Stockholms universitet

Investigating the Free-Surface Stabilization Algorithm in Glacier Simulation: Addressing Minimum Thickness Constraints

Undersöka Free-Surface Stabilization-algoritmen i Glaciär-Simulering: Adressera Minsta Tjockleksrestriktioner

Oskar Hogman

Handledare: Josefin Ahlkrona, Clara Henry
Examinator: Anders Mörtberg
Inlämningsdatum: 2024-05-20

Kandidatuppsats i datalogi
Bachelor Thesis in Computer Science

Abstract

Global warming is causing melting of glaciers and ice sheets globally, contributing to rising sea levels. In order to estimate future rise of sea levels, glaciers and ice sheets are simulated using numerical methods. Glaciers and ice sheets are described using the Stokes equations and the free-surface equation. These equations are solved approximately computationally using the finite element method. However, solving the Stokes equations is prone to numerical instability, demanding a small time-step. In this thesis, a free-surface stabilization algorithm (FSSA) is investigated with regard to a minimum thickness condition, which needs to be imposed for numerical stability. We show that time-stepping with FSSA is stable for a time-step where the same method without the FSSA suffer from instability. We also show that initially it gives the wrong velocities at points where the minimum thickness condition is active. The stability granted by the free-surface stabilization algorithm is in line with previous research. The differing velocities at points for the minimum thickness condition confirms what was expected, as the free-surface stabilization algorithm doesn't take the minimum thickness condition into account. These results exhibits the free-surface stabilization algorithm as a possibly viable method to increase the speed of simulations, as it allows for larger time-steps. The FSSA does come with a cost of less accuracy at the limit of a glacier or ice sheet, where the minimum thickness condition is applied. The free-surface stabilization algorithm shows promise in improving on numerical methods, but the issue with the minimum thickness condition needs to be addressed.

Sammanfattning

Den globala uppvärmningen leder till att glaciärer och inlandsisar smälter globalt, vilket bidrar till stigande havsnivåer. För att kunna uppskatta framtida havsnivåhöjningar simuleras glaciärer och istäcken med numeriska metoder. Glaciärer och inlandsisar beskrivs med hjälp av Stokes ekvationer och ekvationen för fria ytor. Dessa ekvationer löses approximativt beräkningsmässigt med hjälp av finita elementmetoden. Lösningen av Stokes ekvationer är dock känslig för numerisk instabilitet och kräver ett litet tidssteg. I denna avhandling undersöks free-surface stabilization algorithm (FSSA), en stabiliseringsalgoritm, med avseende på ett villkor om minsta tjocklek, som måste införas för numerisk stabilitet. Vi visar att tidsstegning med FSSA är stabil för ett tidssteg där samma metod utan FSSA drabbas av instabilitet. Vi visar också att den initialt ger fel hastigheter i punkter där villkoret för minsta tjocklek är aktivt. Den stabilitet som stabiliseringsalgoritmen med fri yta ger är i linje med tidigare forskning. De olika hastigheterna i punkter för villkoret om minsta tjocklek bekräftar det som förväntades, eftersom stabiliseringsalgoritmen med fri yta inte tar hänsyn till villkoret om minsta tjocklek. Detta visar att algoritmen för stabilisering av fria ytor är en möjlig metod för att öka simuleringarnas hastighet, eftersom den möjliggör större tidssteg. FSSA medför dock en kostnad i form av mindre noggrannhet vid gränsen för en glaciär eller ett istäcke, där villkoret om minsta tjocklek tillämpas. FSSA är lovande när det gäller att förbättra numeriska metoder, men problemet med villkoret för minsta tjocklek måste lösas.

Acknowledgements

First and foremost a huge thank you to my supervisor Clara Henry, for tirelessly week after week helping, assisting and guiding me with this thesis. This thesis would not be what it is without your guidance. Thank you also to Josefin Ahlkrona, for introducing me and awakening my interest to this topic, and guiding me along the way of writing this thesis. A thank you to André Löfgren as well, for helping me get started with Elemer/Ice as well as allotting time to discuss and give insightful comments of the results.

I would also like to give thanks to my fellow co-workers and students, whom have given good comments and helped me along the way. None mentioned, none forgotten.

Lastly I'd like to thank Lars Arvestad, whose advice always have been and continue to be helpful during my journey as a student.

Contents

1 Introduction

The Earth's glaciers and ice sheets are diminishing in size, contributing to rising sea levels. Over the next decades, mass loss is certain for the Greenland Ice Sheet and likely for the Antarctic Ice Sheet even if the global temperature stabilizes [1]. In order to prepare for rising sea levels, we need to know by how much sea levels are expected to rise. Thus, we need to accurately predict changes in glacier and ice sheet mass in the future, accounting for different scenarios. In order to make predictions, Ice sheets and glaciers are described as highly viscous, non-Newtonian fluids and are modelled using partial differential equations, solved using numerical methods. The non-Newtonian nature of ice makes solving its flow computationally expensive and so the continuous development of numerical methods to correctly and efficiently simulate the change of ice sheet and glacier mass in the future is crucial.

In a previous study [2] the free-surface stabilization algorithm (FSSA) was implemented to improve computational efficiency when simulating ice sheets and glaciers. The simulations of ice sheet and glacier melting is governed by two set of equations; (1) the Stokes equations and (2) the free surface equation. These are solved discretely by using the finite element method [3], [4]. Solving these systems of partial differential equations approximately previously demanded a strict time-step constraint to ensure stability, but by using FSSA the authors were able to increase time-steps by up to thirty times with the same accuracy as previous methods and without causing instability. The FSSA does this by approximating the updated surface in each time-step for solving the Stokes equations. The FSSA method used in the article thus shows promise in improving on current ice sheet solvers.

A retreating glacier needs a minimum ice thickness condition, i.e. a lower bound for thickness of the ice, as the numerical method used to model the glacier can not produce a solution with zero thickness. In this thesis, we explore the consequences of the minimum thickness condition when applying FSSA. The equations cannot be solved with a zero or negative thickness, so a minimum thickness constraint must be imposed. The FSSA, however, does not take this minimum thickness constraint into consideration when approximating the new surface in the Stokes equations. This causes an issue with calculating the velocities of the moving ice, potentially resulting in a difference of the surface evolution of a glacier under simulations using the FSSA. In order to investigate if this is the case, a simulation with the FSSA implemented is compared with two simulations without FSSA: a reference solution using a small time-step and a simulation using the same time-step as the simulations with FSSA. The comparison is made in surface evolution and the velocities of the ice. A comparison of the development of the error for each time-step is also made, particularly in the vincinitiy of the minimum ice thickness condition. The glacier used in this thesis is a two-dimensional synthetic glacier with a randomized bedrock topography, created using Perlin noise [5], resulting in a shape that replicates a real-world glacier.

The thesis is structured as follows. In Chapter 2, we introduce the terminology and methods used in this thesis, as well as the governing equations of a glacier. In Chapter 3, The numerical methods for ice-flow modelling, the ice-flow equations are solved. Furthermore, the implementation of the FSSA and the algorithm for the minimum thickness condition are explained. The numeri-

cal experiments to be carried out are also introduced in Chapter 3. The results of the experiments are presented in Chapter 4. It is followed by a discussion of the results in Chapter 5, and the thesis is concluded in Chapter 6.

2 Preliminaries

This chapter starts with some basic concepts for numerical methods in Section 2.1 The chapter continues with a short introduction to partial differential equations, the finite element method and Newton's method in regard to its relevance to ice sheet modeling in Section 2.2. The governing equations of ice sheet modeling are introduced in Section 2.3.

2.1 Basic Concepts for Numerical Methods

To start, some initial definitions and theorems which will be needed later are defined. Both the Frobenius Inner Product, Green's idendity and function spaces are needed later to process the equations used in this thesis. Some basic knowledge of Euler methods and stability in numerical methods is also essential.

2.1.1 The Frobenius Inner Product

Definition 2.1. Let A, B be $n \times n$ matrices. The *Frobenius inner product*, denoted :, is defined as

$$A : B = \sum_{i=0}^{n} \sum_{j=0}^{n} A_{ij} B_{ij}, \tag{2.1}$$

where i is the row and j the column of the matrix.

2.1.2 Green's Identity

Theorem 2.2. Let $\mathbf{u}$ be a vector field and v a scalar field over a domain Ω. Then

$$\int_{\Omega} -(\Delta \cdot \mathbf{u})\, v d\mathbf{x} = \int_{\Omega} \nabla \cdot \mathbf{u} \cdot \nabla v d\mathbf{x} - \int_{\partial\Omega} (\mathbf{u} \cdot \hat{\mathbf{n}})\, v ds, \tag{2.2}$$

where $\hat{\mathbf{n}}$ is the unit normal pointing outward from the boundary, $\int_{\partial\Omega} ds$ is the curve integral on the boundary, and ∇ and Δ are the first order and second order divergences, respectively, introduced in Definitions 2.14 and 2.15. A derivation of the theorem can be found in [6].

2.1.3 Spaces

Next, Hilbert function spaces and Sobolev function spaces are introduced. For more information on function spaces, one can read [6], Chapter 7.

Definition 2.3. A *Hilbert function space* is defined as

$$L^2(\Omega) = \{v : \Omega \to \mathbb{R} : ||v||_{L^2}(\Omega) < \infty\}. \tag{2.3}$$

Here, $|| \cdot ||_{L^2}(\Omega)$ refers to the L^2-norm. Thus, this is the function space of all functions with a bounded L^2-norm on the space Ω. Conventionally, the L^2-norm is used since among its properties both the triangle inequality as well as the Cauchy-Schwartz inequality holds. These are necessary inequalities to calculate error margin for the approximations used in numerical methods.

The Hilbert function space is not enough for derivatives to make sense as derivates are not included in the definition. Thus, Sobolev function spaces are also introduced.

Definition 2.4. A *Sobolev function space* is defined as

$$H^1(\Omega) = \{v \in L^2(\Omega) : ||v||_{L^2}(\Omega) + ||\nabla v||_{L^2}(\Omega) < \infty\}. \tag{2.4}$$

This function space refers to all functions with a bounded gradient in L^2-norm as well.

Onward, using the terms '*"Hilbert space"* or *"Sobolev space"* will refer to these definitions.

2.1.4 Euler Methods

Definitions of the forward Euler and the backward Euler methods will be introduced here. The forward Euler method is also referred to as the explicit Euler method, and in the same way the backward Euler method is also called the implicit Euler method. These are used to approximately solve differential equations (see 2.2.1). In this thesis, Euler methods are used for time-stepping as well as deriving the free-surface stabilization algorithm.

Definition 2.5. The *forward Euler* method, for a function y at a certain time-step k is defined as [7]

$$y_{k+1} = y_k + \Delta t f(t_k, y_k). \tag{2.5}$$

The value Δt is the size of the time-step used and $f(t, y(t)) = y'(t)$. This is an initial value problem, i.e. the values for t_0 and $y(t_0)$ are needed to take the first step.

Definition 2.6. The *backward Euler* method, for a function y at a certain time-step k is defined as [7]

$$y_{k+1} = y_k + \Delta t f(t_{k+1}, y_{k+1}). \tag{2.6}$$

The backward Euler method is a function of y_{k+1} on both sides, so an algebraic equation for y_{k+1} needs to be solved.

2.1.5 Stability

In the last section the Euler methods were defined. These are algorithms to solve differential equations, and as they approximate a solution for each time-step, there'll be an error compared to the true solution. One might ask the question: does the error grow with each step taken? This introduces the concept of stability. To explain this concept, two definitions are borrowed from [8]:

Definition 2.7. The *region of absolute stability* R for a numerical method for solving a differential equation is defined by

$$R := \{\mu : |\lambda_i(\mu)| < 1, \forall \lambda_i(\mu)\} \tag{2.7}$$

where $\lambda_i(\mu) = ah$, for the time-step size h, are the roots of the characteristic equation when the method is applied to the test equation $y' = ay, a \in \mathbb{C}$.

Here, the characteristic equation refers to the one for a differential equation. Taking the forward Euler method as example:

$$y_{n+1} = y_n + hf(t_n, y_n) = y_n + hay_n = (1 + ha)y_n, \tag{2.8}$$

the characteristic equation becomes

$$\lambda = (1 + ha) = (1 + \mu). \tag{2.9}$$

Definition 2.8. A method is called *A-stable* if its region of absolute stability R contains the left half of the complex plane $\{z \in \mathbb{C} : \mathrm{Re}(z) < 0\}$.

A-stability of a method is desirable, because if a true solution decays to zero so does the approximative numerical solution. Rather than A-stable, the term stable will be used onward. A method is unstable, affected by instability, if it is not stable.

Returning to the Euler methods, what does this mean practically? If the numerical method does not fulfill the requirement for the region of absolute stability, the approximation will grow exponentially with each time-step taken. Thus, eventually the approximation will diverge from the true solution. A method may be close to one, but as long as its absolute value according to Definition 2.7 is below one, it will closely approximate the true solution. The error will still grow with each time-step even if the method is stable, due to a small error in the approximation in each time-step, but not as much as with instability.

2.2 Solving Partial Differential Equations with The Finite Element Method

In this section, partial differential equations are defined in Section 2.2.1. The finite element method is is explained in Section 2.2.2. Solving nonlinear partial differential equations need some iterative method to reach convergence, introduced in Section 2.2.3.

2.2.1 Partial Differential Equations

In this section, partial differential equations are introduced. Some important concepts concerning partial differential equations are also defined. An introduction to differential equations can be found in [9], and for more information on partial differential equations, please see [10].

Definition 2.9. A *differential equation* is an equation consisting of the derivative of an unknown function, and possibly the function itself.

Example 1. Let

$$y' + y = f \tag{2.10}$$

be an equation for some arbitrary function f, where y is a function and y' its derivative. Then Equation 2.10 is a differential equation.

Definition 2.10. A *partial differential equation* is an equation in which the function depends on several variables.

Example 2. In Equation 2.10, let $f = f(x, y)$ and $y = g(x, y)$. Then this equation is a partial differential equation.

Partial differential equations are used to describe various different natural phenomena. These are usually solved approximately by using numerical methods, as they often can not be solved analytically. Partial differential equations are classified depending on their properties.

Definition 2.11. The *dimension* of a partial differential equation is equal to the space variables of the function.

Definition 2.12. The *order* of a partial differential equation is equal to the highest order of its derivatives.

Example 3. The differential equation introduced in Example 1 with the properties given in Example 2 has a dimension of two, since there are two space variables: x and y. It is a first order partial differential equation since the highest derivative in the equation is the first derivative.

Definition 2.13. The *partial derivative* of a function $f(x, y)$ is written either f_x or f_y if differentiating with respect to x or y. The second order partial derivative is written f_{xx}, f_{xy}, f_{yx} or f_{yy}.

Definition 2.14. The symbol $\nabla\cdot$ denotes the *divergence*, i.e. the sum of all partial derivatives of a function.

Definition 2.15. The symbol Δ denotes the *Laplacian* (the second order divergence), i.e. the sum of all second order partial derivatives of each individual variable of a function.

Example 4. Let $f(x, y, z)$ be a function. Then

$$\Delta f - \nabla \cdot f = (f_{xx} + f_{yy} + f_{zz}) - (f_x + f_y + f_z) \tag{2.11}$$

according to Definitions 2.14 and 2.15.

Definition 2.16. A *boundary condition* of a partial differential equation, is a constraint of how the partial differential equation behaves at the boundary of the domain the partial differential equation is defined on.

Definition 2.17. A *Dirichlet boundary condition* is one, where a partial differential equation's behaviour on the boundary of a domain can be described by a function g.

Example 5. Let u be a function over the domain Ω, and $\partial\Omega$ be the boundary of the domain. Let

$$-\Delta u = f \qquad \text{in } \Omega, \tag{2.12}$$

$$u_{|\partial\Omega} = 0 \qquad \text{on } \partial\Omega, \tag{2.13}$$

then Equation 2.12 is partial differential equation, whereas Equation 2.13 describes a Dirichlet boundary condition. The partial differential equation defined in 5 is called the *Poisson's* equation. Poisson's equation arises in various types of physical situations.

The notation of $\partial\Omega$ to specify the boundary of a domain Ω will be used henceforth unless otherwise stated. The equation shown in Example 5 is a specific kind of partial differential equation, given a specific name.

A partial differential equation can be classified depending on its algebraic properties. Differentiation will be done into two categories.

Definition 2.18. A *linear* partial differential equation is linear in the unknown functions.

Definition 2.19. A *nonlinear* partial differential equation is nonlinear in the unknown functions.

Nonlinear partial differential equations can be further subdivided into categories depending on which order of derivatives are nonlinear, but the stated definitions are sufficient for this thesis.

2.2.2 The Finite Element Method

Partial differential equations can be solved using the finite element method. Here, the time-independent case of a domain in two dimensions of the finite element method is explained. For a more thorough guide to the finite element method, see [6].

Definition 2.20. A *mesh* of a domain is a mapping or a grid of the domain.

Definition 2.21. A *structured* mesh has regular connectivity. An *unstructured* mesh instead has irregular connectivity.

See Figure 2.1 for examples of a structured and an unstructured mesh.

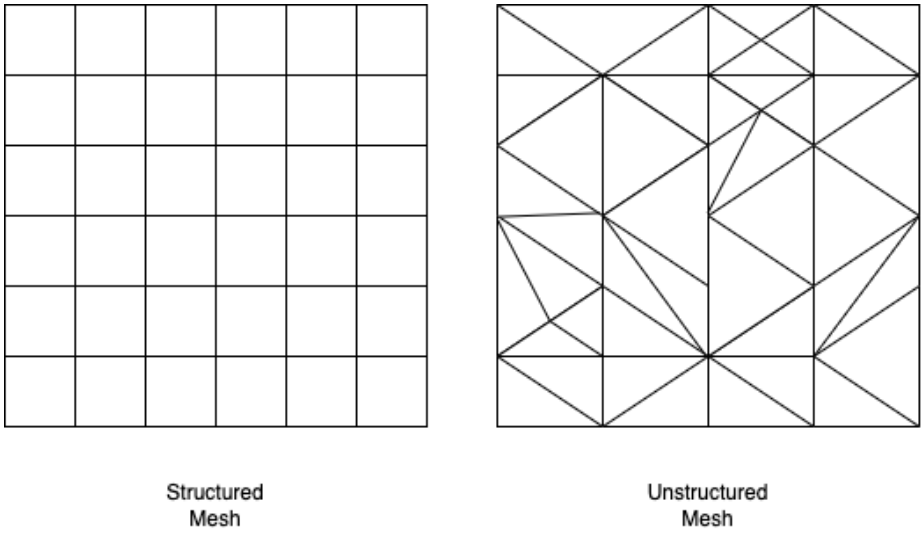

Figure 2.1: Example of a structured mesh and an unstructured mesh.

Definition 2.22. A *finite element* is a specified geometric shape making up the mesh.

A structured mesh can be made out of quadrilaterals while an unstructured mesh can be made out of triangles as finite elements in the plane. Structured meshes have space efficiency and can thus have higher resolutions than unstructured meshes. Unstructured meshes have the advantage of being able to model much more complex domains, as the triangles can be of any size or shape as long as all shapes are triangles (see Figure 2.2).

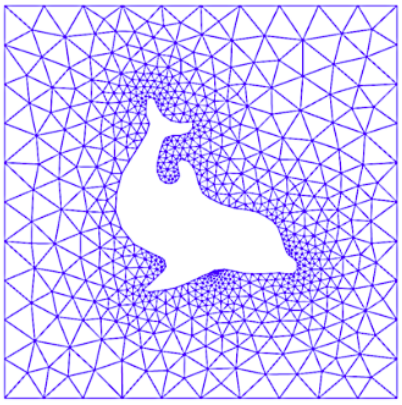

Figure 2.2: An unstructured mesh where the boundary is in the shape of a dolphin. The figure is taken from [11].

Definition 2.23. *Nodes* are the corners of a finite element and an *edge* is made up of the sides of a finite element. The nodes are named N_j, for $j = 1, 2, 3, .., M$ where M is the number of nodes.

Nodes can only touch other corners of other finite elements, never an edge. See Figure 2.3 of a configuration of a small mesh of eight finite elements in the form of triangles, nine nodes and sixteen edges.

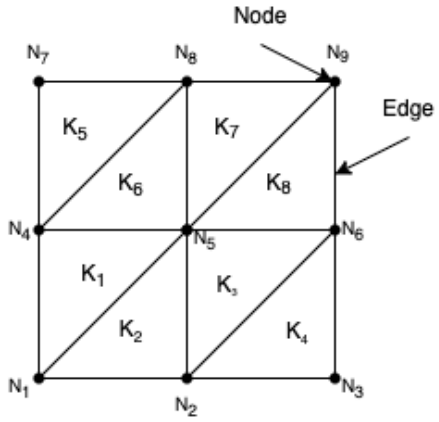

Figure 2.3: A small mesh.

A definition of continuous piecewise polynomial spaces is needed next.

Definition 2.24. Let K be a finite element. A *continuous piecewise polynomial space* is can be defined by

$$P(K) = \{v : v(x) = c_0 + c_1 x_1 + c_2 x_2, x_i \in K \text{ for } i \in \{1, 2\}, c_0, c_1, c_2 \in \mathbb{R}\}. \tag{2.14}$$

Here, a linear continuous piecewise polynomial space is used, but higher order polynomials are possible. The nodal values are used as degrees of freedom, so the natural basis $\{1, x_1, x_2\}$ is not suitable. Instead, a nodal basis is used.

Definition 2.25. The *nodal basis* $\{\gamma_1, \gamma_2, \gamma_3\}$ is defined by

$$\gamma_j(N_i) = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}, i, j = 1, 2, 3. \tag{2.15}$$

With this definition of the nodal basis, any function $v \in P(K)$ can be expressed as

$$v = \alpha_1 \gamma_1 + \alpha_2 \gamma_2 + \alpha_3 \gamma_3 \tag{2.16}$$

where $\alpha_i = v(N_i)$. The basis functions γ_j are called hat functions, after their similarity to pointy hats when drawn on a mesh.

Once a mesh of the domain has been created, the partial differential equation being solved needs to be changed into their variational form. This is the discretization of the partial differential function. To demonstrate the discretization method, Poisson's equation from Example 5 will be used, i.e.

$$-\Delta u = f \qquad \text{in } \Omega, \tag{2.17}$$

$$u_{|\partial\Omega} = 0 \qquad \text{on } \partial\Omega, \tag{2.18}$$

The first step of this is to introduce a test function $v \in V_0$, where

$$V = \{v : ||v||_{L^2(\Omega)} + ||\nabla v||_{L^2(\Omega)} < \infty\}, \tag{2.19}$$

$$V_0 = \{v \in V : v|_{\partial\Omega} = 0\}. \tag{2.20}$$

i.e. a Sobolev space with the boundary condition that the test function v is zero at the boundary. This test function is multiplied with Equation 2.17 to aquire

$$-\Delta u v = f v. \tag{2.21}$$

Integrating over the domain Ω and, replacing the second order divergence, we use Green's identity to get

$$\int_\Omega \nabla u \cdot \nabla v dx - \int_{\partial\Omega} \mathbf{n} \cdot \nabla u v ds = \int_\Omega f v dx \tag{2.22}$$

If the integral over the boundary is solved it becomes

$$-\int_{\partial\Omega} \mathbf{n} \cdot \nabla u v ds = -\left[n \cdot uv\right]_{\partial\Omega}, \tag{2.23}$$

which becomes zero as the test function v is zero at the boundary. This concludes to the following variational formula: find $u \in V_0$ such that

$$\int_\Omega \nabla u \cdot \nabla v dx = \int_\Omega f v dx \tag{2.24}$$

for all $v \in V_0$.

Now, let $\mathcal{K}$ be a mesh of the domain Ω, and V_h the space of continuous piecewise linear functions on $\mathcal{K}$. The subspace $V_{h,0} \subset V_h$ is defined as

$$V_{h,0} = \{v \in V_h : v|_{\partial\Omega} = 0\} \tag{2.25}$$

to adhere to the boundary conditions of the partial differential equation. To obtain the finite element method approximation, V_0 in Equation 2.24 is replaced with $V_{h,0}$: find $u_h \in V_{h,0}$ such that

$$\int_\Omega \nabla u_h \cdot \nabla v_h dx = \int_\Omega f v_h dx \tag{2.26}$$

for all $v_h \in V_{h,0}$.

In order to compute the finite element approximation u_h, a linear system of equations is needed to be derived. First, a base is defined:

Definition 2.26. Define $\{\varphi\}_{i=1}^{n_i}$ as the basis of hat functions for $V_{h,0}$ associated with the n_i interior nodes of the mesh.

Remember, the functions in $V_{h,0}$ are zero on the boundary, so no hat functions are allowed there. All $v_h \in V_{h,0}$ are vectors in the basis of $\{\varphi\}_{i=1}^{n_i}$. Thus, they equivalently can be replaced with the basis vectors instead. Also, since $u_h \in V_{h,0}$, it is also a linear combination of the basis vectors, and can thus be written as

$$u_h = \sum_{j=1}^{n_j} \xi_j \varphi_j. \tag{2.27}$$

Using these two facts, and breaking out the sum of ξ_j from Equation 2.26, the equation

$$\sum_{j=1}^{n_j} \xi_j \int_\Omega \nabla\varphi_j \cdot \nabla\varphi_i dx = \int_\Omega f\varphi_i dx, \quad i = 1, 2, ..., n_i \tag{2.28}$$

is obtained.

Now, two matrices are defined.

Definition 2.27. The $n_i \times n_i$ *system matrix* of the finite element method is defined as

$$A_{ij} = \int_\Omega \nabla\varphi_j \cdot \nabla\varphi_i dx, \quad i, j = 1, 2, ..., n_i. \tag{2.29}$$

Definition 2.28. The $n_i \times 1$ *load vector* of the finite element method is defined as

$$a_i = \int_\Omega f\varphi_i dx, \quad i = 1, 2, ..., n_i. \tag{2.30}$$

With these, Equation 2.28 can be written as

$$\sum_{j=1}^{n_i} A_{ij}\xi_j = a_i, \quad i = 1, 2, ..., n_i, \tag{2.31}$$

or, as a matrix equation,

$$A\xi = a. \tag{2.32}$$

The vector ξ is called the solution vector. At this point, the discretization is complete. By solving this matrix equation, the unknowns ξ_j are obtained, and thus u_h.

For a summation of the finite element method, see Algorithm 2.1 below:

Algorithm 2.1 The Finite Element Method

---

1. Make a mesh $\mathcal{K}$ of the domain Ω and define the continuous piecewise linear functions space $V_{h,0}$ hat function basis $\{\varphi_i\}_{i=1}^{n_i}$.

2. Assemble the system matrix A and the load vector a as such

$$A_{ij} = \int_\Omega \nabla\varphi_j \cdot \nabla\varphi_i dx, \quad a_i = \int_\Omega f\varphi_i dx. \tag{2.33}$$

3. Solve the linear system

$$A\xi = a. \tag{2.34}$$

4. Set

$$u_h = \sum_{j=1}^{n_i} \xi_j\varphi_j. \tag{2.35}$$

A couple of notes on this. Firstly, in this example, conveniently the Poisson's equation had the boundary condition set to zero (Equation 2.18). If this is not the case, the method would still follow the exact same steps, except when constructing the load vector a the boundary conditions for the partial differential equation would need to be included.

Secondly, in this thesis the finite element method tool Elmer/Ice is used. This is a finite element method solver based on Elmer (a finite element method solver), customized to use for ice-flow modelling. The tool does a lot of the work and computations for us, and it is really only necessary to find the variational form of the partial differential equations and plug them in. For this reason, in chapter 3, where the governing equations for ice sheet modelling are discretized, the process is only explained until the variational form is obtained.

For a concise guide on how to use Elmer/Ice, see [17].

2.2.3 Newton's Method

The finite element method described in Section 2.2.2 works well for linear partial differential equations. However, for nonlinear partial differential equations iterative methods need to be implemented to reach convergence for the nonlinear terms. The simulations that are done in this thesis make use of Newton's method for nonlinear partial differential equations. To learn more about solving nonlinear partial differential equations, read chapter 9 in [6].

Newton's method can be applied on equations of the form

$$g(x) = 0, \tag{2.36}$$

where g is assumed to be a scalar non-linear function of x. To derive Newton's method, a solution $\bar{x}$ is split into an initial guess x^0 and a correction δx, so that

$$\bar{x} = x^0 + \delta x. \tag{2.37}$$

Using Taylor expansion on $g(x)$ around $\bar{x}$,

$$g(\bar{x}) = g(x^0) + g'(x^0)\delta x + \mathcal{O}(\delta x^2) \tag{2.38}$$

is acquired, where all extra terms of the Taylor expansion are included in $\mathcal{O}(\delta x^2)$. Ignoring the higher order terms in $\mathcal{O}(\delta u^2)$ and using $g(\bar{x}) = 0$, the approximation

$$0 \approx g(x^0) + g'(x^0)\delta x \tag{2.39}$$

is attained. Solving for δx, the linear equation

$$\delta x = \frac{-g(x^0)}{g'(x^0)} \tag{2.40}$$

is thus acquired. By adding δx to x^0, a closer approximation to $\bar{x}$ is expected. Now, iterating this method with x^0 replaced by $\bar{x}$ until δx is lower than a predecided tolerance, gives a convergence towards x.

As an example, Newton's method for Poisson's equation is derived. Assume the variational form of the nonlinear Poisson's equation:

$$\int_\Omega a(u)\nabla u \nabla v dx = \int_\Omega f v dx, \forall v \in V. \tag{2.41}$$

The nonlinear term is $a(u)$ as it makes u dependant on u itself, i.e. $a(u)$ is a function of u. The sought-after solution u is split into an initial guess u^0 and a correction δu. Then u can be written as

$$u = u^0 + \delta u. \tag{2.42}$$

A Taylor expansion of $a(u^0 + \delta u)$ gives

$$a(u^0 + \delta u) = a(u^0) + a'_u(u^0)\delta u + \mathcal{O}(\delta u^2). \tag{2.43}$$

Neglecting higher order terms $\mathcal{O}(\delta u^2)$, gives the equation

$$\int_\Omega \left(a(u^0) + a'_u(u^0)\delta u \nabla u^0\right) \nabla v dx = \int_\Omega f v dx, \forall v \in V. \tag{2.44}$$

The equation is to be solved for δu, so the other terms are moved to the right hand side of the equation, giving

$$\int_\Omega a'_u(u^0)\delta u \nabla u^0 \nabla v dx = \int_\Omega f v dx - \int_\Omega a(u^0)\nabla u^0 \nabla v dx, \forall v \in V. \tag{2.45}$$

This equation is solved using the finite element method, with δu as the solution vector and the right hand side as the load vector.

This method is iteratively run until a small enough value, lower than a predecided tolerance, is acquired for δu.

Newton's method is a popular method due to the fast convergence rate, however it comes with a couple of weaknesses. Since Taylor expansions are used, it requires an "adequate" initial guess of u^0 to converge at all, and computing g' can be expensive. In cases where an initial guess needs to be acquired, a method called Picard iteration can be used. This method is explained in chapter 9 of [6]. In this thesis, Elmer/Ice only make use of Newton's method.

2.3 Governing Equations

A glacier's rate of change is governed by the Stokes equation and the glacier's surface evolution is governed by the free-surface equation.

2.3.1 The Stokes Equations

The movement of a glacier can be described as a highly viscous, non-Newtonian fluid [12]. Fluids follow the laws of physics, i.e. conservation of momentum as well as conservation of mass, and are governed by the Stokes equations:

$$\nabla \cdot (2\mu(\mathbf{Du})\mathbf{Du}) - \nabla p = \rho g \hat{\mathbf{z}}, \qquad \mathbf{x} \in \Omega, \tag{2.46}$$

$$\nabla \cdot \mathbf{u} = 0, \qquad \mathbf{x} \in \Omega, \tag{2.47}$$

where $\Omega \in \mathbb{R}^d, d \in \{2, 3\}$ depending on the dimension. Here, Equation 2.46 represents conservation of momentum and equation 2.47 conservation of mass, which is an incompressibility condition. The Stokes equation is a nonlinear partial differential equation of two or three variables depending on the dimension (see Section 2.2.1). The ice velocity is $\mathbf{u}$ and the pressure p at $\mathbf{x}$ in the domain Ω. The nonlinearity is because of the viscosity, μ, which is dependent on the velocity $\mathbf{u}$ (Section 2.3.2). The strain-rate tensor is $\mathbf{Du} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$. The parameters used are $\rho = 917$ kg m^{-3} is ice density, and $g = 9.8$ m/s^2 is the acceleration of gravity.

In order to shorten the writing of the Stokes equations, define

$$\mathbf{S}(\mathbf{Du}) = 2\mu(\mathbf{Du})\mathbf{Du} \tag{2.48}$$

and let

$$\mathbf{f} = -\rho g \hat{\mathbf{z}}. \tag{2.49}$$

This gives the following expression of the Stokes equations:

$$\nabla \cdot \mathbf{S}(\mathbf{Du}) - \nabla p = -\mathbf{f}, \qquad \mathbf{x} \in \Omega, \tag{2.50}$$

$$\nabla \cdot \mathbf{u} = 0, \qquad \mathbf{x} \in \Omega. \tag{2.51}$$

2.3.2 Glen's Flow Law

The Glen's viscosity, μ, depends on the temperature and velocity. It is governed by Glen's flow law, [13] and [14],

$$\mu(\mathbf{u}, T') = A(T')^{-\frac{1}{n}} \left(\frac{1}{2} tr(\mathbf{Du}^2) + \mathbf{Du}_0^2 \right)^{\frac{1-n}{2n}}. \tag{2.52}$$

The term $\mathbf{Du}_0^2 = 10^{-10}$ yr^{-2} is a regularization term, added so that zero strain rates do not cause infinite viscosity. The term n, in this case $n = 3$, is called the power-law exponent. The value of $A(T')$ is set to 100 MPa^{-3}a^{-1}.

2.3.3 The Free-Surface Equations

To determine the upper surface position ($z_s = z_s(x, y, t)$) of a glacier at a time t the free surface-equation [12] is used:

$$\frac{\partial z_s}{\partial t} + u_x^s \frac{\partial z_s}{\partial x} + u_y^s \frac{\partial z_s}{\partial y} = u_z^s + a_s. \tag{2.53}$$

Since on a glacier it will snow and melt, a_s describes the accumulation or ablation of mass. From the Stokes equations 2.46 and 2.47 we get for the surface the velocity field $\mathbf{u}^s = (u_x^s, u_y^s, u_z^s)$.

2.3.4 Boundary Conditions

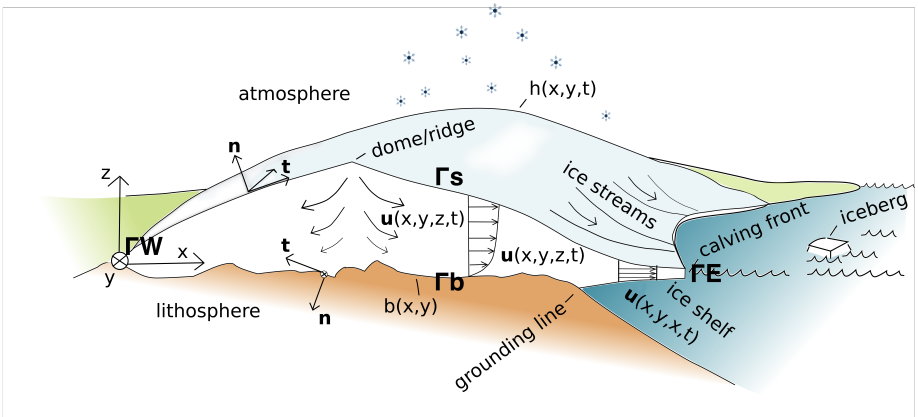

Figure 2.4: An ice sheet with the boundaries Γ_s, Γ_W, Γ_b and Γ_E marked out.

For the glacier boundary $\partial\Omega$ we specify non-overlapping boundary parts Γ_s, Γ_W, Γ_b and Γ_E. Except for the surface Γ_s, the boundaries are stationary. This means the change of the glacier completely depends on the evolution of the surface. A mountain glacier is a smaller ice sheet.

See Figure 2.4 for a schematic of a general ice sheet, with the boundary parts included. The ice-water boundary shown here is not necessary when modelling glaciers. Note that the length of the ice sheet is displayed in orthogonal directions x and y, and the height of the ice sheet is in the z-direction.

The following boundary conditions are considered for the different parts of the boundary

$$\sigma\hat{\mathbf{n}} = \mathbf{0}, \qquad \mathbf{x} \in \Gamma_s, \tag{2.54}$$

$$\mathbf{u} \cdot \hat{\mathbf{n}} = 0, \qquad \mathbf{x} \in \partial\Omega/\Gamma_s, \tag{2.55}$$

$$\hat{\mathbf{t}}_i \cdot \sigma\hat{\mathbf{n}} = -\beta^2 |\mathbf{u}|^{m-1} \mathbf{u} \cdot \hat{\mathbf{t}}_i, \qquad \mathbf{x} \in \Gamma_b, \tag{2.56}$$

The tensor $\sigma = 2\mu\mathbf{D}\mathbf{u} - pI$, where I is the identity matrix, is called the Cauchy stress tensor, $\hat{\mathbf{n}}$ is the unit normal pointing outward from the boundary and $\{\hat{\mathbf{t}}_i\}_{i=0}^{d-1}$ are tangent vectors, spanning the plane defined by $\hat{\mathbf{n}}$. The term β is the drag coefficient and m is an exponent. Equation 2.54 shows a stress-free condition on the glacier surface. This follows from the assumption that the stresses asserted on the surface are negligible compared to internal stresses of the glacier. Equation 2.55 describes the impenetrable condition of the glacier-bedrock. It is set to zero because both accumulation and loss of ice at the bedrock are negligible. Equation 2.56 states that the ice may slip along the bedrock, called a Weertman-type sliding law. We put $m = 1$, so the relation is linear.

3 The Numerical Methods for Ice-Flow Modelling

In this part, the use of numerical methods to solve the governing equations of ice-flow modelling are explained, the Free-surface stabilization algorithm (FSSA) is derived and the algorithm for the minimum thickness condition is introduced. Then, the numerical experiments are introduced.

The hypothesis is that using the Free-surface stabilization algorithm (FSSA) will give different velocities where the minimum thickness condition comes into play, resulting in a different surface. This is because a minimum thickness condition is used when solving the free-surface equation, but FSSA doesn't take this minimum thickness condition into account when approximating the domain the velocities are calculated on.

3.1 Discretization of The Stokes Equations

From Section 2.3.1 we get:

$$\nabla \cdot \mathbf{S}(\mathbf{Du}) - \nabla p = -\mathbf{f} \tag{3.1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{3.2}$$

Let $\mathbf{u} = \mathbf{u}(\mathbf{x})$ where $\mathbf{x} = (x, y)$. In order to apply the finite element method on the Stokes equations, a discretization needs to be done. Firstly, two test spaces are introduced:

$$V = \{v : ||v||_{L^2(\Omega)} + ||\nabla v||_{L^2(\Omega)} < \infty, \mathbf{u}|_{\Gamma_b} = 0\}, \tag{3.3}$$

$$Q = \{q : ||q||_{L^2(\Omega)} < \infty, \int_\Omega q(\mathbf{x}) d\mathbf{x} = 0\}. \tag{3.4}$$

Equation 3.1 is multiplied with -1 and the Stokes equations are multiplied with test functions $\mathbf{v} \in V$ and $q \in Q$ such that

$$-\nabla \cdot \mathbf{S}(\mathbf{Du}) \cdot \mathbf{v} + \nabla p \cdot \mathbf{v} = \mathbf{f} \cdot \mathbf{v}, \tag{3.5}$$

$$\nabla \cdot \mathbf{u} q = 0. \tag{3.6}$$

The equations are then integrated over the domain Ω and this yields

$$-\int_\Omega \nabla \cdot \mathbf{S}(\mathbf{Du}) \cdot \mathbf{v} d\mathbf{x} + \int_\Omega \nabla p \cdot \mathbf{v} d\mathbf{x} = \int_\Omega \mathbf{f} \cdot \mathbf{v} d\mathbf{x}, \tag{3.7}$$

$$\int_\Omega \nabla \cdot \mathbf{u} q d\mathbf{x} = 0. \tag{3.8}$$

For now 3.8 is ignored and 3.7 is dealt with onward. Green's identity is applied to both the first and second integral on the left hand side to acquire

$$\int_\Omega \mathbf{S}(\mathbf{Du}) : \nabla \mathbf{v} d\mathbf{x} - \int_\Gamma \hat{\mathbf{n}} \cdot \nabla \cdot \mathbf{S}(\mathbf{Du}) \cdot \mathbf{v} ds - \int_\Omega p \nabla \cdot \mathbf{v} d\mathbf{x} + \int_\Gamma \hat{\mathbf{n}} \cdot p\mathbf{v} ds = \int_\Omega f \mathbf{v} d\mathbf{x}. \tag{3.9}$$

The integrals over the boundary are simplified on their own. We first collect the terms under the same integral such that

$$-\int_\Gamma \hat{\mathbf{n}} \cdot \nabla \cdot \mathbf{S}(\mathbf{Du}) \mathbf{v} ds + \int_\Gamma \hat{\mathbf{n}} \cdot p\mathbf{v} ds = \int_\Gamma \hat{\mathbf{n}} \cdot (-\nabla \cdot \mathbf{S}(\mathbf{Du}) + p)\mathbf{v} ds = -\int_\Gamma \hat{\mathbf{n}} \cdot (\nabla \cdot \mathbf{S}(\mathbf{Du}) - p)\mathbf{v} ds. \tag{3.10}$$

The parenthesis contains the Cauchy stress tensor, in Section 2.3.4 defined as σ. The integral is divided into the different boundaries, also defined in Section 2.3.4. Thus, this yields

$$-\int_{\Gamma} \sigma\hat{\mathbf{n}} \cdot \mathbf{v} ds = \\ -\int_{\Gamma_s} \sigma\hat{\mathbf{n}} \cdot \mathbf{v} ds - \int_{\Gamma_E} \sigma\hat{\mathbf{n}} \cdot \mathbf{v} ds - \int_{\Gamma_W} \sigma\hat{\mathbf{n}} \cdot \mathbf{v} ds - \int_{\Gamma_b} \sigma\hat{\mathbf{n}} \cdot \mathbf{v} ds. \quad (3.11)$$

Now, due to the boundary conditions defined in Part 2.3.4 together with that $ds = 0$ for Γ_W and Γ_E, all integrals but the one over Γ_b becomes 0, so 3.11 becomes

$$-\int_{\Gamma_b} \sigma\hat{\mathbf{n}} \cdot \mathbf{v} ds. \quad (3.12)$$

Let now the basis of $\hat{\mathbf{n}}$ in two dimensions be defined as

$$\hat{\mathbf{n}} = \mathbf{n}^{\perp} + \mathbf{n}^{\parallel}. \quad (3.13)$$

The theorem for change of basis for orthonormal bases in linear algebra yields

$$\sigma \cdot \hat{\mathbf{n}} = \sigma\mathbf{n}^{\perp} + \sigma\mathbf{n}^{\parallel} = ((\sigma\hat{\mathbf{n}}) \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{n}} + ((\sigma\hat{\mathbf{n}}) \cdot \hat{\mathbf{t}}) \cdot \hat{\mathbf{t}}, \quad (3.14)$$

where $\hat{\mathbf{t}}$ is a tangent vector, spanning the vector defined by $\hat{\mathbf{n}}$. Including this in 3.12 yields the integral

$$-\int_{\Gamma_b} \left[((\sigma\hat{\mathbf{n}}) \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{n}} + ((\sigma\hat{\mathbf{n}}) \cdot \hat{\mathbf{t}}) \cdot \hat{\mathbf{t}}\right] \cdot \mathbf{v} ds = -\int_{\Gamma_b} ((\sigma\hat{\mathbf{n}}) \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{n}} \cdot \mathbf{v} + ((\sigma\hat{\mathbf{n}}) \cdot \hat{\mathbf{t}}) \cdot \hat{\mathbf{t}} \cdot \mathbf{v} ds. \quad (3.15)$$

The first term becomes 0 because of the definition of the space in Equation 3.3. Further, the second term can be rewritten using Equation 2.56 in Section 2.3.4. This yields

$$-\int_{\Gamma_b} -\beta^2 \left[(\mathbf{u} \cdot \hat{\mathbf{t}}) \cdot \hat{\mathbf{t}}\right] \cdot \mathbf{v} ds = \int_{\Gamma_b} \beta^2 (\mathbf{u} - (\mathbf{u} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{n}}) \cdot \mathbf{v} ds = \int_{\Gamma_b} \beta^2 \mathbf{u} \cdot \mathbf{v} ds, \quad (3.16)$$

by dividing $\mathbf{u}$ into its components and solve for $(\mathbf{u} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{n}}$, which becomes zero due to Equation 2.55. Now reintroducing the result from simplifying the integral over the boundary to 3.9 yields

$$\int_{\Omega} \mathbf{S}(\mathbf{Du}) : \nabla \mathbf{v} d\mathbf{x} - \int_{\Omega} p\nabla \cdot \mathbf{v} d\mathbf{x} + \int_{\Gamma_b} \beta^2 \mathbf{u} \cdot \mathbf{v} ds = \int_{\Omega} f\mathbf{v} d\mathbf{x}. \quad (3.17)$$

The result of the discretization, reintroducing Equation 3.8, thus becomes

$$\int_{\Omega} \mathbf{S}(\mathbf{Du}) : \nabla \mathbf{v} d\mathbf{x} - \int_{\Omega} p\nabla \cdot \mathbf{v} d\mathbf{x} + \int_{\Gamma_b} \beta^2 \mathbf{u} \cdot \mathbf{v} ds = \int_{\Omega} f\mathbf{v} d\mathbf{x}, \quad (3.18)$$

$$\int_{\Omega} \nabla \cdot \mathbf{u} q d\mathbf{x} = 0. \quad (3.19)$$

In the actual implementation of the Stokes equations, 3.19 is subtracted from both equations, so the Stokes equations becomes one equation. This yields the variational problem: find $(\mathbf{u}, p) \in V \times Q$ such that

$$\int_{\Omega} \mathbf{S}(\mathbf{Du}) : \nabla \cdot \mathbf{v} d\mathbf{x} - \int_{\Omega} p\nabla \cdot \mathbf{v} d\mathbf{x} + \int_{\Gamma_b} \beta^2 \mathbf{u} \cdot \mathbf{v} ds - \int_{\Omega} \nabla \cdot \mathbf{u} q d\mathbf{x} - \int_{\Omega} f\mathbf{v} d\mathbf{x} = 0 \quad (3.20)$$

for all $\mathbf{v} \in V$ and all $q \in Q$.

3.2 Discretization of The Free-Surface Equation

Let k be the current time step. The upper surface of the glacier is denoted z_s and the lower surface of the glacier z_b. The free-surface equation is discretized with regards to time using a semi-implicit Euler discretization. It is called semi-implicit since it is implicit with regards to the surface z_s, but explicit in terms of $\mathbf{u}$. The discretization depends on that to solve for $\mathbf{u}^k$ the surface area Ω^{k+1} is needed, described by z_s^{k+1}.

We are solving for z_s^{k+1}. The implicit approximation of its derivative is described as

$$\frac{\partial z_s}{\partial t} \approx \frac{z_s^{k+1} - z_s^k}{\Delta t}. \tag{3.21}$$

Here Δt is the time-step from k to $k+1$. Using this approximation in the free-surface equation for two dimensions we acquire

$$\frac{z_s^{k+1} - z_s^k}{\Delta t} + u_x^k \frac{\partial z_s^{k+1}}{\partial x} = u_z^k + a_s^k. \tag{3.22}$$

Multiplication with the time-step and addition of z_s^k solves for z_s^{k+1} and gives us

$$z_s^{k+1} + \Delta t \left(u_x^k \frac{\partial z_s^{k+1}}{\partial x} \right) = z_s^k + \Delta t (u_z^k + a_s^k), \tag{3.23}$$

the complete time-discretization of the free-surface equation.

3.3 The Free-Surface Stabilization Algorithm (FSSA)

A fully explicit forward Euler time discretization solves the Stokes equations on the domain Ω^k for $\mathbf{u}^k$ and then uses $\mathbf{u}^k$ as coefficients in the free-surface equations, obtaining Ω^{k+1}. This approach requires small time steps, to avoid numerical instability [2]. A fully implicit backward Euler time discretization avoids the numerical instability, but instead needs to solve the Stokes equations twice in each iteration, having a higher computational cost.

In Section 3.2 we used a semi-implicit Euler time discretization. Its discretization is implicit in terms of z_s but explicit in terms of $\mathbf{u}^k$. This means it still have the issue with numerical instability, but the free surface stabilization algorithm (FSSA) solves that by mimicking a fully implicit scheme. An estimate of the domain in the next time-step is approximated by applying gravity's impact of force on the domain. This is done using Reynold's transport theorem, which standard form can be found in [12]. The FSSA was first used with mantle-convection simulations [15] and then adapted to ice-sheet modeling in 2022 [2].

The FSSA is based on computing an estimate $(\tilde{\mathbf{u}}^{k+1}, \tilde{p}^{k+1})$ instead of $(\mathbf{u}^{k+1}, p^{k+1})$ for the full-Stokes equations: Find $(\mathbf{u}^{k+1}, p^{k+1}) \in V \times Q$ such that

$$\int_{\Omega^k} \mathbf{S}(\mathbf{D}\mathbf{u}^\mathbf{k}) : \nabla \mathbf{v} d\mathbf{x} - \int_{\Omega^k} p^k \nabla \cdot \mathbf{v} d\mathbf{x} + \int_{\Gamma_b} \beta^2 \mathbf{u}^\mathbf{k} \cdot \mathbf{v} ds = \int_{\Omega^k} f \mathbf{v} d\mathbf{x}, \tag{3.24}$$

$$\int_{\Omega^k} \nabla \cdot \mathbf{u}^\mathbf{k} q d\mathbf{x} = 0, \tag{3.25}$$

for all $\mathbf{v} \in V$ and all $q \in Q$, where V and Q are defined as

$$V = \{v : ||v||_{L^2(\Omega)} + ||\nabla v||_{L^2(\Omega)} < \infty, \mathbf{u}|_{\Gamma_b} = 0\}, \tag{3.26}$$

$$Q = \{q : ||q||_{L^2(\Omega)} < \infty, \int_\Omega q(\mathbf{x})d\mathbf{x} = 0\}. \tag{3.27}$$

To solve for $\mathbf{u}^{k+1}$ the right hand side of Equation 3.24 is modified to an approximation of $\int_{\Omega_{k+1}} \mathbf{f} \cdot \mathbf{v} d\Omega$. This is done using Reynold's transport theorem, which states that

$$\frac{d}{dt}\int_{\Omega(t)} g d\mathbf{x} = \int_{\Omega(t)} \frac{\partial}{\partial t} g d\mathbf{x} + \int_{\partial\Omega(t)} (\mathbf{u} \cdot \hat{\mathbf{n}}) g ds. \tag{3.28}$$

Here $\Omega(t) = \Omega^k$ for a specific time-step k. Since $g = \mathbf{f} \cdot \mathbf{v}$, which depends on constants and $\hat{\mathbf{z}}$, we get

$$\int_{\Omega(t)} \frac{\partial}{\partial t} g d\mathbf{x} = 0. \tag{3.29}$$

The left hand side of Equation 3.28 is approximated in terms of integrals over Ω^k explicitly, such that

$$\frac{d}{dt}\int_{\Omega^k} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} \approx \frac{\int_{\Omega^{k+1}} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} - \int_{\Omega^k} \mathbf{f} \cdot \mathbf{v} d\mathbf{x}}{\Delta t}. \tag{3.30}$$

Combining Equation 3.29 and Equation 3.30, and solving for $\int_{\Omega^{k+1}} \mathbf{f} \cdot \mathbf{v} d\mathbf{x}$ in Reynolds transport theorem gives the approximation

$$\int_{\Omega^{k+1}} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} = \int_{\Omega^k} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} + \Delta t \int_{\partial\Omega^k} (\mathbf{u} \cdot \hat{\mathbf{n}})(\mathbf{f} \cdot \mathbf{v}) ds. \tag{3.31}$$

If we now put this result as the right hand side into the variational formulation of the Stokes equations we get: find $(\tilde{\mathbf{u}}^{k+1}, \tilde{p}^{k+1}) \in V \times Q$ such that

$$\int_{\Omega^k} \mathbf{S}(\mathbf{D}\tilde{\mathbf{u}}^{\mathbf{k}}) : \nabla \mathbf{v} d\mathbf{x} - \int_{\Omega^k} \tilde{p}^{k+1}(\nabla \cdot \mathbf{v}) d\mathbf{x} = \int_{\Omega^k} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} + \Delta t \int_{\partial\Omega^k} (\tilde{\mathbf{u}}^{k+1} \cdot \hat{\mathbf{n}})(\mathbf{f} \cdot \mathbf{v}) ds, \tag{3.32}$$

$$\int_{\Omega^k} \nabla \cdot \tilde{\mathbf{u}}^{k+1} \mathbf{q} d\Omega = 0, \tag{3.33}$$

for all $\mathbf{v} \in V$ and all $q \in Q$. FSSA adapted for glacier modelling needs an additional term for ablation/accumulation. To account for the accumulation, a_s, the surface velocity $\mathbf{u}$ in Reynold's transport theorem is replaced by $\mathbf{u} + a_s\hat{\mathbf{z}}$. The integral for the FSSA is then split on the addition between $\mathbf{u}$ and $a_s\hat{\mathbf{z}}$ and the FSSA term for $\mathbf{u}$ is moved to the left hand side of the equation, giving the final variational form for ice-flow: find $(\tilde{\mathbf{u}}^{k+1}, \tilde{p}^{k+1}) \in V \times Q$ such that

$$\int_{\Omega^k} \mathbf{S}(\mathbf{D}\tilde{\mathbf{u}}^{\mathbf{k}}) : \nabla \mathbf{v} d\mathbf{x} - \int_{\Omega^k} \tilde{p}^{k+1} \nabla \cdot \mathbf{v} d\mathbf{x}$$
$$-\theta\Delta t \int_{\partial\Omega^k} (\tilde{\mathbf{u}}^{k+1} \cdot \hat{\mathbf{n}})(\mathbf{f} \cdot \mathbf{v}) ds = \int_{\Omega^k} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} + \theta\Delta t \int_{\partial\Omega^k} (a_s\hat{\mathbf{z}} \cdot \hat{\mathbf{n}})(\mathbf{f} \cdot \mathbf{v}) ds, \tag{3.34}$$

$$\int_{\Omega^k} \nabla \cdot \tilde{\mathbf{u}}^{k+1} \mathbf{q} d\mathbf{x} = 0, \tag{3.35}$$

for all $\mathbf{v} \in V$ and all $q \in Q$. This is the free-surface stabilization algorithm. A parameter $\theta \in \{0, 1\}$ has been included in Equation 3.34, where $\theta = 0$ gives an explicit solver without FSSA, and $\theta = 1$ gives a quasi-implicit solver with FSSA activated. This is done to easily be able to compare the FSSA with an explicit solver without the FSSA.

3.4 Minimum thickness condition

When we are simulating glaciers, we need to decide a minimum thickness the ice can reach. If this condition is not in place, it's possible for the upper surface, z_s, to go below the lower surface z_b. As this is not physically possible, it's unwanted in our simulations. In order to prevent the upper surface to fall below the lower surface, Elmer/Ice lets us choose a minimum thickness of the ice, $\mathbf{h}_{\min}$, and uses the algorithm described in [16] when solving the discretized free-surface equation (Equation 3.2) under the constraint. We need a fix so that $z_s > z_b + h_{\min}$. The algorithm works as follows.

Define $\mathbf{A}$ as the system matrix and $\mathbf{a}$ the load vector. Then $\mathbf{h}$ is the solution vector containing the values for the upper surface z_s at the nodes. The matrix system of the unconstrained system being solved is

$$\mathbf{A} \cdot \mathbf{h} = \mathbf{a}.$$

We want to solve this equation under the constraint that $\mathbf{h} > \mathbf{h}_{\min}$.

- The algorithm goes through the nodes in $\mathbf{h}$, and if one is found to not satisfy the minimum thickness condition, it is set as active and added to a set of active nodes.
- For each active node i, the Dirichlet condition $h_i = h_{\min i}$ is introduced by changing row i in the system matrix to $A_{ij} = \delta_{ij}$, where

  $$\delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases}$$

  and the ith entry of the body force to $a_i = h_{\min i}$, resulting in the constrained equation

  $$\mathbf{A}' \cdot \mathbf{h}' = \mathbf{a}'.$$

- Solve the constrained equation in the previous step for $\mathbf{h}'$.
- Obtaining $\mathbf{h}'$, it's put into the unconstrained system, replacing $\mathbf{h}$. A residual is defined as

  $$\mathbf{R} = \mathbf{A} \cdot \mathbf{h}' - \mathbf{a}.$$

  This residual can physically be interpreted ad the additionally needed ablation at the nodes for the requirement of the minimum ice thickness.
- If an active node i now complies with $h_i > h_{\min i}$ it is set as inactive and removed from the set if and only if $R_i < 0$.
- This is repeated until there's no change in the set of active nodes and the convergence criteria for the free surface solver is met.

Forcing this constraint using the algorithm results in altered forces within the ice, which in turn results in altered velocities.

3.5 Error estimation

In order to compare the results, an error needs to be estimated. In this section, the equations used for error calculations are introduced.

3.5.1 Error Calculation

Let h be the experiment, and h^* be a reference solution. Then the total error for the variable in question is calculated as

$$\epsilon = \frac{1}{|\Omega_h|} \int_{\Omega_h} |h - h^*| \, dx. \tag{3.36}$$

3.5.2 Velocity Magnitude

Comparing the differences in velocity for v_x and v_z separately might give an idea of the errors, but for a complete picture a comparison of the absolute velocity needs to be done. To be able to calculate the error of the whole velocity, rather than v_x and v_z by themselves, velocity magnitude is used.

Let $v_{x_1}, v_{x_2}, ..., v_{x_n}$ be the velocities in n different directions. Then the velocity magnitude, V, is calculated as

$$V = \sqrt{v_{x_1}^2 + v_{x_2}^2 + ... + v_{x_n}^2}. \tag{3.37}$$

3.6 Numerical experiments

The aim of the experiments are to investigate how simulations behave using the FSSA together with the minimum thickness condition. The algorithm for minimum thickness is already implemented in Elmer/Ice, so we run simulations without FSSA first as benchmarks to compare with the simulation using FSSA.

As described in part 3.3, the FSSA approximates the domain for the next time step, to more accurately calculate the velocities of the ice when solving the discretization of the Stokes equations. The FSSA, however, may not fully account for the minimum thickness condition, which is applied when calculating the surface. With this in mind, our hypothesis is that the velocity field for the FSSA will be different, as it is possible the condition $z_s > z_b + h_{min}$ will not hold when solving the Stokes equations when the glacier is retreating. For the point at which the glacier reaches h_{min}, the word limit is used.

All experiments are simulated for 500 a (years), where we chose to look at the period from 100 a to 200 a. The reason for this is that before 100 a, the glacier was advancing at some moments. We are interested in the retreating case, as that is when the nodes become active with the minimum thickness condition. If the glacier is advancing, there is no need to check that $\mathbf{h} > \mathbf{h}_{\min}$ as we are increasing thickness. For comparison, there is no need to use a longer period than 100 a.

The first experiment is our reference solution, using forward Euler. Here we use a much smaller time step to get a more exact solution to compare our next experiments with, i.e. $\theta = 0$ for the simulation in equation 3.34. Experiment 2 is with the same solver as the reference solution, with a larger time step. Experiment 3 is the same solver, but with FSSA implemented, i.e. the simulation

with $\theta = 1$, with the same time step as Experiment 2. Experiment 4 compares the evolution of the surface and the velocities for 10 a for a simulation with $\theta = 0$ and a simulation with $\theta = 1$, starting at 100 a, with a smaller time-step but still larger than the reference solution.

3.6.1 Perlin Glacier

The glacier used for the experiment is a 2D "Perlin" synthetic glacier. This means the bedrock is generated by using Perlin noise, that produces smooth, psuedo-randomly generated terrain. How this is done is described in detail in [5]. The glacier used here is the moderately sloping glacier in that thesis, with some data pre-processing used on the lower limit. For a picture of the glacier at the start of our simulations (0 a), see Figure 3.1.

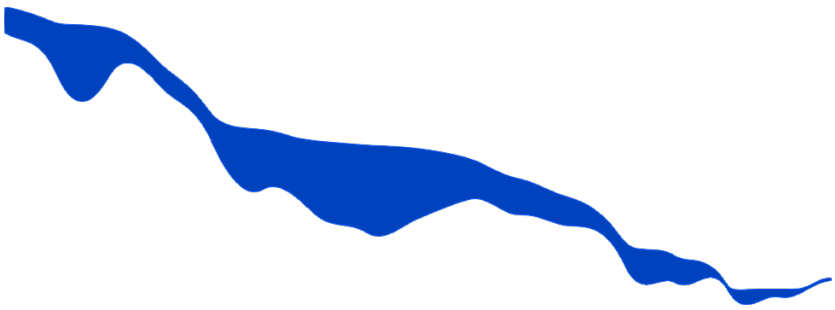

Figure 3.1: The initial look of the Perlin glacier.

The horizontal extent of the domain is $L_x = 8000$ m, and the bedrock is sloping, with undulations, starting at a high point of 800 m at $x = 0$ and going down to 5 m at $x = 8000$. To build up a glacier on the bedrock a positive accumulation function is used:

$$a(x) = max\left(1 - \frac{3x}{L_x}, 0\right).$$

This function will linearly decay along the horizontal coordinates, with a maximum at $x = 0$.

Accumulation and sliding conditions are present. The boundary conditions imposed are those described in Section 2.3.4.

The data pre-processing used was to set the minimum thickness condition to $\mathbf{h}_{\text{min}} = 10$. This was done by creating a copy of the file describing the bedrock and modifying it. The bedrock is described in a file consisting of two

columns. The first column contained the position in the horizontal plane, and the second column described the height of the bedrock. In order to set the minimum thickness condition to $\mathbf{h}_{\min} = 10$, the value 10 was added to all values in the second column using a Python script. This new file was then used as z_s lower limit.

Note that since the glacier is two dimensional, the y-dimension does not exist for the domain. Only the x-axis and the z-axis are considered.

3.6.2 Mesh

The mesh that is used is a structured mesh, with an equidistant grid in the x-direction, having different distances between nodes in the z-direction depending on the ice-thickness. The mesh uses triangles as the finite elements. There are five equidistant layers of nodes in the z-direction, with five nodes in the z-direction being placed every 20 meters in the x-direction. This means the mesh consists of a total of 2000 nodes.

3.6.3 Solution Algorithm of Elmer/Ice for the Explicit Euler Method

The time-stepping used in these simulations are the explicit Euler method. How Elmer/Ice updates the Stokes and the free-surface equation each time-step, is explained in Algorithm 3.1. The procedure for the explicit Euler method is taken from [5].

Algorithm 3.1 Explicit Euler Method

Let Δt be the size of the time-step and k the current time-step.
1. Solve the Stokes equations for $\mathbf{u}^k$ on the domain Ω^k.
2. Solve the free-surface equation for the new velocities $\mathbf{u}^k$ acquiring the new domain $\Omega^{k+\Delta t}$.
3. Start over with the next time-step $k + 1$.

3.6.4 Experiment 1: Reference Solution

The first experiment is a reference solution, a benchmark to compare our other experiments with. We expect the simulations to become less accurate the larger the time-step, so for this experiment we let the time-step be small: 0.5 a. The simulation was performed with $\theta = 0$ in Equation 3.34, i.e. FSSA is not activated.

We need a reference simulation so we can compare the performance of our other simulations with something we expect to be accurate.

3.6.5 Experiment 2: Larger Time-Step

Experiment 2 is the same simulation as in Experiment 1, with the difference that the time-step now is 5 a. With a smaller time-step, the simulations take longer to run, as in each time-step the discretized Stokes equations and free-surface equation have to be solved.

3.6.6 Experiment 3: FSSA Implementation

For this experiment, we set $\theta = 1$ in Equation 3.34. We use the same time-step as in Experiment 2, i.e. 5 a. Hence, we can both compare this experiment with how Experiment 2 perform, and the accuracy using the reference simulation in Experiment 1.

3.6.7 Experiment 4: Comparison with smaller Time-Step

This experiment was designed after seeing the results of the Experiment 1, Experiment 2 and Experiment 3. The purpose of this experiment is to look at the geometry at the nodes close to the limit, where the minimum thickness constraint comes into play.

Starting at the state of the reference simulation at 100 a, a simulation with $\theta = 0$ and a simulation with $\theta = 1$, both with a time-step of 1 a, is performed until 110 a. The smaller time-step is to avoid instabilities. To explain how the FSSA together with the minimum thickness constraint affect the velocities at the limit of the glacier, errors for each time-step between the simulation with $\theta = 0$ and the simulation with $\theta = 1$ are shown.

4 Results

Here, the results of the four experiments performed are presented. The glacier advanced for some parts of the first century, and since our interest lies in the retreating case, the experiments start at 100 a.

4.1 Experiment 1: Reference Solution

Experiment 1 is the reference solution, with a time-step of 0.5 a. As this simulation is for 500 a, just like Experiment 2 and Experiment 3, there are 1000 time-steps in which the Stokes equation and the free-surface equation are solved. This can be seen in Table 4.1, where Experiment 1 has a five times longer runtime than Experiment 3.

For the reference solution, we see in Figure 4.1 the glacier melting as time passes. For 100 a and 150 a, the limit of the glacier, where the minimum thickness condition is found, is between the points $x = 6500$ and $x = 7000$. For 200 a the limit is at $x \approx 5750$. Thus, the glacier is retreating during our time period.

We can also find the limit by looking at Figure 4.2. Here we clearly see that the ice thickness reaches the minimum value of $h_{\text{min}} = 10$ at $x = 7000$ for 100 a, about $x \approx 6750$ for 150 a, and around $x \approx 5750$ for 200 a. From this we can conclude that the glacier retreated faster in the period 150-200 a than in the period 100-150 a.

Figure A.1 in the Appendix shows the velocity in both x- and z-direction for the limit of the glacier. The velocity of the ice slows down over time, but tends toward a steady state.

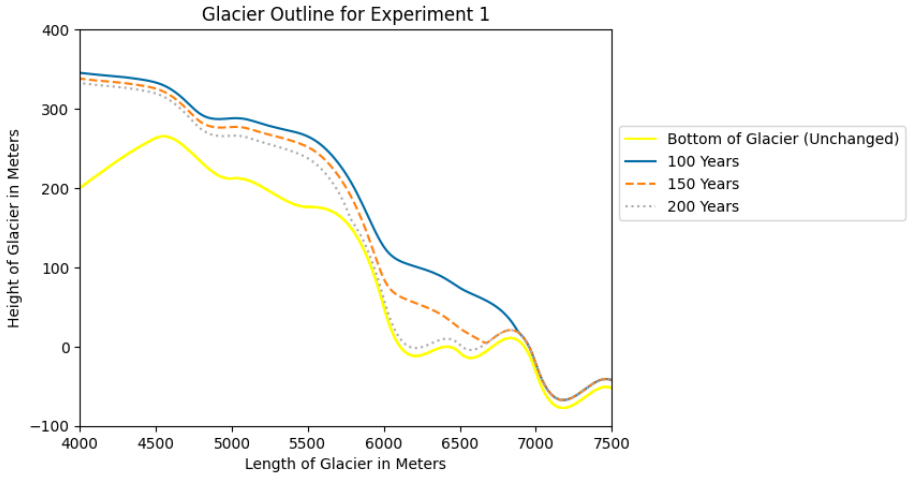

Figure 4.1: The development of the glacier at the limit of Experiment 1: Reference solution.

Experiment	Runtime
Experiment 1	49 minutes, 48 seconds
Experiment 2	4 minutes, 35 seconds
Experiment 3	9 minutes, 27 seconds

Table 4.1: Table over the runtime for each experiment.

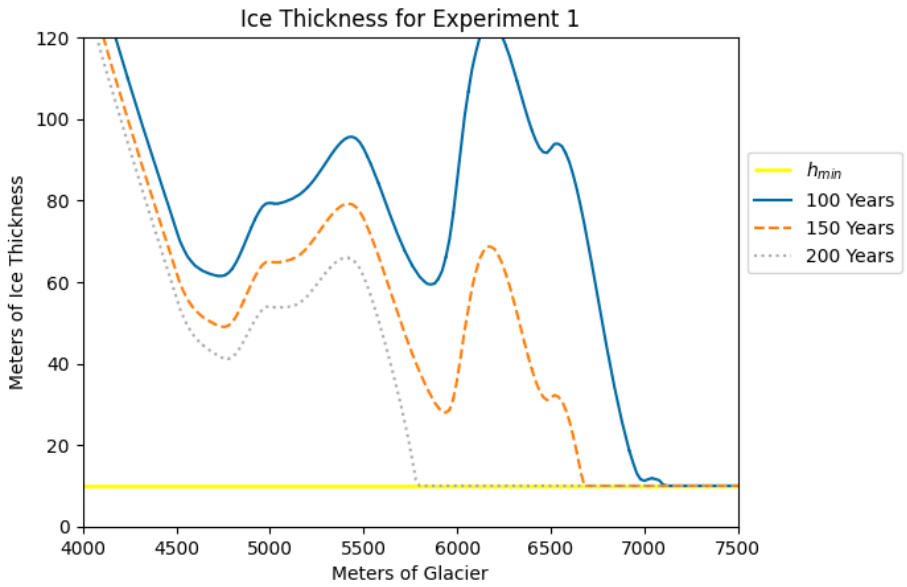

Figure 4.2: Ice thickness at the limit of Experiment 1.

4.2 Experiment 2: Larger Time-Step

In Experiment 2, we run the same simulation as Experiment 1, but with a larger time-step of 5 a. Thus, it is the same experiment as Experiment 3, but with the difference that $\theta = 0$, i.e. FSSA inactivated.

Figure A.2 in the Appendix shows that the limit is at $x \approx 4500$ for all three time-steps looked at. Moreover, the glacier is growing in thickness as time goes on. For 100 a, there are two isolated regions past the limit where the ice thickness is larger than $h_{\min} = 10$. In Figure A.3 in the Appendix, this becomes more clear, as we see the three tops for 100 a, and that the line for 150 a is above 100 a, and the line for 200 a is above them both. The velocity, found in Figure A.4 in the Appendix, we find part of the explanation for the growing glacier. For the horizontal velocity, v_x, the values of the velocity dips for 100 a at $x \approx 4500$, while for 150 a and 200 a the values of the velocity is more even in that period.

The development of the glacier for Experiment 2 differ a lot from Experiment 1. This becomes apparent in Figure A.8, A.9 and A.10 in the Appendix. We see that the outline for the glacier of Experiment 2 is lower than for Experiment

Error	Experiment 2	Experiment 3
Surface Elevation		
100 a	40.27	2.06
150 a	24.36	2.16
200 a	13.65	1.06
Velocity Magnitude		
100 a	3.88	0.18
150 a	2.20	0.11
200 a	1.59	0.05

Table 4.2: Table over the error of the surface and velocity magnitude for Experiment 2: Larger time step and Experiment 3: FSSA implementation, compared to Experiment 1: Reference solution, rounded to two decimals.

1, even if the difference gets smaller the more time passes. The velocity for Experiment 2 is closer to zero for both v_x and v_z compared to Experiment 1.

The magnitude of the difference in Velocity Magnitude and Surface Elevation between Experiment 2 and Experiment 1 can be seen in Table 4.2. How the error is calculated is described in Section 3.5.1. The meaning of "Velocity Magnitude" is described in Section 3.5.2. The running time of Experiment 2, however, is the shortest, as seen in Table 4.1.

4.3 Experiment 3: FSSA Implementation

Experiment 3 is the same experiment as Experiment 2, but now with the FSSA implementation, with $\theta = 1$, as described in Section 3.3.

The glacier outline for Experiment 3 can be seen in Figure A.5 in the Appendix. The limit for this experiment is between $x = 6500$ and $x = 7000$ for all three time-steps, with large differences between the time steps between $x = 6000$ and $x = 7000$. The large difference in ice thickness between $x = 6000$ and $x = 7000$ is also apparent in Figure A.6 in the Appendix. The thickness does not change as much between the time-steps for $x < 6000$. Surface velocity for Experiment 3 is shown in Figure A.7 in the Appendix. The same trend as A.5 and A.6 is seen for the velocity, with large differences between $x = 6000$ and $x = 7000$. For that space, the velocity of both v_x and v_z at 200 a is essentially zero. The velocities for the nodes at $x < 5000$ do not change much, with the velocities for 150 a and 200 a being more or less the same.

Experiment 3 follows Experiment 1, the reference solution, closer than Experiment 2, the same simulation as Experiment 3 except without FSSA. In Figure A.11, 4.3 and 4.4 it is sometimes hard to see the line for Experiment 3, as it closely follows the lines of Experiment 1. We also see this in Table 4.2, where the error of Experiment 3 is about $5\% - 10\%$ of the error of Experiment 2, both in surface area and velocity magnitude.

The running time of Experiment 3 is about twice that of Experiment 2, as seen in Table 4.1.

Comparing where the error is largest between Experiment 2 and Experiment 3, we can see in Figure 4.5 that Experiment 3, where $\theta = 1$ is more exact in the

middle of the glacier, but have a higher error at the limit. This results is, as previously stated, probably tainted by instability in Experiment 2.

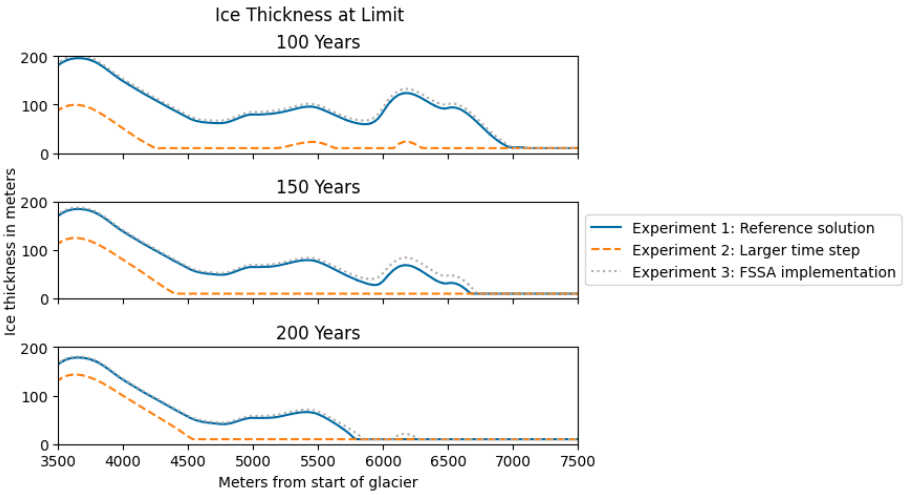

Figure 4.3: Ice thickness at the limit of the simulation for Experiment 1: Reference Solution, Experiment 2: Larger Time-Step and Experiment 3: FSSA Implementation.

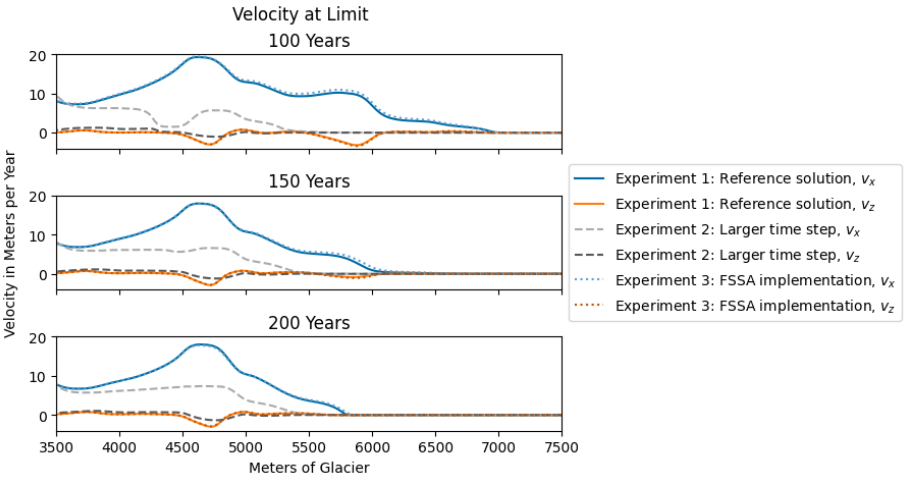

Figure 4.4: Velocity in v_x and v_z for the surface of the glacier at the limit for Experiment 1: Reference Solution, Experiment 2: Larger Time-Step and Experiment 3: FSSA Implementation.

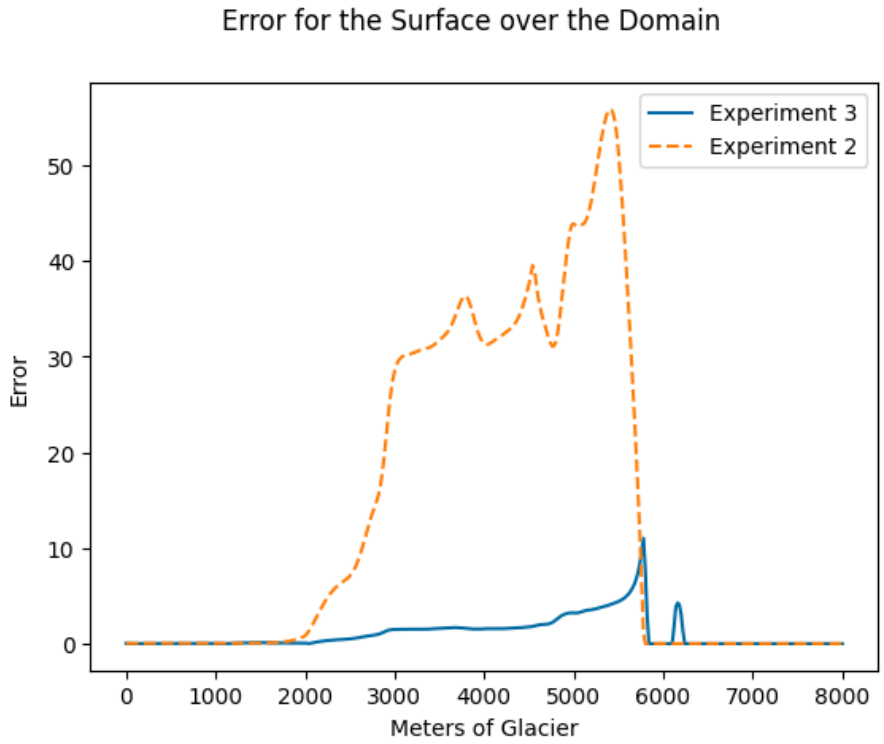

Figure 4.5: The error at each node of the surface for Experiment 2: Larger Time-Step and Experiment 3: FSSA Implementation.

4.4 Experiment 4: Comparison with smaller Time-Step

This experiment consists of two simulations. Both are started at the point for the reference simulation at 100 a, and both run for 10 a, with a time-step size of 1 a. The difference between them is that the first simulation has $\theta = 0$ for the θ for FSSA in Section 3.3, while the second simulation has $\theta = 1$. In other words, the first simulation runs without FSSA implemented, and the second simulation does have FSSA implemented.

For the results of this experiment, the velocity magnitude is not calculated, but rather v_x and v_z are separated. This is to separate the difference between ice flow in the x-direction and in the z-direction. For the earlier experiments, the goal was to investigate how large of a difference there was between Experiment 2 and Experiment 3. In this experiment the goal is instead to see *why* there is a difference. We are looking at differences in velocities, because velocities are calculated with the Stokes equations, and it is for the Stokes equations the FSSA is implemented. Thus, by looking at the velocities in the different directions separately, one can deduce wether there is a larger difference in how the velocities differ.

The Figures 4.6, 4.7, 4.8 and A.12 (the last one found in the Appendix) are all taken from 103 a. This year was chosen because it is the first year where there is a difference of ice thickness between $\theta = 0$ and $\theta = 1$, see 4.3.

From these figures, it is easy to see that at the limit of the glacier the

Diff	v_x	v_z	Thickness	Surface
101 a	0.0425	0.0079	0.0000	0.0000
102 a	0.0413	0.0075	0.0000	0.0029
103 a	0.0402	0.0071	0.0032	0.0063
104 a	0.0392	0.0069	0.0062	0.0088
105 a	0.0382	0.0066	0.0089	0.0114
106 a	0.0372	0.0064	0.0115	0.0142
107 a	0.0363	0.0062	0.0140	0.0165
108 a	0.0353	0.0060	0.0162	0.0184
109 a	0.0344	0.0059	0.0184	0.0201
110 a	0.0335	0.0058	0.0204	0.0226

Table 4.3: Table over the the absolute difference of value for the horizontal velocity, the vertical velocity, Ice Thickness and Surface Elevation between Experiment 4: $\theta = 0$ and Experiment 4: $\theta = 1$, rounded to four decimals.

horizontal velocity is lower for the $\theta = 1$ simulation, but higher for the vertical velocity for the $\theta = 1$ simulation, both compared to the $\theta = 0$ simulation.

In Table 4.3 the absolute value of the difference in values between the $\theta = 0$ simulation and the $\theta = 1$ simulation are shown for each time-step for this experiment. In the table, one can see that the difference between ice thickness is the greatest at the last time-step, since the different velocities for the two simulations at that point have made maximum impact for this simulation. Something else to notice as well is that for each time-step taken the difference in velocity between the $\theta = 0$ simulation and the $\theta = 1$ simulation is decreasing for both v_x and v_z. This could mean both of these value are converging towards a steady-state for velocities.

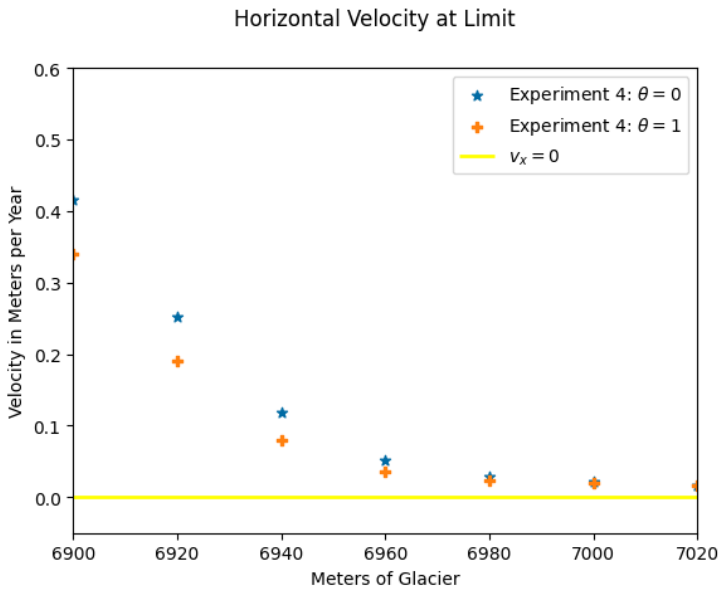

Figure 4.6: The velocities for v_x for Experiment 1: Reference Solution, Experiment 4: $\theta = 0$ and Experiment 4: $\theta = 1$ at the limit for 103 a.

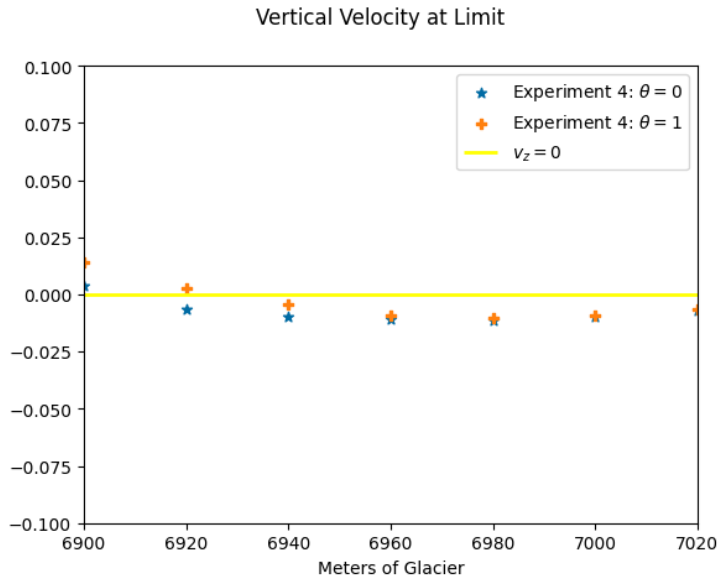

Figure 4.7: The velocities for v_z for Experiment 1: Reference Solution, Experiment 4: $\theta = 0$ and Experiment 4: $\theta = 1$ at the limit for 103 a.

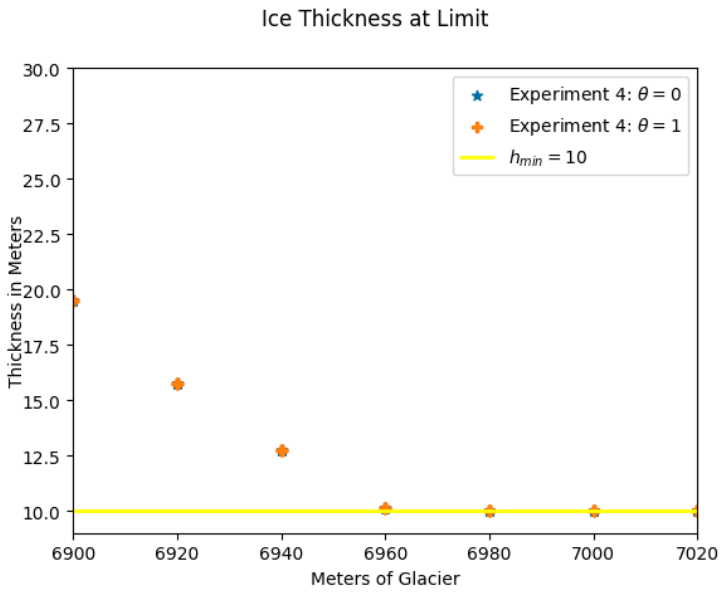

Figure 4.8: The ice thickness for Experiment 1: Reference Solution, Experiment 4: $\theta = 0$ and Experiment 4: $\theta = 1$ at the limit for 103 a.

5 Discussion

In Section 4 four experiments were carried out. Experiment 1, 2 and 3 were all for the time period 100-200 a, and Experiment 4 was for the time period 100-110 a. The time periods were chosen based on our interest in the retreating case of the glacier, and for the first century of the simulations the glacier was at occations advancing.

5.1 Analysis of Results

Experiment 1 was a reference solution, with a small time-step of 0.5 a without FSSA. Experiment 2 was a simulation with a large time-step of 5 a, also without FSSA. Experiment 3 was a simulation also with a large time-step of 5 a, with FSSA activated, $\theta = 1$ for the θ in Equation 3.34. This means that when the velocities were calculated by solving the Stokes equations, they were done so over an approximation of how the surface looked in the next time-step. Experiment 4 was two simulations with a time-step of 1 a, one without FSSA and the other with FSSA.

From our results we see that Experiment 3 was much more accurate than Experiment 2 compared to Experiment 1, both in velocity magnitude and surface elevation. Even though Experiment 3 was slower to run than Experiment 2, it is possible that we could take larger time steps, speeding up the simulation, and still be in an acceptable range of accuracy. This was shown in [2] for an advancing glacier, where the time-stepping could be increased by a factor of thirty or more and without loss of accuracy.

The surprise was that Experiment 2 showed significant error compared to the reference solution. This might be because of instability in Experiment 2, since the time-step was 5 a, which is quite large. The large differences in velocity between the experiments can be explained by that there is a large difference in the surface at these time-steps. The errors of both Experiment 2 and 3 decreases as time passes. This could possibly be explained by that the glacier is retreating. This means there's fewer amount of nodes where there can be an error, as the minimum thickness condition is met for all experiments on a larger part of the glacier the longer the simulations run. Graphing where the errors were the largest between Experiment 2 where $\theta = 0$ and Experiment 3 where $\theta = 1$ shows that the errors were larger for $\theta = 1$ at the limit of the glacier, but smaller at the middle of the glacier. This last find was however tainted by instability in Experiment 2.

For Experiment 4, a difference in the velocities was found between the two simulations. The consequence of this was a growing difference between the two simulations for the ice thickness and surface for each time-step. The difference for the velocities between the simulations with $\theta = 0$ and $\theta = 1$ did however get smaller for each time-step. This could be explained by the fact that the glacier is retreating, just as explained in the previous paragraph. Since the glacier is retreating, it is flatter in time-step $k + 1$ compared to time-step k.

The slower horizontal velocities for $\theta = 1$ can be explained by the FSSA not taking the minimum thickness condition into consideration. This means the slope of the glacier could become more flat, giving slower horizontal velocities. The higher speeds for the vertical velocities for $\theta = 1$ could then be a consequence of the slower horizontal velocities. Equation 2.47 says that the derivative

of the horizontal velocities and the vertical velocities must be zero:

$$\nabla \cdot \mathbf{u} = 0, \mathbf{x} \in \Omega. \tag{5.1}$$

The slower horizontal velocities gives a smaller change of velocity than for $\theta = 0$, making the vertical velocities have a larger change of velocity to compensate, to fulfill the condition given in 2.47.

Our hypothesis was that solving the Stokes equations using FSSA would give incorrect velocities at the limit, since it does not take the minimum thickness condition into consideration. This was confirmed by Experiment 4, as the velocities at the limit were different for $\theta = 1$ compared to $\theta = 0$. The difference in velocities was, however, minimized with each time-step. In simulating glaciers, we are mostly concerned with the evolution of the surface. This is because the surface evolution shows the mass loss of the glacier over time. The velocities of the glacier are only calculated to be used in calculations of the surface evolution for each time-step. The question if the different velocities calculated by using the FSSA causes a large difference in surface further down the line. Experiment 2 and 3 showed that the method using the FSSA was more exact, but no conclusions can be drawn since there is a large possibility Experiment 2 suffered from instability.

5.2 Importance of Simulating Glaciers

The reason for simulating ice sheets and glaciers is the expected rise in sea levels due to ice sheets and glaciers losing mass. The mass of glaciers consists of frozen water, and when they melt, the water end up in the oceans. As described in [1], mass loss is certain for the Greenland Ice Sheet and probable for the Antarctica Ice Sheet in the near future. Consequences for mass loss from these massive ice sheets, according to [18], include but are not limited to fresh water availability, severe flooding and coastal hazards for communities placed at low level elevation coastal zones, existing in both developed and developing countries. In order to take counter measures and mitigate the problems rising sea levels cause, the need to accurate model the mass loss of ice sheets and glaciers is crucial.

In this thesis, the FSSA method, just recently applied to ice sheet modelling by [2], is tested for accuracy in geometry and velocity for a retreating glacier. If the method is proven reliable for simulating glaciers, it could be applied to get more accurate estimates of future rising sea levels.

5.3 Further Development of Numeric Methods

The Antarctic Ice Sheet and the Greenland Ice Sheet are the largest on earth. As such, their contribution to rising sea levels are expected to be great if they continue to lose mass. At the same time, their size make them hard to model.

Today's simulations of ice sheet and glacier melt relies on solving the Stokes equations. These are computationally expensive, and thus require better and more effective numerical methods to reliably be solved with accuracy and speed. Faster models can also simulate ice sheets at a higher resolution, i.e. more nodes in the mesh, giving more reliable predictions. The development of new numerical methods for simulations needs to be tested and assured to be robust as well, as to not give inaccurate or false predictions.

In order to check that the simulation is robust, it is compared to a reference simulation, in this thesis represented by Experiment 1. Taking a smaller time-step is expected to give a more accurate solution, as those are not as prone to being affected by numerical instabilities.

The FSSA is a stabilization algorithm, implemented with the intention to be take larger time-steps without causing instability. It does this by solving the Stokes equations over an approximation of how the surface will look in the next time-step, by using Reynold's transport theorem, explained in Section 3.3.

In a previous article, [2], on using FSSA for ice sheet modeling, it was shown the FSSA increased the largest stable time-step by a factor of over 30, speeding up simulations without losing accuracy. The largest increase in time-stepping was seen for a semi-implicit Euler method. These experiments were done on an advancing glacier. The model with the FSSA was less exact at the limit of the glacier, but more exact in the middle of the glacier.

This experiment, using an explicit Euler method, showed that for the limit at the glacier, where the minimum thickness condition is applied, the error for the velocities are decreased with each time-step. This might be because of that the glacier is retreating. The results of the earlier experiments was most likely tainted by instability, but the conclusion that implementing the FSSA stabilize time-stepping with larger time-steps can still be drawn. Just as in [2], the larger errors of the surface were close to the limit, but it was more exact for in the middle. These findings in this thesis was most likely tainted by instability. As opposed to [2], these experiments were done on a retreating glacier.

The results from the earlier article, together with the results from this thesis, shows promise for using the FSSA as a stabilization algorithm. The results of this thesis also needs to be compared to results of using an explicit Euler method for the time-stepping instead.

5.4 Outlook

Moving forward, Experiment 4 should be repeated for a longer time duration. Keeping the time-step at 1 a is recommended to avoid instability for the non-FSSA simulation. The experiment should run for a longer time than 10 years to see if the velocities for the simulations with $\theta = 1$ and $\theta = 0$ keeps getting closer, and check if this eventually trivialize the difference in velocities between $\theta = 0$ and $\theta = 1$. This experiment can also check if the initial differences in velocities cause large differences in the surface of the glacier and ice thickness over a longer time-period than 10 years. Since it is the surface evolution that is most interesting when simulating glaciers, this last suggestion is of great interest. The error of the surface should be analyzed using these simulations in order to see if the error for the simulation with $\theta = 1$ is still lower in the middle but higher at the limit than the simulation using $\theta = 0$.

The results from Experiment 4 should also be compared to a reference simulation to see if simulations with $\theta = 1$ have a larger error than simulations with $\theta = 0$, rather than just conclude that there is a difference in velocities.

A fix for the larger error where the minimum thickness condition is applied is also needed to be developed for the FSSA. A suggestion is some kind of condition, that if the node for the approximated domain falls below $h_{\min}$, a special kind of calculation takes place. This could be accomplished by using an

if-statement. The if-condition checks wether $z_s + \tilde{\mathbf{u}} \cdot \hat{\mathbf{n}} < z_b + h_{\min}$ in the FSSA calculation. In that case, replace $z_s + \tilde{\mathbf{u}} \cdot \hat{\mathbf{n}}$ with $z_b + h_{\min}$.

One could also look at different values for θ, rather than just 0 and 1, and investigate how this changes the simulations with regards to a reference solution. Does a tweaking of the influence of θ by using a different number than $\theta = 1$ decrease or increase the error at the limit of the glacier?

Once enough is known about the FSSA for ice sheet modelling, one would need to compare this method with other numerical methods on real ice sheet environments. The Marine Ice Sheet Model Intercomparison Project (MISMIP) [19] aim to compare different numerical methods of the grounding line for ice sheets. The grounding line is the zone where a floating ice sheet meets the ocean. It would be interesting to see how the FSSA does in velocity and geometry compared to other numerical methods in simulating this complicated area of ice sheet modelling.

6 Conclusion

In this thesis, the free-surface stabilization algorithm (FSSA) was investigated with regards to the minimum thickness condition using Elmer/Ice. The FSSA uses Reynold's transport theorem to approximate the domain in the next time-step when solving the Stokes equations, thus mimicking a fully implicit time-stepping scheme. The minimum thickness condition is an algorithm that makes sure the ice thickness of the glacier does not go below some decided $h_{\min}$. The FSSA, when approximating the domain in the next time-step, does not take the minimum thickness condition into consideration. The method using FSSA was compared to a method with the same time-step, and a reference solution. The behaviour of the FSSA at the limit, where the minimum thickness condition becomes relevant, was also investigated. All simulations were done on a retreating glacier.

The main finding was that the method using the FSSA was more stable for the same time-step as the method without FSSA. The error of the surface for the method using FSSA was lower in the middle but higher at the limit of the glacier. Another finding was that different velocities were calculated for the FSSA at the limit compared to a method without the FSSA where the minimum thickness condition becomes relevant, for a small time-step. The differences in velocity for the FSSA for a small time-step, though, became smaller with each time-step.

These findings points towards the FSSA showing promise as a stabilization algorithm for ice sheet modelling, however more research is needed to prove the FSSA as a robust algorithm for improving glacier and ice sheet simulations.

Appendices

A Extra Graphs

A.1 Graphs on Experiment 1

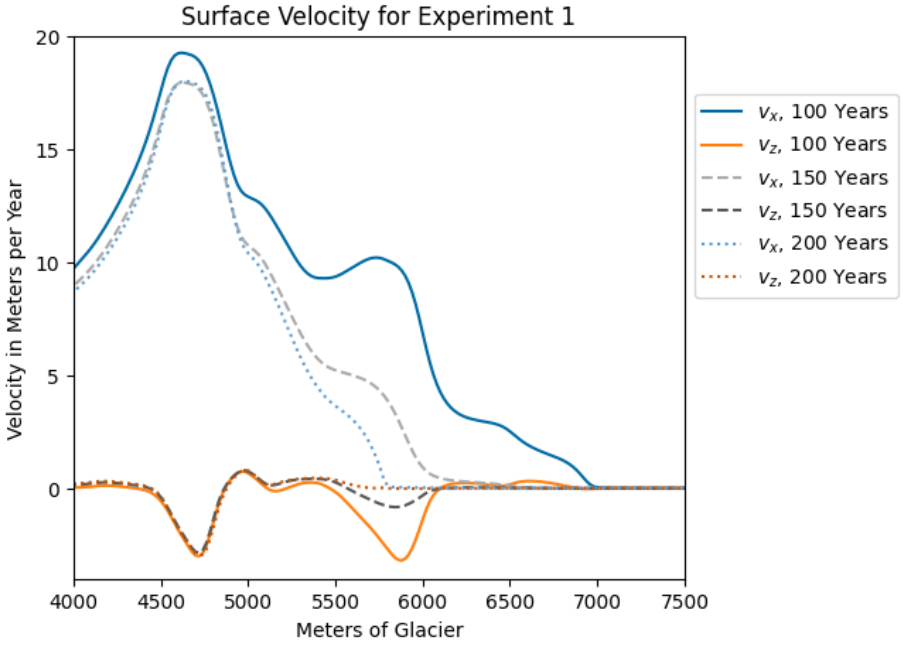

Figure A.1: Velocity in v_x and v_z for the surface of Experiment 1: Reference Solution.

A.2 Graphs on Experiment 2

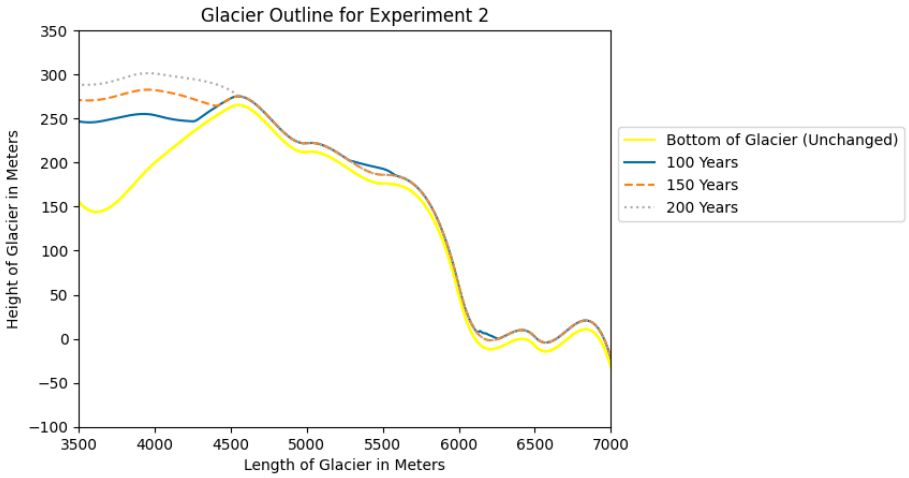

Figure A.2: The development of the glacier at the limit of Experiment 2: Larger Time-Step.

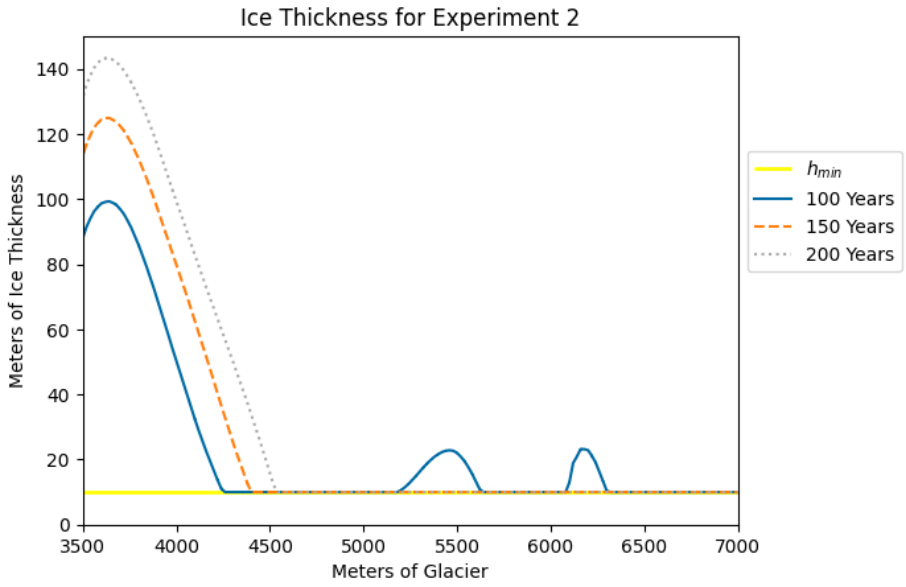

Figure A.3: Ice thickness at the limit of Experiment 2: Larger Time-Step.

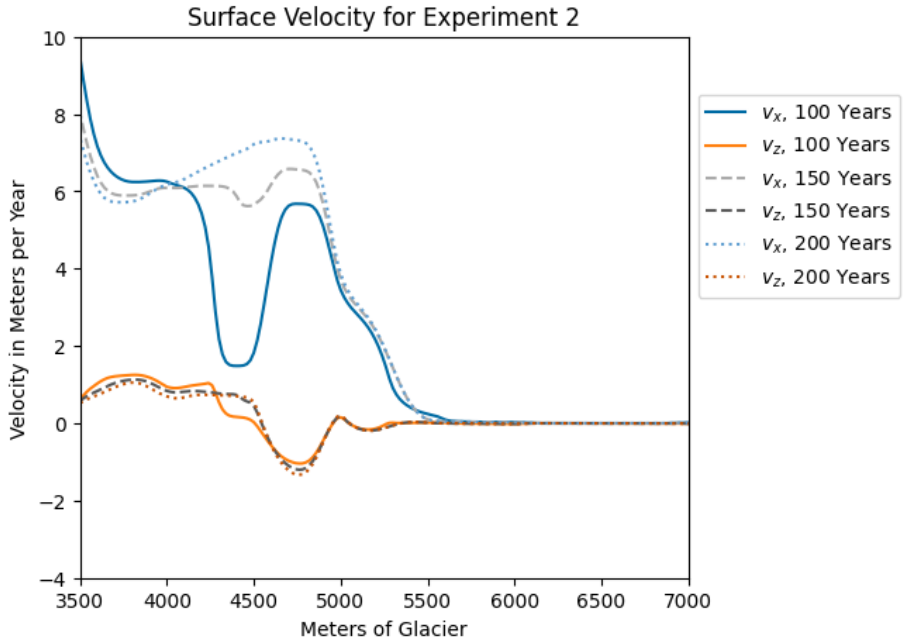

Figure A.4: Velocity in v_x and v_z for the surface of the glacier at the limit for Experiment 2: Larger Time-Step.

A.3 Graphs on Experiment 3

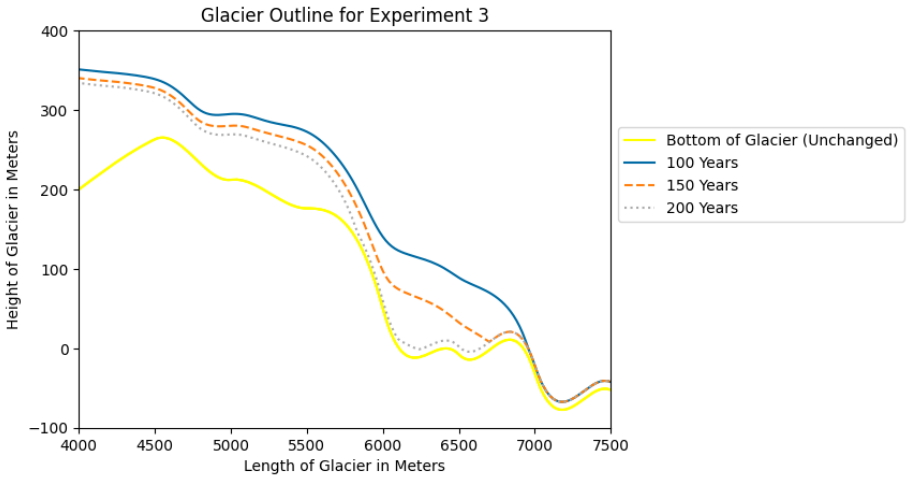

Figure A.5: The development of the glacier at the limit of Experiment 3: FSSA Implementation.

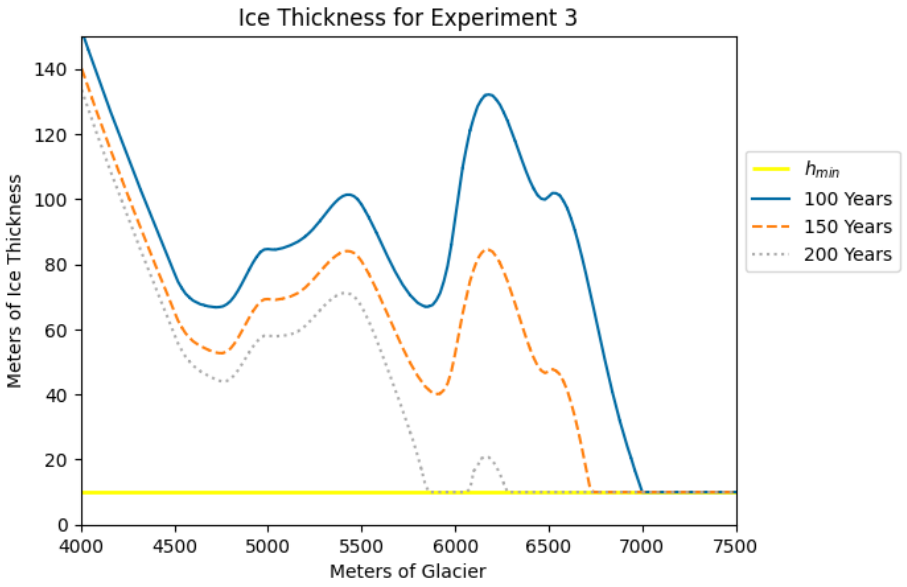

Figure A.6: Ice thickness at the limit of Experiment 3: FSSA Implementation.

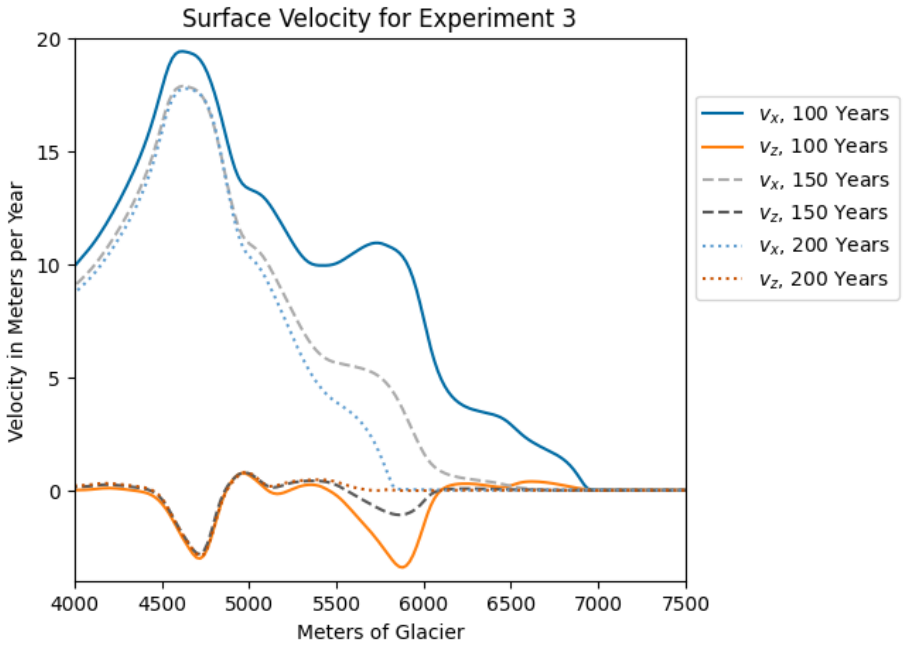

Figure A.7: Velocity in v_x and v_z for the surface of the glacier at the limit for Experiment 3: FSSA Implementation.

A.4 Graphs comparing Experiment 1 and Experiment 2

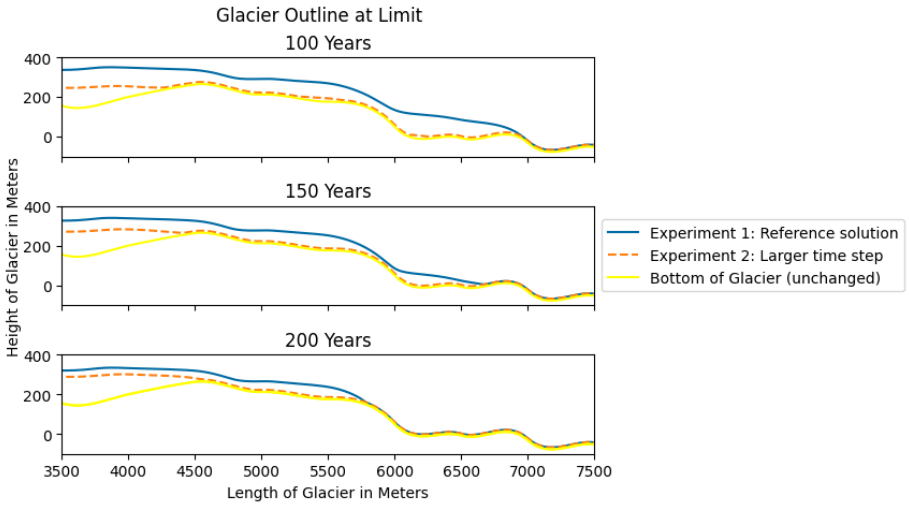

Figure A.8: The development of the glacier at the limit of Experiment 1: Reference Solution and Experiment 2: Larger Time-Step.

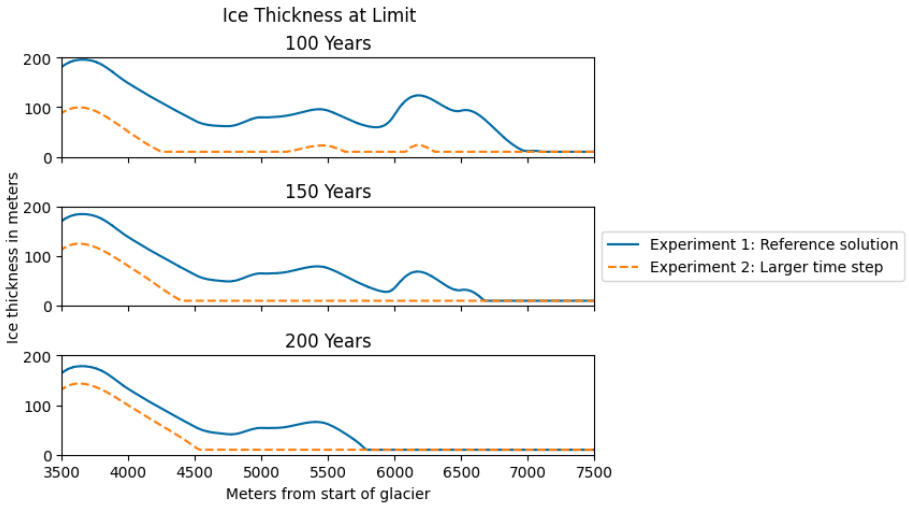

Figure A.9: Ice thickness at the limit of Experiment 1: Reference Solution and Experiment 2: Larger Time-Step.

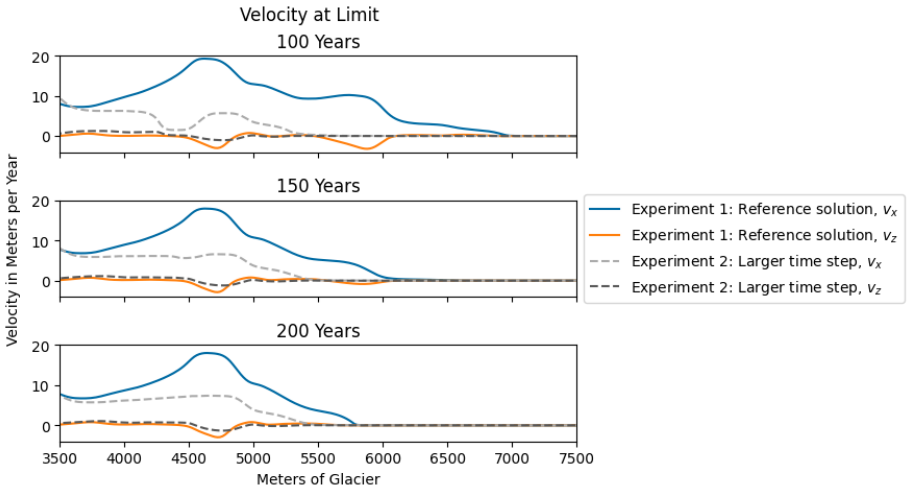

Figure A.10: Velocity in v_x and v_z for the surface of the glacier for Experiment 1: Reference Solution and Experiment 2: Larger Time-Step.

A.5 Graph comparing the outline of Experiment 1, 2 and 3

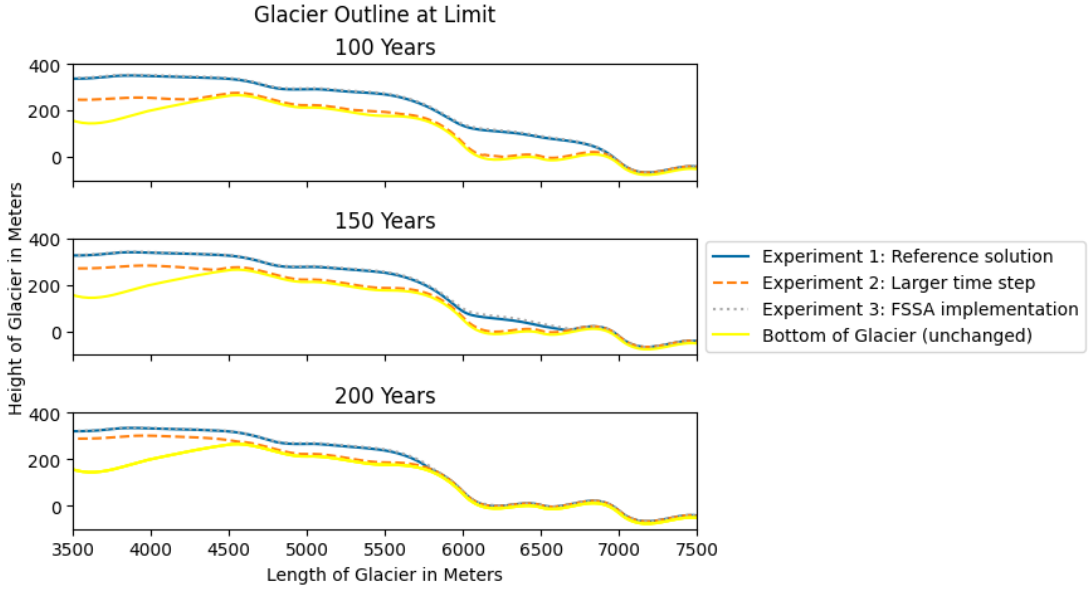

Figure A.11: The development of the glacier at the limit of the glacier for Experiment 1: Reference Solution, Experiment 2: Larger Time-Step and Experiment 3: FSSA Implementation.

A.6 Graphs for Experiment 4

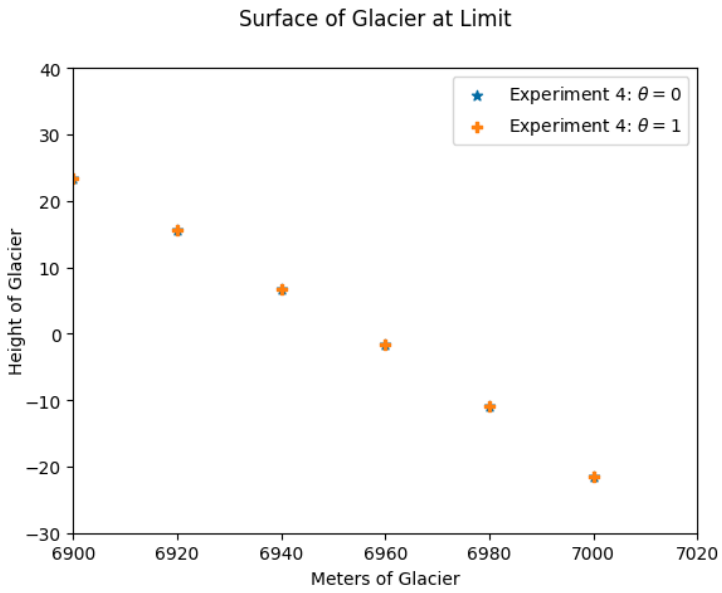

Figure A.12: The surface for Experiment 1: Reference Solution, Experiment 4: $\theta = 0$ and Experiment 4: $\theta = 1$ at the limit for 103 a.

References

[1] Valérie Masson-Delmotte et al. "Climate change 2021: the physical science basis". In: *Contribution of working group I to the sixth assessment report of the intergovernmental panel on climate change* 2 (2021).

[2] André Löfgren, Josefin Ahlkrona, and Christian Helanow. "Increasing stable time-step sizes of the free-surface problem arising in ice-sheet simulations". In: *Journal of Computational Physics: X* 16 (2022), p. 100114.

[3] Christian Helanow and Josefin Ahlkrona. "Stabilized equal low-order finite elements in ice sheet modeling–accuracy and robustness". In: *Computational Geosciences* 22 (2018), pp. 951–974.

[4] O Gagliardini et al. "Capabilities and performance of Elmer/Ice, a new-generation ice sheet model". In: *Geoscientific Model Development* 6.4 (2013), pp. 1299–1318.

[5] André Löfgren et al. "Increasing numerical stability of mountain valley glacier simulations: implementation and testing of free-surface stabilization in Elmer/Ice". In: *EGUsphere* 2023 (2023), pp. 1–22. DOI: 10.5194/egusphere-2023-1507. URL: https://egusphere.copernicus.org/preprints/2023/egusphere-2023-1507/.

[6] Mats G. Larson and Fredrik Bengzon. *The Finite Element Method: Theory, Implementation, and Applications*. Springer-Verlag, 2013. ISBN: 9783642332876.

[7] Qingkai Kong, Timmy Siauw, and Alexandre Bayen. *Python programming and numerical methods: A guide for engineers and scientists*. Academic Press, 2020.

[8] Walter Gander, Martin J Gander, and Felix Kwok. *Scientific computing-An introduction using Maple and MATLAB*. Vol. 11. Springer Science & Business, 2014.

[9] Arne Persson and Lars-Christer Böiers. *Analys i en variabel*. Third edition. Studentlitteratur, 2010. ISBN: 9789144067650.

[10] Wolfgang Arendt and Karsten Urban. *Partial Differential Equations: An Introduction to Analytical and Numerical Methods*. Trans. by James B. Kennedy. Berlin, Germany: Springer Spektrum, 2018. Chap. 1. ISBN: 9783031133787.

[11] FEniCS Project. *Marking subdomains of a mesh*. 2014. URL: https://fenicsproject.org/olddocs/dolfin/1.4.0/python/demo/documented/subdomains/python/documentation.html. (accessed: 08.05.2024).

[12] Ralf Greve and Heinz Blatter. *Dynamics of ice sheets and glaciers*. Springer Science & Business Media, 2009.

[13] John W Glen. "The creep of polycrystalline ice". In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 228.1175 (1955), pp. 519–538.

[14] John Frederick Nye. "The distribution of stress and velocity in glaciers and ice-sheets". In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 239.1216 (1957), pp. 113–133.

[15] Boris JP Kaus, Hans Mühlhaus, and Dave A May. "A stabilization algorithm for geodynamic numerical simulations with a free surface". In: *Physics of the Earth and Planetary Interiors* 181.1-2 (2010), pp. 12–20.

[16] Thomas Zwinger et al. “A full Stokes-flow thermo-mechanical model for firn and ice applied to the Gorshkov crater glacier, Kamchatka”. In: *Annals of Glaciology* 45 (2007), pp. 29–37.

[17] Thomas Zwinger (CSC) Samuel Cook (UNIL) and Olivier Gagliardini (IGE). *Elmer/Ice beginner's course, CSC Helsinki, nov. 6+7, 2023*. 2023. URL: `https://elmerice.elmerfem.org/wiki/doku.php?id=courses:2023:helsinki2023`. (accessed: 09.05.2024).

[18] Hans-Otto Pörtner et al. “The ocean and cryosphere in a changing climate”. In: *IPCC special report on the ocean and cryosphere in a changing climate* 1155 (2019).

[19] Frank Pattyn et al. “Results of the marine ice sheet model intercomparison project, MISMIP”. In: *The Cryosphere Discussions* 6.1 (2012), pp. 267–308.

Datalogi
www.math.su.se

Beräkningsmatematik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm