Marine outlet glacier dynamics in Greenland

Bachelor project
Written by *Helene Pehrsson*

Supervised by
Christine Schøtt Hvidberg

University of Copenhagen

UNIVERSITY OF COPENHAGEN

Name of Institute:	Niels Bohr Institute
Name of Department:	Faculty of Science
Author(s):	Helene Pehrsson
Email:	helene.pehrsson@nbi.ku.com
Title and subtitle:	Marine outlet glacier dynamics in Greenland -
Supervisor(s):	Christine Schøtt Hvidberg
Handed in:	10.60.2020
Defended:	19.06.2020

Name Helene Pehrsson

Signature

Date 10/06/2020

Abstract

The focus of this project is to understand the physical properties of marine outlet glaciers. It also aims to build two dynamic glacier models to study the effects of the climate forcings on marine outlet glaciers and how changes in is this can cause instabilities in the glacier.

The simplest model is based upon the assumption, that the evolution of the ice sheet can be described only by the length of the glacier and the climate forcings. The second model takes a more complex approach in 3 dimensions and is based upon the principle of mass conservation and Glen's flow law.

The latter model is the main focus of the project and is used to model conditions at the Helheim glacier in south-eastern Greenland. The model gives nice results when exploring how changes in different parameters, can affect the glaciers stability and result in mass gain or loss. However a number of flaws where found, when it came to recreating the conditions of the Helheim glacier. Here bottom sliding has a big influence on the behavior of the glacier and can therefore not be neglected, as it has been in the model.

Contents

1 Introduction

Over the past decades, ice sheets all over the world have been retreating. And in the last couple of years the process has sped up for especially the Greenland ice sheet. As described in IPCC's 2019 SROCC report [1], this has let to the Greenland ice sheet surpassing Antarctica and now being one of the biggest contributors to the global mass loss. According to the IPCC's 2013 5th assessment report [2] this is due to the warming of the oceans, causing the glaciers to melt faster than new ice can accumulate. This results in a higher melt rate, leading to mass loss. This has sparked a growing interest in the behavior of the Greenland ice sheet, because of its large impact on the global sea level, coupled atmospheric and oceanic circulation in the North Atlantic.

The focus of this project is to understand the parameters resulting in stability of marine outlet glaciers and test how small changes can affect the balance. The goal is to build a dynamic glacier model to study the effects of the climate forcings on marine outlet glaciers and how they can cause instability in the glaciers. In the IMBIE Team's 2019 article [3], they describe how the discharge for many years has been a larger contributor to mass loss than the surface mass balance. However, in the past couple of years the surface melting has increased, but it may also impact the marine glaciers and cause further retreat.

Building a dynamic model of a marine outlet glacier can very quickly get complicated, due to the sheer number of physical properties involved. This project takes a simple approach to glacier dynamics, consisting of the building and testing of two simple glacier models.

The simplest of the models relies on very few assumptions and is used here to better understand the climate forcings that drives the advance and retreat of a marine outlet glacier. This model is based upon the model discribed in Oerlemans and Nick (2005) [4]. The more complicated model relies on more physical properties of glacier dynamics, but still makes a number of approximations to simplify the process. This model is based upon Oerlemans (1981) [5]. The latter model is the main focus of the project and has been used to model conditions at the Helheim glacier in the south-eastern Greenland. This is done in an attempt to mimic the behavior of Helheim as described in Nick et. al. 2009 paper [6].

2 Description of the models

The physics of a marine outlet glacier is very complicated thus requiring a huge amount of parameters to be taken into account when modeling the dynamics of such a glacier. In an attempt to simplify the processes of such a marine outlet glacier, this project deals with two simplified numerical models. One model is based upon the simple assumption, that the evolution of the ice sheet can be described only by the length of the glacier and the climate forcing, in this case the mass budget. The second model takes a more complex approach in 3 dimensions and is based upon the principle of mass conservation and Glen's flow law.

2.1 Minimal model

This model is a very minimal glacier model described by Oerlemans and Nick in their 2005 paper [4]. The model is built on the basic assumption that the mean ice thickness H_m and the ice thickness at the glacier front H_f can be related to the length L of the glacier in these two simple equations:

$$H_m = \alpha_m L^{1/2} \tag{1}$$

$$H_f = max\{\alpha_f L^{1/2}; -\delta d\}. \tag{2}$$

Here α_m and α_f are constants related to the bulk flow parameter, d is the elevation of the bed and δ is the water to ice density ratio.

The evolution of the ice sheet is found from the conservation of mass, which can be written as

$$\frac{dV}{dt} = B + F, \tag{3}$$

where B is the total gain or loss of ice at the surface and F is the ice flux at the glacier front. F can be written as a function of the elevation of the bed d, the calving rate c and the height at the glacier front H_f:

$$F = min\{0; cdH_f\}. \tag{4}$$

It is shown that the change in glacier length can be written as

$$\frac{dL}{dt} = \frac{2(B+F)}{3\alpha_m} L^{1/2}, \tag{5}$$

The change in glacier length is determined by the total change in the mass budget and the shape of the bed, where the total mass change is the calving rate and the surface mass balance averaged over the whole ice sheet.

This entire model is build upon eq. 5, which finds the length of the current time step based upon the length of the previous time step, the total gain of ice and the ice flux of the previous time step. This new length is then used to find the new values for B and F. Here B and F are two climate forcing parameters, that can be changed to test the model. Since this model only relies on the length to grow, the glacier cannot grow from nothing. The model requires an initial length of the ice sheet > 0. For this project the initial value has been $L = 1$, since the model reacts quite fast.

2.2 Vertically integrated model

The grounds for this model are based upon a vertically integrated ice sheet model described by Oerlemans in 1981 [5], which solves the zero-order shallow ice approximation of the stress equilibrium equations with Glen's flow law:

$$\epsilon = A\tau^n. \tag{6}$$

This equation describes the relationship between the dominant shear stress τ and the corresponding strain rate ϵ. The flow parameter A has a strong dependence on the ice temperature and the fabric of the ice, as well as crystal orientation and debris content of the ice. Values that can differ from ice sheet to ice sheet. The power n is approximately a constant ranging from about 1.5 to 4.2. In this case it is assumed that $n = 3$, which is most consistent with field data.

This model is a three-dimensional model, meaning that it is dependent on the width as well as the height of the ice sheet and can therefore not be described using only the height and the laminar velocity. The evolution of the ice sheet is also depending on the conservation of mass and can be written as

$$\frac{\partial H}{\partial t} = -\frac{1}{W}\frac{\partial(H\overline{u}W)}{\partial x} + b, \tag{7}$$

where H is the ice thickness, $\overline{u}$ is the laminar velocity, W is the width of the ice sheet and b denotes the mass balance.

Another way of looking at the evolution of the ice sheet is to see the transport of mass, from accumulations to ablation zones, as a diffusive process. Knowing this eq. 7 can be written as

$$\frac{\partial H}{\partial t} = -\frac{1}{W}\frac{\partial}{\partial x}\Big(WD\frac{\partial S}{\partial x}\Big) + b, \tag{8}$$

where the diffusivity D is positive and found to be

$$D = \frac{2A(\rho g)^n}{n+2}H^{n+2}\Big(\frac{\partial S}{\partial x}\Big)^{n-1}. \tag{9}$$

Here S is the ice surface, ρ is the ice density and g is the gravitational acceleration. So now the spread of the ice sheet can be described using a nonlinear diffusion equation, where the diffusivity increases with the slope of the surface and the thickness of the ice sheet. This implies that D tends to be larger where the ice sheet is thicker. Therefore to reduce variations in D, the ice sheet will have larger surface slopes where thickness is smaller.

2.2.1 Ice shelf

When the glacier is grounded it ends in a steep slope towards the ground. But as the glacier begins to advance into the ocean, the ice front will either have to break off or begin to float. To make the model more realistic in this situation, an ice shelf is attached at the tide water margin. This prevents the model from advancing when the glacier isn't grounded. The shelf is a confined shelf of uniform width and follows the same rules as the shelf described in Cuffey and Paterson's chapter 8.9.3 [7].

Figure 1 shows a cross-section of the ice shelf, where H_M is the full height of the shelf and H_S is height of the shelf portion under water. Since the shelf is floating on the seawater and not grounded H_S is not equal to the water depth but can be written as

$$H_S = \frac{\rho H_M}{\rho_w}, \tag{10}$$

where ρ is the ice density and ρ_w is the seawater density. Since the condition of floating equilibrium requires that

$$H\frac{dS}{dx} = \Big(1 - \frac{\rho}{\rho_w}\Big)H\frac{dH}{dx}, \tag{11}$$

the driving force can be written as

$$F_D = \frac{1}{2}\rho g\Big(1 - \frac{\rho}{\rho_w}\Big)H^2. \tag{12}$$

Assuming that the shelf is in steady state with a constant width and mass balance rate, we can neglect the friction at the sides and assume that the shelf is very wide. With these assumptions the longitudinal spread can be written as

$$\dot{\epsilon_x} = A\Big(\frac{\rho g}{4}(1 - \frac{\rho}{\rho_w})\Big)^n H^n \tag{13}$$

and the mass balance can be written as

$$M = H\frac{\partial u}{\partial x} + u\frac{\partial u}{\partial x}. \tag{14}$$

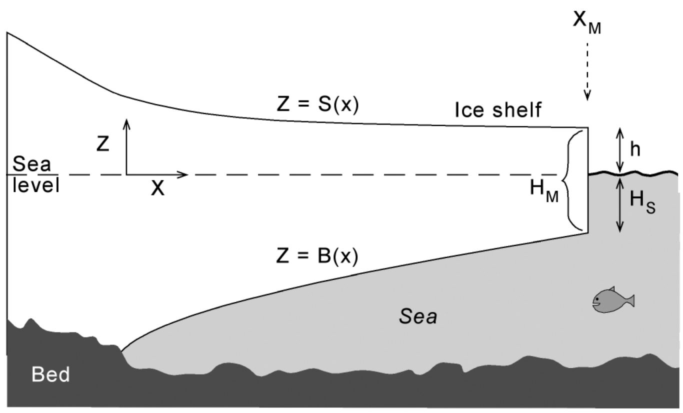

Figure 1: Cross-section of an ice shelf (dimensions not to scale). The figure is from Cuffey & Paterson [7], chapter 8.9.3 page 374.

The shelf grows from the flow of ice flowing into the shelf from the whole area on the ice sheet and looses mass due to melt and calving into the ocean. When the flow of mass into the shelf is higher than the melt off $M > 0$, the shelf will become thicker and at some point get grounded. When this happens the ice sheet will advance further into the water. However when the flow of mass into the shelf is smaller than the melt off $M < 0$, the shelf starts to thin out and the glacier will start to retreat and possibly start floating further in.

3 Implementing the model

To implement the vertically integrated model, all the physical properties of the glacier have to be in a format that can work within a computer model. This means that all the equations have to be rewritten so that they can work on a grid structure. In this grid the $i = 0, ..., n$ refers to the x-coordinates and $j = 0, ..., k$ refers to the time-coordinates.

The model is driven by an accumulation rate, which can be constant as well as vary over time and the model is not dependent on having an initial height profile. This means that the ice sheet can grow from no ice and still reach a steady state profile within a reasonable amount of time, depending on accumulation and ablation.

3.1 Staggered grid

To avoid problems with the growth of the ice sheet the spatial discretization is done on a staggered grid, where the surface, height and bottom of the glacier is calculated with a collocated grid. Where as the diffusivity and the surface slope is calculated with a staggered grid. This means that the values S, H and B are evaluated at the regular grid points, where D and $\frac{\partial S}{\partial x}$ are evaluated exactly halfway between the grid points. Using a staggered grid instead of only a collocated grid allows for a more accurate assessment of the equations, as well as providing the most simple geometry needed for solving different equations simultaneously. The staggered grid makes sure that the height of the glacier is never $H = 0$ at where $\frac{\partial S}{\partial x}$ is calculated, since this would result in an infinite velocity.

3.2 The physics of the model

Since the height of the glacier is calculated on a collocated grid, rewriting eq. 7 is quite simple

$$\left(\frac{\partial H}{\partial t}\right)_i^j = \frac{H_i^{j+1} - H_i^j}{dt}. \tag{15}$$

From eq. 9 we know that the diffusivity is depending on the height, therefore H has to be calculated on a staggered grid as well

$$H_{i+\frac{1}{2}}^j = \frac{H_{i+1}^j + H_i^j}{2}. \tag{16}$$

The surface slope $\frac{\partial S}{\partial x}$ has to be calculated on the staggered grid as well and can be written as

$$\left(\frac{\partial S}{\partial x}\right)_{i+\frac{1}{2}}^j = \frac{S_{i+1}^j - S_i^j}{dx}. \tag{17}$$

When these two values are calculated on a staggered grid, it is important to remember that they can only be calculated up to $i = n - 1$, where as the values calculated on a collocated grid can be calculated up to $i = n$.

Now using the expressions for H and $\frac{\partial S}{\partial x}$ on the staggered grid, the diffusivity from eq. 9 can be written as

$$D^j_{++\frac{1}{2}} = \frac{2A(\rho g)^n}{n+2}\left(H^j_{i+\frac{1}{2}}\right)^{n+2}\left(\left(\frac{\partial S}{\partial x}\right)^j_{i+\frac{1}{2}}\right)^{n-1} \tag{18}$$

and the width can be written as

$$W_{i+\frac{1}{2}} = \frac{W_{i+1}+W_{i-1}}{2}. \tag{19}$$

Using the three expressions for $\frac{\partial S}{\partial x}$, W and D the following expression can be derived from eq. 8 and can be written as

$$\left[WD\frac{\partial S}{\partial x}\right]^j_i = \frac{1}{dx}\left(W_{i+\frac{1}{2}}D_{i+\frac{1}{2}}\left(\frac{\partial S}{\partial x}\right)^j_{i+\frac{1}{2}} - W_{i-\frac{1}{2}}D_{i-\frac{1}{2}}\left(\frac{\partial S}{\partial x}\right)^j_{i-\frac{1}{2}}\right). \tag{20}$$

To reduce the complexity of the following equations, a new expression for the diffusivity including the width is introduced here

$$\tilde{D}^j_{i\pm\frac{1}{2}} = W^j_{i\pm\frac{1}{2}} \cdot D^j_{i\pm\frac{1}{2}}. \tag{21}$$

Using the new expression for diffusivity, we can derive an equation to describe the height of the glacier at the same grid point for the next time step. The equation can be written as

$$H^{j+1}_i = H^j_i + dt\left[b^j_i + \frac{1}{(dx)^2}\left[S^j_{i-1}\frac{\tilde{D}^j_{i-\frac{1}{2}}}{W_i} - S^j_i\frac{\tilde{D}^j_{i-\frac{1}{2}}+\tilde{D}^j_{i+\frac{1}{2}}}{W_i} + S^j_{i+1}\frac{\tilde{D}^j_{i+\frac{1}{2}}}{W_i}\right]\right]. \tag{22}$$

To make it easier for the model to compute a matrix called $\underline{\underline{C}}$ of size $[nxn]$ is introduced. The different elements of the matrix are given as

$$C_{i,i-1} = \frac{\tilde{D}^j_{i-\frac{1}{2}}}{W_i}\ ,\quad C_{i,i} = \frac{\tilde{D}^j_{i-\frac{1}{2}}+\tilde{D}^j_{i+\frac{1}{2}}}{W_i}\ ,\ C_{i,i+1} = \frac{\tilde{D}^j_{i+\frac{1}{2}}}{W_i}. \tag{23}$$

Now the expression for glacier height eq. 22 can be written as the following vector function

$$\underline{H}^{j+1} = \underline{H}^j + dt\left[\underline{b}^j + \frac{1}{(dx)^2}\,\underline{\underline{C}}^j\underline{S}^j\right]. \tag{24}$$

This function calculates all the glacier heights at all grid points in the same time step simultaneously. This function is what drives the model and the main parameter to change when testing the model will be the mass balance b.

3.3 Growing profile

When the surface profile of the model ends on land it grows by letting the front move one grid point forward, when the ice is thick enough. When it moves forward this grid point has a very small height resulting in a tiny triangle forming and growing, until the height at this point is big enough to let the front move again, this can be seen in figure 2.a. This way of growing results in an oscillation of the height of the glacier front.

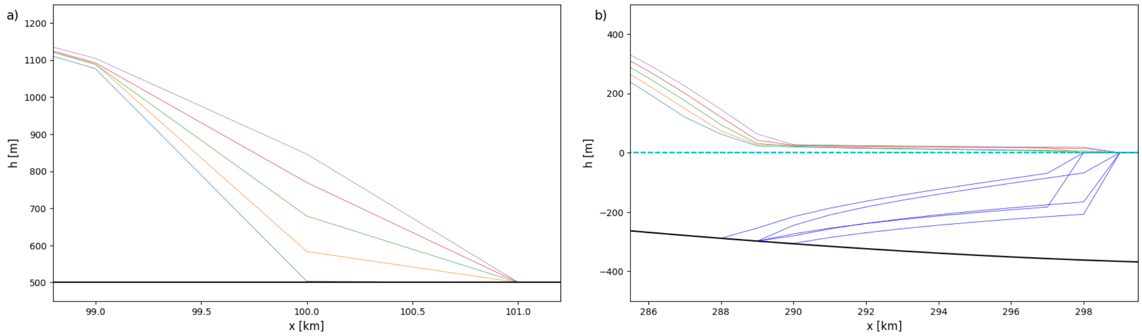

Figure 2: **a)** This figure shows how the glacier advances on land, by jumping to the next grid point.
b) This figure shows how the glacier advances in the ocean, by jumping to the next grid point.

When the surface profile ends in water, an ice shelf is put on the end of the glacier. This shelf is basically passive and has no influence on the behavior of the ice sheet it self. The shelf just makes sure that the glacier can float when the water is to deep compared to the glacier height. When the model has to advance in water it grows the ice shelf thicker until the point where the glacier is grounded and the model thereforee is able to jump to the next grid point. When the grounding line of the glacier jumps it allows the front of the shelf to thin out and jump to the next grid point, this can be seen in figure 2.b.

4 Testing the models

To verify that the two models are giving reasonable results, they have been run through two simple tests, both thought out by Oerlemans and Nick in their 2005 paper [4]. The first test considers a case with uniform mass balance through out the ice sheet and calculates how the glacier evolves over time until it reaches steady state. In the second case the mass balance is dependent on the altitude of the ice sheet, with an equilibrium line that oscillates over time, with accumulation above it and ablation below it.

For both the models the bottom topography has been chosen rather arbitrarily and is calculated like this

$$d(x) = d_0 - s_x + \lambda \, e^{-[(x-x_s)/\sigma]^2}. \tag{25}$$

This equation describes a bed with a linear downward slope, with a Gaussian bump. The height of the bump is λ, the width is σ and the position of the bump is determined by x_s. The reason for choosing a bedrock with a bump is to study how the glacier evolves when it advances or retreats over increasing and decreasing water depths.

4.1 Test 1: The minimal model

The first test of the minimal model considers a case with uniform mass balance over the ice sheet. For this case the total gain or loss of ice at the surface can be calculated as

$$B = La \tag{26}$$

where the accumulation rate a is increased by 0.0005 $m\ yr^{-1}$ and the calving rate of the glacier is $c = 2.4\ m\ yr^{-1}$. The Gaussian bump on the bed is located at $x_s = 40\ km$ from the ice divide, with a height of $\lambda = 300\ m$ and a width of $\sigma = 10\ km$.

The model was set to run for 5000 years and gave the results shown in figure 3.a. In the figure it is clearly seen how the length of the glacier grows over time, slowing down at around $L = 20\ km$ when it reaches sea level and making a big jump forward when the bump begins. When the glacier passes the bump the glacier approaches an equilibrium, when the ocean becomes so deep, that the glacier cannot stay grounded.

The figure also shows the progression of the water depth, the height at the front and the two components of the mass balance. Here there is also a clear growth until the glacier reaches the bump and again after it passes the bump. As seen the total gain of ice at the surface begins to increase and the negative flux decreases as expected.

4.2 Test 2: The minimal model

In the second test the mass balance is dependent on the altitude of the ice sheet, with an oscillating equilibrium line. This is a more realistic approximation for marine outlet glaciers located in a warmer climate and therefore have large ablation zones. Here the flux is the same as for the first test (eq. 4), but the total gain of ice at the surface is calculated as

$$B = \beta(h_m - E)L, \tag{27}$$

where the balance gradient is $\beta = 0.005\ yr^{-1}$ and h_m is the mean surface elevation. E is the equilibrium line which is calculated as

$$E = E_0 + A_E\ sin\Big(\frac{2\pi t}{P_E} + f\Big). \tag{28}$$

In this case $E_0 = 100\ m$, $A_E = 350\ m$, $P_E = 5\ kyr$ and $f = 1.6$. The glacier has accumulation below the equilibrium line and ablation above it.

The results of this test can be found in figure 3.b, where it is clear that the glacier does not reach an equilibrium when it passes the bump, like it did in the other test. This is due to the ablation of ice caused by the oscillating equilibrium line. When the glacier reaches its maximum length, it slowly begins to loose mass until it passes the bump, where the retreat increases rapidly until the glacier is gone again. This process is cyclic and will repeat itself, with growing the glacier and melting it away again.

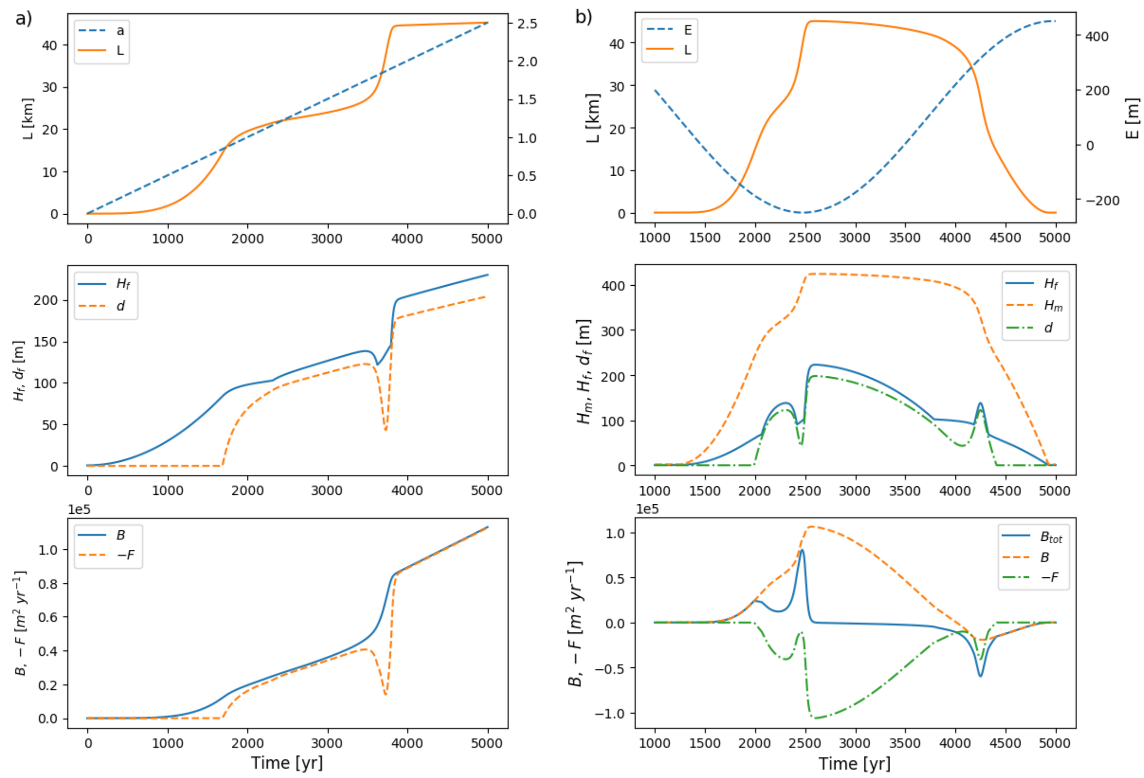

Figure 3: **a)** This figure shows the test on the minimal model, with an accumulation rate increasing linearly over time by 0.0005 $m\ yr^{-1}$. The first panel shows the length and accumulation rate. The second panel shows the ice thickness and the water depth at the front. The third panel shows the two components of the mass budget.
b)This figure shows the result for test of the minimal glacier with mass balance depending on altitude and a linear balance gradient and periodic forcing ($P_E = 5kyr$). The first panel shows the glacier length and the equilibrium line altitude E. The second panel shows the mean ice thickness, the thickness at the glacier front and water depth. The components of the mass budget are plotted in panel three.

4.3 Test 1: The vertically integrated model

This model was also run through the same two test as the minimal model. The first test was with a constant mass balance of 1 $m\ y^{-1}$ on the ice sheet and $-1\ m\ yr^{-1}$ on the shelf. The same bottom topography from eq. 25 was used here, with the Gaussian bump located at $x_s = 350\ km$ from the ice divide, with a height of $\lambda = 1\ km$ and a width of $\sigma = 35\ km$. Since this model reacts slower than the minimal model and has a much larger distribution, the model was given an initial Vialov profile of length $L = 265km$ and set to run for 150 kyr. This length was chosen so the model would not have to run unnecessarily long, but also ensuring the glacier would not have reached sea level yet.

In figure 4.a the result for the test can be seen. Here it is clear that the glacier grows steadily until it passes the bump. The fluctuations in the length and the height at the front is due to the shelf building up and becoming grounded. When the glacier passes the bump the growth slows down and comes to a stop as the seawater becomes to deep for the shelf too be grounded.

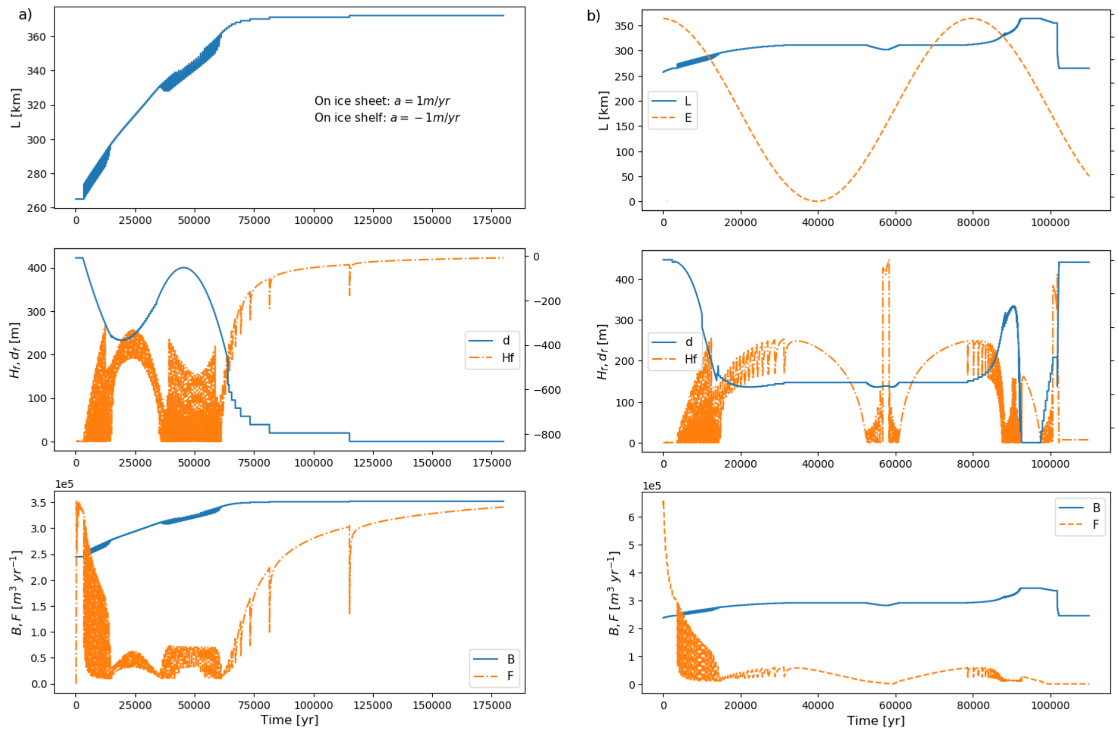

Figure 4: **a)** This figure shows the test with constant accumulation and mass balance on the vertically integrated model. The first panel shows the length of the glacier, the second panel shows height at the glacier front and the water depth and the third panel shows the total mass balance and the flux at the front.
b) This figure shows the test with mass balance depending on altitude in the vertically integrated model. The first panel shows the length of the glacier and the equilibrium line with a forcing period of $P_E = 80\ kyr$, the second panel shows height at the glacier front and the water depth and the third panel shows the total mass balance and the flux at the front.

As seen in figure 4.a the height of the glacier at the front, looks to be very unstable and oscillates up and down. This is due to the way the model advances (see figure 2). Due to this H_f is not as smooth as expected, but jumps up and down as the glacier moves forward. These jumps are only seen when the glacier is on land or grounded on the ocean floor. This is because the glacier grows differently when on land compared to the growth in water.

When the glacier front is floating in the water, the test shows downward spikes in the water depth, this is due to the way the glacier grows in water. it does this by making the shelf thicker, until it is thick enough to move to the next grid point. The dips in water depth happens when the shelf moves forward (see figure 2).

Since the flux though the glacier front F is depending on the height at the front, the flux also jumps up and down as the glacier moves forward. And it can be seen in the figure that these jumps in flux and height happens at the same locations throughout the test.

4.4 Test 2: The vertically integrated model

In the second test the mass balance is depending on the altitude of the glacier. Here the same formula (eq. 28) for the equilibrium line was used. In this case $E_0 = 1000\ m$, $A_E = 2000\ m$, $P_E = 80\ kyr$ and $f = 1.6$ and with a mass balance of 1 $m\ yr^{-1}$ below the equilibrium line and $-1\ m\ yr^{-1}$ above and on the shelf. The same bottom and the same initial profile was used for this test as for the first test.

The results of this test is shown in figure 4.b, were it is seen how the glacier grows with time and makes a jump when it reaches the bump. After the bump the growth slows to a stop and then the ice sheet jumps back past the bump, resembling a length like the initial profile. This process is also cyclic and will begin to repeat it self, when the equilibrium line is at the right height.

5 Data analysis

To test how the vertically integrated model performs for real, a set of data for a flow line for the Helheim glacier is used. The flow line can be seen in figure 5 and figure 6. Helheim is one of Greenland's largest outlet glaciers and as described in Moon et. al. (2018) [12] and Nick et. al. (2009) [6], it has a very large overdeepening in the ocean floor and ends in a very narrow fjord, resulting in a complex behavior of the glacier.

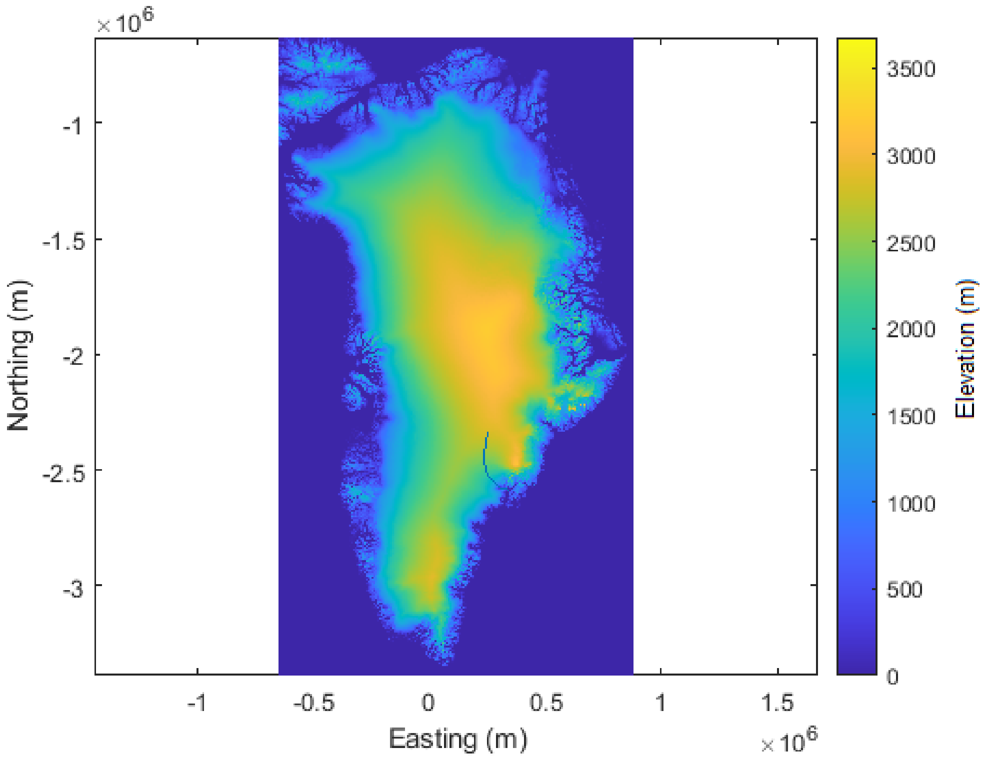

Figure 5: This figure was made by Christine Schøtt Hvidberg for this project and shows the flow line for Helheim on a surface elevation map. The surface elevation data comes from GIMPDEM version 1. by Howat et al. (2014) [10].

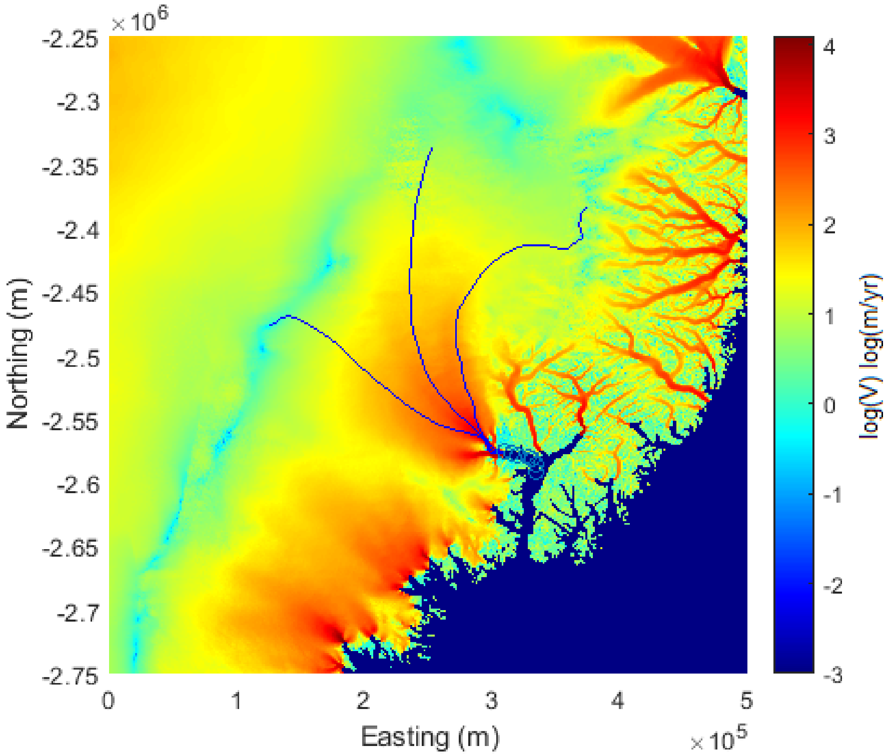

Figure 6: This figure was made by Christine Schøtt Hvidberg for this project and shows the flow line (in the middle) for Helheim on a velocity map. The two other lines represent the width of the glacier. The data for surface velocity comes from the complete map of Greenland ice velocities by Joughin et al. (2017) [9].

The data used in this project comes from four different articles. The bottom topography and the ice sheet thickness data, comes from the Bedmachine v3 article by Morlighem et al. (2017) [8]. The surface velocity data, which is plotted with the flow line in figure 6, comes from the complete map of Greenland ice velocities by Joughin et al. (2017) [9]. The surface elevation data, which is plotted with the flow line in figure 5, comes from GIMPDEM version 1. by Howat et al. (2014) [10]. And lastly the surface mass balance data is from the regional climate model SMB from RACMO2 by Ettema et al. (2009) [11].

In figure 7 the different data for the Helheim flow line has been plotted. The top panel shows the initial surface profile of the flow line and the flow lines bottom topography, which consists of a fairly flat part over sea level. It slopes up toward the ocean an ends in a plateau of about 500 m in height. The plateau ends in a steep slope into the ocean followed by a high bump on the ocean floor. The surface profile is very smooth over land, due to many years of accumulating snow and ice. The profile goes into the water and ends right past the bump in the bed.

The second panel in the figure shows the flow lines surface mass balance and how it changes towards the front. The surface mass balance is low towards the beginning of the flow line, where the ice sheet is thickest and grows towards the coast as the ice sheet thins. When the ice sheet reaches the ocean the surface mass balance falls drastically but has a small peak around the bump in the ocean floor. The last panel in the figure shows the surface velocity of the flow line. The velocity is very small until the ice sheet reaches the coast and becomes thin, here the velocity grows rapidly both down the plateau and over the bump.

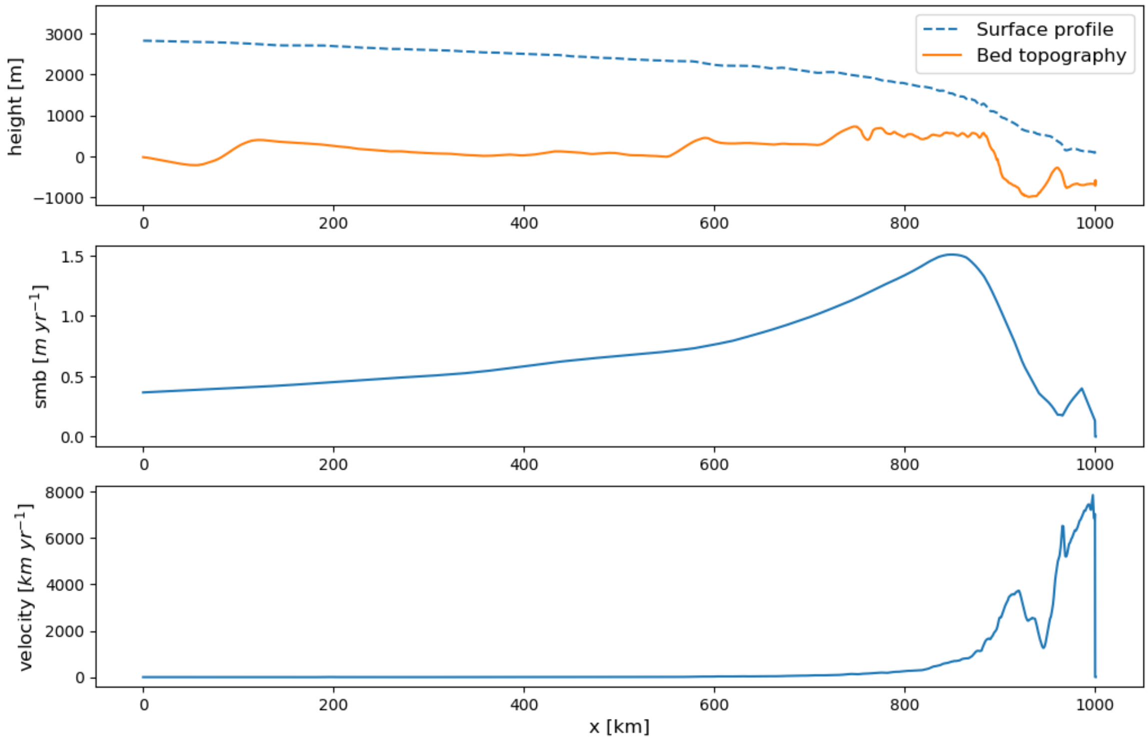

Figure 7: The first panel of this figure shows the surface profile of the flow line (Howat et al. 2014 [10]) and the bottom topography (Morlighem et al. 2017 [8]). The second panel shows the surface mass balance of the flow line (Ettema et al. 2009 [11]) and the third panel shows the surface velocity of the flow line (Joughin et al. 2017 [9]).

5.1 Testing the model

Figure 8 shows the surface profile after running the vertically integrated model for 5000 years. It also shows the surface profiles for four arbitrarily chosen values for the temperature, with a surface mass balance of 1 $m\ yr^{-1}$. The figure only shows about the last 40-90 km of the ice sheet. The complete run for each temperature can be found in appendix A.

The results in the figure shows that the lower the temperature is on the ice sheet, the slower it will retreat. All the temperatures are so high that the ice sheet becomes so thin over the bump, that it retreats past the bump and grounds it self on the opposite side, where it again reach steady state if the conditions do not change. Due to the low accumulation on the ice sheet, none of the temperatures are low enough to reach steady state on the bump. Therefore a series of runs were done with higher accumulation, which can be found in figure 10.

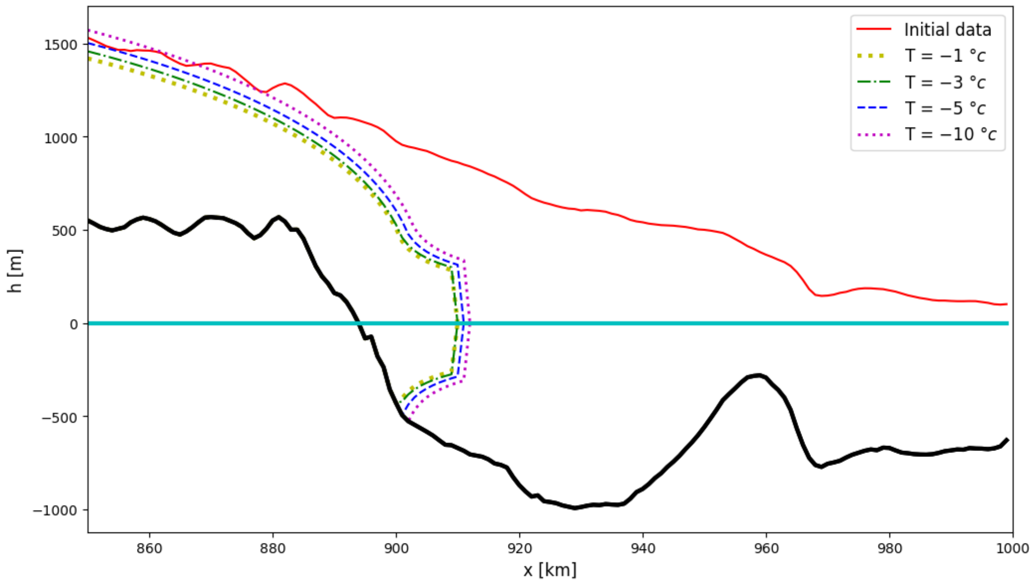

Figure 8: This figure shows the model's results for four different runs with different temperatures. The red line is the observed profile. All runs were 5 kyr and had the flow line's surface profile as initial profile. The black line is the bottom topography and the light green line is the sea level.

Figure 9 shows the surface profile after running the model for 5000 years, with the observed surface mass balance and a temperature of $-3\ °C$. It also shows the surface profiles for five arbitrarily chosen values for the surface mass balance. The five chosen surface mass balances are constant over the entire ice sheet, where as the observed value is not constant. The figure only shows about the last 40-90 km of the ice sheet. The complete run for each surface mass balance can be found in appendix B.
From the results in figure 9 it is clear, that when the surface mass balance is high, the retreat of the ice sheet is slow and when the surface mass balance is low, the retreat is faster. If the surface mass balance is set to be lower than 1 $m\ yr^{-1}$, the ice sheet will be so thin over the bump, that it will retreat past the bump and ground it self on the opposite side. Here it will slowly stop retreating until it reaches steady state. If the surface mass balance is 1 $m\ yr^{-1}$ or higher, the ice sheet will ground it self on the bump, where it will slowly reach steady state if the conditions do not change.

The observed surface mass balance on Helheim is not uniform all over the ice sheet, as that of the other 5 runs. However if the surface mass balance should be approximated to a constant, it is seen in figure 9 that a value of around 0.8 $m\ yr^{-1}$ would be a reasonable approximation.

Figure 10 shows the model's results for 4 different surface mass balances at a temperature of $-10\ °C$. In this figure the accumulation of ice is high enough to let all the profiles ground on the bump, rather than after the bump near the coast. Here we can see that a lower surface mass balance grounds the glacier higher up on the bump than a higher value for surface mass balance. The full runs for the different surface mass balances can be found in appendix C.

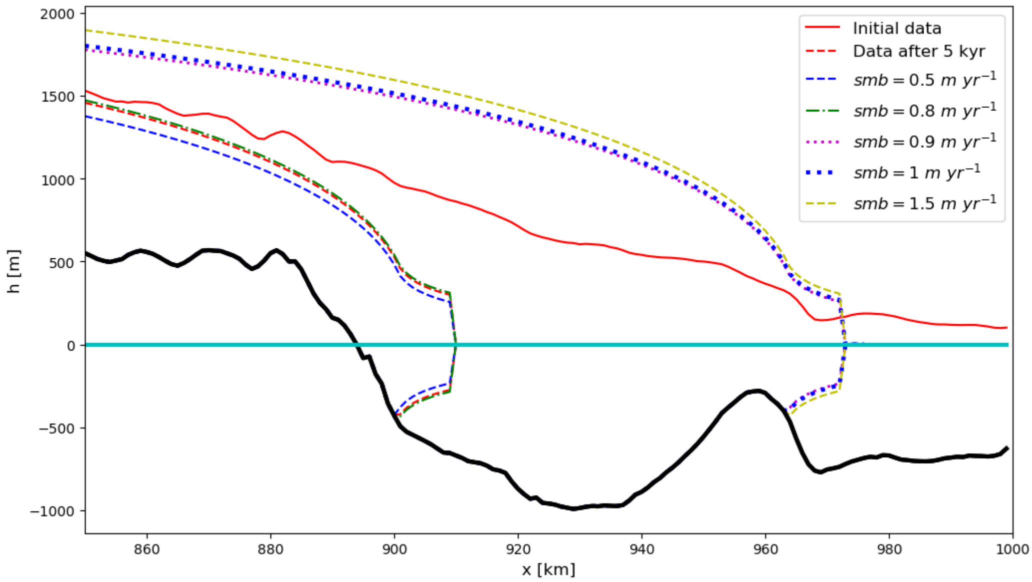

Figure 9: This figure shows the model's results for six different surface mass balances. The red line is the observed and the rest have been chosen arbitrarily. All runs were 5 kyr and had the flow lines surface profile as initial profile and a temperature of $-3\ °C$. The black line is the bottom topography and the light green line is the sea level.

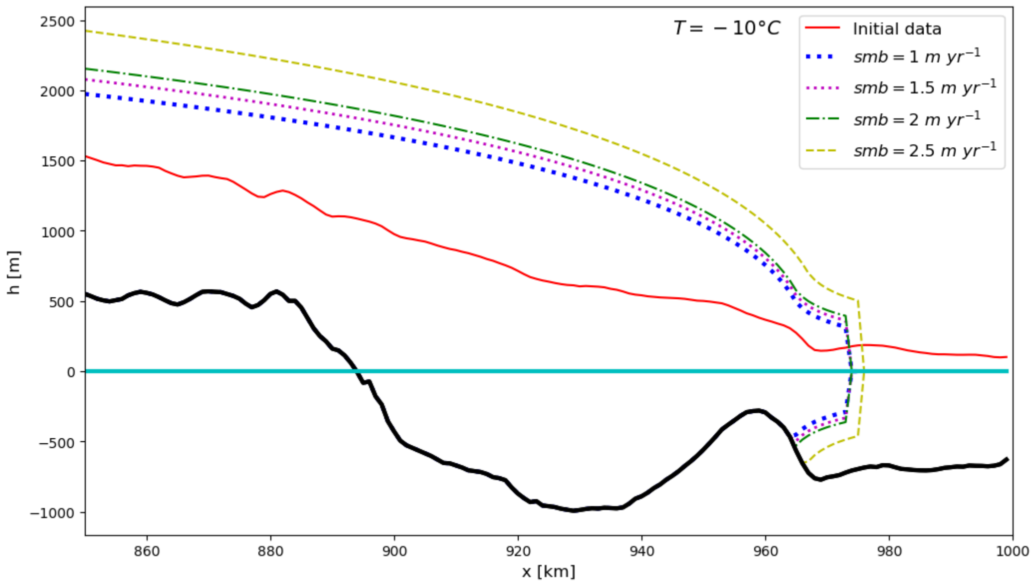

Figure 10: This figure shows the model's results for 4 different surface mass balances (smb) at a temperature of $-10\ °C$. All runs were 5 kyr and had the initial surface profile of the flow line. The black line is the bottom topography and the light green line is the sea level.

Figure 11 shows three runs for three different values for n in Glens flow law (eq. 6). These values have been chosen because the model does not take bottom sliding of the glacier into account. Choosing a value for n that is higher than 3, will make the velocity profile of the glacier more steep and can make up for the lack of bottom sliding. From the results in the figure, it is clear that a higher value for n lets the glacier ground it self on the bump, as well as resulting in a thinner shelf.

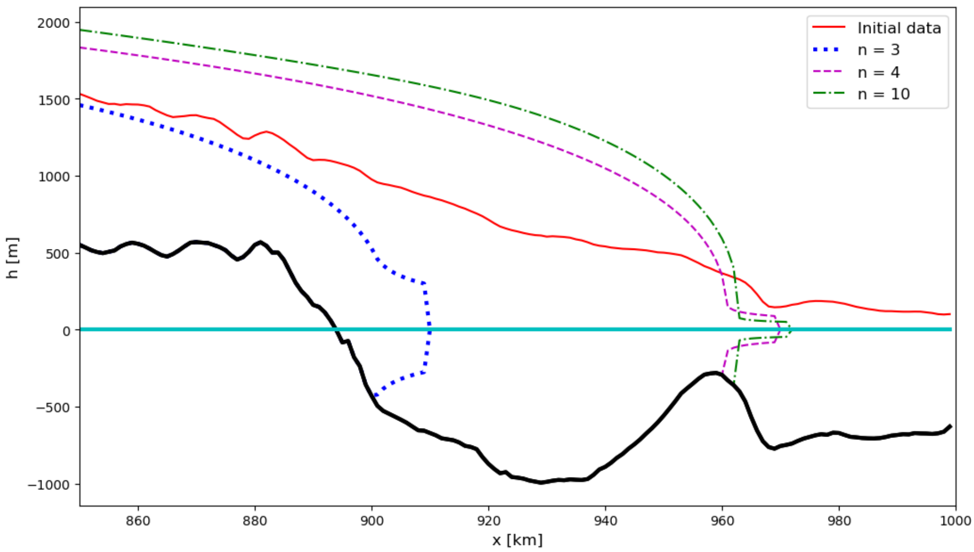

Figure 11: This figure shows the model's results for 3 different values of n. All runs were 5 kyr, had a temperature of $-3\ °C$ and had the flow lines surface profile and observed mass balance as initial profile. The black line is the bottom topography and the light green line is the sea level

Figure 12 shows the model's result for surface velocity of a glacier at $-3\ °C$, with the observed surface mass balance and the observed values for the surface velocity. It also shows the model's calculated velocities under the same conditions after 5 kyr and with two different values of n. It is clear from the observed data that the velocity grows rapidly as it nears the end of the plateau and again as it nears the glacier front, with a distinct peak at the bump. Both the calculated velocities are far lower than the observed values and the velocity profile for n = 4 does not show changes at the end of the plateau. It is clear that the model does not give as high velocity values near the glacier front as expected.

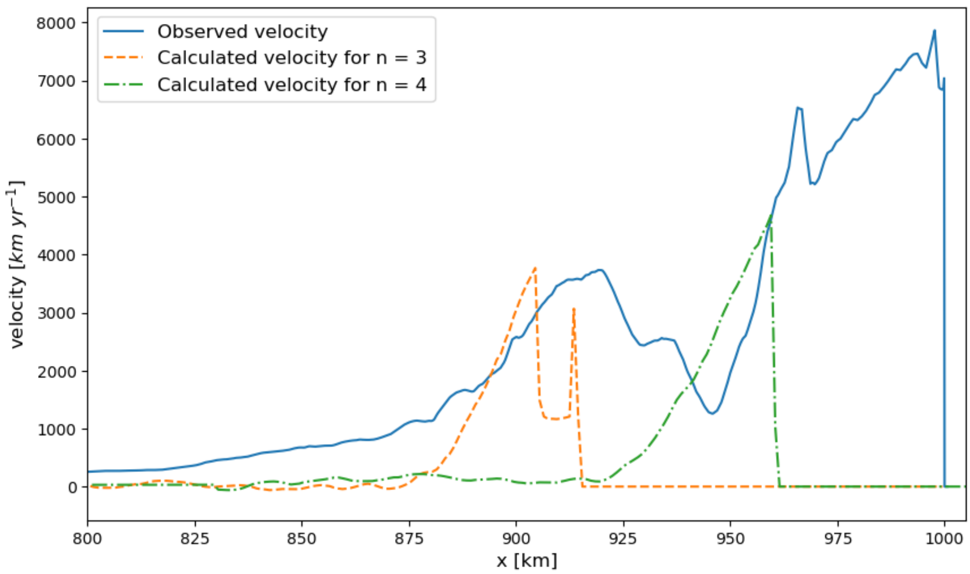

Figure 12: This figure shows the model's result for surface velocity of a glacier at $-3\ °C$, with the observed surface mass balance. In blue is the observed surface velocity.

6 Discussion

The two tests done on the model in section 4 clearly shows, that the minimal model from Oerlemans and Nick (2005) [4] reacts much faster than the vertically integrated model. In figure 3.a we see how the glacier can advance past the bump in the ocean floor and reach steady state within 5000 years. In figure 4 we see how the vertically integrated model needs 175.000 years to do the same.

The reason that the simple model is so much faster than the vertically integrated model, is because it is so simple. It has less parameters that need to be at the right values and it has no shelf to ground before it can advance further into the water. The complexity of the vertically integrated model, although still simple compared to a real glacier, results in more conditions to be just right, before the glacier can advance.

It is clear from the results in figure 8, that vertically integrated model has trouble staying grounded on, or before, the bump in the ocean floor. Lowering the temperature below the -3 to -5 $°C$ that has been used to model Helheim before (This has among others been done by Nick et. al. in their 2009 paper [6]) did nothing to help this problem. This points towards there being a problem with how the model grounds itself, when the glacier ends in deep water.

Figure 9 also shows the same grounding problem in deep water. The results show that when accumulation of ice resembles the observed values, the glacier will retreat rapidly past the bump instead of grounding on or before the bump. This again is not consistent with what is observed at Helheim, where the glacier is grounded further out than the bump.

From the results in figure 10 it is clear, that for the model to behave as expected both the temperature and the mass balance has to be changed drastically. The figure shows that with much higher accumulation of ice and a much lower temperature the model can ground it self on the bump. However this is still much further back than what is actually observed at Helheim and therefore this can not help to explain the problem.

A part of the problem might be due to the lack of bottom sliding in the model. This was one of the physical properties neglected in this model, in order to keep it as simple as possible. Bottom sliding however is an important factor in the dynamics of Helheim and have to be taken in to account when modeling it, since it has a large effect on the thinning of the ice sheet, as described in the 2009 article about Helheim by Nick et. al. [6].

In order to try and make up for the missing sliding a series of runs for higher values of n was done. Having a higher value of $n = 4$ or higher, will make the velocity profile down through the ice sheet steeper and can mimic the effects of bottom sliding. The results for this test can be seen in figure 11 and show that a higher value for n does allow the glacier to ground on the bump. In figure 12 the surface velocities of $n = 3$ and $n = 4$ are plotted. This shows that even with a higher value the model cannot mimic the high surface velocities, which are a result of the bottom sliding.

The velocity profile for $n = 3$ has a noticeable peak where the plateau ends and another one where the glacier ends, but the values are not nearly as high as the observed values. The same goes for the velocity profile for $n = 4$, however this profile does not have a peak at the end of the plateau. This is due to the fact, that the ice sheet is so thick at the end of the plateau, as can be seen in figure 11. This shows that giving n a larger value than 3, does not make up for the thinning of the glacier towards the end of it. There is an offset in the placement of the peaks, because the glaciers profiles ends at different lengths, due to the growth. It can however be seen that all the peaks are placed at the points where the respective glacier profiles end.

The large velocities towards the end of the glacier in the observed profile, points towards a large water pressure towards the bottom, due to the bottom sliding of the glacier. Since there is no bottom sliding in the model and raising the value of n, did not have the desired effect, it is not possible to recreate this high pressure resulting in the high velocities.

Another reason for the models grounding issues can be due to the way the ice shelf works. The shelf is passive and has a constant melting rate of $-1\ m\ yr^{-1}$ all over the shelf. This is however not how an ice shelf actually works. In the 2018 paper by Moon et. al. [12] and the 2019 paper by Shean et al. [13], it is described how the melting rate of the shelf is lowest at the shelf front and highest at the grounding line. This factor could result in the glacier having problems with grounding it self.

The topography surrounding Helheim probably also has a big impact on this problem. As described in Moon et. al. (2018) [12] and Nick et. al. (2009) [6], Helheim ends in a very narrow fjord where the ice spreads to the coast on both sides. This creates a large friction between the glacier sides and the fjord shore, resulting in the glacier retreating much slower than it would otherwise do. Due to this the floating shelf on Helheim is almost none-existing and the glacier is grounded at the very front. This along with the lack of bottom sliding can explain why the model cannot stay grounded far out in the fjord, when that is in fact what Helheim does in real life.

7 Conclusion

The vertically integrated model build for this project, is a very fine simple glacier model and is useful when exploring how the changes in different parameters, can affect the glaciers stability an result in mass gain or lose. The model works well with complicated bottom topographies and surface mass balances and it behaves as expected when advancing and retreating in both water and on land. It gave very reasonable results in the two tests, despite reacting slower than the simple model.

However a number of flaws where found, when it came to recreating the conditions of the Helheim glacier. Here bottom sliding has a big influence on the behavior of the glacier and can therefore not be neglected, as it has been in the model. The bottom sliding at Helheim, results in both thinning of the ice sheet and changes in the velocity profile, which cannot be recreated with this model. Implementing bottom sliding in the model would require many new parameters, as seen in Nick et. al (2009) [6] and would take away from the simplicity of the model.

Another flaw in the model is the passive shelf. The model would retreat less drastically, if the shelf had a non uniform melting rate, resembling the actual melting rate of a shelf. The retreat of the model, would also be slowed down if the model included the friction between the glacier sides and the surrounding topography.

A way to move forward with this model in the future, would be to find a way to include bottom sliding in the model, without compromising the simplicity of the model too much. This would make the model more versatile and it would therefore be more useful, when modelling many different glaciers.

Another way to continue the development of this model, would be to expand it, so that it could cover more of or all of, the glaciers on Greenland all at once. This would further the benefits of this model and make it more useful when looking at the global mass lose and how the Greenland glaciers affect it.

Acknowledgements

Firstly I would like to thank my supervisor Christine Schøtt Hvidberg, for all the help and guidance she has provided through out this complex journey into the physics of dynamical glacier modelling. I would also like to thank her for providing me with the necessary data and material needed to make this project.

I would also like to thank Kasper Holst Lund for helping me debug my code, when I was too tired or frustrated with the project and for listing to all my complaining when the model wasn't working.

References

[1] Meredith, M., M. Sommerkorn, S. Cassotta, C. Derksen, A. Ekaykin, A. Hollowed, G. Kofinas, A. Mackintosh, J. Melbourne-Thomas, M.M.C. Muelbert, G. Ottersen, H. Pritchard, and E.A.G. Schuur, 2019: Polar Regions. In: *IPCC Special Report on the Ocean and Cryosphere in a Changing Climate* [H.-O. Pörtner, D.C. Roberts, V. Masson-Delmotte, P. Zhai, M. Tignor, E. Poloczanska, K. Mintenbeck, A. Alegría, M. Nicolai, A. Okem, J. Petzold, B. Rama, N.M. Weyer (eds.)].

[2] PCC, 2013: Climate Change 2013: *The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovern-mental Panel on Climate Change* [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (eds.)]. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 1535 pp.

[3] Shepherd, A., Ivins, E., Rignot, E. et al. (2020). *Mass balance of the Greenland Ice Sheet from 1992 to 2018.* Nature 579, 233–239. doi: 10.1038/s41586-019-1855-2

[4] Oerlemans, J., & Nick, F. (2005). *A minimal model of a tidewater glacier.* Annals of Glaciology, 42, 1-6. doi:10.3189/172756405781813023

[5] J. Oerlemans (1981). *Some basic experiments with a vertically-integrated ice sheet model.* Tellus, 33:1, 1-11. doi: 10.3402/tellusa.v33i1.10690

[6] Nick, F., Vieli, A., Howat, I. et al. (2009). *Large-scale changes in Greenland outlet glacier dynamics triggered at the terminus.* Nature Geosci 2, 110–114. doi: 10.1038/ngeo394

[7] K.M. Cuffey & W.s.B. Paterson. *The Physics of Glaciers*, fourth edition. Chapter 8: 8.1.1-2 (Balance velocity), 8.2.1 (driving stress), 8.3.1-2 (laminar flow), 8.9.3 (Ice Shelves), 8.10 (surface profiles of ice sheets).

[8] Morlighem M. M., C. N. Williams, E. Rignot, L. An, J. E. Arndt, J. L. Bamber, G. Catania, N. Chauché, J. A. Dowdeswell, B. Dorschel, I. Fenty, K. Hogan, I. Howat, A. Hubbard, M. Jakobsson, T. M. Jordan, K. K. Kjeldsen, R. Millan, L. Mayer, J. Mouginot, B. P. Y. Noël, C. O'Cofaigh, S. Palmer, S. Rysgaard, H. Seroussi, M. J. Siegert, P. Slabon, F. Straneo, M. R. van den Broeke, W. Weinrebe, M. Wood, and K. B. Zinglersen, (2017). *BedMachine v3: Complete bed topography and ocean bathymetry mapping of Greenland from multi-beam echo sounding combined with mass conservation*, Geophys. Res. Lett., 44, doi:10.1002/2017GL074954.

[9] Joughin, I., B. E. Smith, and I. M. Howat. (2017). *A complete map of Greenland ice velocity derived from satellite data collected over 20 years.* Journal of Glaciology, doi: 10.1017/jog.2017.73.

[10] Howat, I., A. Negrete, and B. Smith. (2014). *The Greenland Ice Mapping Project (GIMP) land classification and surface elevation data sets.* The Cryosphere. 8. 1509-1518. doi: 10.5194/tc-8-1509-2014

[11] Ettema J., M.R. van den Broeke, E. van Meigaard, W.J. van de Berg, J.L. Bamber, J.E. Box, and R.C. Bales (2009. *Higher surface mass balance of the Greenland ice sheet revealed by high-resolution climate modeling.* Geophys. Res. Lett., 36, L12501, doi:10.1029/2009GL038110

[12] Moon, T., Sutherland, D.A., Carroll, D. et al. (2018). *Subsurface iceberg melt key to Greenland fjord freshwater budget.* Nature Geosci 11, 49–54. doi: 10.1038/s41561-017-0018-z

[13] Shean, David E. et. al. (2019). *Ice shelf basal melt rates from a high-resolution digital elevation model (DEM) record for Pine Island Glacier, Antarctica.* The Cryosphere 13,10, 2633-2656. doi: 10.5194/tc-13-2633-2019

Appendix A

Appendix A shows the models result for four different temperatures, with the observed surface mass balance. All runs were 5000 years and the glacier profile has been plotted every 1000 years.

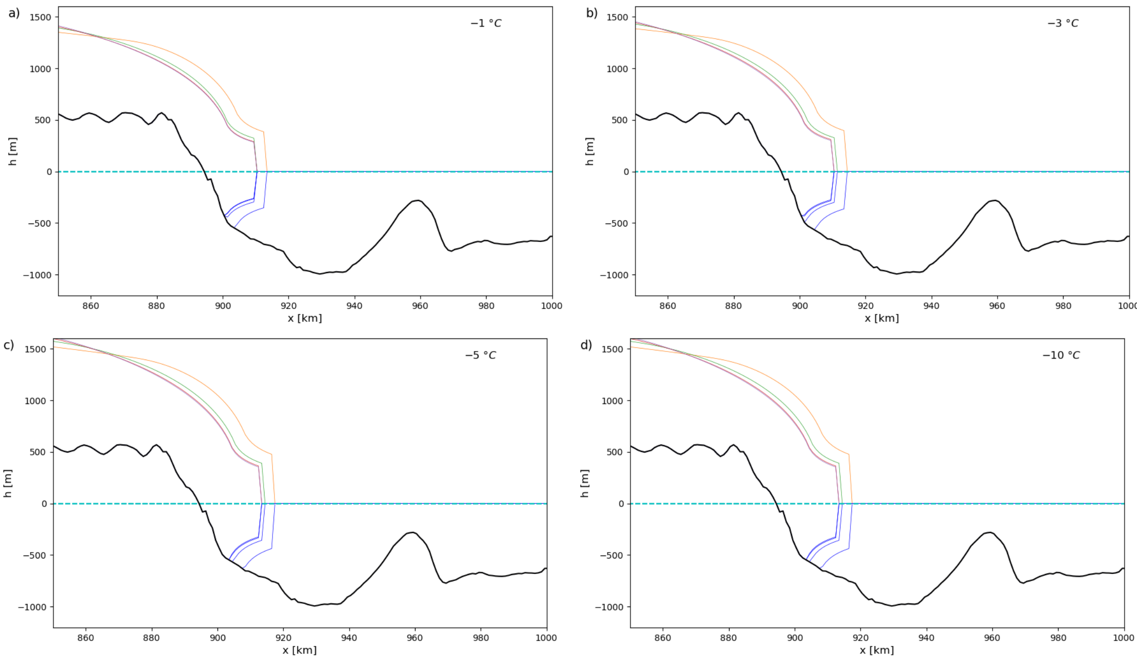

Figure 13: This figure shows the models result for four different temperatures, with the observed surface mass balance. All runs were 5 kyr.

Appendix B

Appendix B shows the models results six different surface mass balances, with ta temperature of -3 $°C$. All runs were 5000 years and the glacier profile has been plotted every 1000 years.

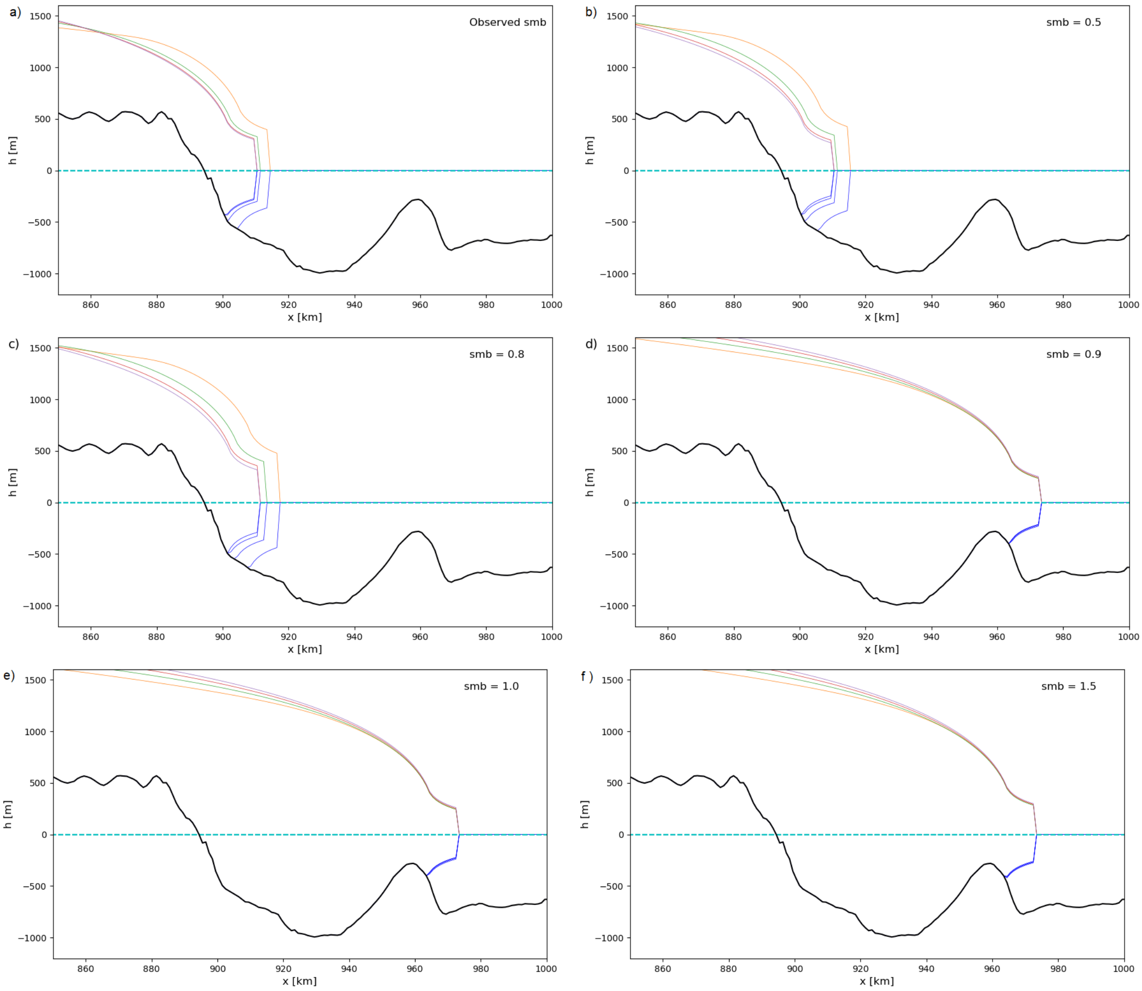

Figure 14: This figure shows the models result for six different surface mass balances, with ta temperature of -3 $°C$. All runs were 5 kyr.

Appendix C

Appendix C shows the models result for 4 different surface mass balances, with ta temperature of $-10\ °C$. All runs were 5000 years and the glacier profile has been plotted every 1000 years.

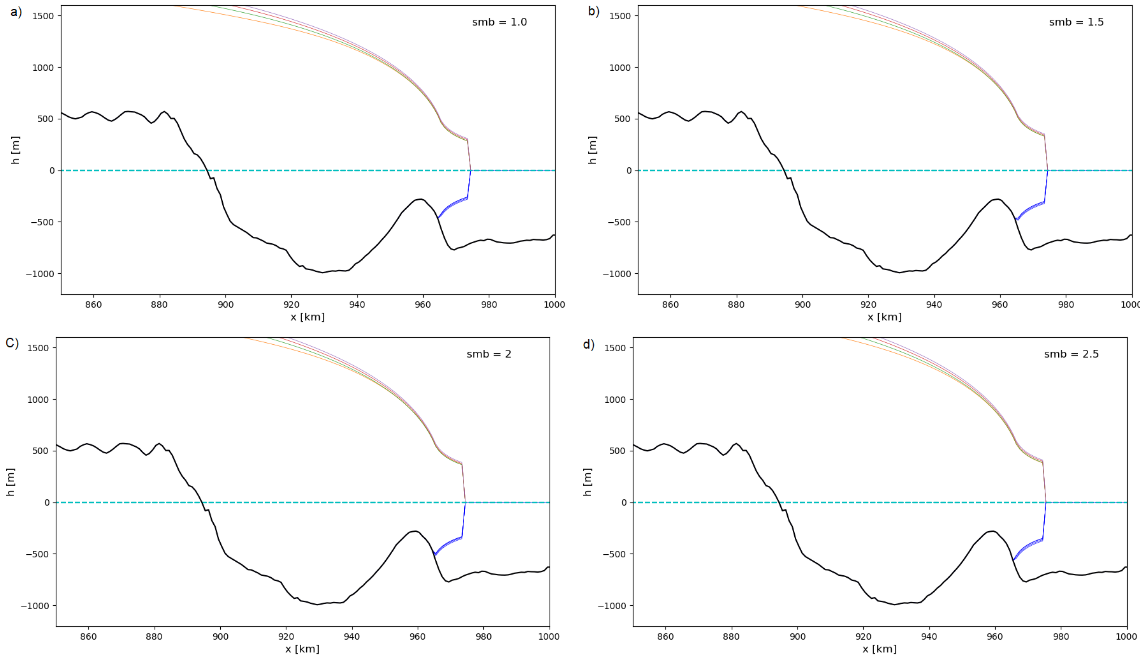

Figure 15: This figure shows the models result for 4 different surface mass balances, with ta temperature of $-10\ °C$. All runs were 5 kyr.

Appendix D

Below is the code for the simple model.

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
```

This simulatin creates a glacier with a uniform mass balance, like the one from Oerlemans og Nick. The mass balance is describe by $B = aL$, this means that the mass balnce is independant of x.

7.0.1 Defining my functions

[3]:
```
def dL(B, F, L, am):
    return (2*(B+F)/(3*am)*L**(-0.5))
```

[4]:
```
def B_func(a, L):
    return a*L
```

[5]:
```
def F_func(c, d, Hf):
    return np.min([0, c*d*Hf])
```

[6]:
```
def Hm_func(am, L):
    return am*L**(0.5)
```

[7]:
```
def Hf_func(af, L, eps, delt, d):
    return np.max([af*L**(0.5), -eps*delt*d])
```

[8]:
```
def d_func(x):
    d_0 = 200       # m
    s = 0.014
    lam = 300       # m
    xs = 40000      # m
    sig = 10000     # m
    return d_0 -s*x + lam * np.exp(-((x-xs)/sig)**2)
```

7.0.2 Defining constants

[9]:
```
N = 5001            # number of timesteps (yr)
am = 2              # constant related to the bulk flow parameter (m^1/
 ↪2)
af = 0.7            # constant related to the bulk flow parameter (m^1/
 ↪2)
delt = 1.127        # ratio of water density to ice density
c = 2.4             # calving rate (yr^-1)
eps = 1
dt = 1              # timestep (yr)
```

7.0.3 Creating arrays

Here I create arrays for all the values that change over time. Currently all elements are 0.

[10]:
```
L = np.zeros(N)
d = np.zeros(N)
Hm = np.zeros(N)
Hf = np.zeros(N)
F = np.zeros(N)
B = np.zeros(N)
a = np.zeros(N)
```

Here I set the starting values for all my arrays. This makes sure that I don't just get 0 for all my time steps.

[11]:
```
a[0] = 0.0005
L[0] = 1
d[0] = L[0]
Hm[0] = Hm_func(am, L[0])
Hf[0] = Hf_func(af, L[0], eps, delt, d[0])
F[0] = F_func(c, d[0], Hf[0])
B[0] = B_func(a[0], L[0])
```

7.0.4 Creating simulation

I create my simulation using a forloop. I loope one less time, than my arrays are long.

I use the the current step of my arrays to create the next L and use that L to create the next step of the rest.

[12]:
```
for i in range(N-1):
    a[i+1] = a[i] + 0.0005
    L[i+1] = L[i] + dt * dL(B[i], F[i], L[i], am)
    d[i+1] = d_func(L[i+1])
    Hm[i+1] = Hm_func(am, L[i+1])
    Hf[i+1] = Hf_func(af, L[i+1], eps, delt, d[i+1])
    F[i+1] = F_func(c, d[i+1], Hf[i+1])
```

```
    B[i+1] = B_func(a[i+1], L[i+1])
```

Here i set my d values for over 0 too 0, so I can multiply by -1 later, and create a time array.

[13]:
```
d[d > 0] = 0
time = np.arange(N)
```

7.0.5 Plotting stuff

[23]:
```
fig, ax = plt.subplots(3, 1, figsize=(6, 10))

ax[0].plot([1000,1001], [0,1], '--', label='a')
ax[0].plot([1000,1001], [0,1], 'w', linewidth=2)
ax[0].plot(time, L/1000, label='L')
ax[0].set_ylabel('L [km]')
ax[0].legend(loc=2)
ax[0].text(-900, 45, 'a)', fontsize=14)

axa = ax[0].twinx()
axa.plot(time, a, '--', label='a')

ax[1].plot(time, Hf, label='$H_f$')
ax[1].plot(time, -d, '--',  label='$d$')
ax[1].set_ylabel('$H_f$, $d_f$ [m]')
ax[1].legend()

ax[2].plot(time, B, label='$B$')
ax[2].plot(time, -F, '--',  label='$-F$')
ax[2].set_ylabel('$B, -F$ [$m^2\ yr^{-1}$]')
ax[2].set_xlabel('Time [yr]', fontsize=12)
ax[2].ticklabel_format(style='sci', axis='y', scilimits=(0,0))
ax[2].legend()

plt.savefig('simple_uniform.png', dpi=100, bbox_inches='tight')
```

Appendix E

Below is the code for the simple model.

[247]:
```
import numpy as np
import matplotlib.pyplot as plt
from sympy import diff
import pandas as pd
import matplotlib.patches as patches
```

[248]:
```
dt = 0.01    # timestep
k = 100    # number of timesteps
time = 0     # current time
kstart = 0  #number of timesteps at start (for restarting a run)
kslut = kstart+k

# sea level varies over time
sl = np.zeros(kslut+1)
```

[249]:
```
nf = 3       # Power in Glens flow law
g = 9.82     # Gravity [m s^-2]
rho = 918    # Ice density [kg m^-3]
rho_w = 1000 # water density [kg m^-3]

# Flow law rate factor A
isotemp = 270
ar = 1.35 * 10**(-5) * np.exp(-60000/8.3143/isotemp)
kfl = 2 * ar * (rho*g)**(nf)/(nf+2)
cs = ar * (rho*g/4*(1 - rho/rho_w))**nf
#kfl = np.zeros(k)
#kfl[0] = 2 * ar * (rho*g)**(nf)/(nf+2)
```

[250]:
```
n = 1001    # Gridpoint
dx = 1000   # Distance in x
```

[251]:
```
# Grid along x
x = np.zeros(n)

for i in range(n):
    x[i] = ((i)*dx)

#[print(y) for y in x]

# staggered grid along x
xs = np.zeros(n-1)
```

```
for i in range(n-1):
    xs[i] = ((i)*dx) + dx/2
```

[252]:
```
def bottom(X):
    d_0 = 200       # m
    s = 0.014
    lam = 1000       # m
    Xs = 350000      # m
    sig = 35000     # m
    bot = (d_0 -s*X + lam * np.exp(-((X-Xs)/sig)**2)) + 3500
    for i in range(len(x)):
        if bot[i] > 200:
            bot[i] = -0.009*X[i] + 2450
        if bot[i] > 300:
            bot[i] = -0.004*X[i] + 1260
        if bot[i] > 500:
            bot[i] = 500
    return bot
bed = bottom(x)          # bottom topography
```

[253]:
```
mb = np.zeros(n)
smb = np.ones(n)    #surface mass balance
ms = -1.0           # surface mass balance shelf
lmax = 10000        # shelf max length
```

[254]:
```
h = np.zeros([n,k])      # ice thickness
s = np.zeros([n,k])      # surface elevation

# H initial
hdiv = ((smb[0]/(2*ar))**(1/nf) * (nf+2)**(1/nf) * 2/(rho*g))**(1/
 ↪(2+2/nf)) * 280000**0.5
h[:101,0] = (1-((x[:101])/100000)**(4/3))**(3/8) * hdiv


xs0 = x[101]
bed0 = bed[101]
b = np.zeros([n,k])
hinit = h
b[:,0] = bed

s[:,0] = h[:,0] + b[:,0]          # surface
left = np.arange(n)*0.08
right = -np.arange(n)*0.08 + 100
w = right-left                    # width
```

```
L = np.zeros(k)
L[0] = 200
```

[255]:
```
lc = 0                  # shelf length at start
lv = 0
imask = np.zeros(n)   # land-water mask

for i in range(n):
    if s[i,0] >= 0:
        imask[i] = 1
    else:
        imask[i] = 0

mb = imask * smb
flux = np.zeros(n-1)
```

[256]:
```
vol_tot = 0        # total volume
masbal_tot = 0     # total mass balance

for i in range(n-1):
    masbal_tot = masbal_tot + 0.5 * (mb[i] + mb[i+1]) * dx
    vol_tot = vol_tot + 0.5 * (h[i,0] + h[i+1,0]) * dx

dvol = 0
vol = vol_tot
```

[274]:
```
difftest = 0
hsg = np.zeros(n)
dsdx = np.zeros(n)
diff = np.zeros(n)
timestep = np.zeros(k)
q = np.zeros([n-1,k])
Hf = np.zeros(k)
L = np.zeros(k)
las_ind = 0
d = np.zeros(k)
Bal = np.zeros(k)
F = np.zeros(k)
F1 = np.zeros(k)

fig, ax = plt.subplots(1, 1, figsize=(10, 6))


for j in range(kstart, kslut-1):
    #kfl[j+1] = 0.5 + kfl[j]
```

```
    for i in range(n-1):
        hsg[i] = (h[i+1,j]+h[i,j])/2
        dsdx[i] = (s[i+1,j]-s[i,j])/dx
        diff[i] = kfl*hsg[i]**5 * dsdx[i]**2
        flux[i] = -diff[i] * dsdx[i]

    difftest = np.max(diff)
    if difftest > 0:
        dt = dx * dx/difftest/6
        dt = np.min([dt,1.0])
        dt = np.max([dt,0.0001])
    else:
        dt = 0.0001

    time = time + dt
    timestep[j] = dt

    c = np.zeros([n,n])

    diffid = kfl/2 * h[0,j]*h[1,j]**4 * ((s[2,j] - s[0,j])/
↪(2*dx))**2 * (1/2*(w[1]+w[0]))
    c[0,0] = -diffid
    c[0,2] = diffid

    for i in range(1,n-1):
        c[i,i-1] = diff[i-1]
        c[i,i] = -diff[i-1]-diff[i]
        c[i,i+1] = diff[i]

    c[n-1,n-1] = 1

    h[:,j+1] = h[:,j] + dt*(mb+1/dx**2*c.dot(s[:,j]))
    h[0,j+1] = h[1,j+1]
    b[:,j+1] = bed[:]

    imask = np.zeros(n)
    for i in range(n):
        if h[i,j+1] <= 0 :
            h[i,j+1] = 0
        else:
            imask[i] = 1

    xs0 = 0
    is0 = 0
    ism = 0
```

```
    for i in range(n-1):
        if is0 == 0:
            imask[i] = 1
            mb[i] = smb[i]

            if bed[i]+sl[j+1]<0 and h[i,j+1]>0 and h[i+1,j+1]<=
↪rho_w/rho*(sl[j+1]-bed[i+1]):
                is0 = i
                xs0 = x[i]
                bed0 = bed[i]
                h0 = h[i,j+1]
                u0 = flux[i-1]/hsg[i-1]
                if ms>= 0:
                    lc = lmax
                else:
                    lc = np.min([h0*u0/abs(ms), lmax])
            ism = i
        else:
            if x[i] < xs0+lc:
                if ms == 0:
                    h[i,j+1] = (h0**(-nf-1)+cs*(nf+1)/h0/
↪u0*(x[i]-xs0))**(-1/(nf+1))
                else:
                    h[i,j+1] = ((u0/
↪(ms*(x[i]-xs0)+h0*u0))**(nf+1)*(1-cs/ms*h0**(nf+1))+cs/ms)**(-1/
↪(nf+1))

                if h[i,j+1] >= rho_w/rho*(sl[j+1]-bed[i]):
                    imask[i] = 1
                    b[i,j+1] = bed[i]
                else:
                    b[i,j+1] = -h[i,j+1]*rho/rho_w + sl[j+1]
                    imask[i] = 1
                ism = i
                mb[i] = ms
            else:
                h[i,j+1] = 0
                b[i,j+1] = sl[j+1]
                imask[i] = 0
                mb[i] = 0
    s[:,j+1] = b[:,j+1] + h[:,j+1]
    q[:,j+1] = flux[:]

    H = pd.Series(h[:,j+1])
```

```
    las_ind = H[H != 0].index[-1]
    Hf[j+1] = h[las_ind,j+1]
    L[j+1] = las_ind*1000
    Bal[j+1] = smb[0]*(L[j+1]-10000)+(-1*10000)
    F[j+1] = flux[las_ind-10]
    F1[j+1] = flux[las_ind-1]
    if bed[las_ind] < 0:
        d[j+1] = -bed[las_ind]
    else:
        d[j+1] = 0

    if j % 20 == 0:
        ax.plot(x[:]/1000, s[:,j+1], linewidth=0.5)
        ax.plot(x[:]/1000, b[:,j+1], 'b', linewidth=0.5)
        ax.plot(x/1000, 0*b[:,j+1]+sl[j+1], '--c')
            #if xs0 > 0:
                #ax[0].plot(xs0,bed0, '.r')
            #ax[1].plot(xs/1000, q[:,j+1])
    #if j == 0:
        #ax.plot(x[:]/1000, s[:,j+1], 'w', linewidth=20)

    mb = imask * mb
    s[:,j+1] = s[:,j+1] * imask[:]
    h[:,j+1] = h[:,j+1] * imask[:]
    b[:,j+1] = bed[:]
ax.plot(x/1000, bed, 'k')
ax.set_xlim([98.8, 101.2])
ax.set_ylim([450, 1250])
#ax[1].set_xlim([0, 400])
ax.text(98.55, 1210, 'a)', fontsize=14)

ax.set_xlabel('x [km]', fontsize=12)
ax.set_ylabel('h [m]', fontsize=12)
#ax[1].set_ylabel('flux [£m^{3} yr^{-1}£]')
plt.savefig('front.png', dpi=100, bbox_inches='tight')
```