i,1)=c(i,1)*10000;
   c1(1,2)=c(1,33);
   c1(i,2)=c(i,33);
   A0(i,1)=c(i,33)-c(i,3);
   B0(i,1)=c(i,1);
   C0(i,1)=c(i,2);
   PFp0 = find(A0>0,1,'first');
   PFm0 = find(A0<0,1,'last');
   PF0 = (B0(PFp0)+B0(PFm0))/2; %calulate pressure
```

```
   dF0 = (C0(PFp0)+C0(PFm0))/2; %calculate crust+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,5)=cth0;
w(2,5)=Q00;
w(3,5)=Tad0;

syms P0
eqn1 = slope2*P0+(Tad0+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P00 = min(vpasolve(eqn1,P0));
P00 = double(P00); %pressure where solidus and adiabat intersect
w(4,5)=P00;
w(5,5)=PF0;
w(6,5)=dF0;

dTdPsolidus0 = 2*(-4.877*PF0)+120.2; %slope of solidus Duncan et al.
(2018)
dTdF0 = dTdFa*PF0+dTdFb;
w(7,5)=dTdPsolidus0;
w(8,5)=dTdF0;

% Use P0 and PF to calculate F (percentage of melt produced)
F0 = ((dTdPsolidus0-slope2)/((deltaHF/C_P)+dTdF0))*(P00-PF0)*100;
w(9,5)=F0;

% Calculate the composition of the melt and error
SiO20 = (SiO2A*PF0+SiO2B)*F0^2+(SiO2C*PF0+SiO2D)*F0+(SiO2E*PF0+SiO2F);
TiO20 = TiO2A*F0^TiO2B;
Al2O30 =
(Al2O3A*PF0+Al2O3B)*F0^2+(Al2O3C*PF0+Al2O3D)*F0+(Al2O3E*PF0+Al2O3F);
Cr2O30 =
(Cr2O3A*PF0+Cr2O3B)*F0^2+(Cr2O3C*PF0+Cr2O3D)*F0+(Cr2O3E*PF0+Cr2O3F);
FeO0 = (FeOA*PF0+FeOB)*F0^2+(FeOC*PF0+FeOD)*F0+(FeOE*PF0+FeOF);
MnO0 = (MnOA*PF0+MnOB)*F0^2+(MnOC*PF0+MnOD)*F0+(MnOE*PF0+MnOF);
MgO0 = (MgOA*PF0+MgOB)*F0^2+(MgOC*PF0+MgOD)*F0+(MgOE*PF0+MgOF);
CaO0 = CaOA*F0^3+CaOB*F0^2+CaOC*F0+CaOD;
Na2O0 = (Na2OA*PF0+Na2OB)*F0^2+(Na2OC*PF0+Na2OD)*F0+(Na2OE*PF0+Na2OF);
K2O0 = K2OA*F0^K2OB;
P2O50 = P2O5A*F0^P2O5B;
w(10,5)=SiO20;
w(11,5)=TiO20;
w(12,5)=Al2O30;
w(13,5)=Cr2O30;
w(14,5)=FeO0;
w(15,5)=MnO0;
w(16,5)=MgO0;
w(17,5)=CaO0;
w(18,5)=Na2O0;
w(19,5)=K2O0;
w(20,5)=P2O50;

% Calculate the residual using batch melting-will become mantle lithos
SiO2res0 = (mantleSiO2-((F0/100)*SiO20))/(1-(F0/100));
TiO2res0 = (mantleTiO2-((F0/100)*TiO20))/(1-(F0/100));
Al2O3res0 = (mantleAl2O3-((F0/100)*Al2O30))/(1-(F0/100));
Cr2O3res0 = (mantleCr2O3-((F0/100)*Cr2O30))/(1-(F0/100));
```

```
FeOres0 = (mantleFeO-((F0/100)*FeO0))/(1-(F0/100));
MnOres0 = (mantleMnO-((F0/100)*MnO0))/(1-(F0/100));
MgOres0 = (mantleMgO-((F0/100)*MgO0))/(1-(F0/100));
CaOres0 = (mantleCaO-((F0/100)*CaO0))/(1-(F0/100));
Na2Ores0 = (mantleNa2O-((F0/100)*Na2O0))/(1-(F0/100));
K2Ores0 = (mantleK2O-((F0/100)*K2O0))/(1-(F0/100));
P2O5res0 = (mantleP2O5-((F0/100)*P2O50))/(1-(F0/100));
w(21,5)=SiO2res0;
w(22,5)=TiO2res0;
w(23,5)=Al2O3res0;
w(24,5)=Cr2O3res0;
w(25,5)=FeOres0;
w(26,5)=MnOres0;
w(27,5)=MgOres0;
w(28,5)=CaOres0;
w(29,5)=Na2Ores0;
w(30,5)=K2Ores0;
w(31,5)=P2O5res0;
%End of present day calculations


%% Now plot the results.
%The areotherms and adiabats are plotted against depth on the left
axis,while the solidus is plotted against pressure on the right axis
vt1 = 1/255*[232,119,34];
str1 = '#001932'; %'#000032';
color1 = sscanf(str1(2:end),'%2x%2x%2x',[1 3])/255;
str2 = '#00366C'; %'#000096';
color2 = sscanf(str2(2:end),'%2x%2x%2x',[1 3])/255;
str3 = '#0068D0'; %'#0000FA';
color3 = sscanf(str3(2:end),'%2x%2x%2x',[1 3])/255;
str4 = '#359AFF'; %'#5F5FFF';
color4 = sscanf(str4(2:end),'%2x%2x%2x',[1 3])/255;
str5 = '#99CCFF';
color5 = sscanf(str5(2:end),'%2x%2x%2x',[1 3])/255;

yyaxis left
plot(x(:,33),x(:,2),'-','color',color1,'LineWidth',2,'DisplayName','4
Ga') %areotherm
hold on
plot (y(:,33),y(:,2),'-.','color',color2,'LineWidth',2,'DisplayName','3
Ga')
hold on
plot (z(:,33),z(:,2),'--','color',color3,'LineWidth',2,'DisplayName','2
Ga')
hold on
plot (b(:,33),b(:,2),':','color',color4,'LineWidth',2,'DisplayName','1
Ga')
hold on
plot (c(:,33),c(:,2),'-','color',color5,'LineWidth',2,'DisplayName','0
Ga')
hold on
plot (x(:,3),x(:,2),'-
','color',color1,'LineWidth',1.5,'DisplayName','adiabat') %adiabat
hold on
```

```
plot (y(:,3),y(:,2),'-
.','color',color2,'LineWidth',1.5,'DisplayName','adiabat')
hold on
plot (z(:,3),z(:,2),'--
','color',color3,'LineWidth',1.5,'DisplayName','adiabat')
hold on
plot
(b(:,3),b(:,2),':','color',color4,'LineWidth',1.5,'DisplayName','adiaba
t')
hold on
plot (c(:,3),c(:,2),'-
','color',color5,'LineWidth',1.5,'DisplayName','adiabat')
legend('location','southwest');

ylabel('Depth (km)')
ylim([0 x(3000,2)])
set(gca,'YDir','reverse','YMinorTick','on','Ycolor','k')

title('Areotherms Average Melt')
xlabel('Temperature (K)')
set(gca,'XAxisLocation','top','linestyleorder','-
','XMinorTick','on','Xcolor','k')
xlim([200 2000])

yyaxis right
plot(x(:,4),x(:,1),'color',vt1,'LineWidth',2,'DisplayName','solidus')
%solidus
ylim([0 x(3000,1)])
ylabel('Pressure (GPa)')
set(gca,'YDir','reverse','YMinorTick','on','Ycolor','k')

set(gca,'TickDir','both')
%% Save the results
save Areotherm4GaAvemelt.txt x -ascii
save Areotherm3GaAvemelt.txt y -ascii
save Areotherm2GaAvemelt.txt z -ascii
save Areotherm1GaAvemelt.txt b -ascii
save Areotherm0GaAvemelt.txt c -ascii
save pt0Avemelt.txt c1 -ascii
save pt1Avemelt.txt b1 -ascii
save pt2Avemelt.txt z1 -ascii
save pt3Avemelt.txt y1 -ascii
save pt4Avemelt.txt x1 -ascii
```

B.2 Melt-Layer Model

```
%-----------------------------------------------------------------------
% Project: Areotherm_MeltLayer.m
% Author: Fiona McGroarty
% Date Written: 19 October 2019
% Last Updated: 19 May 2021
% Purpose: This program will calculate the HPE concentration in the
depleted mantle lithosphere and then use those values to calculate the
parameters for an areotherm through time and plot the result. It will
calculate the composition of melt produced at a particular time and the
composition of the residual, with error. It will then repeat the process
using the previous calculation as the starting value for the next timestep
and the previous melt as the most recent layer in the crust.
% Addiitonal files needed: MeltModel.xlsx
%-----------------------------------------------------------------------

clear
clc

%% Load the initial values from the Excel file meltmodel.xlsx
% Load values needed later to calculate F (dT/dPsolidus, dT/dPadiabat,
deltaHF, C_P. dT/dF). P0 and Pf will be calculated in a later step
prior to calculating F

% 1. Parameter read in
param = importdata('OxideCoeff.txt');

% 2. Variable read in
SiO2A = param(1,1);
SiO2B = param(1,2);
SiO2C = param(1,3);
SiO2D = param(1,4);
SiO2E = param(1,5);
SiO2F = param(1,6);

TiO2A = param(2,1);
TiO2B = param(2,2);
TiO2C = param(2,3);
TiO2D = param(2,4);
TiO2E = param(2,5);
TiO2F = param(2,6);

Al2O3A = param(3,1);
Al2O3B = param(3,2);
Al2O3C = param(3,3);
Al2O3D = param(3,4);
Al2O3E = param(3,5);
Al2O3F = param(3,6);

Cr2O3A = param(4,1);
Cr2O3B = param(4,2);
Cr2O3C = param(4,3);
Cr2O3D = param(4,4);
Cr2O3E = param(4,5);
```

```
Cr2O3F = param(4,6);

FeOA = param(5,1);
FeOB = param(5,2);
FeOC = param(5,3);
FeOD = param(5,4);
FeOE = param(5,5);
FeOF = param(5,6);

MnOA = param(6,1);
MnOB = param(6,2);
MnOC = param(6,3);
MnOD = param(6,4);
MnOE = param(6,5);
MnOF = param(6,6);

MgOA = param(7,1);
MgOB = param(7,2);
MgOC = param(7,3);
MgOD = param(7,4);
MgOE = param(7,5);
MgOF = param(7,6);

CaOA = param(8,1);
CaOB = param(8,2);
CaOC = param(8,3);
CaOD = param(8,4);
CaOE = param(8,5);
CaOF = param(8,6);

Na2OA = param(9,1);
Na2OB = param(9,2);
Na2OC = param(9,3);
Na2OD = param(9,4);
Na2OE = param(9,5);
Na2OF = param(9,6);

K2OA = param(10,1);
K2OB = param(10,2);
K2OC = param(10,3);
K2OD = param(10,4);
K2OE = param(10,5);
K2OF = param(10,6);

P2O5A = param(11,1);
P2O5B = param(11,2);
P2O5C = param(11,3);
P2O5D = param(11,4);
P2O5E = param(11,5);
P2O5F = param(11,6);

%% Mars parameters
rhoc = 2900; %kg/m^3 crustal density
rhoml = 3400; %kg/m^3 mantle lithosphere density
rhom = 3700; %kg/m^3 mantle density
```

```
T0 = 220; %Temperature at the surface in K
g = 3.71; %m/s^2 gravitational acceleration on Mars
slope = 0.18; %adiabat slope (deg/km)
slope2 = (slope/1000)/(g*rhoml)*1E9; %adiabat slope
deltaHF = 640000; %J/kg (Kiefer, 2003)
C_P = 1200; %J/K*kg (Kiefer, 2003)
rad = 3390; %km radius planet
d = 5000; %The number of rows of the array to hold the values
step = 0.1; %step for depth km

dTdFa = -0.639; %calculated from experimental data
dTdFb = 5.2926; %calculated from experimental data

%Input the oxides in the undepleted mantle (Dreibus and Wanke 1985)
mantleSiO2 = 44.4;
mantleTiO2 = 0.14;
mantleAl2O3 = 3.02;
mantleFeO = 17.9;
mantleCr2O3 = 0.76;
mantleMnO = 0.47;
mantleMgO = 30.2;
mantleCaO = 2.45;
mantleNa2O = 0.50;
mantleK2O = 0.04;
mantleP2O5 = 0.16;

%Initial thermal conductivity guess
kc = 2.5; %crust thermal condutivity(W/mK)
klth = 4.0; %mantle lithosphere thermal conductivity(W/mK)

% Heat Production Constants
U238 = 0.992834; %fraction of U238 in the solar system
U235 = 0.00711; %fraction of U235
U238t = 4.469E9; %half-life in yr
U235t = 0.704E9; %half-life in yr
H238U = 9.464E-5; %rate of heat release 238U (W/kg)
H235U = 5.687E-4; %rate of heat release 235U (W/kg)

Th232 = 0.9998; %fraction of Th232
Th232t = 14.00E9; %half-life in yr
H232Th = 2.638E-5; %rate of heat release 232Th (W/kg)

K40 = 1.193E-4; %fraction of K40
K40t = 1.268E9; %half-life in yr
H40K = 2.97E-5; %rate of heat release 40K (W/kg)

%Bulk peridotite/melt partition coefficients
DK = 0.003; %Davis et al. 2011 EPSL, garnet peridodite
DU = 1.1e-4; %Beatie 1993 EPSL, spinel peridodite
DTh = 1.7e-4; %Beatie 1993 EPSL, spinel peridotite

%HPE concentrations
Uc = 180/1E9; %Uranium in crust (Taylor and McLennon, 2009)
Thc = 700/1E9; %Thorium in crust (Taylor and McLennon, 2009)
Kc = 3740/1E6; %Potassium in crust (Taylor and McLennon, 2009)
```

```
Um = 16/1E9; %Uranium in mantle Dreibus and Wanke
Thm = 56/1E9; %Thorium in mantle Dreibus and Wanke
Km = 315/1E6; %Potassium in mantle Dreibus and Wanke


%% Areotherm 4 Ga
% Input the initial parameters TO BE CHANGED WITH TIME
t = 4*10^9; %time in years, 0 is present
Q04 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad4 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Forig = 0.1; %initial guess for melt fraction for first crust
Uc4 = Um/(Forig+(1-Forig)*DU); %Uranium in crust
Thc4 = Thm/(Forig+(1-Forig)*DTh); %Thorium in crust
Kc4 = Km/(Forig+(1-Forig)*DK); %Potassium in crust
Um4 = (Um-Forig*Uc4)/(1-Forig); %Solve for mantle lithosphere
concentration
Thm4 = (Thm-Forig*Thc4)/(1-Forig); %Solve for mantle lithosphere
concentration
Km4 = (Km-Forig*Kc4)/(1-Forig); %Solve for mantle lithosphere
concentration

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)
A1c = Atotc4;
A1m = Atotm4;
Am = Atotm;

%Calculate the layer thicknesses
lay1c = 20; %initial crustal thickness from Taylor 2009
vol1c = (4/3)*pi*(rad^3-(rad-lay1c)^3); %km^3 volume of crustal layer
mass1c = vol1c*(1000^3)*rhoc; %kg mass crustal layer
```

```
mass1m = mass1c/Forig-mass1c; %kg mass lithosphere layer
vol1m = mass1m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay1m = (rad-lay1c)-((rad-lay1c)^3-vol1m/((4/3)*pi))^(1/3); %km
thicknes of lithosphere layer

cth4 = lay1c; %total thickness of crust in km
cmth4 = cth4+lay1m;
%% Calculate areotherm at 4 Ga
x = zeros(d,33);
x1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
x(1,1) = 0; %Initial Pressure (GPa)
x(1,2) = 0; %Initial Depth (km)
x(1,3) = Tad4+273;
x(1,4) = (-4.877*x(1,1)^2+120.2*x(1,1)+1088)+273.15; %solidus
x(1,5) = Q04/1000; %This is the surface heat flow (W/m^2)
x(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   x(i,2)=x(i-1,2)+step; %km
   if x(i,2)<cth4
       x(i,1)=(rhoc*g*x(i,2)*1000)/10^9;
   else
       x(i,1)=(rhoc*g*cth4*1000)/10^9+(rhoml*g*((x(i,2)-
cth4)*1000))/10^9;
   end

   %These are the adiabat and solidus
   x(i,3)=slope*x(i,2)+x(1,3);
   x(i,4)=(-4.877*x(i,1)^2+120.2*x(i,1)+1088)+273.15; %solidus

   %This is the q value
   if x(i,2)<lay1c
       q=x(i-1,5)-A1c*(x(i,2)-x(i-1,2))*1000;
   elseif x(i,2)>lay1c && x(i,2)<cmth4
       q=x(i-1,5)-A1m*(x(i,2)-x(i-1,2))*1000;
   else
       q=x(i-1,5)-Am*(x(i,2)-x(i-1,2))*1000;
   end
   x(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if x(i,2)<lay1c
       k=kc;
   else
       k=klth;
   end
   x(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this z value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
```

```
   if x(i,2)<lay1c
       T=x(i-1,7)+x(i,5)*((x(i,2)-x(i-1,2))*1000)/x(i,6)-A1c*((x(i,2)-
x(i-1,2))*1000)^2/(2*x(i,6));
   elseif x(i,2)>lay1c && x(i,2)<cmth4
       T=x(i-1,7)+x(i,5)*((x(i,2)-x(i-1,2))*1000)/x(i,6)-A1m*((x(i,2)-
x(i-1,2))*1000)^2/(2*x(i,6));
   else
       T=x(i-1,7)+x(i,5)*((x(i,2)-x(i-1,2))*1000)/x(i,6)-Am*((x(i,2)-
x(i-1,2))*1000)^2/(2*x(i,6));
   end
   x(i,7)=T;

   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
       %Step 5: Calculate k using most recent temperature calculation
       if x(i,2)<lay1c
           k=kc*(1+0.0015*x(i,2))/(1+0.0001*(x(i,j-1)-273));
       elseif x(i,2)>lay1c
           if x(i,j-1)<500
               k=1/(0.0741+0.000502*x(i,j-1));
           else %Take the larger of these values
               k1=1/(0.0741+0.000502*x(i,j-1))+0.0023*(x(i,j-1)-500);
               %or
               k2=1.26*(1+x(i,2)/1000)+0.0023*(x(i,j-1)-500);
               if k1>k2
                   k=k1;
               else
                   k=k2;
               end
           end
       end
       x(i,j)=k; %Save the applicable k in the array

       x(1,j+1)=T0; %set T on first row

       %Step 6: Calculate the temperature using the most recent k
       if x(i,2)<lay1c
           T=x(i-1,j+1)+x(i,5)*(x(i,2)-x(i-1,2))*1000/x(i,j)-
A1c*((x(i,2)-x(i-1,2))*1000)^2/(2*x(i,j));
       elseif x(i,2)>lay1c && x(i,2)<cmth4
           T=x(i-1,j+1)+x(i,5)*(x(i,2)-x(i-1,2))*1000/x(i,j)-
A1m*((x(i,2)-x(i-1,2))*1000)^2/(2*x(i,j));
       else
           T=x(i-1,j+1)+x(i,5)*(x(i,2)-x(i-1,2))*1000/x(i,j)-
Am*((x(i,2)-x(i-1,2))*1000)^2/(2*x(i,j));
       end
       x(i,j+1)=T; %Stores the values for T in the array
   end %Ends the Step 5 and 6 loop
   x1(i,1)=x(i,1)*10000;
   x1(1,2)=x(1,33);
   x1(i,2)=x(i,33);
   A4(i,1)=x(i,33)-x(i,3);
   B4(i,1)=x(i,1);
   C4(i,1)=x(i,2);
   PFp4 = find(A4>0,1,'first');
   PFm4 = find(A4<0,1,'last');
```

```
   PF4 = (B4(PFp4)+B4(PFm4))/2; %calulate pressure
   dF4 = (C4(PFp4)+C4(PFm4))/2; %calculate lithosphere+crust thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w=zeros(31,5);
w(1,1)=cth4;
w(2,1)=Q04;
w(3,1)=Tad4;

syms P0
eqn1 = slope2*P0+(Tad4+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P04 = min(vpasolve(eqn1,P0)); %pressure of adiabat and solidus
intersection
P04 = double(P04);
w(4,1)=P04;
w(5,1)=PF4;
w(6,1)=dF4;

dTdPsolidus4 = 2*(-4.877*PF4)+120.2; %solidus slope
dTdF4 = dTdFa*PF4+dTdFb;
w(7,1)=dTdPsolidus4;
w(8,1)=dTdF4;

% Use P0 and PF to calculate F (percentage of melt produced)
F4 = ((dTdPsolidus4-slope2)/((deltaHF/C_P)+dTdF4))*(P04-PF4)*100;
w(9,1)=F4;

% Calculate the composition of the melt and error
SiO24 = (SiO2A*PF4+SiO2B)*F4^2+(SiO2C*PF4+SiO2D)*F4+(SiO2E*PF4+SiO2F);
TiO24 = TiO2A*F4^TiO2B;
Al2O34 =
(Al2O3A*PF4+Al2O3B)*F4^2+(Al2O3C*PF4+Al2O3D)*F4+(Al2O3E*PF4+Al2O3F);
Cr2O34 =
(Cr2O3A*PF4+Cr2O3B)*F4^2+(Cr2O3C*PF4+Cr2O3D)*F4+(Cr2O3E*PF4+Cr2O3F);
FeO4 = (FeOA*PF4+FeOB)*F4^2+(FeOC*PF4+FeOD)*F4+(FeOE*PF4+FeOF);
MnO4 = (MnOA*PF4+MnOB)*F4^2+(MnOC*PF4+MnOD)*F4+(MnOE*PF4+MnOF);
MgO4 = (MgOA*PF4+MgOB)*F4^2+(MgOC*PF4+MgOD)*F4+(MgOE*PF4+MgOF);
CaO4 = CaOA*F4^3+CaOB*F4^2+CaOC*F4+CaOD;
Na2O4 = (Na2OA*PF4+Na2OB)*F4^2+(Na2OC*PF4+Na2OD)*F4+(Na2OE*PF4+Na2OF);
K2O4 = K2OA*F4^K2OB;
P2O54 = P2O5A*F4^P2O5B;
w(10,1)=SiO24;
w(11,1)=TiO24;
w(12,1)=Al2O34;
w(13,1)=Cr2O34;
w(14,1)=FeO4;
w(15,1)=MnO4;
w(16,1)=MgO4;
w(17,1)=CaO4;
w(18,1)=Na2O4;
w(19,1)=K2O4;
w(20,1)=P2O54;

% Calculate the residual using batch melting- will become mantle lithos
SiO2res4 = (mantleSiO2-((F4/100)*SiO24))/(1-(F4/100));
TiO2res4 = (mantleTiO2-((F4/100)*TiO24))/(1-(F4/100));
```

```
Al2O3res4 = (mantleAl2O3-((F4/100)*Al2O34))/(1-(F4/100));
Cr2O3res4 = (mantleCr2O3-((F4/100)*Cr2O34))/(1-(F4/100));
FeOres4 = (mantleFeO-((F4/100)*FeO4))/(1-(F4/100));
MnOres4 = (mantleMnO-((F4/100)*MnO4))/(1-(F4/100));
MgOres4 = (mantleMgO-((F4/100)*MgO4))/(1-(F4/100));
CaOres4 = (mantleCaO-((F4/100)*CaO4))/(1-(F4/100));
Na2Ores4 = (mantleNa2O-((F4/100)*Na2O4))/(1-(F4/100));
K2Ores4 = (mantleK2O-((F4/100)*K2O4))/(1-(F4/100));
P2O5res4 = (mantleP2O5-((F4/100)*P2O54))/(1-(F4/100));
w(21,1)=SiO2res4;
w(22,1)=TiO2res4;
w(23,1)=Al2O3res4;
w(24,1)=Cr2O3res4;
w(25,1)=FeOres4;
w(26,1)=MnOres4;
w(27,1)=MgOres4;
w(28,1)=CaOres4;
w(29,1)=Na2Ores4;
w(30,1)=K2Ores4;
w(31,1)=P2O5res4;
%End of 4 Ga (will become one distinct layer)


%% 3 Ga
% Input the initial parameters TO BE CHANGED WITH TIME
t = 3*10^9; %time in years, 0 is present
Q03 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad3 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc3 = Um/(F4+(1-F4)*DU); %Uranium in crust
Thc3 = Thm/(F4+(1-F4)*DTh); %Thorium in crust
Kc3 = Km/(F4+(1-F4)*DK); %Potassium in crust

Um3 = Um-F4*Uc3/(1-F4); %Uranium in crust
Thm3 = Thm-F4*Thc3/(1-F4); %Thorium in crust
Km3 = Km-F4*Kc3/(1-F4); %Potassium in crust

HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
```

```
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat
production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)
A2c = Atotc3;
A1c = Atotc4;
A1m = Atotm4;
A2m = Atotm3;
Am = Atotm;

%Calculate the layer thicknesses
lay2c = 20*(F4/100); %km
vol2c = (4/3)*pi*(rad^3-(rad-lay2c)^3); %km^3 volume of crustal layer
mass2c = vol2c*(1000^3)*rhoc; %kg mass crustal layer

lay1c = (rad-lay2c)-((rad-lay2c)^3-vol1c/((4/3)*pi))^(1/3); %km new
thickness crustal layer

lay1m = (rad-lay2c-lay1c)-((rad-lay2c-lay1c)^3-vol1m/((4/3)*pi))^(1/3);
%km new thickness lithosphere layer

mass2m = mass2c/(F4/100)-mass2c; %kg mass lithosphere layer
vol2m = mass2m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay2m = (rad-lay2c-lay1c-lay1m)-((rad-lay2c-lay1c-lay1m)^3-
vol2m/((4/3)*pi))^(1/3); %km

%Calculate base (depth) of each layer
cth3 = lay2c+lay1c; %total thickness of crust in km
cmth3a = cth3+lay1m;
cmth3 = cmth3a+lay2m;
%% Calculate areotherm 3 Ga

y = zeros(d,33);
y1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
y(1,1) = 0; %Initial Pressure (GPa)
y(1,2) = 0; %Initial Depth (km)
y(1,3) = Tad3+273;
```

```
y(1,4) = (-4.877*y(1,1)^2+120.2*y(1,1)+1088)+273.15; %solidus
y(1,5) = Q03/1000; %This is the surface heat flow (W/m^2)
y(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   y(i,2)=y(i-1,2)+step; %km
   if y(i,2)<cth3
       y(i,1)=(rhoc*g*y(i,2)*1000)/10^9;
   else
       y(i,1)=(rhoc*g*cth3*1000)/10^9+(rhoml*g*((y(i,2)-
cth3)*1000))/10^9;
   end

   %These are the adiabat and solidus
   y(i,3)=slope*y(i,2)+y(1,3);
   y(i,4)=(-4.877*y(i,1)^2+120.2*y(i,1)+1088)+273.15; %solidus

   %This is the q value
   if y(i,2)<lay2c
       q=y(i-1,5)-A2c*(y(i,2)-y(i-1,2))*1000;
   elseif y(i,2)>lay2c && y(i,2)<cth3
       q=y(i-1,5)-A1c*(y(i,2)-y(i-1,2))*1000;
   elseif y(i,2)>cth3 && y(i,2)<cmth3a
       q=y(i-1,5)-A1m*(y(i,2)-y(i-1,2))*1000;
   elseif y(i,2)>cmth3a && y(i,2)<cmth3
       q=y(i-1,5)-A2m*(y(i,2)-y(i-1,2))*1000;
   else
       q=y(i-1,5)-Am*(y(i,2)-y(i-1,2))*1000;
   end
   y(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if y(i,2)<cth3
       k=kc;
   else
       k=klth;
   end
   y(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this z value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if y(i,2)<lay2c
       T=y(i-1,7)+y(i,5)*((y(i,2)-y(i-1,2))*1000)/y(i,6)-A2c*((y(i,2)-
y(i-1,2))*1000)^2/(2*y(i,6));
   elseif y(i,2)>lay2c && y(i,2)<cth3
       T=y(i-1,7)+y(i,5)*((y(i,2)-y(i-1,2))*1000)/y(i,6)-A1c*((y(i,2)-
y(i-1,2))*1000)^2/(2*y(i,6));
   elseif y(i,2)>cth3 && y(i,2)<cmth3a
       T=y(i-1,7)+y(i,5)*((y(i,2)-y(i-1,2))*1000)/y(i,6)-A1m*((y(i,2)-
y(i-1,2))*1000)^2/(2*y(i,6));
   elseif y(i,2)>cmth3a && y(i,2)<cmth3
       T=y(i-1,7)+y(i,5)*((y(i,2)-y(i-1,2))*1000)/y(i,6)-A2m*((y(i,2)-
y(i-1,2))*1000)^2/(2*y(i,6));
   else
```

```
        T=y(i-1,7)+y(i,5)*((y(i,2)-y(i-1,2))*1000)/y(i,6)-Am*((y(i,2)-
y(i-1,2))*1000)^2/(2*y(i,6));
    end
    y(i,7)=T;

    %Steps 5 and 6 must be iterated 10 times; use a for loop
    for j=8:2:33
        %Step 5: Calculate k using most recent temperature calculation
        if y(i,2)<lay2c
            k=kc*(1+0.0015*y(i,2))/(1+0.0015*(y(i,j-1)-273));
        elseif y(i,2)>lay2c && y(i,2)<cth3
            k=kc*(1+0.0015*y(i,2))/(1+0.0001*(y(i,j-1)-273));
        elseif y(i,2)>cth3
            if y(i,j-1)<500
                k=1/(0.0741+0.000502*y(i,j-1));
            else %Take the larger of these values
                k1=1/(0.0741+0.000502*y(i,j-1))+0.0023*(y(i,j-1)-500);
                %or
                k2=1.26*(1+y(i,2)/1000)+0.0023*(y(i,j-1)-500);
                if k1>k2
                    k=k1;
                else
                    k=k2;
                end
            end
        end
        y(i,j)=k; %Save the applicable k in the array

        y(1,j+1)=T0; %set T on first row

        %Step 6: Calculate the temperature using the most recent k
        if y(i,2)<lay2c
            T=y(i-1,j+1)+y(i,5)*(y(i,2)-y(i-1,2))*1000/y(i,j)-
A2c*((y(i,2)-y(i-1,2))*1000)^2/(2*y(i,j));
        elseif y(i,2)>lay2c && y(i,2)<cth3
            T=y(i-1,j+1)+y(i,5)*(y(i,2)-y(i-1,2))*1000/y(i,j)-
A1c*((y(i,2)-y(i-1,2))*1000)^2/(2*y(i,j));
        elseif y(i,2)>cth3 && y(i,2)<cmth3a
            T=y(i-1,j+1)+y(i,5)*(y(i,2)-y(i-1,2))*1000/y(i,j)-
A1m*((y(i,2)-y(i-1,2))*1000)^2/(2*y(i,j));
        elseif y(i,2)>cmth3a && y(i,2)<cmth3
            T=y(i-1,j+1)+y(i,5)*(y(i,2)-y(i-1,2))*1000/y(i,j)-
A2m*((y(i,2)-y(i-1,2))*1000)^2/(2*y(i,j));
        else
            T=y(i-1,j+1)+y(i,5)*(y(i,2)-y(i-1,2))*1000/y(i,j)-
Am*((y(i,2)-y(i-1,2))*1000)^2/(2*y(i,j));
        end
        y(i,j+1)=T; %Stores the values for T in the array
    end %Ends the Step 5 and 6 loop
    y1(i,1)=y(i,1)*10000;
    y1(1,2)=y(1,33);
    y1(i,2)=y(i,33);
    A3(i,1)=y(i,33)-y(i,3);
    B3(i,1)=y(i,1);
    C3(i,1)=y(i,2);
    PFp3 = find(A3>0,1,'first');
```

```
   PFm3 = find(A3<0,1,'last');
   PF3 = (B3(PFp3)+B3(PFm3))/2; %calulate pressure
   dF3 = (C3(PFp3)+C3(PFm3))/2; %calculate crust+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,2)=cth3;
w(2,2)=Q03;
w(3,2)=Tad3;

syms P0
eqn1 = slope2*P0+(Tad3+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P03 = min(vpasolve(eqn1,P0)); %pressure of solidus-adiabat intersection
P03 = double(P03);
w(4,2)=P03;
w(5,2)=PF3;
w(6,2)=dF3;

dTdPsolidus3 = 2*(-4.877*PF3)+120.2; %slope of solidus Duncan et al.
(2018)
dTdF3 = dTdFa*PF3+dTdFb;
w(7,2)=dTdPsolidus3;
w(8,2)=dTdF3;

% Use P0 and PF to calculate F (percentage of melt produced)
F3 = ((dTdPsolidus3-slope2)/((deltaHF/C_P)+dTdF3))*(P03-PF3)*100;
w(9,2)=F3;

% Calculate the composition of the melt and error
SiO23 = (SiO2A*PF3+SiO2B)*F3^2+(SiO2C*PF3+SiO2D)*F3+(SiO2E*PF3+SiO2F);
TiO23 = TiO2A*F3^TiO2B;
Al2O33 =
(Al2O3A*PF3+Al2O3B)*F3^2+(Al2O3C*PF3+Al2O3D)*F3+(Al2O3E*PF3+Al2O3F);
Cr2O33 =
(Cr2O3A*PF3+Cr2O3B)*F3^2+(Cr2O3C*PF3+Cr2O3D)*F3+(Cr2O3E*PF3+Cr2O3F);
FeO3 = (FeOA*PF3+FeOB)*F3^2+(FeOC*PF3+FeOD)*F3+(FeOE*PF3+FeOF);
MnO3 = (MnOA*PF3+MnOB)*F3^2+(MnOC*PF3+MnOD)*F3+(MnOE*PF3+MnOF);
MgO3 = (MgOA*PF3+MgOB)*F3^2+(MgOC*PF3+MgOD)*F3+(MgOE*PF3+MgOF);
CaO3 = CaOA*F3^3+CaOB*F3^2+CaOC*F3+CaOD;
Na2O3 = (Na2OA*PF3+Na2OB)*F3^2+(Na2OC*PF3+Na2OD)*F3+(Na2OE*PF3+Na2OF);
K2O3 = K2OA*F3^K2OB;
P2O53 = P2O5A*F3^P2O5B;
w(10,2)=SiO23;
w(11,2)=TiO23;
w(12,2)=Al2O33;
w(13,2)=Cr2O33;
w(14,2)=FeO3;
w(15,2)=MnO3;
w(16,2)=MgO3;
w(17,2)=CaO3;
w(18,2)=Na2O3;
w(19,2)=K2O3;
w(20,2)=P2O53;

% Calculate the residual using batch melting- will become mantle lithos
SiO2res3 = (mantleSiO2-((F3/100)*SiO23))/(1-(F3/100));
TiO2res3 = (mantleTiO2-((F3/100)*TiO23))/(1-(F3/100));
```

```
Al2O3res3 = (mantleAl2O3-((F3/100)*Al2O33))/(1-(F3/100));
Cr2O3res3 = (mantleCr2O3-((F3/100)*Cr2O33))/(1-(F3/100));
FeOres3 = (mantleFeO-((F3/100)*FeO3))/(1-(F3/100));
MnOres3 = (mantleMnO-((F3/100)*MnO3))/(1-(F3/100));
MgOres3 = (mantleMgO-((F3/100)*MgO3))/(1-(F3/100));
CaOres3 = (mantleCaO-((F3/100)*CaO3))/(1-(F3/100));
Na2Ores3 = (mantleNa2O-((F3/100)*Na2O3))/(1-(F3/100));
K2Ores3 = (mantleK2O-((F3/100)*K2O3))/(1-(F3/100));
P2O5res3 = (mantleP2O5-((F3/100)*P2O53))/(1-(F3/100));
w(21,2)=SiO2res3;
w(22,2)=TiO2res3;
w(23,2)=Al2O3res3;
w(24,2)=Cr2O3res3;
w(25,2)=FeOres3;
w(26,2)=MnOres3;
w(27,2)=MgOres3;
w(28,2)=CaOres3;
w(29,2)=Na2Ores3;
w(30,2)=K2Ores3;
w(31,2)=P2O5res3;
%End of 3 Ga (becomes next distinct layer)



%% 2 Ga
t = 2*10^9; %time in years, 0 is present
Q02 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad2 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc2 = Um/(F3+(1-F3)*DU); %Uranium in crust
Thc2 = Thm/(F3+(1-F3)*DTh); %Thorium in crust
Kc2 = Km/(F3+(1-F3)*DK); %Potassium in crust

Um2 = Um-F3*Uc2/(1-F3); %Uranium in mantle lithosphere
Thm2 = Thm-F3*Thc2/(1-F3); %Thorium in mantle lithosphere
Km2 = Km-F3*Kc2/(1-F3); %Potassium in mantle lithosphere

HU238c2 = Uc2*U238*H238U*exp(t*log(2)/U238t);
HU235c2 = Uc2*U235*H235U*exp(t*log(2)/U235t);
HTh232c2 = Thc2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c2 = Kc2*K40*H40K*exp(t*log(2)/K40t);

HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
```

```
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m2 = Um2*U238*H238U*exp(t*log(2)/U238t);
HU235m2 = Um2*U235*H235U*exp(t*log(2)/U235t);
HTh232m2 = Thm2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m2 = Km2*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc2 = (HU238c2+HU235c2+HTh232c2+HK40c2)*rhoc; %volumetric heat
production crust (W/m3)
Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat
production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm2 = (HU238m2+HU235m2+HTh232m2+HK40m2)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)
A3c = Atotc2;
A2c = Atotc3;
A1c = Atotc4;
A1m = Atotm4;
A2m = Atotm3;
A3m = Atotm2;
Am = Atotm;

%Calculate the layer thicknesses
lay3c = 20*(F3/100); %km
vol3c = (4/3)*pi*(rad^3-(rad-lay3c)^3); %km^3 volume of crustal layer
mass3c = vol3c*(1000^3)*rhoc; %kg mass crustal layer

lay2c = (rad-lay3c)-((rad-lay3c)^3-vol2c/((4/3)*pi))^(1/3); %km

lay1c = (rad-lay3c-lay2c)-((rad-lay3c-lay2c)^3-vol1c/((4/3)*pi))^(1/3);
%km

lay1m = (rad-lay3c-lay2c-lay1c)-((rad-lay3c-lay2c-lay1c)^3-
vol1m/((4/3)*pi))^(1/3); %km

lay2m = (rad-lay3c-lay2c-lay1c-lay1m)-((rad-lay3c-lay2c-lay1c-lay1m)^3-
vol2m/((4/3)*pi))^(1/3); %km
```

```
mass3m = mass3c/(F3/100)-mass3c; %kg mass lithosphere layer
vol3m = mass3m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay3m = (rad-lay3c-lay2c-lay1c-lay1m-lay2m)-((rad-lay3c-lay2c-lay1c-
lay1m-lay2m)^3-vol3m/((4/3)*pi))^(1/3); %km

%Calculate base (depth) of each layer
cth2a = lay3c+lay2c;
cth2 = lay3c+lay2c+lay1c; %total thickness of crust in km
cmth2a = cth2+lay1m;
cmth2b = cmth2a+lay2m;
cmth2 = cmth2b+lay3m;
%% Calculate areotherm 2Ga
z = zeros(d,33);
z1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
z(1,1) = 0; %Initial Pressure (GPa)
z(1,2) = 0; %Initial Depth (km)
z(1,3) = Tad2+273;
z(1,4) = (-4.877*z(1,1)^2+120.2*z(1,1)+1088)+273.15; %solidus
z(1,5) = Q02/1000; %This is the surface heat flow (W/m^2)
z(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   z(i,2)=z(i-1,2)+step; %km
   if z(i,2)<cth2
       z(i,1)=(rhoc*g*z(i,2)*1000)/10^9;
   else
       z(i,1)=(rhoc*g*cth2*1000)/10^9+(rhoml*g*((z(i,2)-
cth2)*1000))/10^9;
   end

   %These are the adiabat and solidus
   z(i,3)=slope*z(i,2)+z(1,3);
   z(i,4)=(-4.877*z(i,1)^2+120.2*z(i,1)+1088)+273.15; %solidus

   %This is the q value
   if z(i,2)<lay3c
       q=z(i-1,5)-A3c*(z(i,2)-z(i-1,2))*1000;
   elseif z(i,2)<lay3c && z(i,2)<cth2a
       q=z(i-1,5)-A2c*(z(i,2)-z(i-1,2))*1000;
   elseif z(i,2)>cth2a && z(i,2)<cth2
       q=z(i-1,5)-A1c*(z(i,2)-z(i-1,2))*1000;
   elseif z(i,2)>cth2 && z(i,2)<cmth2a
       q=z(i-1,5)-A1m*(z(i,2)-z(i-1,2))*1000;
   elseif z(i,2)>cmth2a && z(i,2)<cmth2b
       q=z(i-1,5)-A2m*(z(i,2)-z(i-1,2))*1000;
   elseif z(i,2)>cmth2b && z(i,2)<cmth2
       q=z(i-1,5)-A3m*(z(i,2)-z(i-1,2))*1000;
   else
       q=z(i-1,5)-Am*(z(i,2)-z(i-1,2))*1000;
   end
```

```
   z(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if z(i,2)<cth2
       k=kc;
   else
       k=klth;
   end
   z(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this z value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if z(i,2)<lay3c
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-A3c*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   elseif z(i,2)>lay3c && z(i,2)<cth2a
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-A2c*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   elseif z(i,2)>cth2a && z(i,2)<cth2
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-A1c*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   elseif z(i,2)>cth2  && z(i,2)<cmth2a
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-A1m*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   elseif z(i,2)>cmth2a && z(i,2)<cmth2b
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-A2m*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   elseif z(i,2)>cmth2b && z(i,2)<cmth2
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-A3m*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   else
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-Am*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   end
   z(i,7)=T;

   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
       %Step 5: Calculate k using most recent temperature calculation
       if z(i,2)<cth2a
           k=kc*(1+0.0015*z(i,2))/(1+0.0015*(z(i,j-1)-273));
       elseif z(i,2)>cth2a && z(i,2)<cth2
           k=kc*(1+0.0015*z(i,2))/(1+0.0001*(z(i,j-1)-273));
       elseif z(i,2)>cth2
           if z(i,j-1)<500
               k=1/(0.0741+0.000502*z(i,j-1));
           else %Take the larger of these values
               k1=1/(0.0741+0.000502*z(i,j-1))+0.0023*(z(i,j-1)-500);
               %or
               k2=1.26*(1+z(i,2)/1000)+0.0023*(z(i,j-1)-500);
               if k1>k2
                   k=k1;
               else
                   k=k2;
               end
           end
```

```
       end
       z(i,j)=k; %Save the applicable k in the array

       z(1,j+1)=T0; %set T on first row

       %Step 6: Calculate the temperature using the most recent k
       if z(i,2)<lay3c
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
A3c*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       elseif z(i,2)>lay3c && z(i,2)<cth2a
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
A2c*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       elseif z(i,2)>cth2a && z(i,2)<cth2
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
A1c*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       elseif z(i,2)>cth2 && z(i,2)<cmth2a
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
A1m*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       elseif z(i,2)>cmth2a && z(i,2)<cmth2b
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
A2m*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       elseif z(i,2)>cmth2b && z(i,2)<cmth2
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
A3m*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       else
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
Am*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       end
       z(i,j+1)=T; %Stores the values for T in the array
   end %Ends the Step 5 and 6 loop
   z1(i,1)=z(i,1)*10000;
   z1(1,2)=z(1,33);
   z1(i,2)=z(i,33);
   A2(i,1)=z(i,33)-z(i,3);
   B2(i,1)=z(i,1);
   C2(i,1)=z(i,2);
   PFp2 = find(A2>0,1,'first');
   PFm2 = find(A2<0,1,'last');
   PF2 = (B2(PFp2)+B2(PFm2))/2; %calulate pressure
   dF2 = (C2(PFp2)+C2(PFm2))/2; %calculate crusth+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,3)=cth2;
w(2,3)=Q02;
w(3,3)=Tad2;


syms P0
eqn1 = slope2*P0+(Tad2+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P02 = min(vpasolve(eqn1,P0));
P02 = double(P02); %pressure where solidus and adiabat intersect
w(4,3)=P02;
w(5,3)=PF2;
w(6,3)=dF2;


dTdPsolidus2 = 2*(-4.877*PF2)+120.2; %slope of solidus Duncan et al.
(2018)
```

```
dTdF2 = dTdFa*PF2+dTdFb;
w(7,3)=dTdPsolidus2;
w(8,3)=dTdF2;

% Use P0 and PF to calculate F (percentage of melt produced)
F2 = ((dTdPsolidus2-slope2)/((deltaHF/C_P)+dTdF2))*(P02-PF2)*100;
w(9,3)=F2;

% Calculate the composition of the melt and error
SiO22 = (SiO2A*PF2+SiO2B)*F2^2+(SiO2C*PF2+SiO2D)*F2+(SiO2E*PF2+SiO2F);
TiO22 = TiO2A*F2^TiO2B;
Al2O32 =
(Al2O3A*PF2+Al2O3B)*F2^2+(Al2O3C*PF2+Al2O3D)*F2+(Al2O3E*PF2+Al2O3F);
Cr2O32 =
(Cr2O3A*PF2+Cr2O3B)*F2^2+(Cr2O3C*PF2+Cr2O3D)*F2+(Cr2O3E*PF2+Cr2O3F);
FeO2 = (FeOA*PF2+FeOB)*F2^2+(FeOC*PF2+FeOD)*F2+(FeOE*PF2+FeOF);
MnO2 = (MnOA*PF2+MnOB)*F2^2+(MnOC*PF2+MnOD)*F2+(MnOE*PF2+MnOF);
MgO2 = (MgOA*PF2+MgOB)*F2^2+(MgOC*PF2+MgOD)*F2+(MgOE*PF2+MgOF);
CaO2 = CaOA*F2^3+CaOB*F2^2+CaOC*F2+CaOD;
Na2O2 = (Na2OA*PF2+Na2OB)*F2^2+(Na2OC*PF2+Na2OD)*F2+(Na2OE*PF2+Na2OF);
K2O2 = K2OA*F2^K2OB;
P2O52 = P2O5A*F2^P2O5B;
w(10,3)=SiO22;
w(11,3)=TiO22;
w(12,3)=Al2O32;
w(13,3)=Cr2O32;
w(14,3)=FeO2;
w(15,3)=MnO2;
w(16,3)=MgO2;
w(17,3)=CaO2;
w(18,3)=Na2O2;
w(19,3)=K2O2;
w(20,3)=P2O52;

% Calculate the residual using batch melting-will become mantle lithos
SiO2res2 = (mantleSiO2-((F2/100)*SiO22))/(1-(F2/100));
TiO2res2 = (mantleTiO2-((F2/100)*TiO22))/(1-(F2/100));
Al2O3res2 = (mantleAl2O3-((F2/100)*Al2O32))/(1-(F2/100));
Cr2O3res2 = (mantleCr2O3-((F2/100)*Cr2O32))/(1-(F2/100));
FeOres2 = (mantleFeO-((F2/100)*FeO2))/(1-(F2/100));
MnOres2 = (mantleMnO-((F2/100)*MnO2))/(1-(F2/100));
MgOres2 = (mantleMgO-((F2/100)*MgO2))/(1-(F2/100));
CaOres2 = (mantleCaO-((F2/100)*CaO2))/(1-(F2/100));
Na2Ores2 = (mantleNa2O-((F2/100)*Na2O2))/(1-(F2/100));
K2Ores2 = (mantleK2O-((F2/100)*K2O2))/(1-(F2/100));
P2O5res2 = (mantleP2O5-((F2/100)*P2O52))/(1-(F2/100));
w(21,3)=SiO2res2;
w(22,3)=TiO2res2;
w(23,3)=Al2O3res2;
w(24,3)=Cr2O3res2;
w(25,3)=FeOres2;
w(26,3)=MnOres2;
w(27,3)=MgOres2;
w(28,3)=CaOres2;
w(29,3)=Na2Ores2;
w(30,3)=K2Ores2;
```

```
w(31,3)=P2O5res2;
%End of 2 Ga calculations


%% 1 Ga
t = 1*10^9; %time in years, 0 is present
Q01 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad1 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc1 = Um/(F2+(1-F2)*DU); %Uranium in crust
Thc1 = Thm/(F2+(1-F2)*DTh); %Thorium in crust
Kc1 = Km/(F2+(1-F2)*DK); %Potassium in crust

Um1 = Um-F2*Uc1/(1-F2); %Uranium in mantle lithosphere
Thm1 = Thm-F2*Thc1/(1-F2); %Thorium in mantle lithosphere
Km1 = Km-F2*Kc1/(1-F2); %Potassium in mantle lithosphere

HU238c1 = Uc1*U238*H238U*exp(t*log(2)/U238t);
HU235c1 = Uc1*U235*H235U*exp(t*log(2)/U235t);
HTh232c1 = Thc1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c1 = Kc1*K40*H40K*exp(t*log(2)/K40t);

HU238c2 = Uc2*U238*H238U*exp(t*log(2)/U238t);
HU235c2 = Uc2*U235*H235U*exp(t*log(2)/U235t);
HTh232c2 = Thc2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c2 = Kc2*K40*H40K*exp(t*log(2)/K40t);

HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m2 = Um2*U238*H238U*exp(t*log(2)/U238t);
HU235m2 = Um2*U235*H235U*exp(t*log(2)/U235t);
HTh232m2 = Thm2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m2 = Km2*K40*H40K*exp(t*log(2)/K40t);
```

```
HU238m1 = Um1*U238*H238U*exp(t*log(2)/U238t);
HU235m1 = Um1*U235*H235U*exp(t*log(2)/U235t);
HTh232m1 = Thm1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m1 = Km1*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc1 = (HU238c1+HU235c1+HTh232c1+HK40c1)*rhoc; %volumetric heat production crust (W/m3)
Atotc2 = (HU238c2+HU235c2+HTh232c2+HK40c2)*rhoc; %volumetric heat production crust (W/m3)
Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat production mantle litho (W/m3)
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat production mantle litho (W/m3)
Atotm2 = (HU238m2+HU235m2+HTh232m2+HK40m2)*rhoml; %volumetric heat production mantle litho (W/m3)
Atotm1 = (HU238m1+HU235m1+HTh232m1+HK40m1)*rhoml; %volumetric heat production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production mantle (W/m3)
A4c = Atotc1;
A3c = Atotc2;
A2c = Atotc3;
A1c = Atotc4;
A1m = Atotm4;
A2m = Atotm3;
A3m = Atotm2;
A4m = Atotm1;
Am = Atotm;

%Calculate the layer thicknesses
lay4c = 20*(F2/100); %km
vol4c = (4/3)*pi*(rad^3-(rad-lay4c)^3); %km^3 volume of crustal layer
mass4c = vol4c*(1000^3)*rhoc; %kg mass crustal layer

lay3c = (rad-lay4c)-((rad-lay4c)^3-vol3c/((4/3)*pi))^(1/3); %km

lay2c = (rad-lay4c-lay3c)-((rad-lay4c-lay3c)^3-vol2c/((4/3)*pi))^(1/3); %km

lay1c = (rad-lay4c-lay3c-lay2c)-((rad-lay4c-lay3c-lay2c)^3-vol1c/((4/3)*pi))^(1/3); %km

lay1m = (rad-lay4c-lay3c-lay2c-lay1c)-((rad-lay4c-lay3c-lay2c-lay1c)^3-vol1m/((4/3)*pi))^(1/3); %km
```

```
lay2m = (rad-lay4c-lay3c-lay2c-lay1c-lay1m)-((rad-lay4c-lay3c-lay2c-
lay1c-lay1m)^3-vol2m/((4/3)*pi))^(1/3); %km

lay3m = (rad-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m)-((rad-lay4c-lay3c-
lay2c-lay1c-lay1m-lay2m)^3-vol3m/((4/3)*pi))^(1/3); %km

mass4m = mass4c/(F2/100)-mass3c; %kg mass lithosphere layer
vol4m = mass4m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay4m = (rad-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)-((rad-lay4c-
lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)^3-vol4m/((4/3)*pi))^(1/3); %km

%Calculate base (depth) of each layer
cth1a = lay4c+lay3c;
cth1b = lay4c+lay3c+lay2c;
cth1 = lay4c+lay3c+lay2c+lay1c; %total thickness of crust in km
cmth1a = cth1+lay1m;
cmth1b = cmth1a+lay2m;
cmth1c = cmth1b+lay3m;
cmth1 = cmth1c+lay4m;
%% Calculate areotherm 1 Ga
b = zeros(d,33);
b1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
b(1,1) = 0; %Initial Pressure (GPa)
b(1,2) = 0; %Initial Depth (km)
b(1,3) = Tad1+273;
b(1,4) = (-4.877*b(1,1)^2+120.2*b(1,1)+1088)+273.15; %solidus
b(1,5) = Q01/1000; %This is the surface heat flow (W/m^2)
b(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   b(i,2)=b(i-1,2)+step; %km
   if b(i,2)<cth1
       b(i,1)=(rhoc*g*b(i,2)*1000)/10^9;
   else
       b(i,1)=(rhoc*g*cth1*1000)/10^9+(rhoml*g*((b(i,2)-
cth1)*1000))/10^9;
   end

   %These are the adiabat and solidus
   b(i,3)=slope*b(i,2)+b(1,3);
   b(i,4)=(-4.877*b(i,1)^2+120.2*b(i,1)+1088)+273.15; %solidus

   %This is the q value
   if b(i,2)<lay4c
       q=b(i-1,5)-A4c*(b(i,2)-b(i-1,2))*1000;
   elseif b(i,2)<lay4c && b(i,2)<cth1a
       q=b(i-1,5)-A3c*(b(i,2)-b(i-1,2))*1000;
   elseif b(i,2)<cth1a && b(i,2)<cth1b
       q=b(i-1,5)-A2c*(b(i,2)-b(i-1,2))*1000;
   elseif b(i,2)>cth1b && b(i,2)<cth1
```

```
       q=b(i-1,5)-A1c*(b(i,2)-b(i-1,2))*1000;
   elseif b(i,2)>cth1 && b(i,2)<cmth1a
       q=b(i-1,5)-A1m*(b(i,2)-b(i-1,2))*1000;
   elseif b(i,2)>cmth1a && b(i,2)<cmth1b
       q=b(i-1,5)-A2m*(b(i,2)-b(i-1,2))*1000;
   elseif b(i,2)>cmth1b && b(i,2)<cmth1c
       q=b(i-1,5)-A3m*(b(i,2)-b(i-1,2))*1000;
   elseif b(i,2)>cmth1c && b(i,2)<cmth1
       q=b(i-1,5)-A4m*(b(i,2)-b(i-1,2))*1000;
   else
       q=b(i-1,5)-Am*(b(i,2)-b(i-1,2))*1000;
   end
   b(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if b(i,2)<cth1
       k=kc;
   else
       k=klth;
   end
   b(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this b value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if b(i,2)<lay4c
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A4c*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   elseif b(i,2)>lay4c && b(i,2)<cth1a
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A3c*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   elseif b(i,2)>cth1a && b(i,2)<cth1b
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A2c*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   elseif b(i,2)>cth1b && b(i,2)<cth1
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A1c*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   elseif b(i,2)>cth1 && b(i,2)<cmth1a
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A1m*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   elseif b(i,2)>cmth1a && b(i,2)<cmth1b
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A2m*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   elseif b(i,2)>cmth1b && b(i,2)<cmth1c
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A3m*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   elseif b(i,2)>cmth1c && b(i,2)<cmth1
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A4m*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   else
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-Am*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   end
   b(i,7)=T;

   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
```

```
        %Step 5: Calculate k using most recent temperature calculation
        if b(i,2)<cth1b
            k=kc*(1+0.0015*b(i,2))/(1+0.0015*(b(i,j-1)-273));
        elseif b(i,2)>cth1b && b(i,2)<cth1
            k=kc*(1+0.0015*b(i,2))/(1+0.0001*(b(i,j-1)-273));
        elseif b(i,2)>cth1
            if b(i,j-1)<500
                k=1/(0.0741+0.000502*b(i,j-1));
            else %Take the larger of these values
                k1=1/(0.0741+0.000502*b(i,j-1))+0.0023*(b(i,j-1)-500);
                %or
                k2=1.26*(1+b(i,2)/1000)+0.0023*(b(i,j-1)-500);
                if k1>k2
                    k=k1;
                else
                    k=k2;
                end
            end
        end
        b(i,j)=k; %Save the applicable k in the array

        b(1,j+1)=T0; %set T on first row

        %Step 6: Calculate the temperature using the most recent k
        if b(i,2)<lay4c
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A4c*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        elseif b(i,2)>lay4c && b(i,2)<cth1a
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A3c*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        elseif b(i,2)>cth1a && b(i,2)<cth1b
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A2c*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        elseif b(i,2)>cth1b && b(i,2)<cth1
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A1c*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        elseif b(i,2)>cth1 && b(i,2)<cmth1a
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A1m*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        elseif b(i,2)>cmth1a && b(i,2)<cmth1b
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A2m*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        elseif b(i,2)>cmth1b && b(i,2)<cmth1c
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A3m*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        elseif b(i,2)>cmth1c && b(i,2)<cmth1
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A4m*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        else
            T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
Am*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
        end
        b(i,j+1)=T; %Stores the values for T in the array
    end %Ends the Step 5 and 6 loop
    b1(i,1)=b(i,1)*10000;
    b1(1,2)=b(1,33);
```

```
   b1(i,2)=b(i,33);
   A1(i,1)=b(i,33)-b(i,3);
   B1(i,1)=b(i,1);
   C1(i,1)=b(i,2);
   PFp1 = find(A1>0,1,'first');
   PFm1 = find(A1<0,1,'last');
   PF1 = (B1(PFp1)+B1(PFm1))/2; %calulate pressure
   dF1 = (C1(PFp1)+C1(PFm1))/2; %calculate crusth+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,4)=cth1;
w(2,4)=Q01;
w(3,4)=Tad1;

syms P0
eqn1 = slope2*P0+(Tad1+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P01 = min(vpasolve(eqn1,P0));
P01 = double(P01); %pressure where solidus and adiabat intersect
w(4,4)=P01;
w(5,4)=PF1;
w(6,4)=dF1;

dTdPsolidus1 = 2*(-4.877*PF1)+120.2; %slope of solidus Duncan et al.
(2018)
dTdF1 = dTdFa*PF1+dTdFb;
w(7,4)=dTdPsolidus1;
w(8,4)=dTdF1;

% Use P0 and PF to calculate F (percentage of melt produced)
F1 = ((dTdPsolidus1-slope2)/((deltaHF/C_P)+dTdF1))*(P01-PF1)*100;
w(9,4)=F1;

% Calculate the composition of the melt and error
SiO21 = (SiO2A*PF1+SiO2B)*F1^2+(SiO2C*PF1+SiO2D)*F1+(SiO2E*PF1+SiO2F);
TiO21 = TiO2A*F1^TiO2B;
Al2O31 =
(Al2O3A*PF1+Al2O3B)*F1^2+(Al2O3C*PF1+Al2O3D)*F1+(Al2O3E*PF1+Al2O3F);
Cr2O31 =
(Cr2O3A*PF1+Cr2O3B)*F1^2+(Cr2O3C*PF1+Cr2O3D)*F1+(Cr2O3E*PF1+Cr2O3F);
FeO1 = (FeOA*PF1+FeOB)*F1^2+(FeOC*PF1+FeOD)*F1+(FeOE*PF1+FeOF);
MnO1 = (MnOA*PF1+MnOB)*F1^2+(MnOC*PF1+MnOD)*F1+(MnOE*PF1+MnOF);
MgO1 = (MgOA*PF1+MgOB)*F1^2+(MgOC*PF1+MgOD)*F1+(MgOE*PF1+MgOF);
CaO1 = CaOA*F1^3+CaOB*F1^2+CaOC*F1+CaOD;
Na2O1 = (Na2OA*PF1+Na2OB)*F1^2+(Na2OC*PF1+Na2OD)*F1+(Na2OE*PF1+Na2OF);
K2O1 = K2OA*F1^K2OB;
P2O51 = P2O5A*F1^P2O5B;
w(10,4)=SiO21;
w(11,4)=TiO21;
w(12,4)=Al2O31;
w(13,4)=Cr2O31;
w(14,4)=FeO1;
w(15,4)=MnO1;
w(16,4)=MgO1;
w(17,4)=CaO1;
w(18,4)=Na2O1;
w(19,4)=K2O1;
```

```
w(20,4)=P2O51;

% Calculate the residual using batch melting-will become mantle lithos
SiO2res1 = (mantleSiO2-((F1/100)*SiO21))/(1-(F1/100));
TiO2res1 = (mantleTiO2-((F1/100)*TiO21))/(1-(F1/100));
Al2O3res1 = (mantleAl2O3-((F1/100)*Al2O31))/(1-(F1/100));
Cr2O3res1 = (mantleCr2O3-((F1/100)*Cr2O31))/(1-(F1/100));
FeOres1 = (mantleFeO-((F1/100)*FeO1))/(1-(F1/100));
MnOres1 = (mantleMnO-((F1/100)*MnO1))/(1-(F1/100));
MgOres1 = (mantleMgO-((F1/100)*MgO1))/(1-(F1/100));
CaOres1 = (mantleCaO-((F1/100)*CaO1))/(1-(F1/100));
Na2Ores1 = (mantleNa2O-((F1/100)*Na2O1))/(1-(F1/100));
K2Ores1 = (mantleK2O-((F1/100)*K2O1))/(1-(F1/100));
P2O5res1 = (mantleP2O5-((F1/100)*P2O51))/(1-(F1/100));
w(21,4)=SiO2res1;
w(22,4)=TiO2res1;
w(23,4)=Al2O3res1;
w(24,4)=Cr2O3res1;
w(25,4)=FeOres1;
w(26,4)=MnOres1;
w(27,4)=MgOres1;
w(28,4)=CaOres1;
w(29,4)=Na2Ores1;
w(30,4)=K2Ores1;
w(31,4)=P2O5res1;
%End of 1 Ga calculations


%% Present day
t = 0*10^9; %time in years, 0 is present
Q00 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad0 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc0 = Um/(F1+(1-F1)*DU); %Uranium in crust
Thc0 = Thm/(F1+(1-F1)*DTh); %Thorium in crust
Kc0 = Km/(F1+(1-F1)*DK); %Potassium in crust

Um0 = Um-F1*Uc0/(1-F1); %Uranium in mantle lithosphere
Thm0 = Thm-F1*Thc0/(1-F1); %Thorium in mantle lithosphere
Km0 = Km-F1*Kc0/(1-F1); %Potassium in mantle lithosphere

HU238c0 = Uc0*U238*H238U*exp(t*log(2)/U238t);
HU235c0 = Uc0*U235*H235U*exp(t*log(2)/U235t);
HTh232c0 = Thc0*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c0 = Kc0*K40*H40K*exp(t*log(2)/K40t);

HU238c1 = Uc1*U238*H238U*exp(t*log(2)/U238t);
HU235c1 = Uc1*U235*H235U*exp(t*log(2)/U235t);
HTh232c1 = Thc1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c1 = Kc1*K40*H40K*exp(t*log(2)/K40t);

HU238c2 = Uc2*U238*H238U*exp(t*log(2)/U238t);
HU235c2 = Uc2*U235*H235U*exp(t*log(2)/U235t);
```

```
HTh232c2 = Thc2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c2 = Kc2*K40*H40K*exp(t*log(2)/K40t);

HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m2 = Um2*U238*H238U*exp(t*log(2)/U238t);
HU235m2 = Um2*U235*H235U*exp(t*log(2)/U235t);
HTh232m2 = Thm2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m2 = Km2*K40*H40K*exp(t*log(2)/K40t);

HU238m1 = Um1*U238*H238U*exp(t*log(2)/U238t);
HU235m1 = Um1*U235*H235U*exp(t*log(2)/U235t);
HTh232m1 = Thm1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m1 = Km1*K40*H40K*exp(t*log(2)/K40t);

HU238m0 = Um0*U238*H238U*exp(t*log(2)/U238t);
HU235m0 = Um0*U235*H235U*exp(t*log(2)/U235t);
HTh232m0 = Thm0*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m0 = Km0*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc0 = (HU238c0+HU235c0+HTh232c0+HK40c0)*rhoc; %volumetric heat
production crust (W/m3)
Atotc1 = (HU238c1+HU235c1+HTh232c1+HK40c1)*rhoc; %volumetric heat
production crust (W/m3)
Atotc2 = (HU238c2+HU235c2+HTh232c2+HK40c2)*rhoc; %volumetric heat
production crust (W/m3)
Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat
production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
```

```
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat production mantle litho (W/m3)
Atotm2 = (HU238m2+HU235m2+HTh232m2+HK40m2)*rhoml; %volumetric heat production mantle litho (W/m3)
Atotm1 = (HU238m1+HU235m1+HTh232m1+HK40m1)*rhoml; %volumetric heat production mantle litho (W/m3)
Atotm0 = (HU238m0+HU235m0+HTh232m1+HK40m0)*rhoml; %volumetric heat production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production mantle (W/m3)
A5c = Atotc0;
A4c = Atotc1;
A3c = Atotc2;
A2c = Atotc3;
A1c = Atotc4;
A1m = Atotm4;
A2m = Atotm3;
A3m = Atotm2;
A4m = Atotm1;
A5m = Atotm0;
Am = Atotm;

%Calculate the layer thicknesses
lay5c = 20*(F1/100); %km
vol5c = (4/3)*pi*(rad^3-(rad-lay5c)^3); %km^3 volume of crustal layer
mass5c = vol5c*(1000^3)*rhoc; %kg mass crustal layer

lay4c = (rad-lay5c)-((rad-lay5c)^3-vol4c/((4/3)*pi))^(1/3); %km

lay3c = (rad-lay5c-lay4c)-((rad-lay5c-lay4c)^3-vol3c/((4/3)*pi))^(1/3); %km

lay2c = (rad-lay5c-lay4c-lay3c)-((rad-lay5c-lay4c-lay3c)^3-vol2c/((4/3)*pi))^(1/3); %km

lay1c = (rad-lay5c-lay4c-lay3c-lay2c)-((rad-lay5c-lay4c-lay3c-lay2c)^3-vol1c/((4/3)*pi))^(1/3); %km

lay1m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c)-((rad-lay5c-lay4c-lay3c-lay2c-lay1c)^3-vol1m/((4/3)*pi))^(1/3); %km

lay2m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m)-((rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m)^3-vol2m/((4/3)*pi))^(1/3); %km

lay3m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m)-((rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m)^3-vol3m/((4/3)*pi))^(1/3); %km

lay4m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)-((rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)^3-vol4m/((4/3)*pi))^(1/3); %km

mass5m = mass5c/(F4/100)-mass5c; %kg mass lithosphere layer
vol5m = mass5m/((1000^3)*rhoml); %km^3 volume lithosphere layer
```

```
lay5m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m-lay4m)-
((rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m-lay4m)^3-
vol5m/((4/3)*pi))^(1/3); %km

%Calculate base (depth) of each layer
cth0a = lay5c+lay4c;
cth0b = lay5c+lay4c+lay3c;
cth0c = lay5c+lay4c+lay3c+lay2c;
cth0 = lay5c+lay4c+lay3c+lay2c+lay1c; %total thickness of crust in km
cmth0a = cth0+lay1m;
cmth0b = cmth0a+lay2m;
cmth0c = cmth0b+lay3m;
cmth0d = cmth0c+lay4m;
cmth0 = cmth0d+lay5m;
%% Calculate areotherm 0 Ga
c = zeros(d,33);
c1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
c(1,1) = 0; %Initial Pressure (GPa)
c(1,2) = 0; %Initial Depth (km)
c(1,3) = Tad0+273;
c(1,4) = (-4.877*c(1,1)^2+120.2*c(1,1)+1088)+273.15; %solidus
c(1,5) = Q00/1000; %This is the surface heat flow (W/m^2)
c(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   c(i,2)=c(i-1,2)+step; %km
   if c(i,2)<cth0
       c(i,1)=(rhoc*g*c(i,2)*1000)/10^9;
   else
       c(i,1)=(rhoc*g*cth0*1000)/10^9+(rhoml*g*((c(i,2)-
cth0)*1000))/10^9;
   end

   %These are the adiabat and solidus
   c(i,3)=slope*c(i,2)+c(1,3);
   c(i,4)=(-4.877*c(i,1)^2+120.2*c(i,1)+1088)+273.15; %solidus

   %This is the q value
   if c(i,2)<lay5c
       q=c(i-1,5)-A5c*(c(i,2)-c(i-1,2))*1000;
   elseif c(i,2)<lay5c && c(i,2)<cth0a
       q=c(i-1,5)-A4c*(c(i,2)-c(i-1,2))*1000;
   elseif c(i,2)<cth0a && c(i,2)<cth0b
       q=c(i-1,5)-A3c*(c(i,2)-c(i-1,2))*1000;
   elseif c(i,2)<cth0b && c(i,2)<cth0c
       q=c(i-1,5)-A2c*(c(i,2)-c(i-1,2))*1000;
   elseif c(i,2)>cth0c && c(i,2)<cth0
       q=c(i-1,5)-A1c*(c(i,2)-c(i-1,2))*1000;
   elseif c(i,2)>cth0 && c(i,2)<cmth0a
       q=c(i-1,5)-A1m*(c(i,2)-c(i-1,2))*1000;
   elseif c(i,2)>cmth0a && c(i,2)<cmth0b
```

```
       q=c(i-1,5)-A2m*(c(i,2)-c(i-1,2))*1000;
   elseif c(i,2)>cmth0b && c(i,2)<cmth0c
       q=c(i-1,5)-A3m*(c(i,2)-c(i-1,2))*1000;
   elseif c(i,2)>cmth0c && c(i,2)<cmth0d
       q=c(i-1,5)-A4m*(c(i,2)-c(i-1,2))*1000;
   elseif c(i,2)>cmth0d && c(i,2)<cmth0
       q=c(i-1,5)-A5m*(c(i,2)-c(i-1,2))*1000;
   else
       q=c(i-1,5)-Am*(c(i,2)-c(i-1,2))*1000;
   end
   c(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if c(i,2)<cth0
       k=kc;
   else
       k=klth;
   end
   c(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this z value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if c(i,2)<lay5c
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A5c*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   elseif c(i,2)>lay5c && c(i,2)<cth0a
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A4c*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   elseif c(i,2)>cth0a && c(i,2)<cth0b
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A3c*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   elseif c(i,2)>cth0b && c(i,2)<cth0c
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A2c*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   elseif c(i,2)>cth0c && c(i,2)<cth0
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A1c*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   elseif c(i,2)>cth0 && c(i,2)<cmth0a
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A1m*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   elseif c(i,2)>cmth0a && c(i,2)<cmth0b
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A2m*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   elseif c(i,2)>cmth0b && c(i,2)<cmth0c
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A3m*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   elseif c(i,2)>cmth0c && c(i,2)<cmth0d
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A4m*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   elseif c(i,2)>cmth0d && c(i,2)<cmth0
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A5m*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   else
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-Am*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   end
```

```
   c(i,7)=T;

   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
       %Step 5: Calculate k using most recent temperature calculation
       if c(i,2)<cth0c
           k=kc*(1+0.0015*c(i,2))/(1+0.0015*(c(i,j-1)-273));
       elseif c(i,2)>cth0c && c(i,2)<cth0
           k=kc*(1+0.0015*c(i,2))/(1+0.0001*(c(i,j-1)-273));
       elseif c(i,2)>cth0
           if c(i,j-1)<500
               k=1/(0.0741+0.000502*c(i,j-1));
           else %Take the larger of these values
               k1=1/(0.0741+0.000502*c(i,j-1))+0.0023*(c(i,j-1)-500);
               %or
               k2=1.26*(1+c(i,2)/1000)+0.0023*(c(i,j-1)-500);
               if k1>k2
                   k=k1;
               else
                   k=k2;
               end
           end
       end
       c(i,j)=k; %Save the applicable k in the array

       c(1,j+1)=T0; %set T on first row

       %Step 6: Calculate the temperature using the most recent k
       if c(i,2)<lay5c
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A5c*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       elseif c(i,2)>lay5c && c(i,2)<cth0a
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A4c*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       elseif c(i,2)>cth0a && c(i,2)<cth0b
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A3c*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       elseif c(i,2)>cth0b && c(i,2)<cth0c
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A2c*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       elseif c(i,2)>cth0c && c(i,2)<cth0
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A1c*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       elseif c(i,2)>cth0 && c(i,2)<cmth0a
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A1m*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       elseif c(i,2)>cmth0a && c(i,2)<cmth0b
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A2m*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       elseif c(i,2)>cmth0b && c(i,2)<cmth0c
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A3m*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       elseif c(i,2)>cmth0c && c(i,2)<cmth0d
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A4m*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       elseif c(i,2)>cmth0d && c(i,2)<cmth0
```

```
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A5m*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       else
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
Am*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       end
       c(i,j+1)=T; %Stores the values for T in the array
   end %Ends the Step 5 and 6 loop
   c1(i,1)=c(i,1)*10000;
   c1(1,2)=c(1,33);
   c1(i,2)=c(i,33);
   A0(i,1)=c(i,33)-c(i,3);
   B0(i,1)=c(i,1);
   C0(i,1)=c(i,2);
   PFp0 = find(A0>0,1,'first');
   PFm0 = find(A0<0,1,'last');
   PF0 = (B0(PFp0)+B0(PFm0))/2; %calulate pressure
   dF0 = (C0(PFp0)+C0(PFm0))/2; %calculate crusth+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,5)=cth0;
w(2,5)=Q00;
w(3,5)=Tad0;

syms P0
eqn1 = slope2*P0+(Tad0+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P00 = min(vpasolve(eqn1,P0));
P00 = double(P00); %pressure where solidus and adiabat intersect
w(4,5)=P00;
w(5,5)=PF0;
w(6,5)=dF0;

dTdPsolidus0 = 2*(-4.877*PF0)+120.2; %slope of solidus Duncan et al.
(2018)
dTdF0 = dTdFa*PF0+dTdFb;
w(7,5)=dTdPsolidus0;
w(8,5)=dTdF0;

% Use P0 and PF to calculate F (percentage of melt produced)
F0 = ((dTdPsolidus0-slope2)/((deltaHF/C_P)+dTdF0))*(P00-PF0)*100;
w(9,5)=F0;

% Calculate the composition of the melt and error
SiO20 = (SiO2A*PF0+SiO2B)*F0^2+(SiO2C*PF0+SiO2D)*F0+(SiO2E*PF0+SiO2F);
TiO20 = TiO2A*F0^TiO2B;
Al2O30 =
(Al2O3A*PF0+Al2O3B)*F0^2+(Al2O3C*PF0+Al2O3D)*F0+(Al2O3E*PF0+Al2O3F);
Cr2O30 =
(Cr2O3A*PF0+Cr2O3B)*F0^2+(Cr2O3C*PF0+Cr2O3D)*F0+(Cr2O3E*PF0+Cr2O3F);
FeO0 = (FeOA*PF0+FeOB)*F0^2+(FeOC*PF0+FeOD)*F0+(FeOE*PF0+FeOF);
MnO0 = (MnOA*PF0+MnOB)*F0^2+(MnOC*PF0+MnOD)*F0+(MnOE*PF0+MnOF);
MgO0 = (MgOA*PF0+MgOB)*F0^2+(MgOC*PF0+MgOD)*F0+(MgOE*PF0+MgOF);
CaO0 = CaOA*F0^3+CaOB*F0^2+CaOC*F0+CaOD;
Na2O0 = (Na2OA*PF0+Na2OB)*F0^2+(Na2OC*PF0+Na2OD)*F0+(Na2OE*PF0+Na2OF);
K2O0 = K2OA*F0^K2OB;
P2O50 = P2O5A*F0^P2O5B;
```

```
w(10,5)=SiO20;
w(11,5)=TiO20;
w(12,5)=Al2O30;
w(13,5)=Cr2O30;
w(14,5)=FeO0;
w(15,5)=MnO0;
w(16,5)=MgO0;
w(17,5)=CaO0;
w(18,5)=Na2O0;
w(19,5)=K2O0;
w(20,5)=P2O50;

% Calculate the residual using batch melting-will become mantle lithos
SiO2res0 = (mantleSiO2-((F0/100)*SiO20))/(1-(F0/100));
TiO2res0 = (mantleTiO2-((F0/100)*TiO20))/(1-(F0/100));
Al2O3res0 = (mantleAl2O3-((F0/100)*Al2O30))/(1-(F0/100));
Cr2O3res0 = (mantleCr2O3-((F0/100)*Cr2O30))/(1-(F0/100));
FeOres0 = (mantleFeO-((F0/100)*FeO0))/(1-(F0/100));
MnOres0 = (mantleMnO-((F0/100)*MnO0))/(1-(F0/100));
MgOres0 = (mantleMgO-((F0/100)*MgO0))/(1-(F0/100));
CaOres0 = (mantleCaO-((F0/100)*CaO0))/(1-(F0/100));
Na2Ores0 = (mantleNa2O-((F0/100)*Na2O0))/(1-(F0/100));
K2Ores0 = (mantleK2O-((F0/100)*K2O0))/(1-(F0/100));
P2O5res0 = (mantleP2O5-((F0/100)*P2O50))/(1-(F0/100));
w(21,5)=SiO2res0;
w(22,5)=TiO2res0;
w(23,5)=Al2O3res0;
w(24,5)=Cr2O3res0;
w(25,5)=FeOres0;
w(26,5)=MnOres0;
w(27,5)=MgOres0;
w(28,5)=CaOres0;
w(29,5)=Na2Ores0;
w(30,5)=K2Ores0;
w(31,5)=P2O5res0;
%End of present day calculations


%% Now plot the results.
%The areotherms and adiabats are plotted against depth on the left
axis, while the solidus is plotted against pressure on the right axis
vt1 = 1/255*[232,119,34];
str1 = '#001932'; %'#000032';
color1 = sscanf(str1(2:end),'%2x%2x%2x',[1 3])/255;
str2 = '#00366C'; %'#000096';
color2 = sscanf(str2(2:end),'%2x%2x%2x',[1 3])/255;
str3 = '#0068D0'; %'#0000FA';
color3 = sscanf(str3(2:end),'%2x%2x%2x',[1 3])/255;
str4 = '#359AFF'; %'#5F5FFF';
color4 = sscanf(str4(2:end),'%2x%2x%2x',[1 3])/255;
str5 = '#99CCFF';
color5 = sscanf(str5(2:end),'%2x%2x%2x',[1 3])/255;

yyaxis left
plot(x(:,33),x(:,2),'-','color',color1,'LineWidth',2,'DisplayName','4
Ga') %areotherm
```

```
hold on
plot (y(:,33),y(:,2),'-.','color',color2,'LineWidth',2,'DisplayName','3
Ga')
hold on
plot (z(:,33),z(:,2),'--','color',color3,'LineWidth',2,'DisplayName','2
Ga')
hold on
plot (b(:,33),b(:,2),':','color',color4,'LineWidth',2,'DisplayName','1
Ga')
hold on
plot (c(:,33),c(:,2),'-','color',color5,'LineWidth',2,'DisplayName','0
Ga')
hold on
plot (x(:,3),x(:,2),'-
','color',color1,'LineWidth',1.5,'DisplayName','adiabat') %adiabat
hold on
plot (y(:,3),y(:,2),'-
.','color',color2,'LineWidth',1.5,'DisplayName','adiabat')
hold on
plot (z(:,3),z(:,2),'--
','color',color3,'LineWidth',1.5,'DisplayName','adiabat')
hold on
plot
(b(:,3),b(:,2),':','color',color4,'LineWidth',1.5,'DisplayName','adiaba
t')
hold on
plot (c(:,3),c(:,2),'-
','color',color5,'LineWidth',1.5,'DisplayName','adiabat')
legend('location','southwest');

ylabel('Depth (km)')
ylim([0 x(3000,2)])
set(gca,'YDir','reverse','YMinorTick','on','Ycolor','k')

title('Areotherms')
xlabel('Temperature (K)')
set(gca,'XAxisLocation','top','linestyleorder',{'-
'},'XMinorTick','on','Xcolor','k')
xlim([200 2000])

yyaxis right
plot(x(:,4),x(:,1),'color',vt1,'LineWidth',2,'DisplayName','solidus')
%solidus
ylim([0 x(3000,1)])
ylabel('Pressure (GPa)')
set(gca,'YDir','reverse','YMinorTick','on','Ycolor','k')

set(gca,'TickDir','both')
%% Save the results
save Areotherm4GaMelt.txt x -ascii
save Areotherm3GaMelt.txt y -ascii
save Areotherm2GaMelt.txt z -ascii
save Areotherm1GaMelt.txt b -ascii
save Areotherm0GaMelt.txt c -ascii
save pt0Melt.txt c1 -ascii
save pt1Melt.txt b1 -ascii
```

```
save pt2Melt.txt z1 -ascii
save pt3Melt.txt y1 -ascii
save pt4Melt.txt x1 -ascii
```

B.3 Melt-Average Model

```
%----------------------------------------------------------------------
% Project: Areotherm_MeltAverage.m
% Author: Fiona McGroarty
% Date Written: 19 October 2019
% Last Updated: 19 May 2021
% Purpose: This program will calculate the HPE concentration in the
depleted mantle lithosphere and then use those values to calculate the
parameters for an areotherm through time and plot the result. It will
calculate the composition of melt produced at a particular time and the
composition of the residual, with error. It will then repeat the process
using the previous calculation as the starting value for the next timestep
and the previous melt as the most recent layer in the crust.
% Additional files needed: MeltModel.xlsx
%----------------------------------------------------------------------

clear
clc

%% Load the initial values from the Excel file meltmodel.xlsx
% Load values needed later to calculate F (dT/dPsolidus, dT/dPadiabat,
deltaHF, C_P. dT/dF). P0 and Pf will be calculated in a later step
prior to calculating F

% 1. Parameter read in
param = importdata('OxideCoeff.txt');

% 2. Variable read in
SiO2A = param(1,1);
SiO2B = param(1,2);
SiO2C = param(1,3);
SiO2D = param(1,4);
SiO2E = param(1,5);
SiO2F = param(1,6);

TiO2A = param(2,1);
TiO2B = param(2,2);
TiO2C = param(2,3);
TiO2D = param(2,4);
TiO2E = param(2,5);
TiO2F = param(2,6);

Al2O3A = param(3,1);
Al2O3B = param(3,2);
Al2O3C = param(3,3);
Al2O3D = param(3,4);
Al2O3E = param(3,5);
Al2O3F = param(3,6);

Cr2O3A = param(4,1);
Cr2O3B = param(4,2);
Cr2O3C = param(4,3);
Cr2O3D = param(4,4);
```

```
Cr2O3E = param(4,5);
Cr2O3F = param(4,6);

FeOA = param(5,1);
FeOB = param(5,2);
FeOC = param(5,3);
FeOD = param(5,4);
FeOE = param(5,5);
FeOF = param(5,6);

MnOA = param(6,1);
MnOB = param(6,2);
MnOC = param(6,3);
MnOD = param(6,4);
MnOE = param(6,5);
MnOF = param(6,6);

MgOA = param(7,1);
MgOB = param(7,2);
MgOC = param(7,3);
MgOD = param(7,4);
MgOE = param(7,5);
MgOF = param(7,6);

CaOA = param(8,1);
CaOB = param(8,2);
CaOC = param(8,3);
CaOD = param(8,4);
CaOE = param(8,5);
CaOF = param(8,6);

Na2OA = param(9,1);
Na2OB = param(9,2);
Na2OC = param(9,3);
Na2OD = param(9,4);
Na2OE = param(9,5);
Na2OF = param(9,6);

K2OA = param(10,1);
K2OB = param(10,2);
K2OC = param(10,3);
K2OD = param(10,4);
K2OE = param(10,5);
K2OF = param(10,6);

P2O5A = param(11,1);
P2O5B = param(11,2);
P2O5C = param(11,3);
P2O5D = param(11,4);
P2O5E = param(11,5);
P2O5F = param(11,6);

%% Mars parameters
rhoc = 2900; %kg/m^3 crustal density
rhoml = 3400; %kg/m^3 mantle lithosphere density
```

```
rhom = 3700; %kg/m^3 mantle density
T0 = 220; %Temperature at the surface in K
g = 3.72; %m/s^2 gravitational acceleration on Mars
slope = 0.18; %adiabat slope (deg/km)
slope2 = (slope/1000)/(g*rhoml)*1E9; %adiabat slope
deltaHF = 640000; %J/kg (Kiefer, 2003)
C_P = 1200; %J/K*kg (Kiefer, 2003)
rad = 3390; %km radius planet
d = 5000; %The number of rows of the array to hold the values
step = 0.1; %step for depth km

dTdFa = -0.639; %calculated from experimental data
dTdFb = 5.2926; %calculated from experimental data

%Input the oxides in the undepleted mantle (Dreibus and Wanke 1985)
mantleSiO2 = 44.4;
mantleTiO2 = 0.14;
mantleAl2O3 = 3.02;
mantleFeO = 17.9;
mantleCr2O3 = 0.76;
mantleMnO = 0.47;
mantleMgO = 30.2;
mantleCaO = 2.45;
mantleNa2O = 0.50;
mantleK2O = 0.04;
mantleP2O5 = 0.16;

%Initial thermal conductivity guess
kc = 2.5; %crust thermal condutivity(W/mK)
klth = 4.0; %mantle lithosphere thermal conductivity(W/mK)

% Heat Production Constants
U238 = 0.992834; %fraction of U238 in the solar system
U235 = 0.00711; %fraction of U235
U238t = 4.469E9; %half-life in yr
U235t = 0.704E9; %half-life in yr
H238U = 9.464E-5; %rate of heat release 238U (W/kg)
H235U = 5.687E-4; %rate of heat release 235U (W/kg)

Th232 = 0.9998; %fraction of Th232
Th232t = 14.00E9; %half-life in yr
H232Th = 2.638E-5; %rate of heat release 232Th (W/kg)

K40 = 1.193E-4; %fraction of K40
K40t = 1.268E9; %half-life in yr
H40K = 2.97E-5; %rate of heat release 40K (W/kg)

%Bulk peridotite/melt partition coefficients
DK = 0.003; %Davis et al. 2011 EPSL, garnet peridodite
DU = 1.1e-4; %Beatie 1993 EPSL, spinel peridodite
DTh = 1.7e-4; %Beatie 1993 EPSL, spinel peridodite

%HPE concentrations
Uc = 180/1E9; %Uranium in crust (Taylor and McLennon, 2009)
Thc = 700/1E9; %Thorium in crust (Taylor and McLennon, 2009)
```

```
Kc = 3740/1E6; %Potassium in crust (Taylor and McLennon, 2009)

Um = 16/1E9; %Uranium in mantle Dreibus and Wanke
Thm = 56/1E9; %Thorium in mantle Dreibus and Wanke
Km = 315/1E6; %Potassium in mantle Dreibus and Wanke


%% Areotherm 4 Ga
% Input the initial parameters TO BE CHANGED WITH TIME
t = 4*10^9; %time in years, 0 is present
Q04 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad4 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Forig = 0.1; %initial guess for melt fraction for first crust
Uc4 = Um/(Forig+(1-Forig)*DU); %Uranium in crust
Thc4 = Thm/(Forig+(1-Forig)*DTh); %Thorium in crust
Kc4 = Km/(Forig+(1-Forig)*DK); %Potassium in crust
Um4 = (Um-Forig*Uc4)/(1-Forig); %Solve for mantle lithosphere
concentration
Thm4 = (Thm-Forig*Thc4)/(1-Forig); %Solve for mantle lithosphere
concentration
Km4 = (Km-Forig*Kc4)/(1-Forig); %Solve for mantle lithosphere
concentration
% Um4 = Um*0.5; %Uranium in mantle lithosphere
% Thm4 = Thm*0.5; %Thorium in mantle lithosphere
% Km4 = Km*0.5; %Potassium in mantle lithosphere

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)
A4c = Atotc4;
A4m = Atotm4;
Am = Atotm;

%Calculate the layer thicknesses
```

```
lay1c = 20;
vol1c = (4/3)*pi*(rad^3-(rad-lay1c)^3); %km^3 volume of crustal layer
mass1c = vol1c*(1000^3)*rhoc; %kg mass crustal layer

mass1m = mass1c/Forig-mass1c; %kg mass lithosphere layer
vol1m = mass1m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay1m = (rad-lay1c)-((rad-lay1c)^3-vol1m/((4/3)*pi))^(1/3); %km
thicknes of lithosphere layer

cth4 = lay1c; %total thickness of crust in km
cmth4 = cth4+lay1m;
%% Calculate areotherm at 4 Ga
x = zeros(d,33);
x1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
x(1,1) = 0; %Initial Pressure (GPa)
x(1,2) = 0; %Initial Depth (km)
x(1,3) = Tad4+273;
x(1,4) = (-4.877*x(1,1)^2+120.2*x(1,1)+1088)+273.15; %solidus
x(1,5) = Q04/1000; %This is the surface heat flow (W/m^2)
x(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   x(i,2)=x(i-1,2)+step; %km
   if x(i,2)<cth4
       x(i,1)=(rhoc*g*x(i,2)*1000)/10^9;
   else
       x(i,1)=(rhoc*g*cth4*1000)/10^9+(rhoml*g*((x(i,2)-
cth4)*1000))/10^9;
   end

   %These are the adiabat and solidus
   x(i,3)=slope*x(i,2)+x(1,3);
   x(i,4)=(-4.877*x(i,1)^2+120.2*x(i,1)+1088)+273.15; %solidus

   %This is the q value
   if x(i,2)<cth4
       q=x(i-1,5)-A4c*(x(i,2)-x(i-1,2))*1000;
   elseif x(i,2)>cth4 && x(i,2)<cmth4
       q=x(i-1,5)-A4m*(x(i,2)-x(i-1,2))*1000;
   else
       q=x(i-1,5)-Am*(x(i,2)-x(i-1,2))*1000;
   end
   x(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if x(i,2)<cth4
       k=kc;
   else
       k=klth;
   end
```

```
   x(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this z value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if x(i,2)<cth4
       T=x(i-1,7)+x(i,5)*((x(i,2)-x(i-1,2))*1000)/x(i,6)-A4c*((x(i,2)-
x(i-1,2))*1000)^2/(2*x(i,6));
   elseif x(i,2)>cth4 && x(i,2)<cmth4
       T=x(i-1,7)+x(i,5)*((x(i,2)-x(i-1,2))*1000)/x(i,6)-A4m*((x(i,2)-
x(i-1,2))*1000)^2/(2*x(i,6));
   else
       T=x(i-1,7)+x(i,5)*((x(i,2)-x(i-1,2))*1000)/x(i,6)-Am*((x(i,2)-
x(i-1,2))*1000)^2/(2*x(i,6));
   end
   x(i,7)=T;

   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
       %Step 5: Calculate k using most recent temperature calculation
       if x(i,2)<cth4
           k=kc*(1+0.0015*x(i,2))/(1+0.0001*(x(i,j-1)-273));
       elseif x(i,2)>cth4
           if x(i,j-1)<500
               k=1/(0.0741+0.000502*x(i,j-1));
           else %Take the larger of these values
               k1=1/(0.0741+0.000502*x(i,j-1))+0.0023*(x(i,j-1)-500);
               %or
               k2=1.26*(1+x(i,2)/1000)+0.0023*(x(i,j-1)-500);
               if k1>k2
                   k=k1;
               else
                   k=k2;
               end
           end
       end
       x(i,j)=k; %Save the applicable k in the array

       x(1,j+1)=T0; %set T on first row

       %Step 6: Calculate the temperature using the most recent k
       if x(i,2)<cth4
           T=x(i-1,j+1)+x(i,5)*(x(i,2)-x(i-1,2))*1000/x(i,j)-
A4c*((x(i,2)-x(i-1,2))*1000)^2/(2*x(i,j));
       elseif x(i,2)>cth4 && x(i,2)<cmth4
           T=x(i-1,j+1)+x(i,5)*(x(i,2)-x(i-1,2))*1000/x(i,j)-
A4m*((x(i,2)-x(i-1,2))*1000)^2/(2*x(i,j));
       else
           T=x(i-1,j+1)+x(i,5)*(x(i,2)-x(i-1,2))*1000/x(i,j)-
Am*((x(i,2)-x(i-1,2))*1000)^2/(2*x(i,j));
       end
       x(i,j+1)=T; %Stores the values for T in the array
   end %Ends the Step 5 and 6 loop
   x1(i,1)=x(i,1)*10000;
   x1(1,2)=x(1,33);
   x1(i,2)=x(i,33);
   A4(i,1)=x(i,33)-x(i,3);
```

```
   B4(i,1)=x(i,1);
   C4(i,1)=x(i,2);
   PFp4 = find(A4>0,1,'first');
   PFm4 = find(A4<0,1,'last');
   PF4 = (B4(PFp4)+B4(PFm4))/2; %calulate pressure
   dF4 = (C4(PFp4)+C4(PFm4))/2; %calculate lithosphere+crust thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w=zeros(31,5);
w(1,1)=cth4;
w(2,1)=Q04;
w(3,1)=Tad4;

syms P0
eqn1 = slope2*P0+(Tad4+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P04 = min(vpasolve(eqn1,P0)); %pressure of adiabat and solidus
intersection
P04 = double(P04);
w(4,1)=P04;
w(5,1)=PF4;
w(6,1)=dF4;

dTdPsolidus4 = 2*(-4.877*PF4)+120.2; %solidus slope
dTdF4 = dTdFa*PF4+dTdFb;
w(7,1)=dTdPsolidus4;
w(8,1)=dTdF4;

% Use P0 and PF to calculate F (percentage of melt produced)
F4 = ((dTdPsolidus4-slope2)/((deltaHF/C_P)+dTdF4))*(P04-PF4)*100;
w(9,1)=F4;

% Calculate the composition of the melt and error
SiO24 = (SiO2A*PF4+SiO2B)*F4^2+(SiO2C*PF4+SiO2D)*F4+(SiO2E*PF4+SiO2F);
TiO24 = TiO2A*F4^TiO2B;
Al2O34 =
(Al2O3A*PF4+Al2O3B)*F4^2+(Al2O3C*PF4+Al2O3D)*F4+(Al2O3E*PF4+Al2O3F);
Cr2O34 =
(Cr2O3A*PF4+Cr2O3B)*F4^2+(Cr2O3C*PF4+Cr2O3D)*F4+(Cr2O3E*PF4+Cr2O3F);
FeO4 = (FeOA*PF4+FeOB)*F4^2+(FeOC*PF4+FeOD)*F4+(FeOE*PF4+FeOF);
MnO4 = (MnOA*PF4+MnOB)*F4^2+(MnOC*PF4+MnOD)*F4+(MnOE*PF4+MnOF);
MgO4 = (MgOA*PF4+MgOB)*F4^2+(MgOC*PF4+MgOD)*F4+(MgOE*PF4+MgOF);
CaO4 = CaOA*F4^3+CaOB*F4^2+CaOC*F4+CaOD;
Na2O4 = (Na2OA*PF4+Na2OB)*F4^2+(Na2OC*PF4+Na2OD)*F4+(Na2OE*PF4+Na2OF);
K2O4 = K2OA*F4^K2OB;
P2O54 = P2O5A*F4^P2O5B;
w(10,1)=SiO24;
w(11,1)=TiO24;
w(12,1)=Al2O34;
w(13,1)=Cr2O34;
w(14,1)=FeO4;
w(15,1)=MnO4;
w(16,1)=MgO4;
w(17,1)=CaO4;
w(18,1)=Na2O4;
w(19,1)=K2O4;
w(20,1)=P2O54;
```

```
% Calculate the residual using batch melting- will become mantle lithos
SiO2res4 = (mantleSiO2-((F4/100)*SiO24))/(1-(F4/100));
TiO2res4 = (mantleTiO2-((F4/100)*TiO24))/(1-(F4/100));
Al2O3res4 = (mantleAl2O3-((F4/100)*Al2O34))/(1-(F4/100));
Cr2O3res4 = (mantleCr2O3-((F4/100)*Cr2O34))/(1-(F4/100));
FeOres4 = (mantleFeO-((F4/100)*FeO4))/(1-(F4/100));
MnOres4 = (mantleMnO-((F4/100)*MnO4))/(1-(F4/100));
MgOres4 = (mantleMgO-((F4/100)*MgO4))/(1-(F4/100));
CaOres4 = (mantleCaO-((F4/100)*CaO4))/(1-(F4/100));
Na2Ores4 = (mantleNa2O-((F4/100)*Na2O4))/(1-(F4/100));
K2Ores4 = (mantleK2O-((F4/100)*K2O4))/(1-(F4/100));
P2O5res4 = (mantleP2O5-((F4/100)*P2O54))/(1-(F4/100));
w(21,1)=SiO2res4;
w(22,1)=TiO2res4;
w(23,1)=Al2O3res4;
w(24,1)=Cr2O3res4;
w(25,1)=FeOres4;
w(26,1)=MnOres4;
w(27,1)=MgOres4;
w(28,1)=CaOres4;
w(29,1)=Na2Ores4;
w(30,1)=K2Ores4;
w(31,1)=P2O5res4;
%End of 4 Ga (will become one distinct layer)


%% 3 Ga
% Input the initial parameters TO BE CHANGED WITH TIME
t = 3*10^9; %time in years, 0 is present
Q03 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad3 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc3 = Um/(F4+(1-F4)*DU); %Uranium in crust
Thc3 = Thm/(F4+(1-F4)*DTh); %Thorium in crust
Kc3 = Km/(F4+(1-F4)*DK); %Potassium in crust

Um3 = Um-F4*Uc3/(1-F4); %Uranium in crust
Thm3 = Thm-F4*Thc3/(1-F4); %Thorium in crust
Km3 = Km-F4*Kc3/(1-F4); %Potassium in crust

HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
```

```
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat
production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)

%Calculate the layer thicknesses
lay2c = 20*(F4/100); %km
vol2c = (4/3)*pi*(rad^3-(rad-lay2c)^3); %km^3 volume of crustal layer
mass2c = vol2c*(1000^3)*rhoc; %kg mass crustal layer

lay1c = (rad-lay2c)-((rad-lay2c)^3-vol1c/((4/3)*pi))^(1/3); %km

lay1m = (rad-lay2c-lay1c)-((rad-lay2c-lay1c)^3-vol1m/((4/3)*pi))^(1/3);
%km

mass2m = mass2c/(F4/100)-mass2c; %kg mass lithosphere layer
vol2m = mass2m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay2m = (rad-lay2c-lay1c-lay1m)-((rad-lay2c-lay1c-lay1m)^3-
vol2m/((4/3)*pi))^(1/3); %km

%Calculate base (depth) of each layer
cth3 = lay2c+lay1c; %total thickness of crust in km
cmth3 = cth3+lay1m+lay2m;

A3c = Atotc4*(lay1c/cth3)+Atotc3*(lay2c/cth3);
A3m = Atotm4*(lay1m/(cmth3-cth3))+Atotm3*(lay2m/(cmth3-cth3));
Am = Atotm;
%% Calculate areotherm 3 Ga
y = zeros(d,33);
y1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
y(1,1) = 0; %Initial Pressure (GPa)
y(1,2) = 0; %Initial Depth (km)
y(1,3) = Tad3+273;
```

```
y(1,4) = (-4.877*y(1,1)^2+120.2*y(1,1)+1088)+273.15; %solidus
y(1,5) = Q03/1000; %This is the surface heat flow (W/m^2)
y(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   y(i,2)=y(i-1,2)+step; %km
   if y(i,2)<cth3
       y(i,1)=(rhoc*g*y(i,2)*1000)/10^9;
   else
       y(i,1)=(rhoc*g*cth3*1000)/10^9+(rhoml*g*((y(i,2)-
cth3)*1000))/10^9;
   end

   %These are the adiabat and solidus
   y(i,3)=slope*y(i,2)+y(1,3);
   y(i,4)=(-4.877*y(i,1)^2+120.2*y(i,1)+1088)+273.15; %solidus

   %This is the q value
   if y(i,2)<cth3
       q=y(i-1,5)-A3c*(y(i,2)-y(i-1,2))*1000;
   elseif y(i,2)>cth3 && y(i,2)<cmth3
       q=y(i-1,5)-A3m*(y(i,2)-y(i-1,2))*1000;
   else
       q=y(i-1,5)-Am*(y(i,2)-y(i-1,2))*1000;
   end
   y(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if y(i,2)<cth3
       k=kc;
   else
       k=klth;
   end
   y(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this z value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if y(i,2)<cth3
       T=y(i-1,7)+y(i,5)*((y(i,2)-y(i-1,2))*1000)/y(i,6)-A3c*((y(i,2)-
y(i-1,2))*1000)^2/(2*y(i,6));
   elseif y(i,2)>cth3 && y(i,2)<cmth3
       T=y(i-1,7)+y(i,5)*((y(i,2)-y(i-1,2))*1000)/y(i,6)-A3m*((y(i,2)-
y(i-1,2))*1000)^2/(2*y(i,6));
   else
       T=y(i-1,7)+y(i,5)*((y(i,2)-y(i-1,2))*1000)/y(i,6)-Am*((y(i,2)-
y(i-1,2))*1000)^2/(2*y(i,6));
   end
   y(i,7)=T;

   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
       %Step 5: Calculate k using most recent temperature calculation
       if y(i,2)<cth3
           k=kc*(1+0.0015*y(i,2))/(1+0.0001*(y(i,j-1)-273));
```

```
       elseif y(i,2)>cth3
           if y(i,j-1)<500
               k=1/(0.0741+0.000502*y(i,j-1));
           else %Take the larger of these values
               k1=1/(0.0741+0.000502*y(i,j-1))+0.0023*(y(i,j-1)-500);
               %or
               k2=1.26*(1+y(i,2)/1000)+0.0023*(y(i,j-1)-500);
               if k1>k2
                   k=k1;
               else
                   k=k2;
               end
           end
       end
       y(i,j)=k; %Save the applicable k in the array

       y(1,j+1)=T0; %set T on first row

       %Step 6: Calculate the temperature using the most recent k
       if y(i,2)<cth3
           T=y(i-1,j+1)+y(i,5)*(y(i,2)-y(i-1,2))*1000/y(i,j)-
A3c*((y(i,2)-y(i-1,2))*1000)^2/(2*y(i,j));
       elseif y(i,2)>cth3 && y(i,2)<cmth3
           T=y(i-1,j+1)+y(i,5)*(y(i,2)-y(i-1,2))*1000/y(i,j)-
A3m*((y(i,2)-y(i-1,2))*1000)^2/(2*y(i,j));
       else
           T=y(i-1,j+1)+y(i,5)*(y(i,2)-y(i-1,2))*1000/y(i,j)-
Am*((y(i,2)-y(i-1,2))*1000)^2/(2*y(i,j));
       end
       y(i,j+1)=T; %Stores the values for T in the array
   end %Ends the Step 5 and 6 loop
   y1(i,1)=y(i,1)*10000;
   y1(1,2)=y(1,33);
   y1(i,2)=y(i,33);
   A3(i,1)=y(i,33)-y(i,3);
   B3(i,1)=y(i,1);
   C3(i,1)=y(i,2);
   PFp3 = find(A3>0,1,'first');
   PFm3 = find(A3<0,1,'last');
   PF3 = (B3(PFp3)+B3(PFm3))/2; %calulate pressure
   dF3 = (C3(PFp3)+C3(PFm3))/2; %calculate crust+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,2)=cth3;
w(2,2)=Q03;
w(3,2)=Tad3;

syms P0
eqn1 = slope2*P0+(Tad3+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P03 = min(vpasolve(eqn1,P0)); %pressure of solidus-adiabat intersection
P03 = double(P03);
w(4,2)=P03;
w(5,2)=PF3;
w(6,2)=dF3;
```

```
dTdPsolidus3 = 2*(-4.877*PF3)+120.2; %slope of solidus Duncan et al. (2018)
dTdF3 = dTdFa*PF3+dTdFb;
w(7,2)=dTdPsolidus3;
w(8,2)=dTdF3;

% Use P0 and PF to calculate F (percentage of melt produced)
F3 = ((dTdPsolidus3-slope2)/((deltaHF/C_P)+dTdF3))*(P03-PF3)*100;
w(9,2)=F3;

% Calculate the composition of the melt and error
SiO23 = (SiO2A*PF3+SiO2B)*F3^2+(SiO2C*PF3+SiO2D)*F3+(SiO2E*PF3+SiO2F);
TiO23 = TiO2A*F3^TiO2B;
Al2O33 = (Al2O3A*PF3+Al2O3B)*F3^2+(Al2O3C*PF3+Al2O3D)*F3+(Al2O3E*PF3+Al2O3F);
Cr2O33 = (Cr2O3A*PF3+Cr2O3B)*F3^2+(Cr2O3C*PF3+Cr2O3D)*F3+(Cr2O3E*PF3+Cr2O3F);
FeO3 = (FeOA*PF3+FeOB)*F3^2+(FeOC*PF3+FeOD)*F3+(FeOE*PF3+FeOF);
MnO3 = (MnOA*PF3+MnOB)*F3^2+(MnOC*PF3+MnOD)*F3+(MnOE*PF3+MnOF);
MgO3 = (MgOA*PF3+MgOB)*F3^2+(MgOC*PF3+MgOD)*F3+(MgOE*PF3+MgOF);
CaO3 = CaOA*F3^3+CaOB*F3^2+CaOC*F3+CaOD;
Na2O3 = (Na2OA*PF3+Na2OB)*F3^2+(Na2OC*PF3+Na2OD)*F3+(Na2OE*PF3+Na2OF);
K2O3 = K2OA*F3^K2OB;
P2O53 = P2O5A*F3^P2O5B;
w(10,2)=SiO23;
w(11,2)=TiO23;
w(12,2)=Al2O33;
w(13,2)=Cr2O33;
w(14,2)=FeO3;
w(15,2)=MnO3;
w(16,2)=MgO3;
w(17,2)=CaO3;
w(18,2)=Na2O3;
w(19,2)=K2O3;
w(20,2)=P2O53;

% Calculate the residual using batch melting- will become mantle lithos
SiO2res3 = (mantleSiO2-((F3/100)*SiO23))/(1-(F3/100));
TiO2res3 = (mantleTiO2-((F3/100)*TiO23))/(1-(F3/100));
Al2O3res3 = (mantleAl2O3-((F3/100)*Al2O33))/(1-(F3/100));
Cr2O3res3 = (mantleCr2O3-((F3/100)*Cr2O33))/(1-(F3/100));
FeOres3 = (mantleFeO-((F3/100)*FeO3))/(1-(F3/100));
MnOres3 = (mantleMnO-((F3/100)*MnO3))/(1-(F3/100));
MgOres3 = (mantleMgO-((F3/100)*MgO3))/(1-(F3/100));
CaOres3 = (mantleCaO-((F3/100)*CaO3))/(1-(F3/100));
Na2Ores3 = (mantleNa2O-((F3/100)*Na2O3))/(1-(F3/100));
K2Ores3 = (mantleK2O-((F3/100)*K2O3))/(1-(F3/100));
P2O5res3 = (mantleP2O5-((F3/100)*P2O53))/(1-(F3/100));
w(21,2)=SiO2res3;
w(22,2)=TiO2res3;
w(23,2)=Al2O3res3;
w(24,2)=Cr2O3res3;
w(25,2)=FeOres3;
w(26,2)=MnOres3;
w(27,2)=MgOres3;
w(28,2)=CaOres3;
```

```
w(29,2)=Na2Ores3;
w(30,2)=K2Ores3;
w(31,2)=P2O5res3;
%End of 3 Ga (becomes next distinct layer)



%% 2 Ga
t = 2*10^9; %time in years, 0 is present
Q02 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad2 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc2 = Um/(F3+(1-F3)*DU); %Uranium in crust
Thc2 = Thm/(F3+(1-F3)*DTh); %Thorium in crust
Kc2 = Km/(F3+(1-F3)*DK); %Potassium in crust

Um2 = Um-F3*Uc2/(1-F3); %Uranium in mantle lithosphere
Thm2 = Thm-F3*Thc2/(1-F3); %Thorium in mantle lithosphere
Km2 = Km-F3*Kc2/(1-F3); %Potassium in mantle lithosphere

HU238c2 = Uc2*U238*H238U*exp(t*log(2)/U238t);
HU235c2 = Uc2*U235*H235U*exp(t*log(2)/U235t);
HTh232c2 = Thc2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c2 = Kc2*K40*H40K*exp(t*log(2)/K40t);

HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m2 = Um2*U238*H238U*exp(t*log(2)/U238t);
HU235m2 = Um2*U235*H235U*exp(t*log(2)/U235t);
HTh232m2 = Thm2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m2 = Km2*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);
```

```
Atotc2 = (HU238c2+HU235c2+HTh232c2+HK40c2)*rhoc; %volumetric heat
production crust (W/m3)
Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat
production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm2 = (HU238m2+HU235m2+HTh232m2+HK40m2)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)

%Calculate the layer thicknesses
lay3c = 20*(F3/100); %km
vol3c = (4/3)*pi*(rad^3-(rad-lay3c)^3); %km^3 volume of crustal layer
mass3c = vol3c*(1000^3)*rhoc; %kg mass crustal layer

lay2c = (rad-lay3c)-((rad-lay3c)^3-vol2c/((4/3)*pi))^(1/3); %km

lay1c = (rad-lay3c-lay2c)-((rad-lay3c-lay2c)^3-vol1c/((4/3)*pi))^(1/3);
%km

lay1m = (rad-lay3c-lay2c-lay1c)-((rad-lay3c-lay2c-lay1c)^3-
vol1m/((4/3)*pi))^(1/3); %km

lay2m = (rad-lay3c-lay2c-lay1c-lay1m)-((rad-lay3c-lay2c-lay1c-lay1m)^3-
vol2m/((4/3)*pi))^(1/3); %km

mass3m = mass3c/(F3/100)-mass3c; %kg mass lithosphere layer
vol3m = mass3m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay3m = (rad-lay3c-lay2c-lay1c-lay1m-lay2m)-((rad-lay3c-lay2c-lay1c-
lay1m-lay2m)^3-vol3m/((4/3)*pi))^(1/3); %km

%Calculate base (depth) of each layer
cth2 = lay3c+lay2c+lay1c; %total thickness of crust in km
cmth2 = cth2+lay1m+lay2m+lay3m;

A2c = Atotc4*lay1c/cth2+Atotc3*lay2c/cth2+Atotc2*lay3c/cth2;
A2m = Atotm4*lay1m/(cmth2-cth2)+Atotm3*lay2m/(cmth2-
cth2)+Atotm2*lay3m/(cmth2-cth2);
Am = Atotm;
%% Calculate areotherm 2Ga
z = zeros(d,33);
z1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
z(1,1) = 0; %Initial Pressure (GPa)
z(1,2) = 0; %Initial Depth (km)
z(1,3) = Tad2+273;
z(1,4) = (-4.877*z(1,1)^2+120.2*z(1,1)+1088)+273.15; %solidus
```

```
z(1,5) = Q02/1000; %This is the surface heat flow (W/m^2)
z(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   z(i,2)=z(i-1,2)+step; %km
   if z(i,2)<cth2
       z(i,1)=(rhoc*g*z(i,2)*1000)/10^9;
   else
       z(i,1)=(rhoc*g*cth2*1000)/10^9+(rhoml*g*((z(i,2)-
cth2)*1000))/10^9;
   end

   %These are the adiabat and solidus
   z(i,3)=slope*z(i,2)+z(1,3);
   z(i,4)=(-4.877*z(i,1)^2+120.2*z(i,1)+1088)+273.15; %solidus

   %This is the q value
   if z(i,2)<cth2
       q=z(i-1,5)-A2c*(z(i,2)-z(i-1,2))*1000;
   elseif z(i,2)>cth2 && z(i,2)<cmth2
       q=z(i-1,5)-A2m*(z(i,2)-z(i-1,2))*1000;
   else
       q=z(i-1,5)-Am*(z(i,2)-z(i-1,2))*1000;
   end
   z(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if z(i,2)<cth2
       k=kc;
   else
       k=klth;
   end
   z(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this z value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if z(i,2)<cth2
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-A2c*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   elseif z(i,2)>cth2 && z(i,2)<cmth2
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-A2m*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   else
       T=z(i-1,7)+z(i,5)*((z(i,2)-z(i-1,2))*1000)/z(i,6)-Am*((z(i,2)-
z(i-1,2))*1000)^2/(2*z(i,6));
   end
   z(i,7)=T;

   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
       %Step 5: Calculate k using most recent temperature calculation
       if z(i,2)<cth2
           k=kc*(1+0.0015*z(i,2))/(1+0.0001*(z(i,j-1)-273));
       elseif z(i,2)>cth2
```

```
           if z(i,j-1)<500
               k=1/(0.0741+0.000502*z(i,j-1));
           else %Take the larger of these values
               k1=1/(0.0741+0.000502*z(i,j-1))+0.0023*(z(i,j-1)-500);
               %or
               k2=1.26*(1+z(i,2)/1000)+0.0023*(z(i,j-1)-500);
               if k1>k2
                   k=k1;
               else
                   k=k2;
               end
           end
       end
       z(i,j)=k; %Save the applicable k in the array

       z(1,j+1)=T0; %set T on first row

       %Step 6: Calculate the temperature using the most recent k
       if z(i,2)<cth2
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
A2c*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       elseif z(i,2)>cth2 && z(i,2)<cmth2
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
A2m*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       else
           T=z(i-1,j+1)+z(i,5)*(z(i,2)-z(i-1,2))*1000/z(i,j)-
Am*((z(i,2)-z(i-1,2))*1000)^2/(2*z(i,j));
       end
       z(i,j+1)=T; %Stores the values for T in the array
   end %Ends the Step 5 and 6 loop
   z1(i,1)=z(i,1)*10000;
   z1(1,2)=z(1,33);
   z1(i,2)=z(i,33);
   A2(i,1)=z(i,33)-z(i,3);
   B2(i,1)=z(i,1);
   C2(i,1)=z(i,2);
   PFp2 = find(A2>0,1,'first');
   PFm2 = find(A2<0,1,'last');
   PF2 = (B2(PFp2)+B2(PFm2))/2; %calulate pressure
   dF2 = (C2(PFp2)+C2(PFm2))/2; %calculate crusth+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,3)=cth2;
w(2,3)=Q02;
w(3,3)=Tad2;

syms P0
eqn1 = slope2*P0+(Tad2+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P02 = min(vpasolve(eqn1,P0));
P02 = double(P02); %pressure where solidus and adiabat intersect
w(4,3)=P02;
w(5,3)=PF2;
w(6,3)=dF2;

dTdPsolidus2 = 2*(-4.877*PF2)+120.2; %slope of solidus Duncan et al.
(2018)
```

```
dTdF2 = dTdFa*PF2+dTdFb;
w(7,3)=dTdPsolidus2;
w(8,3)=dTdF2;

% Use P0 and PF to calculate F (percentage of melt produced)
F2 = ((dTdPsolidus2-slope2)/((deltaHF/C_P)+dTdF2))*(P02-PF2)*100;
w(9,3)=F2;

% Calculate the composition of the melt and error
SiO22 = (SiO2A*PF2+SiO2B)*F2^2+(SiO2C*PF2+SiO2D)*F2+(SiO2E*PF2+SiO2F);
TiO22 = TiO2A*F2^TiO2B;
Al2O32 =
(Al2O3A*PF2+Al2O3B)*F2^2+(Al2O3C*PF2+Al2O3D)*F2+(Al2O3E*PF2+Al2O3F);
Cr2O32 =
(Cr2O3A*PF2+Cr2O3B)*F2^2+(Cr2O3C*PF2+Cr2O3D)*F2+(Cr2O3E*PF2+Cr2O3F);
FeO2 = (FeOA*PF2+FeOB)*F2^2+(FeOC*PF2+FeOD)*F2+(FeOE*PF2+FeOF);
MnO2 = (MnOA*PF2+MnOB)*F2^2+(MnOC*PF2+MnOD)*F2+(MnOE*PF2+MnOF);
MgO2 = (MgOA*PF2+MgOB)*F2^2+(MgOC*PF2+MgOD)*F2+(MgOE*PF2+MgOF);
CaO2 = CaOA*F2^3+CaOB*F2^2+CaOC*F2+CaOD;
Na2O2 = (Na2OA*PF2+Na2OB)*F2^2+(Na2OC*PF2+Na2OD)*F2+(Na2OE*PF2+Na2OF);
K2O2 = K2OA*F2^K2OB;
P2O52 = P2O5A*F2^P2O5B;
w(10,3)=SiO22;
w(11,3)=TiO22;
w(12,3)=Al2O32;
w(13,3)=Cr2O32;
w(14,3)=FeO2;
w(15,3)=MnO2;
w(16,3)=MgO2;
w(17,3)=CaO2;
w(18,3)=Na2O2;
w(19,3)=K2O2;
w(20,3)=P2O52;

% Calculate the residual using batch melting-will become mantle lithos
SiO2res2 = (mantleSiO2-((F2/100)*SiO22))/(1-(F2/100));
TiO2res2 = (mantleTiO2-((F2/100)*TiO22))/(1-(F2/100));
Al2O3res2 = (mantleAl2O3-((F2/100)*Al2O32))/(1-(F2/100));
Cr2O3res2 = (mantleCr2O3-((F2/100)*Cr2O32))/(1-(F2/100));
FeOres2 = (mantleFeO-((F2/100)*FeO2))/(1-(F2/100));
MnOres2 = (mantleMnO-((F2/100)*MnO2))/(1-(F2/100));
MgOres2 = (mantleMgO-((F2/100)*MgO2))/(1-(F2/100));
CaOres2 = (mantleCaO-((F2/100)*CaO2))/(1-(F2/100));
Na2Ores2 = (mantleNa2O-((F2/100)*Na2O2))/(1-(F2/100));
K2Ores2 = (mantleK2O-((F2/100)*K2O2))/(1-(F2/100));
P2O5res2 = (mantleP2O5-((F2/100)*P2O52))/(1-(F2/100));
w(21,3)=SiO2res2;
w(22,3)=TiO2res2;
w(23,3)=Al2O3res2;
w(24,3)=Cr2O3res2;
w(25,3)=FeOres2;
w(26,3)=MnOres2;
w(27,3)=MgOres2;
w(28,3)=CaOres2;
w(29,3)=Na2Ores2;
w(30,3)=K2Ores2;
```

```
w(31,3)=P2O5res2;
%End of 2 Ga calculations


%% 1 Ga
t = 1*10^9; %time in years, 0 is present
Q01 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad1 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc1 = Um/(F2+(1-F2)*DU); %Uranium in crust
Thc1 = Thm/(F2+(1-F2)*DTh); %Thorium in crust
Kc1 = Km/(F2+(1-F2)*DK); %Potassium in crust

Um1 = Um-F2*Uc1/(1-F2); %Uranium in mantle lithosphere
Thm1 = Thm-F2*Thc1/(1-F2); %Thorium in mantle lithosphere
Km1 = Km-F2*Kc1/(1-F2); %Potassium in mantle lithosphere

HU238c1 = Uc1*U238*H238U*exp(t*log(2)/U238t);
HU235c1 = Uc1*U235*H235U*exp(t*log(2)/U235t);
HTh232c1 = Thc1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c1 = Kc1*K40*H40K*exp(t*log(2)/K40t);

HU238c2 = Uc2*U238*H238U*exp(t*log(2)/U238t);
HU235c2 = Uc2*U235*H235U*exp(t*log(2)/U235t);
HTh232c2 = Thc2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c2 = Kc2*K40*H40K*exp(t*log(2)/K40t);

HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m2 = Um2*U238*H238U*exp(t*log(2)/U238t);
HU235m2 = Um2*U235*H235U*exp(t*log(2)/U235t);
HTh232m2 = Thm2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m2 = Km2*K40*H40K*exp(t*log(2)/K40t);
```

```
HU238m1 = Um1*U238*H238U*exp(t*log(2)/U238t);
HU235m1 = Um1*U235*H235U*exp(t*log(2)/U235t);
HTh232m1 = Thm1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m1 = Km1*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc1 = (HU238c1+HU235c1+HTh232c1+HK40c1)*rhoc; %volumetric heat
production crust (W/m3)
Atotc2 = (HU238c2+HU235c2+HTh232c2+HK40c2)*rhoc; %volumetric heat
production crust (W/m3)
Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat
production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm2 = (HU238m2+HU235m2+HTh232m2+HK40m2)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm1 = (HU238m1+HU235m1+HTh232m1+HK40m1)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)

%Calculate the layer thicknesses
lay4c = 20*(F2/100); %km
vol4c = (4/3)*pi*(rad^3-(rad-lay4c)^3); %km^3 volume of crustal layer
mass4c = vol4c*(1000^3)*rhoc; %kg mass crustal layer

lay3c = (rad-lay4c)-((rad-lay4c)^3-vol3c/((4/3)*pi))^(1/3); %km

lay2c = (rad-lay4c-lay3c)-((rad-lay4c-lay3c)^3-vol2c/((4/3)*pi))^(1/3);
%km

lay1c = (rad-lay4c-lay3c-lay2c)-((rad-lay4c-lay3c-lay2c)^3-
vol1c/((4/3)*pi))^(1/3); %km

lay1m = (rad-lay4c-lay3c-lay2c-lay1c)-((rad-lay4c-lay3c-lay2c-lay1c)^3-
vol1m/((4/3)*pi))^(1/3); %km

lay2m = (rad-lay4c-lay3c-lay2c-lay1c-lay1m)-((rad-lay4c-lay3c-lay2c-
lay1c-lay1m)^3-vol2m/((4/3)*pi))^(1/3); %km

lay3m = (rad-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m)-((rad-lay4c-lay3c-
lay2c-lay1c-lay1m-lay2m)^3-vol3m/((4/3)*pi))^(1/3); %km

mass4m = mass4c/(F2/100)-mass3c; %kg mass lithosphere layer
vol4m = mass4m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay4m = (rad-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)-((rad-lay4c-
lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)^3-vol4m/((4/3)*pi))^(1/3); %km
```

```
%Calculate base (depth) of each layer
cth1 = lay4c+lay3c+lay2c+lay1c; %total thickness of crust in km
cmth1 = cth1+lay1m+lay2m+lay3m+lay4m;

A1c =
Atotc4*lay1c/cth1+Atotc3*lay2c/cth1+Atotc2*lay3c/cth1+Atotc1*lay4c/cth1
;
A1m = Atotm4*lay1m/(cmth1-cth1)+Atotm3*lay2m/(cmth1-
cth1)+Atotm2*lay3m/(cmth1-cth1)+Atotm1*lay4m/(cmth1-cth1);
Am = Atotm;
%% Calculate areotherm 1 Ga
b = zeros(d,33);
b1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
b(1,1) = 0; %Initial Pressure (GPa)
b(1,2) = 0; %Initial Depth (km)
b(1,3) = Tad1+273;
b(1,4) = (-4.877*b(1,1)^2+120.2*b(1,1)+1088)+273.15; %solidus
b(1,5) = Q01/1000; %This is the surface heat flow (W/m^2)
b(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   b(i,2)=b(i-1,2)+step; %km
   if b(i,2)<cth1
       b(i,1)=(rhoc*g*b(i,2)*1000)/10^9;
   else
       b(i,1)=(rhoc*g*cth1*1000)/10^9+(rhoml*g*((b(i,2)-
cth1)*1000))/10^9;
   end

   %These are the adiabat and solidus
   b(i,3)=slope*b(i,2)+b(1,3);
   b(i,4)=(-4.877*b(i,1)^2+120.2*b(i,1)+1088)+273.15; %solidus

   %This is the q value
   if b(i,2)<cth1
       q=b(i-1,5)-A1c*(b(i,2)-b(i-1,2))*1000;
   elseif b(i,2)>cth1 && b(i,2)<cmth1
       q=b(i-1,5)-A1m*(b(i,2)-b(i-1,2))*1000;
   else
       q=b(i-1,5)-Am*(b(i,2)-b(i-1,2))*1000;
   end
   b(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if b(i,2)<cth1
       k=kc;
   else
       k=klth;
   end
```

```
   b(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this b value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if b(i,2)<cth1
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A1c*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   elseif b(i,2)>cth1 && b(i,2)<cmth1
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-A1m*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   else
       T=b(i-1,7)+b(i,5)*((b(i,2)-b(i-1,2))*1000)/b(i,6)-Am*((b(i,2)-
b(i-1,2))*1000)^2/(2*b(i,6));
   end
   b(i,7)=T;

   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
       %Step 5: Calculate k using most recent temperature calculation
       if b(i,2)<cth1
           k=kc*(1+0.0015*b(i,2))/(1+0.0001*(b(i,j-1)-273));
       elseif b(i,2)>cth1
           if b(i,j-1)<500
               k=1/(0.0741+0.000502*b(i,j-1));
           else %Take the larger of these values
               k1=1/(0.0741+0.000502*b(i,j-1))+0.0023*(b(i,j-1)-500);
               %or
               k2=1.26*(1+b(i,2)/1000)+0.0023*(b(i,j-1)-500);
               if k1>k2
                   k=k1;
               else
                   k=k2;
               end
           end
       end
       b(i,j)=k; %Save the applicable k in the array

       b(1,j+1)=T0; %set T on first row

       %Step 6: Calculate the temperature using the most recent k
       if b(i,2)<cth1
           T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A1c*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
       elseif b(i,2)>cth1 && b(i,2)<cmth1
           T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
A1m*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
       else
           T=b(i-1,j+1)+b(i,5)*(b(i,2)-b(i-1,2))*1000/b(i,j)-
Am*((b(i,2)-b(i-1,2))*1000)^2/(2*b(i,j));
       end
       b(i,j+1)=T; %Stores the values for T in the array
   end %Ends the Step 5 and 6 loop
   b1(i,1)=b(i,1)*10000;
   b1(1,2)=b(1,33);
   b1(i,2)=b(i,33);
   A1(i,1)=b(i,33)-b(i,3);
```

```
   B1(i,1)=b(i,1);
   C1(i,1)=b(i,2);
   PFp1 = find(A1>0,1,'first');
   PFm1 = find(A1<0,1,'last');
   PF1 = (B1(PFp1)+B1(PFm1))/2; %calulate pressure
   dF1 = (C1(PFp1)+C1(PFm1))/2; %calculate crusth+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,4)=cth1;
w(2,4)=Q01;
w(3,4)=Tad1;

syms P0
eqn1 = slope2*P0+(Tad1+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P01 = min(vpasolve(eqn1,P0));
P01 = double(P01); %pressure where solidus and adiabat intersect
w(4,4)=P01;
w(5,4)=PF1;
w(6,4)=dF1;

dTdPsolidus1 = 2*(-4.877*PF1)+120.2; %slope of solidus Duncan et al.
(2018)
dTdF1 = dTdFa*PF1+dTdFb;
w(7,4)=dTdPsolidus1;
w(8,4)=dTdF1;

% Use P0 and PF to calculate F (percentage of melt produced)
F1 = ((dTdPsolidus1-slope2)/((deltaHF/C_P)+dTdF1))*(P01-PF1)*100;
w(9,4)=F1;

% Calculate the composition of the melt and error
SiO21 = (SiO2A*PF1+SiO2B)*F1^2+(SiO2C*PF1+SiO2D)*F1+(SiO2E*PF1+SiO2F);
TiO21 = TiO2A*F1^TiO2B;
Al2O31 =
(Al2O3A*PF1+Al2O3B)*F1^2+(Al2O3C*PF1+Al2O3D)*F1+(Al2O3E*PF1+Al2O3F);
Cr2O31 =
(Cr2O3A*PF1+Cr2O3B)*F1^2+(Cr2O3C*PF1+Cr2O3D)*F1+(Cr2O3E*PF1+Cr2O3F);
FeO1 = (FeOA*PF1+FeOB)*F1^2+(FeOC*PF1+FeOD)*F1+(FeOE*PF1+FeOF);
MnO1 = (MnOA*PF1+MnOB)*F1^2+(MnOC*PF1+MnOD)*F1+(MnOE*PF1+MnOF);
MgO1 = (MgOA*PF1+MgOB)*F1^2+(MgOC*PF1+MgOD)*F1+(MgOE*PF1+MgOF);
CaO1 = CaOA*F1^3+CaOB*F1^2+CaOC*F1+CaOD;
Na2O1 = (Na2OA*PF1+Na2OB)*F1^2+(Na2OC*PF1+Na2OD)*F1+(Na2OE*PF1+Na2OF);
K2O1 = K2OA*F1^K2OB;
P2O51 = P2O5A*F1^P2O5B;
w(10,4)=SiO21;
w(11,4)=TiO21;
w(12,4)=Al2O31;
w(13,4)=Cr2O31;
w(14,4)=FeO1;
w(15,4)=MnO1;
w(16,4)=MgO1;
w(17,4)=CaO1;
w(18,4)=Na2O1;
w(19,4)=K2O1;
w(20,4)=P2O51;
```

```
% Calculate the residual using batch melting-will become mantle lithos
SiO2res1 = (mantleSiO2-((F1/100)*SiO21))/(1-(F1/100));
TiO2res1 = (mantleTiO2-((F1/100)*TiO21))/(1-(F1/100));
Al2O3res1 = (mantleAl2O3-((F1/100)*Al2O31))/(1-(F1/100));
Cr2O3res1 = (mantleCr2O3-((F1/100)*Cr2O31))/(1-(F1/100));
FeOres1 = (mantleFeO-((F1/100)*FeO1))/(1-(F1/100));
MnOres1 = (mantleMnO-((F1/100)*MnO1))/(1-(F1/100));
MgOres1 = (mantleMgO-((F1/100)*MgO1))/(1-(F1/100));
CaOres1 = (mantleCaO-((F1/100)*CaO1))/(1-(F1/100));
Na2Ores1 = (mantleNa2O-((F1/100)*Na2O1))/(1-(F1/100));
K2Ores1 = (mantleK2O-((F1/100)*K2O1))/(1-(F1/100));
P2O5res1 = (mantleP2O5-((F1/100)*P2O51))/(1-(F1/100));
w(21,4)=SiO2res1;
w(22,4)=TiO2res1;
w(23,4)=Al2O3res1;
w(24,4)=Cr2O3res1;
w(25,4)=FeOres1;
w(26,4)=MnOres1;
w(27,4)=MgOres1;
w(28,4)=CaOres1;
w(29,4)=Na2Ores1;
w(30,4)=K2Ores1;
w(31,4)=P2O5res1;
%End of 1 Ga calculations


%% Present day
t = 0*10^9; %time in years, 0 is present
Q00 = 6.25*(t/10^9)+25; %surface heat flow (mW/m^2), 25 is present
Tad0 = 1345+9.43*(t/10^9)+4.3*(t/10^9)^2; %mantle potential temp (C)
modern Filiberto 2017

%Calculate the heat produced in the crust and mantle lithosphere
Uc0 = Um/(F1+(1-F1)*DU); %Uranium in crust
Thc0 = Thm/(F1+(1-F1)*DTh); %Thorium in crust
Kc0 = Km/(F1+(1-F1)*DK); %Potassium in crust

Um0 = Um-F1*Uc0/(1-F1); %Uranium in mantle lithosphere
Thm0 = Thm-F1*Thc0/(1-F1); %Thorium in mantle lithosphere
Km0 = Km-F1*Kc0/(1-F1); %Potassium in mantle lithosphere

HU238c0 = Uc0*U238*H238U*exp(t*log(2)/U238t);
HU235c0 = Uc0*U235*H235U*exp(t*log(2)/U235t);
HTh232c0 = Thc0*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c0 = Kc0*K40*H40K*exp(t*log(2)/K40t);

HU238c1 = Uc1*U238*H238U*exp(t*log(2)/U238t);
HU235c1 = Uc1*U235*H235U*exp(t*log(2)/U235t);
HTh232c1 = Thc1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c1 = Kc1*K40*H40K*exp(t*log(2)/K40t);

HU238c2 = Uc2*U238*H238U*exp(t*log(2)/U238t);
HU235c2 = Uc2*U235*H235U*exp(t*log(2)/U235t);
HTh232c2 = Thc2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c2 = Kc2*K40*H40K*exp(t*log(2)/K40t);
```

```
HU238c3 = Uc3*U238*H238U*exp(t*log(2)/U238t);
HU235c3 = Uc3*U235*H235U*exp(t*log(2)/U235t);
HTh232c3 = Thc3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c3 = Kc3*K40*H40K*exp(t*log(2)/K40t);

HU238c4 = Uc4*U238*H238U*exp(t*log(2)/U238t);
HU235c4 = Uc4*U235*H235U*exp(t*log(2)/U235t);
HTh232c4 = Thc4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40c4 = Kc4*K40*H40K*exp(t*log(2)/K40t);

HU238m4 = Um4*U238*H238U*exp(t*log(2)/U238t);
HU235m4 = Um4*U235*H235U*exp(t*log(2)/U235t);
HTh232m4 = Thm4*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m4 = Km4*K40*H40K*exp(t*log(2)/K40t);

HU238m3 = Um3*U238*H238U*exp(t*log(2)/U238t);
HU235m3 = Um3*U235*H235U*exp(t*log(2)/U235t);
HTh232m3 = Thm3*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m3 = Km3*K40*H40K*exp(t*log(2)/K40t);

HU238m2 = Um2*U238*H238U*exp(t*log(2)/U238t);
HU235m2 = Um2*U235*H235U*exp(t*log(2)/U235t);
HTh232m2 = Thm2*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m2 = Km2*K40*H40K*exp(t*log(2)/K40t);

HU238m1 = Um1*U238*H238U*exp(t*log(2)/U238t);
HU235m1 = Um1*U235*H235U*exp(t*log(2)/U235t);
HTh232m1 = Thm1*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m1 = Km1*K40*H40K*exp(t*log(2)/K40t);

HU238m0 = Um0*U238*H238U*exp(t*log(2)/U238t);
HU235m0 = Um0*U235*H235U*exp(t*log(2)/U235t);
HTh232m0 = Thm0*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m0 = Km0*K40*H40K*exp(t*log(2)/K40t);

HU238m = Um*U238*H238U*exp(t*log(2)/U238t);
HU235m = Um*U235*H235U*exp(t*log(2)/U235t);
HTh232m = Thm*Th232*H232Th*exp(t*log(2)/Th232t);
HK40m = Km*K40*H40K*exp(t*log(2)/K40t);

Atotc0 = (HU238c0+HU235c0+HTh232c0+HK40c0)*rhoc; %volumetric heat
production crust (W/m3)
Atotc1 = (HU238c1+HU235c1+HTh232c1+HK40c1)*rhoc; %volumetric heat
production crust (W/m3)
Atotc2 = (HU238c2+HU235c2+HTh232c2+HK40c2)*rhoc; %volumetric heat
production crust (W/m3)
Atotc3 = (HU238c3+HU235c3+HTh232c3+HK40c3)*rhoc; %volumetric heat
production crust (W/m3)
Atotc4 = (HU238c4+HU235c4+HTh232c4+HK40c4)*rhoc; %volumetric heat
production crust (W/m3)
Atotm4 = (HU238m4+HU235m4+HTh232m4+HK40m4)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm3 = (HU238m3+HU235m3+HTh232m3+HK40m3)*rhoml; %volumetric heat
production mantle litho (W/m3)
```

```
Atotm2 = (HU238m2+HU235m2+HTh232m2+HK40m2)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm1 = (HU238m1+HU235m1+HTh232m1+HK40m1)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm0 = (HU238m0+HU235m0+HTh232m1+HK40m0)*rhoml; %volumetric heat
production mantle litho (W/m3)
Atotm = (HU238m+HU235m+HTh232m+HK40m)*rhom; %volumetric heat production
mantle (W/m3)

%Calculate the layer thicknesses
lay5c = 20*(F1/100); %km
vol5c = (4/3)*pi*(rad^3-(rad-lay5c)^3); %km^3 volume of crustal layer
mass5c = vol5c*(1000^3)*rhoc; %kg mass crustal layer

lay4c = (rad-lay5c)-((rad-lay5c)^3-vol4c/((4/3)*pi))^(1/3); %km

lay3c = (rad-lay5c-lay4c)-((rad-lay5c-lay4c)^3-vol3c/((4/3)*pi))^(1/3);
%km

lay2c = (rad-lay5c-lay4c-lay3c)-((rad-lay5c-lay4c-lay3c)^3-
vol2c/((4/3)*pi))^(1/3); %km

lay1c = (rad-lay5c-lay4c-lay3c-lay2c)-((rad-lay5c-lay4c-lay3c-lay2c)^3-
vol1c/((4/3)*pi))^(1/3); %km

lay1m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c)-((rad-lay5c-lay4c-lay3c-
lay2c-lay1c)^3-vol1m/((4/3)*pi))^(1/3); %km

lay2m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m)-((rad-lay5c-lay4c-
lay3c-lay2c-lay1c-lay1m)^3-vol2m/((4/3)*pi))^(1/3); %km

lay3m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m)-((rad-lay5c-
lay4c-lay3c-lay2c-lay1c-lay1m-lay2m)^3-vol3m/((4/3)*pi))^(1/3); %km

lay4m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)-((rad-
lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m)^3-
vol4m/((4/3)*pi))^(1/3); %km

mass5m = mass5c/(F4/100)-mass5c; %kg mass lithosphere layer
vol5m = mass5m/((1000^3)*rhoml); %km^3 volume lithosphere layer
lay5m = (rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m-lay4m)-
((rad-lay5c-lay4c-lay3c-lay2c-lay1c-lay1m-lay2m-lay3m-lay4m)^3-
vol5m/((4/3)*pi))^(1/3); %km

%Calculate base (depth) of each layer
cth0 = lay5c+lay4c+lay3c+lay2c+lay1c; %total thickness of crust in km
cmth0 = cth0+lay1m+lay2m+lay3m+lay4m+lay5m;

A0c =
Atotc4*lay1c/cth0+Atotc3*lay2c/cth0+Atotc2*lay3c/cth0+Atotc1*lay4c/cth0
+Atotc0*lay5c/cth0;
A0m = Atotm4*lay1m/(cmth0-cth0)+Atotm3*lay2m/(cmth0-
cth0)+Atotm2*lay3m/(cmth0-cth0)+Atotm1*lay4m/(cmth0-
cth0)+Atotm0*lay5m/(cmth0-cth0);
```

```
Am = Atotm;
%% Calculate areotherm 0 Ga
c = zeros(d,33);
c1 = zeros(d,2);

%Fill in the initial row of the array so that the loops can build the
rest
c(1,1) = 0; %Initial Pressure (GPa)
c(1,2) = 0; %Initial Depth (km)
c(1,3) = Tad0+273;
c(1,4) = (-4.877*c(1,1)^2+120.2*c(1,1)+1088)+273.15; %solidus
c(1,5) = Q00/1000; %This is the surface heat flow (W/m^2)
c(1,7) = T0;

% Now that row 1 is filled out, it is time to fill in the remaining
columns
for i=2:d
   c(i,2)=c(i-1,2)+step; %km
   if c(i,2)<cth0
       c(i,1)=(rhoc*g*c(i,2)*1000)/10^9;
   else
       c(i,1)=(rhoc*g*cth0*1000)/10^9+(rhoml*g*((c(i,2)-
cth0)*1000))/10^9;
   end

   %These are the adiabat and solidus
   c(i,3)=slope*c(i,2)+c(1,3);
   c(i,4)=(-4.877*c(i,1)^2+120.2*c(i,1)+1088)+273.15; %solidus

   %This is the q value
   if c(i,2)<cth0
       q=c(i-1,5)-A0c*(c(i,2)-c(i-1,2))*1000;
   elseif c(i,2)>cth0 && c(i,2)<cmth0
       q=c(i-1,5)-A0c*(c(i,2)-c(i-1,2))*1000;
   else
       q=c(i-1,5)-Am*(c(i,2)-c(i-1,2))*1000;
   end
   c(i,5)=q; %Store the q value in the array

   %Step 1: Calculate initial guess k values
   if c(i,2)<cth0
       k=kc;
   else
       k=klth;
   end
   c(i,6)=k; %Put this k value into the array

   %Step 2: Calculate the temperature at this z value
   %T(z)=T0+qT*deltaz/k-A*deltaz^2/2*k
   if c(i,2)<cth0
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A0c*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   elseif c(i,2)>cth0 && c(i,2)<cmth0
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-A0m*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   else
```

```
       T=c(i-1,7)+c(i,5)*((c(i,2)-c(i-1,2))*1000)/c(i,6)-Am*((c(i,2)-
c(i-1,2))*1000)^2/(2*c(i,6));
   end
   c(i,7)=T;

   %Steps 5 and 6 must be iterated 10 times; use a for loop
   for j=8:2:33
       %Step 5: Calculate k using most recent temperature calculation
       if c(i,2)<cth0
           k=kc*(1+0.0015*c(i,2))/(1+0.0001*(c(i,j-1)-273));
       elseif c(i,2)>cth0
           if c(i,j-1)<500
               k=1/(0.0741+0.000502*c(i,j-1));
           else %Take the larger of these values
               k1=1/(0.0741+0.000502*c(i,j-1))+0.0023*(c(i,j-1)-500);
               %or
               k2=1.26*(1+c(i,2)/1000)+0.0023*(c(i,j-1)-500);
               if k1>k2
                   k=k1;
               else
                   k=k2;
               end
           end
       end
       c(i,j)=k; %Save the applicable k in the array

       c(1,j+1)=T0; %set T on first row

       %Step 6: Calculate the temperature using the most recent k
       if c(i,2)<cth0
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A0c*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       elseif c(i,2)>cth0 && c(i,2)<cmth0
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
A0m*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       else
           T=c(i-1,j+1)+c(i,5)*(c(i,2)-c(i-1,2))*1000/c(i,j)-
Am*((c(i,2)-c(i-1,2))*1000)^2/(2*c(i,j));
       end
       c(i,j+1)=T; %Stores the values for T in the array
   end %Ends the Step 5 and 6 loop
   c1(i,1)=c(i,1)*10000;
   c1(1,2)=c(1,33);
   c1(i,2)=c(i,33);
   A0(i,1)=c(i,33)-c(i,3);
   B0(i,1)=c(i,1);
   C0(i,1)=c(i,2);
   PFp0 = find(A0>0,1,'first');
   PFm0 = find(A0<0,1,'last');
   PF0 = (B0(PFp0)+B0(PFm0))/2; %calulate pressure
   dF0 = (C0(PFp0)+C0(PFm0))/2; %calculate crusth+lithosphere thickness
end %Ends the for loop that fills in the rest of the array
%% Calculate F, melt and residual compositions, and error
w(1,5)=cth0;
w(2,5)=Q00;
w(3,5)=Tad0;
```

```
syms P0
eqn1 = slope2*P0+(Tad0+273) == (-4.877*P0^2+120.2*P0+1088)+273.15;
P00 = min(vpasolve(eqn1,P0));
P00 = double(P00); %pressure where solidus and adiabat intersect
w(4,5)=P00;
w(5,5)=PF0;
w(6,5)=dF0;

dTdPsolidus0 = 2*(-4.877*PF0)+120.2; %slope of solidus Duncan et al.
(2018)
dTdF0 = dTdFa*PF0+dTdFb;
w(7,5)=dTdPsolidus0;
w(8,5)=dTdF0;

% Use P0 and PF to calculate F (percentage of melt produced)
F0 = ((dTdPsolidus0-slope2)/((deltaHF/C_P)+dTdF0))*(P00-PF0)*100;
w(9,5)=F0;

% Calculate the composition of the melt and error
SiO20 = (SiO2A*PF0+SiO2B)*F0^2+(SiO2C*PF0+SiO2D)*F0+(SiO2E*PF0+SiO2F);
TiO20 = TiO2A*F0^TiO2B;
Al2O30 =
(Al2O3A*PF0+Al2O3B)*F0^2+(Al2O3C*PF0+Al2O3D)*F0+(Al2O3E*PF0+Al2O3F);
Cr2O30 =
(Cr2O3A*PF0+Cr2O3B)*F0^2+(Cr2O3C*PF0+Cr2O3D)*F0+(Cr2O3E*PF0+Cr2O3F);
FeO0 = (FeOA*PF0+FeOB)*F0^2+(FeOC*PF0+FeOD)*F0+(FeOE*PF0+FeOF);
MnO0 = (MnOA*PF0+MnOB)*F0^2+(MnOC*PF0+MnOD)*F0+(MnOE*PF0+MnOF);
MgO0 = (MgOA*PF0+MgOB)*F0^2+(MgOC*PF0+MgOD)*F0+(MgOE*PF0+MgOF);
CaO0 = CaOA*F0^3+CaOB*F0^2+CaOC*F0+CaOD;
Na2O0 = (Na2OA*PF0+Na2OB)*F0^2+(Na2OC*PF0+Na2OD)*F0+(Na2OE*PF0+Na2OF);
K2O0 = K2OA*F0^K2OB;
P2O50 = P2O5A*F0^P2O5B;
w(10,5)=SiO20;
w(11,5)=TiO20;
w(12,5)=Al2O30;
w(13,5)=Cr2O30;
w(14,5)=FeO0;
w(15,5)=MnO0;
w(16,5)=MgO0;
w(17,5)=CaO0;
w(18,5)=Na2O0;
w(19,5)=K2O0;
w(20,5)=P2O50;

% Calculate the residual using batch melting-will become mantle lithos
SiO2res0 = (mantleSiO2-((F0/100)*SiO20))/(1-(F0/100));
TiO2res0 = (mantleTiO2-((F0/100)*TiO20))/(1-(F0/100));
Al2O3res0 = (mantleAl2O3-((F0/100)*Al2O30))/(1-(F0/100));
Cr2O3res0 = (mantleCr2O3-((F0/100)*Cr2O30))/(1-(F0/100));
FeOres0 = (mantleFeO-((F0/100)*FeO0))/(1-(F0/100));
MnOres0 = (mantleMnO-((F0/100)*MnO0))/(1-(F0/100));
MgOres0 = (mantleMgO-((F0/100)*MgO0))/(1-(F0/100));
CaOres0 = (mantleCaO-((F0/100)*CaO0))/(1-(F0/100));
Na2Ores0 = (mantleNa2O-((F0/100)*Na2O0))/(1-(F0/100));
K2Ores0 = (mantleK2O-((F0/100)*K2O0))/(1-(F0/100));
```

```
P2O5res0 = (mantleP2O5-((F0/100)*P2O50))/(1-(F0/100));
w(21,5)=SiO2res0;
w(22,5)=TiO2res0;
w(23,5)=Al2O3res0;
w(24,5)=Cr2O3res0;
w(25,5)=FeOres0;
w(26,5)=MnOres0;
w(27,5)=MgOres0;
w(28,5)=CaOres0;
w(29,5)=Na2Ores0;
w(30,5)=K2Ores0;
w(31,5)=P2O5res0;
%End of present day calculations


%% Now plot the results.
%The areotherms and adiabats are plotted against depth on the left
axis, while the solidus is plotted against pressure on the right axis
vt1 = 1/255*[232,119,34];
str1 = '#001932'; %'#000032';
color1 = sscanf(str1(2:end),'%2x%2x%2x',[1 3])/255;
str2 = '#00366C'; %'#000096';
color2 = sscanf(str2(2:end),'%2x%2x%2x',[1 3])/255;
str3 = '#0068D0'; %'#0000FA';
color3 = sscanf(str3(2:end),'%2x%2x%2x',[1 3])/255;
str4 = '#359AFF'; %'#5F5FFF';
color4 = sscanf(str4(2:end),'%2x%2x%2x',[1 3])/255;
str5 = '#99CCFF';
color5 = sscanf(str5(2:end),'%2x%2x%2x',[1 3])/255;

yyaxis left
plot(x(:,33),x(:,2),'-','color',color1,'LineWidth',2,'DisplayName','4
Ga') %areotherm
hold on
plot (y(:,33),y(:,2),'-.','color',color2,'LineWidth',2,'DisplayName','3
Ga')
hold on
plot (z(:,33),z(:,2),'--','color',color3,'LineWidth',2,'DisplayName','2
Ga')
hold on
plot (b(:,33),b(:,2),':','color',color4,'LineWidth',2,'DisplayName','1
Ga')
hold on
plot (c(:,33),c(:,2),'-','color',color5,'LineWidth',2,'DisplayName','0
Ga')
hold on
plot (x(:,3),x(:,2),'-
','color',color1,'LineWidth',1.5,'DisplayName','adiabat') %adiabat
hold on
plot (y(:,3),y(:,2),'-
.','color',color2,'LineWidth',1.5,'DisplayName','adiabat')
hold on
plot (z(:,3),z(:,2),'--
','color',color3,'LineWidth',1.5,'DisplayName','adiabat')
hold on
```

```
plot
(b(:,3),b(:,2),':','color',color4,'LineWidth',1.5,'DisplayName','adiaba
t')
hold on
plot (c(:,3),c(:,2),'-
','color',color5,'LineWidth',1.5,'DisplayName','adiabat')
legend('location','southwest');

ylabel('Depth (km)')
ylim([0 x(3000,2)])
set(gca,'YDir','reverse','YMinorTick','on','Ycolor','k')

title('Areotherms Average Melt')
xlabel('Temperature (K)')
set(gca,'XAxisLocation','top','linestyleorder',{'-
'},'XMinorTick','on','Xcolor','k')
xlim([200 2000])

yyaxis right
plot(x(:,4),x(:,1),'color',vt1,'LineWidth',2,'DisplayName','solidus')
%solidus
ylim([0 x(3000,1)])
ylabel('Pressure (GPa)')
set(gca,'YDir','reverse','YMinorTick','on','Ycolor','k')

set(gca,'TickDir','both')
%% Save the results
save Areotherm4GaAvemelt.txt x -ascii
save Areotherm3GaAvemelt.txt y -ascii
save Areotherm2GaAvemelt.txt z -ascii
save Areotherm1GaAvemelt.txt b -ascii
save Areotherm0GaAvemelt.txt c -ascii
save pt0Avemelt.txt c1 -ascii
save pt1Avemelt.txt b1 -ascii
save pt2Avemelt.txt z1 -ascii
save pt3Avemelt.txt y1 -ascii
save pt4Avemelt.txt x1 -ascii
```