Ice Sheet Modeling: Accuracy of First-Order Stokes Model with Basal Sliding

Iståckemodellering: Noggrannhet hos första ordningens Stokes modell med basalskjutning

Eskil Jonsson
Ice Sheet Modeling: Accuracy of First-Order Stokes Model with Basal Sliding

Eskil Jonsson
Abstract

Ice Sheet Modeling: Accuracy of First-Order Stokes Model with Basal Sliding
Eskil Jonsson

Some climate models are still lacking features such as dynamical modelling of ice sheets due to their computational cost which results in poor accuracy and estimates of e.g. sea level rise. The need for low-cost high-order models initiated the development of the First-Order Stokes (or Blatter-Pattyn) model which retains much of the accuracy of the full-Stokes model but is also cost-effective. This model has proven accurate for ice sheets and glaciers with frozen bedrocks, or no-slip basal boundary conditions. However, experimental evidence seems to be lacking regarding its accuracy under sliding, or stress-free, bedrock conditions (ice-shelf conditions). Hence, it became of interest to investigate this.

Numerical experiments were set up by formulating the first-order Stokes equations as a variational finite element problem, followed by implementing them using the open-source FEniCS framework. Two types of geometries were used with both no-slip and slip basal boundary conditions. Specifically, experiments B and D from the Ice Sheet Model Intercomparison Project for Higher-Order ice sheet Models (ISMIP-HOM) were used to benchmark the model. Local model errors were investigated and a convergence analysis was performed for both experiments.

The results yielded an inherent model error of about 0.06% for ISMIP-HOM B and 0.006% for ISMIP-HOM D, mostly relating to the different types of geometries used. Errors in stress-free regions were greater and varied on the order of 1%. This was deemed fairly accurate, and probably enough justification to replace models such as the Shallow Shelf Approximation with the First-Order Stokes model in some regions. However, more rigorous tests with real-world geometries may be warranted.

Also noteworthy were inconsistent results in the vertical velocity under slippery conditions (ISMIP-HOM D) which could either be due to coding errors or an inherent problem with the decoupling of the horizontal and vertical velocities of the First-Order Stokes model. This should be further investigated.

Keywords: Ice sheet modelling, First-Order Stokes model, basal sliding, Stokes equations

Degree Project D in Earth Science, 1GV013, 15 credits
Supervisor: Gong Cheng
Department of Earth Sciences, Uppsala University, Villavägen 16, SE-752 36 Uppsala (www.geo.uu.se)

ISSN 1650-6553, Examensarbete vid Institutionen för geovetenskaper, No. 443, 2018

The whole document is available at www.diva-portal.org
Populärvetenskaplig sammanfattning

Istäckemodellering: Noggrannhet hos första ordningens Stokes modell med basalskjutning

Eskil Jonsson

Vissa klimatmodeller saknar fortfarande funktioner så som dynamisk modellering av istäcken på grund av dess höga beräkningskostnad, vilket resulterar i låg noggrannhet och uppskattningar av t.ex. havsnivåhöjning. Behovet av enkla modeller med hög noggrannhet satte igång utvecklingen av den s.k. Första Ordningens Stokes (eller Blatter-Pattyn) modellen. Denna modell behåller mycket av noggrannheten i den mer exakta full-Stokes-modellen men är också väldigt kostnadseffektiv. Denna modell har visat sig vara noggrann för istäcken och glaciärer med frusna berggrunder eller s.k. no-slip randvillkor. Experimenterella bevis tycks dock saknas med avseende på dess noggrannhet under glidning, eller stressfria, berggrundsförhållanden (t.ex. vid ishyllor). Därför ville vi undersöka detta.

Numeriska experiment upprättades genom att formulera Blatter-Pattyn ekvationer som ett variationsproblem (via finita elementmetoden), följt av att implementera dem med hjälp av den öppna källkoden FEniCS. Två typer av geometrier användes med både glidande och stressfria basala randvillkor. Specifikt användes experiment B och D från Ice Sheet Model Intercomparison Project for Higher-Order ice sheet Models (ISMIP-HOM) för att testa modellen. Lokala fel undersöktes och en konvergensanalys utfördes för båda experimenten.

Resultaten gav ett modellfel på ca 0,06 % för ISMIP-HOM B och 0,006 % för ISMIP-HOM D, vilka var mest relaterade till de olika typerna av geometrier som användes. Fel i stressfria regioner var större och varierade i storleksordningen 1 %. Detta ansågs vara ganska noggrant och sannolikt tillräckligt för att ersätta modeller så som Shallow Shelf Approximationen med Blatter-Pattyn-modellen i vissa regioner. Dock krävs mer noggranna tester med mer verkliga geometrier för att dra konkreta slutsatser.

Också anmärkningsvärt var motsägande resultat i den vertikala hastigheten under glidande förhållanden (ISMIP-HOM D) som antingen kan ha berott på kodningsfel eller ett modellproblem som härstammar utifrån särkopplingen mellan den horizontella- och den vertikala hastigheten i Blatter-Pattyn-modellen. Detta bör undersökas vidare.

Keywords: Istäckemodellering, Blatter-Pattyn approximationen, basalgidning, Stokes-ekvationerna

Examensarbete D i Geovetenskap, 1GV013, 15 hp
Handledare: Gong Cheng
Institutionen för geovetenskaper, Uppsala universitet, Villavägen 16, 752 36 Uppsala (www.geo.uu.se)

ISSN 1650-6553, Examensarbete vid Institutionen för geovetenskaper, No. 443, 2018

Hela publikationen finns tillgänglig på www.diva-portal.org
# Table of Contents

1 Introduction .............................................................................................................................. 1

2 Background .............................................................................................................................. 3  
   2.1 Ice Sheet Modelling – The Governing Equations ............................................................... 3  
   2.2 The First-Order Stokes Model ........................................................................................... 3  
   2.3 Boundary Conditions ......................................................................................................... 7  
   2.4 Weak Formulation of the FOS Problem .......................................................................... 10

3 Methodology ............................................................................................................................... 13

4 Results and Discussion ............................................................................................................ 15  
   4.1 ISMIP-HOM B Test and Convergence Analysis .............................................................. 15  
   4.2 ISMIP-HOM D Test ............................................................................................................ 17

5 Conclusions ............................................................................................................................. 20

Acknowledgements ........................................................................................................................ 21

References ....................................................................................................................................... 22

Internet Resources .......................................................................................................................... 23

Appendices ..................................................................................................................................... 24

A Python Code ................................................................................................................................. 24  
   A.1 Initialization, Domain, and Physical Parameters ............................................................... 24  
   A.2 Ice Sheet Geometry, Computational Mesh, and Boundary Conditions ....................... 25  
   A.3 Strain Rate, Viscosity, and Source Terms ........................................................................ 27  
   A.4 Defining the Weak Formulation and Solving the Nonlinear System ............................... 28

Glossary ........................................................................................................................................ 29

Acronyms ....................................................................................................................................... 30

Symbols ......................................................................................................................................... 31

Software ........................................................................................................................................ 33
Preface

Although the target audience is master-level Earth Science students some sections require a deeper understanding of computational science, specifically regarding the finite element method. These sections may be glossed over since they have little bearing on the initial glacier theory and the final results.

The intention was to eventually cover more realistic 3D cases in this project as well, however these were very prone to errors and took too long to implement, even with pre-built packages like CSLVR and VarGlaS. These would’ve been more appropriate if this was a 30 credit project. Because of this, there’s ample room for improvement in future studies.
1 Introduction

The Greenland and Antarctic ice sheets continue to lose mass at an accelerating rate—resulting in significant contributions to global sea level rise (Vaughan et al., 2013). Regardless, it was only until recently that climate models such as CMIP5 have been able to resolve ice sheet dynamics whereas other models such as CESM are still lacking this feature (Slangen & Lenaerts, 2016). This exclusion of ice dynamics has been due to a number of reasons, such as the inability of simplified, low-order, ice sheet models—typically the Shallow Ice Approximation (SIA) or the Shallow Shelf Approximation (SSA)—to accurately replicate the observed dynamics of ice sheets. Alternatively, there’s the higher-order so-called full-Stokes models (see Figure 1 for a physical comparison). However, these models take into account all the physical stresses acting on the ice and are thus too computationally expensive to apply to large regions (Tezaur et al., 2015). Hence, the demand of high-order low-cost models warranted the development of more sophisticated approximations.

One of these developments was initiated by Blatter, 1995, and later refined by Pattyn, 2003, to what is now known as the Blatter-Pattyn (B-P), or the First Order Stokes (FOS), model (both names are used interchangeably). This model retains much of the simplicity of the lower-order models, but some of the important higher-order stress terms of the Stokes equations are still accounted for. In addition, the model decouples the horizontal and vertical velocity dimensions, effectively turning the complexity of the three-dimensional (3D) flow-field of the ice sheet into a horizontal flow-field problem. The vertical velocity profile may then be computed from the horizontal field (Ahlkrona, 2016). This development has allowed more practical long-term simulations to be employed over larger regions. Regardless, although the model has proved accurate for ice sheets with frozen bed topography (i.e. under no-slip conditions), there appears to be a lack of experimental evidence regarding its accuracy under sliding conditions, such as those prevailing in ice streams and ice shelves. Hence, it is of our interest to investigate this.

Figure 1: The primary stresses acting on a slab of ice on an incline for a) the Stokes model and b) the SIA. Adapted from http://www.antarcticglaciers.org.
Our model of the physics is only as good as our numerical method however. The Finite Difference Method (FDM), which approximates the derivatives using truncated Taylor series expansions, remain the go-to solution for many model problems. Although the typical explicit schemes are simple and straightforward to implement they often come with stricter stability conditions and may introduce more significant truncation errors. Additionally, some regions of ice sheets, particularly near the margins, may exhibit rather steep velocity gradients due to local topography compared to the normally shallow gradients further inland. FDMs typically do not provide efficient adaptation of the mesh to account for these variations which introduces more discretization errors, thus requiring a higher resolution mesh to be used (see Figure 2a). Because of this we typically employ the Finite Element Method (FEM) in computational ice sheet dynamics and will do so in this paper. As seen in Figure 2b, due to its adaptivity, the FEM typically has a much smaller discretization error, despite much fewer computational cells (elements).

Figure 2: Discretization meshes for a) the FDM and b) the FEM for an arbitrary domain Ω. Discretization errors are highlighted in red. Adapted from Ferreira et al., 2015.

Despite the many benefits of the FEM, it is also fairly complicated to fully implement, even for simple geometries. As such, we rely on pre-built software packages, in this case the open-source FEniCS framework by Alnæs et al., 2015. The code for setting up the finite element problem is written in the Python programming language which is then exported to and solved in FEniCS. This is done via Linux or an equivalent virtual Linux machine such as Docker containers for Windows (in our case, the latter).

We primarily focus on 2D cases in this paper, as this is sufficient to establish the accuracy of the FOS model. More realistic 3D cases were intended, but there was insufficient time to cover them in this 15 credit thesis.
2 Background

2.1 Ice Sheet Modelling – The Governing Equations

Since glacier ice is polycrystalline it moves largely due to internal deformation under the force of gravity. As such, it behaves as a plastically deforming solid. Most, if not all, fluids or plastically deforming solids can be modelled by the Navier-Stokes equations, which are a set of non-linear hyperbolic Partial Differential Equations (PDEs) that dictate the momentum balance. Due to their non-linearity, which give rise to e.g. turbulent eddies, they’re either extremely difficult- or impractically expensive to solve.

Ice flow is highly viscous, slow-moving, and laminar however (low Reynolds number), which allows for a number of approximations to be made to the Navier-Stokes equations. These approximations, where advection and acceleration terms have been neglected, results in the linearized Navier-Stokes, or simply the Stokes equations which, in Cartesian coordinates, takes the form

\[
\nabla : \sigma = \rho g \\
\n\nabla \cdot \mathbf{u} = 0,
\]

where \(\sigma\) is the Cauchy stress tensor, \(\mathbf{u} = (u, v, w)\) the velocity field, \(\rho\) the ice density, and \(g = (0, 0, g)\) the acceleration due to gravity. The colon operator (\(\cdot\)) denotes the inner product for tensors—which in this case is effectively just the vector consisting of the dot product between \(\nabla\) and each row of \(\sigma\). In essence, the divergence of the stress is balanced by an applied force, in this case gravity (Batchelor, 2000). Equation (2) is simply the continuity equation, where we have assumed the ice to be incompressible (\(\rho = \text{constant}\)). This is the steady-state Stokes system.

2.2 The First-Order Stokes Model

Following Pattyn, 2003, we assume hydrostatic balance, i.e. \(\partial_z \sigma_{zz} \approx \rho g\), which reduces equation (1) to

\[
\partial_z \sigma_{xx} + \partial_y \sigma_{xy} + \partial_x \sigma_{xz} = 0 \\
\partial_z \sigma_{yx} + \partial_y \sigma_{yy} + \partial_x \sigma_{yz} = 0 \\
\partial_z \sigma_{zz} = \rho g.
\]

This implies that surface and basal stresses (Figure 1) do not vary in the lateral and longitudinal directions. Although, intuitively, these terms would change significantly whenever there’s a transition in the flow regime (e.g. from frozen to sliding basal conditions) it turns out that these terms are still negligible
compared to the other normal and shear stresses (Pattyn, 2000).

For incompressible materials, such as ice, deformations depend on the stress deviations rather than the total stress, particularly deviations from hydrostatic pressure (Cuffey & Paterson, 2010). Hence, we split the Cauchy stress tensor $\sigma$ into a hydrostatic stress component $\sigma_{\text{hyd}}$ and a deviatoric stress component $\sigma'$

$$\sigma = \sigma' + \sigma_{\text{hyd}},$$

where

$$\sigma_{\text{hyd}} = \begin{bmatrix} \sigma_{\text{hyd}} & 0 & 0 \\ 0 & \sigma_{\text{hyd}} & 0 \\ 0 & 0 & \sigma_{\text{hyd}} \end{bmatrix}.$$  (7)

Hydrostatic stress is just the average of the three diagonal (normal) components of the Cauchy stress tensor, i.e.

$$\sigma_{\text{hyd}} = \frac{\sigma_{xx} + \sigma_{yy} + \sigma_{zz}}{3} = \frac{1}{3} \sigma_{kk},$$  (8)

where the Einstein summation convention has been used. Hence, we can express the deviatoric stress with

$$\sigma_{ij} = \sigma'_{ij} + \frac{1}{3} \delta_{ij} \sigma_{kk},$$  (9)

where $\delta_{ij}$ is the Kronecker delta. From this, we get the following expressions for $\sigma_{xx}$ and $\sigma_{yy}$

$$\sigma_{xx} = \frac{3}{2} \sigma'_{xx} + \frac{1}{2} (\sigma_{yy} + \sigma_{zz})$$  (10)

$$\sigma_{yy} = \frac{3}{2} \sigma'_{yy} + \frac{1}{2} (\sigma_{xx} + \sigma_{zz}),$$  (11)

whereas for $i \neq j$ we simply have that $\sigma_{ij} = \sigma'_{ij}$. Inserting (11) into (10) and repeating the same procedure for $\sigma_{yy}$ we then obtain the following expressions

$$\sigma_{xx} = 2\sigma'_{xx} + \sigma'_{yy} + \sigma_{zz}$$  (12)

$$\sigma_{yy} = 2\sigma'_{yy} + \sigma'_{xx} + \sigma_{zz}$$  (13)

$$\sigma_{ij} = \sigma'_{ij}, \quad i \neq j.$$  (14)
By integrating (5) from a depth \( z \) to the surface \( s(x, y) \), neglecting atmospheric pressure, we also obtain an expression for \( \sigma_{zz} \)

\[
\sigma_{zz} = \rho g (s - z).
\] (15)

Finally, inserting (12-15) into (3-4) we obtain

\[
\begin{align*}
\partial_x (2\sigma'_{xx} + \sigma'_{yy}) + \partial_y \sigma'_{xy} + \partial_z \sigma'_{xz} &= \rho g \partial_x s, \\
\partial_y (2\sigma'_{yy} + \sigma'_{xx}) + \partial_x \sigma'_{xy} + \partial_z \sigma'_{yz} &= \rho g \partial_y s
\end{align*}
\] (16)

The strain rate tensor \( \dot{\varepsilon} \), which describes the rate of normal and shear deformation, is defined by the velocity gradients as follows

\[
\dot{\varepsilon}_{ij} = \frac{1}{2} \left( \partial_j u_i + \partial_i u_j \right),
\] (18)

where \( \partial_j \) refers to the local derivative with respect to the \( j \)-th component of \( x = (x, y, z) \) and \( u_i \) refers to the \( i \)-th component of \( u \). So, for example, we have that \( \dot{\varepsilon}_{xy} = \frac{1}{2} (\partial_y u + \partial_x v) \).

Assuming that the horizontal gradients of vertical velocity are negligible \((\partial_x w, \partial_y w \ll \partial_z u, \partial_z v)\), which is accurate for most of the ice sheet domain, we can rewrite (18) as

\[
\begin{pmatrix}
\dot{\varepsilon}_{xx} & \dot{\varepsilon}_{xy} & \dot{\varepsilon}_{xz} \\
\dot{\varepsilon}_{yx} & \dot{\varepsilon}_{yy} & \dot{\varepsilon}_{yz} \\
\dot{\varepsilon}_{zx} & \dot{\varepsilon}_{zy} & \dot{\varepsilon}_{zz}
\end{pmatrix} =
\begin{pmatrix}
\partial_x u & \frac{1}{2} (\partial_y u + \partial_z v) & \frac{1}{2} \partial_z u \\
\frac{1}{2} (\partial_y u + \partial_z v) & \partial_y v & \frac{1}{2} \partial_z v \\
\frac{1}{2} \partial_z u & \frac{1}{2} \partial_z v & \partial_z w
\end{pmatrix}.
\] (19)

The fractional rate of volume change is given by \( \dot{\varepsilon}_I = \dot{\varepsilon}_{xx} + \dot{\varepsilon}_{yy} + \dot{\varepsilon}_{zz} \) and is known as the first invariant of \( \dot{\varepsilon} \). The second invariant \( \dot{\varepsilon}_E \) of \( \dot{\varepsilon} \), or the effective strain rate, is essentially just the total magnitude of the tensor (the sum of its components), which is analogous to the length of a vector. Its definition in the literature may vary by a constant factor of 2 (Cuffey & Paterson, 2010; Dukowicz et al., 2010). Note that since the strain rate tensor is symmetric (see 19) the off-diagonal terms have been combined to eliminate the factor of \( \frac{1}{2} \) for those terms

\[
\dot{\varepsilon}_E^2 = \frac{1}{2} (\dot{\varepsilon}_{xx}^2 + \dot{\varepsilon}_{yy}^2 + \dot{\varepsilon}_{zz}^2) + \dot{\varepsilon}_{xx}^2 + \dot{\varepsilon}_{xy}^2 + \dot{\varepsilon}_{yz}^2.
\] (20)

By rearranging the continuity equation (2) as \( \partial_z w = -(\partial_x u + \partial_y v) \) and inserting this into (20) we get

\[
\dot{\varepsilon}_E^2 = \dot{\varepsilon}_{xx}^2 + \dot{\varepsilon}_{yy}^2 + \dot{\varepsilon}_{xx} \dot{\varepsilon}_{yy} + \dot{\varepsilon}_{xy}^2 + \dot{\varepsilon}_{xz}^2 + \dot{\varepsilon}_{yz}^2.
\] (21)
Laboratory experiments have shown that the rheology of glacier ice follows a power-law relation between the strain rate and the shear stress known as Glen’s flow law

\[ \sigma'_{ij} = 2\eta \varepsilon_{ij}. \]  

(22)

This is a constitutive relation, which links stress to deformation (Cuffey & Paterson, 2010; Dukowicz et al., 2010). Here, \( \eta \) is the effective viscosity, given by

\[ \eta = \frac{1}{2} A(\theta^*)^{-\frac{1}{n}} (\dot{\varepsilon}_E + \dot{\varepsilon}_0)^{\frac{1-n}{n}}, \]  

(23)

where \( n \) is the Glen parameter, usually taken to be \( n = 3 \). \( A \) is the rate factor which depends on \( \theta^* \)—the temperature corrected for pressure melting. Since the viscosity also depends on the effective stress glacial ice mostly behaves as a non-Newtonian fluid. Essentially, as the deviatoric stress increases, the ice softens (Pattyn, 2003; Cuffey & Paterson, 2010). Lastly, \( \dot{\varepsilon}_0 \) is the critical shear rate which is just a small number to prevent singularities in the shear rate, particularly at the ice divide under no-slip boundary conditions (BCs).

The rate factor (also known as creep factor or flow parameter) takes the following form

\[ A(\theta^*) = A_0 \exp \left( -\frac{Q}{R\theta^*} \right), \]  

(24)

where \( Q \) is called the activation energy and \( R \) is the universal gas constant. The general physical and chemical properties of the rate factor are still not fully understood. However, experiments show that ice plasticity varies greatly depending on \( A \), and good accuracy of the temperature field essential to predict deformation rates (Cuffey & Paterson, 2010).

Now combining (22) with (16-17), we obtain

\[ \partial_x (2\eta[\dot{\varepsilon}_{xx} + \dot{\varepsilon}_{yy}]) + \partial_y (2\eta[\dot{\varepsilon}_{xy}]) + \partial_z (2\eta[\dot{\varepsilon}_{xz}]) = \rho g \partial_x s \]  

(25)

\[ \partial_x (2\eta[\dot{\varepsilon}_{xy}]) + \partial_y (2\eta[2\dot{\varepsilon}_{yy} + \dot{\varepsilon}_{xx}]) + \partial_z (2\eta[\dot{\varepsilon}_{yz}]) = \rho g \partial_y s. \]  

(26)

It is convenient to define a new tensor for the B-P approximation from this system as follows

\[ \dot{\varepsilon}_{BP} = \begin{pmatrix} 2\dot{\varepsilon}_{xx} + \dot{\varepsilon}_{yy} & \dot{\varepsilon}_{xy} & \dot{\varepsilon}_{xz} \\ \dot{\varepsilon}_{xy} & \dot{\varepsilon}_{xx} + 2\dot{\varepsilon}_{yy} & \dot{\varepsilon}_{yz} \end{pmatrix}. \]  

(27)

With this, we can rewrite (25-26) as
\( \nabla : (2\eta \dot{e}_{BP}) = \rho g \nabla s. \) \hspace{1cm} (28)

This is the governing system for the horizontal motion. Meanwhile, the vertical velocity \( w(z) \) is simply obtained by integrating the continuity equation (2) from the base \( b(x, y) \) to the height \( z \)

\[
w(z) - w(b) = - \int_b^z (\partial_x u + \partial_y v) \, dz.
\] \hspace{1cm} (29)

To account for temperature variations in the ice, an additional equation, the thermodynamic equation, can be added to the Stokes system (eqs. 1,2) or the B-P system (28)

\[
\rho c_p \frac{d\theta}{dt} = k \nabla^2 \theta + \Phi,
\] \hspace{1cm} (30)

where the specific heat capacity and thermal conductivity of ice, \( c_p \) and \( k \) respectively, have been assumed constant. Additionally, \( \Phi \) is the frictional heating due to internal deformation, which is directly proportional to the effective strain rate and also the effective stress \( \sigma_E \) which, as in (20), is the second invariant of the stress tensor (Cuffey & Paterson, 2010)

\[
\Phi = 2\dot{\varepsilon}_E \sigma_E, \quad \sigma_E^2 = \frac{1}{2} (\sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2) + \sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2.
\] \hspace{1cm} (31)

For this project we will assume isothermal ice sheets with the fixed pressure melting temperature \( \theta^* \). Hence, equation (30) simplifies to \( \Phi = 0 \), and is thus no longer necessary.

2.3 Boundary Conditions

We assume that the ice sheet is only bounded by two surfaces—the interface between the atmosphere and the ice sheet, and the interface between the bedrock and the ice sheet. The BCs of these surfaces completes the Stokes, as well as the B-P problem. These surfaces are also assumed to have shallow slopes, implying that \( \partial_x s, \partial_y s, \partial_x b, \partial_y b \ll 1 \).

In accordance with Dukowicz et al., 2010 we define two outward-pointing unit normal vectors, \( n^{(s)} \) and \( n^{(b)} \), for the upper and lower surfaces, \( s \) and \( b \) respectively. These can be simplified by neglecting 2nd-order terms, which yields

\[
n^{(s)} = \left( n_x^{(s)}, n_y^{(s)}, n_z^{(s)} \right)^T = \frac{(-\partial_x s, -\partial_y s, 1)^T}{\sqrt{1^2 + (\partial_x s)^2 + (\partial_y s)^2}}
\] \hspace{1cm} (32)

\[
= (-\partial_x s, -\partial_y s, 1)^T + \mathcal{O}(\zeta^2),
\] \hspace{1cm} (33)
and

\[
\mathbf{n}^{(b)} = (n_x^{(b)}, n_y^{(b)}, n_z^{(b)})^T = \frac{(\partial_x b, \partial_y b, -1)^T}{\sqrt{1^2 + (\partial_x b)^2 + (\partial_y b)^2}}
\]

\[
= (\partial_x b, \partial_y b, -1)^T + \mathcal{O}(\zeta^2),
\]

(34)

where \(\zeta\) is the aspect ratio of the ice sheet, i.e. the ratio between its characteristic height \([H]\) and length \([L]\) scales (Blatter, 1995)

\[
\zeta = \frac{[H]}{[L]}.
\]

Before implementing the boundary conditions it helps to refresh on some vector algebra. We can decompose any vector into a normal and tangential component, i.e. \(\mathbf{u} = \mathbf{u}^\perp + \mathbf{u}^\parallel\) such that \(\mathbf{u}^\perp \cdot \mathbf{u}^\parallel = 0\), or simply \(u_i^\perp u_i^\parallel = 0\) in the Einstein summation convention. Additionally, we know that \(\mathbf{u}^\perp = (\mathbf{u} \cdot \mathbf{n}) \mathbf{n}\) (or \(u_i^\perp = (u_j n_j) n_i\)), where \(\mathbf{n}\) is a unit normal vector to the surface, in this case either \(\mathbf{n}^{(s)}\) or \(\mathbf{n}^{(b)}\). It then follows that the parallel component is just given by \(\mathbf{u}^\parallel = \mathbf{u} - \mathbf{u}^\perp = \mathbf{u} - (\mathbf{u} \cdot \mathbf{n}) \mathbf{n}\).

The first requirement for the basal BC is that the ice sheet does not penetrate the bedrock. This implies that the normal component of the velocity at the bed is zero, i.e.

\[
\mathbf{u}^\perp = 0 \quad \text{or} \quad \mathbf{u} \cdot \mathbf{n}^{(b)} = 0.
\]

(37)

It then follows that the basal velocity is tangential to the bedrock surface, \(\mathbf{u} = \mathbf{u}^\parallel\). Additionally, using (34), this condition may be written as

\[
w = u\partial_x b + v\partial_y b.
\]

(38)

To account for sliding conditions, Dukowicz et al., 2010 and others often assume a linear relationship between the surface friction (or the normal components of the shear stress) and the basal velocity, where \(\beta \geq 0\) is the drag factor

\[
f^\parallel = \left(\sigma : \mathbf{n}^{(b)}\right)^\parallel = -\beta \mathbf{u}^\parallel.
\]

(39)

Although observations support a non-linear relationship, such as Weertman- or Lliboutry sliding at the local scale, we will stick with the linear relationship which is more appropriate at the regional scale (Fowler, 2010). To make physical sense of this boundary condition, consider an infinitesimal slab of ice where the stress \(\sigma\) acts (Figure 3). This stress tensor is made up of three so-called traction vectors, \(\sigma_x, \sigma_y, \sigma_z\).
\(\sigma_y\) and \(\sigma_z\), one accounting for the force vector acting on each surface axis.

\[ \text{Figure 3: Illustration of the traction vectors acting on an infinitesimal slab of ice.} \]

Assuming our coordinate system is initially aligned with the surface, so that \(n = (0, 0, 1)^T\), we have that the \(x\)-component of \(\sigma : n\) is given by \(\sigma_x \cdot n = \sigma_{xz}\). Doing the same analysis for the other faces and taking the parallel component, it then follows that \((\sigma : n) \parallel = (\sigma_{xz}, \sigma_{yz}, 0)^T\). Since our BC is \((\sigma : n) \parallel = -\beta u\) we then have that \(\sigma_{xz} = -\beta u\). At first glance, this result may not make much sense.

Now consider two infinitesimal ice slabs (Figure 4). The bed topography tends to force these volumes to displace upwards or downwards relative to each other. This causes, in this case, vertical shearing \((\sigma_{xz})\) between the slabs. This shearing is then what generates, or corresponds to, the horizontal motion \(-\beta u\).

A more intuitive interpretation follows from recognizing that the stress tensor is symmetric (due to conservation of angular momentum). This implies that \(\sigma_{zx} = \sigma_{xz} = -\beta u\), i.e. that the horizontal velocity is governed by the shearing stress acting at the base, or the frictional force.

\[ \text{Figure 4: Illustration of how the vertical shearing between infinitesimal slabs of ice due to changes in topography generates horizontal motion.} \]
If the material substance was completely rigid, such that the infinitesimal volumes could not displace up or down, then \( \beta \to \infty \) which would imply that \( u \to 0 \), i.e. no horizontal motion due to plasticity could occur because no deformation of the material is possible.

Note that the above analysis only applies initially for an infinitesimal displacement. After a significant displacement, the ice surface is no longer aligned with the coordinate system, and the expression \((\sigma : n)^\parallel = -\beta u\) becomes less obvious. Regardless, the above should provide a decent physical interpretation of the term.

Proceeding with simplifying the BC; recalling from (6) that \( \sigma = \sigma' + \sigma_{\text{hyd}} \) where \( \sigma_{\text{hyd}} \equiv \frac{1}{3} \delta_{ij} \sigma_{kk} \), we observe that the parallel component of the hydrostatic stress is zero, since

\[
\left( \sigma_{\text{hyd}} \cdot n_j^{(b)} \right)^\parallel = \left( \frac{1}{3} \delta_{ij} \sigma_{kk} n_j^{(b)} \right)^\parallel = \frac{1}{3} \delta_{ij} \sigma_{kk} \left( n_j^{(b)} \right)^\parallel = 0. \tag{40}
\]

Hence, we can write (39) solely in terms of the deviatoric stress

\[
(\sigma' : n^{(b)})^\parallel = -\beta u^\parallel. \tag{41}
\]

Since we neglected air pressure earlier, the upper surface, exposed only to the atmosphere, is assumed stress-free, \( \beta = 0 \) (also neglecting any shear stresses from winds)

\[
\sigma' : n^{(s)} = 0. \tag{42}
\]

We also have the terminal and lateral BCs which will vary on a case-by-case basis. For the terminal boundaries we mostly apply periodic conditions, meaning that whatever exits the lower terminus of the ice sheet or glacier enters the upper terminus.

Lateral boundary conditions are only necessary for the 3D cases, however these will mostly have no-slip, or highly longitudinally shearing, conditions since ice sheets and ice fields are often funneled through narrow corridors in the form of ice streams and valley glaciers. Some exceptions would be ice caps and cirque glaciers where lateral boundaries would presumably be largely stress-free.

### 2.4 Weak Formulation of the FOS Problem

This section can be quite abstract and require the reader to have, at least, an introductory level of understanding of the FEM. However, it may be glossed over without loss of context for subsequent sections.

Continuing from (28), which is known as the strong form of the PDE, and following the Galerkin FEM from Larson & Bengzon, 2013 and Tezaur et al., 2015, we want to find \( \dot{\varepsilon}_{\text{BP}}(u, v) \), the so-called trial function, such that:
\[ \nabla : (2\eta \dot{e}_{\text{BP}}) = \rho g \nabla s \quad u, v \in \Omega \tag{43} \]
\[ (\sigma' : n^{(b)}) || = -\beta u || \quad u, v, \in \Gamma_b \tag{44} \]
\[ \sigma' : n^{(s)} = 0, \quad u, v, \in \Gamma_s \tag{45} \]

where \( \Gamma_b \) and \( \Gamma_s \) refers to the bottom and surface boundaries respectively. To find the weak formulation of this problem we take the inner product of the PDE (43) and a vector test function \( \phi \), where

\[ \phi \in H^1_0(\Omega) = \{ \phi : \phi \in H^1(\Omega), \phi|_{\Gamma} = 0 \} \subset H^1(\Omega) = \{ \phi : ||\phi|| + ||\nabla \phi|| < \infty \}. \tag{46} \]

I.e. \( \phi \) and its first derivatives are square-integrable (or bounded in the \( L^2 \)-norm) and vanish at the boundaries of \( \Omega \). This is a requirement since the solution is already known at the boundaries (Langtangen & Mardal, 2018). \( H^1(\Omega) \) is known as a Hilbert space. This space is chosen because it allows for discontinuities, which in turn allows for the use of the piecewise polynomial function spaces that the FEM relies on (Langtangen & Logg, 2017). A more detailed description can be found in any introductory textbook on the finite element method (such as Larson & Bengzon, 2013). Multiplying (43) by \( \phi \) is equivalent to projecting the PDE onto the test space defined by (46), see Figure 5.

![Figure 5](image)

**Figure 5:** Projection of PDE solution from the real space to the Hilbert space. Adapted from Larson & Bengzon, 2013.

Subsequently, we integrate over the domain to obtain the weak formulation of the problem: Find \( \dot{e}_{\text{BP}} \in V = \{ \phi \in H^1(\Omega) : \phi|_{\Gamma_b} = \dot{e}_D \} \) (where \( \dot{e}_D \) is the Dirichlet BC for \( \dot{e}_{\text{BP}} \)) such that

\[ \int_{\Omega} \nabla : (2\eta \dot{e}_{\text{BP}}) \phi \, d\Omega = \int_{\Omega} \rho g \phi \nabla s \, d\Omega, \quad \forall \phi \in H^1_0, \tag{47} \]

where \( d\Omega \) is a differential element of the spatial domain \( \Omega \). Applying Green’s Theorem to the left-hand side, we then get
\[- \int_{\Omega} 2\eta \hat{\varepsilon}_{BP} : \nabla \phi \, d\Omega + \int_{\Gamma} n \cdot (2\eta \hat{\varepsilon}_{BP}) \cdot \phi \, d\Gamma = \int_{\Omega} \rho g \phi \nabla s \, d\Omega, \quad \forall \phi \in H^1_0. \quad (48)\]

This is known as the weak form of the PDE, but the BC is strongly imposed since \( \phi \in H^1_0 \). It is weakly imposed if \( \phi \in V \) like our trial function (hence why the trial function is a weak solution). We’ve reformulated the problem by only seeking solutions in the Hilbert space. Regardless, the solution to this weak form still corresponds to the real solution (this is shown in more advanced text books on the FEM).

Simplifying the boundary term using (41) we then obtain

\[- \int_{\Omega} 2\eta \hat{\varepsilon}_{BP} : \nabla \phi \, d\Omega + \int_{\Gamma_b} \beta \mathbf{u} \parallel \cdot \phi \, d\Gamma_b = - \int_{\Omega} \rho g \phi \nabla s \, d\Omega, \quad \forall \phi \in H^1_0. \quad (49)\]

With a given finite element mesh, the FEniCS software solves this system by letting the test function be a set of so-called Lagrange polynomials for each node in the mesh (discretizing the infinite-dimensional space). This ultimately transforms (49) into a matrix system that is solved.

Dirichlet BCs are typically employed using Nitsche’s method (e.g. Massing et al., 2014). However, we cheat a bit by using a simpler but similar method known as the Penalty method. Normally, this is not a stable way of enforcing BCs, but it works in our case.

Take the \( \int_{\Gamma_b} \beta \mathbf{u} \parallel \cdot \phi \, d\Gamma_b \) boundary term from (49) as an example. By letting \( \beta \to \gamma \), where \( \gamma \) is an appropriately scaled large number (in our case \( \frac{10^6}{\text{cell diameter}} \)), we force the velocity \( \mathbf{u} \) to approach zero—thereby enforcing no-slip BCs.

If we want to enforce a specific velocity \( \mathbf{u}_D \) as a Dirichlet BC we can instead write this term as \( \int_{\Gamma_b} \beta (\mathbf{u} \parallel - \mathbf{u}_D) \cdot \phi \, d\Gamma_b \). This implies that when we let \( \beta \to \gamma \) then \( (\mathbf{u} - \mathbf{u}_D) \to 0 \), or simply \( \mathbf{u} \to \mathbf{u}_D \).

A more in-depth description and discussion of the Penalty and Nitsche’s method can be found in Babuška, 1973; Becker et al., 2009 and Nguyen et al., 2017.
3 Methodology

As noted in earlier sections, we formulate the FEM problem in Python and then utilize the FEniCS framework to solve it. Following Langtangen & Logg, 2017, we start off by importing the relevant Python packages, followed by defining the 2D computational domain size \( L_x \) and \( L_y \) and the various parameters from earlier sections—including the rate factor \( A(\theta^*) \), Glen’s flow parameter \( n_i \), the ice density \( \rho \), acceleration due to gravity \( g \), and the slope of the ice sheet \( \alpha \)—as constants. The numerical values of these are provided in the source code (see Section A.1 for all the details).

Next, we define the computational mesh by specifying how many nodes and layers we want (Section A.2). We distinguish here between what’s called a structured and an unstructured mesh (see Figure 6). For our series of tests we’ll be using the structured mesh as it makes it easier to analyze convergence rates by controlling the number of layers (Figure 6a).

This is followed by defining the boundaries of the mesh and the boundary conditions we covered in Section 2.3. Additionally, the steady-state ice sheet geometry is set up here. These definitions (functions) are then called further down in the code when needed (Section A.2). Details on what geometry is used for each test is provided in Section 4.

In the following code segment (Section A.3) we define the strain rate tensor \( \dot{\varepsilon}_{BP} \), the effective strain rate \( \dot{\varepsilon}_0 \), viscosity \( \eta \), and the deviatoric B-P-stress \( \sigma'_{BP} = 2\eta\dot{\varepsilon}_{BP} \). Additionally, we compute the right hand side source terms of (26) using the provided geometry from Section A.2.

In the next section we formulate and solve our finite element problem. The code structure is fairly similar to how we wrote it in Section 2.4.‡ We define our TrialFunction and TestFunction, along with other variables, followed by writing out the system (49) in the form \( F = 0 \), which is then solved:

‡ Note that the FEniCS symbolism is a bit different: \( dx \equiv d\Omega \) and \( da \equiv d\Gamma \) from previous sections.
# Assembly:

\[
\begin{align*}
\mathbf{u} &= \text{TrialFunction}(V) \\
\phi &= \text{TestFunction}(V) \\
h &= \text{CellDiameter}(\text{mesh}) \\
\beta &= \text{Expression}(`1e-3*(1+\sin(2\pi x[0]/Lx))`, Lx=Lx, \text{degree}=1) \\
f &= \text{Source}(\text{degree}=2)
\end{align*}
\]

\[
F = (\text{dot} (\text{deviatoric}_\text{BP}(\mathbf{u}), \text{grad}(\phi)) \text{+} \beta \mathbf{u} \phi \text{+} f \phi) \text{dx}
\]

# Solve nonlinear system:

\[
solve(F == 0, \mathbf{u})
\]

# Solve continuity equation for the vertical velocity:

\[
\begin{align*}
\mathbf{w} &= \text{TrialFunction}(V) \\
F &= (\text{grad}(\mathbf{u})[0] \phi + \text{grad}(\mathbf{w})[1] \phi) \text{dx} + 1e6/h \times (\mathbf{w} - \mathbf{u} \tan(\alpha)) \phi \text{ds}(4)
\end{align*}
\]

\[
solve(F == 0, \mathbf{w})
\]

Preceding this, we marked (or tagged) our boundaries so that we could dictate where each integral of (49) was applied (Section A.4). For example, `\text{bottom.mark(boundaries, 4)}` marks our bottom boundary with the number 4. So, when we later write the term `\beta \mathbf{u}_\text{parallel} \phi \text{ds}(4)` this is only integrated over the bottom boundary, not the interior domain or the surface boundary. Similarly, the `f \phi \text{dx}` is only integrated over the interior domain (which is defined by `\text{dx} \equiv d\Omega`).

Finally, our solutions were exported to a VTK format and visualized and post-processed in Paraview and MATLAB. For more details on the numerics of how the problem is solved the reader is encouraged to check out Langtangen & Logg, 2017.
4 Results and Discussion

Before using the model to simulate any realistic scenarios it is recommended to put it through a set of standardized tests developed by Pattyn & Payne, 2006, known as the Ice Sheet Model Intercomparison Project for Higher-Order ice sheet Models (ISMIP-HOM). These tests are used to benchmark ice models to determine how closely they replicate the full-Stokes model, and hence real ice flow.

4.1 ISMIP-HOM B Test and Convergence Analysis

We focus on simple 2D geometries. ISMIP-HOM B considers a gently sloping ice flow (with an incline $\alpha = 0.5^\circ$) over a rippled (sinusoidal) bedrock with no-slip BCs at the base and periodic BCs at the terminus. The structured finite element mesh for this ice geometry can be seen in Figure 7.

![Structured finite element mesh for ISMIP-HOM B with 50 horizontal elements per layer and 10 vertical layers.](image)

The no-slip BC is obtained by just setting $\beta$ to a large number which, equivalently to Nitsche’s method, forces the velocity to approach zero. The resulting horizontal velocity profiles, $u(x, z)$ for the FOS model and fine-mesh full-Stokes model with the ISMIP-HOM B geometry can be seen in Figures 8a and 8b.

As observed, the FOS model retains the overall horizontal velocity profile of the full-Stokes model, with decreases in the velocity at the bedrock crests and increases at the troughs, as intuitively expected. The margin of error varies around 0-15 [m a$^{-1}$], with the largest errors occurring over the bedrock crest and at mid-depth over the trough. Meanwhile, the lowest errors occur around the surface of the trough and near the bedrock surface (Figure 8c). If we take a closer look at the error distribution over the bedrock crest (Figure 8c) we can observe that it consists of two maxima, where the velocity changes most drastically in Figures 8a and 8b.

One possible explanation for this discrepancy hearkens back to Section 2.2 where we neglected shear stresses acting on the vertical axis in the longitudinal directions, i.e. $\partial_x \sigma_{zz}$. Via Glen’s flow law (22) and (18), this means we neglected $\partial_z u, \partial_x w$ in some terms ($\partial_x \sigma_{zz} = \partial_x (2\eta \dot{\varepsilon}_{zz}) = \partial_x (\eta \partial_z u)$). With the ice thickness decreasing significantly in this region the horizontal velocity is forced to increase. However, due to the no-slip BC, this increase is non-uniform throughout the vertical. As a result, the vertical shear ($\partial_z u$) increases—a term we implicitly neglected in equation (5) for the FOS model.
Figure 8: Horizontal velocity profile for ISMIP-HOM B for (a) the FOS model, and (b) the full-Stokes model, along with (c) the absolute error between these.

Figures 9a and 9b showcases the vertical velocity distribution for ISMIP-HOM B. As observed again, the greatest errors occur where velocity profile changes are the greatest—likely stemming from the hydrostatic approximation made in Section 2.2 (Figure 9c).

Figure 9: Vertical velocity profile for ISMIP-HOM B for (a) the FOS model, and (b) the full-Stokes model, with (c) the absolute error between these.

The FOS and full-Stokes models were tested with a variety of horizontal step sizes ranging from 20-200 m. A fine-mesh full-Stokes model simulation (1000×50 elements) was used as a baseline for all error estimates. The convergence rate for each model in this interval is showcased in Figure 14.
The discretization error is expected to follow the same trend for both models. Since we’re comparing the full-Stokes to itself at a higher resolution its trend is purely due to discretization. For the FOS model however there is a model error superimposed on the discretization error. As observed, while the convergence rate for the full-Stokes model is approximately quadratic, the FOS model is converging relatively slowly at this step size as indicated by the flat-line. This is most likely due to the model error of the FOS being at least one order of magnitude larger than the discretization error. I.e. the approximations we made in Section 2.2 have an inherent error of about 0.06% for ISMIP-HOM B.

![Figure 10](image.png)

**Figure 10**: Relative error in the horizontal velocity compared to the horizontal step size ($dx$) between the FOS and full-Stokes model for ISMIP-HOM B. The quadratic convergence rate for the full-Stokes model has been plotted for reference (green dashed line). The convergence rate of the FOS model has flat-lined around 0.06% for $20 < dx < 200$ [m]. A fine-mesh full-Stokes simulation ($1000 \times 50$ elements) was used as a baseline for all error estimates.

### 4.2 ISMIP-HOM D Test

Similar to ISMIP-HOM B, ISMIP-HOM D considers a gently sloping ice flow, again with a shallow incline of $\alpha = 0.1^\circ$ (Figure 11).

![Figure 11](image.png)

**Figure 11**: Structured finite element mesh for ISMIP-HOM D with 50 horizontal elements per layer and 10 vertical layers.

As observed, in this case the bedrock is flat but we instead have a longitudinally varying drag factor which is defined as follows
\[ \beta^2(x) = 1000 + 1000 \sin(\omega x) , \]  

(50)

where \( \omega = \frac{2\pi}{L_x} \) for \( L_x = 10,000 \) m. This \( \beta \)-profile can be seen in Figure 12.

![Figure 12: Variation in drag factor (\( \beta \)) with distance in ISMIP-HOM D.](image)

The corresponding velocity profiles and their respective errors are plotted in Figure 13. As observed in Figure 13a, the error is the largest in the region of lowest drag and lowest in the region of high drag.

This is probably due to the artificial forcing of a semi-no-slip BC in this region and not due to a model error of the FOS. As intuitively expected, if we force basal velocities to approach zero for both models then we also expect the error between them to approach zero. This artifact naturally propagates upward towards the surface such that total errors are smaller in the no-slip compared to the slip region. Due to this, we can always expect better results for no-slip BCs compared to slip BCs.

Focusing on the slip region we observe that max errors are on the order of 0.1 \([\text{m a}^{-1}]\) in the horizontal velocity (Figure 13a), or about 1%. This seems acceptable, particularly given the low resolution used in the FOS model (Figure 11). Albeit, even with a fine-mesh FOS \((500 \times 10 \text{ elements})\) this error persists.

The error in the vertical velocity however has a strange pattern. It exhibits minima of 0.02 \([\text{m a}^{-1}]\) in the region of maximum positive vertical velocity (where \( \beta \) increases) and a maxima of 0.08 \([\text{m a}^{-1}]\) on the lee side where \( \beta \) decreases. The reason for this is unknown, but could be related to the boundary condition that was imposed via the Penalty method on the vertical velocity (see Section A.4). Alternatively, this could be an inherent problem with the decoupling of the horizontal and vertical velocity dimensions for the FOS model. Although vertical velocity errors were not plotted in Pattyn & Payne, 2006 (Figure 9,
Figure 13: (a) Horizontal and (b) vertical velocity profiles for ISMIP-HOM D, with row 1 being the FOS model, row 2 the full-Stokes model, and row 3 the relative or absolute error between these.

$L = 40$, the norm of the velocity vectors, and their standard deviations, were plotted and hint at similar asymmetries around $x = 0.25$. It is difficult to discern whether these are significant however since the horizontal velocities are around one order of magnitude larger than the vertical velocities.

In the convergence analysis for the entire ISMIP-HOM D region, we observe similar results to ISMIP-HOM B. The FOS model is again flat-lining below $dx = 200$ m. However, the corresponding error is one order of magnitude smaller than for ISMIP-HOM B, now reaching 0.005%, likely due to the simpler geometry for which the FOS model is more valid. In contrast to ISMIP-HOM B, here the full-Stokes model starts flat-lining after the error dips below $10^{-5}$. This is expected however since the Newton iteration method used to solve the full-Stokes problem had a relative tolerance level of $10^{-4}$. Again, errors were obtained by comparing all results to a fine-mesh full-Stokes run.

Figure 14: Relative error in the horizontal velocity compared to the horizontal step size ($dx$) between the FOS and full-Stokes model for ISMIP-HOM D. The convergence rate of the FOS model has flat-lined around 0.006% for $20 < dx < 200$ [m]. Similarly, the full-Stokes model has also started flat-lining below 200 m. Again, a fine-mesh full-Stokes simulation was used as a baseline for all error estimates.
5 Conclusions

The FOS model is a cost-effective substitute for the full-Stokes and an accurate substitute for the SIA and SSA models. Its accuracy is well-founded for ice sheets with frozen bedrocks (no-slip basal BCs), however, experimental evidence is lacking for its accuracy under sliding conditions. Hence, we investigated this by solving the FOS equations using the FEM on a select few bedrock topographies with various BCs.

As shown in Section 4.2, the relative error of the FOS model with varying sliding conditions is around 0.006%, which is significantly less than for the no-slip conditions in Section 4.1. This difference is most likely attributable to the differences in geometry however.

Additionally, by just checking the absolute error in Section 4.2 where the drag varies we can observe maxima of about 0.1 [m a\(^{-1}\)] for the horizontal velocity in the sliding regions (Figure 13a) (1%).

Although this is by no means a rigorous analysis it should rule out any major concerns about the accuracy of the FOS model under sliding conditions. Hence, it should suffice as a cost-effective substitute for both the SIA and SSA while retaining much of the accuracy of the full-Stokes model, particularly in regions closer to the terminal boundary of ice sheets.

Lastly, we observed inexplicable anomalies in the computed vertical velocity that could be due to any number of reasons—a coding or boundary condition error, or perhaps an issue inherent to the decoupling of the velocity dimensions in the FOS model. This should be investigated further.

With the accuracy of the FOS model established in 2D the next step was to experiment with 3D geometries. However, this turned out to be fairly complicated, and traditional packages like CSLVR and VarGlaS seemed to have outdated instructions as of August 2018—making them difficult to install and use with the newer FEniCS versions. The next step would be to try and utilize these packages, which come bundled with real-world geometries (Figure 15), and test the FOS model for more realistic scenarios.

![Figure 15: (a) Observed Greenland horizontal velocities and (b) modeled \(\beta\)-field of from data assimilation using VarGlaS. Source: Quantitative Study of Snow and Ice (QSSI) contributors, 2014](image)
Acknowledgements

I’d like to thank my thesis advisor and Ph.D. student Cheng Gong for taking his time to aid me throughout the project—providing me with both source code and in-depth explanations when I needed them. This particular project was cut a bit short due to insufficient time on my schedule so I hope to be working with Cheng on ice sheet modelling again in the future.

Lastly, I’d also like to thank professors Rickard Pettersson and Lina von Sydow for their inputs and suggestions.
References


**Internet Resources**


Appendices

A  Python Code

A.1  Initialization, Domain, and Physical Parameters

```python
# Packages:
from fenics import *
import numpy as np
import math

# Computational Domain:
Lx = 10000
Ly = 1000

# Physical Parameters (units: km-Mpa-a):
A = 100
n_i = 3.0
rho = 9.03721418e-19
g = 9756234293760000
alpha = 0.1 /180 * pi
```
A.2 Ice Sheet Geometry, Computational Mesh, and Boundary Conditions

# Mesh Size:
\[ n_x = 50 \]
\[ v_1 = 10 \]
\[ h_{\text{max}} = 50 \]

# Sub-domain for bottom Dirichlet BC:
```python
class BottomBoundary(SubDomain):
    def inside(self, x, on_boundary):
        return bool(((x[1]+x[0]*tan(alpha)+Ly) < DOLFIN_EPS+Ly/v1/2.0) and on_boundary)
```

# Sub-domain for terminal Periodic BC:
```python
class PeriodicBoundary(SubDomain):
    # Left boundary is "target domain" G
    def inside(self, x, on_boundary):
        return bool(x[0] < DOLFIN_EPS and x[0] > -DOLFIN_EPS and on_boundary)
    
    # Map right boundary (H) to left boundary (G)
    def map(self, x, y):
        y[0] = Lx - x[0]  # z[0] = Lx
        y[1] = x[1] + Lx*tan(alpha)
```

# Generate specific ice sheet geometry:
```python
def ISMIPHOM_B(nx, alpha, Lx, Ly):
    x = np.array([n*Lx*1./nx for n in range(nx + 1)])
    tops = -1.*x*np.tan(alpha)
    bots = tops - Ly + 0.5*Ly*np.sin(2*pi*x/Lx)
    return x, tops, bots

def ISMIPHOM_D(nx, alpha, Lx, Ly):
    x = np.array([n*Lx*1./nx for n in range(nx + 1)])
    tops = -1.*x*np.tan(alpha)
    bots = tops - Ly
    return x, tops, bots
```
# Map the given geometry to structured mesh with vl vertical layers:
def createStructuredMesh(xvec, topsurf, bottomsurf, vl):
    hl = len(xvec)-1

    # unit square
    mesh = RectangleMesh(Point(0.0, 0.0), Point(1.0, 1.0), hl, vl)

    # First make
    x = mesh.coordinates()[:,0]
y = mesh.coordinates()[:,1]
x0 = min(xvec)
x1 = max(xvec)
xnew = x0 + x*(x1 - x0)
zs = np.interp(xnew, xvec, topsurf)
zb = np.interp(xnew, xvec, bottomsurf)
ynew = zb + y*(zs - zb)
xy_new_coor = np.array([xnew, ynew]).transpose()
mesh.coordinates()[:,] = xy_new_coor

    return mesh

def createUnstructuredMesh(xvec, topsurf, bottomsurf, kmax):
    domain_points = list()
x = len(xvec)

    # Put bottom from z=0->z=L and then top from z=L->z=0
    domain_points = [Point(xvec[i],bottomsurf[i]) for i in range(x)]
domain_points += [Point(xvec[i],topsurf[i]) for i in reversed(range(x))]

    # Polygon vertices must be given in counter clockwise order.
domain = Polygon(domain_points)

    # Generate mesh
    mesh = generate_mesh(domain, kmax)

    return mesh

# Create Mesh:
xvec, topsurf,bottomsurf = ISMIPHOM_D(nx, alpha, Lx, Ly)

mesh = createStructuredMesh(xvec, topsurf, bottomsurf, vl)
# mesh = createUnstructuredMesh(xvec, topsurf, bottomsurf, kmax)
A.3 Strain Rate, Viscosity, and Source Terms

```python
# Compute strain rate and nonlinear viscosity:
def strainrate_BP(u):
    C = Constant(((2.0, 0.0),(0.0, 0.5)))
    return C*grad(u)

def eff_strainrate(u):
    return sqrt(grad(u)[0]**2 + 0.25*grad(u)[1]**2 + 1e-10)

def viscosity(u):
    return 0.5*A**(-1/n_i) * (eff_strainrate(u))**((1-n_i)/n_i)

def deviatoric_BP(u):
    return 2*viscosity(u)*strainrate_BP(u)

# Source Term:
def computedSdx(xp):
    dx = Lx/nx/100.0
    xp_p1 = xp + dx
    xp_m1 = xp - dx
    yp_p1 = np.interp(xp_p1, xvec, topsurf)
    yp_m1 = np.interp(xp_m1, xvec, topsurf)
    return (yp_p1-yp_m1)/dx*0.5

class Source(Expression):
    def eval(self, values, x):
        values[0] = computedSdx(x[0])*rho*g
```
A.4 Defining the Weak Formulation and Solving the Nonlinear System

```cpp
# Initialize mesh function for interior domains
domains = CellFunction("size_t", mesh)

# Initialize mesh function for boundary domains:
boundaries = FacetFunction("size_t", mesh)
bottom = BottomBoundary()
bottom.mark(boundaries, 4)

# Define new measures associated with the interior domains and exterior boundaries:
dx = Measure('dx', domain=mesh, subdomain_data=domains)
ds = Measure('ds', domain=mesh, subdomain_data=boundaries)

# Set periodical conditions for the left and right Boundaries
V = FunctionSpace(mesh, "CG", 2, constrained_domain=PeriodicBoundary())

# Assembly:
u = Function(V)
phi = TestFunction(V)
h = CellDiameter(mesh)
beta = Expression('1e-3*(1+sin(2*pi*x[0]/Lx))', Lx=Lx, degree=2)
f = Source(degree=2)
u_parallel = u  # at base, which follows from dot(n,sigma) = dot(sigma_nn,n) + dot(sigma_nt,t) where n is the normal and t the parallel unit vectors

F = (dot(deviatoric_BP(u), grad(phi)) * dx
    + beta*u_parallel*phi*ds(4)
    + f*phi*dx)

# Solve nonlinear system:
solve(F == 0, u)  #, bc)

# Solve incompressible equation for the vertical velocity:
w = Function(V)
F = (grad(u)[0]*phi + grad(w)[1]*phi)*dx + 1e6/h *(w-u*tan(alpha))*phi*ds(4)
solve(F == 0, w)

# Save solution in VTK format:
ufilename_pvd = File("firstOrderStoke_u.pvd")
ufile_pvd << u
vfilename_pvd = File("firstOrderStoke_w.pvd")
vfile_pvd << w
```

28
Glossary

**Cauchy stress tensor**  A tensor that completely defines the total stress at a point, including all normal and shear stresses. 3, 4, 29

**creep**  Deformation due to applied stress. 29

**Einstein summation convention**  Repeated indices, or products with repeated indices, such as $u_i v_{ij}$, are implicit summations over those indices, i.e. $u_i v_{ij} = \sum_i u_i v_{ij}$. 4, 8, 29

**Green’s Theorem**  Also known as *integration by parts* in two dimensions. Relates the double integral over a domain $\Omega$, bounded by a closed curve $\Gamma$, to the line integral around the curve $\Gamma$

$$-\int_{\Omega} \nabla \cdot (a \nabla \phi) \, d\Omega = \int_{\Omega} a \nabla u \cdot \nabla \phi \, d\Omega - \int_{\Gamma} n \cdot (a \nabla u) \phi \, d\Gamma$$

where $u, \phi,$ and $a$ are all functions of the space $\Omega$. 11, 29

**Hilbert space**  Among other subspaces, it contains the space of square-integrable functions. I.e. the space of functions that satisfy: $\int_{\Omega} u(\Omega)^2 d\Omega < \infty$. 11, 12, 29

**hydrostatic stress**  The negative of the hydrostatic pressure, which acts equally in all directions. 4, 29

**ice divide**  The boundary of an ice sheet that separates opposite flow directions. 6, 29

**ice shelf**  A thick floating ice platform formed as an ice sheet or glacier enters the ocean. 1, 29

**ice stream**  A region where the ice moves significantly faster than the surrounding ice—typically in sloping regions near the terminal boundaries of ice sheets. 1, 10, 29

**internal deformation**  The displacement of ice due to the alignment and subsequent sliding of layers of ice crystals. This alignment of ice crystals is caused by the weight of the overlying ice. Since the pressure at the surface is negligible this type of deformation only occurs within the ice, hence the term *internal* deformation. 3, 7, 29

**invariant**  A matrix property that remains unchanged despite coordinate transformation. 5, 29

**Newtonian fluid**  A fluid where the local stress is proportional to the strain rate by a constant viscosity tensor. If the viscosity tensor in itself depends on the stress or the velocity then the fluid is non-Newtonian. Alternatively, a fluid is Newtonian if the Glen parameter $n = 1$. 6, 29
**polycrystalline** An aggregate of individual grain-sized crystals. 3, 29

**rheology** The deformation properties of fluids or plastically deforming bodies. 6, 29

**Stokes model** A special case of the Navier-Stokes equations where advection terms are small compared to viscosity terms. It very closely models real ice flow. 1, 15–17, 19, 20, 29

**test function** A well-behaved square-integrable arbitrary function that vanishes at the boundaries of the domain. 11, 12, 29

**traction** A vector $\tau$ that is the force vector $F$ acting per unit area

$$\tau = \frac{F}{\text{Area}}.$$  

Not to be confused with the pressure, or normal stress, which is the normal component of the traction vector. The component of $\tau$ parallel to the surface is the shear stress. 8, 29

**trial function** Essentially just the function of the sought solution which satisfies the boundary conditions. 10, 12, 29

**Acronyms**

**B-P** Blatter-Pattyn. 1, 6, 7, 13, 29

**BC** boundary condition. 6–12, 15, 18, 20, 29

**CSLVR** Cryospheric Problem Solver. 5, 20, 29

**FDM** Finite Difference Method. 2, 29

**FEM** Finite Element Method. 2, 10–13, 20, 29

**FEniCS** Finite Element Computational Software. 2, 12, 13, 20, 29

**FOS** First Order Stokes. 1, 2, 15–20, 29

**ISMIP-HOM** Ice Sheet Model Intercomparison Project for Higher-Order ice sheet Models. 15–19, 29

**PDE** Partial Differential Equation. 3, 10–12, 29

**SIA** Shallow Ice Approximation. 1, 20, 29
SSA  Shallow Shelf Approximation.  1, 20, 29

VarGlaS  Variational Glacier Simulator.  5, 20, 29

Symbols

$A$  Rate factor.  29

$L_x$  Longitudinal domain length.  29

$\Gamma$  Boundary domain.  29

$\Omega$  Spatial domain.  29

$\Phi$  Frictional heating due to internal deformation.  29

$\alpha$  Steady-state surface slope angle.  29

$\beta$  Drag factor.  29

$\delta_{ij}$  Kronecker delta.  29

$\dot{\varepsilon}_0$  Small constant strain rate.  29

$\dot{\varepsilon}_E$  Effective strain rate.  29

$\dot{\varepsilon}$  Strain rate.  29

$\dot{\varepsilon}_{BP}$  Blatter-Pattyn strain rate.  29

$\eta$  Effective viscosity.  29

$\rho$  Ice density.  29

$\sigma_E$  Effective stress.  29

$\theta^*$  Ice temperature, corrected for pressure melting.  29

$\theta$  Ice temperature.  29

$\sigma'$  Deviatoric stress tensor.  29

$\sigma_{hyd}$  Hydrostatic stress tensor.  29

$\sigma$  Cauchy stress tensor.  29
\( u \) Velocity vector. 29

\( b \) Base topography elevation. 29

\( c_p \) Specific heat capacity for ice. 29

\( g \) Gravitational acceleration. 29

\( k \) Thermal conductivity for ice. 29

\( n \) Glen’s parameter. 29

\( s \) Ice surface elevation. 29
Software

FEniCS Project

- Developer: FEniCS Community
- Year: 2017
- Version: 1.5
- Available at: https://fenicsproject.org/

Python

- Developer: Python Software Foundation
- Year: 2017
- Version: 3.5.2
- Available at: https://www.python.org/

Docker

- Developers: Docker, Inc.
- Year: 2018
- Version: 18.06.0-ce
- Available at: https://www.docker.com/

ParaView

- Developers: Sandia National Laboratory, Kitware Inc, Los Alamos National Laboratory
- Year: 2018
- Version: 5.5.0
- Available at: https://www.paraview.org/
MATLAB

- Developer: The MathWorks, Inc.
- Year: 2018
- Version: R2018a
- Available at: https://mathworks.com/