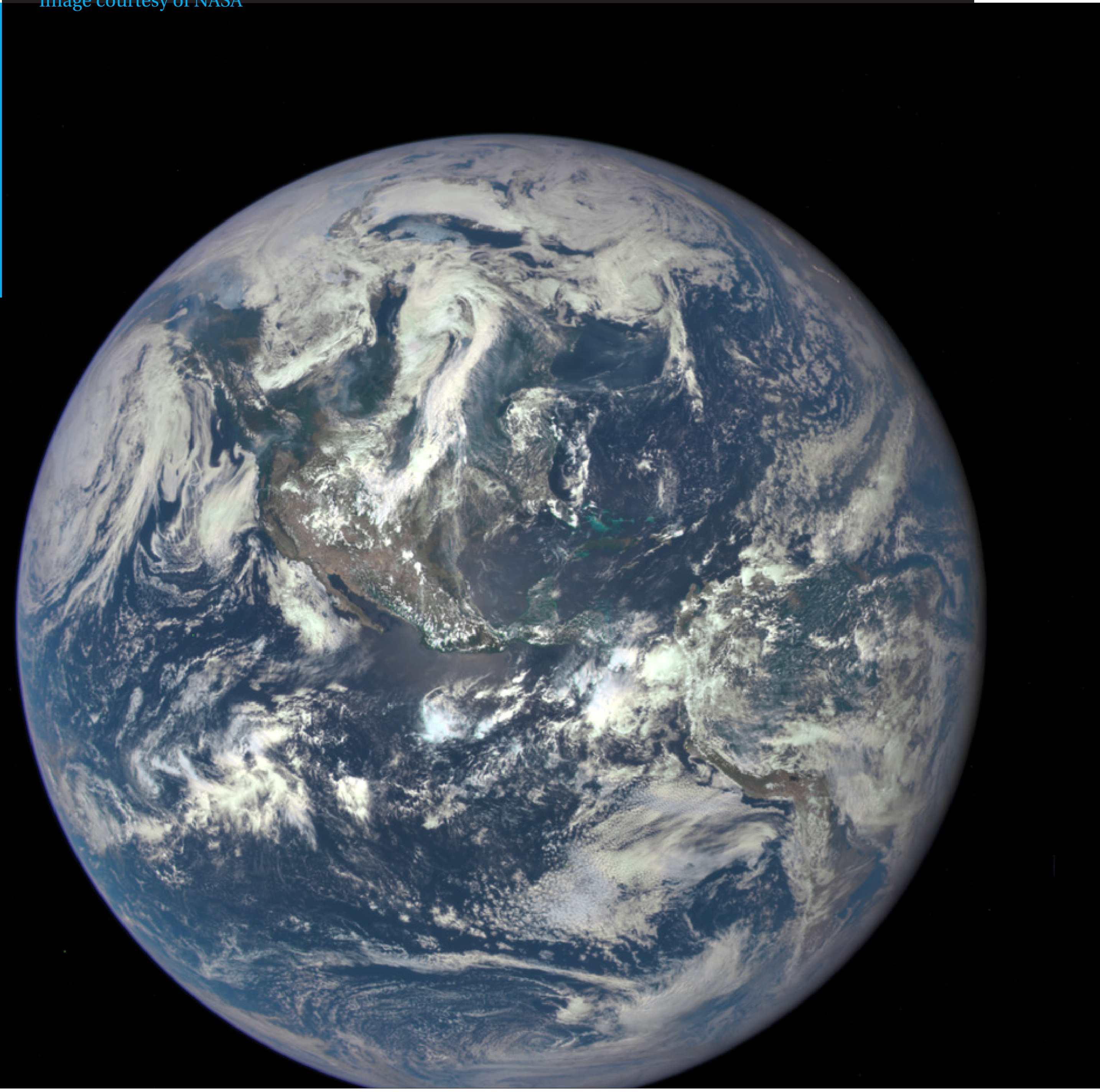
Influence of the combination of ambient and glacial stresses on the response of a visco-elastic Earth

Master Thesis in Aerospace Engineering

Fabrizio Morra	4934539
Date of Delivery	February 4, 2021
Image courtesy of NASA	

Delft University of Technology



TUDelft Delft University of Technology

Challenge the future

Preface

My decision to start this research project has been quite long and complex, just like the thesis project itself. I still remember a fairly large number of visits to the 9th floor of the Aerospace building to talk to every single one of the possible thesis supervisors, a number still smaller than the hours spent deciding what unknown developments to undertake by choosing my Master Thesis project.

Because of its length and complexity and the unfortunate coincidence of the Coronavirus pandemic happening with all its social consequences, the weight of this thesis work on my shoulders felt even heavier. This makes the support and encouragement, or just the willingness to listen to my problems, received during this period all the more precious. A gentle summer rain washing sadness, doubts and regrets away and bringing images and impressions of serenity and warmth.

I would first of all like to thank my supervisor Wouter van der Wal for being very understanding and kind when I felt completely lost in my own project and overwhelmed or confused by the tasks I had to carry out and the steps I had to take, always helping with his technical expertise or empathetic insight. I honestly did not expect such help and compassion and I am very grateful to him for being close to what I have in my mind as the image of the ideal teacher.

I also thank Bas Blank for the initial help given to me when it came to understanding how ABAQUS and the model worked, since it would have been much harder without him, and how to solve many of the initial problems related to learning this new software.

I want to thank my family for the support shown throughout the thesis and especially when I was living especially dire moments. Most importantly, I feel grateful for them (almost) always trusting my judgement and allowing me to prioritize carrying out my thesis work over their need for my help. Honorable mention goes to my sister and aunt Alessia and Angela who always reminded me of how stable and stalwart an adult worthy of this name should be in the hardships or difficult moments of life and how living a thousand stories can develop and shape a single individual by making them theirs. I also want to thank my cousin Eleonora for all the tips given to me about how to work on research projects and sharing with me her PhD experiences, both at its high and low points, allowing me to always keep my hope in the future unwavering and never lose my focus even when I was feeling completely down.

Antonino, not a hundred books could describe what your friendship has been to me than a simple sentence, and that is I would not be the person that I am today without you and I will hopefully be what I want to be in the future. You have always been an example of what I should strive to be, and of a rare blend of solid composure and open friendliness. Just be careful not to become 80 when you're half that age.

Federico, you might not realize but you been an incredible example of how to put things in perspective and not surrender your entire self to the impression the others might have of you. Your insight and decision to leave home first, to me unexpected as I admit to have been one of those, long ago, who also was misguided by appearances and the others' judgement, have also inspired me to leave my safe haven in Naples and come to Delft. And by the way, I'm waiting for you here, I am sure you would enjoy the green green grass of the Netherlands...

Stefano, your steadfast calm and gentle but resolute demeanor, be it when arguing with someone or confidently, as only a few people would be able to do, looking ahead at the future, are a rare anchor of stability and a safe haven in the storm of events that the external world has become, recently especially, and the internal worlds of some have always been. And you know what would be even more rare? You actually taking a stance when deciding what programs to make for the night.

Andrea, your experience of Erasmus abroad back when in the bachelor demonstrated that the world can be as open as the mind of those who decide to expand their horizons. If what Federico did was the first step, your decision was in a way what made me walk the path of studying abroad at take the plunge in an open sea. The open sea you never wanted to swim with me in in summer because you were too lazy to do so.

I have always admired Stefano's ability to always and sometimes conceal his profound insight and vast culture behind a lighthearted facade, which always make him a prime choice for educational but never too serious conversation. Such historical knowledge would make Alberto Angela and all of us Italian inheritors of a long past legacy proud. Speaking of which, your crusade for the demonstration of the Italian cultural supremacy in its highest vessel, the bidet, is meeting some resistances but I am confident in our victory.

Andrea, also known as Pino (Pine tree), constantly discussing our own insecurities and weaknesses has been a lifesaver through mutual help, as it taught me how fulfilling it is to bring peace to the others' inner conflicts and how selfish it is when I complain too much about my own issues. On the other hand, the fun moments we shared together will also never be forgotten, and I am sure the memory of this will carry me through difficult times. Exactly like the ones you had when trying to stay awake during game nights.

I also need to thank the rest of my group of friends from Naples. Andrea, Federica, Antonio, Francesca, William, thanks for all the fun shared together either remotely or in person when I could go back home before everything went to hell with the pandemic.

Simone e Piergiorgio, I am grateful for our enduring friendship throughout the years (almost as long as Simone's bachelor and the time Piergiorgio takes to get ready when going out) during and after high school and for the possibility of whipping out all my pure and unadulterated cynicism when spending time with both of you. It is always nice to share a radically different view of the world once in a while.

Simon and Rose, I don't think i would have made it that far here in this country without you. Before coming to the Netherlands, I honestly had a series of stereotypes about Dutch people which actually made me feel uneasy but you tore almost every single one of them down. I need to thank you for helping me understand how things work in the Netherlands and for the many different tips about studying at a university here. You have been a constant and comforting presence here, always quick to make me feel at peace again. I would say as quickly as Simon manages to end conversations.

I am grateful for having been able to maintain the friendship with Alex and Shashank after the introduction program, which made me feel a lot less alone especially during my first year in the Netherlands and showed me how unexpectedly long-lasting some bonds can be. Almost long lasting as our barbecues together when the famously idyllic Dutch weather allowed them.

Amaia, the soup sous chef, the discount connoisseur, the money saver extraordinaire. It is rare to always remain unfazed by almost anything (except when people accuse you of hiding puzzle pieces) and just as rare to always have faith that things are going to work out in the end. This enriching example which I am still trying to make, for whatever measure, part of me and the fact that you listened to countless complaints (almost as many as the euros I saved thanks to your tips) about anything really makes me cherish our friendship.

Benedetta has been for me an example of unrelenting ambition, I think she is one of the people who better represent the motto "Sky is the limit". Never faltering when you have a clear goal in mind, and pushing to get there is one of the main ingredients for success, especially when you manage to propagate your own drive to those who are around you. For the cynical ones, luck can only do that much when you do not know how to take advantage of it. I suppose that drive is what kept your soul from leaving the body after our training sessions in the park.

Georgios, the living proof that age is not just a number, but also a wealth of life experience to share with us little kids, showed me with his point of view how little I knew and how large was my presumption of knowing about the variety of experiences and hardships someone undergoes, and the wisdom one can derive from them.

Camera, despite not having had the possibility of interacting a lot, I am thankful for the feeling I have that you were the only one I could have conversations about some more complex or boring topics with with and to whom their messages would not be lost or spun around changing their meaning like leaves in the wind. Your resolute and always composed attitude when dealing with something you perceive, most of the time, as negative and disrespectful is the quality of a noble. And this behaviour is what I emulated when I tried your mac n cheese.

Sofie showed me how being a very empathetic person can brighten up others' days and always, no matter what, be more or less optimistic about what happened, happens and might happen. She is also a constant reminder of how not to make lasagna.

Last but not least, I would like to express my admiration for a true champion of unwavering trustworthiness, the elevator of the Aerospace Engineering faculty that never failed me when going to the 9th floor offices, allowing me to always look presentable.

Abstract

The evolution of ice loads during the last Ice Age causes an ongoing Earth response with deformations and stress fields. The viscosity of the mantle depends on stresses, therefore the model stresses should be combined with ambient stresses such as due to mantle convection. This combination has been successfully simulated in the FEM software ABAQUS.

At each timestep background stresses are added to existing component and the Mises stress is recalculated to be used in a custom creep law. This leads to a new viscosity leading to different stress components, which is solved an iterative procedure.

This work shows that the initial model provided can be adapted to simulate this iterative process with a loop inside each timestep of the computation. Results show that stresses increase with time while deformations and viscosity decrease, an opposite trend compared to a non-ambient stresses case.

Contents

List of abbreviations

The list of physical quantities and other abbreviations will be presented in this chapter. The physical quantities are presented first, in the list below.

1. Pa - Pascal
2. MPa - Mega - Pascal, 1e6 Pascals

1. GIA - Glacial Isostatic Adjustment
2. FEM - Finite Element Method
3. PDE - Python Development Environment
4. RSL - Relative Sea Level
5. CMB - Core-Mantle Boundary
6. CoG - Center of Gravity
7. ENU - East, North, Up
8. AOI - Area Of Interest

List of Figures

LIST OF TABLES

1
Introduction

1.1. Background

Many planetary phenomena are constantly happening all over the Earth, with some of them sudden, violent and quick such as volcanic eruptions while others cannot be appreciated as easily and happen over geological timescales. Examples of these are the motion of a plate with its small velocity as shown by Bouin and Vigny (2000) and uplifts of some areas of the Earth such as in Scandinavia as described for example by Pohjola et al. (2018). Planetary sciences studies exist to try to shed light into these events so that it is possible to either defend ourselves from them and avoid widespread destruction/loss of lives or derive information about Earth properties and dynamics which cannot be observed directly. The aim of this work belongs to the latter category, and more specifically to investigating a phenomenon that can be directly and indirectly studied but on timescales in the order of centuries, and that is the response of our planet to the disappearance of the ice load during some of the glacial cycles of the last Ice Age. This response is called GIA (Glacial Isostatic Adjustment) and many models have been developed to simulate it over the past few decades, going through different iterations as more and more precise data became available in different areas of the globe such as what happened for the GIA model developed by Peltier et al. (2015), which is part of the last in a long series. One of the main divisions between ice models can be done by considering what area they are tailored to study: some are global, while others are specifically developed, for example, for Fennoscandia or Antarctica. An example of the former is the ICE-6G model cited before, while examples of the latter are the W12 Antarctica model by Whitehouse et al. (2012a) and Whitehouse et al. (2012b) and the Iceland model proposed by Schmidt et al. (2012).

GIA models try to infer the composition of the Earth's interior and the development and evolution of other quantities which are not directly observable, such as stresses, by simulating the response of a viscoelastic Earth over time with a varying ice load based on a supplied ice history, which usually changes from model to model. While the response of a viscoelastic Earth causes the generation of uplifts in some areas of the globe such as Fennoscandia as for example shown in the review paper by Steffen and Wu (2011), it also generates stresses in the Earth's interior, which can only be evaluated indirectly. The only stresses which have been directly observed have been crustal stresses as reported by Zoback (1992) for the creation of the World Stress Project, which used many different techniques for stress study. These have been the observation of earthquake focal mechanisms (meaning the deformation type in its source region), fault slip data and borehole breakouts among others (Zoback (1992)). Conversely, while it is known that earthquakes also originate at lower depths such as in the mantle (Karato et al. (2001)) especially due to plate subduction, the observation techniques described by Zoback (1992) cannot be applied to them and these background stresses must mainly be derived through modelling such as in the case of the study by Osei Tutu et al. (2018).

Between observable stresses such as the ones studied in the work by Zoback (1992) and the observable deformations due to the removal of the ice load, the latter are especially important as they can be compared to GNSS data reading the surface deformations in determinate areas for the verification of GIA models. The GIA models usually use parametric studies which vary a series of inputs, the parameters, such as the grain size to see which set(s) of inputs best fits the observable response of the viscoelastic Earth. These GIA models are what constitutes one of the bases of this thesis work and will be used for the model simulation. Because the GIA model is already a verified component (as described by Blank et al. (2020) of the base modelling process to build upon, there will be no need to conduct any GIA validity and quality studies as the deformation comparison cited before.

However, despite how difficult it is to simulate and study Earth stresses, this endeavor is a crucial one, as their buildup in a certain area of the crust is for example what causes earthquakes, which are generated by their sudden release. For this reason, global scope projects such as the aforementioned World Stress Map Project by Zoback (1992), aimed at producing a global map of first (meaning dominated by large scale phenomena such as compression at plate boundaries) and second order stresses (which can be interpreted as local perturbation, as explained shortly by Zoback (1992)) over the Earth's crust, and local ones such as a stress map of Italy Montone et al. (2004) have been conducted.

Unfortunately, not much literature exists on the stresses acting in the mantle, especially its lower part, as they are complicated to simulate and quantify because of the lack of data concerning them. However, these mantle stresses cannot be ignored as shown by the influence they have on the crustal stress state as demonstrated by Osei Tutu et al. (2018) and also, regarding GIA, when considering the response of a viscoelastic Earth to an evolving ice load. Combining ambient stress components with present stresses due to ice load instead of working with a Mises stress that already includes the combination of different stresses (like in previous studies such as the one by Schmeling (1987)) leads in fact to an iterative procedure in the FEM software, because adding ambient components means modifying the Mises stress which must be recalculated. The new Mises stress causes a change in viscosity since it is dependent on the Mises stress as discussed in van der Wal et al. (2013) and demonstrated in Equation 1.2 based on the same study, which in turn modifies the stresses generated by the Earth as a response to the ice load at that timestep. These new stresses coming from the model are again combined with background stresses in the same timestep, leading to a new stress iteration, which means changing the viscosity again. While past studies about the interaction between background and glacial stresses exist, they were focused on different dynamics and not on the iterative process generated from the stress (and therefore viscosity) change. A short review of some of them will be presented here.

The study by Schmeling (1987) is one of the first investigations carried out on the interaction between a stress deriving from ice load and an ambient stress due to mantle processes. These two processes generate two different flows whose magnitudes are varied to investigate their effect on the viscosity. However, this study does not make use of a FEM method and its main scope has been the attribution to the mantle of a Newtonian or non-Newtonian rheology, which means that no iterative process due to stress changes has been taken into account. The study demonstrates that a case such as the one for superposition of convection and glacial stresses leads to the convective flow being enhanced or retarded in rebound areas. As will be described in Chapter 4 and Chapter 5, an enhancement of stresses in the rebound areas where the ice load is applied is exactly what is observed when stresses are coupled.

The study by Wu (1992) is the first one to attempt the study of a viscoelastic Earth response to a box car load using power law rheology (the one used for this thesis) using a finite element model to have a more rigorous solution to the problem. The results show that a power law rheology does not seem to be "seen" by the rebound data used for verification, and it is indeed shown that variations in creep parameters do not seem to influence the Earth response, especially as time goes on. An example of this is shown in Figure 1.1 and is one of the plots generated by Wu (1992) for his paper. Even though an instantaneous melting has been considered for the plot generation, it could be considered an approximation of the acceleration of the melting rate of the ice load over Antarctica. This is an important result which will be discussed in Chapter 5, so even though the scope of the paper has been pretty different from this thesis, it is knowledge to build upon.

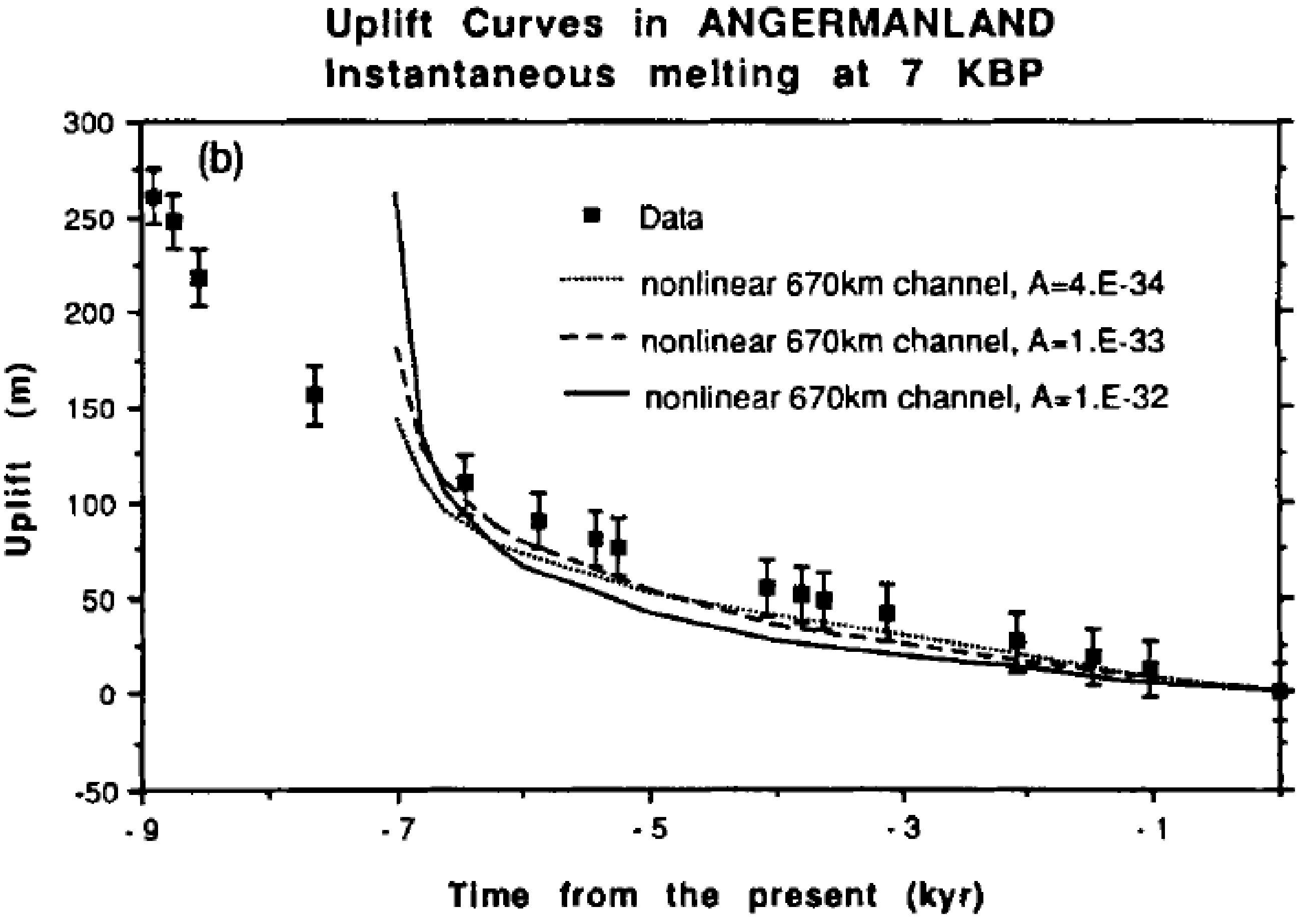


Figure 1.1: Comparison of uplift data in Angermanland to uplift with varying creep parameters over time from the study by Wu (1992).

The study by Wu (2001) uses a finite element approach to simulate the Earth's response when undergoing ice loading together with tectonic background stresses and decide whether using a non-Newtonian mantle rheology can explain sea level observations in the Boston area. The conclusion of the paper is that this is possible if the background stress due to tectonic processes is 10 MPa, while ambient stresses that are too small (smaller than 1 MPa) or too large (larger than 100 Mpa) do not agree with sea level measurements. Again the scope of the paper corresponds to the study of a linear or non linear mantle rheology, with an emphasis on the possibility of proving the validity of a non-linear rheology. The viscoelastic iterative process due to stress change is not discussed. This gap between the presence of stress combination studies and the need for a more precise investigation on one of its implications, which is the effect of this combination on the viscoelastic Earth response when stress components are combined instead of supposing a new Mises stress from the start, is the context in which this thesis work is proposed to help.

1.2. Research question

As explained before, combining mantle and ice load stresses in an ice history timestep means modifying the total Mises stress which is used to calculate creep law and viscosity, which follow the laws of Equation 1.1 and Equation 1.2 respectively. Both laws are the same as the ones developed for the study by van der Wal et al. (2013).

$$\epsilon = B_{diff} q \Delta t + B_{disl} q^n \Delta t \tag{1.1}$$

$$\eta_{eff} = \frac{q}{3B_{diff} q + 3B_{disl} q^n} \tag{1.2}$$

The formulation of B_{diff} and B_{disl} is based on the olivine flow law from the work by Hirth and Kohlstedf (2003) and follows Equation 1.3. The parameters p, r, A, E, V change based on a dry or wet rheology or diffusion or dislocation creep as explained by van der Wal et al. (2013), therefore both parameters come from the same law.

$$B = Ad^{-p} \exp A\phi f H_2O^r \exp\left(\frac{E + pV}{RT}\right) \tag{1.3}$$

In Equation 1.3, A is a constant, d the grain size, p the grain size exponent, fH_2O the water fugacity, r the water fugacity exponent, ϕ the melt fraction, α a constant, E and V the activation energy and volume respectively, P the pressure, R the gas constant and T the absolute temperature. In Equation 1.1, B_{diff} and B_{disl} are the diffusion and dislocation creep parameters respectively, q is the Von Mises stress and Δt is the time interval between timesteps. n is an exponent regulating the weight of the Mises stress, and in this case it is equal to 3.5 from the work of Hirth and Kohlstedf (2003). ϵ is the strain tensor component resulting from the creep development. In Equation 1.2, η_{eff} is the viscosity and q is the Mises stress while all the other parameters are the same as Equation 1.1. Discussing these equations clearly shows how changing the stresses by combining the background and ice load ones affects the viscosity, which in turn affects the ice load stresses at next step before being again combined with mantle stresses. This originates an iterative process in the FEM software, and because no one has ever attempted to do this before, any result can take place, ranging from an insensitive model to convergence after 2 stress iterations.

Based on the background presented in Section 1.1 and the information derived from Equation 1.1 and Equation 1.2, the research question serving as the base of this thesis is whether, how and in what measure, the combination of background mantle stresses and the ones deriving from the ice load influences the simulation results. The geographic area for this investigation has been chosen as Antarctica, with a focus on the West Antarctica area. The reason for this choice has mainly been the availability of a ready-made base GIA model described in Blank et al. (2020) and developed at TU Delft on which to build on in order to simulate the combination of glacial and ambient stresses. The research question can be further broken into:

1. How can an interaction between background and ice load stresses be simulated in a FEM software such as ABAQUS?
2. How does an interaction between the Earth background stresses and the ones coming from the ice load over West Antarctica influence the simulation outputs such as stresses, deformations and viscosity over the area of West Antarctica over time?
3. Does changing the background stresses and rheology also change the way these stresses influence the simulation output values for West Antarctica over time?

The methodology used to answer these research questions will be shortly summarized in Section 1.3 and then further elaborated on in Chapter 2.

1.3. Methodology

The basis of the work to obtain the answer to the research question of Section 1.2 is a series of Python scripts supplied by the Delft University of Technology to be used in the ABAQUS, which is a prominent FEM software developed by Dassault Systemes. These scripts build on the seminal work by Wu (2004), who first described how to tackle the problem of representing the non linearity of GIA modelling in commercial FEM software, and have been mainly developed by Bas Blank and Haiyang Yu of TU Delft. Almost all of the commands can be given to ABAQUS using Python scripting through its integrated PDE (Pyhton Development Environment), which is a Python interface for ABAQUS scripting, and this is how the 3D Earth model is generated and undergoes the simulation process. The only commands to not be supplied in Python are given in FORTRAN and they are the ones related to the creep law, which is custom-defined from Equation 1.1. The iterative simulation process is based on the recalculation of viscosity through this creep law which is implemented in FORTRAN as shown in Equation 1.4. The first equation uses the FORTRAN notation, while the second one uses the same symbols introduced in Section 1.1, to show their equivalence.

$$\begin{aligned} DECRA &= ALIN * QTILD * DTIME + A * QTILD * * AN * DTIME \\ \epsilon &= B_{diff} q \Delta t + B_{disl} q^n \Delta t \end{aligned} \tag{1.4}$$

In this equation, $DECRA$ is the strain tensor component resulting from the creep development, $ALIN$ is B_{diff}, $QTILD$ is the Mises stress coming from the model and is going to be changed when combining the glacial and background stresses, $DTIME$ the time interval, A is B_{disl}, AN the n exponential factor.

Both these groups of scripts have been modified to allow for the combination of stresses resulting from the model at each stress iteration inside a timestep with realistic background stress due to mantle convection, which is the most prominent large-scale mantle phenomenon. The stresses due to ice load are read from the model and background stresses are added to them, with the Mises stress being recalculated using Equation 1.5. This definition, taken from the work of Kim (2009), is displayed in Equation 1.5 and is an approximation of the procedure ABAQUS uses to calculate the Mises stress whose description can be found in the user's manual. In the equation, the components use the same notation as Equation 2.1 after the change to standard Cartesian while q is the Mises stress.

$$q = \sqrt{\frac{(S_{11} - S_{22})^2 + (S_{22} - S_{33})^2 + (S_{33} - S_{11})^2 + 6(S_{12}^2 + S_{23}^2 + S_{13}^2)}{2}} \tag{1.5}$$

This will cause the creep law and viscosity values to change inside a timestep because the Mises stress changes, so this variation will then cause the output data, with stresses among them, to change. These stresses are again read and combined with background stresses and a new model is created, generating new Mises stresses which in turn modify the viscosity for all elements. Again, the new viscosity causes a modification in the output stresses and an iterative process is started. Once convergence is reached for the coupled stresses the timestep is incremented and a new loop starts. The data produced at each stress iteration can then be reorganized and plotted with a framework devised for this work and then written in Python (for data organization) and MATLAB (for plotting). A summary of the stress coupling loop as it is implemented in ABAQUS is shown in Figure 1.2.

After this chapter, the thesis work will be presented as divided into several chapters. These are: the description of how data generation, processing and plotting has taken place in Chapter 2, the presentation and analysis of results in Chapter 4 and the conclusions stemming from those results, together with suggestions for future developments, in Chapter 5. Another chapter, Chapter 3, describes how it was made sure that the results displayed in the thesis work would have a proof of validity and some tools used for this case study were developed.

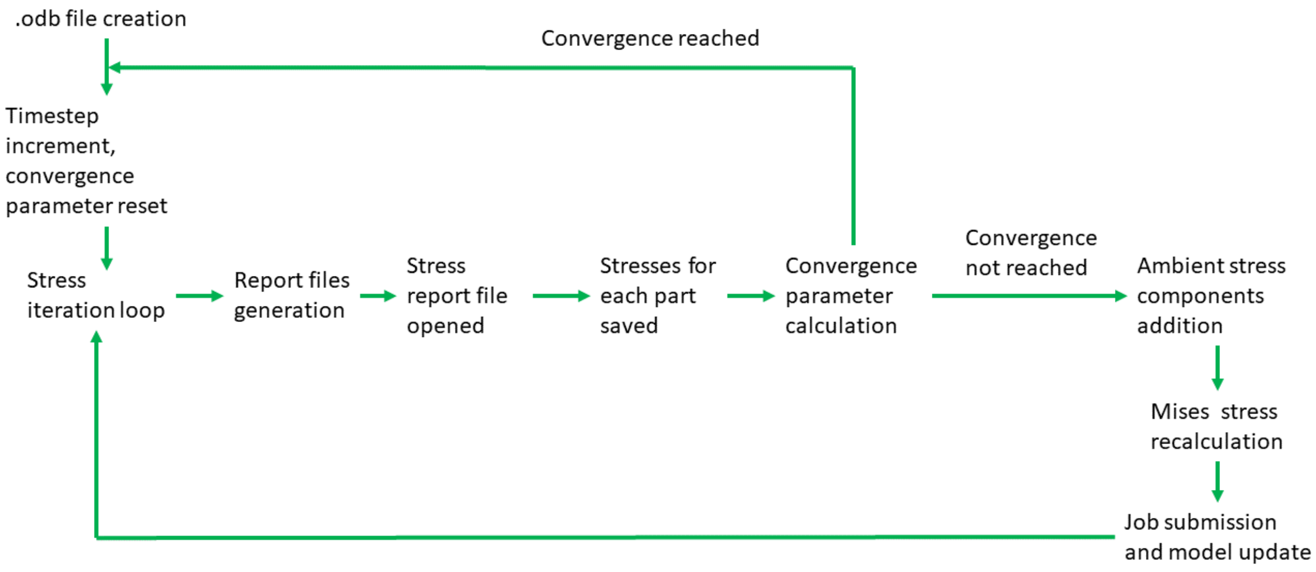

.odb file creation
Convergence reached
Timestep increment, convergence parameter reset
Stress iteration loop
Report files generation
Stress report file opened
Stresses for each part saved
Convergence parameter calculation
Convergence not reached
Ambient stress components addition
Mises stress recalculation
Job submission and model update

Figure 1.2: Scheme of the stress coupling loop implementation in ABAQUS for this thesis work.

2

METHODOLOGY

This chapter will present all the new tools that have been developed for this thesis and give all of their theoretical background where necessary. Section 2.1 describes all the operations related to the change in reference systems to be applied to the ABAQUS output data. Section 2.2 will give an overview of the data generation process through the model definition and iterative operations to be performed in ABAQUS, Section 2.3 will describe how the data coming out of ABAQUS has been organized and manipulated for better handling and plotting and finally Section 2.4 will describe the data plot generation procedure.

2.1. REFERENCE SYSTEM TRANSFORMATIONS

ABAQUS uses two custom-defined Cartesian references frames for this particular simulation, one to display the model in the GUI (Graphic User interface) and one to actually save coordinate and stress/deformation results. The former frame is a Cartesian reference frame whose axes are in the Z, X, Y order for the 1, 2, 3 axes instead of the standard Cartesian reference frame where the 1, 2, 3 axes are associated to X, Y, Z. This reference system with Z, X, Y for the 1, 2, 3 axes will from now on be referred to as display frame, so that it is going to be clear what its role is. The latter reference frame, more important for this work because it is one the data to save is in, has has its axes in the Y, Z, X order on the 1, 2, 3 directions. This reference frame will be called data frame. A conversion to the ENU coordinate system is necessary since the end goal of the thesis also consists of mapping data using latitude and longitude, but the coordinate transformation for vectors and tensors is defined for a X, Y, Z to lat, lon, R reference system change. Both vectors and tensors will need to be plotted since deformations are in vector form and stresses are in tensor form.

Therefore, two coordinate system changes have been necessary for this thesis work: one between the data and Cartesian reference frames and another between the Cartesian and ENU (East, North, Up) reference frame, where the ENU reference frame has been defined according to Grewal et al. (2007). The first reference system transformation does not involve any rotations and can basically be performed without introducing transformation matrices, only by "swapping" the data columns for coordinates, stresses or deformations. The sequence to swap the columns, which also means the original axis configuration, has been found through a procedure described in Chapter 3.

The first transformation (data to standard cartesian frame) needs to be carried out not only on the centroid or node coordinates, but also on the deformation vector and stress tensor components. Stresses and deformations are in fact calculated in two different places in the model, with the former in the mean position of all the nodes over all coordinates and the latter at the vertices of each of those elements, called nodes. This happens because the stress tensor is defined for an infinitesimal volume, while movements such as deformations are defined for points like the nodes, and the stress output for that volume can be reduced to a centroid. This difference is also important for the data processing procedure for stresses, since ABAQUS does not calculate the centroid coordinates but simply gives the nodes making up each centroid. A longer procedure involving the calculation of the centroid coordinates of each element has therefore to be carried out based on the coordinates of the nodes making it up.

This first transformation is necessary because changing the reference system obviously causes a change of vector components, but it also causes the tensor components to "move around", and for tensors this change is more difficult to visualize. Therefore, the tensor component change due to this first reference frame transformation is shown in Figure 2.1 for clarity, and it displays how the stress components have to be "swapped around" when the axes also move around.

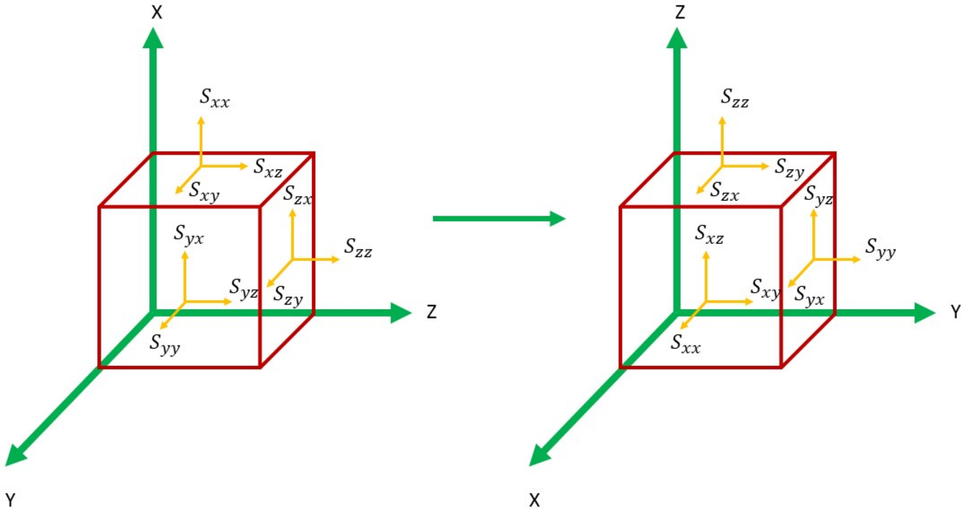


Figure 2.1: Sketch of the tensor components change between the ABAQUS data frame and the base 3D Cartesian reference system, to be applied consequently to the coordinate system coordinate change.

Such operation corresponds to moving around the data columns relative to node and centroid coordinates and corresponding deformation and stress components. This happens because the transformation takes place between two Cartesian systems with different axes definitions. If a change to a spherical or ENU reference systems, such as the second one to be applied for this thesis work, took place instead, the transformation would have needed to take place through multiplication with a transformation tensor. The reference system change of Figure 2.1 is equivalent to the change of tensor components from Equation 2.1 in tensor notation.

$$\begin{bmatrix} S_{yy} & S_{yz} & S_{yx} \\ S_{zy} & S_{zz} & S_{zx} \\ S_{xy} & S_{xz} & S_{xx} \end{bmatrix} \rightarrow \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} \tag{2.1}$$

These "column swaps" will be explained next for clarity. The node coordinates are swapped around in the following order, for Y, Z, X to X, Y, Z:

1. Y and X, columns 1 and 3 of the coordinate file;
2. Z and Y, columns 2 and 3 of the coordinate file.

This is the same order to be used for deformation components and centroid coordinates. Nodes and centroid are processed independently, the former when working on deflections and the latter on stresses, so the column swapping for nodes happens independently from the one for centroids even if they follow the same order. The stress components are swapped around in the following order:

1. S_{11} and S_{33}, columns 2 and 4 of the stress file;
2. S_{22} and S_{33}, columns 3 and 4 of the stress file;
3. S_{12} and S_{23}, columns 5 and 7 of the stress file;
4. S_{12} and S_{13}, columns 5 and 6 of the stress file.

After this preliminary transformation, the main one between Cartesian and ENU reference systems can be carried out. The ENU reference system used for this work is depicted in Figure 2.2, and based on the definition by Grewal et al. (2007), E is the longitudinal direction directed eastward, N is the latitudinal direction based towards the North pole and U is the vertical direction in the specified point P.

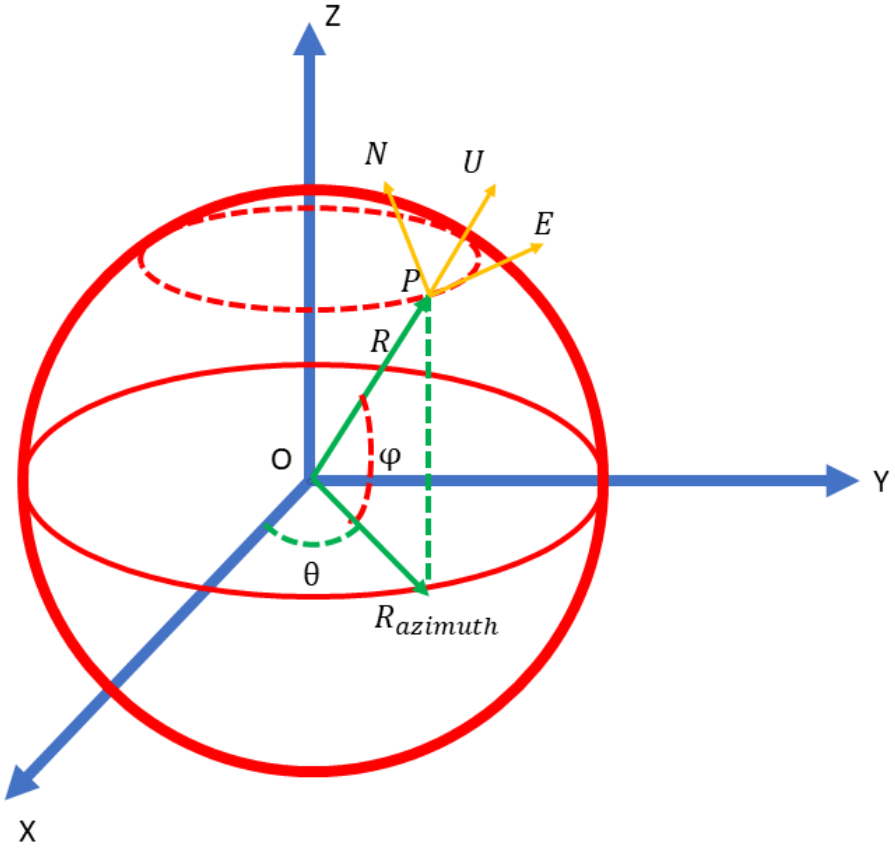


Figure 2.2: ENU reference system used in the thesis work to represent the Earth using latitude and longitude. θ is the longitude angle, ϕ the latitude, R the radius of a certain point P on the sphere and $R_{azimuth}$ the radius on the horizontal plane.

This reference system change requires the definition of latitude, longitude and radial distance starting from Cartesian coordinates. These relations are displayed in Equation 2.2. This series of relations, together with all the others presented until the the beginning of Section 4.5, is based on the equations contained in Grewal et al. (2007) unless specified otherwise.

$$\begin{aligned} R &= \sqrt{X^2+Y^2+Z^2} \\ lat &= \arcsin\left(\frac{Z}{R}\right)\frac{180}{\pi} \\ lon &= \arctan\left(\frac{Y}{X}\right)\frac{180}{\pi} \end{aligned} \tag{2.2}$$

The transformation angles corresponding to this reference system are shown in Equation 2.3, and their formulation can be readily derived from Figure 2.2.

$$\begin{aligned} R_{3D} &= \sqrt{X^2+Y^2+Z^2} \\ R_{azimuth} &= \sqrt{X^2+Y^2} \\ \cos\theta &= X/R_{azimuth} \\ \sin\theta &= Y/R_{azimuth} \\ \cos\phi &= R_{azimuth}/R_{3D} \\ \sin\phi &= Z/R_{3D} \end{aligned} \tag{2.3}$$

The next step after the definition of the transformation angles for this reference system transformation is the definition of the corresponding transformation tensor. Again, the work by Grewal et al. (2007) has been used as a reference for this task and the process will be displayed below. The explanation for the reference system change is better discussed when starting from how to change the reference system for a vector. In this case, the deformation vector case can be taken as an example, with its base components being u_x, u_y and u_z as displayed in Equation 2.4.

$$u_{cart} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{2.4}$$

This vector has to change components based on the new reference frame as shown in Equation 2.5,

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \rightarrow \begin{bmatrix} u_\theta \\ u_\phi \\ u_R \end{bmatrix} \tag{2.5}$$

and this is done through the use of a transformation tensor called A, generating an equation as displayed in Equation 2.6.

$$\begin{bmatrix} u_\theta \\ u_\phi \\ u_R \end{bmatrix} = A \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{2.6}$$

Equation 2.6 can be summarized as shown in Equation 2.7.

$$u_{ENU} = A u_{cart} \tag{2.7}$$

The next step is the definition of the transformation tensor and this can be done by starting from breaking down a vector into components relative to unit vectors corresponding to the two different reference systems. In this case, this means expressing the deformation vector as shown in Equation 2.8. This equation means that the vector is the same, but it can be expressed with different components in the Cartesian and ENU reference systems. In this equation, u_x, u_y, u_z are the Cartesian vector components, $\mathbf{i}_x, \mathbf{i}_y.\mathbf{i}_z$ the unit vectors along the Cartesian axes, u_θ, u_ϕ, u_R are the ENU vector components and $\mathbf{i}_\theta, \mathbf{i}_\phi, \mathbf{i}_R$ are the unit vectors in the ENU reference system.

$$\mathbf{u} = u_x\mathbf{i}_x + u_y\mathbf{i}_y + u_z\mathbf{i}_z = u_\theta\mathbf{i}_\theta + u_\phi\mathbf{i}_\phi + u_R\mathbf{i}_R \tag{2.8}$$

Since every component can be written as the product between the transpose of the unit vector and the vector itself, this means for example that

$$u_\theta = \mathbf{i}_\theta^T\mathbf{u} \tag{2.9}$$

and because of the equivalence presented in Equation 2.8, the components in the reference system to adopt, ENU in this case, can be written as displayed in Equation 2.10.

$$\begin{aligned} u_\theta &= \mathbf{i}_\theta^T\mathbf{u} = u_x\mathbf{i}_\theta^T\mathbf{i}_x + u_y\mathbf{i}_\theta^T\mathbf{i}_y + u_z\mathbf{i}_\theta^T\mathbf{i}_z \\ u_\phi &= \mathbf{i}_\phi^T\mathbf{u} = u_x\mathbf{i}_\phi^T\mathbf{i}_x + u_y\mathbf{i}_\phi^T\mathbf{i}_y + u_z\mathbf{i}_\phi^T\mathbf{i}_z \\ u_R &= \mathbf{i}_R^T\mathbf{u} = u_x\mathbf{i}_R^T\mathbf{i}_x + u_y\mathbf{i}_R^T\mathbf{i}_y + u_z\mathbf{i}_R^T\mathbf{i}_z \end{aligned} \tag{2.10}$$

Equation 2.10 can be written in matrix form, giving Equation 2.11.

$$\begin{bmatrix} u_\theta \\ u_\phi \\ u_R \end{bmatrix} = \begin{bmatrix} \mathbf{i}_\theta^T\mathbf{i}_x & \mathbf{i}_\theta^T\mathbf{i}_y & \mathbf{i}_\theta^T\mathbf{i}_z \\ \mathbf{i}_\phi^T\mathbf{i}_x & \mathbf{i}_\phi^T\mathbf{i}_y & \mathbf{i}_\phi^T\mathbf{i}_z \\ \mathbf{i}_R^T\mathbf{i}_x & \mathbf{i}_R^T\mathbf{i}_y & \mathbf{i}_R^T\mathbf{i}_z \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{2.11}$$

This means that now the transformation tensor has been defined as

$$A = \begin{bmatrix} \mathbf{i}_\theta^T \mathbf{i}_x & \mathbf{i}_\theta^T \mathbf{i}_y & \mathbf{i}_\theta^T \mathbf{i}_z \\ \mathbf{i}_\phi^T \mathbf{i}_x & \mathbf{i}_\phi^T \mathbf{i}_y & \mathbf{i}_\phi^T \mathbf{i}_z \\ \mathbf{i}_R^T \mathbf{i}_x & \mathbf{i}_R^T \mathbf{i}_y & \mathbf{i}_R^T \mathbf{i}_z \end{bmatrix} \tag{2.12}$$

and this definition also allows for the use of Equation 2.13 to perform the reference frame change on a tensor aside from a vector. S_{ENU} is the new stress tensor in the ENU reference system and S_{cart} the stress tensor in the original Cartesian reference system. This equation has been taken from Bower (2009).

$$S_{ENU} = AS_{cart}A^T \tag{2.13}$$

After describing the governing equation for the coordinate change, Equation 2.7 for the deformations and Equation 2.13 for the stresses, it is time to show the components of the unit vectors making up the transformation matrix A. They are still taken from Grewal et al. (2007) and only the components will be shown without the transpose and multiplication operations inside the transformation matrix, as it would be overly complicated and useless since vector multiplications and transpositions can easily be performed in any programming language such as, in this case, Python. The unit vectors for the ENU system, in Cartesian ECEF reference system, are shown in Equation 2.14.

$$\mathbf{i}_\theta = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix}; \mathbf{i}_\phi = \begin{bmatrix} -\cos\theta\sin\phi \\ \sin\theta\sin\phi \\ \cos\phi \end{bmatrix}; \mathbf{i}_R = \begin{bmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \sin\phi \end{bmatrix} \tag{2.14}$$

Conversely, the Cartesian unit vectors expressed in ENU coordinates are displayed in Equation 2.15.

$$\mathbf{i}_x = \begin{bmatrix} -\sin\theta \\ -\cos\theta\sin\phi \\ \cos\theta\cos\phi \end{bmatrix}; \mathbf{i}_y = \begin{bmatrix} \cos\theta \\ -\sin\theta\sin\phi \\ \sin\theta\cos\phi \end{bmatrix}; \mathbf{i}_z = \begin{bmatrix} 0 \\ \cos\phi \\ \sin\phi \end{bmatrix} \tag{2.15}$$

After discussing all the necessary theory behind the reference system changes to be applied for this work, it is time to describe how the data from ABAQUS is generated, processed and plotted.

2.2. Generation of the ABAQUS model, iterative simulation process and production of results

This section will give a general explanation of the steps used to generate the results to be plotted and interpreted for each timestep. All of the procedures illustrated here are carried out on the Hipparchos machine from TU Delft, using MobaXterm as an interface with the servers and tmux for multiplexing management and to remotely give commands without keeping the interface open. A summary of the scripts is presented in Chapter , but an overview of model input parameters and modifications used for the implementation of stress coupling will nonetheless be presented here because they have been one of the main objectives of this work.

The input parameters are a series of values used by all the scripts as fundamental units to build the model. The vast majority of them is found in the first two scripts to be run, and the first ones to be defined are constants: the gravitational constant G, the value of π, the value of a ka (kiloannum, 1e3 years), the density of water, ice and sediments and the degrees to radians conversion factor. The next parameters are the Earth's angular rotation speed and a series of flags for additional effects to be included in the model. The first flag to consider is the presence of a liquid core, which if taken into account generates centrifugal forces on the Earth which will have to be translated to a pressure load when building the model. The next one controls whether to take into account centrifugal effects and is set to False, then another flag set to True fixes the center of the model and if true the model does not allow the center to move. The next flag takes into account the iterative sea level equation procedure and is set to True, but it won't have any influence on the model for this work since the scope of this study is to evaluate stress combination in the ice history timesteps and the results have been generated before the sea level equation ran. The next flag decides whether the sea level change is eustatic only (sea level change if the oceans were all in a single basin) and is set to False, which leads to more realistic results due to the distribution of the Earth's seawater. The presence of an initial ice load is set to true and this creates initial forces and stresses due to the ice on the model and the presence of a sedimentation history to take into account is set to False, meaning the additional sediment load is not taken into account. After this, it is determined whether ice loss has to be implemented in the ice history and since the sea level equation is going to be run the ice loss is set to True to determine the sea level increase. The next flag is the one controlling the ramp load settings and is set to true in order to have a more realistic load evolution in time instead of instantaneous variation between timesteps using a step load. There is then a flag set to False used to decide whether to reshape the ice load by giving the ice load values at each timestep the ones from the previous timestep or not. The last flag, set to True, decides whether to use a scattered grid for the ice load and this decides whether to call the ice load gridding function or not in the next scripts. The grid resolution is given together with this flag. A short description of the ice history model will be presented to give a better understanding of how it works.

The ice history model is based on the one used by Barletta et al. (2018), where the ice history is divided into two main components: one with a mass trend of -130 Gt/year between 2002 and 2014 and another one between 1900 and 2002. The ice history until 1900 has been extrapolated based on ice flux observations collected since 1970. Then, it is supposed that the ice load between 1900 and the initial time has not changed in order to establish isostatic equilibrium as described by Blank et al. (2020). This is why the ice history timestep is composed by 11 values where there is a 30 ka jump between the first and second timestep value, where the ice load was constant, and then 10 of them are very close in time. The ice history times vector is composed of times in ascending order, starting from 31.115 ka ago, with ice load at the very first two timesteps, meaning until 1900, and the way it is generated in ABAQUS displayed Figure 2.3. The initial very high and not realistic ice load increase is due to the creation of the load in ABAQUS, using a ramp load option to avoid it appearing instantaneously.

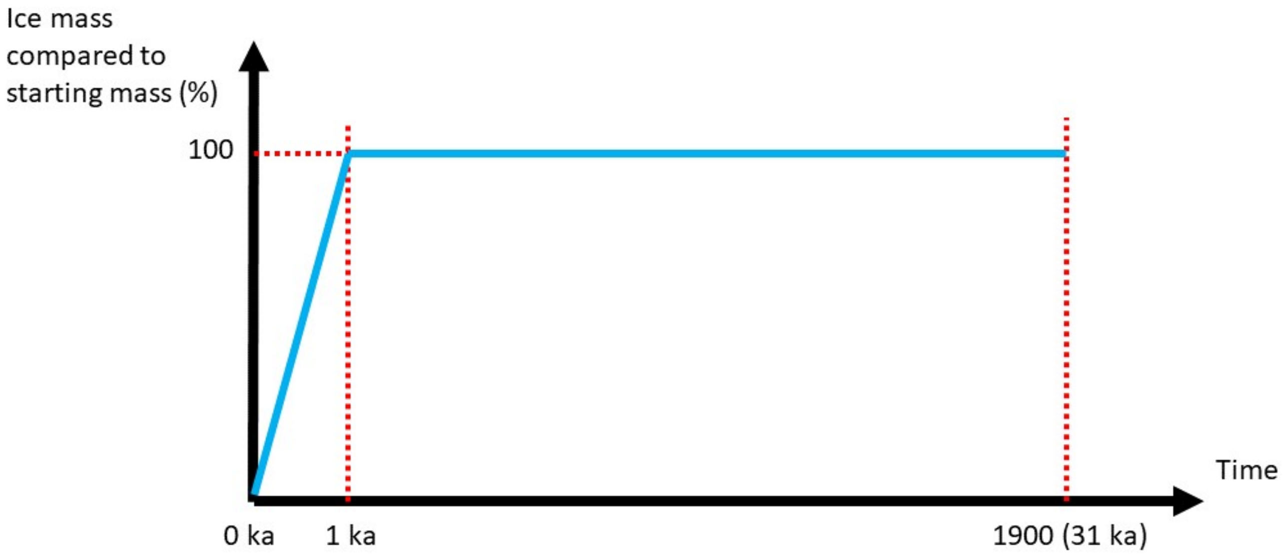


Figure 2.3: Input ice load for the first two timesteps following the description by Blank et al. (2020).

The all the timesteps except the very first is instead represented in Figure 2.4, again from the work by Blank et al. (2020).

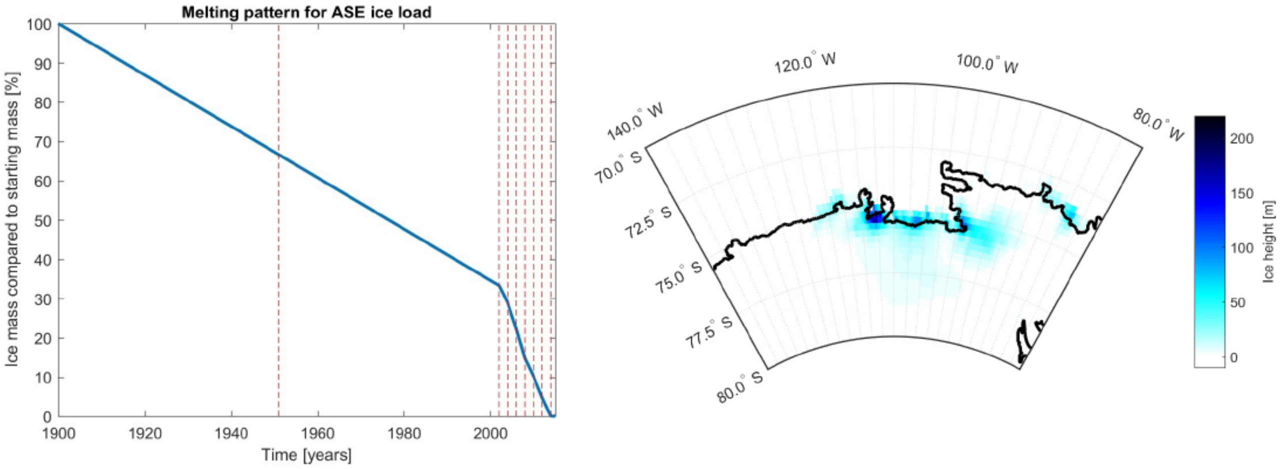


Figure 2.4: Input ice load for the base load as displayed by Blank et al. (2020), from the work by Barletta et al. (2018).

All the timesteps except the first are represented in Figure 2.4 by the red dotted lines at the timestep in 1900, the timestep in 2017 and the 8 dotted lines between them. Interesting results have been generated for this work from the first two timesteps already even though they have the same ice load as discussed in the ice load description and this is why it has been chosen to focus on those, even though the scenario for the very first timestep is not very realistic due to the high ice load increase in a short time span. However, the study of the effects of stress combinations per se inside the timestep still stands and the comparison with baseline result is valid as long as the same timesteps are selected, so there are enough reasons to analyze this part of the model. The combination of ambient stresses and the ones coming from a rapidly decreasing variable ice load at successive timesteps could be interesting for future developments.

It has been chosen to set the creep error tolerance to 1e-2 instead of the original 1e-5 to improve calculation speed due to the small effect varying this parameter has on the simulation results. The next section of the code is about spherical harmonics coefficients, where their degree is set, then it is chosen whether to compensate for the center of gravity asymmetry with respect to the center (set to True) and to disable the spherical harmonics coefficients of degree 0 and 1 (set to False). The number of mesh seeds (markers along the edge of a region to describe the mesh density), polar radii and other seeds are defined and Poisson's ratio is set at 0.49999999. Finally, the coordinates of the center of the high resolution mesh area over Antarctica, where the ASE (Amundsen Sea Embayment) lies as described by Blank et al. (2020) the South Pole are also defined.

It is important to notice that Poisson's ratio is extremely close to 0.5 (0.4999), meaning that hybrid elements will need to be selected in the mesh settings before starting the simulation. This happens because the material is incompressible and its volume cannot change under, for example, uniform hydrostatic pressure coming from all sides. The choice of incompressible material is necessary as explained by Wu (2004) to avoid model instability due to internal buoyancy, which with an incompressible material vanishes. Therefore, pressure stress cannot be computed from the displacements of the nodes (since they don't move) and a pure displacement formulation is inadequate for any element with incompressible material behavior MIT (2020). Hybrid elements are defined with an additional degree of freedom to determine the element pressure stress directly, without using the nodes. The nodal displacements are used only to calculate the shear strains and stresses MIT (2020). Other parameters have to be defined in a separate script called model_data2, and these are the radius for each layer of the model, its density, shear modulus and viscosity. They have been displayed in Table 2.1. It is important to notice that the viscosity is extremely high for the crust to approximate the fact that it is elastic and therefore theoretically infinitely viscous, while it is 0 for the liquid outer core.

Depth [km]	**Density [kg/m3]**	**Shear modulus [Pa]**	**Viscosity [Pa s]**
0	3037	0.50605e11	1e44
30	3438	0.70363e11	4.98E19
420	3871	1.05490e11	4.98E19
670	4978	2.28340e11	1E21
1171	4978	2.28340e11	2E21
2911	10681	1E-20	0

Table 2.1: Earth layer parameters as supplied together with the model scripts.

The rheology is expressed through a rheology file called e.dat which contains B_{diff} and B_{disl} for every element, calculated using Equation 1.3. The values in this original file have been modified in order to maximize the result change due to change in background stresses as will be explained in Section 4.4 and uses a dry rheology with a grain size of 20 mm as one of the results discussed by van der Wal et al. (2013).

Heavy changes have been implemented in the file where the original iterative process took place, because it is where the code to simulate the response due to the combination of glacial and background stresses, meaning the stress coupling, is located. The GIA model used in this thesis work would originally submit an initial job request to create the model database (.odb) file and then send a job request every time the loads are updated through the sea level equation, with a total of 3-4 iterations since it has been described by Wu (2004) that only a low number of iterations is necessary to reach convergence. This has been modified to accommodate the new problem of simulating the interaction between background and glacial stresses generating the aforementioned iterative process. This means introducing an iteration within each of the 11 ice history timesteps after the initial job submission in every aforementioned sea level iteration with the sea level equation running after all the timesteps. For this work, the program did not reach the sea level equation part of the code as the main results were generated early in the iterative process and the sea level recalculation with new loads was not the focus of this thesis.

At the beginning of each of the ice history timesteps, a convergence parameter and the stress iteration counter are initialized. Such measure is needed because the while cycle contained in each timestep can stop in two ways. The first one is when a certain number of stress iterations has been run, which has been dismissed after the scripts have been finalized but has been especially useful during testing. The second one used currently is through the convergence parameter recalculated at each stress iteration being smaller than a certain threshold. The while loop based on this parameter inside each timestep is started and stresses are read from the .odb file, transferred to report files and then saved to .csv files for each part. The convergence parameter is calculated for the current stress iteration and has been defined as the absolute value of the difference between the sums of the Mises stress for all elements coming from the previous and current stress iterations, and if it is smaller than 100 Pa then convergence is reached. The considered stresses are only for part EARTH, because it is the only one where stress coupling can be performed. Three convergence parameter cases need to be defined: one is the most frequently occurring one, for any timestep but not in the first stress iteration. The stress tensor generated for the current cycle in the previous lines of code is read and together with the one generated in the previous stress iteration. The Mises stresses are read from both of them and the absolute values of the difference of the sum of their respective matrix columns is computed. If this value is lower than the threshold of 100 Pa defined in each while cycle then the stress iteration interrupts. The next case is the very first stress iteration, when the timestep is 0 and the iteration is 1, and nothing must be done if this condition is true as there are no previous cycles where data has been created. The last condition is for every timestep that is not the first and the stress iteration number is 1, for any iteration. In this case, the previous stress iteration cannot be defined as current stress iteration-1, because stress iteration 0 does not exist. It is therefore necessary to extract the highest stress iteration number in the previous step and read the matrix defined by these attributes as the one where the "stress iteration 0" would be, as that would be the moment in time immediately preceding the one when the stress iteration counter is 1. The convergence parameter calculation is then done in the same way as explained earlier. The section of the code then terminates to go on to the generation of the deflection reports, where again deformations are read, stored in report files and then saved to .csv files.

The next large addition to the iterative process script has been the stress coupling process proper. The first action to take for stress coupling is the definition of the stresses to couple, which have been set at different orders of magnitude: for this work, 0.008 and 1 MPa have been selected both for normal component S11 and shear component S12. These stresses can be substituted with more realistic ones, whose definition can start from the theory of Section 4.5 and the explanation of the code from Section A.6. The base stresses are then read from the .csv file and the background stresses are added to the selected components using Equation 2.16, then the Mises stress is recalculated with Equation 1.5 and stored in the rheology file.

$$\begin{bmatrix} S_{11} & S_{12} & S_{13}\\ S_{21} & S_{22} & S_{23}\\ S_{31} & S_{32} & S_{33}\end{bmatrix} + \begin{bmatrix} 1e6 & 1e6 & 0\\ 1e6 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \rightarrow \begin{bmatrix} \S 11_{coupled} & S12_{coupled} & S_{13}\\ S21_{coupled} & S_{22} & S_{23}\\ S_{31} & S_{32} & S_{33}\end{bmatrix} \tag{2.16}$$

This rheology file will be used for the creep law defined by Equation 1.1 so that it can use the new coupled Mises stress. The updated creep law is then used in a new job definition and submission inside the same timestep. This is a novel procedure which has never been described before, since as stated in Section 1.1 past studies focused on the use of ambient stresses as pre-added to the Mises stress. Here the ambient stresses are instead added inside the timestep to generate an iterative process. This job definition and submission starts an iterative process because the model will be updated and originate new stresses, which are read and again combined with background stresses to update the rheology file. This can generate a different viscosity after stress coupling which will cause different stresses to be generated in the model after again sending the job request and so on. This iterative process inside the timestep stops after convergence is reached. Convergence is checked on the Mises stress after reading it from the model as described earlier: if the absolute value of the difference of the sum of the Mises stresses is in absolute value smaller than a certain threshold (100 Pa in this case), the timestep is incremented.

The new iteration scheme can be summarized as in Figure 2.5. The hierarchy is as follows: the uppermost line follows the iteration determined by variable Num, the second line from the top follows the timesteps described by N_step in the script, the second line from the bottom follows the stress iterations in each timestep described by the variable $cycle\_counter$ while the bottom line describes the general sequence of operations for each stress iteration. The stress iterations are contained in the timestep loop which is contained in the sea level iteration loop. In every sea level iteration, the job is submitted before the timestep iteration, the timestep iteration takes place and finally sph_tools_TPW7_4SLE is run and the new loads are defined and applied as portrayed on the top of the figure. It can be seen that the smallest time unit for this thesis work will be the stress iteration. The original scripts provided only consisted in the three uppermost iterations, where each of them took around one day and a half to complete.

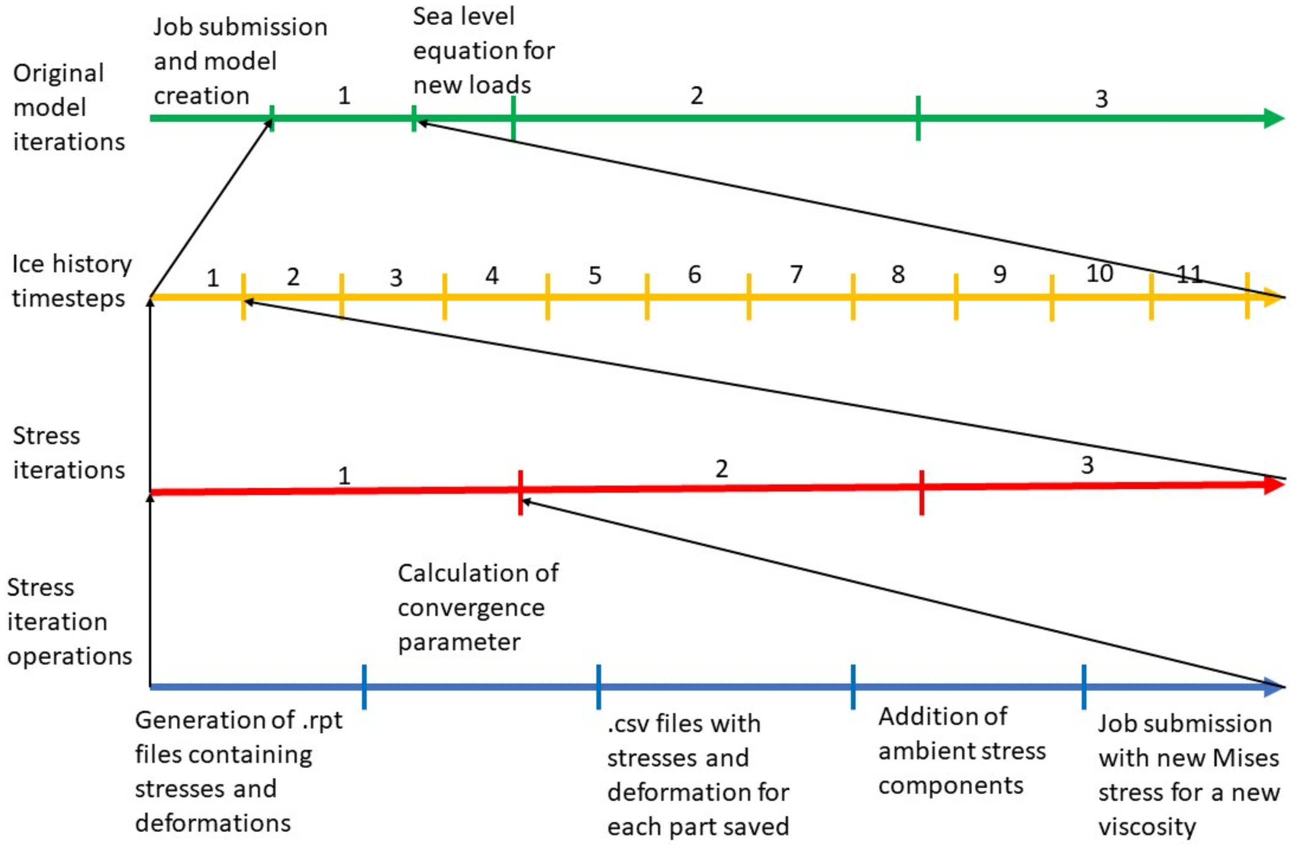

Job submission and model creation
Sea level equation for new loads
Original model iterations
1
2
3
Ice history timesteps
1
2
3
4
5
6
7
8
9
10
11
Stress iterations
1
2
3
Calculation of convergence parameter
Stress iteration operations
Generation of .rpt files containing stresses and deformations
.csv files with stresses and deformation for each part saved
Addition of ambient stress components
Job submission with new Mises stress for a new viscosity

Figure 2.5: Iteration scheme for the simulation on which the thesis is based.

This new setup requires between 4 and 5 hours to complete each stress iteration, and it can be deduced that running the entire simulation with 2 or 4 iterations containing 11 timesteps of 3 stress iterations is very time consuming and even having to restart it after stopping at a certain point can be very problematic. However, since a single timestep already encompasses an entire iterative process of interaction between background and glacial stresses as explained earlier and this method has never been tried before, it was not necessary to run all the timesteps, let alone all the iterations with sea level equation, to generate results to analyze and present as for this thesis work. The reason the sea level equation does not need to be considered here is that the study of the Earth response to stress combination can be done with respect to the ice load itself at any ice history timestep, which means that it is not required to reach the load recalculation due to ice load and sea level redistribution. This is why the results presented in this thesis work will only encompass the first timestep and the beginning of the second one. It has nonetheless been decided to add a mechanism to start the simulation again after an interruption from a "checkpoint" instead of the very beginning so that testing while developing the code modifications would be quicker and future work using this code would be more efficient. The way these "checkpoints" work is by encapsulating some operations such as the most time-consuming one of job submission and completion in if cycles. The cycle checks for the existence of a text file relative to that operation and, if it is not found, the script executes the operation and then produces the file. Job submissions are some of the most time consuming operations, which either update existing files in the run folder or create new ones that are then stored in the stress iteration or run folders. When the script has to rerun because of errors causing the program to crash or freeze and reaches one of those operations, the script searches for the existence of the corresponding .txt file, and if that file is found, the operation is skipped. Skipping operations can be done because the results of these operations, which as said before are either file updates or file creation, are stored in the corresponding folder right before the creation of the .txt file. This setup has been used to generate, among others, a .txt file every time the job submission for a certain stress iteration had been completed, which means that the script was able to go through different iterations in a short time before reaching the point where the execution was interrupted. The same thing has been done for many other actions that either consumed time or could not be rerun for that stress iteration if others were skipped. An example of the latter is the stress coupling procedure.

The output of this procedure will be a series of report files where the stresses and deformations for each stress iteration are stored together with the .csv matrices where this data has been transferred, together with a rheology file for each iteration and a single Earth.dat file for every stress iteration.

2.3. Generation of data matrices in Python for MATLAB plotting

The next step in the data plotting procedure consists in reorganizing the data in the report files so that is can be easily read and plotted. The inputs used for the data processing procedure are matrices containing the stresses and deformations for every element and node of every part at every stress iteration of the simulation, and the Earth.dat produced once and containing the node coordinates for all the nodes in the model. More importantly, this file also contains the nodes that make up each element, as they are needed to calculate the coordinates of each element centroid. Other files supplied are rheology file at any stress iteration and the rheology file containing the latest coupled stresses. The structure of the input folder for a given run with all its sub-folders is displayed in Figure 2.6.

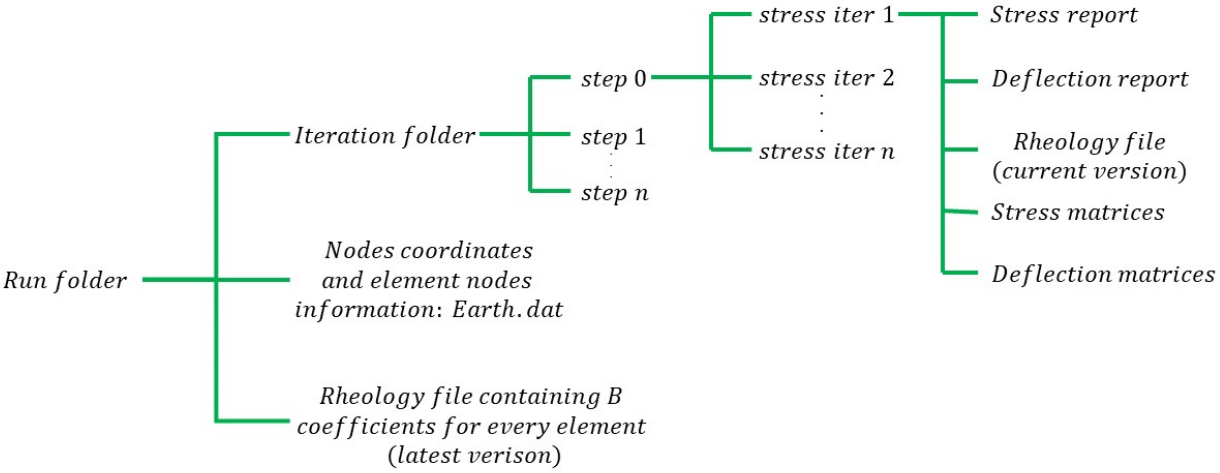

Run folder
Iteration folder
step 0
step 1
step n
stress iter 1
stress iter 2
stress iter n
Stress report
Deflection report
Rheology file (current version)
Stress matrices
Deflection matrices
Nodes coordinates and element nodes information: Earth.dat
Rheology file containing B coefficients for every element (latest verison)

Figure 2.6: Structure of the input data before processing it.

These inputs are given by downloading the folder generated on the Hipparchos servers and containing these files plus the report files for that stress iteration and the .csv matrices containing the coupled stresses for the stress iteration. The user needs to manually create the folder where this downloaded directory has to be stored, and will contain the Earth.dat file and the rheology file called for the last stress iteration of that run aside from the iteration directories. This rheology file is the result of using Equation 1.3 on every model element and therefore supplies B_{diff} and B_{disl} for each of them. It does not matter what stress iteration the rheology file is for since the parameters used to calculate the viscosity (B_{diff} and B_{disl} to plot do not change with the stress iteration, but only with the rheology.

The data processing procedure is illustrated in the flowchart of Figure 2.7.

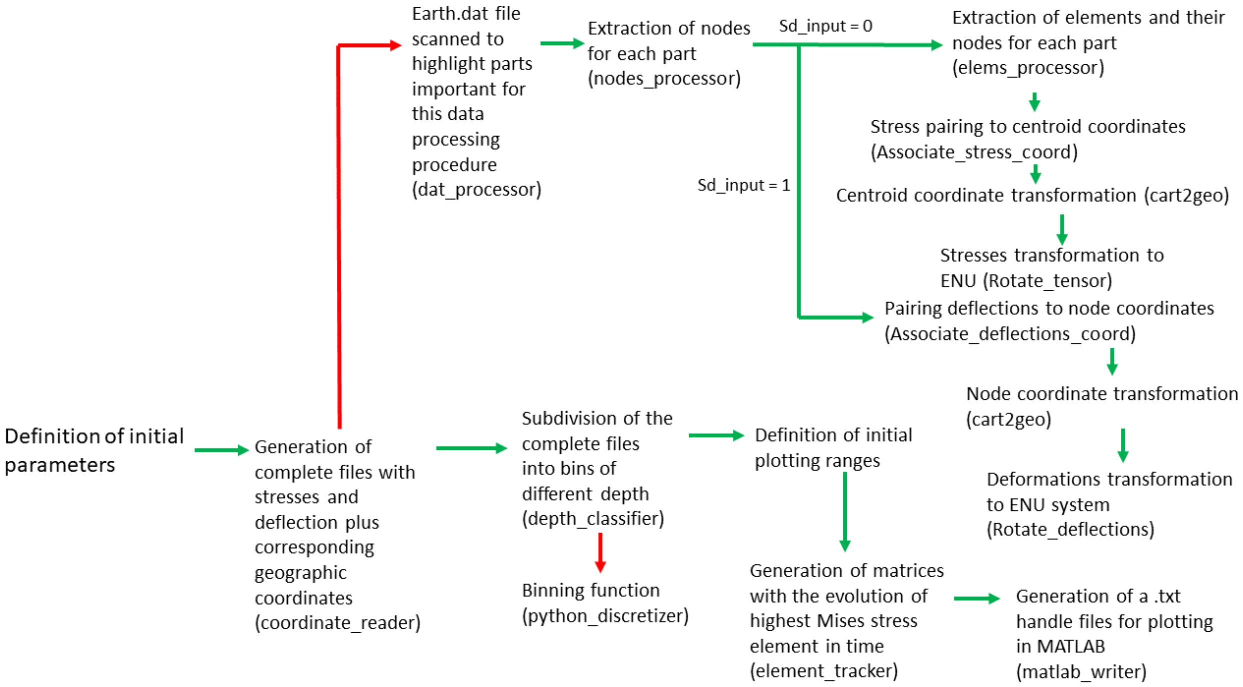

Earth.dat file scanned to highlight parts important for this data processing procedure (dat_processor)
Extraction of nodes for each part (nodes_processor)
Sd_input = 0
Extraction of elements and their nodes for each part (elems_processor)
Stress pairing to centroid coordinates (Associate_stress_coord)
Sd_input = 1
Centroid coordinate transformation (cart2geo)
Stresses transformation to ENU (Rotate_tensor)
Pairing deflections to node coordinates (Associate_deflections_coord)
Node coordinate transformation (cart2geo)
Definition of initial parameters
Generation of complete files with stresses and deflection plus corresponding geographic coordinates (coordinate_reader)
Subdivision of the complete files into bins of different depth (depth_classifier)
Definition of initial plotting ranges
Deformations transformation to ENU system (Rotate_deflections)
Binning function (python_discretizer)
Generation of matrices with the evolution of highest Mises stress element in time (element_tracker)
Generation of a .txt handle files for plotting in MATLAB (matlab_writer)

Figure 2.7: Data processing flowchart, starting from the input files of Figure 2.6 to the generation of files ready to be plotted in MATLAB.

This chart contains a model of the main script with all its subordinate functions. The red arrows mean that the functions after those arrows are all called as subordinates of the functions from which the red arrows start, while the green arrows indicate a sequential process.

The outputs of the data processing procedure are a series of .csv files for each part to be processed (in this case only EARTH, but the user can adjust the files to process by modifying the processing filters) containing the stress or deflection values associated to the centroid or node coordinates of that element or node, its radial distance from the center, depth, latitude, longitude both in a X, Y, Z and ENU coordinates. The files generated in these two reference systems are stored in different folders to avoid making the .csv files too large. Other outputs are the centroid or node coordinates for each part and the depth classified centroids or nodes with all the aforementioned values for every part. The most important file to be generated at the end of the procedure is however a .txt file containing the location for all of these matrices that MATLAB needs to access for plotting purposes and the type of variables that need to be plotted.

The new folder structure after the data processing procedure, with all these new files, is much more complex and displayed in Figure 2.8. The red arrows indicate in this case the new folders with all of their files to be created. There can potentially be more than one iteration folder because this name refers to the high-level iterations where the sea-level equation is carried out in ABAQUS at the end of each of them. "Complete files" refers to the files containing stress and deformations paired to their corresponding centroid and node coordinates respectively, while "depth files" refers to the complete files after having been broken into different files based on the number of selected depth bins. The "processing results" folders refer to the fact that the files contained in them will now be organized data that can be readily used for plot generation in the post-processing step. The input paths for the post-processing procedure are highlighted with the red "branches", while in orange are the input file(s).

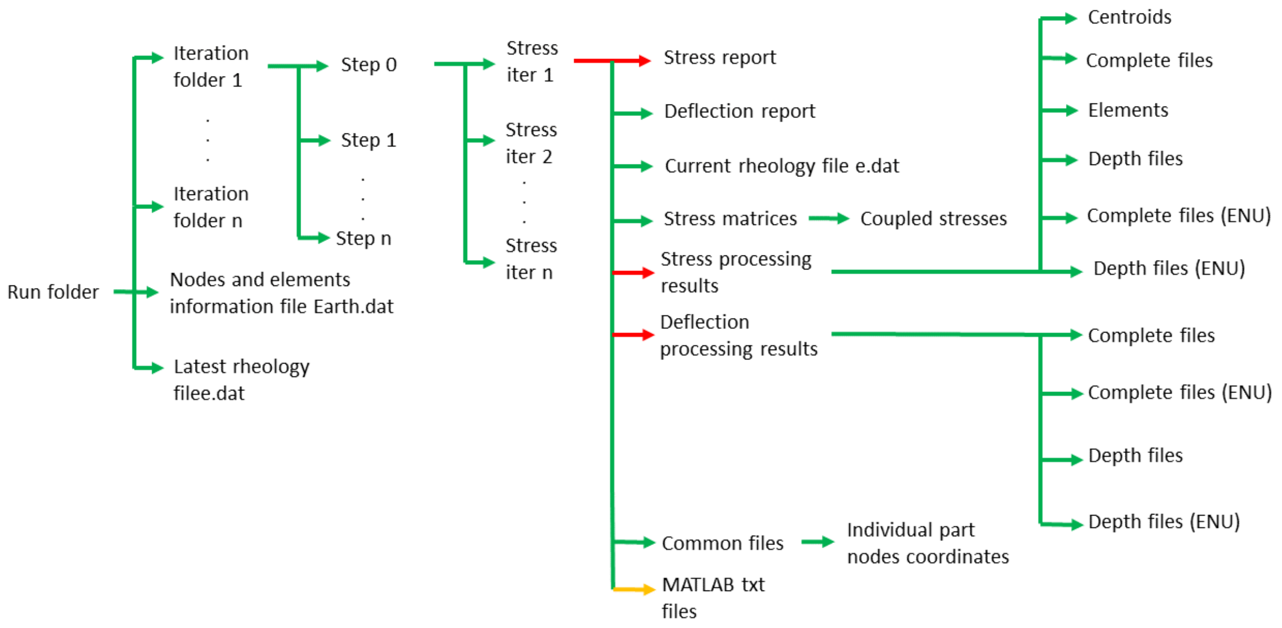


Figure 2.8: Input folder scheme after running the processing algorithms in Python.

2.4. Generation of plots with MATLAB using the data matrices generated in Python

The last section of the code is dedicated to plotting the data with some additional operations. The scheme used for this is similar to what has been done during the processing phase, with a main file calling a series of functions. The main blocks in which the program is divided are shown in the list below, with the corresponding flowchart displayed in Figure 2.9. Since this is a processing flowchart and it does not illustrate the files generated after the process, the red arrow indicates subordinate functions which are only called in the function their red arrow departs from. The three main functions on the horizontal line can potentially all be called independently from each other depending on the operation to perform.

The final folder configuration after the plot generation is shown in Figure 2.10. The folder paths generated during this procedure are again highlighted with red branches, since this is another picture displaying the folder setup after the plot generation process and does not describe the process itself.

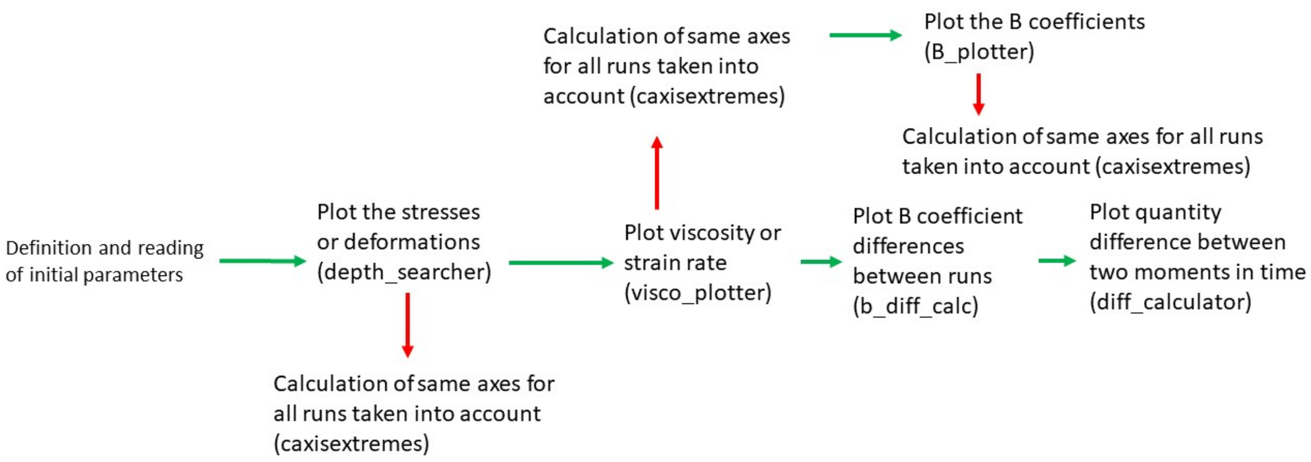

Definition and reading of initial parameters
Plot the stresses or deformations (depth_searcher)
Calculation of same axes for all runs taken into account (caxisextremes)
Plot viscosity or strain rate (visco_plotter)
Calculation of same axes for all runs taken into account (caxisextremes)
Plot the B coefficients (B_plotter)
Calculation of same axes for all runs taken into account (caxisextremes)
Plot B coefficient differences between runs (b_diff_calc)
Plot quantity difference between two moments in time (diff_calculator)

Figure 2.9: Flowchart of the main file for the plotting procedure.

It can be seen that the procedure is much more straightforward than the processing one displayed in Figure 2.7. The cause of this is that this section is simply dedicated to plot generation.

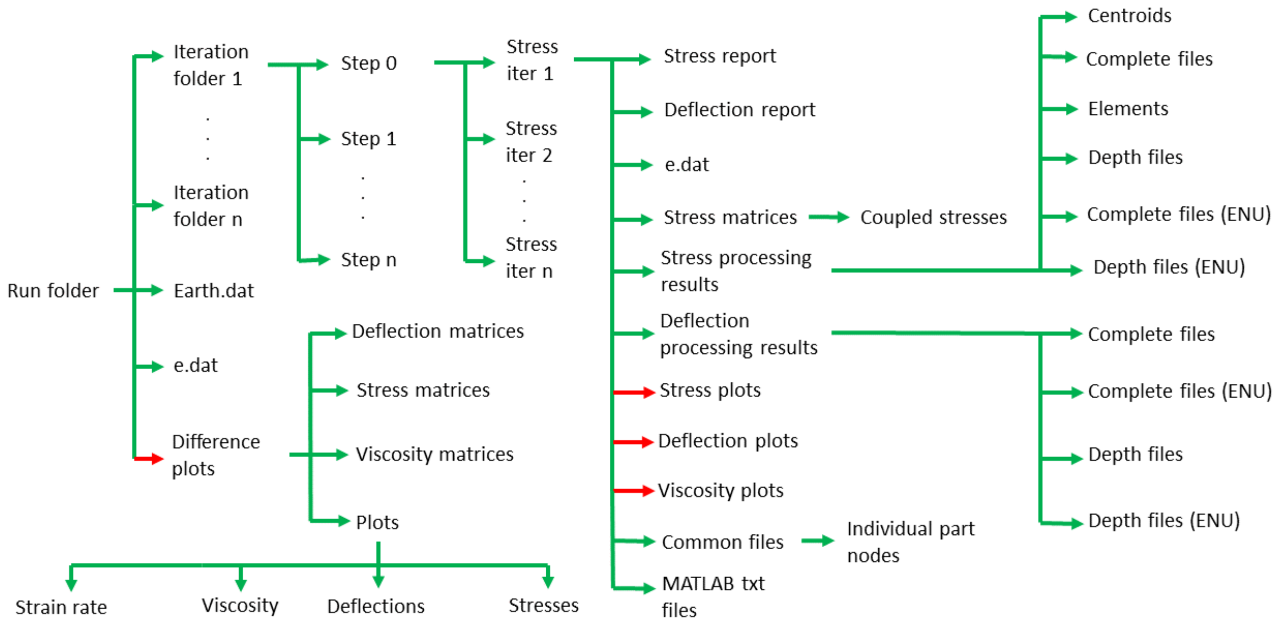


Figure 2.10: Structure of the run folder after running the MALTAB scripts for plotting.

3

Verification process and example run

3.1. Pre-results generation verification

Before the results generated in different cases and presented in this work could be accepted, it had to be verified that the difference in results between the files generated by Bas Blank and supplied at the beginning of the thesis work and the by base model used for this work were minimal. A short description of the verification process for this model will also be provided, with all of it coming from Blank et al. (2020).

The base model, following a setup described by Martinec et al. (2018), has been benchmarked against two different models both coming from the aforementioned study. Both benchmarks consist in the comparison of deformations over time, but the second one uses a model with added oceans with variable shorelines. For clarity, the base model by Blank et al. (2020) to be verified will be called the B model, the model for the first benchmark will be called WW model and the second benchmark model will be called D1 model, following the naming conventions of Blank et al. (2020). For both benchmark tests, the 5-layered Earth model coming from Spada et al. (2011) is used. The first benchmark test is against the rotational symmetric FE model developed by Wouter van der Wal, called WW model, and discussed in Martinec et al. (2018). The test has been conducted using a 1 km thick ice sheet with a load composed of a flat disk surrounded by a slope in order to minimize the differences coming from the load discretization over any random mesh and to avoid the insurgence of high local stresses deriving from a pure disk load. Because the resolution of the WW model is 0.5 by 0.5 degrees and it is not possible to do the same for a full 3D FEM model without a strong impact on computation times, it has been chosen to start with a higher resolution and gradually decrease it to see whether the results of the B model would approach the one coming from the WW model. This first benchmark test indeed shows that the larger the model resolution, the smaller the relative difference between the WW and B models is, going from 0.6 to 0.15%. This means that the base model can be considered verified due to the low relative difference.

The second benchmark test has been between the B and D1 model, the latter again discussed by Martinec et al. (2018). This time, the load per unit area is a spherical cap with a maximum height of 1.5 km and density ρ=931 kg/m^3. In this case it was necessary iterate the model several times to find an input topography close (less than 1 m) to the one used as the D1 input topography. This happened because the D1 model uses the topography at t_0, initial ice history time, as input while the B model uses the present topography for it. The model did not reach full convergence because of computational and time constraints. This mainly has an influence on the found RSL (Relative Sea Level) near the basin, which in turn influences the deformation in the basin area. Therefore the difference in deformation is higher near the basin area than in the area around the ice load. Again, the resolution of the mesh is smaller than the D1 model, which leads to errors in the exact loads and structural flexibility of the model. The results of this test nonetheless show a difference of 1.3 % between the uplift rates of the high resolution area which have been the main benchmarking factor, and this difference has been decided as small enough for this benchmarking test which has therefore been passed too. This means that the model can be considered verified.

Plotting the differences between the results coming from first operational model to be generated in ABAQUS and the ones sent at the beginning of the thesis work was necessary to ensure that all input parameters for the model, as described in Section 2.2, were right. These differences have been mapped over a latitude/longitude 2D chart for each of the files sent for verification at the beginning of the thesis and the most significant of them have been added here, being displayed in Figure 3.1 and Figure 3.2 for two different timesteps, showing that the differences in results are low for both of the supplied verification files towards the end of the ice history vector. All the values resulting from the difference between the outputs of this model and the one given at the beginning of the thesis are presented here and mainly belong to an order of magnitude of 1e-6, with some maximum values of around 1e-5. The smaller difference values appear where the order of magnitude of the deformations is around 1e-3 while the larger ones appear where the order of magnitude of the deformations is around 1e0 (deformation of 5 to 6 meters). This difference is at most 3 orders of magnitude smaller than the results themselves, so it is not significant, and is even smaller in the areas of largest deformation. The input parameters are therefore all correct.

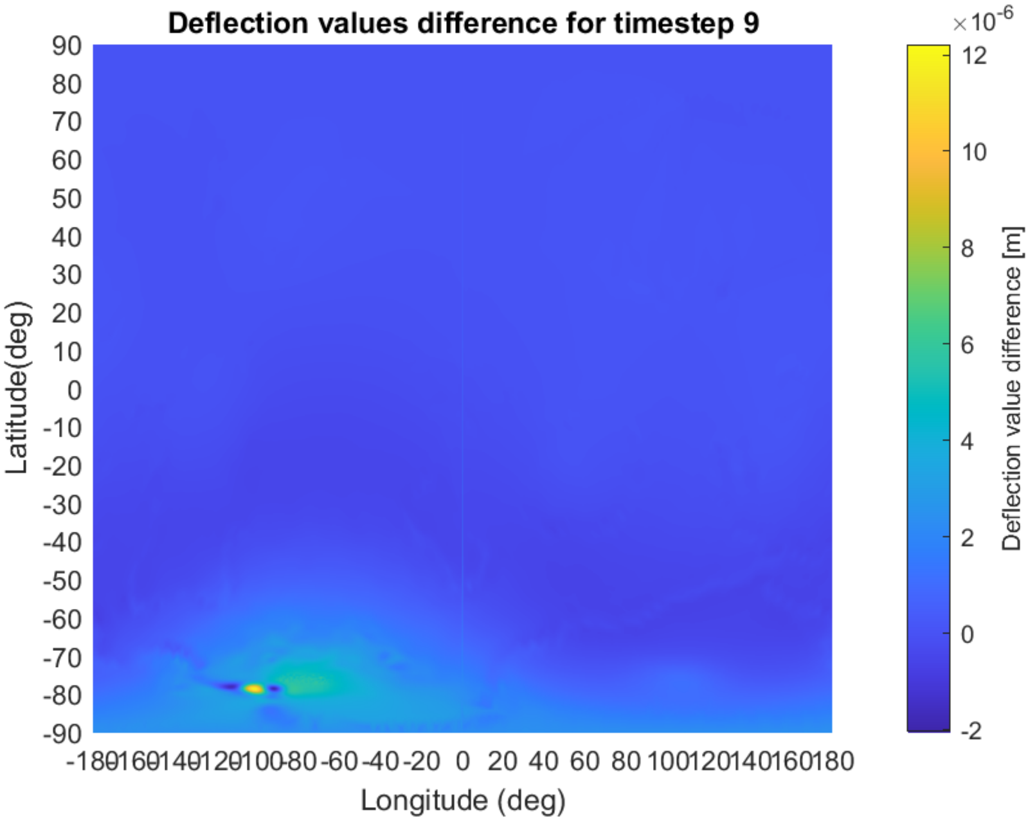


Figure 3.1: Absolute difference for the deformation magnitude between the model and its settings used in this work and the files supplied at the beginning of the thesis at step 9.

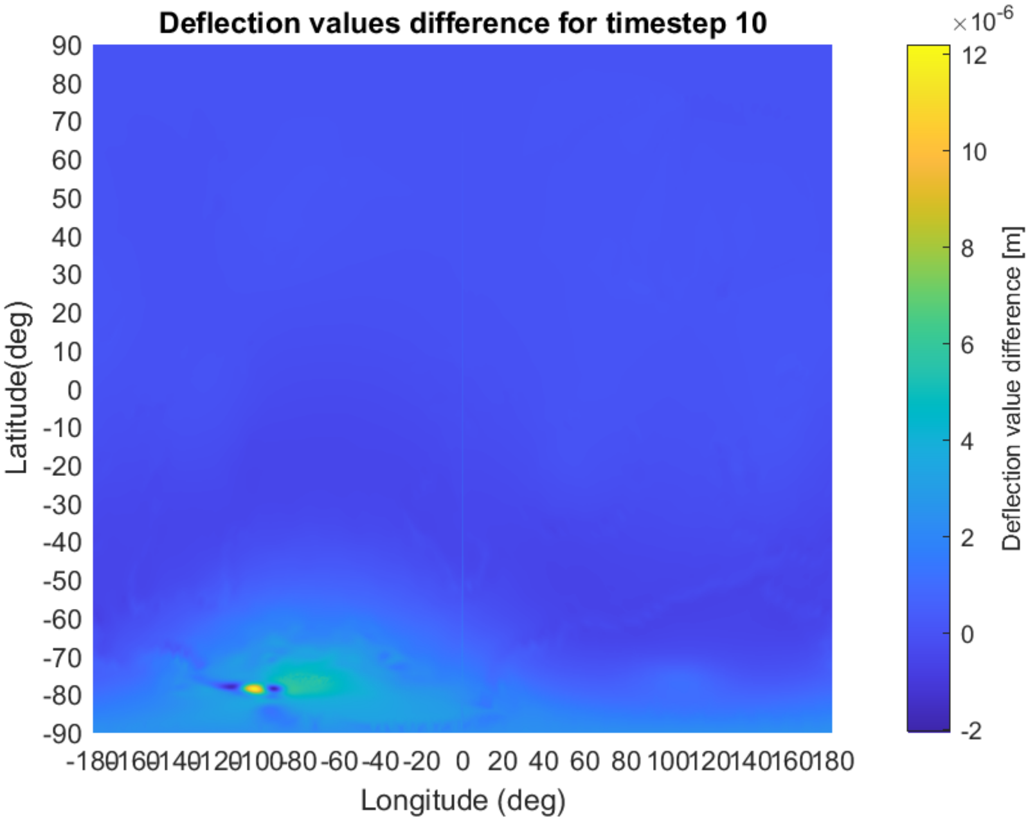


Figure 3.2: percentage difference for the deformation magnitude between the model and its settings used in this work and the files supplied at the beginning of the thesis at step 10.

3.2. From the data reference system to the standard Cartesian reference system

It has been explained in Section 2.1 that the first reference system change to perform has been the one from the data to the standard Cartesian frame. However, it was not known how the data was saved, meaning which stress or deformation component in which column. The order in which the coordinates are saved has been found by using a small MATLAB function called sysver used to read the complete deformation files generated in Python by joining the deformation values and the corresponding node coordinates. This function is described in Section A.6.

Different combinations have been tried and the one that showed the deformations to be in the right area, meaning in the South Pole over West Antarctica, has been the one where the first and third coordinate columns are swapped first followed by swapping the second and third, which corresponds to an original Y, Z, X configuration if the transformations are applied in reverse. The column manipulation sequence for the deformations has then been used for nodes and centroids and has then been adapted for the stresses, as described in Section 2.1.

The results for unchanged ABAQUS coordinates are shown in Figure 3.3, the ones for a coordinate change based on the model display reference frame in Figure 3.4 and the results for the correct coordinate system change are shown in Figure 3.5. It is clear that the right sequence of coordinate changes is the one where the deformations are concentrated in West Antarctica, and that is the sequence that has been used for all the stress and deformation plots which show higher stresses and deformations where they are supposed to be.

Finding the correct order to swap the node coordinates has been crucial to then verify whether the second coordinate transformation from the Cartesian to ENU reference frames was correct. At this point, it was known that every component was in the right data column because of the plots displayed below. Therefore, if for example the largest deformation value ended up in a column different from the third one (Up) where it was supposed to be in the ENU reference system, the transformation matrix A from cartesian to ENU would have been wrong. This is true because, as explained earlier, the uplift resulting from the ice load mainly causes deformations on the vertical direction, which in the ENU coordinate system is the third coordinate (Up).

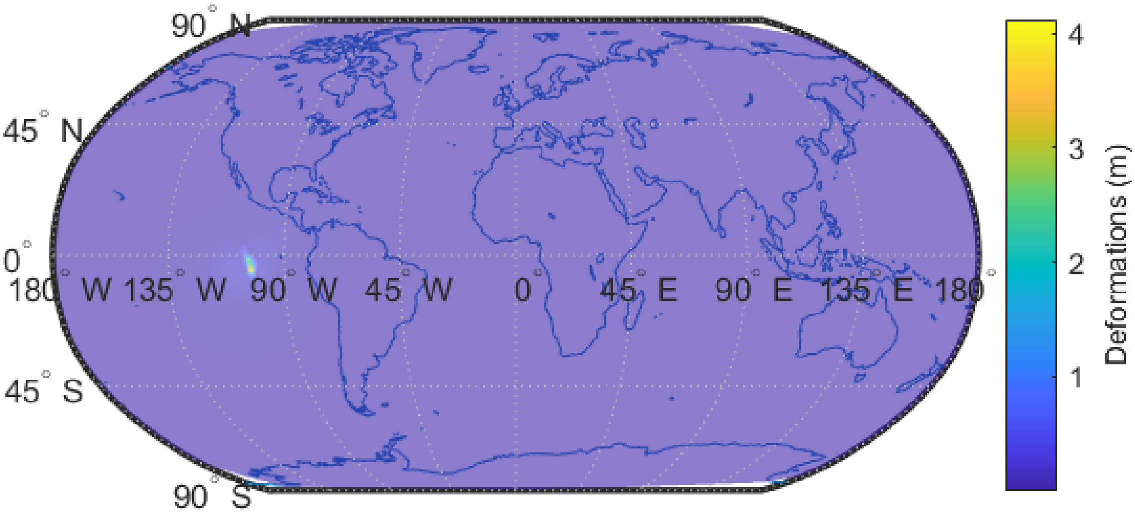


Figure 3.3: Deformation magnitude plot with no change from the ABAQUS data frame.

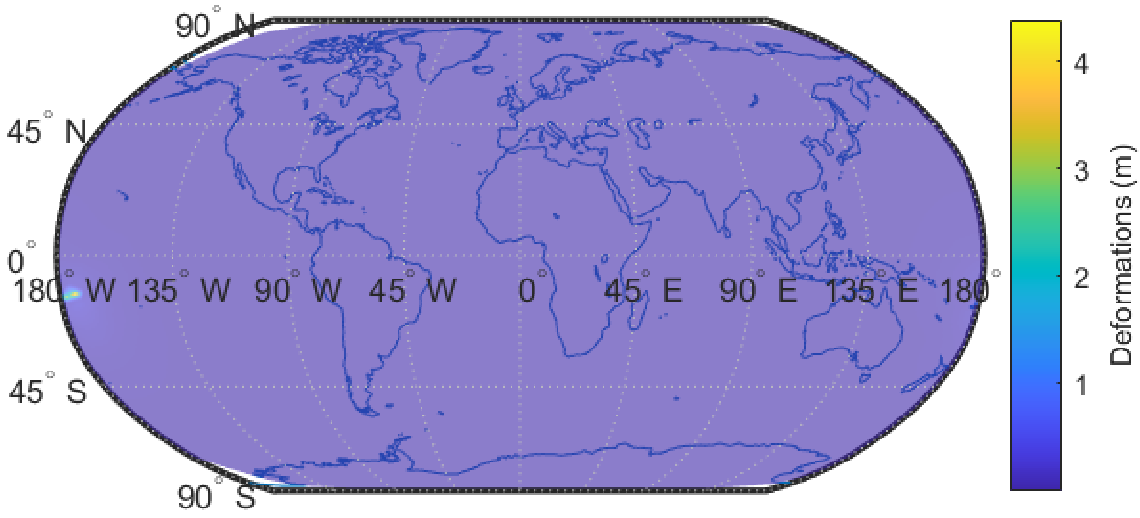


Figure 3.4: Deformation magnitude with a reference system change based on the ABAQUS display frame.

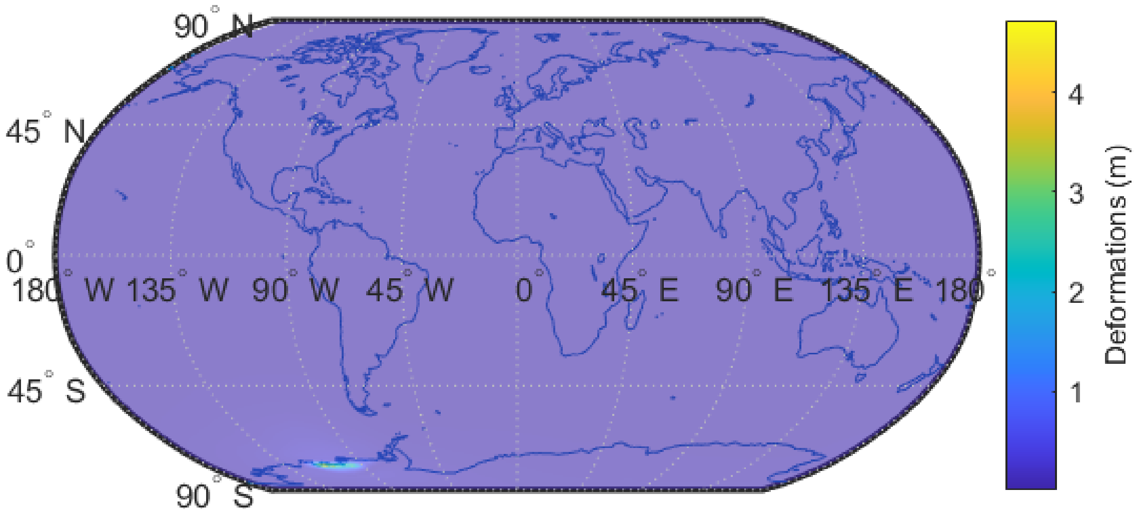


Figure 3.5: Deformation magnitude with a reference system change based on the ABAQUS data frame, showing deformations in the correct positions.

3.3. Mises stress formula verification

After submitting the very first job request to create the mode .odb file, a file with stresses and deformation results is created. Background stresses, in this case added on the S11 and S12 components, have to be added to the ones coming from the model and written in report files. This obviously changes the Mises stress, which needs to be recalculated. This new Mises stress needs to be written in the rheology file to calculate the new viscosity, which in turn will change the stress outputs an therefore requires another job submission. The job is submitted with the new rheology file and gives new stress and deformation results which again need to be paired with background stresses until convergence in a timestep is reached or a set number of these stress iterations has been reached. In this case the timestep is incremented when either of these is true, but it has been observed from the plots and tables in Chapter 4 that convergence is already reached in the number of timesteps given as limit, meaning that using the convergence criterion only will lead to the same results.

Combining the background and ice load stresses therefore required either finding the formula used by ABAQUS to calculate the Mises stress or a good approximation of it and apply this formula to to the new Mises stress calculation after verifying it. Equation 1.5, from Kim (2009) has been used for this and implemented in a Python function described in

Section A.6, where a stress matrix was read and its components used to calculate the Mises stress as per Equation 1.5. The absolute difference between this Mises stress and the one red by the matrix would then be calculated to verify that the differences would be below a certain threshold. The maximum percentage relative difference between the ABAQUS Mises stress and the one calculated with Equation 1.5 is always around 0.2%. This relative error corresponds to an absolute maximum difference of 2 Pa between the output stresses and the ones calculated with Equation 1.5, so the approximation given by this formula is good enough for this thesis work.

3.4. Choice of depth ranges for plotting

Both the nodes and centroids making up the model are all situated at slightly different radial distances from the model origin, which means at slightly different depths. Because of this, it is impossible to plot the results at the same depth and a depth range has instead to be selected. However, it is not possible to use a single depth range for both stresses and deflections because of the difference in location of the points where they are defined, meaning centroids and nodes respectively. The choice of depth ranges to use is greatly aided by the files produced during the depth binning procedure, as their names contain the depth ranges and the points in that depth range. If the file size is too small, it means selecting that depth range is not advisable as there are few or no points in it. However, there is another consideration to take into account: it might be that, despite the high number of points in a file, the finer mesh for that depth range over the AOI gets "cut out" if the number of points is large, but not sufficiently so. Therefore, a supplementary depth range check will be presented in this section.

An easy way to visualize the points filtered by the chosen depth, latitude and longitude ranges is by producing scatterplots of the z-coordinate over the three point coordinates. Figure 3.6 shows how selecting a depth range of 145 to 150 km which still contains a rather large number of points (with a file of around 1500 kB) still leaves large "holes" in the finer mesh. These holes can be filled by selecting a depth range between 145 and 160 km, and the result of this change is shown in Figure 3.7. The visual checks for all the depth ranges chosen for this work are displayed from Figure 3.8 to Figure 3.10. These checks prove that the chosen depth ranges not only have enough points to generate a coherent plot, but they are also in a number large enough to have a complete finer mesh over the AOI.

Generating these plots takes a very short time but they can considerably improve the way the results are displayed, so it is recommended to run this check for every selected depth range of every chosen quantity, which are stresses or deflections.

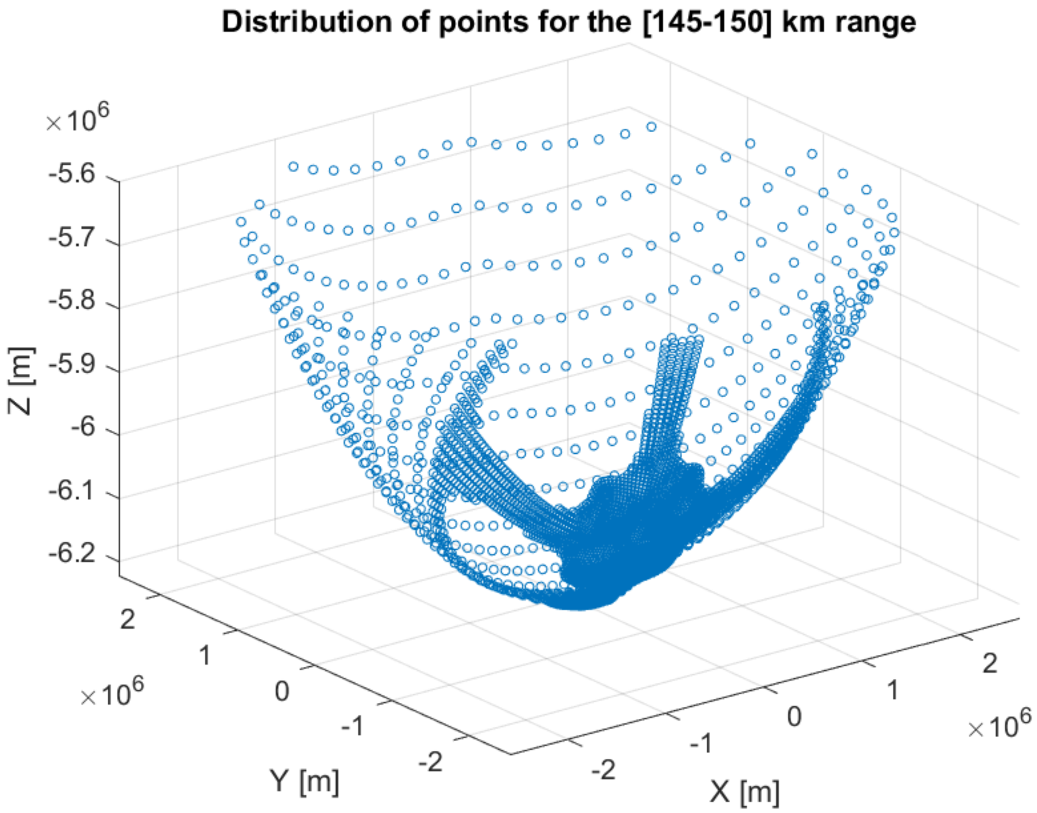


Figure 3.6: distribution of filtered mesh points for a depth range of 145 to 150 km, showing "holes"in the mesh.

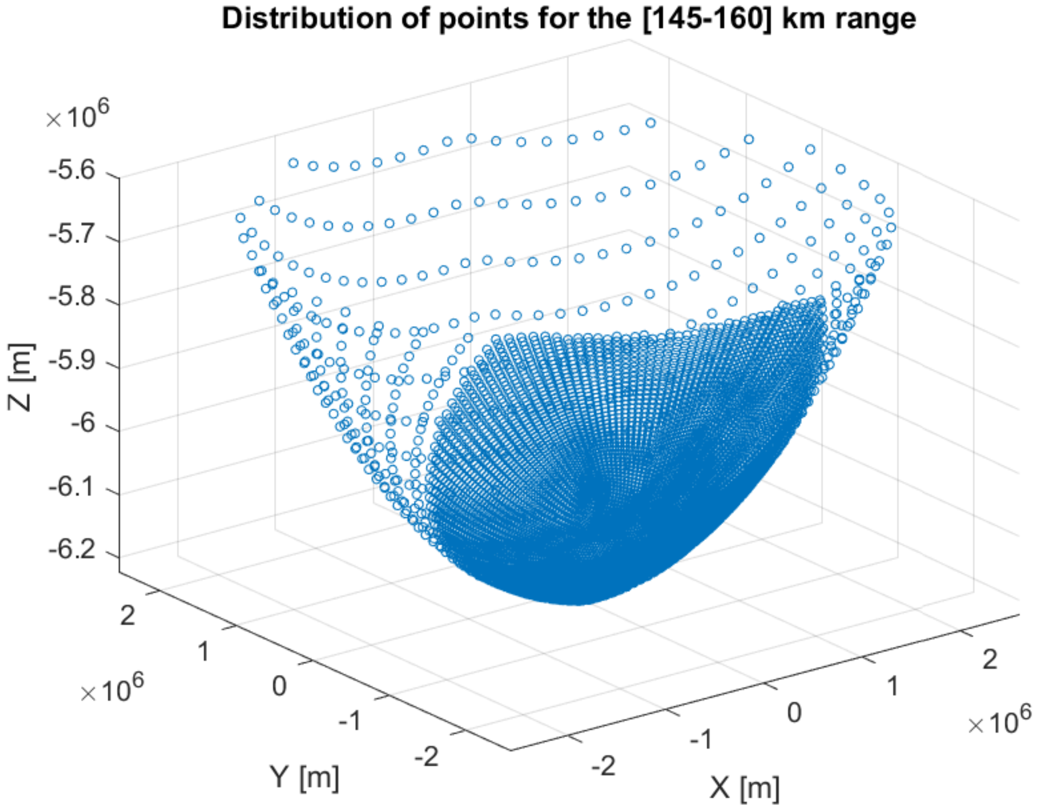


Figure 3.7: Distribution of filtered mesh points for a depth range of 145 to 160 km, the lower depth range for stresses, with the finer mesh not having any holes now.

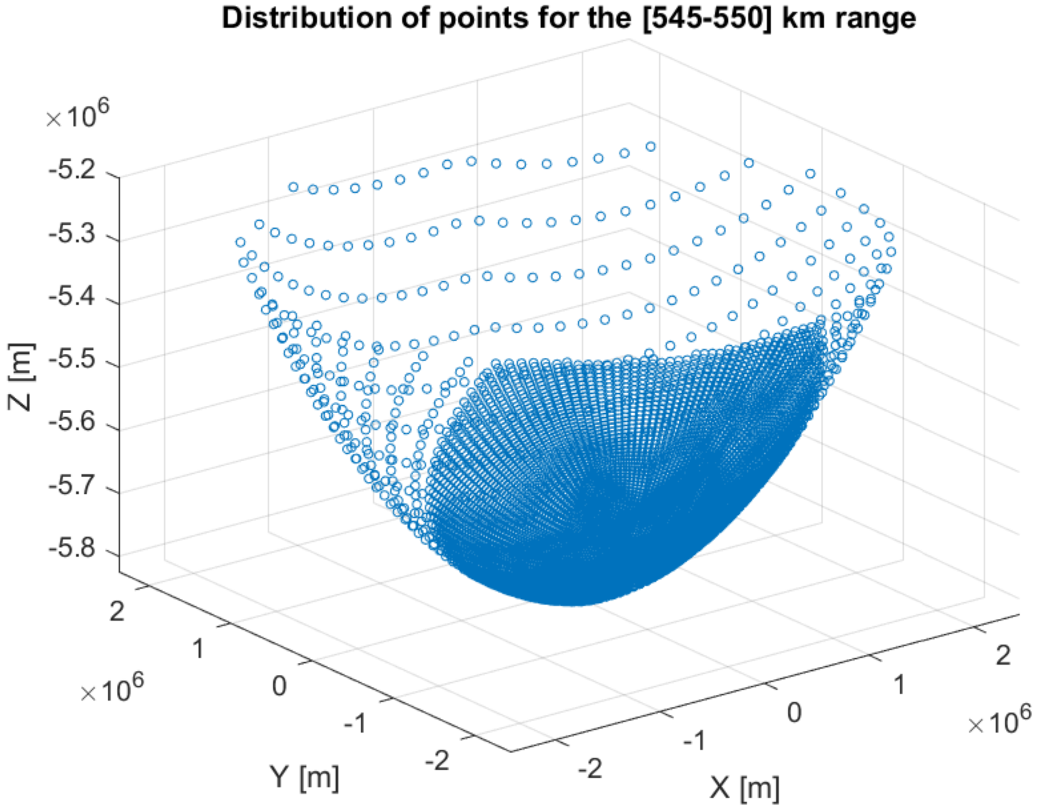


Figure 3.8: Distribution of filtered mesh points for a depth range of 545 to 550 km, the higher depth range for stresses.

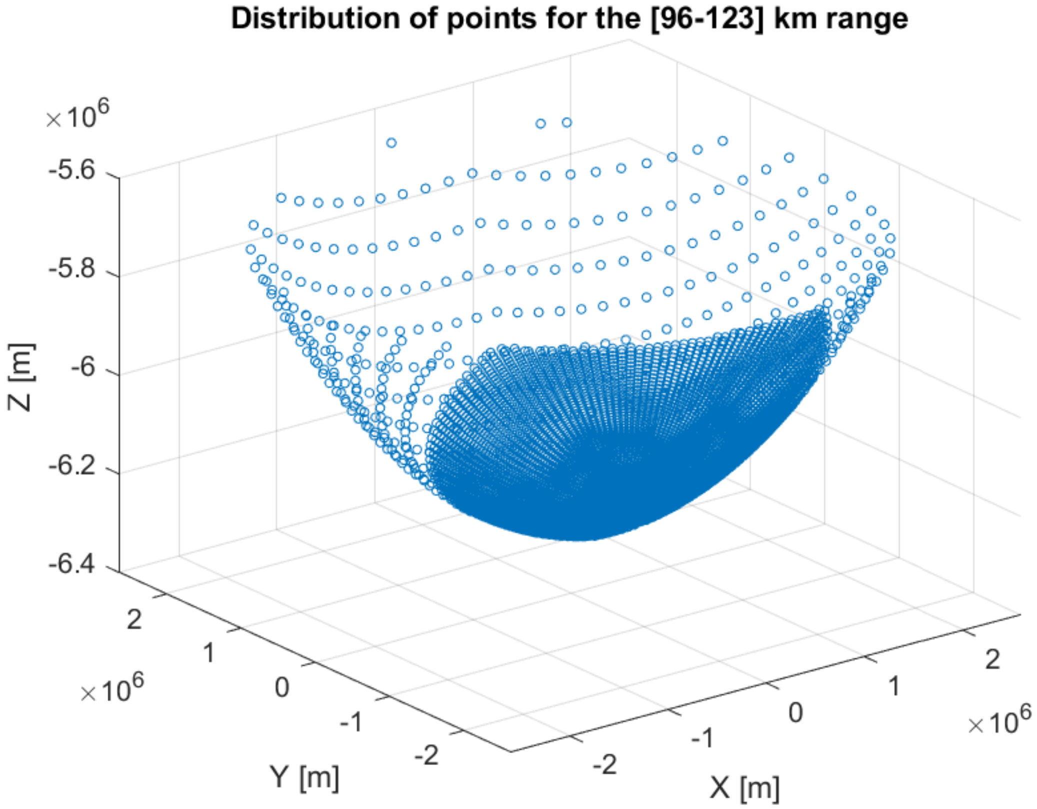


Figure 3.9: Distribution of filtered mesh points for a depth range of 96 to 123 km, the lower depth range for deflections.

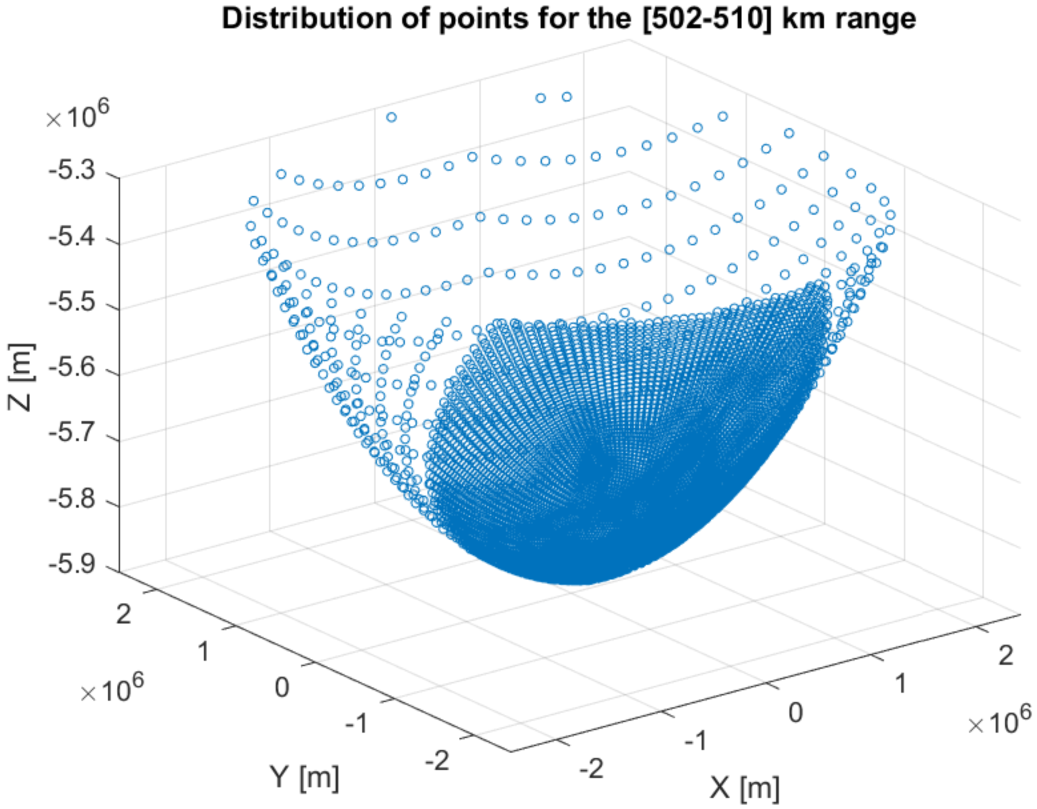


Figure 3.10: Distribution of filtered mesh points for a depth range of 502 to 510 km, which is a higher depth range for the deflections.

3.5. Demo files and example run

The example files used for the run can be found at Morra (2020a) and the Python and MATLAB scripts with a short description for each of them can be found at Morra (2020b). The example given uses a background stress of 0.008 MPa (since the results for that background stress and the ones presented here for 1 MPa are the same for this thesis work), and has all the necessary files to generate all the results described in Chapter 2, from the .csv to the plots. The initial folder follows the scheme displayed in Figure 2.6.

This folder needs to be downloaded and left in the same configuration as it has been uploaded on GitHub, then the scripts can be run freely to have a demonstration of their functionality. The run number must be set to the one indicted in the GitHub folder name and the iteration number to 1, but steps and cycles can be varied a little more. The scripts can then be freely modified by the user to better suit their needs once they have taken confidence in the script functionalities.

4

PRESENTATION AND ANALYSIS OF THE RESULTS

The results coming from the simulated viscoelastic iterative process will be presented in this chapter both in the form of plots and tables. The plots will display stresses, deformations and viscosity over Antarctica, while the tables will mainly be focused on the evolution of the viscosity and stresses of the element with highest Mises stress in the West Antarctica region. The plots have all been generated for the first two timesteps of the simulation, while taking into account all stress iterations of the first timestep and the first stress iteration of the second timestep. All the plots have been generated in the ENU reference system and every component uses the same scale across all runs displayed here. The only exceptions are the difference plots between iterations 2 and 3, where some of them have been generated instead of the ones for stress iteration 3 due to convergence being reached very quickly inside a timestep as it will be explained later. The plots have all been generated for a longitude between -180 and 180 ° and a latitude of -90 to 65 °, while the depth will be specified in the pictures as it will vary between two depth ranges.

All the component plots will be generated for the two rheology files called e.dat and two different background stresses of 8e-3 and 1 MPa. The difference between the two rheology files is that one of them has been generated for a dry rheology with a 20 mm grain size, while the second one uses a wet rheology and a smaller grain size of 10 mm. A 20 mm grain size has been chosen to maximize the effect of the dislocation creep for a dry rheology, while a wet rheology uses a smaller grain size to reduce the viscosity as much as possible. Two stresses have been chosen because the former is closer to the model outputs in order to check whether forces with the same order of magnitude as the ones deriving from ice loads would generate an effect that does not "overshadow" the base results. The latter, on the other hand, is closer to the supposed range of tectonic stresses presented in Wu (2001) where a value on its lower end has been chosen to take into account an average stress value over the entire Earth over the entire mantle and to be closer to the value of 1 MPa used in the analysis by Schmeling (1987). Moreover, the plots will be generated for two depth ranges which will be used again in the tables from Section 4.4 to give an overview of what happens at both high and low depths. These ranges will be the lower depth range of 145 to 160 km and the higher depth range of 545 to 550 km in order to investigate whether there is a different response at different mantle depths. Depth ranges had to be used instead of single depth values due to the fact that nodes and centroid in ABAQUS are all at slightly different depths. Slightly different ranges had to be chosen for deformations compared to stresses as they are calculated in the nodes and not the centroids with the difference being explained in Section 2.1, and these have been a 96 to 123 km for lower depths and 502 to 510 km for higher depths. All the depth ranges have been chosen in accordance with the checks described in Section 3.4. It has been chosen to generate many of the plots at different snapshots in time at these two depth ranges only for the one run (the one for 1 MPa background stress because it is closer to a realistic uniform tectonic stress value from the aforementioned study by Schmeling (1987)), since many of the trends observed for this value are repeated with different rheology and background stresses combinations. The only exception is the comparison between old and new rheology which compares results belonging to two different runs.

For a shorter notation, the two rheology files used for the result generation will be described here so that, instead of their description in every paragraph title, dry and wet rheology file will be used to denote their differences. The dry rheology refers to the original rheology (dry and 20 mm grain size) where its B_{diff} factor has been reduced by a factor of 10 compared to the baseline results and B_{disl} is unaltered, while the wet e.dat file uses a different rheology with a higher water content and a smaller, 10 mm grain size which reduces the model viscosity. Both files have been chosen in order to maximize the effect that changing the background stress could have on the results, with the wet rheology being especially important in this sense. The following notation will be used for the stresses and deformations cited and displayed both in this chapter and Chapter A.6.3:

1. S11 and S12 are the normal eastward stress component and the tangential northward stress component respectively, as shown in Figure 2.1 but with the E,N,U directions for X,Y, and Z respectively;
2. U3 is the deformation components in the vertical (Up) direction.

Before introducing the results of this simulation, it is convenient to show the Mises stress, deformation magnitude and viscosity at the two timesteps used for the base simulation and obeying the ice load described in Section 2.2. This can help the interpretation of the results coming out of the simulation modified to include stress combination and it also makes the comparison between base and modified simulation results much clearer. The Mises stress plots are shown in Figure 4.1. The reduction in stress is consistent with an ice load being the same for a period of 30ka as described in

Section 2.2, where being the same means that it is being applied with the same intensity on the ABAQUS model at these two time steps.

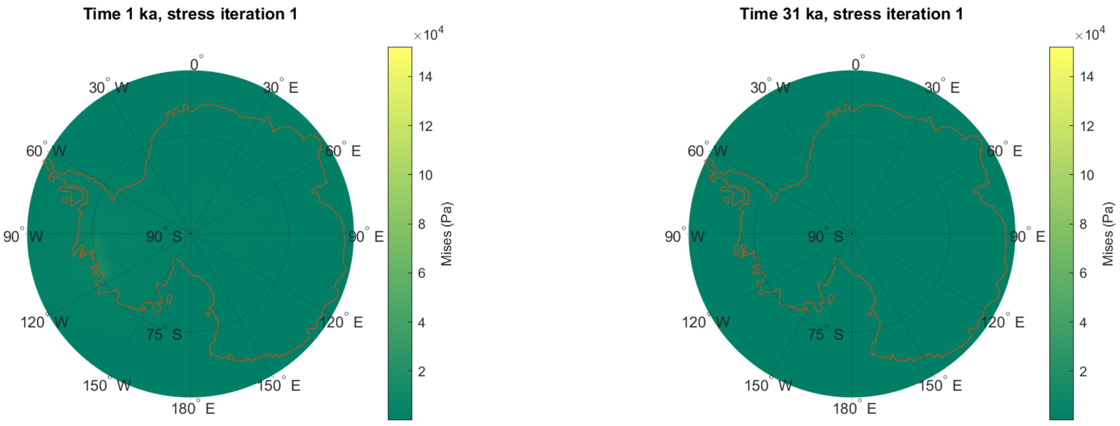


Figure 4.1: Mises stress plots for a base run and a depth range of 145 to 160 km and timesteps 1 and 2.

The deformation magnitude is shown in Figure 4.2. The deformations increase but only slightly, which is consistent with a ice load being the same for a very long period of time. The disappearance or reduction of the ice load would have led to larger deformations at the second timestep due to the reduction in the pressure on the Earth's surface.

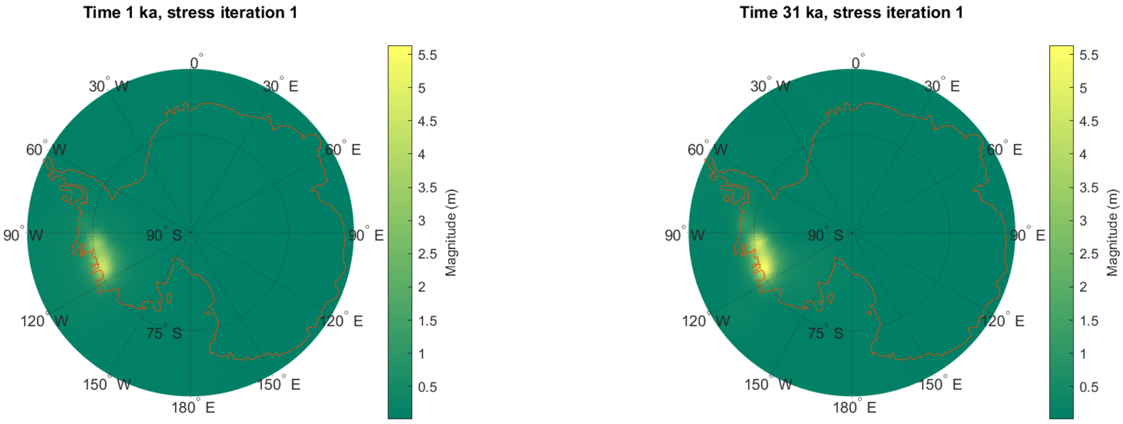


Figure 4.2: Deformation magnitude stress plots for a base run and a depth range of 96 to 123 km and timesteps 1 and 2.

The viscosity is shown in Figure 4.3, and it does not exhibit any visible changes despite the decrease in stresses. This could be due to the fact that for this particular case, the effect of the Mises stress reduction at both the numerator and denominator of Equation 1.2 cancel each other.

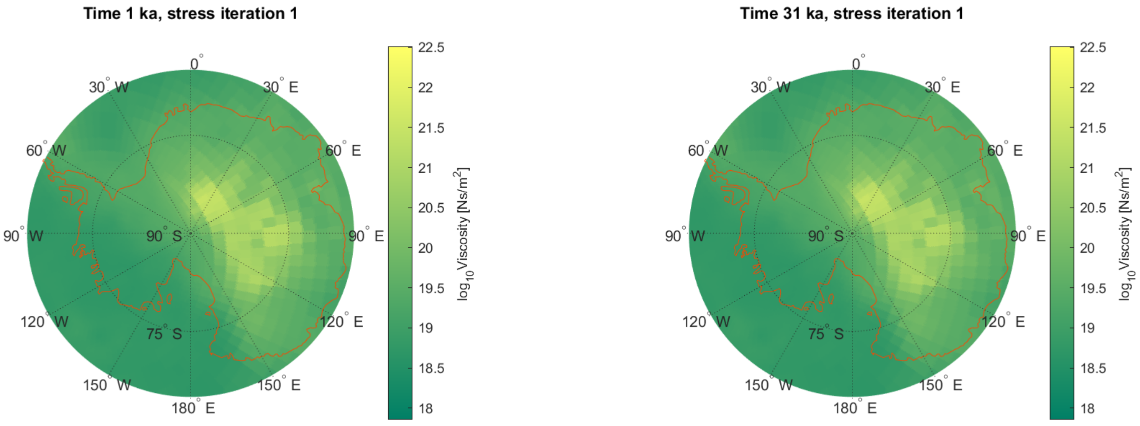


Figure 4.3: Viscosity plots for a base run and a depth range of 145 to 160 km and timesteps 1 and 2.

Comparison with these results at each timestep will be offered in the discussion of the results coming from the modified simulation for each relevant set of figures. Because of this, all the plots use the same colorbar scale for each component and depth range chosen. All the plots, be them for stress, deformation or viscosity, already show convergence of the simulation results after stress iteration 2 in a timestep. This is easily observable from the output difference between two stress iterations (2 and 3) which is too small to be discerned and can be basically interpreted as numerical errors if there is any difference at all. This quick convergence is important because adding background stresses to the ones coming from the simulation at every stress iteration means that the input Mises stress changes and with it the viscosity, since the Mises stress is internally used by ABAQUS for the viscosity calculation. What is observed is that for this case study this effect does not "pile up" bringing the model to divergence for both background stresses and rheologies used. These additional stresses can be compared to a perturbation, with the model relatively stable with respect to them and quickly stabilizing on new values at each timestep. Comparing this situation with the base case, it could be argued that adding background stresses causes the Earth to relax faster due to lower viscosity and dissipate stresses more quickly without accumulating them. Deformations are also reduced due to this faster relaxation. This is especially evident from a comparison between Figure 4.3 and Figure 4.16, which shows how an area of low viscosity arises where the ice load is applied. This comparison also shows that the viscosity difference between the two timesteps is much more pronounced with values decreasing with time once background stresses are added in the ice load application area. The fact that the difference in output, be it stresses or deformations, is so small is actually one of the main results of this thesis work and shows how quickly convergence is reached for the iterative process of stress combination. An example of this is shown in Figure 4.4, which shows the difference in values between stress iterations 2 and 3 for both the Mises stress and deformation magnitude. The same happens for the viscosity because of its strong dependence on the Mises stress, which in turn depends on the other stress components, and this is why the viscosity plot has been omitted here. More detailed descriptions for the behavior of each quantity will be found in the next corresponding sections. It is worth noting that, because of the model convergence between stress iterations 2 and 3, the values belonging to these two iterations can be considered the same aside from small numerical errors. For this reason, stress iteration 2 of timestep 1 can be compared with values to stress iteration 1 of timestep 2 instead of stress iteration 3 of timestep 1 (since the values are the same), as will be done in this work.

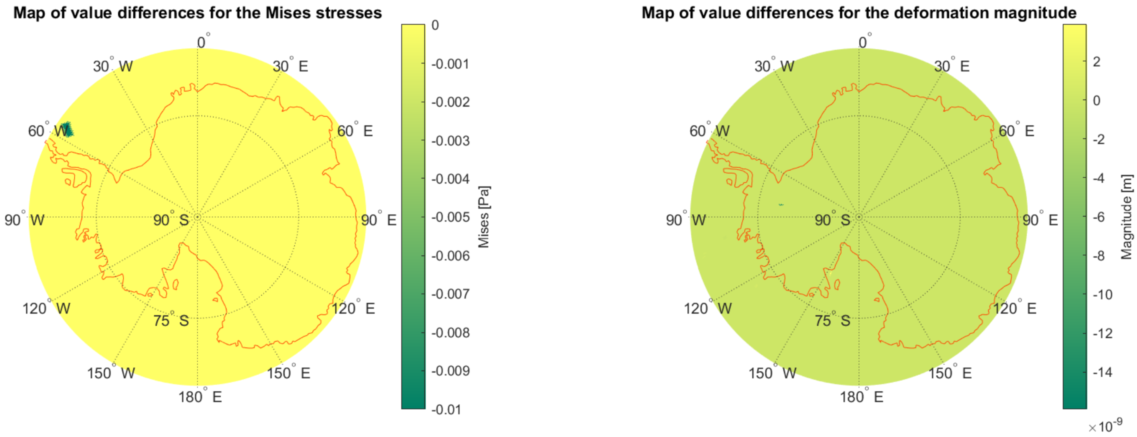


Figure 4.4: Plots of the difference in the outputs between stress iterations 2 and 3 for both the Mises stress and deformation magnitude, for a depth of 145 to 160 km, a background stress of 1 MPa and dry rheology.

Another important finding is that the model output at every stress iteration analyzed in this work does not change at all when changing the background stress value. This means that at each stress iteration, while the coupled stresses change with the background stresses due to the different ambient stress module to be added, the resulting output stresses after updating the model with the creep law of Equation 1.1 using the coupled stresses are the same. Such a result shows that all the quantities here presented do not seem to be sensitive to a change in background stress, which could be caused by the stresses used here not being different enough to induce a change in the creep law and subsequent viscosity, the viscosity change not being significant between ambient stresses or not having a realistic distribution due to being constant and with the same sign everywhere. A more detailed explanation for this phenomenon will be given in Section 4.4. This is true for both dry and wet rheologies and low and high depths, as displayed in Section 4.4. Even though those tables are related to stress and viscosity, the fact that viscosity is the same for both ambient stresses also means that, leaving all other variables fixed, the resulting deformations will also be the same with different ambient stresses. Therefore, all the plots will only be presented in the case of one background stress, which is 1 MPa. Unless specified otherwise, all the plots have been displayed for the aforementioned dry rheology.

4.1. Stress plots

The stress plots will be presented for the Mises stress in this section, with the normal eastward (S_{11}) and shear northward (S_{12}) components plotted in Section B.1. The plots show that the Mises stress increases in absolute value, a trend mirrored by S11 and S12 as in Section B.1, as time passes when considering a low depth range value of 145 to 160 km both for both dry and wet rheology. This can be seen in Figure 4.5, where different snapshots in time (stress iteration 1 for timestep 1 and 2) of the normal and shear stress components are presented. Many of the considerations resulting from the observation of the Mises stress are applicable to the S11 and S12 stress components, so they have been been added to the appendix for comparison as they would be superfluous for a discussion here. The main difference between the baseline plots and these ones is that the Mises stress decreases in the baseline plots while it (and S11 and S12 too from Section B.1) displays a net increase at low depths between timesteps 1 and 2 when adding ambient stresses, as shown by Figure 4.5. This should have been expected, because constantly adding ambient stresses at each stress iteration inside a timestep means that the Mises stress is in the end going to increase between the timesteps. This increment of stresses in rebound areas has also been observed by Schmeling (1987). This constant ambient stress addition does not seem to influence the convergence of values reached inside every timestep due to higher relaxation. This means that the Mises stress stabilizes on values which increase at every timestep. Moreover, the stresses are generally higher compared to the baseline plots, again to be expected due to the constant background stress addition.

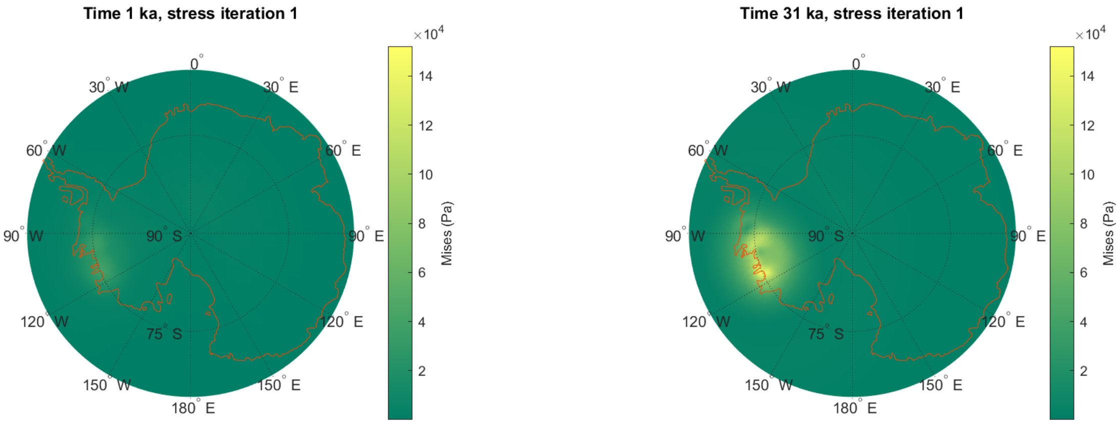


Figure 4.5: Mises stress for two different time snapshots, one at stress iteration 1 of timestep 1 and one at stress iteration 1 of timestep 2, for a depth range of 145 to 160 km, showing a net stress increase.

However, the stresses actually slightly decrease when going from stress iteration 3 of timestep 1 to stress iteration 1 of timestep 2 despite the presence of ambient stresses, as it is shown in Figure 4.6 and is especially clear from Table 4.2. This can be explained by the fact that the ice load is reapplied before the generation of the results of stress iteration 1 of timestep 2, with the same value as what is found in timestep 1 and stress iteration 1. This time the ice load has been the same for a very long time compared to a really steep ramp load application as per Figure 2.3, which leads to a slight decrease in stresses which is mitigated by the close distance in depth between the ice load and the stresses displayed here. Moreover, the faster relaxation rate and more efficient stress dissipation which is the main cause of convergence of results at each timestep also helps reducing the stresses produced.

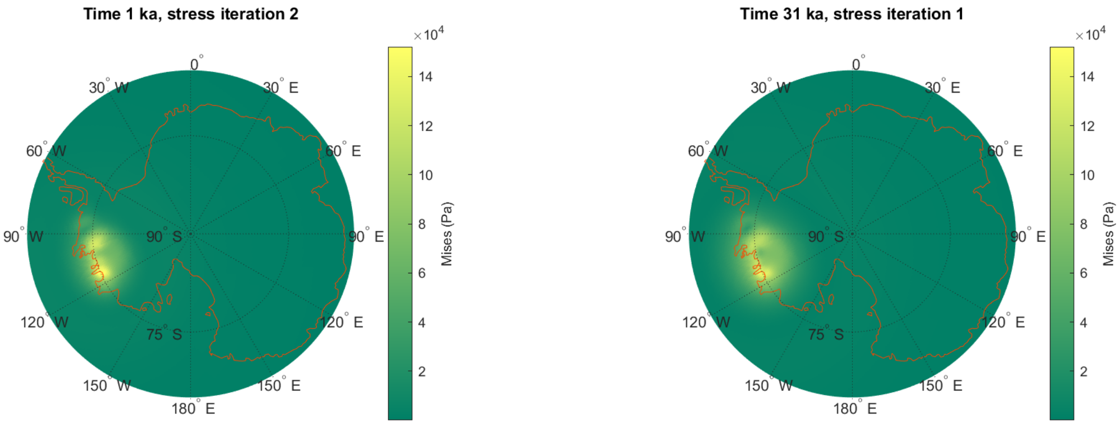


Figure 4.6: Mises stress at two different time snapshots, stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2, for a depth range of 145 to 160 km, showing the slight stress decrease.

When a higher depth value range of 545 to 550 km is used, Figure 4.7 and Figure 4.8 show a different trend compared to the lower depths with a net decrease in the stress value between timestep 1 and 2. This is also mirrored for example by component S11 from Chapter A.6.3, which decreases in absolute value. It is interesting to note that at this higher depth range the Mises stress decrease is continuous, being observable both between stress iterations 1 and 2 of timestep 1 and stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2. This decrease in stresses can be ascribed to the higher depths meaning a larger distance from the ice load causing the glacial stresses to be smaller compared to the lower depth range. Smaller glacial stresses (by one order of magnitude) which cannot compensate the faster relaxation lead to a continuous decrease in stresses even though the same ambient stresses are being added at each stress iteration. The decrease in Mises stress is shown in Figure 4.7.

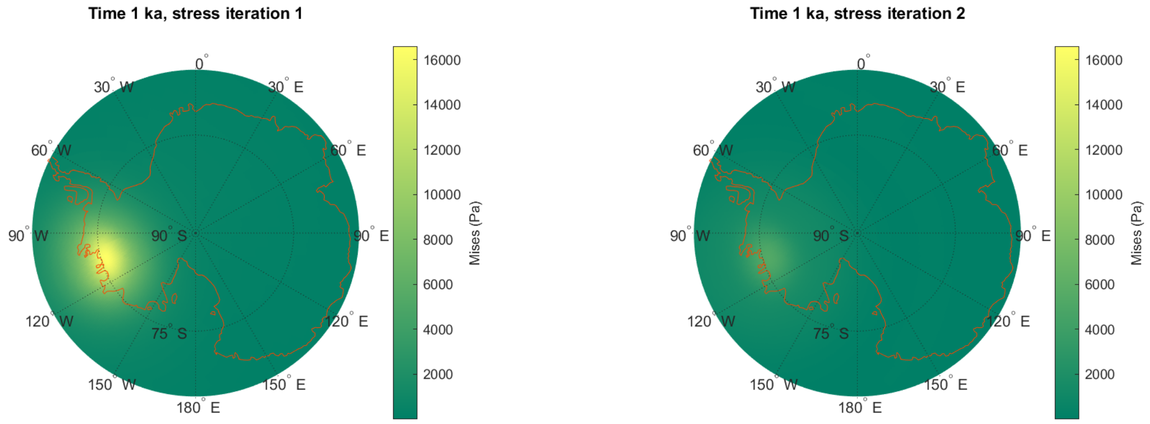


Figure 4.7: Mises stress for two different time snapshots, one at stress iteration 1 of timestep 1 and one at stress iteration 2 of timestep 1 for a depth range of 545 to 550 km.

The decrease between stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2 is shown in Figure 4.8. The Mises stress was already (slightly) reduced between these two time snapshots at low depths, which means that for this high depth range the distance with respect to an ice load reapplied with the same value after time of 30 ka leads to the Mises stress being reduced to near zero values as shown in Table 4.3 too. Moreover, it can be observed that the Mises stress is around one order of magnitude smaller compared to a lower depth range.

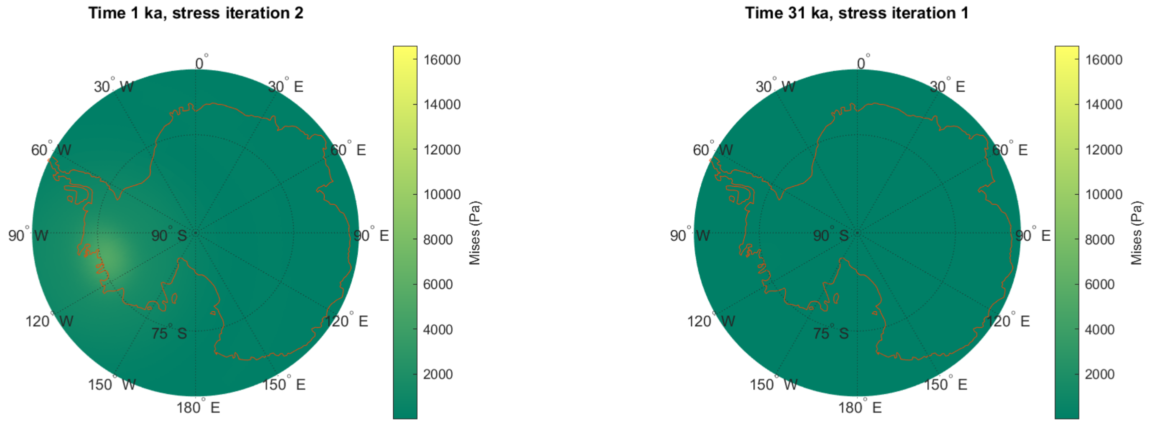


Figure 4.8: Mises stress for two different time snapshots, one at stress iteration 2 of timestep 1 and one at stress iteration 1 of timestep 2 for a depth range of 545 to 550 km.

All of the plots show large value differences at stress iteration 1 of timestep 1 when using a wet rheology which quickly become much smaller for stress iteration 2 already, with more precise values from Section 4.4 showing differences of more than double the stress values themselves for the first stress iteration becoming zero for all stress components when comparing dry and wet rheology tables. This strong difference at the first stress iteration which quickly decreases could be explained by a small sensitivity of the stresses used for this thesis work and viscosity to rheology which then transfers to deformations, as will be discussed further in Chapter 5, or by differences in rheology parameters not being very large themselves. Both options seems to be true. The inability of the FEM model to see differences in a power law rheology has already been described in the work by Wu (1992) using pre-coupled Mises stresses in Section 1.1, so the same could be happening when adding background stresses at each stress iteration as done here. The latter option also seems to be true as it can be observed in the plots of Section B.3, which show that B_{disl} is the same while B_{diff} does not substantially change between dry and wet rheology rheology, being around the same order of magnitude for the AOI in both cases. Difference plots have been added for B_{diff} only for both depth ranges to better represent its variation between a dry and wet rheology. The initial difference in Mises stress has been displayed in Figure 4.9 and the absence of such difference at stress iteration 2 has been displayed Figure 4.10. This reduction happens for both depth ranges, so it has not been necessary to show it again for these higher depth range case. Again, the same goes for component S11 from Section B.1.

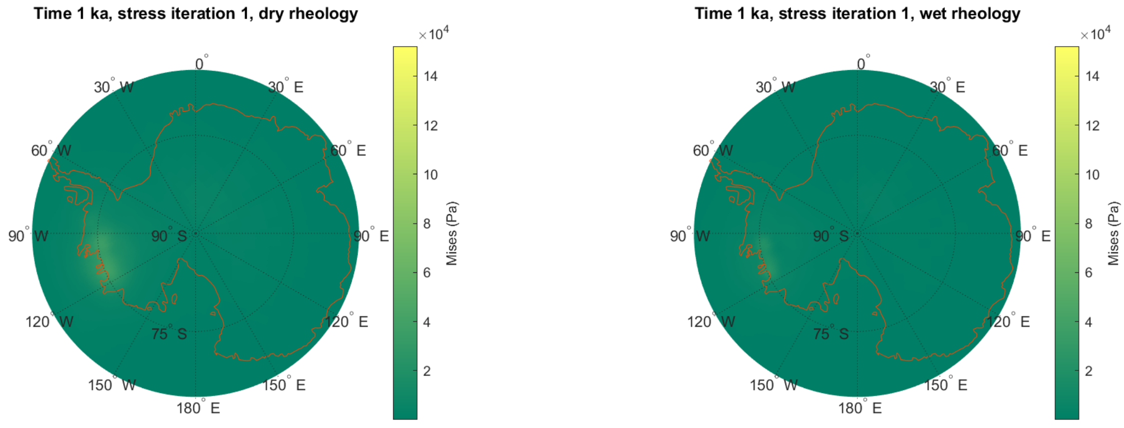


Figure 4.9: Comparison of the results from the first stress iteration of the first time step for the Mises stress for different rheologies, with a background stress of 1 MPa and for a depth range of 145 to 160 km, showing discernible differences in the plotted values.

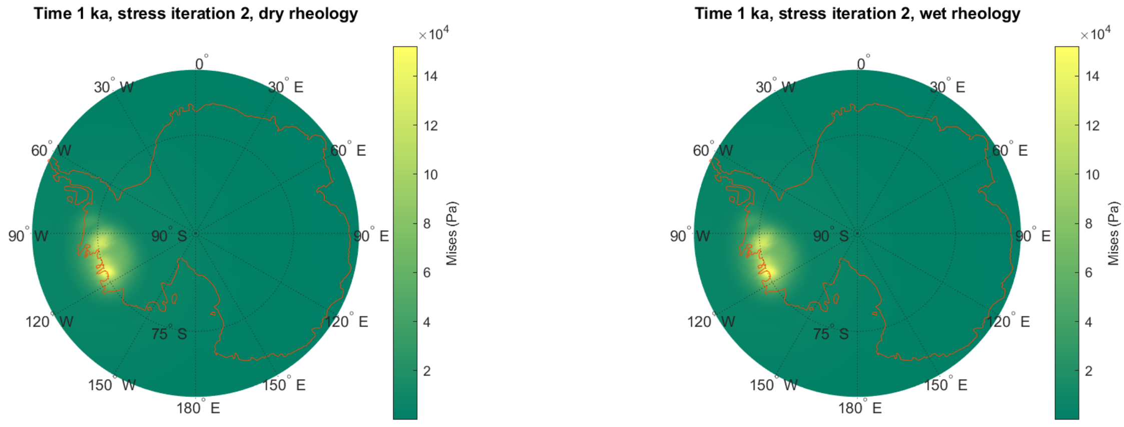
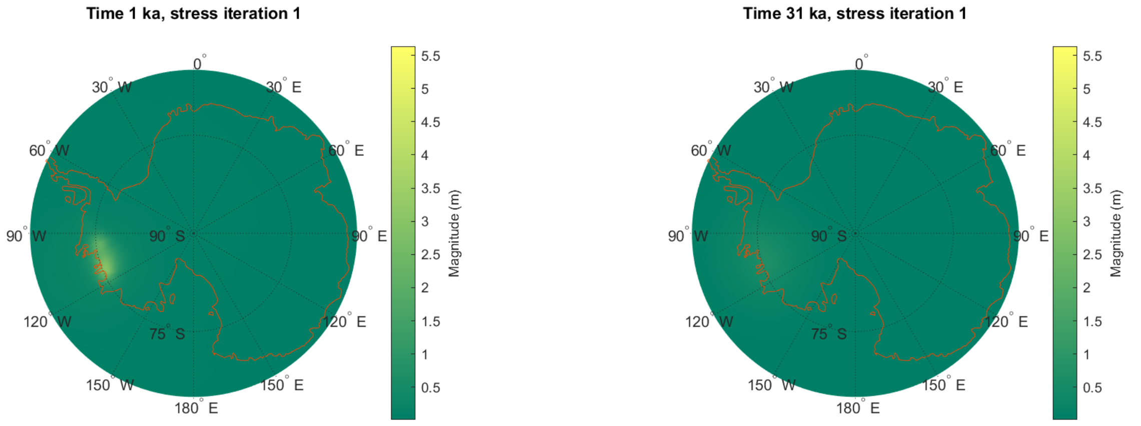


Figure 4.10: Comparison of the results from the second stress iteration of the first time step for the Mises stress for different rheologies, with a background stress of 1 MPa and for a depth range of 145 to 160 km, showing that the differences have strongly reduced.

The S11 and S12 stress components plots together with their time evolution and short explanation of their behavior can be found in Section B.1.

4.2. Deformation plots

Deformation plots show a different trend with respect to the stress plots, with deformations decreasing until stress iteration 2 of timestep 1 as shown by Figure 4.11. These plots have only been generated for the deformation magnitude, which is almost completely made up of the vertical deformation component which has been displayed in some plots in Section B.2 for comparison. The net reduction of the deformation magnitude is again an opposite trend compared to the one observed in Figure 4.2 where the deformations increase between two timesteps, and it might seem strange that a lower viscosity as will be shown in Section 4.3 leads to smaller deformations. However, this can be again explained by a faster relaxation, which means that the Earth is dissipating stresses at each timestep by also deforming at every stress iteration, and through this action the remaining deformations in stress iteration 1 of timestep 2 are smaller. For this reason, the deformations are also in general smaller compared to the baseline case.

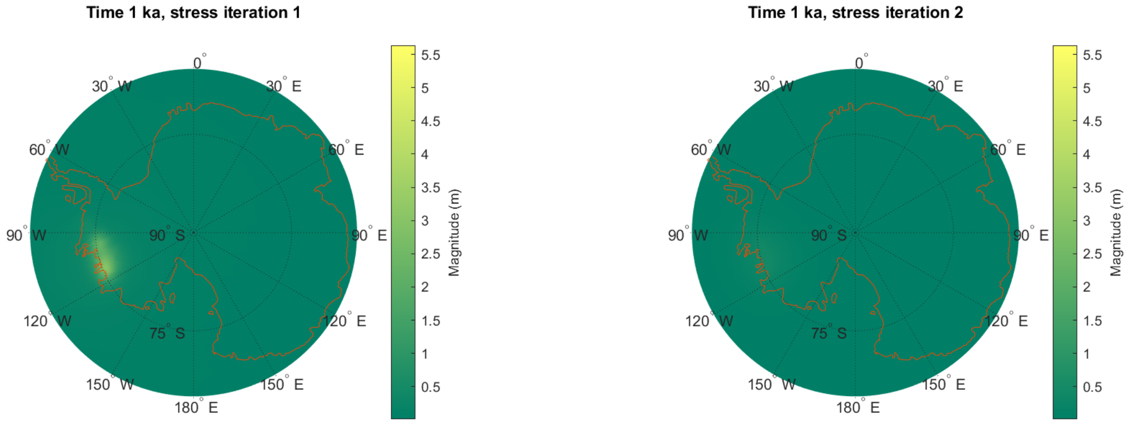


Time 1 ka, stress iteration 1

Time 31 ka, stress iteration 1

Figure 4.11: Deformation magnitude for two different time snapshots, one at stress iteration 1 of timestep 1 and one at stress iteration 1 of timestep 2 for a depth range of 96 to 123 km.

However, deformations do not constantly decrease, with a deformation decrease between stress iteration 1 and 2 of timestep 1 shown in Figure 4.12 and an increase between stress iteration 2 of timestep 1 (which has the same values as stress iteration 3) and stress iteration 1 of timestep 2 as shown in Figure 4.13. This evolution mirrors the one described for stresses in the previous section (increase and then decrease for stresses). The decrease between stress iteration 1 and 2 of timestep 1 can be explained by the high relaxation which leads to almost all the deformations being dissipated in the very first stress iteration, leaving smaller deformations at stress iterations 2.



Time 1 ka, stress iteration 1

Time 1 ka, stress iteration 2

Figure 4.12: Deformation magnitude for two different time snapshots, one at stress iteration 1 of timestep 1 and one at stress iteration 2 of timestep 1 for a depth range of 96 to 123 km.

When considering stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2, reapplying the same ice load with the same value and after a long time of 30 ka causes the deformations to increase, albeit very slightly due to the higher relaxation. Because this decrease and then increase is also observed at higher depths and it is easier to visualize it there due to the different scale used for the two depths which at higher depths is more sensitive to small variations, it has been decided to display the deformation increase between stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2 for the 502 to 510 km range as shown in Figure 4.13. The fact that deformation evolution is the same for both depth ranges is different with respect to the Mises stress, which constantly decreases with time at high depths. It has already been explained why the Mises stress behaves that way, while for deformations the same explanation applies as for the lower depth range but with a smaller order of magnitude due to the distance from the ice load. Figure 4.13 also shows that again in the same fashion as stresses, the entire evolution of deformations with time at higher depths has a less detailed distribution at the different time snapshots due to the distance from the load and its display can therefore be skipped. Only the first stress iterations of timestep 1 and 2 will be displayed here in order to show how much smaller the deformations are at depth (by around a factor of 10), due to the much smaller influence of the ice load.

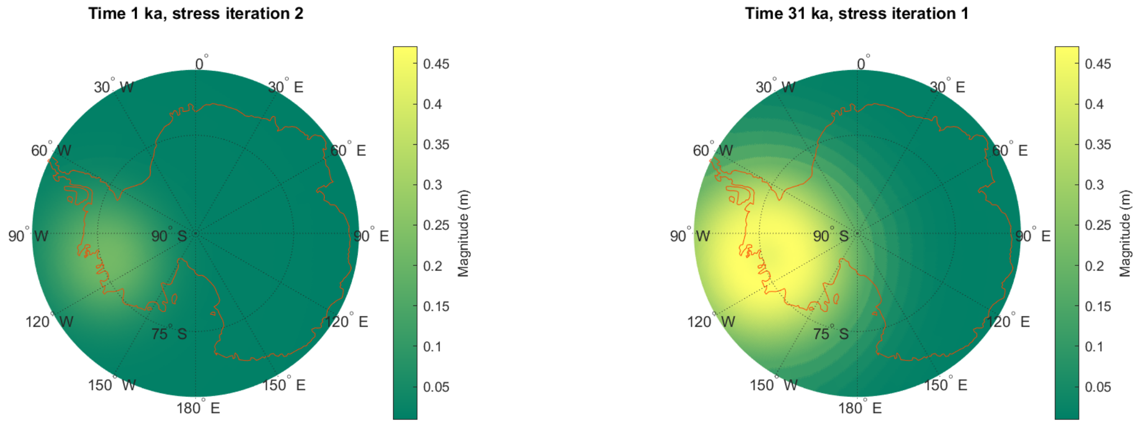


Figure 4.13: Deformation magnitude for two different time snapshots, one at stress iteration 2 of timestep 1 and one at stress iteration 1 of timestep 2 for a depth range of 502 to 510 km.

Using a wet rheology, in an opposite trend with respect to stresses, increases deformations for the same time snapshot as expected due to a lower general viscosity, but again with a noticeable effect only for the first stress iteration of the firs timestep, as displayed in Figure 4.14.

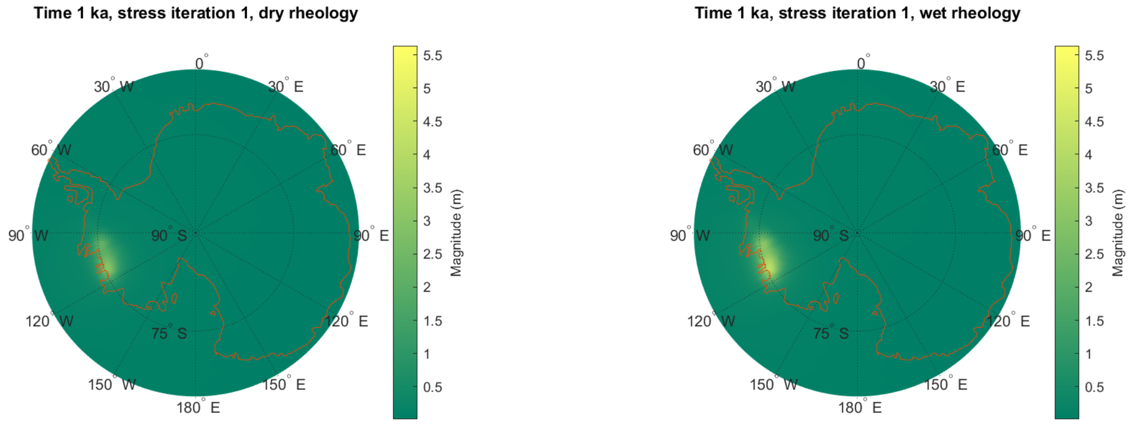


Figure 4.14: Comparison of the results from the first stress iteration of the first time step for the deformation magnitude between results for different rheologies, with a background stress of 1 MPa and depth range of 96 to 123 km, showing discernible differences in the plotted values.

The differences between dry and wet rheologies quickly go down as for the stresses and this is shown in Figure 4.15, where it can be seen that the values for both plots are the same.

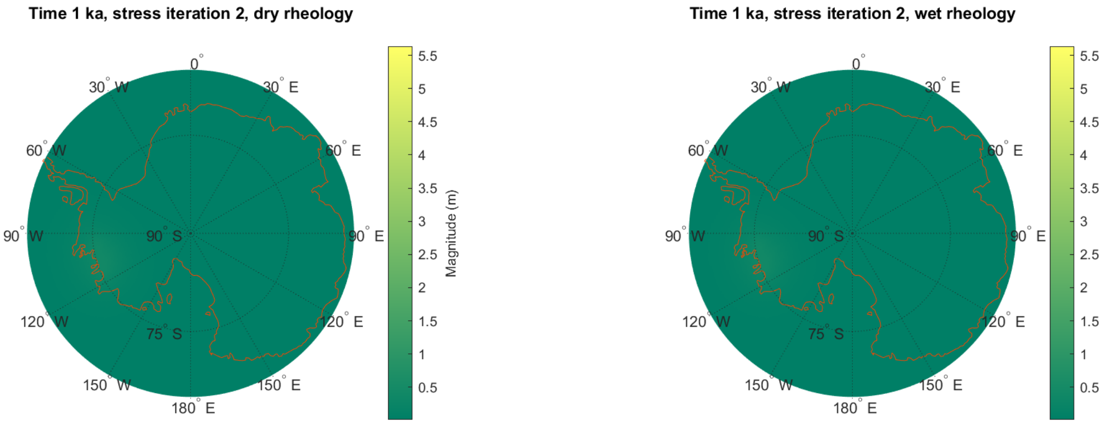


Figure 4.15: Comparison of the results from the second stress iteration of the first time step for the deformation magnitude between results for different rheologies, with a background stress of 1 MPa and depth range of 96 to 123 km, showing that the differences have strongly reduced.

4.3. Viscosity plots

Viscosity plots show a pattern strongly dominated by the values of the B coefficients, with substantial lateral variations in viscosity between West Antarctica and East Antarctica. These coefficients have been calculated using the formulation displayed in Equation 1.3, from van der Wal et al. (2013) which in turn uses values for p, R, A, E, V from Hirth and Kohlstedf (2003). The lateral variation of the B coefficients has been suggested for example by Kaufmann et al. (2005) and demonstrated to provide a better fit to GPS-measured uplift rates in the study by van der Wal et al. (2015). The B coefficients, as shown by their plots in Chapter A.6.3, obey this lateral variation between West and East Antarctica especially at low depths and consequently the viscosity also follows this dichotomy. The evolution of viscosity with time has been plotted for the lower depth range of 145 to 160 km at two different snapshots in time. The viscosity time evolution closely follows that of stresses for low depths, while this close tie is lost at higher depths. A reduction in viscosity by around 2 orders of magnitude as displayed in Figure 4.16 which mirrors the increase in stress is present between stress iteration 1 of timestep 1 and stress iteration 1 of timestep 2. A much smaller viscosity reduction, mirroring the very slight stress increase again due to the reapplication of the ice load, is found between stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2. Because of this, it has not been necessary to plot, as in the case of the stresses, these latter additional two snapshots and only the results for stress iteration 1 for both timesteps 1 and 2 have been generated.

In the 145 to 160 km depth range, it is not only possible to observe the influence of the B coefficients on the viscosity distribution, but also the one due to the Mises stress displayed in Section 4.1 resulting from the ice load. This influence is translated to a dark green area in West Antarctica denoting a lower viscosity compared to the surroundings in a very localized area between 90 and 120 ° W towards the Antarctic coast. The observable "tessellation" in the right-hand side of the plots is due to the interpolation method used, which required a 3D interpolation over the cartesian coordinates of every point in the plot using the 'nearest' method provided by MATLAB, as other interpolation method all failed. The plots for the lower depth range in Figure 4.16 show a total reduction in viscosity, especially in the West Antarctic area. A figure comparison with the baseline case displayed in Figure 4.3 shows that viscosity is generally higher, which can be explained by the larger Mises stress causing a higher viscosity from Equation 1.2 due to the numerator being larger than the denominator. However, the combination between the background and ice load stresses actually introduces an area of low viscosity and higher relaxation compared to the surroundings. This area of low viscosity and increased relaxation with respect to the baseline seems to be strongly localized in the area where the ice load is applied (the "stain" described earlier). The viscosity decreases of around 2 orders of magnitude between timesteps 1 and 2 when introducing stress coupling while the in the base case it was basically the same. The low viscosity and high relaxation be explained by considering that the sum of the glacial and background stresses in an area where the glacial stresses are significant reduces the viscosity due to the Mises stress being so large that from Equation 1.2 it leads to a larger increase in denominator compared to the numerator due to n=3.5. This is also proven by the fact that, for a higher depth where the ice stresses are smaller, the viscosity reduction due to the introduced relaxation disappears and the values are almost unchanged between timesteps 1 and 2.

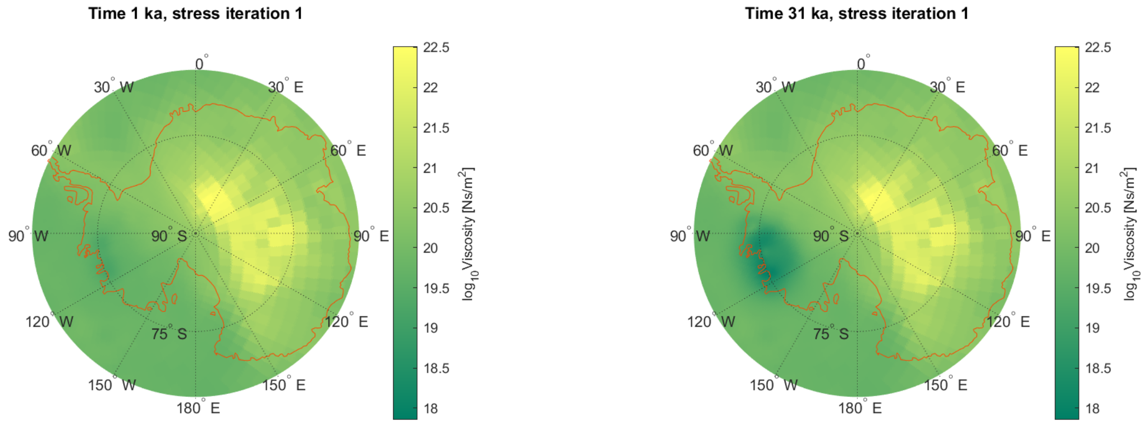


Figure 4.16: Viscosity for two different time snapshots, one at stress iteration 1 of timestep 1 and one at stress iteration 1 of timestep 2 for a depth range of 145 to 160 km.

At higher depth the plots change a lot, with no discernible time evolution and a higher viscosity in the area where the load was present which is also displayed in Table 4.3. However, the influence of the B coefficients (which can be compared from Section B.3) is still visible and actually it is even more evident as the "stain" on the left of the plot left by the Mises stress, which denoted its influence, goes away. This is easy to explain due to the distance from an ice load which as said before is very localized, meaning unable to exert a strong influence even at these middle depths.

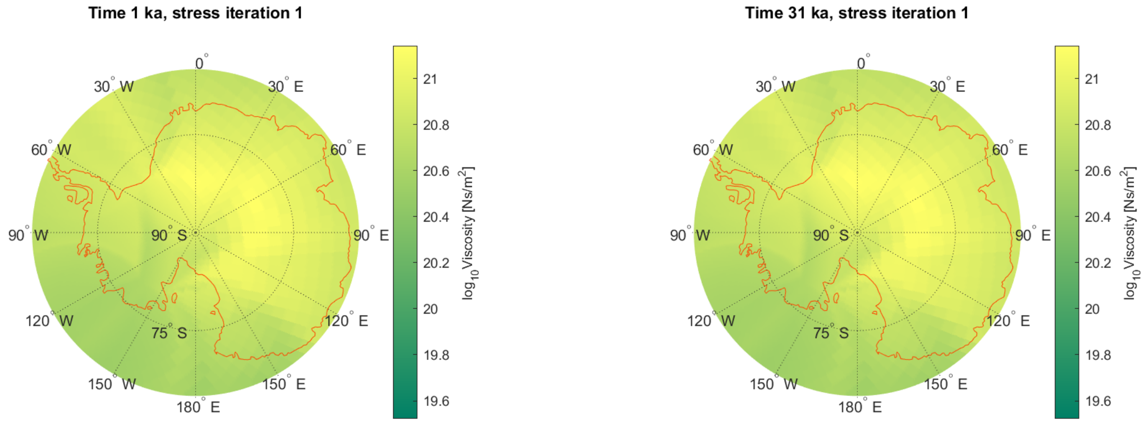


Figure 4.17: Viscosity for two different time snapshots, one at stress iteration 1 of timestep 1 and one at stress iteration 1 of timestep 2 for a depth range of 545 to 550 km.

As expected, using a wet rheology causes a general decrease in viscosity, which is once again especially evident at stress iteration 1 of timestep 1 as displayed in Figure 4.18 by looking at how the entire plot leans towards lower viscosity values.

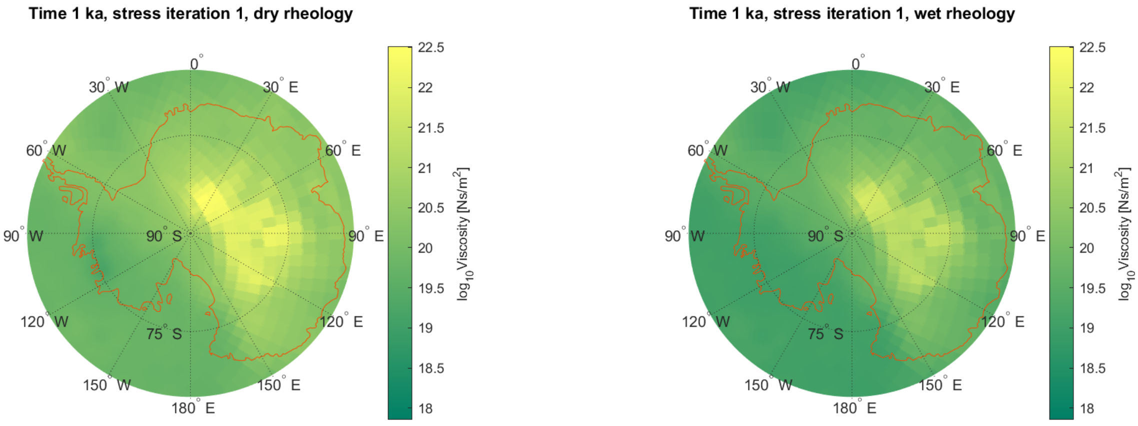

Time 1 ka, stress iteration 1, dry rheology
Time 1 ka, stress iteration 1, wet rheology
0°
30° W
30° E
60° W
60° E
90° W
90° S
90° E
120° W
75° S
120° E
150° W
150° E
180° E
22.5
22
21.5
21
20.5
20
19.5
19
18.5
18
$\log_{10}$Viscosity [Ns/m^2]

Figure 4.18: Comparison of the results from the first stress iteration of the first time step for the viscosity results for a dry and wet rheology, with a background stress of 1 MPa and depth range of 145 to 160 km, showing discernible differences in the plotted values.

This difference is then reduced, but still present, when the stress iteration is incremented to 2 as shown in Figure 4.19. This is a different situation with respect to stresses and deformations, whose differences between dry and wet rheology just turn to 0 at stress iteration 2. This can be better observed in Section 4.4, and is due to the fact that even though the Mises stress is the same between dry and wet rheology, the B coefficients change and with them the viscosity which as shown by Equation 1.2 is directly calculated from them. Notably, this also seems to affect the area of high relaxation, with it being less distinguishable with respect to the surroundings when using a wet rheology. This was to be expected, since a higher B_{diff} due to a wet rheology (as per the plots in Section B.3 reduces the viscosity everywhere and a lower Mises stress as shown in Section 4.1 means that the viscosity is going to be higher in the area of high relaxation.

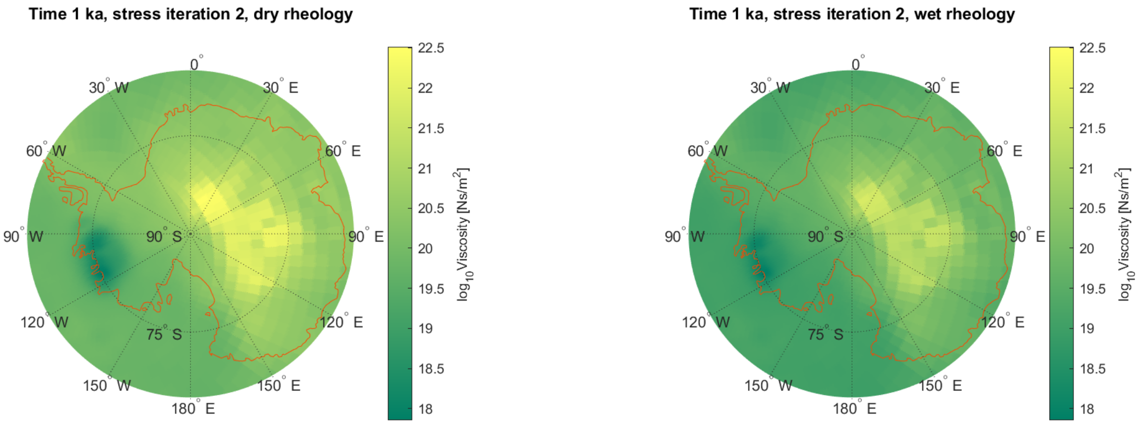

Time 1 ka, stress iteration 2, dry rheology
Time 1 ka, stress iteration 2, wet rheology
0°
30° W
30° E
60° W
60° E
90° W
90° S
90° E
120° W
75° S
120° E
150° W
150° E
180° E
22.5
22
21.5
21
20.5
20
19.5
19
18.5
18
$\log_{10}$Viscosity [Ns/m^2]

Figure 4.19: Comparison of the results from the second stress iteration of the first time step for the viscosity results for a dry and wet rheology, with a background stress of 1 MPa and depth range of 145 to 160 km, still showing differences in the plotted values.

4.4. Tables for element with the highest Mises stress

This section contains tables generated through the element_tracker function and show the evolution of the element with highest Mises stress at every stress iteration and the corresponding viscosity. All the tables have been produced for the specific area of West Antarctica where the load is applied, with a latitude between 65 and 90 degrees south and a longitude of 60 to 150 degrees west. Because the Mises stress is much higher there compared to the surrounding area as Section 4.1 shows, using the entire longitude range of -180 to 180 ° would not have changed the results displayed in the tables. Another reason for the tables being restricted to West Antarctica is that this area tends to have a lower viscosity compared to the eastern region as also discussed by van der Wal et al. (2015). The plots have instead been generated over the entirety of Antarctica to give a better overview of the entire area. The tables displayed below are for two different background stresses (0.008 and 1 MPa), dry and wet rheology and two depth ranges, one at shallow depth (between 145 and 160 km) and one at intermediate depth (between 545 and 550 km). It has been decided to generate tables for two different depth ranges to show more precisely how the quantities displayed in the plots tend to change when farther away from the ice load. Comparing rheologies is also important as shown in the plots from the previous sections. Changing rheology means in fact changing the B_{diff} and B_{disl} parameters, which in turn influence the way the Earth responds to the coupled Mises stress due to the change that B_{diff} and B_{disl} cause to the viscosity as per Equation 1.2.

An important disclaimer to make is that even if some of the results of the tables do not seem to agree with the plots, it has to be considered that the element with highest Mises stress does not necessarily coincide with the elements with the other highest stress values in absolute value. For example, S11 increases in absolute value as shown by a comparison between Figure B.1 and Figure B.2 from Section B.1, while it might seem from the tables that it has reduced. However, the point of highest Mises stress and the one of highest S11 stress have a distance of around 10 degrees longitude between each other, so the point with highest S11 in value is not reported in the table.

These tables offer a lot of insight into the model behaviour with stress coupling and are mainly aimed at giving a more in depth look at the model evolution when using different settings or looking at different depths, while the plots have more of a function of comparison with respect to the baseline case. First of all, two tables for the same rheology (dry) and depth range (145 to 160 km), but using two different background stresses of 0.008 and 1 MPa, will be compared. These tables are Table 4.1 and Table 4.2 respectively. The two tables again show that changing the background stresses does not have any effect on the model outputs at any given stress iteration, only on the coupled stresses due to the different module of the ambient stresses. As explained earlier, this could be due to the background stress differences not being large enough, the viscosity change not being significant or the fact that the added stresses are exactly the same for every element.

The tables displayed below can answer this uncertainty by looking at the $1/3 * B_{diff}$, $1/3 * B_{disl} * q^(n-1)$ and viscosity values which have also been included in the tables under the "Bdiff", "Bdisl" and "Viscosity" labels respectively. The first two factors are a measure of the weight that the diffusion and dislocation creeps respectively have in the creep law. It is easier to evaluate these factors compared to the creep law terms B_{diff} and $B_{disl} * q^(n-1)$ once simplified by q given how small the terms themselves are. Both these factors and the viscosity are important for attempting to explain the fact that results do not change with the background stress, because this can be explained in two ways. Either the viscosity is too large, with a magnitude change due to background stresses too small, to see results change in the timescales used for this thesis work or the dislocation creep term in Equation 1.1 is too small for a change in background stresses to have any influence on the model results. The values shown in the tables seem to be inconclusive since the Bdisl factor seems to be, as model output, smaller than Bdiff at lower depths and larger than that at higher depth due to the Mises stress decrease as the depth increases. This means $B_{disl} * q^(n-1)$ larger than Bdiff at low depths, which would mean that the second potential explanation is not verified and smaller at high depth, which would instead verify this explanation. The viscosity output is also ambiguous, since it is not very high at low depths, reaching a minimum order of magnitude of around e17, and increases with depth to e20 together with a decrease by around 2 orders of magnitude between smaller and larger background stresses. This would mean that the first explanation is again not verified at low depths but could be true, with a minimum viscosity of order of magnitude e20 and a magnitude change which is not influential enough to modify it, at high depths. However, there is an important consideration to make, which is that these tables are only relative to the element with the highest Mises stress, meaning in the load application area. This means considering the case where the aforementioned $1/3 * B_{disl} * q^(n-1)$ and viscosity are the smallest. If the plots are also considered in order to find an explanation to the problem of results not changing, it can be easily seen that the Mises stress is actually close to 0 over the vast majority of Antarctica except for the ice load area with values orders of magnitude smaller than what is found in the load application area. The viscosity is also of an order of magnitude larger than e21 almost everywhere except the ice load application area at low depths and homogeneously high around e21 at high depths, being slightly lower for a wet rheology. This means that for the vast majority of points, the viscosity is really high and a change due to background stresses will not matter anyway due to the really long timescale that would be necessary to see any effect of the stress change. Moreover, the dislocation creep part would be, due to the small Mises stress, much smaller than the diffusion part, which also leads to the results not changing with background stresses.

Moreover, the tables again demonstrate that all quantities plotted earlier converge between stress iterations 2 and 3 already in a single timestep. The tables display "output" before "input" because output is the data read from the model, while input is the coupled stresses given to ABAQUS. Because of this, the input rows contain stress and viscosity that change with the background stresses, while the output rows are the ones not changing due to the aforementioned explanation. Stresses from the model need to be read in order to couple ambient stresses to them, so with this notation the model output precedes the model input. The way the ambient stresses are added to the ones coming from the model is displayed in Equation 2.16.

Step, stress iteration	Lat [deg]	Lon [deg]	Mises [Pa]	S11 [Pa]	S12 [Pa]	Bdiff	Bdisl	Viscosity [Pa s]
1,1 output	-75.4	-112.5	39197.3	-48404.1	146.7	5.14E+19	2.26E+19	1.57E+19
1,1 input	-74.7	-110.7	51337.6	-24787.8	16343.1	5.1E+19	1.14E+19	9.32E+18
1,2 output	-75.3	-114.4	151813	-34695.7	-8136	5.19E+19	7.78E+17	7.67E+17
1,2 input	-74.8	-113.5	162295.4	9689	18281.8	5.14E+19	6.51E+17	6.42E+17
1,3 output	-75.3	-114.4	151813	-34695.7	-8136	5.19E+19	7.78E+17	7.67E+17
1,3 input	-74.8	-113.5	162295.4	9689	18281.8	5.14E+19	6.51E+17	6.42E+17
2,1 output	-74.8	-113.5	141718	-51723.2	-17840.1	5.14E+19	9.13E+17	8.97E+17
2,1 input	-74.6	-112.5	156251.1	-15443.6	46646.4	5.12E+19	7.12E+17	7.02E+17

Table 4.1: Table displaying the temporal evolution of the element with the highest Mises stress for a longitude range of 60 to 150 °W, latitude range of 65 to 90 °S, depth range of 145 to 160 km and dry rheology, with a background stress of 0.008 MPa.

Step, stress iteration	Lat [deg]	Lon [deg]	Mises [Pa]	S11 [Pa]	S12 [Pa]	Bdiff	Bdisl	Viscosity [Pa s]
1,1 output	-75.4	-112.5	39197.3	-48404.1	146.7	5.14E+19	2.26E+19	1.57E+19
1,1 input	-74.5	-109.7	2027855	934046.2	837884.4	5.1E+19	1.16E+15	1.16E+15
1,2 output	-75.3	-114.4	151813	-34695.7	-8136	5.19E+19	7.78E+17	7.67E+17
1,2 input	-74.7	-110.7	2111514	948028.8	867637	5.1E+19	1.05E+15	1.05E+15
1,3 output	-75.3	-114.4	151813	-34695.7	-8136	5.19E+19	7.78E+17	7.67E+17
1,3 input	-74.7	-110.7	2111514	948028.8	867637	5.1E+19	1.05E+15	1.05E+15
2,1 output	-74.8	-113.5	141718	-51723.2	-17840.1	5.14E+19	9.13E+17	8.97E+17
2,1 input	-74.7	-110.7	2116972	935756.8	872590.1	5.1E+19	1.04E+15	1.04E+15

Table 4.2: Table displaying the temporal evolution of the element with the highest Mises stress for a longitude range of 60 to 150 °W, latitude range of 65 to 90 °S, depth range of 145 to 160 km and dry rheology, with a background stress of 1 MPa.

Using a higher depth and comparing two tables with the same rheology and background stress, in this case Table 4.2 and Table 4.3, shows a higher viscosity in the area of highest stress due to the higher depth and a reduction in stresses due to the smaller influence of the ice load. The tables show that viscosity values are almost constant with time at higher depths, already observed in Figure 4.17. This is true for both dry and wet rheologies, with the main difference being a generally lower viscosity in West Antarctica for the latter. An explanation for this could be the larger influence the glacial load stresses have on the viscosity calculation with respect to the background ones, which also explains why the viscosity change with background stresses is smaller as depth increases: the background stress is always the same for all elements, but the one deriving from the ice load is not and decreases with depth. Because the relative influence of the glacial load is higher in the combined stress, a change in combined stress is more influenced by a change in glacial stress, whose influence decreases with depth. Therefore, the influence of the combined stress also decreases with depth.

Step, stress iteration	Lat [deg]	Lon [deg]	Mises [Pa]	S11 [Pa]	S12 [Pa]	Bdiff	Bdisl	Viscosity [Pa s]
1,1 output	-75.9	-106	16561.1	-14052.4	480.7	4.84E+20	4.33E+23	4.84E+20
1,1 input	-72.8	-103.3	2006799	966515.6	801535.5	4.7E+20	2.56E+18	2.54E+18
1,2 output	-75.9	-106	5427.6	-19957.6	135.1	4.84E+20	7.03E+24	4.84E+20
1,2 input	-70.3	-101.9	2000956	961207.4	742413.8	4.68E+20	2.57E+18	2.56E+18
1,3 output	-75.9	-106	5427.6	-19957.6	135.1	4.84E+20	7.03E+24	4.84E+20
1,3 input	-70.3	-101.9	2000956	961207.4	742413.8	4.68E+20	2.57E+18	2.56E+18
2,1 output	-75.6	-105	70.1	-1307.1	0.6	4.86E+20	3.74E+29	4.86E+20
2,1 input	-82.8	-110.2	2000009	992130.8	962014.1	4.83E+20	2.7E+18	2.68E+18

Table 4.3: Table displaying the temporal evolution of the element with the highest Mises stress for a longitude range of 60 to 150 °W, latitude range of 65 to 90 °S, depth range of 545 to 550 km and dry rheology, with a background stress of 1 MPa.

The new rheology with a smaller grain size and a higher water content also causes the viscosity to be lower for all times, as it causes B_{diff} to be larger, and all the stresses to increase only for the very first stress iteration. This leads to another important observation which has already been displayed in Figure B.8 and Figure B.9, which is that the Mises stress and other components show no difference from stress iteration 2 between dry and wet rheology. This can especially be seen when comparing different rheology tables such as Table 4.2 and Table 4.4. As said before this is not true for the viscosity because even if the Mises stress is the same from stress iteration 2 onwards, the viscosity is directly calculated from B_{diff} and B_{disl} which are different, leading to smaller values with a wet rheology for all stress iterations.

Step, stress iteration	Lat [deg]	Lon [deg]	Mises [Pa]	S11 [Pa]	S12 [Pa]	Bdiff	Bdisl	Viscosity [Pa s]
1,1 output	-75.6	-112.5	24332.1	-18178.2	801.1	1E+19	7.43E+19	8.84E+18
1,1 input	-74.7	-110.7	2012346	954407.4	833732.9	9.97E+18	1.18E+15	1.18E+15
1,2 output	-75.3	-114.4	151813	-34695.7	-8136	1.01E+19	7.78E+17	7.23E+17
1,2 input	-74.7	-110.7	2111514	948028.8	867637	9.97E+18	1.05E+15	1.05E+15
1,3 output	-75.3	-114.4	151813	-34695.7	-8136	1.01E+19	7.78E+17	7.23E+17
1,3 input	-74.7	-110.7	2111514	948028.8	867637	9.97E+18	1.05E+15	1.05E+15
2,1 output	-74.8	-113.5	141718	-51723.2	-17840.1	1E+19	9.13E+17	8.37E+17
2,1 input	-74.7	-110.7	2116972	935756.8	872590.1	9.97E+18	1.04E+15	1.04E+15

Table 4.4: Table displaying the temporal evolution of the element with the highest Mises stress for a longitude range of 60 to 150 °W, latitude range of 65 to 90 °S, depth range of 145 to 160 km and wet rheology, with a background stress of 1 MPa.

4.5. Data for realistic stress definition

One of the results of this thesis work has been the description of an approach to define realistic stresses to apply to the model. Realistic here means stresses that have been derived from a systematic literature study and are closer to actual values for the Earth's interior, since the values used here have mainly been used as test and have been chosen as being close to the output stress values (for 0.008 MPa) and on a higher value of 1 MPa closer to the one used in the work by Schmeling (1987). Despite being outside the scope of this thesis, it has been decided to describe this realistic stress definition process here anyway as it can be a starting point for research in this direction.

The chosen approach for this realistic stresses definition can be summarized in the following steps:

1. Find a map of velocities for the AOI, since they are related to plate traction as discussed by Osei Tutu et al. (2018) and velocities can be used to approximately derive normal and shear stresses as shown in Equation 4.1 from Höink et al. (2012), then use this map to average these velocities over the West Antarctic area in module and direction;
2. Apply these average values for West Antarctica to all elements and nodes over the entire model, at surface and at depth;
3. These average values are then are scaled by depth as explained later in more detail;

4. These scaled stress values are then summed to the old ones and the Mises stress is recalculated;
5. The new Mises stress is used as an input to the new model job definition.

The maps used to ultimately define stress-related velocities due to plate motion over Antarctica come from Osei Tutu et al. (2018). While additional velocity data could be obtained from the work of Bredow and Steinberger (2020) which presents findings about vertical mantle flow, such data is a side result and not the main object of study of this paper and, in general, no verified vertical flow map has been produced until now. However, the former study gives the global traction map over the Earth while the latter gives the vertical flow over Antarctica and more detailed horizontal flow maps, so they can be compared for consistency. The plate motion velocities are shown in Figure 4.20 and the additional data with vertical flow is shown in Figure 4.21.

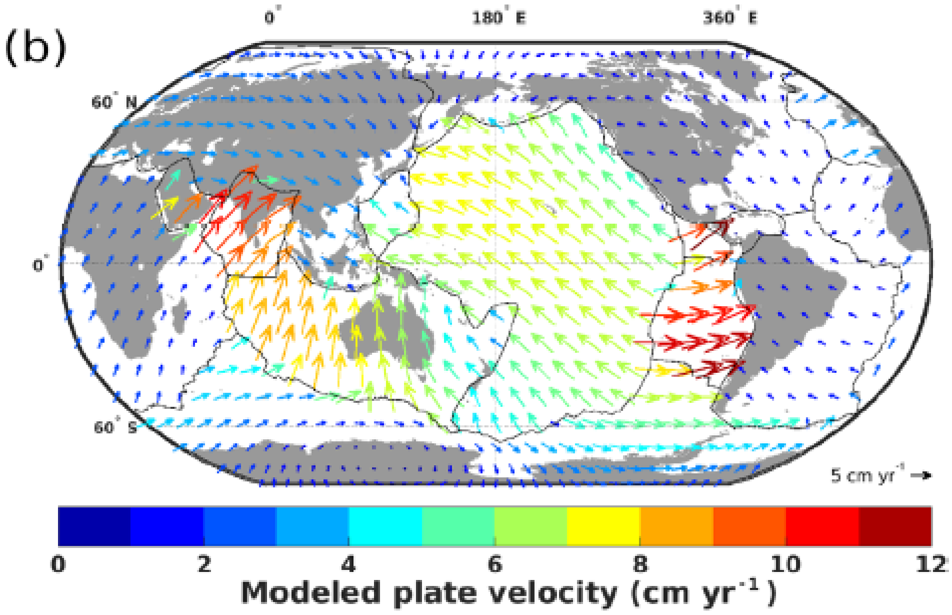


Figure 4.20: Map of crustal velocities related to generation of stress and traction, from the work by Osei Tutu et al. (2018).

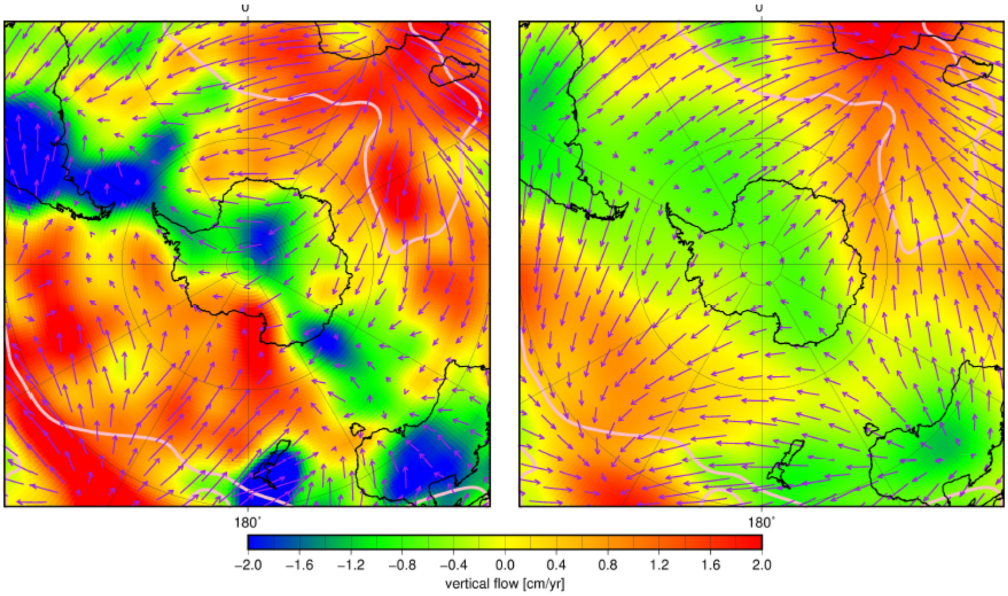


Figure 4.21: Map of horizontal (arrows) and vertical (color scale) mantle flow at a 650 (left) and 2650 (right) km depth, from the work by Bredow and Steinberger (2020).

The two maps more or less coincide when taking into account this flow direction at shallow depths, and the flow velocity can be averaged over West Antarctica from NE to SE with an angle of approximately 60° with respect to the meridian direction and a corresponding crustal velocity of ≈ 2.5e-10 cm/yr. Because the maps are obviously given using a geographical coordinate system, the stresses resulting from crustal velocity will also be in a geographical coordinate system, while ABAQUS only works with cartesian vectors and tensors. However, this is not a problem, because a transformation sequence had to be devised as illustrated in Section 2.1 precisely to convert results from the ABAQUS to the geographical coordinate system to map those results. Therefore, the inverse transformation will be applied to go back to the ABAQUS reference system.

After determining the average vector and tensor values that can be used over the surface of Antarctica and therefore of the entire model, they need to be scaled with depth as a constant velocity-derived stress everywhere would not be an even remotely acceptable approximation of mantle convection. An approximation of the mantle flow selected for this work comes from Wu (2001), which describes mantle flow as a "roller" between the lithosphere-mantle and core-mantle boundaries. This model has been proven effective in describing mantle convection by Spencer et al. (2019) and Bercovici and Karato (2003). The scaling law for stresses is broken into two parts, one for the normal stresses and one for the shear stresses, and is described in Höink et al. (2012) and displayed in Equation 4.1. In the first equation τ_h is the shear stress, μ_m the mantle viscosity, u_0 is the mean horizontal velocity of the plate (surface velocity) and D is the depth of the convecting mantle. In the second equation, τ_n is the normal stress, $\frac{\lambda}{2D}$ the mantle cell aspect ratio and d_p the plate thickness. The parameter which is going to vary for τ_h is the velocity, starting from a value supposed to be the same as the plate velocity once below the low velocity area right underneath the plates to $-u_0$ at the core-mantle boundary, while τ_n will vary with τ_h.

$$
\begin{aligned}
\tau_h &\sim \mu_m \frac{u_0}{0.5D} \\
\tau_n &\sim \frac{\tau_h \lambda}{2Dd_p}
\end{aligned}
\tag{4.1}
$$

It is understandable that, while these equations provide a powerful tool for the approximation of mantle flow, many parameters need to be defined and this operation itself required additional literature study. The values chosen for each parameter, together with their sources, are explained below.

For simplicity, the values used for the viscosity μ_m will come from the volume-averaged viscosity depicted in Kaufmann and Lambeck (2000), which has only two separate values for the upper and lower mantle. The first value is between 100 and 660 km, while the second one is between 660 and 2886 km. u_0, the surface plate velocity module, has been set to an average value of 2.5e-10 m/s, from data about crustal velocities displayed in Osei Tutu et al. (2018). Its direction has been selected as explained in the previous paragraphs. D, the depth of the convecting mantle, has been set to 2786 km from the difference between the depth of the core-mantle boundary at 2886 km displayed in Schubert et al. (2001) and an average of the depth of the asthenosphere under West Antarctica of 100 km from An et al. (2015). The cell aspect ratio $\frac{\lambda}{2D}$ has been demonstrated to be realistic for large wavelengths (Grigné et al. (2007)), and has been set to 8 from research developed by Hewitt et al. (1980) and Höink and Lenardic (2008). d_p, the plate thickness, has been set to 30 km from an average work derived from the map in An et al. (2015). All these simulation values have been summarized in Table 4.5.

Parameter **Earth Layer**	μ_m **[Pa s]**	$u_0[m/s^2]$	**D [km]**	$\lambda/2D$	d_p **[km]**
Crust		$2.5 \cdot 10^{-10}$			30
Upper Mantle	$3.5 \cdot 10^{20}$		2786	8	
Lower Mantle	$2 \cdot 10^{21}$				

Table 4.5: Table displaying a summary of the parameters used for the scaling coefficients of mantle flow with depth.

All the elements for the calculation of the depth scaling coefficients are now here, and it is therefore possible to calculate the τ_h and, from this, the τ_n stress components. However, the results of this procedure are just two stress components, while there should be more even though the velocity vector, and therefore the stress tensor, are in 2D. The first change to apply to the results generated until now is that these stresses are in the reference system of u_0 and as such need to be transferred to the geographical reference system used in Osei Tutu et al. (2018). This can be accomplished through the use of a coordinate system transformation for a 2D stress tensor. The γ angle in the figure will be the complementary angle to the 60° angle between u_0 and the meridian direction cited before, as shown in Figure 4.22.

In the figure, α is the angle between the velocity and any meridian equal for this work to 60°, γ its complementary and the angle the system has to rotate by, x_{u_0} and y_{u_0} are the 2D axes of the velocity reference system while N and E are the East and North direction of the ENU reference system, corresponding to a longitudinal and latitudinal direction respectively.

The corresponding transformation tensor and transformation equation are displayed in Equation 4.2 and Equation 4.3 respectively, then Equation 4.4 is the extended version of Equation 4.3 displaying all of its components. These equations are based on Kelly (2015) with a slight change of notation: the A_{2D} displayed here will be the Q^T of the paper, but the end transformation equation will be the same. In Equation 4.3, A_{2D} is therefore the transformation tensor for a rotation by angle γ, S_{u_0} is the stress tensor in the velocity reference frame and $S_{geo_{2D}}$ the 2D stress tensor in a geographical reference frame.

$$T_{2D} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \tag{4.2}$$

$$S_{geo_{2D}} = A_{2D} S_{u_0} A_{2D}^T \tag{4.3}$$

$$\begin{bmatrix} S_{\theta\theta} & S_{\theta\phi} \\ S_{\phi\theta} & S_{\phi\phi} \end{bmatrix} = \begin{bmatrix} \cos\gamma & \sin\gamma \\ -\sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} \tau_n & \tau_n \\ \tau_n & \tau_n \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{bmatrix} \tag{4.4}$$

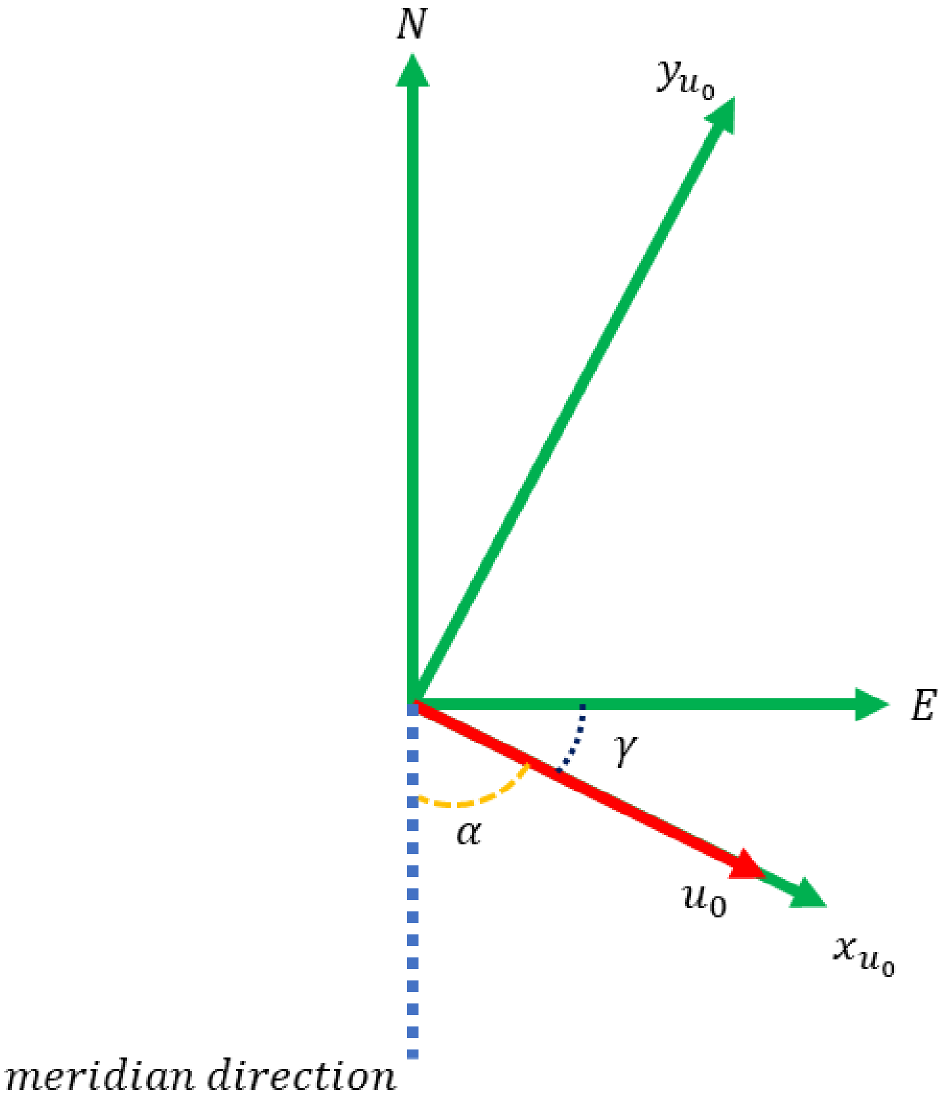


Figure 4.22: Representation of the 2D coordinate transformation to be used for the transformation from velocity to geographical reference frame..

After doing this, it will be possible to apply the inverse coordinate transformation (from ENU to cartesian) with respect to what has been described in Section 2.1. However, a 3-by-3 stress tensor will need to be defined first. In this case, the tensor will have no components on the bottom row and rightmost column, since those correspond to the third dimension but the velocity used to derive the stress components has only two out of three components. This new stress tensor is shown in Equation 4.5.

$$S_{geo} = \begin{bmatrix} S_{\theta\theta} & S_{\theta\phi} & 0 \\ S_{\phi\theta} & S_{\phi\phi} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.5}$$

After this, Equation 4.6 can be applied to obtain the cartesian components.

$$S_{cart} = AS_{ENU}A^T \tag{4.6}$$

Of course, this requires the definition of a different transformation tensor with respect to the one shown in Equation 2.12 since the transformation to perform now is the inverse of the one originally used to plot data, and this new tensor is shown in Equation 4.7.

$$A = \begin{bmatrix} \mathbf{i}_x^T\mathbf{i}_\theta & \mathbf{i}_x^T\mathbf{i}_\phi & \mathbf{i}_x^T\mathbf{i}_R \\ \mathbf{i}_y^T\mathbf{i}_\theta & \mathbf{i}_y^T\mathbf{i}_\phi & \mathbf{i}_y^T\mathbf{i}_R \\ \mathbf{i}_z^T\mathbf{i}_\theta & \mathbf{i}_z^T\mathbf{i}_\phi & \mathbf{i}_z^T\mathbf{i}_R \end{bmatrix} \tag{4.7}$$

The last step is going to be the change from the cartesian X, Y, Z to the ABAQUS Y, Z, X coordinate system, which is done by swapping columns of data in the same fashion as what has been described in Chapter 2. Once these stresses have been transformed to the data reference system (to use the same nomenclature as Chapter 2, stress combination takes place in the same way as described in Section 2.2, which means reading the stress data for part EARTH and adding the realistic stresses to them. This way, a new Mises stress is recalculated and the script proceeds in the same way as has been explained in Section 2.2.

5
CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE DEVELOPMENTS

5.1. CONCLUSIONS

This work has, for the first time, tried to investigate how the viscoelastic Earth behaves when a combination of glacial and background stresses is applied to it. The study of the derived iterative process coming from the implementation of this combination in a FEM software, ABAQUS, goes beyond the work by Schmeling (1987), because the focus of the aforementioned study was the use of a realistic scenario of combination between glacial and ambient stresses (instead of purely glacial stresses) to determine the mantle rheology is Newtonian or non-Newtonian or isotropic or anisotropic. Here the rheology and mantle flow are given and this knowledge is used to investigate what is the effect on the Earth's response to glacial load of a combination of glacial and background stresses. Comparison with the study by Wu (2001) also demonstrates that this work goes beyond its scope, because the main focus of that paper has been the use of a Mises stress which took into account from the beginning of the combination between glacial and ambient stresses to improve the fit to relative sea-level data. The study by Wu (2001) in particular shows that an interaction between glacial and ambient stresses is required to obtain a better fit to sea-level data, thus a more realistic Earth behavior. However, since these studies use a Mises stress where ambient stresses are taken into account from the start, they do not have to add the separate components in the FEM software which then causes the aforementioned iterative process of stress and viscosity change as done in this work. The study by Wu (1992) used a more rudimentary ice load and Earth model, with a completely different scope (demonstrating whether the Earth rheology follows a power law or not). The work by van der Wal et al. (2013), in turn based on the one by Hirth and Kohlstedf (2003), shows that a power law rheology is preferred for a better fit with sea level data and is the one used here for this simulation since it is the one used to build the model. However, this study does not take into account the presence of ambient stresses since it is more focused on the parametric study of a variety of 3D power law rheologies with respect to sea level data to find the best fitting model for GIA in Fennoscandia. Therefore, this work combines the knowledge that both ambient stresses in the Earth and the presence of a power law rheology need to be considered while building a GIA model for more realistic results, which has never been done before, to evaluate the effects of the combination of these two stresses on how the Earth responds to them.

The results presented in Chapter 4 demonstrate that, at present, it is possible to implement an iterative process such as stress coupling and the corresponding viscosity change in a FEM model such as ABAQUS. This can be achieved by introducing a while cycle inside each ice history timestep which can either be stopped when reaching a certain number of iterations (especially during testing) or when a the value lower than a convergence parameter is reached, with this parameter being of variable precision. This while cycle is where the effect of background stresses and their combination with the glacial one are taken into account. For this work, the absolute value of the difference of the sum of the Mises stresses between one stress iteration and another has been used as convergence parameter and set to 100 Pa. However, it has been found that a difference of less than 0.1 Pa is already achieved between stress iterations 2 and 3, so the convergence parameter can be even lower. At every iteration of the inner loop, which simulates the process due to the stress and therefore viscosity change, the stresses are read from the model and combined with background stresses which have been chosen here as constant and positive values of 0.008 and 1 MPa everywhere for two components (normal, S11, and shear, S12). After combining these stresses, the Mises stress is recalculated and the creep law is redefined by using this new coupled Mises stress in Equation 1.1. The model analysis then begins again and once completed it produces new results. This generates new stresses which are read and so on. While the process might be time consuming, a mechanism has been implemented to save results and "snapshots" of the model at each stress iteration to save time when the simulations are interrupted such as due to system crashes. Interesting results can already be observed after around two days of simulation to produce the first results, which is a relatively short time compared to the initial time of the entire simulation of around 4.5 days. This answers the first question introduced in Chapter 1 and leads to the other two.

Regarding the second question, adding the selected background stresses seems to introduce a higher relaxation over a strongly localized area where the influence of the ice load is felt compared to a baseline situation, leading to different situations for each quantity. The stresses are generally larger at the two timesteps and increase between them instead of decreasing which confirms the finding by Schmeling (1987) described in the introduction about an increase in rebound stresses when two flows are combined. This stress increase can also be explained due to the presence of background stresses added at every stress iteration to the ones coming from the model. This increase is not as large as expected due to a small decrease between stress iteration 2 (with the same results as 3 because of the aforementioned convergence) of timestep 1 and stress iteration 1 of timestep 2 caused by stress dissipation due to the higher relaxation leaving lower stresses to be generated when the ice load is reapplied. Deformations are generally smaller than the baseline case and decrease with time instead of increasing due to this change in relaxation, but again they do not decrease as much as expected due to a small increase between stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2 again due to the remaining deformations being small once the ice load is reapplied. The viscosity demonstrates this higher relaxation in the AOI with a strongly localized area of lower values due to high stresses as per Equation 1.2 compared to the surroundings in the area where the ice load is applied and a substantial decrease, by around 2 orders of magnitude, in value between timestep 1 and 2. In the baseline case, the viscosity under the ice load was virtually the same as the surrounding area of West Antarctica and there was almost no change in viscosity between timestep 1 and 2. Therefore, the introduction of ambient stresses leads to a viscosity reduction with time over the load application area compared to basically constant values for a case without them.

Regarding the third question about changes after having added the ambient stress components themselves, changing the background stresses for the same rheology seems to have no effect on the simulation results. The stress components and viscosity have a difference of 0 on the significant digits, which means that the results for two different background stresses are the same. This can be explained by the vast majority of the points over Antarctica having a much smaller Mises stress compared to the ice load application area due to their distance, be it at depth or in latitude and longitude, from the ice load itself. This leads to a high viscosity and low dislocation creep influence in Equation 1.1 everywhere except for the load application area, with both these factors leading to the model being unable to "feel" a change in background stresses. This is due to the viscosity being too high to allow for output changes to be seen in the timescales used for this model and the dislocation creep term, containing the Mises term with highest power, being too small to actually produce any visible effect in the creep law itself, as explained in more detail Section 4.4.

Fixing a depth range, the main differences all seem to come from the different B coefficients of the rheology file with B_{diff} being higher in the case of a wet rheology. The plots for the B coefficients for both rheologies and depth ranges are displayed in Section B.3. The different rheology has different effects on stresses, deformations and viscosity. More specifically, an increase in deformations is observed together with a decrease in the Mises stress and the viscosity, but this difference is easily observable only for the very first stress iteration for both stresses and deformations since the differences between rheologies reach 0 at stress iteration 2 already. This means that the sensitivity of both stresses and deformations does not depend strongly on the rheology. This is similar to what has been discussed in Section 1.1 from the study by Wu (1992) and seems to confirm its findings about the model unable to "see" a power law rheology and therefore differences in the creep parameters for every element resulting from differences in B_{diff} and B_{disl}, which are the ones influencing the creep law, especially as time passes. Another cause for this phenomenon could be the fact that the differences in B_{diff} and B_{disl} are not significant enough to induce changes in the results, which can be seen in Section B.3. Viscosity, on the other hand, slightly decreases with the new rheology even when the Mises stress is the same due to the effect of the different B coefficients since it is directly calculated from them. This decrease in viscosity also seems to reduce the difference in viscosity between the high relaxation area where the ice load influence is strong and the surroundings, which is due to the general effect of the higher B_{diff} and local effect of the lower Mises stress. It might be then investigated how different compositions influence the results and if this influence is going to also extend to stress iterations beyond the very first one. Moreover, it would be interesting to see whether using more realistic stresses, as described in Section 4.5, could lead to changes in results by changing the background stresses. This brings up the discussion of some of the recommendations for future work, which will be treated next.

5.2. Recommendations for future work

Since this work has attempted to generate results which have never been shown before and its main focus has been showing that it is possible to simulate the combination of background mantle and glacial stresses and discuss preliminary results of this combination, there are many points that can be improved or expanded upon.

5.2.1. Application of realistic background stresses

As described from what has been found in Section 4.5, applying more realistic stresses to the model can also bridge the gap between the assumptions made for this simulation as for how it is now and what has been observed for the Earth's behavior, be it input parameters or quantity (such as traction) distribution. However, it has to be considered that this endeavor will probably require studying additional literature to the one presented because the realistic stress study conducted here is very basic. Since these background stresses depend on a variety of parameters such as for example the plate surface velocity and direction, this operation might lead to a parametric study where different parameters combinations are tested to give different results. The deformation results could then be compared to uplift studies in the West Antarctic area for verification, since the resulting stresses due to the iterative process of stress combination have been displayed here for the first time and deformations are directly measurable.

5.2.2. Use of different input parameters

Another modification that could be applied is the change of some of the input parameters for the simulation, such as the ice load, the rheology parameters influencing the B coefficients or the background stresses. If new ice load data is available, it is especially recommended to use it since this would change both the stress distribution and depth reach. If the ice load is sufficiently large, it could be possible to study what happens at around 1000 km, which is the depth limit for part EARTH used for the stress coupling. The same goes for rheology parameters such as he ones the ones described by Hirth and Kohlstedf (2003) and used as a basis for this study, meaning that if more recent studies for the AOI are available with better estimates of parameters such as the T distribution, water content or grain size it is suggested to use them to recalculate the B coefficients. Rheology has a strong influence on the viscosity and it is suggested to try out different rheology files to determine in more detail how stress coupling is influenced by the model composition. Regarding the AOI, it would be a good idea to use West Antarctica again, due to the lower viscosity compared to East Antarctica which would lead to a more pronounced response to the stress combination and change the timesteps studied. If more precise rheology or ice data from newer or localized studies is available only for a subsection of West Antarctica, it would still be interesting to evaluate the Earth response there since high relaxation depends on the introduction of background stresses and not the position of the ice load. Even if the ice load becomes narrower because data is available only for a very limited area, studying the stresses and deformation due to coupling in that area at shallower depths would still be possible and is recommended. Concerning the background stresses, it could be tried to use even higher constant values everywhere for the background one (10 MPa as from the work by Wu (2001) or to use new, realistic background stresses to study the Earth response in a setting that even better coincides with geophysical observations. Another possibility is to change the sign of the added stresses, be them derived as described in Section 4.5 or the same ones used here.

5.2.3. Changing the added stress direction

Another change strongly suggested to implement is the change of direction of the stress components to pair to the base components. This can either be done by using the same stress components as used here, S11 and S12 both equal to 0.008 and 1 MPa for two different cases set to -0.008 and -1 MPa respectively instead, or by introducing new base stresses such ones resulting from a review of previous literature as discussed earlier and invert the direction of each of them. This arguably has a large effect on the model outputs, as Equation 1.5 shows that changing the sign of stresses added, for this example, to S11 is going to cause changes to this component that are amplified by the squared factors at the numerator. Therefore, it is important to investigate whether convergence takes place in this different case for the same number of iteration or not, if convergence takes place at all.

5.2.4. Study results at different ice history timesteps

Another possible change to implement could be investigating what happens at the successive timesteps, especially from timestep 4, where as shown in Figure 2.4 there is a large ice load reduction in a short period of time. This would make a nice comparison wit the situation described here, which uses the same ice load for two timesteps over a long period of time of 30 ka which separates them compared to the timesteps from 100 onwards. Moreover, it could be interesting to also evaluate the Earth response for a larger number of timesteps, stopping the model after 3 or 4 timesteps or more depending on the results obtained and time spent.

5.2.5. Increase processing scripts efficiency

This thesis work required writing a large amount of code due to the amount of operations to perform with the data. Because of the amount of operations and their many different functions implemented in this code, several of these steps can be reviewed and overhauled for better efficiency. The main suggested improvement would be in the processing Python function used to associate each element to its stresses. The operation of finding the node coordinates making up each element takes at least half an hour to complete for each run, which could be reduced. Moreover, in the plotting scripts written in MATLAB, the function used to set the same axes for every run and stress iteration could be optimized for better speed. This is due to the fact that the function has to read all the complete files for all stress iterations and calculate the minimum and maximum of all the quantities that need to be plotted. Increasing the plot resolution and using larger depths in the plots to use more points also increases the computation time, with the resolution being an especially important factor with respect to the longer time spent for plot generation.

5.2.6. Porting the entire code to Python

Because of a series of problems associated to the installation of the mapping toolbox for Python and difficulties with Python plotting in general, the final section of the code dedicated to plot generation has been written in MATLAB. However, it could be tried to transfer the remaining MATLAB code to Python to avoid having to transfer variables from the processing to post-processing parts, instead having all of them in a workflow written entirely in the same programming language.

Code explanation

This appendix contains a detailed explanation of the code used during the thesis work, written in Python and MATLAB with FORTRAN contributions. The code is available on GitHub at Morra (2020a). The Python part will be presented first, followed by the MATLAB one and then the small FORTRAN code used to define the ABAQUS subroutines.

A.1. Python code - Model generation in ABAQUS

The first five scripts are dedicated to the generation of the model database (.cae) file and are only called once, as the base .cae file will only be modified, and not created again, for the remainder of the results generation process.

The model generation and iterative procedure due to the non-linearity of the viscoelastic Earth as discussed in Section 1.2 are carried out through the steps described in the list below.

1. Create the base model file (.cae) in ABAQUS;
2. Read the .cae model and define the first operations on them (submitting a job) through the use of a FORTRAN file to generate the first output database;
3. Iterate over the ice age history timesteps, and in each timestep iterate either until convergence is reached or a certain number of stress iterations has been reached. The following operations take part in each stress iteration.
   (a) The .odb file is opened to generate the stress and deflection reports using a base e.dat file containing creep parameters for the creep law definition;
   (b) The stresses read from the report file are saved in separate .csv files for each part;
   (c) If we are not in the very first stress iteration in the first timestep and iteration, we calculate the parameter used for the convergence criterion and exit the cycle if convergence is met;
   (d) The deflections are read from the report file and saved in separate .csv files for each part;
   (e) Pre-defined stress components are added to the stresses read from the part files;
   (f) The Mises stress for part EARTH, the one of interest for this thesis work, is recalculated;
   (g) The new Mises stress is saved both in the e.dat file read by the FORTRAN script for the creep definition after deleting the old one and in a file with the same name in the sub-folder relative to the cycle;
   (h) The cycle counter is incremented and the cycle begins again.
4. After iterating over all the timesteps, the sph_tools_TPW7_4SLE is called in a separate environment to compute the gravitational perturbation at each step due to the changing loads and saves the individual deflection components at each of those steps.
5. The new load distribution is calculated with the sea level equation at each timestep and for each layer. The .cae file is then modified and we go to the next iteration.

A description of all the scripts used will then be offered next. A common change to all the scripts has been the insertion of a new path, which is the folder where the scripts aptly modified for this thesis are stored, and the command to add to the system both this path and the base path where all the files and folders for this work are stored. This was necessary to prevent import and file not found errors while keeping all the files in different folders to sort them in a more efficient way. Other changes will be specified script by script.

A.1.1. Model_data2_top

Inputs: None
Outputs: Model parameters
This script is used to define the vast majority of the input parameters, such as the timestep vector and Poisson's ratio, used in all the successive scripts which build the model and run the simulation. This is the only script that does not need to import any other script or path, because this is the first one to be run and also the one where the paths inserted at the beginning of every other script are defined.

A.1.2. Model_data2

Inputs: All data from Model_data2_top

Outputs: Additional model parameters, DataIce_in_int.dat

The next script has the main function of defining material constants such as radius, viscosity, density and Young's modulus for each layer based on inputs from Model_data2_top (which is imported here). After this, the ice load is either rearranged or created, and in the workflow for this thesis it is created through the use of the next script, ScatterLoad2BlockLoad. This ice load is saved and loaded and then the polar circle radius is defined. This script, together with Model_data2_top, is always imported in all the others because of all the parameters defined in both of them.

A.1.3. ScatterLoad2BlockLoad

Inputs: All data from Model_data2_top and Model_data2

Outputs: DataIce_in_int.dat

This script is tasked with creating the initial ice load in the form of a grid composed of ice loads in the middle of each grid cell. This kind of load is created for each timestep and then appended to a large matrix containing a n*2n grid stacked 11 times which is the number of time steps, while n is the number of latitude cells. This load will be used in the definition of the base model. No noteworthy modifications have been applied here.

A.1.4. Model_gen_cone_rampV6_3.py

Inputs: All data from Model_data2_top and Model_data2, DataIce_in_int.dat

Outputs: Earth.cae file, S0.dat, Data_in_Slvl.dat

This is the script where the entire model generation takes place based on all the data generated and parameters defined until now. The script generates the core and remaining layers and selects their materials. The mesh parameters are defined and used to generate a mesh which needs to be finer at the poles since Antarctica is the area of interest for this work. After these operations the timesteps for the transient static response are generated based on two cases, one with ramp load during each step and and one with a step load. The mechanical foundations are also defined since this is an integral step of the transition of SLE modelling problems to FEM software, as shown in the work by Wu (2004). The next step is the creation of loads: the first one to be defined is a centrifugal pressure load, followed by the combination of ice and sea load which also generate a sea level data file.

Before defining the total forces acting on the model another script, described in Subsection A.1.5, has to be called to take into account the displacement of the CoG (center of gravity) due to the distribution of the ice and sea load, which has its own gravitational signal. After calculating the displacement vector at each timestep, a load is generated on the .cae file to take this effect into account and is combined with the ice and sea load described earlier. The model outputs are defined (for the case treated in this work, they are stresses and displacements in all of their components) and the model database file is saved at the end of the script. A modification consisting in the absence of the job definition at the end of the file has been added, with the reason for this explained in the Iter_ult_v7_3SLE paragraph.

An important operation to remember to carry out after the model has been generated is the definition of additional parameters by hand as ABAQUS does not seem to recognize them and this definition would normally be skipped, using default options and leading to errors. The user has to redefine the mesh parameters for the EARTH and EARTH_LOW parts in the GUI, and this is done by selecting the mesh module, displaying these two parts, selecting them in their entirety and then only check the boxes relative to "Hybrid formulation" in the dialog window. The material property of M4 also has to be modified to use a user-defined creep law, and this is done by selecting "materials" in the model tree on the left, then right-clicking on M4, removing "Viscoelastic" from the list of parameters, inserting "creep" and then selecting user-defined law. While tedious, these steps are necessary to allow the simulation to run (when it comes to redefining the mesh parameters) and have results in accordance with the base model given for the thesis (regarding the material creep parameter modification).

A.1.5. Initial_CoG_correction

Inputs: Data_in_Slvl.dat

Outputs: CoG_vector.dat

This short function called from Model_gen_cone_rampV6_3 calculates the CoG vector at each timestep. First, the script reads the ice and sea load data files, multiplies them by the ice and seawater density respectively and adds them together. This sum is then multiplied by a gravitational scaling factor which has the same values for each cartesian component and is calculated relative to the geographical harmonics of degree 1 (l=1), which is where the ice and sea load distribution has a gravitational footprint. The script returns a file containing three displacement components for each timestep. No noteworthy modifications have been added here either.

A.2. Model Analysis

This is the part of the ABAQUS code where results are produced by defining jobs on the .cae file at each iteration, which is followed by updating the .odb file every time these results are produced.

A.2.1. ITER_ULT_V7_3SLE

Inputs: All parameters and files produced up until now

Outputs: Many different model files, the most important being Earth.odband Earth.dat; iteration folder with all its subdirectories containing all the data processing files

This is the script where the largest number of modifications has taken place, since its base version contained the iterative process which generates the results to be read. The maximum number of CPUs to use and iterations to run are defined, then a first while cycle starts are defined at the beginning of the script. This first while cycle will always run until the maximum number of iterations is defined.

To save time, a series of if cycles have been added throughout the script. These if cycles encompass some operations to be done in the script and check for the existence of a series of .txt files with different names. These files are generated at the end of each operation, and if they already exist such operations are skipped. This avoids rerunning unnecessary sections of the script if modifications have to be added and these modifications have to be checked for validity, for example, as commands like job submission can take up to one day and a half to complete.

The first operation after entering the while cycle is to check whether the .txt file related to the completion of the job related to each iteration exists or not, so that time is saved by skipping the corresponding code section if the results are already there. If the .txt file exists, this action is skipped and the script goes to the next operation, otherwise the job is defined, submitted and waited for its completion. Since the job definition and not just its submission needs to be executed at each iteration, it had to be moved from the end of the .cae file to the beginning of this script. Using an if cycle here saves between one day and one day and a half. It is worth noting that this job definition for each iteration can refer to both the FORTRAN files, because user_2 will be used for the very first job submission where the .odb file is created while user_2_mises will be used in any other job submission because the coupled stresses this latter file needs to make use to will have been generated. The next operation, which produces files containing node sets for the model, is usually skipped in the current workflow as it is not necessary towards the end results.

Another if cycle for the generation of two save files then starts, one containing the node coordinates and norm of the deflections vector and the other containing the node coordinates and the three deflection components. This operation is again subjected to an if cycle check to save more computation time, even though it is much shorter than the job submission. If a .txt file that is generated at the end of this operation already exists, this action is skipped. This saves about 10 minutes of computation time.

It is after this step that the largest modification to the file takes place, consisting in the combination between stresses resulting from the simulation and external ones defined in the script. First a for cycle looping over each timestep is defined, then the convergence parameter and stress iteration counter are initialized. This is needed because the while cycle contained in each timestep stops when either a certain number of stress iterations have been run or the convergence parameter recalculated at each stress iteration is small than a certain threshold (Here, 100 Pa). This convergence parameter will be described in the coming paragraphs.

After the while cycle for each timestep begins, a series of parameters relative to the files to be saved are defined, starting with whether the .csv files must have headers or not. There are then some base parameters such as the name of the dat files containing the elements and node information for the entire model Earth.dat and the name of the file containing viscosity parameters e.dat. After this lies the definition of the folder where all the data .csv files must be saved, called $level\_dir$, and a list of paths defining the sub-folders to be created inside $level\_dir$ for a better management of all the data produced. After defining all of these necessary paths the next operation, which is opening the .odb file to produce report files containing the results and deformations, can be started. The reports are generated by accessing a determined step and frame in ABAQUS and asking for the generation of a file containing the desired variables outputted in this case at either element centroids (for stresses) or nodes (for deformations). The very first reports (Iteration 1, step 0 and stress iteration 1) to be generated will have their output based on the base e.dat file containing only the B_{diff} and B_{disl} variables for every element.

After generating these reports, their data needs to be extracted and stored in a way that is easy to handle, which means writing their values to .csv files. The stress report is read before the deflection one, and the procedure for the data extraction from the two reports is similar. The first step is defining the path where the report file is located, then the list called $lines$ containing the line numbers where the stresses of each new part are listed, the amount of these lines, the list containing the part names to be used for the name of the .csv files, the list where to save the file headers, the line counter and the keyword to identify the lines in the report files where the stresses of each part start to be defined are initialized.

The report file is then opened and the part or region names and variable names to be used as the .csv headers are extracted and saved for later use. These headers are then manipulated to be ready for use when the .csv file is saved, then a matrix needs to be initialized and created for part EARTH, the largest one. This is necessary because the values for stresses are saved by regions where possible, and since EARTH is a part divided in different regions all the values belonging to the part under different regions need to be unified. This makes the successive manipulations where the stresses are associated to their centroid coordinates much easier, as the node coordinates making up each centroid are saved by parts and not regions. Then, if the last file to be saved as a .csv does not exist yet, the report file needs to be opened again to copy data to the .csv files. This is the kind of check used to save time for report generation and data processing, since the existence of files can be directly checked. Operations that modify the .odb file do not generate any new file, so it is necessary to generate .txt files after their completion.

$lines$ is used to save each part values in the correct .csv files by separating the stress values for each part and an iteration starts over the elements of this list. First it is checked whether the script is in the last part/region of the report file, then the lines of the report file where to extract the stress values are narrowed down by defining a linearly spaced vector ($search\_lines$) using the $lines$ element in the current counter, the $lines$ element in the current counter + 1 and a step of 1. The report file is then opened and the script iterates over it again, but every time a line of this report file is in the range of $search\_lines$ the doubles in it are extracted using a regular expression and, if the vector of these values is sufficiently large (thus excluding possible lines where other numbers are defined and we are not interested in), this vector of doubles containing the element label and its 7 stresses is saved in a row of $stress\_submatrix$, the matrix referring to that specific part. This matrix is then saved based on whether it should have headers or not. If the script reaches the last region in the report file the operation is the same, but instead of iterating between $lines(i), lines(i+1)$ where i is the counter in $lines$, the report file is opened to save the search for and save the last line index. $search\_lines$ is then defined between $lines(i), last_{l}ine$.

After each of these part matrices is saved, a check on whether "Region" is in their name is performed, because these matrices belong to part EARTH and need to be incorporated in it. If this is true then they are appended to the large matrix initialized previously and called EARTH. After being appended, these matrices need to be sorted based on the centroid label, and this is done through the use of a dictionary. After sorting the EARTH part matrix, it is also saved in the same folder as all the other .csv stress files. The stress report processing section of the code is then finished.

The next part is relative to the calculation of the convergence parameter for the current stress iteration, which has been defined as the absolute value of the difference of the sum of the Mises stress vectors coming from the previous and current stress iterations. Three cases need to be defined: one is the most frequently occurring one, for any timestep but not in the first stress iteration. The stress matrix generated for the current cycle in the previous lines of code is read and together with the one generated in the previous stress iteration. The Mises stresses are read from both of them and the absolute values of the difference of the sum of their respective matrix columns is computed. If this value is lower than the threshold of 100 Pa defined in each while cycle then the stress iteration interrupts. The next case is the very first stress iteration, when the timestep is 0 and the iteration is 1, and nothing must be done if this condition is true as there are no previous cycles where data has been created. The last condition is for every timestep that is not the first and the stress iteration number is 1, for any iteration. In this case, the previous stress iteration cannot be defined as current stress iteration-1, because stress iteration 0 does not exist. It is therefore necessary to extract the highest stress iteration number in the previous step and read the matrix defined by these attributes as the one where the "stress iteration 0" would be, as that would be the moment in time immediately preceding the one when the stress iteration counter is 1. The convergence parameter calculation is then done in the same way as explained earlier. The section of the code then terminates to go on to the generation of the deflection reports.

As it has been done for the stress reports, it is also needed to save the deflection data for each part in the corresponding .csv files. This is done exactly in the same way as for the stress values extraction and therefore does not need to be discussed again, with a single difference: deflections are not saved by regions but only by part. This means that the code section used for stress data processing where regional matrices are incorporated in a single one to be sorted using a dictionary is skipped. The stress coupling procedure is next, but it has already been described in Section 2.2 and therefore it will not be repeated here.

The next subsection is still part of this script, but it has been written as a standalone function because of the sheer number of the modifications implemented. This is an extended version of the stress coupling process described in Section 2.2.

A.2.2. Stress coupling

The first action to take for stress coupling is the definition of the stresses to couple, which have been set at different orders of magnitude for two different cases, 0.008 and 1 Mpa. These stresses can be substituted with realistic ones, whose definition can start from the theory of Section 4.5 and the explanation of the code from Section A.6.

The next parts are where the stress coupling proper takes place. First of all, the base rheology e.dat file (the one without a dedicated column for coupled stresses) stored in the base folder for this thesis work is copied in the base folder for this run and a column of zeros is added to it in the very first stress iteration. This is needed because for any other stress iteration, step and iteration the last column will contain the coupled Mises stresses that the modified FORTRAN file needs to read and in order to avoid confusion with the given commands, it has been decided to create a new column first and then substitute the values with the coupled stresses instead of appending it. Such operation is only carried out for the very first stress iteration since the values in this last column will then be overwritten at every stress iteration. The path relative to the current stress iteration, where the e.dat file containing the Mises stress resulting from the coupling of current stresses has to be saved, is also defined. The stress components to couple and the folder where to save the .csv files containing the coupled stresses are then also defined.

The next steps are reading all the files in the stress folder, which contains the .csv files of the stress components for each part, and the selection of the column(s) corresponding to the relevant component(s) to add the previously defined stress to. The .csv files where it is needed to pair the stresses are then selected, and in this case the pairing operation can only be carried out for part EARTH since the number of values in the e.dat file is the same as the number of centroids in part EARTH. The selected part file is then opened and organized into a matrix that can be easily manipulated, then each of the selected stress components from the EARTH.csv file is added to the pre-defined stresses for each element and then the Mises stress for that element is recalculated using Equation 1.5. This new Mises vector of values overwrites the previous Mises column in the .csv file. This new matrix is then saved, the file containing the coupled stresses for the EARTH part is opened and its coupled Mises stress is read and then saved in the last data column of two e.dat files. The first one is the e.dat file in the base folder for the current run, which is deleted and then saved to avoid overwriting ambiguities. The second one is the e.dat saved in the current stress iteration folder to keep track of the changes in results.

Finally, if the .txt file attesting that the job for the current stress iteration has been completed does not exist yet, the job is submitted using user_2_mises as FORTRAN file instead of the base one in order to use the newly coupled Mises stress to calculate both the creep and viscosity. After its completion, the stress iteration counter is incremented and the next stress iteration starts. If the necessary conditions are met, the step number is also incremented.

The last part of the script, after the time step iteration, is dedicated to redefining the ice and sea level loads over the model and has not been the focus of this thesis work. This is first done by calling sph_tools_w7_4SLE, then at each timestep the total ice and sea load is calculated and transformed into a mapped field to be applied to the model. Pressure and load are then derived for the surface of the model. For each timestep, after this operation, the center of gravity displacement load difference if compensation is enabled or just the pressure load difference is calculated and applied. The loads are then applied to each layer too, with an additional load to be defined if the compensation for the CoG displacement is active.

SPH_TOOLS_W7_4SLE

Input: Nodes_config.dat, Data_output.dat, Data_output_hor.dat
Output: (non relevant for this work) CoG_vector.dat, Check_geoid.dat, Data_in_Slvl.dat, RSL.dat, IceFreeOcean.dat, RSL_spatial.dat, deformation files, layer .dat files, Check_deflection.dat
This script runs in a separate Python environment compared to ABAQUS and therefore cannot intervene on any of the model files built up until this point. After importing the necessary libraries and modules, the script defines 3 functions. The first two functions are the same, with the second one being a new version of the first one and is the only one of the two to actually be used. This function generates a linearly interpolated grid of load data over the Earth, while the third one is used to calculate the change in moment of inertia due to these loads and is not used in this work.

The next part of the code only executes if it is not imported, so that if that happens its functions can still be used while the rest of the code is not executed in order to save time. The function computes the three deformations components for each layer of the Earth, and these deformations are used to calculate the corresponding change in moment of inertia for each layer. The deformations are then saved at each timestep, and the last section of the function then computes the relative sea level and eustatic sea level distribution at every timestep, which are then saved. The iteration number is then increased and the next one starts with a new job submission. However, as said before, this part related to the sea level equation has not been relevant for this work and did not need to be reached to generate results worthy of analysis.

This section, related to organizing the data so that they can be effectively plotted in MATLAB, has also been entirely written in Python. To better understand the description of the next scripts, it is helpful to describe the type of output given by ABAQUS and the desired end results from the data processing procedure.

A.3. GENERATION OF DATA MATRICES IN PYTHON FOR MATLAB PLOTTING

As said before, many of these scripts are also going to process other parts, if adjusted for their processing, aside from part EARTH only. This has been decided so that the scripts can be adapted for future uses where there might be the need for the processing of more parts aside from just EARTH. The generation of data matrices for plotting processing steps:

1. Define relevant file paths and variables commonly shared by scripts;
2. Read the Earth.dat file for information and make it more readable;
3. Create the complete stress or deformation matrices to use for plotting;
4. Discretize all nodes and elements based on depth to have a better idea of the depth ranges for plotting;
5. Define the latitude, longitude and depth ranges for the problem;
6. Generate a matrix for the element with highest Mises stress in those ranges at different times;
7. Generate a handle file that MATLAB will use to read all the necessary information to generate plots.

A.3.1. PYTHON_READER_PROCESSING_ONLY

Inputs: Run folder containing the Earth.dat and e.dat file and a series of subdirectories as illustrated in Figure 2.6.
Outputs: Run folder with the addition of a series of .csv files derived from the data processing procedure as shown in Figure 2.8.
This is the main script used for data processing and it calls a series of other functions, with some of them containing other functions. As the name implies, this is a new version of a longer original script where the part relative to report reading and stress coupling was also being tested. This part has been moved to ABAQUS and the script modified to work with the simulation results. The ABAQUS output the script is designed to work on is displayed in Figure 2.6.

The script needs to define a series of parameters before calling any of the functions. The run, iteration, step and stress iteration numbers are defined first, since they are used to initialize the path where all the information we need is stored. The report names are next, then the .dat files names which are e.dat to be used for the viscosity calculation and the Earth.dat file used for pairing stress and deformations to centroids and nodes respectively. The base path, relative to the run and containing the stress iteration subdirectory and the Earth.dat file, is initialized based on the operating system used. The subdirectory relative to the stress iteration we want to analyze is then constructed based on the run, iteration, step and stress iteration numbers initialized earlier. After this a series of paths are initialized, and they are the ones where the report and Earth.dat files exist and a series of directories where to save the matrices resulting from the data processing operations. If these directories do not exist, they are generated.

The boolean to decide whether to generate matrices with or without headers, called $headers\_on$, is also defined in this main function. The user is then prompted to enter a digit to decide whether to work with stresses or deformations, with this input called sd_input, and then the first function coordinate_reader is called. This function includes many other functions inside and performs the creation of the files containing the components associated to the centroid/node coordinates together with the radial distance, depth, latitude and longitude values for each element/node.

A.3.2. COORDINATE_READER

Inputs: boolean to process stresses or deformation, boolean to generate matrices with headers or not, paths to the stress matrices (coupled and uncoupled) and the deflection matrices, a series of paths where to store the generated files
Outputs: Described in detail in the next paragraphs of this box as there are too many to be summarized here
This function calls a series of functions inside to further break down the process of creating the complete files and also because only some data operations can be done on both stresses and deformations. Because of this, the first two functions, dat_processor and node_processor, are run both for stresses and deformations, while after them elems_processor and associate_stress_coord are run for the generation of the complete stress files while only associate_deflection_coord is run for the creation of the complete deformation files. This difference is due to the fact that stress is relative to centroids and not node coordinates, and the former are not contained in any file produced by ABAQUS. Therefore, before associating the stress components to the centroids it is also needed to associate the node coordinates to each node making up an element (this is the information in the Earth.dat file) and then calculate the centroid coordinates as an average between the 3 coordinates over the number of nodes composing that element. A large part of the scripts developed for this function has been re-adapted for a simpler case of single-part processing for the definition of realistic stresses as described in Subsection A.6.3.

A.3.3. DAT_PROCESSOR

Inputs: location of the Earth.dat file, called dat_path
Outputs: three lists used to define regions of the .dat file to extract data from, file names to save the .csv matrices, new Earth.dat file saved at
This function opens the Earth.dat file and makes it easier to handle for the next functions that have to read data from it. The file is opened and some variables are initialized. These are 4 lists containing the line indices of where the data for the nodes and elements of each part begins called $node\_lines$ and $elem\_lines$ respectively, then the line indices where each part name (called $part\_lines$) is stated and finally the lines where the element data for each part ends, stored in $nset\_lines$. These are used to define "regions" of the file to search data from: all nodes for each part are contained between the beginning of the node data and the beginning of the element data, while all element information is contained between the node data and the Nset data. 4 corresponding counters are initialized next to calculate how many of these variables exist in the file and a series of keywords used to search for the lines defined above, with the last keyword called $input\_paragraph\_end$ being the last useful line of search in the entire file for every part since all the data after that is not needed for our operations. The .dat file is then scanned to save the position of the lines described above and count the number of their occurrences. Each time $part\_lines$ finds a new occurrence, a part name is appended to $file\_identifiers$, the list of filenames to use to generate .csv files. All the commas are then substituted by spaces and the "LINE n" word appearing every 5 lines in the Earth.dat file at their beginning where n is a multiple of 5 is deleted, and a new Earth.dat file is saved with these modifications at a location described by new_earth_dat.

A.3.4. NODE_PROCESSOR

Inputs: path where to save the node files called $common\_files\_path$, $node\_lines$ and $elem\_lines$, $file\_identifiers$, dat_path, $headers\_on$
Outputs: $individual\_path$, $large_node_matrix\_path$
This function creates .csv files containing the nodes for each part associated to their coordinates. After initializing the folders where to save these files and the headers for the node matrices, it is checked whether the last matrix to be created in the function already exists or not. If this is not the case, the Earth.dat file is opened and two for cycles are defined, one inside the other. The external one runs on i from 0 to the length-1 of the list of lines called $node_lines$ defining the beginning of the file sections where all the nodes for a part are described. The inner one runs from $node\_lines(i)$ to $elem\_lines(i)$, with $elem\_lines$ being the list of line indices where the description of elements begins and the one for node ends. This way a region of file where to search for nodes for a single part counted by i is defined.

The file is then scanned line by line between those two lines, stripping the line of the end of line character, extracting all floats in it and saving them in a row array. Once this row array is defined, two additional cases are defined to not read useless data lines or remove the number after "LINE" which is a counter present every 5 lines at the beginning of each line. The .csv files are then saved with headers or not based on $headers\_on$ at the $individual\_path$ folder. A larger matrix containing all the nodes for every part is created by again using a dictionary, and this matrix is also saved in a .csv file at $large_node_matrix\_path$. This matrix is created because, as described in the subsection related to associate_stress_coordinate, all the node coordinates for each element will need to be searched for each part, so it is convenient to have a list of all nodes for all parts if other parts aside from EARTH need to be processed.

A.3.5. ELEMS_PROCESSOR

Inputs: $node\_lines, elem\_lines, nset\_lines$, path to save stress processing results called $stress\_processing\_path$, path where to save stress results for every part called $stress\_part\_values$, $dat\_path, file\_identifiers$, $headers\_on$
Outputs: $individual\_element\_paths$
This function is required for the stress processing workflow and associates each element to the nodes making it up, then reads the Z, X, Y (because still in the ABAQUS reference system) coordinates for each of those nodes and calculates the Z, X, Y coordinates of the element centroids as an average over the 8 nodes. Before any other operations, the line indices containing the keyword where every element region definition finishes, contained in $nset\_lines$, have to be "filtered" because the keyword for each part is repeated more than once to describe element sets which are not relevant for this thesis work. To do this, for each line index where the definition of the elements for a certain region starts, contained in new_elem_lines, the first element of $nset\_lines$ larger than the current element of new_elem_lines is extracted. Three large element matrices are then defined containing the elements for all parts in different order if other operations have to be carried on the data. The file headers to use in the .csv files are then defined and again we skip the data processing operation if the last file to be created and saved by the function already exists. If this is not the case, the .dat file is opened for data processing, just as in nodes_processor.

Two matrices are allocated and these will store the elements for each part without and with resetting the label counter in two vectors called $elem\_vector$ and $elem\_vector\_reset$. They have a number of rows equal to the difference between the indices of the .dat file where the section of element definition for that part starts and finishes and 9 columns. For every line in the range of these indices, the function strips the line of the end of line character and extracts all floats, working with different cases based on the length of the float vector. This is necessary because some elements are made up of 6 nodes and others of 8, while any vector shorter than 6 elements has to be discarded as we know it does not have interesting data. The vector is then saved in the current line of $elem\_vector$ and $elem\_vector\_reset$ which are then scanned for lines containing unwanted data.

The .csv matrices for each part are finally saved, again based on $headers\_on$, at $individual\_element\_paths$ and then appended to the large .csv file containing the information for all parts. One of the large element matrices will be again sorted by element label using a dictionary, while the other two, containing the unsorted elements and the unsorted element with a resetting counter for each part, need to be concatenated so that they can be saved as .csv files. The function ends after saving these files.

A.3.6. ASSOCIATE_STRESS_COORD

Inputs: $individual\_element\_paths, stress\_part\_values, large\_node\_matrix\_path, headers\_on$, $stress\_matrices\_path$, folder where the coupled stress matrices are located called $coupled\_stress\_folder$
Outputs: $centroid\_files\_path, complete\_files\_path, geographical\_complete\_files\_path$
This function is similar to associate_deflection_coord, and it is where the files generated up until this point are joined to create the .csv files used for plotting containing complete information for each element(in the next paragraphs, this will be done for the nodes). A series of paths to save the .csv files is defined and created if necessary, then the headers for the complete .csv files and the ones just containing the centroids are declared. Again a check is

performed on the last file to be created in the function to see whether we need to perform the next operations or not, and if it does not exist the file creation procedure is started.

The large matrix containing the nodes for all parts is opened and organized into an array, then the files to process are read from the folder containing the stress .csv files are filtered to take away all files containing "Region" in their name or those that have no data. Other options have been added to avoid processing other matrices, and the user can modify these to process more or less parts. After getting an updated list of files for which to create the complete .csv files, a for cycle is used to iterate over each of them. The element file and the stress file for each part are then opened and also transformed into numerical arrays for better handling. An iteration then starts over the stress matrix, and for each element label, the corresponding list of nodes for the element with that label is read. A matrix with a number of rows equal to the number of nodes making up the current element and three columns is defined, then for each node its coordinates are extracted from the large node matrix using logical indexing and the result is a matrix with 3 coordinates for each node row. This process can take a very long time (around 30 minutes) because of the number of nodes making up a large number of elements, and for each node its coordinates have to be searched in a list of nodes (the large node matrix described earlier). The mean along each column is calculated and is going to be the corresponding coordinate for the centroid of the element with the current label. This is then repeated for every label and selected part.

The next operation is the implementation of the reference system from a cartesian to a geographical frame as explained in Section 2.1. The columns to be swapped, and in what order, have already been described in that section.

The stress component file and centroid coordinate file are horizontally concatenated now that the data is in the correct cartesian reference system. After this procedure has been carried out, the radial distance, depth and latitude and longitude are calculated for every centroid and concatenated to the matrix containing the stress components and corresponding centroid coordinates. Latitude and longitude are calculated with a separate function called cart2geo which will be described later. This matrix is then saved for each part at $complete\_files\_path$ while the corresponding centroid one is saved at the $centroid\_files\_path$ folder. Finally, the stress tensor is converted to geographical coordinates using the rotate_tensor function and therefore this procedure will be described in the next paragraph. This function then terminates.

A.3.7. ROTATE_TENSOR

Inputs: name of the part matrix to process called $part\_matrix$, path where to save the new stress matrices called $geographical\_complete\_files\_path$, $headers\_on, complete\_headers$,
complete part matrix to read path called $complete\_individual\_path$, name of the processed part called $part\_file$
Outputs: None (Directly saves files)
This function multiplies the stress tensor for each element with a transformation tensor to obtain the stress components in the geographical coordinate system. Because the R, depth, latitude and longitude for each point have already been calculated in associate_stress_coord to concatenate them to the cartesian stress components after transforming them as per Figure 2.1, the file containing them at $part\_matrix$ is opened and they are extracted. A loop iterating over each line of the .csv file line starts, and the stress tensor is allocated and filled with the data read from each column. Since the stress tensor is symmetrical, three of the off-diagonal components are saved as equal to the three contained in the .csv file and belonging to the "other side" of the diagonal. The transformation cosines are then defined based on Equation 2.3 and used to define the transformation tensor according to Equation 2.12. Once the stress and transformation tensors have been filled with the necessary values for that centroid, the stress tensor in geographical coordinates is calculated with Equation 2.13. This tensor is then saved in the corresponding line of the spherical_components matrix of zeros allocated previously, and when the matrix is completely filled it is saved in a .csv file based on $headers\_on$ at the folder described by $geographical\_complete\_files\_path$ using a name for each part defined by $part\_file$. The function terminates after this last operation.

A.3.8. CART2GEO

Inputs: cartesian coordinate vectors called $cartesian\_coordinates$
Outputs: lat, lon
This short function transforms vectors of X, Y and Z coordinates to two vectors of latitude and longitude using the relations shown in Equation 2.2, with R being the radial distance of any point from the center of the Earth. The latitude and longitude column vectors called lat, lon respectively are returned to concatenate them to stresses and centroid coordinates together with radial distance and depth and the function is then exited.

A.3.9. ASSOCIATE_DEFLECTION_COORD

Inputs: path where to read the deflection matrices from ABAQUS called $deflection\_path$, $individual\_path$, $headers\_on$, path where to save deformation processing results called $deflection\_processing\_path$
Outputs: $complete\_deflection\_path, geographical\_complete\_deflection\_path$
This function is the equivalent of associate_stress_coord but is used to associate every set of deflections to the correct

node. Since deflections are unequivocally associated to nodes, we do not need an equivalent to the elems_ processor function to be run first and we can directly go to the matrix concatenation. First the filenames containing all of the deflections for each part to process are read and loaded, then the paths where to save the deflections and the matrix headers are defined. The list of .csv deflection files is updated by filtering out empty files and, after a check to determine whether the last file to be generated in the function already exists, the association between deflections and nodes begins.

Once again it is necessary to open both the .csv containing the deflections and the one containing the node coordinates for the current part, then the rows with the same node label are paired. Both the deflection and node coordinate columns need to be swapped around around because of the change from the Z, X, Y to the X, Y, Z reference system, and this change is performed as for the centroid coordinates in associate_stress_coord. Since the nodes and deflections matrices are processed in the same order and have the same labels, it is possible to directly horizontally stack the two. Again, radial distance, depth, latitude and longitude are calculated but this time for each node and then the rotate_deflections function is called to transform the deformations to values in the geographical coordinate system. Radial distance, depth, latitude and longitude are then appended to both cartesian and geographical deformation values. The complete matrices of values for each part in the cartesian and geographical reference system are then both saved at $complete\_deflection\_path$ and $geographical\_complete\_deflection\_path$ respectively and the function terminates when all the parts are finished.

A.3.10. ROTATE_DEFLECTIONS

Inputs: $complete\_deflection\_matrix$
Outputs: $geographical\_complete\_matrix$
This function reads the displacement and coordinate matrices and iterates over each line of the displacements to perform the coordinate transformation, which happens with the same transformation tensor as in rotate_tensor but following Equation 2.6. The transformation tensor is the same as the one in Equation 2.12 while the equation for the change of reference system is different since instead of having a 3-by-3 tensor we have a 3-by-1 deformation vector to be multiplied by $transformation\_tensor$, as shown in Equation 2.11. This procedure is illustrated in S.Widnall (2009).

The new displacements are concatenated to the displacement magnitude in a matrix called $geographical\_complete\_matrix$ and then returned to the associate_deflection_coord function, with the function terminating after this.

After coordinate_reader, the suer is asked to enter the reference system used to plot the components called $components\_to\_plot$ and the function depth_classifier is called. This function divides cartesian or geographical stresses or deflection, based on the user's input, in a number of files based on their depth.

A.3.11. DEPTH_CLASSIFIER

Inputs: $sd\_input, deflection\_processing\_path, individual\_path$, $components\_to\_plot$, $complete\_files\_path, geographical\_complete\_files\_path, headers\_on$
Outputs: $classified\_path, files\_to\_classify$
The next step in the data processing workflow consists in binning the data produced up until this point into different depths. This function is now not a core script anymore due to the implementation of the possibility of plotting using a certain range of depth defined by the user in the MATLAB post-processing part, but it has nonetheless been left for two main reasons. The first one is that this section of the program does not take a long time to run (around 3 minutes). The second one is that the files containing values for each bins can give some idea about whether a certain range of depth contains points or not based on their size, so it can be used as a reference when giving MATLAB a depth range for plotting purposes. ABAQUS in fact already groups points together in layers, so entering some depth ranges might cause a plotting error as there are no points to plot.

The function starts by defining portions of file names and the file headers for both stresses and deflections, then the paths containing the .csv files relative to the depth bins and the histograms relative to the depth distribution of the centroids and nodes belonging to the model are defined. Empty part files and part files we are not interested in are then filtered out. Since the bin files are grouped based on what layer they should belong to at the end of the function, the maximum depth of all centroids or nodes is read and then loaded in a vector containing what are widely accepted as being the different layer depths for the Earth's interior. This vector will be used to group depth files into different Earth layers.

The user is then prompted, if the last file to be generated in the function already exists, to input whether they want to rerun the program and, if so, change the number of bins. If the function runs, all part files to be discretized are concatenated in a large matrix called $large\_depth\_matrix$ and for each element of the layer depth vector in meters all the elements or nodes with a depth smaller than the current one are extracted from $large\_depth\_matrix$ and saved in the relative layer depth .csv file. All the rows with the same indices as the extracted elements are then made

0 and cut out of the matrix to make it smaller and increase processing speed. The number of bins is declared and all the names of the layer .csv files are read to iterate over them. A layer depth folder is created for each of them and each .csv is opened as matrix, then a histogram plot with the depth distribution for each layer is generated. Finally the function python_discretizer is run based on the layer number we are at to group data into different bins, and the bin edges are used to define the names of the .csv files to be saved. A .csv is then saved for each of these bins filling them with the elements of the current layer matrix with the same indices as the ones returned by python_discretizer. This means that the function labels each layer matrix row with a bin number based on its depth and returns a vector of labels, then for each of the labels which are equal in number to the bins we extract the rows binned with that label and save them in separate .csv files.

After all of this, a similar procedure is used to generate a histogram with the depth distribution of centroids or nodes and save the corresponding bin .csv files for the entire part EARTH, without pre-defined layer subdivision. The function then terminates.

A.3.12. PYTHON_DISCRETIZER

Inputs: depths for the current part called $data$, current layer index i, layer depths vector called $layer\_depth\_vector$, number of bins called bin_number

Outputs: $indices, bins$

This function defines the minimum and maximum depth for binning in the current specific layer, then generates the bin extremes based on the current layer boundaries and number of bins. A vector of indices for the layer points called $indices$ is then generated and returned through the digitize function, using the bins defined in the preceding line of code. The vector of bin edges called $bins$ is also returned so that they can be used to define the name of each bin .csv file and help determine whether model points are contained in that depth range or not. Two cases had to be defined for the layer boundaries, one for the first layer and one for the others, because the layer vector only contains the boundaries between layer and not the surface layer at 0 km.

The ranges of latitude, longitude and depth to use for the plots are then defined next; these ranges are also used for the MATLAB plots, with the possibility of overwriting them if the user only wants to run the MATLAB scripts for different ranges without having to run the Python part all over again. The next function used is called element_tracker and is used to generate a table containing the stress and viscosity values for the element with highest Mises stress at different stress iterations.

A.3.13. ELEMENT_TRACKER

Inputs: quantity to plot $components\_to\_plot$, run folder path called $base\_path$, depth, latitude and longitude extremes vector called $area\_params$

Outputs: None (Directly saves files)

This function uses the base run path and list of latitude, longitude and depth ranges as only inputs and is divided into two parts. The first one defines the headers for the .csv file to save and initializes a list called files_to_check with all the filenames where the element with the highest stress has to be extracted. Using a series of for cycles, the function loops over each iteration, timestep and stress iteration folder inside them to find the names of data matrices for coupled and uncoupled stresses, and stores them in files_to_check. A matrix of zeros is initialized with as many rows as the number of files found and a number of columns equal to the number of headers called element_table and a counter is initialized. A for loop iterates over files_to_check, opens the corresponding matrix as an array, only extract the rows whose latitude, longitude and depth are in the given ranges and combine them in a smaller matrix which is then processed. The maximum Mises stress value and its latitude and longitude are found in this matrix and then the script uses the B_{diff} and B_{disl} values extracted from the e.dat file opened as a matrix at the same index as the Mises stress to calculate the corresponding viscosity using Equation 1.2. Moreover, the B_{diff} and B_{disl} factors described in Section 4.4 are calculated in order to also display them in the final table.

The element latitude, longitude, stresses, B factors and viscosity are then all concatenated in the same vector which is stored in element_table and the counter is incremented. After the loop over all the files to read, the columns of element_table are swapped 2 by 2 since the coupled stresses are saved before the ones coming from the model because of the alphabetical order in which they are processed. A list of indices is created by concatenating either "output" or "input" to the aforementioned three values converted to strings, indicating whether the stresses in that row are uncoupled (output, coming from the results read from the model) or coupled (input, because coupled stresses are used as model inputs for the next stress iteration). The table of values is converted to a dataframe using pandas and then saved. This function does not influence the data for the final plots but has been necessary to understand what was happening to the model after each stress iteration in a more precise way than just by looking at the plots.

The last function to be run is matlab_variables_writer, which creates a .txt with handles for MATLAB with all the necessary information to find files for plotting and other information such as the latitude, longitude and depth ranges where to plot the results.

A.3.14. MATLAB_VARIABLES_WRITER

Inputs: stress or deflection handle files for MATLAB to know where to get data for plot generation
Outputs: None (Directly saves files)
This is the last function to be run before the workflow shifts to MATLAB, and it is tasked with the creation of a handle file containing a series of variables defining where to find the data to plot. The function receives as input the filenames for the data to plot, whether to plot stresses or deflections, the current stress iteration path, the path where the depth discretized data is saved, whether to work with cartesian or geographical data, the path where the complete stress or deflection files are saved and the list of plotting ranges mentioned before. All the numbers in the list are then converted to strings and the list is then saved in the handle file, which has a .txt format.

A.4. MATLAB CODE

The MATLAB code has mainly been used for plotting purposes at the end of data processing. The main steps for plotting are displayed below.

1. Reading the Python handles and generation of other base parameters;
2. Generation of the stress or deflection plots;
3. Generation of the viscosity or strain rate plots;
4. Generation of the difference plots of selected quantities between two stress iterations.

A.4.1. GEOGRAPHIC_PLOTS

Inputs: MATLAB handle file generated in Python
Outputs: stresses, deformations, viscosity, strain rate plots and their differences in time depending on the user's input
This is the main file used in MATLAB, and as such many of the base parameters related to file definition and organization are found here. The first parameter to be defined is the base folder path where to find data and save the plots called $python\_base\_path$, followed by a figure counter called $figure\_counter$ and the run and run vector run and run_vec, iteration called $iteration$, step and stress iteration numbers called $step$ and $cycle$. $simul\_time$ is the char vector used for plot titles and refers to the time in ka of each timestep in order to make result interpretation clearer. This separation between a single char run and a vector of doubles for run numbers is due to the fact that, in order to plot all the results for this thesis work, it has been necessary to modify the functions and adapt them for both an iterative process where they are called over each run and a single process where the user selects every run, iteration and step number. Moreover, in order to plot all the figures with the same color scale, it was necessary to define all the run numbers used in the report in order to read all the files with the data where the colorbar extremes could be found. After this the definition of the reference system to use and the quantity (stress or deformation to plot) called $coordinate\_system$ and $quantity$ takes place. sd_input is defined based on $quantity$ and of the folder $python\_variables\_path$ relative to the selected stress iteration takes place together with the construction of the current run folder path called run_folder.

The script then searches for the MATLAB .txt handle file, identified by a "matlab" in the name, and selects the right one for plotting based on reference system and quantity. The MATLAB .txt file is then opened and all the variables contained in it are read and initialized, and these are the .csv with values to process called $files\_to\_classify$, the quantity to plot, the report folder called $report\_path$, the directory for the depth discretized files, the reference system, the directory with the files to plot called $complete\_matrices\_path$. The path where to save the stress and deformation figures and the one where to save the difference matrices and plots for two moments in time, respectively called $figures\_path$ and $diff\_matrix\_path$, are also defined. Finally the latitude, longitude and depth range range extremes called $min\_lat, max\_lat, min\_lon, max\_lon, depths\_to\_plot$ are either read from the .txt file produced in Python or redefined. These extremes can be redefined if needed to generate different plots without having to rerun the Python code, as the only thing that is going to change will be the results between the ones in the $element\_tracker$ output matrix and the results in the plots.

The user is then asked whether it is desired to plot the stresses or deformation or not and based on the user's input, the function depth_searcher is called to generate the stress or deformations plots.

A.4.2. DEPTH_SEARCHER

Inputs: run, sd_input, $coordinate\_system$, $complete\_matrices\_path$, list of files that have been processed called $files\_to\_classify$, path where to generate the plots called $figures\_path$, path where to save the matrix for difference calculation $diff\_matrix\_path$, $iteration, step, cycle$, latitude, longitude and depth range extremes called $min\_lat, max\_lat, min\_lon, max\_lon, depths\_to\_plot$, $run_{f}older$, $figure\_counter$, $python\_base\_path$, run_vec, $simul\_time$
Outputs: nothing to pass to other functions; stress or deformation plots
The first function to be potentially called is used for the generation of stress and deformation plots at a certain stress

iteration. The parameter to plot is decided between stresses and deformations based on sd_input and then the path where to save the .csv file for the calculation of differences in time is defined based on the quantity to plot and reference system to use, then the list of parts for which to plot the values (in this case only EARTH) is initialized. Next is a definition of variables necessary for plotting, such as the plot resolution, ranges and gridded mesh for latitude and longitude, which are then reshaped for successive plot generation. The user is then prompted to enter a list of either stress or deformation components to plot, and for each of them the number of the corresponding column of the part file is selected and saved in a vector. The colorbar extremes across all runs are then calculated for every variable to plot using caxisextremes.

For each of the parts selected, 1 in this case, the depth range for plotting is read and the matrix to read is selected, extracting the depth, latitude and longitude for all centroids or nodes. The indices of the points which respect the condition of having their depth, latitude and longitude in the intervals defined in the main files are saved and then the matrix where to store values for the difference in time, called $matrix\_for\_difference$, is allocated. This matrix will contain the selected stress or deformation component together with the latitude, longitude and depth only for the points respecting the range conditions described earlier.

A for cycle iterating over the selected columns to plot starts, where the variable to plot is defined by only extracting the values of the current variable at the indices satisfying the depth, latitude and range condition. The latitude and longitude are also trimmed to just the values in the above ranges, they are saved together with the current trimmed variable to plot in $matrix\_for\_difference$. The headers for $matrix\_for\_difference$ are also defined here, and after this the plot is generated. Producing the plot requires the additional definition of the radial distance from the origin of the reference system for all of the points in the selected ranges, which is done by subtracting their depth from the radius property of the uniform Earth sphere object defined in MATLAB. This is necessary since the viscosity plots require a 3D cartesian interpolation to be performed or would otherwise display numerical errors with a "strippy" pattern. Since the viscosity plots have been improved this way, it has been chosen to also use this method for all the other plots to generate. After calling the latitude, longitude and radius of all the points in the desired range, two conversions to cartesian coordinates are performed, one on this set of coordinates and another on the reshaped latitude, longitude and r vector defined at the beginning of the script. After performing this operation a scatter plot is generated with the point distribution for the selected mesh to perform the visual check of Section 3.4 and the data is then interpolated with griddata, using the set of cartesian points in the depth range as input points and the cartesian coordinates of the latitude, longitude and r vectors defined at the beginning of the script as query points using the 'nearest' option. The plot is then generated using the $geoloc2grid$ function using the reshaped latitude and longitude from the script beginning together with the interpolated variable to plot and the resolution and the colorbar limits from caxisextremes. The plot is then saved. $matrix\_for\_difference$ is then also saved after exiting the for cycle over each component, because while the plots have to be generated for each component, the matrix contains all of them in the same file. The function then terminates.

CAXISEXTREMES

Inputs: $sd_input, min_lat, max_lat, min_lon, max_lon, depths_to_plot$, list of components to plot called $selected_components$, $viscosity\_input$,b_input, $python\_base\_path$, run_vec, $coordinate\_system$
Outputs: vector of minima and maxima for colorbar plotting called $colorbarlimits$
This function is used to give the same color scale to all stresses and deformations components to plot and also to additional values such as the viscosity and B coefficients. Because of this, it contains different sections dedicated to different possible quantities to plot and returns a colorbar extremes vector based on the minimum and maximum values for any of the desired components found across the selected run folders. This has been done to better compare plots from different moments in time and using different rheologies and background stresses.

The function iterates over the numbers of the runs to process supplied as a vector of one or more elements, reads a list of all the files with the same name in the same run folder (in this case related to part EARTH in ENU coordinates) in a path defined in the main and based on the function inputs. The function iterates over each run number, and if stresses or deflections have to be processed, the function reads all the component vector across all stress iteration directories in the run folder and extracts the minimum and maximum for each of them in the latitude, longitude and depth ranges selected for plotting. These minimum and maximum values are organized with minimum and maximum in succession for each component on each column and these values are returned for every stress iteration on each row. The minimum and maximum are then found over each corresponding column. The resulting vector of minima and maxima is then assigned to a row of a matrix called $extremes\_matrix$ containing the colorbar limits for each component and run, with the components being on the columns and the runs on the rows. This procedure is the same for any variable to plot. For the viscosity, the stress file is opened and strain rate and viscosity are calculated, then the chosen quantity (in this case viscosity) has its minimum and maximum values displayed on two columns on one row for each stress iteration. In this case too, the function calculates the minimum and maximum across the rows to have a value for the entire run. Again, this colorbar limits vector is stored in the corresponding run row of $extremes\_matrix$ in order to calculate the minimum and maximum of the viscosity over all the runs. If the B coefficients plots have to be generated, they follow a similar procedure to the aforementioned ones. After iterating over all the runs, The function calculates the minima and maxima for each column of $extremes\_matrix$ and returns a vector of minima and maxima, with a number of minima equal to the components to plot followed by an equal number of maxima in the case of stresses, deformations and B coefficients and just with one minimum and one maximum for the viscosity.

The user is prompted to enter whether they want to plot the viscosity or strain rate or not after defining $viscosity\_figures\_path$, the path where to save the resulting plots. If the user desires, plots are generated with the function visco_plotter.

A.4.3. VISCO_PLOTTER

Inputs: sd_input, base stress iteration folder path called $python\_variables\_path$, $coordinate\_system$, $complete\_matrices\_path$, stress or deformation report path called $report\_path$, min_lat, max_lat, min_lon, max_lon, $depths\_to\_plot$, $iteration$, $step$, $cycle$, $diff\_matrix\_path$, run_folder, run, $figure\_counter$, $python\_base\_path$, run_vec, $simul\_time$
Outputs: viscosity or strain rate plots, path where to find matrices for generation of plots of viscosity or strain rate differences in time
This function is similar to depth_searcher in that its only task is creating variable plots, but it can only be used on part EARTH. This happens because viscosity is calculated based on B_{diff} and B_{disl}, which are only given for EARTH. Again, the quantity to plot first and the path where to save the matrix for the difference calculation of viscosity or strain rate in time called $visco\_diff\_path$ second, are defined. Again, a series of vectors used for interpolation and plotting has to be defined at the beginning of the function, This will be the same in the B coefficient plot generation function. The path where to save the plots is read or created, the colorbar limits are generated across all runs then the e.dat file and complete stress file for EARTH are opened and the relevant values for viscosity calculation are read from their columns. The is calculated as shown in Equation 1.2 while the strain rate is calculated with Equation 1.1 divided by Δt.

The viscosity, strain rate and Mises stress are all saved in a matrix which is then concatenated to the one for the current part. In the equation, W is the Mises stress and pow is equal to 3.5, while

$$\begin{aligned} \dot{\epsilon} &= B_{diff} W + B_{disl} W^{pow} \\ \mu &= \frac{W}{3\dot{\epsilon}} \end{aligned} \tag{A.1}$$

Again, the range conditions for depth, latitude and longitude are stated and the viscosity or strain rate vectors trimmed together with the latitude and longitude ones based on these conditions. The $matrix\_for\_difference$ and its headers are also again defined, then the matrix gets filled with the trimmed variable to plot, latitude and longitude. The plot is then generated after the same interpolation process presented in depth_searcher and saved and with it $matrix\_for\_difference$, since this time we do not have to iterate over different components to plot first. The B_{diff} and B_{disl} plots are then also generated with B_$plots$, then the function is exited.

B_PLOTS

Inputs: path to the viscosity plot called $viscosity\_figures\_path$, B coefficients called with their FORTRAN names $alin$, a, for B_{diff}, B_{disl} respectively, point indices in the selected range $data\_points\_indices$, part name $part$, min_depth, max_depth, min_lat, max_lat, min_lon, max_lon, lat, lon, run, run_folder, $viscosity\_input$, $python\_base\_path$, $depth$, run_vec
Outputs: B coefficients plots
This function uses as input the opened values for B_{diff} and B_{disl} coming from visco_plotter and generates a plot iterating over each of them by using the same interpolation procedure and plot generation commands of depth_searcher and visco_plotter. The use made of caxisextremes is also repeated here.

After this, the user is prompted to input whether they want to plot the difference plots of a chosen quantity between two stress iterations, and if so, the diff_calculator function starts. Another function can also be used to plot the differences between dry and wet rheologies for the B coefficients.

A.4.4. DIFF_CALCULATOR

Inputs: $diff\_matrix\_path\_incomplete$, min_lat, max_lat, min_lon, max_lon, run, $depths\_to\_plot$, $iteration$, $step$, $figure\_counter$
Outputs: difference plots of a selected quantity for the same run between two stress iterations
This last function generates plots of the difference between all the available quantities that exist in two timesteps.

The first operation in the function is the definition of what quantity to plot(stresses, deformations, viscosity or strain rate) and, if stress or strain rate are selected, the quantity itself and then the reference system. Viscosity and strain rate only require the definition of the quantity. The folders for the $diff\_matrices$ to be used and the plots to be generated are also defined here accordingly, and the latter is created if it does not already exist. The user is then prompted to input two quintuplets of values containing the iteration, step, stress iteration and depth range for the quantity to plot. For both quintuplets, matching $diff\_matrices$ are searched for in their folder and returned if they are found.

With these two matrices and the depth range, the headers are extracted and compared for common labels after taking out the latitude and longitude names, since they will also be in common. The components contained in both matrices are extracted and a for loop iterates over their number. For each of them, the difference between the two columns of the two matrices at different times is calculated and then plotted based on the corresponding latitude and longitude vectors. The plots are then generated and saved in the corresponding plot folder.

A.4.5. B_DIFF_CALC

Inputs: $b\_diff\_plots\_folder, python\_base\_path, min\_lat, max\_lat, min\_lon, max\_lon, run, depths\_to\_plot, figure\_counter$

Outputs: difference plots of a selected quantity for the same run between two stress iterations

This function is very similar to the previous one both in principle and how it works, but it had to be separated from it due to the fact that the B coefficients only vary based on the rheology and not the selected moments in time. Therefore, the only main difference with diff_calc here aside from different file paths is that instead of using 5 values for iteration, step, stress iteration and depth range extremes to generate the difference plots only three values for the run (odd for dry rheology and even for wet) and depth extremes are used to find the files where the values for the B plots are stored and calculate the differences between them.

A.5. DESCRIPTION OF THE FORTRAN FILES

Analyzing the model in ABAQUS requires the use of one or more FORTRAN files which are one of the inputs to the job definition commands if the user needs a user-defined input such as material laws (in this case, the creep behavior). Two files have been used for this thesis work, but since the differences between them are small and in small number, they will both described in this section. user_2 is the base FORTRAN file received among the other python files at the start of the thesis, while user_2_mises is its modified version to be used in the analysis loop defined for this thesis project.

The user_2 file is divided in two parts, one for each declared subroutine. The first one, UEXTERNALDB, is used to define file paths which need to be used by other subroutines and always needs to be used at the beginning of the script. In our case it is used to declare the parameters to use for the creep law and the file where they are included. In the original file there are two of them read from the e.dat file, the creep parameters B_{diff} and B_{disl}, which for diffusion and dislocation creep. The next subroutine is CREEP, where the creep law is defined with Equation 1.4.

The user_2_mises script is very similar to user_2, with the following differences:

1. The path to the e.dat file is different for better file management;
2. There is an additional variable, QMISES, which is the coupled Mises stress;
3. The variable QTILD now takes the value of coupled Mises stresses (new variable QMISES) read from the e.dat file instead of using the compressive stress taken from the model.

Because the definition of the additional variable $QMISES$ needs the coupled Mises stress to be calculated, the user_2_mises file is used in all the model analysis stress iterations except the first one, where the .odb file is first created and with it the first base stresses.

A.6. ADDITIONAL FUNCTIONS

A.6.1. SYSVER

This small MATLAB function reads one of the files generated in Python after joining coordinates and deformations and then applies a coordinate transformation specified by the user in the form of swapping data columns. Once this is done, the function generates a 3D scatterplot of a selected quantity, only used as a means to check whether it is in the right place on the Earth or not.

A.6.2. MISES_VERIFIER

This Python function reads one of the stress matrices generated in ABAQUS and downloaded from Hipparchos (any of them will work), then opens it as an array and uses Equation 1.5 to calculate the Mises stress from the stress components. It then calculates the absolute value of the difference between the Mises vector calculated this way and the Mises vector read from the matrix and then the relative difference between the two by dividing this difference vector by the Mises stress read from the matrix. The maximum value of this relative difference is then printed.

A.6.3. FORCE_DEFINITON_ROUTINE

The objective of this section of code is to infer, from the distribution of flow velocities at a certain height, the depth trend of the stress components from that depth to the lower mantle. This part of the program will be the translation to Python scripting of the literature study presented in Section 4.5 and has been developed towards the end of the thesis work, using many of the tools developed for the data processing step described in Section 2.3. Therefore, a large part of this section of code has been taken from that and slightly modified. However, even though this part has been added last and has used code developed for later operations, the way this code works will be explained here so that the reader won't have to jump back and forth between this and Section 2.3 for an explanation of the code.

As a first step, the node and element information for part EARTH has to be extracted from the model .dat file. This is done by opening such file and defining a series of lists which are going to contain the starting and ending lines for the definition of elements for all parts. Keywords delimiting the data regions for part EARTH and the section of the file containing data of interest for this work are also defined. The file is scanned to find all keywords occurrences to find the file section where the node definition for part EARTH starts and all the sections where elements start and stop being defined for all parts. The file line where nodes stop being defined is extracted as the first value in the list of lines where elements start being defined, called $elem\_lines$. The file lines in that interval are then scanned to save all the nodes in part EARTH. After this something similar is done for the elements: the file line where elements stop being defined is extracted as the first value in the list of lines where element sets start being defined, called $nset\_lines$. All the elements for part EARTH are then also read and stored.

The stress matrix for part EARTH is then opened and each of the labels of the stress matrix is paired with the corresponding element so that the stress tensor values are paired with a series of nodes, the nodes making up the element with that same label. Once this has been done, each node of the element is read and stored in a matrix and their coordinates are searched in the node matrix using their index. The result is a matrix of a certain number of nodes (based on the element type) with 3 columns containing the Z, X and Y coordinates of each node since the ABAQUS reference system is still being used. The mean of every coordinate is then calculated to obtain the centroid coordinate for each element. This is then done for all elements of part EARTH and the centroid coordinates together with the corresponding stress values are then saved in a .csv file. An if cycle has been implemented to skip this part if the file already exists since generating this matrix can take around 20 minutes due to having to search for each node coordinate making up each element.

Once this has been done, the Earth radius taken from Williams (2020) is defined and the centroid coordinates are used to calculate the radial distance of each point from the origin of the model at the center of the Earth with R equal to R_3D in Equation 2.3 and their depth, derived as displayed in Equation A.2.

$$depth = Earth\_radius - R\_3D \tag{A.2}$$

The R_3D and depth column vectors are concatenated to the matrix containing stresses and centroid coordinates, called $data\_matrix$. The matrix will keep this name throughout this section of code. The next operation is dividing all the stress data based on the depth of every point, and this is done through the use of depth bins and the numpy digitize function. Moreover, the bin index of the first bin with an upper edge equal to 660 km, which is the depth of the boundary between upper and lower boundary, is saved. This is needed because viscosity changes between the two mantle layers and therefore the value to use to calculate τ_h, as per Equation 4.1. The next step is the calculation of the scaling the base shear and normal stresses displayed im Equation 4.1 with depth, since the velocity u_0 is going to vary, as said before, with a "roller" law between a maximum of u_0 below the plate to a minimum of $-u_0$ at the CMB. This is done through the use of the digitize numpy function, which uses as input bins of values defined through the number of bins and then maximum and minimum depth of all the points. Once these bins have been defined, a linearly spaced vector of velocities is generated also based on the number of bins and the τ_h and τ_n are calculated for each of these velocities. Then, based on bin index vector generated previously, each point with a certain index is associated to the velocity value in the corresponding position of the velocity list. The τ_h and τ_n vectors with the values for each point are then concatenated to $data\_matrix$.

The last procedure to take place is the combination of realistic stresses and the ones coming from the model. The angle of rotation between the velocity and the geographical reference system is defined and two matrices are initialized. These matrices will contain the cartesian and geographical stress components for all points, called S_matrix and S_geo respectively. A for cycle starts and iterates over every $data\_matrix$ line, which means over every element. Here τ_h and τ_n are used to fill a 2-by-2 matrix which is converted to geographical coordinates as shown by Equation 4.3. This matrix is then used to fill a 3-by-3 geographical stress tensor as described in Equation 4.5, which is converted to cartesian coordinates as shown in Equation 4.6.

The stress components are now in the ABAQUS reference system and they are used to fill 6 out of the 7 columns of S_matrix. Since the tensor is symmetric, ABAQUS only reports the ones on the upper side of the diagonal and those will be the ones to be swapped around. The first column is filled with the Mises stress, which is calculated with the law used by ABAQUS and described in Equation 1.5. This law has been taken from Kim (2009), and it has been verified through a small MATLAB script used to calculate with Equation 1.5 the Mises stress through the components read from any .csv file and then to subtract the Mises stress data column from the resulting vector of values. This matrix with the realistic stresses for each element is then concatenated to $data\_matrix$, which is then overwritten.

Additional plots

This section of the appendix will include other plots which have not been discussed in the main part of the report but are nonetheless interesting to look at.

B.1. Normal and shear components stress plots

The plots included here are the S11 and S12 stress components, which are the components to which background stresses have been added at each stress iteration to simulate stress coupling. The first plots to be shown, Figure B.1 and Figure B.2, display a net stress increase between stress iteration 1 of timesteps 1 and 2 for both the normal and shear stress component again due to the constant addition of ambient stresses. Again, all the plots will be displayed for a 1 MPa background stress.

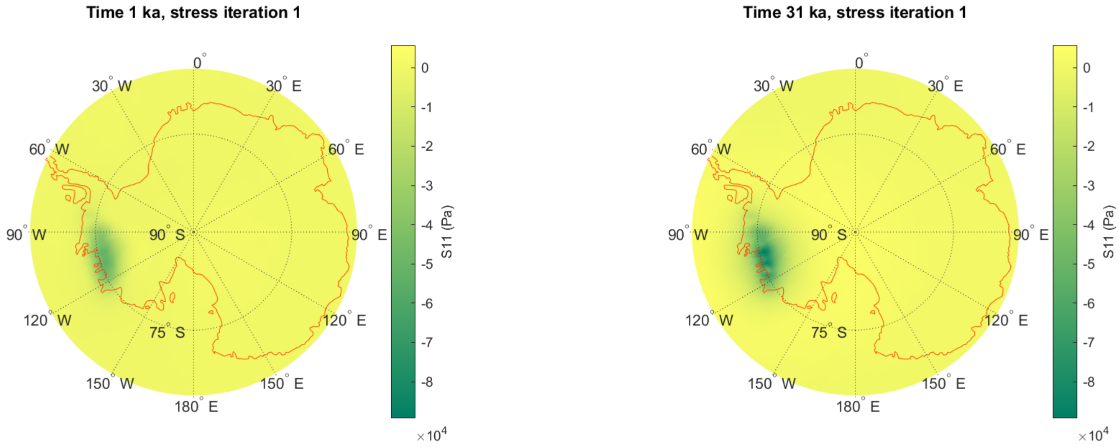


Figure B.1: Stress component S11 for two different time snapshots, one at stress iteration 1 of timestep 1 and one at stress iteration 1 of timestep 2 for a depth range of 145 to 160 km.

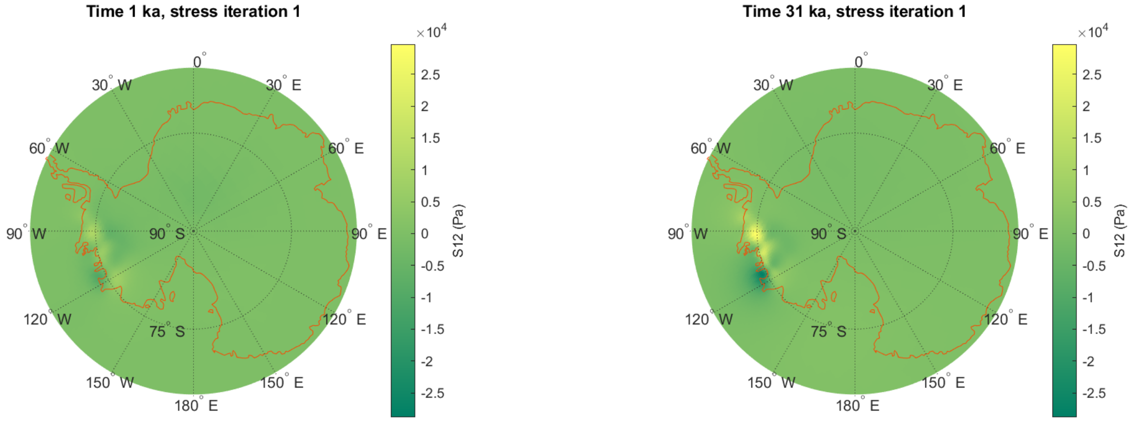
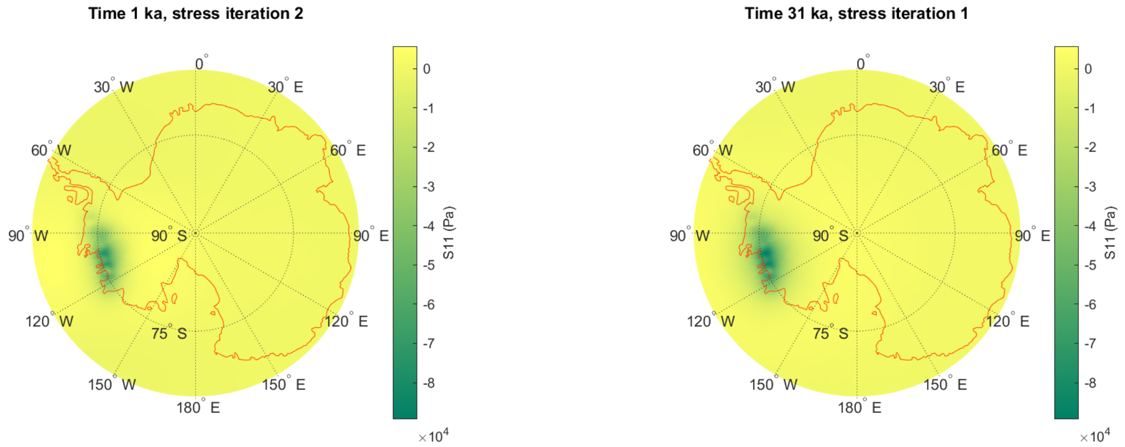


Figure B.2: Stress component S12 for two different time snapshots, one at stress iteration 1 of timestep 1 and one at stress iteration 1 of timestep 2 for a depth range of 145 to 160 km.

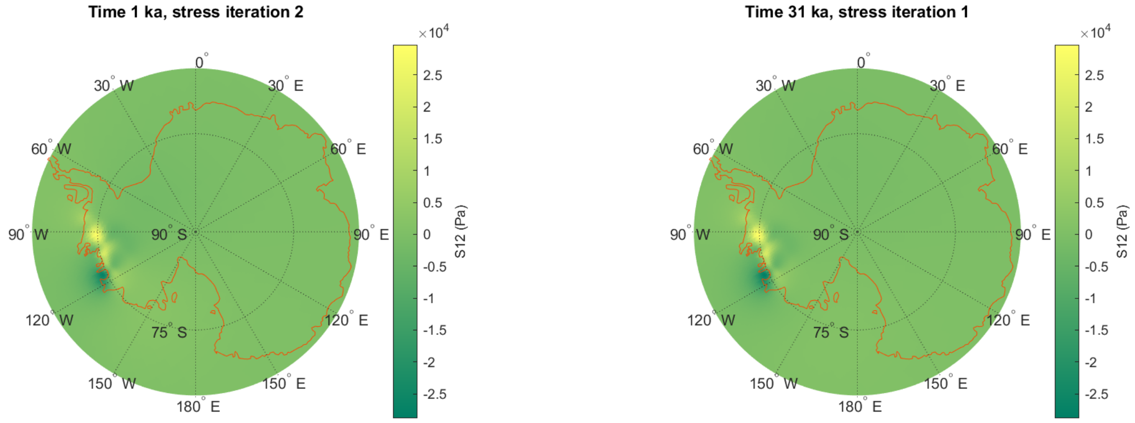
Figure B.3 shows a small stress decrease between stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2 for stress component S11.


Time 1 ka, stress iteration 2

Time 31 ka, stress iteration 1

Figure B.3: Stress component S11 at two different time snapshots, stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2, for a depth range of 145 to 160 km, showing the small increase in stresses.

Compared to S11, S12 slightly decreases between the last two time snapshots, more closely following the Mises stress evolution.


Time 1 ka, stress iteration 2

Time 31 ka, stress iteration 1

Figure B.4: Stress component S12 at two different time snapshots, stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2, for a depth range of 145 to 160 km, showing the small decrease in stresses.

At a higher depth range, S11 and S12 behave slightly differently because S11 increases in absolute value and S12 decreases between stress iterations 1 and 2 of timestep 1. However, both components decrease between stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2 and both of them show a net decrease between stress iteration 1 of timestep 1 and 2. This net decrease is again explainable by the large distance from the ice load and higher relaxation, which compared to low depths does not allow the stresses to increase due to the lack of glacial ones. For this reason, the only plots generated for S12 will be the ones showing a different behaviour with respect for S11, which means stress iterations 1 and 2 of timestep 1. Figure B.5 shows a stress increase between stress iterations 1 and 2 of timestep 1 for a depth range of 545 to 550 km for stress component S11.

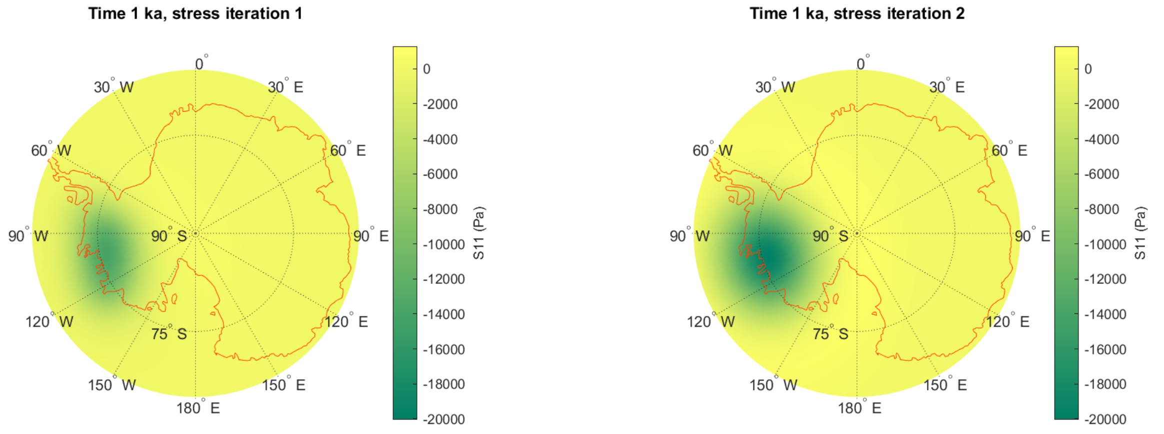


Figure B.5: Stress component S11 for two different time snapshots, one at stress iteration 1 of timestep 1 and one at stress iteration 2 of timestep 1 for a depth range of 545 to 550 km.

Figure B.6 shows how S12, compared to S11, decreases between the same time snapshots as the previous couple of plots.

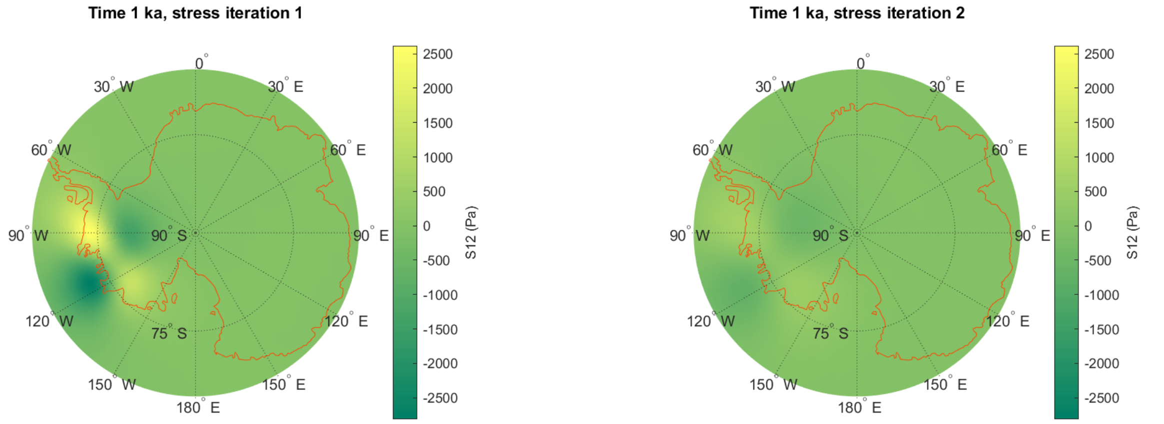


Figure B.6: Stress component S12 for two different time snapshots, one at stress iteration 1 of timestep 1 and one at stress iteration 2 of timestep 1 for a depth range of 545 to 550 km.

Figure B.7 shows the strong stress reduction at higher depths between stress iteration 2 of timestep 1 and stress iteration 1 of timestep 2 for the higher depth range. This is again observed for both S11 and S12 and for this reason only S11 is plotted here.

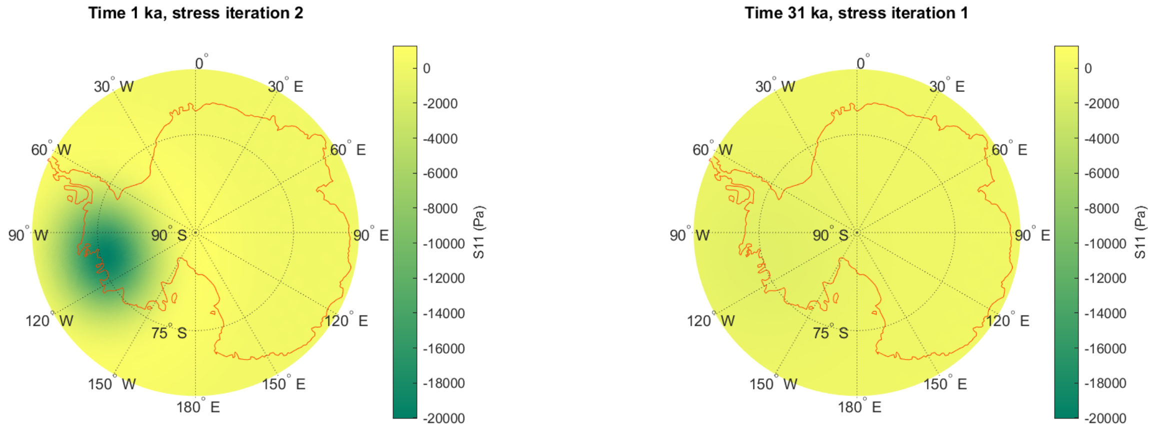


Figure B.7: Stress component S11 for two different time snapshots, one stress iteration 2 of timestep 1 and one at stress iteration 1 of timestep 2 for a depth range of 545 to 550 km.

The fact that the Mises stress seems to be more in agreement with the evolution of S12 is due to the fact that S11, S22 and S33 are all negative, which means from Equation 1.5 that they reduce each other's influence in the calculation of the Mises stress, which therefore depends more on the shear stress components. Because they are all negative, they increase in absolute value where the Mises stress which is positive decreases but due to being at odds with each other's values they end up not influencing the Mises stress a lot.

Figure B.8 shows noticeable stress differences when considering stress iteration 1 of timestep 1 between dry and wet rheology. This initial difference in results between rheologies which quickly turns to 0 is observed for both stress components, so only S11 will be displayed.

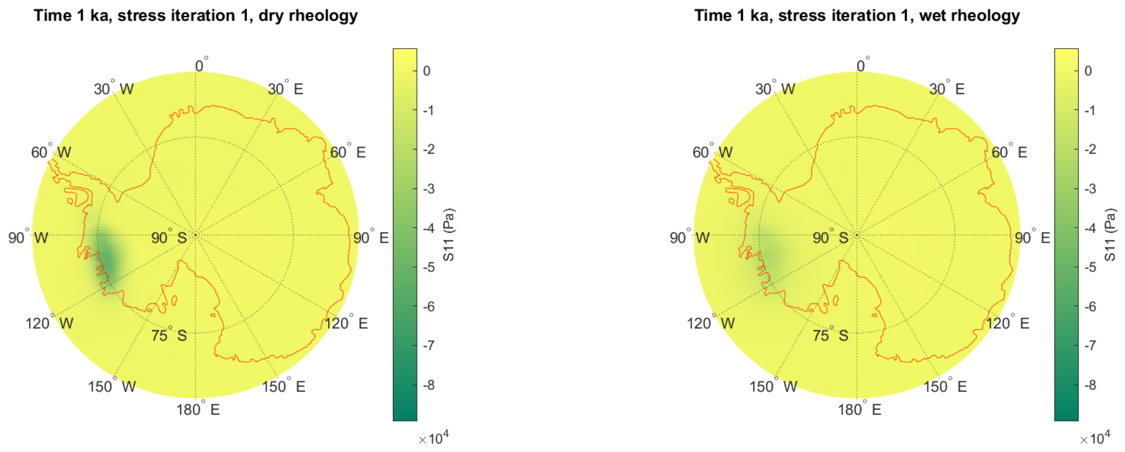


Figure B.8: Comparison of the results from the first stress iteration of the first time step for the S11 stress component between results for a dry and wet rheology, with a background stress of 1 MPa and for a depth range of 145 to 160 km, showing discernible differences in the plotted values.

Figure B.9 shows how the differences of the previous couple of plots are reduced to 0 for stress iteration 2 of timestep 1 already.

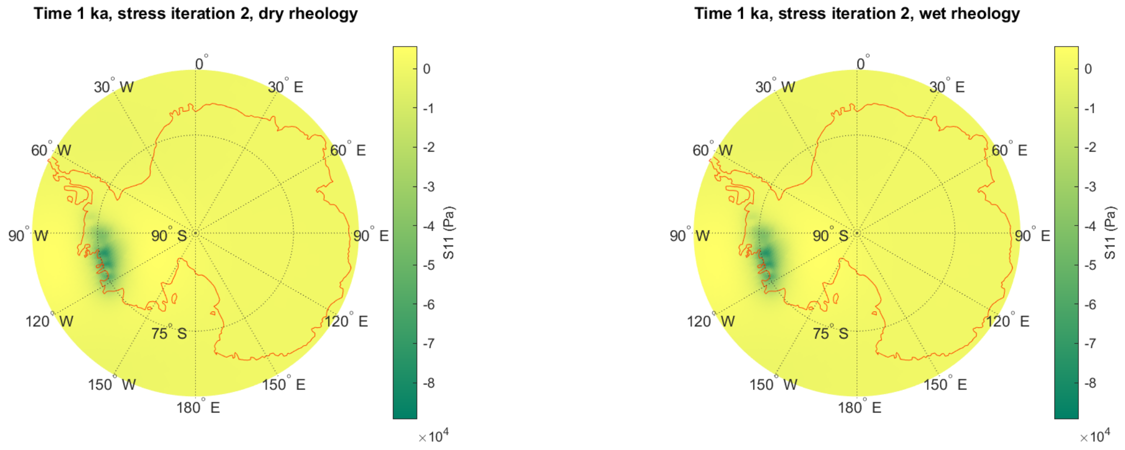


Figure B.9: Comparison of the results from the second stress iteration of the first time step for the S11 stress component between results for a dry and wet rheology, with a background stress of 1 MPa and for a depth range of 145 to 160 km, showing that the differences have strongly reduced.

B.2. Vertical deformation component plots

The plots for the vertical deformation component U3 have been displayed here, showing how similar they are to the deformation magnitude plots due to the fact that vertical deformation is the largest component making up the total magnitude.

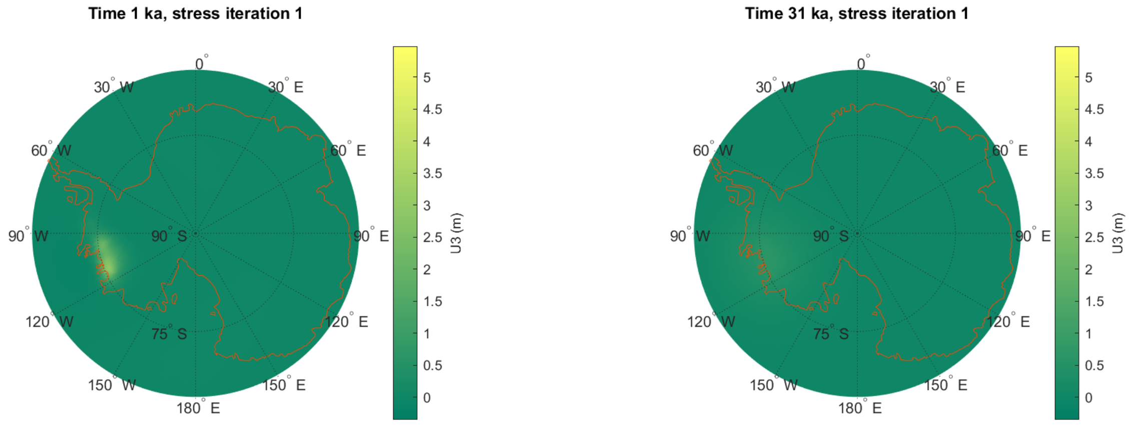


Figure B.10: Vertical deformation component U3 for two different time snapshots, one at stress iteration 1 of timestep 1 and one at stress iteration 1 of timestep 2 for a depth range of 96 to 123 km.

The higher depth range still shows the same evolution as the deformation magnitude.

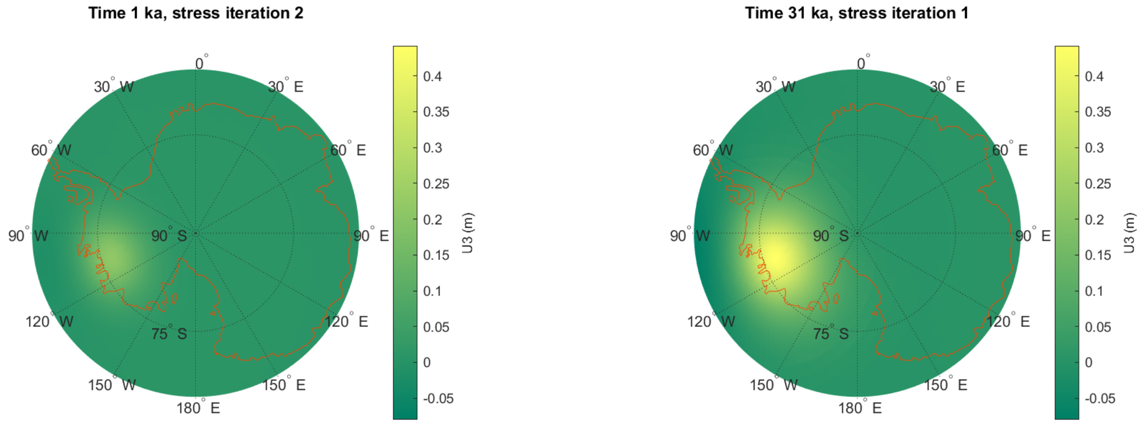


Figure B.11: Vertical deformation component U3 for two different time snapshots, one at stress iteration 2 of timestep 1 and one at stress iteration 1 of timestep 2 for a depth range of 502 to 510 km.

Again, using a wet rheology causes noticeable differences in deformations at stress iteration 1 of timestep 1.

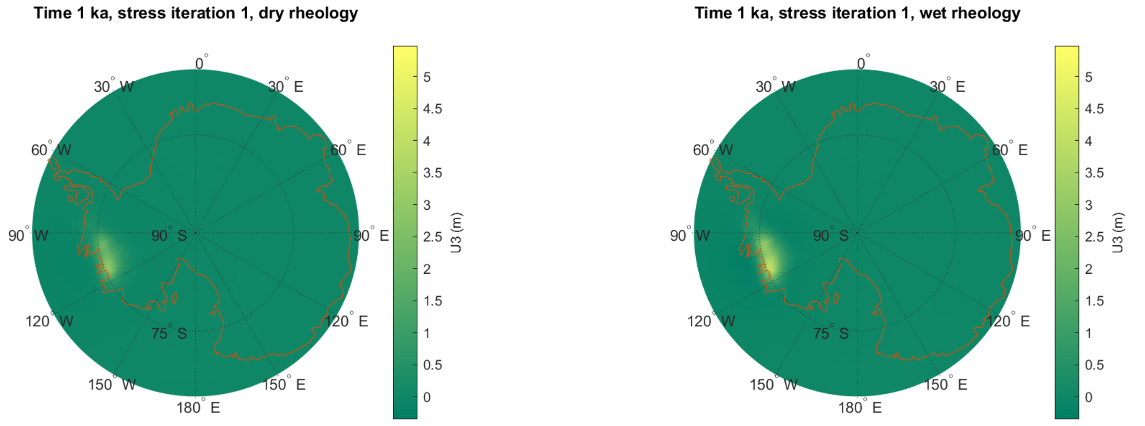


Figure B.12: Comparison of the results from the first stress iteration of the first time step for the vertical deformation component U3 between results for different rheologies, with a background stress of 1 MPa and depth range of 96 to 123 km, showing discernible differences in the plotted values.

The differences between dry and wet rheologies are again quickly reduced to 0 for stress iteration 2, just like the deformation magnitude.

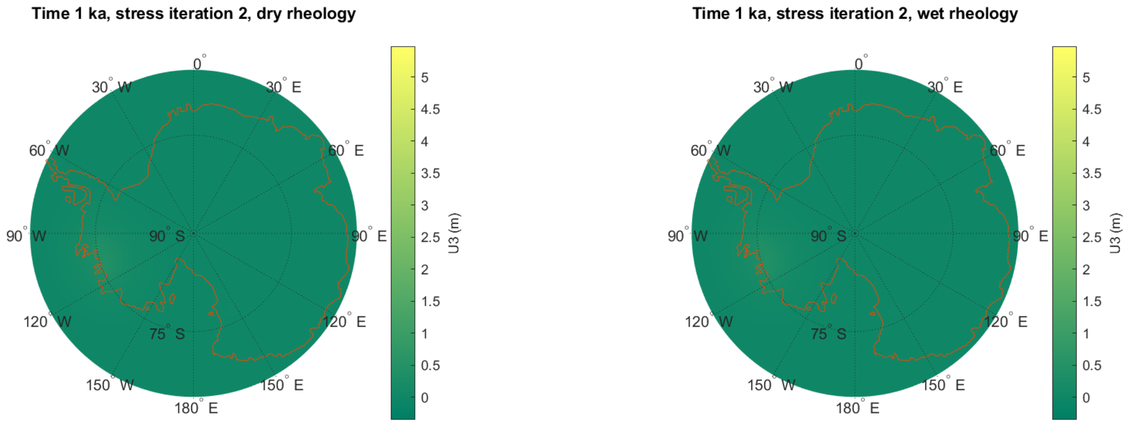


Figure B.13: Comparison of the results from the second stress iteration of the first time step for the vertical deformation component U3 between results for different rheologies, with a background stress of 1 MPa and depth range of 96 to 123 km, showing that the differences have strongly reduced.

B.3. B COEFFICIENTS PLOTS

As explained before, these plots show how the viscosity decreases due to a larger B_{diff} for a wet rheology and how the viscosity pattern is basically only influenced by their distribution at higher depths. This makes sense considering that the influence of the ice load and the deriving stresses on the Earth's surface tends to become smaller and smaller as the depth increases, thus removing the high relaxation area which is visible at lower depths. Moreover, lower depths clearly display the viscosity difference between East and West Antarctica. Compared to the Mises stress, the B_{diff} and B_{disl} variables only change based on the rheology file used and the selected depth range, which leads to a smaller amount of plots. Figure B.14 and Figure B.15 show that B_{diff} increases when using a wet rheology, while B_{disl} is the same, at lower depths.

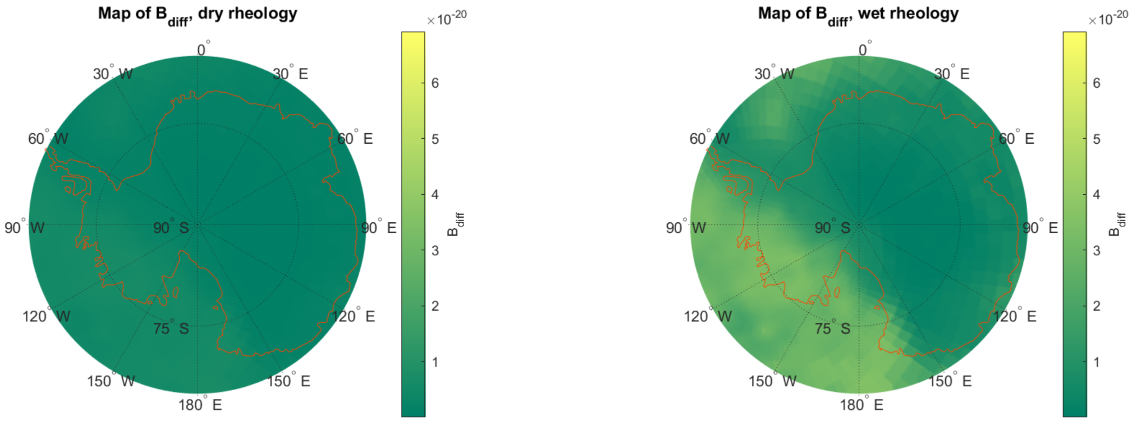


Figure B.14: B_{diff} comparison for a depth range of 145 to 160 km and a dry rheology on the right and a wet rheology on the left.

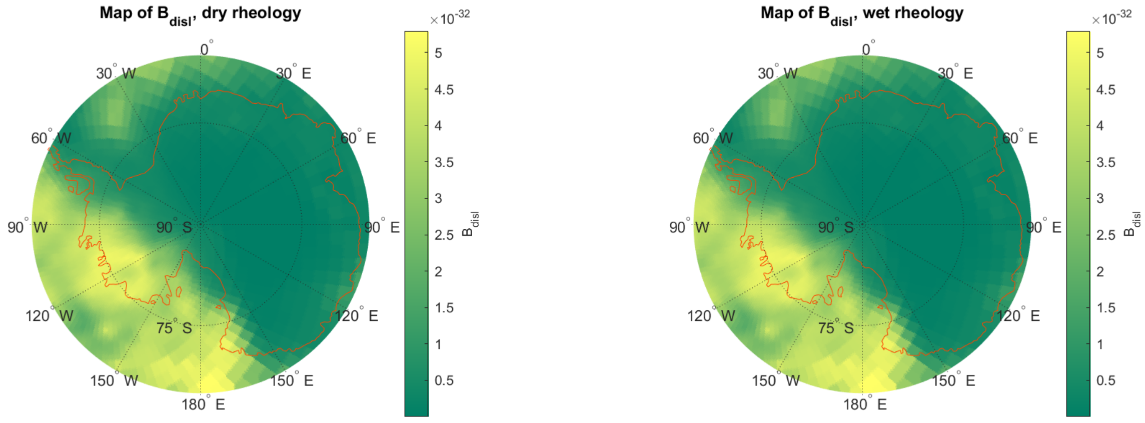


Figure B.15: B_{disl} comparison for a depth range of 145 to 160 km and a dry rheology on the right and a wet rheology on the left.

The same trend, with a B_{diff} increase and unchanged B_{disl}, is found in Figure B.16 and Figure B.17 for higher depths.

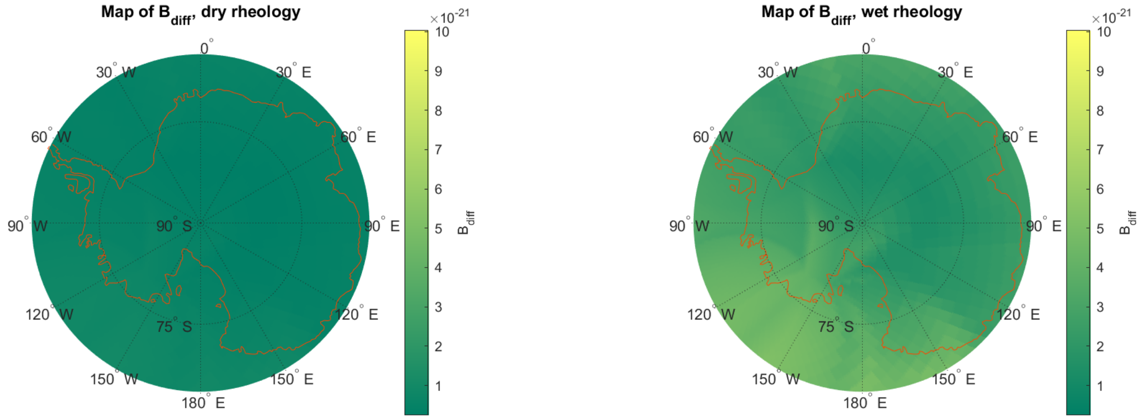


Figure B.16: B_{diff} comparison for a depth range of 545 to 550 km and a dry rheology on the right and a wet rheology on the left.

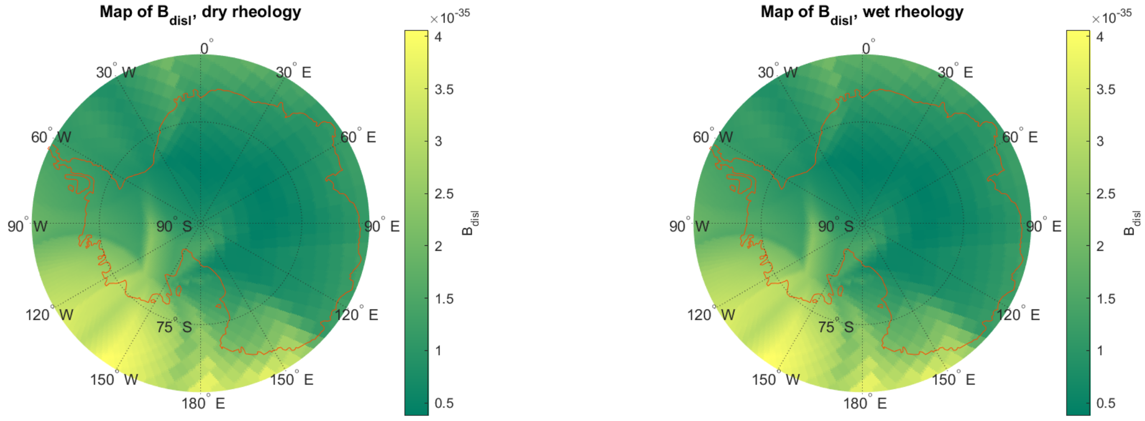


Figure B.17: B_{disl} comparison for a depth range of 545 to 550 km and a dry rheology on the right and a wet rheology on the left.

For a more comprehensive overview of the B_{diff} differences between dry and wet rheology, the plots displayed in Figure B.18 have been generated. These plots show more clearly that B_{diff} tends to stay at the same order of magnitude between the two rheologies, which is the cause of the results not changing except for the very first stress iteration for stresses and deformations when the rheology is changed.

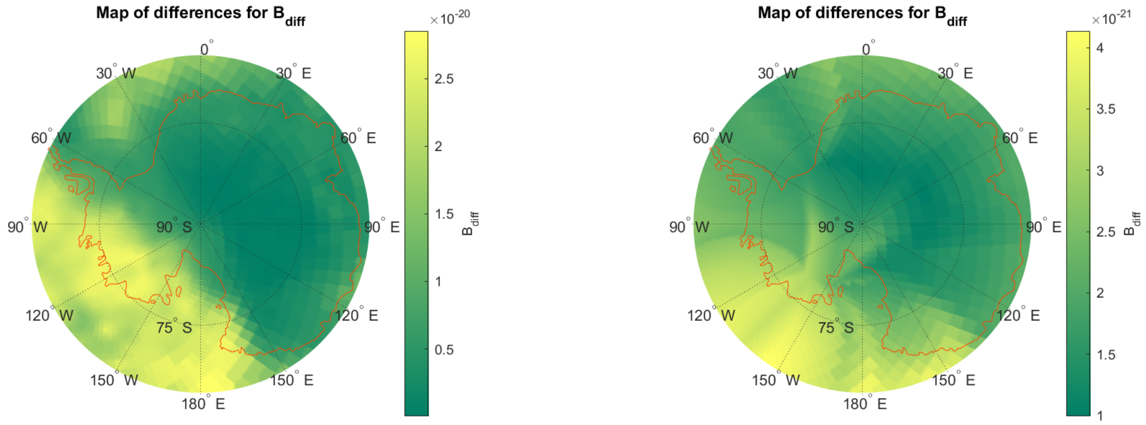


Figure B.18: B_{diff} difference plots between dry and wet rheology for a depth range of 145 to 160 km on the left and 545 to 550 km on the right.

Bibliography

M. An, D. A. Wiens, Y. Zhao, M. Feng, A. Nyblade, M. Kanao, Y. Li, A. Maggi, and J.-J. Lévêque. Temperature, lithosphere-asthenosphere boundary, and heat flux beneath the antarctic plate inferred from seismic velocities. *Journal of Geophysical Research: Solid Earth*, 120(12):8720–8742, 2015.

V. R. Barletta, M. Bevis, B. E. Smith, T. Wilson, A. Brown, A. Bordoni, M. Willis, S. A. Khan, M. Rovira-Navarro, I. Dalziel, et al. Observed rapid bedrock uplift in amundsen sea embayment promotes ice-sheet stability. *Science*, 360(6395): 1335–1339, 2018.

D. Bercovici and S.-i. Karato. Whole-mantle convection and the transition-zone water filter. *Nature*, 425(6953):39–44, 2003.

B. Blank, V. R. Barletta, H. Hu, F. Pappa, and W. van der Wal. Evaluation of differences between 1d, 3d and 4d models in the amundsen sea embayment region by use of finite element based gia models. *Geochemistry, Geophysics, Geosystems*, 2020.

M.-N. Bouin and C. Vigny. New constraints on antarctic plate motion and deformation from gps data. *Journal of Geophysical Research: Solid Earth*, 105(B12):28279–28293, 2000.

A. F. Bower. *Applied Mechanics of Solids by Allan F. Bower* . CRC Press LLC, 2009. ISBN 0471491101.

E. Bredow and B. Steinberger. Mantle convection and possible mantle plumes beneath antarctica - insights from geodynamic models and implications for topography. *The Geological Society Books*, 2020.

M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global positioning systems, inertial navigation, and integration*. John Wiley & Sons, 2007.

C. Grigné, S. Labrosse, and P. Tackley. Convection under a lid of finite conductivity in wide aspect ratio models: Effect of continents on the wavelength of mantle flow. *Journal of Geophysical Research: Solid Earth*, 112(B8), 2007.

J. Hewitt, D. McKenzie, and N. Weiss. Large aspect ratio cells in two-dimensional thermal convection. *Earth and Planetary Science Letters*, 51(2):370–380, 1980.

G. Hirth and D. Kohlstedf. Rheology of the upper mantle and the mantle wedge: A view from the experimentalists. *Geophysical Monograph-American Geophysical Union*, 138:83–106, 2003.

T. Höink and A. Lenardic. Three-dimensional mantle convection simulations with a low-viscosity asthenosphere and the relationship between heat flow and the horizontal length scale of convection. *Geophysical research letters*, 35(10), 2008.

T. Höink, A. Lenardic, and M. Richards. Depth-dependent viscosity and mantle stress amplification: implications for the role of the asthenosphere in maintaining plate tectonics. *Geophysical Journal International*, 191(1):30–41, 2012.

S.-i. Karato, M. R. Riedel, and D. A. Yuen. Rheological structure and deformation of subducted slabs in the mantle transition zone: implications for mantle circulation and deep earthquakes. *Physics of the Earth and Planetary Interiors*, 127 (1-4):83–108, 2001.

G. Kaufmann and K. Lambeck. Mantle dynamics, postglacial rebound and the radial viscosity profile. *Physics of the Earth and Planetary Interiors*, 121(3-4):301–324, 2000.

G. Kaufmann, P. Wu, and E. R. Ivins. Lateral viscosity variations beneath antarctica and their implications on regional rebound motions and seismotectonics. *Journal of Geodynamics*, 39(2):165–181, 2005.

P. Kelly. Lecture Notes: An introduction to Solid Mechanics. http://homepages.engineering.auckland.ac.nz/~pkel015/SolidMechanicsBooks/Part_III/Chapter_1_Vectors_Tensors/Vectors_Tensors_13_Coordinate_Transformation_Tensors.pdf, 2015.

N.-H. Kim. Failure Theories. https://mae.ufl.edu/nkim/eas4200c/VonMisesCriterion.pdf, 2009.

Z. Martinec, V. Klemann, W. van der Wal, R. Riva, G. Spada, Y. Sun, D. Melini, S. Kachuck, V. Barletta, K. Simon, et al. A benchmark study of numerical implementations of the sea level equation in gia modelling. *Geophysical Journal International*, 215(1):389–414, 2018.

MIT. Hybrid elements. https://abaqus-docs.mit.edu/2017/English/SIMACAEGSARefMap/simagsa-c-ctmhybrid.htm, 2020.

P. Montone, M. T. Mariucci, S. Pondrelli, and A. Amato. An improved stress map for italy and surrounding regions (central mediterranean). *Journal of Geophysical Research: Solid Earth*, 109(B10), 2004.

F. Morra. sample_run_folder. https://github.com/fmorra/sample_run_folder, 2020a.

F. Morra. code_thesis. https://github.com/fmorra/code_thesis, 2020b.

A. Osei Tutu, B. Steinberger, S. V. Sobolev, I. Rogozhina, and A. A. Popov. Effects of upper mantle heterogeneities on the lithospheric stress field and dynamic topography. *Solid Earth*, 9:649–668, 2018.

W. Peltier, D. Argus, and R. Drummond. Space geodesy constrains ice age terminal deglaciation: The global ice-6g_c (vm5a) model. *Journal of Geophysical Research: Solid Earth*, 120(1):450–487, 2015.

J. Pohjola, J. Turunen, T. Lipping, A. Sivula, and M. Marila. *Historical Perspectives to Postglacial Uplift: Case Studies from the Lower Satakunta Region*. Springer, 2018.

H. Schmeling. On the interaction between small-and large-scale convection and postglacial rebound flow in a power-law mantle. *Earth and planetary science letters*, 84(2-3):254–262, 1987.

P. Schmidt, B. Lund, T. Arnadottir, and H. Schmeling. Glacial isostatic adjustment constrains dehydration stiffening beneath iceland. *Earth and Planetary Science Letters*, 359:152–161, 2012.

G. Schubert, D. L. Turcotte, and P. Olson. *Mantle convection in the Earth and planets*. Cambridge University Press, 2001.

G. Spada, V. R. Barletta, V. Klemann, R. Riva, Z. Martinec, P. Gasperini, B. Lund, D. Wolf, L. Vermeersen, and M. King. A benchmark study for glacial isostatic adjustment codes. *Geophysical Journal International*, 185(1):106–132, 2011.

C. Spencer, J. Murphy, C. Hoiland, S. Johnston, R. Mitchell, and W. Collins. Evidence for whole mantle convection driving cordilleran tectonics. *Geophysical Research Letters*, 46(8):4239–4248, 2019.

H. Steffen and P. Wu. Glacial isostatic adjustment in fennoscandia—a review of data and modeling. *Journal of geodynamics*, 52(3-4):169–204, 2011.

S.Widnall. Lecture L3- Vectors,Matrices and Coordinate Transformations. https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-07-dynamics-fall-2009/lecture-notes/MIT16_07F09_Lec03.pdf, 2009.

W. van der Wal, A. Barnhoorn, P. Stocchi, S. Gradmann, P. Wu, M. Drury, and B. Vermeersen. Glacial isostatic adjustment model with composite 3-d earth rheology for fennoscandia. *Geophysical Journal International*, 194(1):61–77, 2013.

W. van der Wal, P. L. Whitehouse, and E. J. Schrama. Effect of gia models with 3d composite mantle viscosity on grace mass balance estimates for antarctica. *Earth and Planetary Science Letters*, 414:134–143, 2015.

P. L. Whitehouse, M. J. Bentley, and A. M. Le Brocq. A deglacial model for antarctica: geological constraints and glaciological modelling as a basis for a new model of antarctic glacial isostatic adjustment. *Quaternary Science Reviews*, 32:1–24, 2012a.

P. L. Whitehouse, M. J. Bentley, G. A. Milne, M. A. King, and I. D. Thomas. A new glacial isostatic adjustment model for antarctica: calibrated and tested using observations of relative sea-level change and present-day uplift rates. *Geophysical Journal International*, 190(3):1464–1482, 2012b.

D. R. Williams. Earth Fact Sheet. https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html, 2020.

P. Wu. Deformation of an incompressible viscoelastic flat earth with powerlaw creep: a finite element approach. *Geophysical journal international*, 108(1):35–51, 1992.

P. Wu. Postglacial induced surface motion and gravity in laurentia for uniform mantle with power-law rheology and ambient tectonic stress. *Earth and Planetary Science Letters*, 186(3-4):427–435, 2001.

P. Wu. Using commercial finite element packages for the study of earth deformations, sea levels and the state of stress. *Geophysical Journal International*, 158(2):401–408, 2004.

M. L. Zoback. First-and second-order patterns of stress in the lithosphere: The world stress map project. *Journal of Geophysical Research: Solid Earth*, 97(B8):11703–11728, 1992.