Assessing the benefits of approximately exact step sizes for Picard and Newton solver in simulating ice flow (FEniCS-full-Stokes v.1.3.1)

Niko Schmidt[1], Angelika Humbert[2,3], and Thomas Slawig[1,4]

[1]Dep. of Computer Science, Kiel University, Kiel, Germany
[2]Alfred-Wegener-Institut Helmholtz-Zentrum für Polar- und Meeresforschung, Bremerhaven, Bremen, Germany
[3]Faculty of Geosciences, University of Bremen, Bremen, Germany
[4]Kiel Marine Science (KMS) – Centre for Interdisciplinary Marine Science

Correspondence: Niko Schmidt (nis@informatik.uni-kiel.de)

Abstract. Solving the momentum balance is the computationally expensive part of simulating the evolution of ice sheets. The momentum balance is described by the nonlinear full-Stokes equations, which are solved iteratively. We use the Picard iteration and Newton's methods combined with Armijo step sizes and approximately exact step sizes, respectively, to solve these equations. Only the approximately exact step sizes are used for the Picard iteration. We compare the Picard iteration and the variants of Newton's method in benchmark experiments, called ISMIP-HOM experiments A, B, E1, and E2. The ISMIP-HOM experiments consist of a more realistic domain and are designed to test the quality of ice models. For an even more realistic test case, we simulate the experiments E1 and E2 with a time-dependent surface. We obtain that the Picard iteration and Newton's method with approximately exact step sizes greatly reduce the necessary number of iterations with nearly no increase of the computation time for each iteration.

1 Introduction

Simulating the evolution of the ice sheets in Greenland and Antarctica in adequate physics and resolution is a challenging task. The dynamics of ice sheets is described as a fluid mechanical problem with the momentum balance reduced to a Stokes problem as acceleration and Coriolis forces are negligible. In the past computational constraints led to the reduction of the problem by approximating the momentum balance. If the spatial resolution cannot be choosen sufficiently large, the benefit from solving the Stokes problem is lost. Consequently, in practical terms, Stokes models are leading to large problems and thus efficient solvers are inevitable. This is what this study is focusing on.

The full-Stokes equations are nonlinear partial differential equations described as shear thinning, which means that the viscosity depends non-linearly on the symmetric gradient. More precisely, we consider the stationary variant of these equations in the variational formulation. In ice models, the stationary equations are solved to calculate the velocity field. Then, the velocity field is used to calculate the new shape of the glacier. The variational formulation is needed to calculate a solution with finite elements. A common method to calculate the solution of these equations is the Picard iteration, (see Colinge and Rappaz, 1999). The Picard iteration fixes the nonlinear viscosity, calculates a new velocity, and updates the viscosity.

The Picard iteration is, for example, used in the ice model *ISSM*, Larour et al. (2012), and *FELIX-S*, Leng et al. (2012). *Elmer/Ice*, Gagliardini et al. (2013), uses the Picard iteration for the first few iterations and the Newton method for the last iterations. This approach was also extended to a nonlinear friction law. However, *Elmer/Ice* does not use a step size control. Thereby, Newton's method does not converge from every initial point, Gagliardini et al. (2013). The ice models *Elmer/Ice* and *FELIX-S* are compared in Zhang et al. (2017). For some glacier simulations *COMSOL multiphysics* is used, (see Rückamp et al., 2022). Newton's method in *COMSOL multiphysics* switches between choosing the negative gradient and Newton directions. Additionally, a trust-region method is used to determine the step sizes. The trust-region radius is the maximum step size that one would trust the step size to be a good choice, for more mathematical details, see (Dennis and Schnabel, 1996, section 6.4.2). The trust-region method in *COMSOL multiphysics* uses the residual norm, see (COM, 2018, section The Fully Coupled Attribute and the Double Dogleg Method). Instead, we will discuss another approach, which is algorithmically simpler and uses the problem-specific information of a convex function. Additionally, the convex function allows us to use different step size methods.

We employ Newton's method by formulating the variational formulation as a root problem. If we start near the solution, Newton's method is superlinear convergent, (Hinze et al., 2009). Thus, the error between the approximation and the real solution reduces faster than the linear convergent Picard iteration (see Fraters et al., 2019). However, starting with an unsuitable initial velocity field for Newton's method could lead to a diverging velocity. A step size control guarantees convergence from every initial guess. This step size control is constructed by defining a function that we want to minimize. One variant is presented in Fraters et al. (2019). We consider another approach that only needs to calculate integrals. It allows us to use two different step size controls. Newton's method with one of these step sizes converges from every initial guess to the solution, (Schmidt, 2023). Additionally, we employ approximately exact step sizes for the Picard iteration to provide a possibility to reduce the necessary number of iterations without implementing Newton's method. The exact step sizes are the solution of a one-dimensional minimization problem. As we can only approximate these step sizes arbitrarily precise, we called them approximately exact and will call them exact step sizes for brevity.

The computation of the step size is computationally cheap compared to solving the linear systems of equations in each iteration. The work of Habbal et al. (2017) considers different solvers to reduce the simulation time for solving the system of linear equations. Nonetheless, for all solvers, the system of linear equations is still the main computational effort. Our step size control reduces the computation time by reducing the necessary number of iterations.

As a test case, we use the ISMIP-HOM experiments A, B, $E1$, and $E2$. These experiments are designed to test the quality of glaciological models. They reflect a large domain of the glaciers, a large aspect ratio, and a sinusoidal bedrock and the Glacier d'Arolla, respectively. We simulate the Glacier d'Arolla experiments $E1$ and $E2$ also time-dependent.

The manuscript has the following structure: In Sect. 2, we introduce the equations in the variational formulation and the Picard iteration. In the subsequent section, we formulate Newton's method. In Sect. 4, we introduce the new idea, the step size control that decreases the number of iterations and verifies convergence from every initial guess. In Sect. 5, we discuss the stationary ISMIP-HOM experiments A and B and compare them with the results in Pattyn et al. (2008). In Sect. 6 , we

solve instationary problems with and without sliding derived from the ISMIP-HOM experiments $E1$ and $E2$. Finally, we give a summary in Sect. 7 and an outlook in Sect. 8.

2 The full-Stokes equations as a root problem

Let $\Omega \subseteq \mathbb{R}^N$ with $N \in \{2,3\}$. For describing the movement of ice, we need the second-order tensor $\boldsymbol{\sigma}$, the density ρ, and the gravitational acceleration $\boldsymbol{g}$. These quantities describe the full-Stokes equations, the most complex equations for simulating ice, by:

$$\begin{aligned} -\mathrm{div}\boldsymbol{\sigma} &= -\rho\boldsymbol{g}, \\ \mathrm{div}\boldsymbol{v} &= 0 \end{aligned} \tag{1}$$

on the domain Ω. We describe the stress tensor $\boldsymbol{\sigma}$ with the pressure p, the identity tensor (matrix) $\boldsymbol{I}$, the symmetric gradient D, the velocity $\boldsymbol{v}$, and the viscosity μ by $\boldsymbol{\sigma} := p\boldsymbol{I} - \mu D\boldsymbol{v}$. We define the nonlinear viscosity μ as

$$\mu = B\big(|D\boldsymbol{v}|^2 + \delta^2\big)^{\frac{1-n}{2n}}, \quad (D\boldsymbol{v})_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right), \quad |D\boldsymbol{v}|^2 := D\boldsymbol{v} : D\boldsymbol{v} := \sum_{i,j=1}^{N} |(D\boldsymbol{v})_{ij}|^2 \tag{2}$$

with $n \in (1,\infty)$ and $B = B(x_1, x_2, x_3)$, $\delta > 0$. The constant $\delta > 0$ guarantees $\mu < \infty$. We choose $n = 3$ for the experiments as in Pattyn et al. (2008). The boundary consists of the bedrock Γ_b, the surface Γ_s, and the lateral boundary Γ_ℓ. Our boundary conditions are:

$$\begin{aligned} \boldsymbol{v} &= \boldsymbol{0} \quad \text{on } \Gamma_b \cup \Gamma_\ell, \\ \boldsymbol{\sigma}\cdot\boldsymbol{n} &= \boldsymbol{0} \quad \text{on } \Gamma_s \end{aligned} \tag{3}$$

with the outer normal vector $\boldsymbol{n}$. Here, $\boldsymbol{\sigma}\cdot\boldsymbol{n}$ is the inner tensor-product (matrix-vector multiplication).

We derive the variational formulation in infinite dimensions because we can implement it directly in *FEniCS*, (see Logg et al., 2012). We determine the variational formulation by multiplying with test functions and using partial integration, and second, we explain the function spaces used. We define an operator $G : H \times L \to H^* \times L^*$ by

$$\langle G(\boldsymbol{v},p),(\boldsymbol{\phi},\psi)\rangle = \int_\Omega B\big(|D\boldsymbol{v}|^2+\delta^2\big)^{\frac{1-n}{2n}} D\boldsymbol{v} : \nabla\boldsymbol{\phi}\,dx + \mu_0\int_\Omega \nabla\boldsymbol{v} : \nabla\boldsymbol{\phi}\,dx - \int_\Omega p\,\mathrm{div}\boldsymbol{\phi}\,dx - \int_\Omega \mathrm{div}\boldsymbol{v}\,\psi\,dx + \int_\Omega \rho\boldsymbol{g}\cdot\boldsymbol{\phi}\,dx, \tag{4}$$

where $\boldsymbol{v} \in H$ and $p \in L$ are the solution, and $\boldsymbol{\phi} \in H$ and $\psi \in L$ are test functions. The square brackets on the left-hand side of the equation are used because, formally, we have a function that maps to the dual space. The dual space is denoted by the star after the space, e.g., H^* and L^*. The solution of the full-Stokes equations are $(\boldsymbol{v},p) \in H \times L$ with

$$\langle G(\boldsymbol{v},p),(\boldsymbol{\phi},\psi)\rangle = 0 \quad \text{for all } (\boldsymbol{\phi},\psi) \in H \times L \tag{5}$$

We added the diffusive term $\mu_0 > 0$ to get a well-posed directional derivative and a well-posed Picard iteration, for details, see Appendix A1. Additionally, we are now in a Hilbert space formulation and set $H := \{\boldsymbol{v} \in H^1(\Omega)^N; \boldsymbol{v}|_{\Gamma_b\cup\Gamma_\ell} = \boldsymbol{0}\}$, where

$H^1(\Omega)^N$ is the space of vector-valued square integrable functions with a square integrable derivative. We set $L := \{p \in L^2(\Omega);\ \int_\Omega p\,dx = 0\}$ for the space of square integrable functions with zero integral. There exists a unique solution to that problem for $\mu_0 > 0$, (see Schmidt, 2023). Nonetheless, we perform all experiments with $\mu_0 = 0$.

We formulate the problem in infinite-dimensional spaces H and L. In these infinite-dimensional spaces, mathematical convergence properties are independent of the mesh resolution and the used finite elements, as long as the finite elements are a subspace of the infinite-dimensional spaces. Ice models often use finite elements. Moreover, the formulation in discretized spaces is identical, only the functions are from finite-dimensional spaces.

A common method to solve the variational formulation of the full-Stokes equations in glaciological models is the Picard iteration, see Algorithm 1. It is used in *ISSM*, see Larour et al. (2012) , *FELIX-S*, see Leng et al. (2012). *Elmer/Ice* can use the Picard iteration and Newton's method, which we will introduce in the next section. *Elmer/Ice* can use the Picard iteration to get near the solution and then can use Newton's method, see Gagliardini et al. (2013).

Algorithm 1 Picard iteration

1: Let $\boldsymbol{v_0} \in H$ and $p_0 \in L$ be given.

2: **for** $k = 0, 1, \ldots$ **do**

3: Calculate $\boldsymbol{v_{k+1}} \in H$ and $p_{k+1} \in L$ with

$$\int_\Omega B\left(|D\boldsymbol{v_k}|^2 + \delta^2\right)^{\frac{1-n}{2n}} D\boldsymbol{v_{k+1}} : \nabla\boldsymbol{\phi}\,dx + \mu_0 \int_\Omega \nabla\boldsymbol{v_{k+1}} : \nabla\boldsymbol{\phi}\,dx - \int_\Omega p_{k+1}\mathrm{div}\boldsymbol{\phi}\,dx - \int_\Omega \mathrm{div}\boldsymbol{v_{k+1}}\psi\,dx = -\int_\Omega \rho\boldsymbol{g}\cdot\boldsymbol{\phi}\,dx$$

for all $\phi \in H$ and $\psi \in L$.

4: **end for**

3 Newton's method

The Picard iteration converges slowly (see Fraters et al., 2019). Thus, it can be beneficial to consider faster converging algorithms. Newton's method is often superlinear convergent, also in infinite dimensions, (see Hinze et al., 2009). For Newton's method, the calculation of the derivative is necessary. Due to the variational formulation, we can only express the derivative of G in terms of the direction and the test functions. The derivative of G in $(\boldsymbol{v}, p)$ in direction $(\boldsymbol{w}, q)$ is

$$\begin{aligned}\langle G'(\boldsymbol{v},p)(\boldsymbol{w},q),(\boldsymbol{\phi},\psi)\rangle = &\int_\Omega \frac{1-n}{n} B\left(|D\boldsymbol{v}|^2+\delta^2\right)^{\frac{1-3n}{2n}} (D\boldsymbol{v} : D\boldsymbol{w})\,(D\boldsymbol{v} : \nabla\boldsymbol{\phi})\,dx + \mu_0 \int_\Omega \nabla\boldsymbol{w} : \nabla\boldsymbol{\phi}\,dx \\ &+ \int_\Omega B\left(|D\boldsymbol{v}|^2+\delta^2\right)^{\frac{1-n}{2n}} D\boldsymbol{w} : \nabla\boldsymbol{\phi}\,dx - \int_\Omega q\mathrm{div}\boldsymbol{\phi}\,dx - \int_\Omega \mathrm{div}\boldsymbol{w}\psi\,dx. \qquad (6)\end{aligned}$$

A mathematical proof that G is differentiable in all directions $(\boldsymbol{w}, q)$ is presented in Schmidt (2023). A more detailed deduction of the derivative is in Subsect. $A2$.

Newton's method can solve the full-Stokes equations by Algorithm 2. Because Newton's method is only locally convergent, we use a step size control in Algorithm 2. We explain the step size control in the next section.

Algorithm 2 Globalized Newton's method

1: Let $(\boldsymbol{v_0}, p_0)$ be given.
2: **for** $k = 0, 1, \ldots$ **do**
3: Calculate $(\boldsymbol{w_k}, q_k)$ with

$$\langle G'(\boldsymbol{v_k}, p_k)(\boldsymbol{w_k}, q_k), (\boldsymbol{\phi}, \psi)\rangle = -\langle G(\boldsymbol{v_k}, p_k), (\boldsymbol{\phi}, \psi)\rangle \text{ for all } \boldsymbol{\phi} \in H, \psi \in L.$$

4: Set $\boldsymbol{v_{k+1}} := \boldsymbol{v_k} + \alpha_k \boldsymbol{w_k}$ and $p_{k+1} := p_k + \alpha_k q_k$ with a suitable $\alpha_k > 0$.
5: **end for**

4 Step size control

In this section, we derive a global convergent Newton method by using a step size control: We have the current velocity field $\boldsymbol{v}$ and the direction $\boldsymbol{w}$. Instead of setting our new field $\tilde{\boldsymbol{v}} := \boldsymbol{v} + \boldsymbol{w}$, we choose $\alpha > 0$ with $\tilde{\boldsymbol{v}} := \boldsymbol{v} + \alpha \boldsymbol{w}$. We want an algorithm for choosing this α. Classical approaches for determining the step size α check, if the norm $\|G(\boldsymbol{v_{k+1}}, p_{k+1})\|$ reduces enough compared to $\|G(\boldsymbol{v_k}, p_k)\|$. What enough reduction means is, for example, discussed in Hinze et al. (2009). However, we use an alternative approach. Solving $G(\boldsymbol{v}, p) = 0$ is equivalent to minimizing $J : H \to \mathbb{R}$

$$J(\boldsymbol{v}) = \int_\Omega \frac{n}{1+n} B\big(|D\boldsymbol{v}|^2 + \delta^2\big)^{\frac{1+n}{2n}}\, dx + \frac{\mu_0}{2}\int_\Omega |\nabla \boldsymbol{v}|^2\, dx + \int_\Omega \rho \boldsymbol{g}\cdot\boldsymbol{v}\, dx - \int_\Omega p\mathrm{div}\boldsymbol{v}\, dx, \tag{7}$$

see Schmidt (2023). We need the last summand because the time-dependent experiments lead to initial guesses for the velocity field that are not divergence-free: We start with a divergence-free initial guess, calculate the velocity field, and use the velocity field to calculate the new domain. (We explain the calculation of the new domain in subsect. 6.2.) The grid points with the velocity information are moved correspondingly to fit the new domain. On this new domain, our old velocity field is our initial guess and slightly not divergence-free. (Despite $\mathrm{div}\boldsymbol{v}$ being near 0, the step size control did not work without the last summand in Eq. (7) for time-dependent problems.)

The convex functions were also used in Hirn (2013) for $\mu_0 = 0$ with Dirichlet boundary conditions, and Chen et al. (2013) for $\delta = 0$ and $\mu_0 = 0$ with more realistic boundary conditions. The equivalence between minimizing this convex function and solving the full-Stokes equations is clear because the minimizer of the function and the root of the derivative are at the same point for strict convex functions.

A classical approach to determine a suitable step size α is the use of an Armijo step size as in Hinze et al. (2009), see Algorithm 3. We describe the idea of Armijo step sizes: For a function, the negative gradient is the direction of the steepest descent. To find the minimum, we need enough reduction compared to the steepest descent multiplied with the step size, and a factor between zero and one as we expect less reduction than the steepest descent. This condition is stated in line 4 of Algorithm 3. Line 3 guarantees that the step size is not too small. For Newton's method one chooses $\alpha := 1$, (see Nocedal and Wright, 2006, Algorithm 3.1). Newton's method converges fast for $\gamma \in (0, 1/2)$, (see Nocedal and Wright, 2006, Theorem

Algorithm 3 Armijo step size

Let $\gamma \in (0, 1/2)$.
for $i = 0, 1, \ldots$ **do**
  Set $\alpha := 1$.
  while $J(\boldsymbol{v} + \alpha \boldsymbol{w}) - J(\boldsymbol{v}) > \alpha\gamma J'(\boldsymbol{v})\boldsymbol{w}$ **do**
    Set $\alpha := 0.5\alpha$.
  end while
end for
return α

3.6). Typically, $\gamma := 10^{-4}$ is chosen, see Nocedal and Wright (2006). However, we choose $\gamma := 10^{-10}$ as 10^{-4} was to strict. We only need a direction that reduces the function value to use Armijo step sizes.

However, we can exploit the strict convexity of J for constructing other step sizes: We define the auxiliary function

$$J_k(\alpha) := J(\boldsymbol{v_k} + \alpha \boldsymbol{w_k}). \tag{8}$$

The function J_k is strictly convex, $J_k'(\alpha) = J'(\boldsymbol{v_k} + \alpha \boldsymbol{w_k})\boldsymbol{w_k}$ is negative for $\alpha = 0$ and positive for big enough α because of the choice of $\boldsymbol{w_k}$ and the strict convexity of J_k. As J_k' is continuous, a simple bisection, see Algorithm 4, calculates the minimum of J_k. In practice, we approximate the exact step size arbitrarily precise. Thus, we denote the approximate exact step size as the exact step size. Exact step sizes have the advantage that we really calculate the minimum in a direction instead of just having some reduction. Nonetheless, the exact step size is only rarely used in practice. One needs more conditions on the problem, here the strict convexity of J and J_k, respectively. We could not find the minimum of

$$\|G(\boldsymbol{v_k} + \alpha \boldsymbol{w_k}, p_k + \alpha q_k)\|$$

for the direction $(\boldsymbol{w_k}, q_k)$, because we have no information where the minimum is.

For simplicity, we choose 25 iterations to approximate the exact step size. As we expect step sizes of length 1 to be often a good choice, we chose the maximum step size of 4 in our implementation, we obtain an accuracy of $4/2^{25} \approx 10^{-7}$ for the step size α. The calculation of α is computationally not expensive.

We modify the Picard iteration, see Algorithm 1, by a relaxation: We set

$$\tilde{\boldsymbol{v}}_{k+1} := (1 - \alpha_k)\boldsymbol{v_k} + \alpha_k \boldsymbol{v_{k+1}}, \quad \tilde{p}_{k+1} := (1 - \alpha_k)p_k + \alpha_k p_{k+1} \tag{9}$$

and choose α_k as α in Algorithm 4.

5 Stationary experiments

We analyze the four algorithms we introduced: The classical Picard iteration as a reference, exact step sizes for the Picard iteration and Newton's method, and Armijo step sizes for Newton's method. We implemented all these algorithms in *FEniCS*

Algorithm 4 Exact step size

Set $a, b \in [0, \infty)$ with $a < b$.
for $i = 0, 1, \ldots$ **do**
  if $J_k'((a+b)/2) > 0$ **then**
    Set $b := (a+b)/2$.
  else
    Set $a := (a+b)/2$.
  end if
end for
return $\alpha := (a+b)/2$

version 2019.1.0, (see Logg et al., 2012). *FEniCS* is a library that allows to implement variational formulations easily. Hence, it allows fast testing of algorithms without implementing them in complex codes. We determine the performance of these algorithms by comparing each iteration step with a reference solution for the experiments ISMIP-HOM A and B, (see Pattyn et al., 2008). The reference solution is calculated with 80 prescribed Picard iterations as the Picard iteration converges slowly and the reference solution should be more accurate than the solutions calculated by the compared methods. In Pattyn et al. (2008), the authors described the ISMIP-HOM experiments to analyze the quality of ice models. Moreover, they compared simulation results.

We set the physical variables according to Pattyn et al. (2008): $B := 0.5 \cdot (10^{-16})^{-1/3}$ (Pa)$^{-3}$ a^{-1}, $\rho := 910$ kg m^{-3}, and $\boldsymbol{g} := (0, 9.81)$ m s^{-2}.

We set the constant $\delta := 10^{-12}$a^{-1} and $\mu_0 := 0$ kg a m^{-1} s^{-2}. We derive the unit for μ_0 by $[\mu_0 |\nabla \boldsymbol{v}|^2] = [\rho \boldsymbol{g} \cdot \boldsymbol{v}]$. In the experimental design, the nonlinear term is $2B(0.5|D\boldsymbol{v}|^2 + \delta^2)^{(1-n)/(2n)}$ instead of $B(|D\boldsymbol{v}|^2 + \delta^2)^{(1-n)/(2n)}$, (see Pattyn et al., 2008). We choose the constant δ such that δ is smaller than the typical magnitude of $D\boldsymbol{v}$, $3 \cdot 10^{-4}$ a^{-1} and 3 a^{-1}, multiplied with the machine precision eps:

$$\delta < eps \sqrt{0.5} |D\boldsymbol{v}| \tag{10}$$

for typical values of $|D\boldsymbol{v}|$. We defined the infinite-dimensional algorithms for $\mu_0 > 0$ as they are not well-posed for $\mu_0 = 0$. However, we do all simulations with $\mu_0 = 0$ because the additional diffusion term is not used in ice models and the diffusion term is not needed in the finite dimensional finite element spaces, Hirn (2013). In all experiments, we calculate the initial velocity by replacing $(0.5|D\boldsymbol{v}|^2 + \delta^2)^{(1-n)/(2n)}$ with 10^6 and solving this linear problem. As starting with $|D\boldsymbol{v}| = 3 \cdot 10^{-4}$ a^{-1} leads to $(0.5|D\boldsymbol{v}|^2 + \delta^2)^{(1-n)/(2n)} \approx 281$ a$^{2/3}$ and starting with a constant velocity field leads to $(0.5|D\boldsymbol{v}|^2 + \delta^2)^{(1-n)/(2n)} = 10^8$ a$^{2/3}$, we chose 10^6 between both values.

5.1 The original experiment ISMIP-HOM B

In this subsection, we introduce details from Pattyn et al. (2008) that are specific to the experiment ISMIP-HOM B. This experiment has a domain with a sinusoidal, slightly tilted $(0.5°)$, bottom. The boundaries at the left and right are vertical, and the boundary at the top has a linear slope of $0.5°$. Furthermore, periodic boundary conditions are used at Γ_ℓ. The experiment prescribes Dirichlet zero boundary conditions, $\boldsymbol{v} = \mathbf{0}$ on Γ_b and $\boldsymbol{\sigma} \cdot \boldsymbol{n} = \mathbf{0}$ on Γ_s.

The length $L := 5$ km is the horizontal extent. The angle $\beta := 0.5°$ describes a slight decline at the surface and the bottom by

$$z_s(x) = -x \tan(\beta), \quad z_b(x) = z_s(x) - 1000 + 500 \cdot \sin(\omega x) \tag{11}$$

with $\omega := 2\pi/L$.

5.2 Modifications to the experiment ISMIP-HOM B

Formulating the convex function J, see Eq. (7), that corresponds to periodic boundary conditions is complicated. Thus, we use the alternative introduced in the supplement of Pattyn et al. (2008), by copying the glacier to the right and the left. We have three copies to the right and the left, see Fig. 1. At the lateral boundaries Γ_ℓ, we impose Dirichlet zero boundary conditions. Also, the resolution at the outer copies is lower than for the original domain. This reduces for the two-dimensional experiment the number of elements by 30 % and in three dimensions by 51 %. Nevertheless, the three-dimensional experiment was performed on a high performance computer. In two dimensions, the local refinement has no relevant impact on the solution. Also, one can simulate the two-dimensional experiment on a laptop.

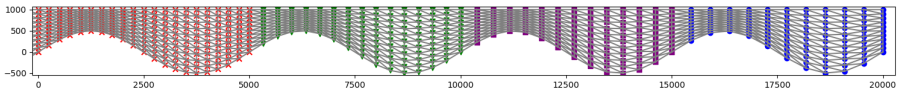

Figure 1. The domain with a grid with red dots and three copies to the right with green, purple, and blue dots.

Instead of the slope, we rotate the gravity. Thus, we should rotate the lateral boundaries Γ_ℓ of the domain. We neglect this and stick to vertical boundaries at the left and the right.

5.3 Results for experiment ISMIP-HOM B

Our velocity fields at the surface (see Fig. 2) are close to the mean of the full-Stokes simulations in Pattyn et al. (2008). Also, all our methods produce very similar velocity fields at the surface, as displayed in Fig. 3. Next, we compare how many iterations are necessary to reduce the relative difference and relative local difference compared to the reference solution. We calculate the relative difference and the relative local difference for a velocity $\boldsymbol{v}$ and the reference solution $\boldsymbol{v}_{\mathbf{ref}}$ by

$$\sqrt{\frac{\int_\Omega |\boldsymbol{v} - \boldsymbol{v}_{\mathbf{ref}}|^2 \, dx}{\int_\Omega |\boldsymbol{v}_{\mathbf{ref}}|^2 \, dx}} \text{ and } \sqrt{\int_\Omega \frac{|\boldsymbol{v} - \boldsymbol{v}_{\mathbf{ref}}|^2}{\max(|\boldsymbol{v}_{\mathbf{ref}}|^2, c^2)} \, dx} \tag{12}$$

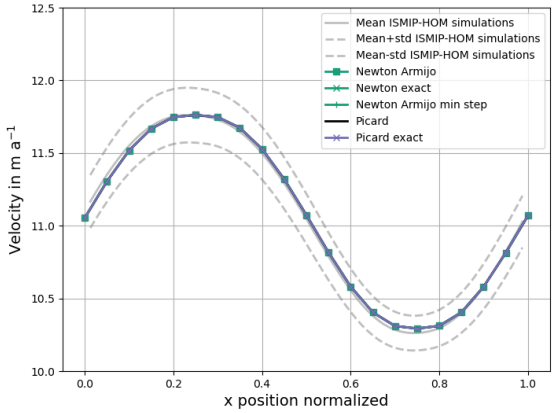

Figure 2. Simulated surface velocity for different solvers for ISMIP-HOM B. All our calculated velocity fields overlap each other. In grey are plotted the mean and the standard deviation (std) from Pattyn et al. (2008) with 9 models. The mean and standard deviation have no values at $x = 0$ and $x = 1$ due to missing values.

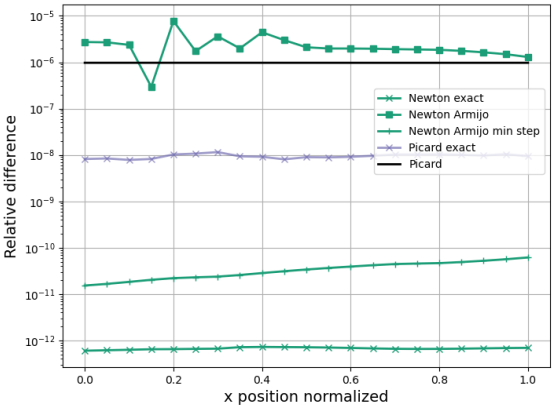

Figure 3. Relative difference of $|\boldsymbol{v} - \boldsymbol{v}_{\mathbf{ref}}|/\boldsymbol{v}_{\mathbf{ref}}$ for each grid point at the surface. The reference solution is the solution from 80 Picard iterations.

with $c = 1$ mm a^{-1}. Then, c is much smaller than 10m a^{-1} and below this velocity speed slight differences are seen as not so important, (Joughin et al., 2010, section 2.3). We use two error measurements because one method could be better for one purpose and the other for another. The local relative difference reflects that regions with small velocities should also be represented with a small relative error. Both error measurements consider the velocity field for the whole domain of the glacier. In contrast, the original experiment (Pattyn et al., 2008) only considers the velocity field at the surface.

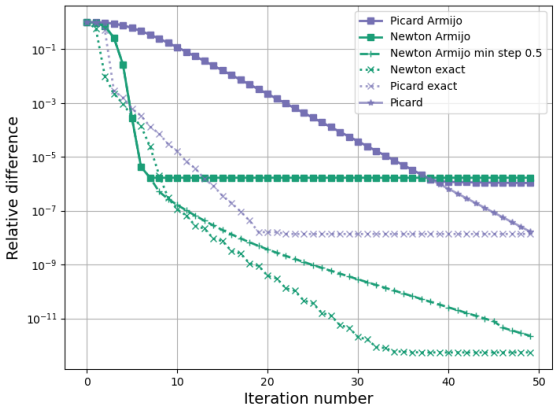

Figure 4. Relative difference compared to the reference solution for ISMIP-HOM B.

Figure 4 displays the relative difference over the iteration number. The classic Picard iteration has a slow convergence rate. It needs 39 iterations to obtain a reduction to 10^{-6}. Newton's method using Armijo step sizes obtains this reduction after only 7 iterations. This reduces the necessary number of iterations by 82 %. We see this even better if we consider just the first 9 iterations (Fig. 5). After a few iterations, Newton's method does not reduce the relative difference anymore. This could be seen

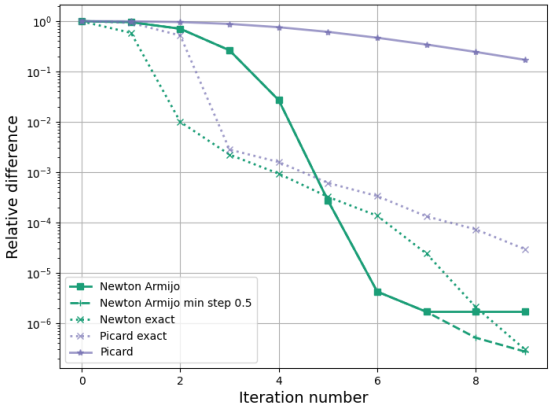

Figure 5. Relative difference compared to the reference solution for ISMIP-HOM B for the first 9 iterations.

as either a mistake in the step size control or Newton's method. However, the Picard iteration with Armijo step sizes has the same problem. It is identical to the Picard iteration up to iteration 39 and chooses then smaller step sizes to stall at a similar difference as Newton's method.

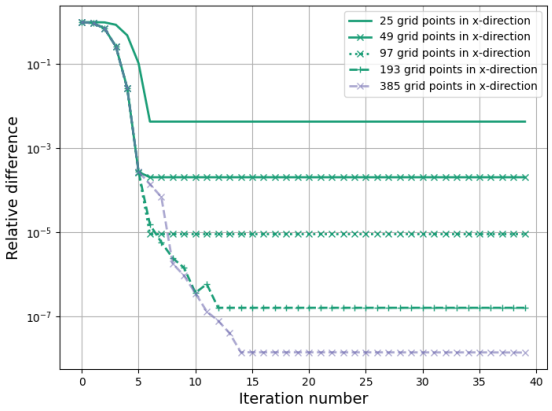

Figure 6. Experiments with 25, 49, 97, 193, and 385 grid poiints in x-direction and 3, 5, 9, 17, and 34 grid points in z-direction for the experiment ISMIP-HOM B for Newton's method with Armijo step sizes.

We think the reason for this behavior is that the discretized minimum of the convex functional and the root of the full-Stokes equations are slightly different. To verify this claim, we solved the full-Stokes equations on one grid and refinements of it, ensuring that all grid points on the coarser grids are also on the finer grids. We see the dependence on the resolution in Fig. 6. By halving the grid size, we reduce the relative difference by a bit more than a factor of 10. The increase in the relative difference from iteration 10 to 11 for the resolution with 193 grid points in the x-direction seems unintuitive. But, we remind that the Armijo step size control tries to minimize the functional not to find the root. Imposing a minimal step size of 0.5 helps to circumvent this problem. Then Newton's method reduces the relative difference up to iteration 39.

Also, Hirn (2013) reports accuracy problems for a small value of δ. Hirn considered a channel flow with $\rho \boldsymbol{g} = \mathbf{0}$ and $N \in \{2, 10/3, 5, 10\}$. The stopping criterium is not reached for $N = 10$ and for $N = 5$ for higher resolutions. In the second experiment, Hirn introduced $\delta > 0$ with $\delta = \delta_0 h^{2/(1+1/N)}$ and $\delta_0 \in \{1, 10\}$ and the mesh size h. Both variants converged to the wanted accuracy. Additionally, the calculated solutions for all resolutions were not too different from the analytical solution compared to the original problem with $\delta_0 = 0$. Finally, Hirn (2013) counted the number of Newton iterations to reach the wanted accuracy: The variants with $\delta_0 \in \{1, 10\}$ converge always and need a lesser number of iterations compared to $\delta_0 = 0$.

Thus, a higher δ value could lead to the expected quadratic convergence. In contrast, the exact step sizes do not seem to have this problem as they do not rely on evaluating the functional. Newton's method with exact step sizes has the advantage that the error reduces even more without using a minimal step size. Thus, one less parameter needs to be selected. Even the Picard iteration with exact step sizes is much better than the Picard iteration. It only needs 15 iterations to obtain the accuracy, for which the Picard iteration needs 39 iterations. That corresponds to a reduction of 62 %. The latter approach also has the advantage that there is no need to implement a new method to solve the problem. Only the relatively simple calculation of the

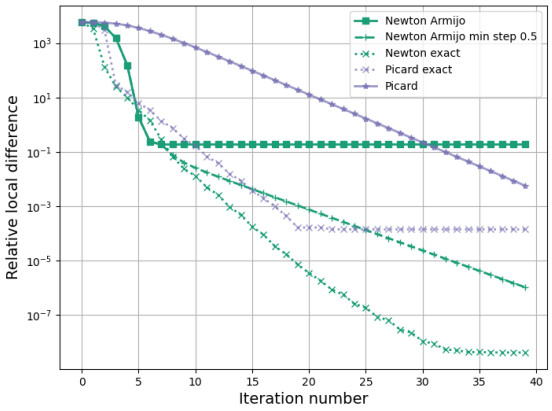

Figure 7. Relative local difference compared to the reference solution for ISMIP-HOM B.

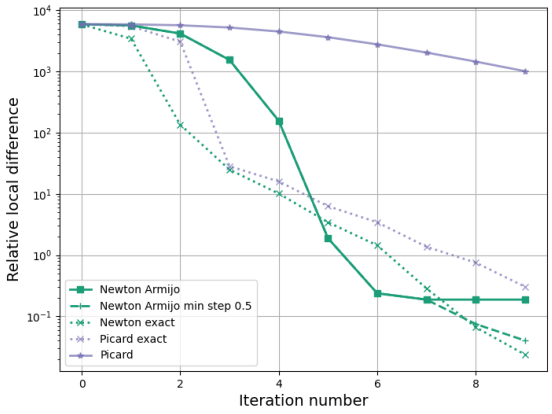

Figure 8. Relative local difference compared to the reference solution for ISMIP-HOM B for the first 9 iterations.

step sizes needs to be implemented. The results are really similar for our second measure of the accuracy, the relative local error, see Fig. 7.

All our algorithms are better than the classic Picard iteration in this measurement. The reduction with Newton's approach with both step size controls is 77 % now. The fast convergence is again impressive, especially for the first 9 iterations, see Fig. 8.

5.4 The experiment ISMIP-HOM A

Because real-world applications are three-dimensional, we consider experiment ISMIP-HOM A. This experiment extends ISMIP-HOM B to three dimensions. All chosen constants are the same as in the experiment ISMIP-HOM B. The experiment ISMIP-HOM A has a sinusoidal bottom in both horizontal dimensions. Again, we have three copies of the glacier in both horizontal directions. Thus, we have in total 48 copies. We describe the surface and bottom by

$$z_s(x,y) = -x\tan(\beta), \quad z_b(x,y) = z_s(x,y) - 1000 + 500 \cdot \sin(\omega x)\sin(\omega y). \tag{13}$$

5.5 Results for experiment ISMIP-HOM A

All our methods produce very similar results and are overlapping, see Fig. 9 and Fig. 10. Our simulations reproduce the surface velocity at $y = L/4$ from Pattyn et al. (2008) for the full-Stokes simulations for the majority of the glacier. But they produce higher velocity values than the mean plus the standard deviation around $x = L/3$. Nonetheless, the maximum relative difference is less than 0.02, see Fig. 9.

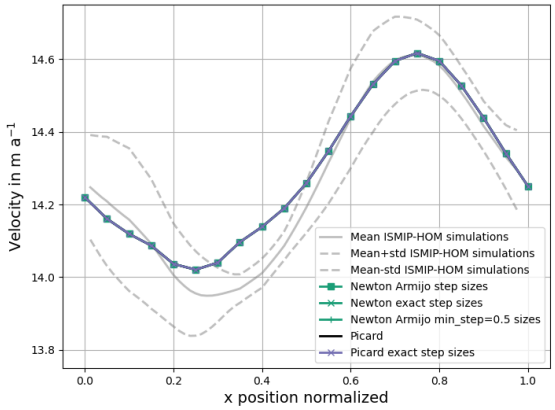

Figure 9. Simulated surface velocity at $y = L/4$ in meters per year for different solvers for ISMIP-HOM A. All our calculated velocity fields overlap each other. In grey are plotted the mean and the standard deviation from Pattyn et al. (2008) with 5 models. The mean and standard deviation have no values at $x = 0$ and $x = 1$ due to missing values.

The general convergence behavior for the three-dimensional experiment is similar to the two-dimensional experiment. However, the Armijo step sizes are even better for Newton's method in three dimensions, see Fig. 11. Again zooming to the first few iterations states the benefit from Newton's method and the step size control more impressing, see Fig. 12. The Picard iteration needs 39 iterations to have the same accuracy as Newton's method using Armijo step sizes after 6 iterations. Thus, the necessary number of iterations is reduced by more than 85 %. Again, a minimum step size of $\alpha = 0.5$ helps to reduce the relative difference after a few iterations. The exact step sizes for Newton's method are even better. They decrease the relative

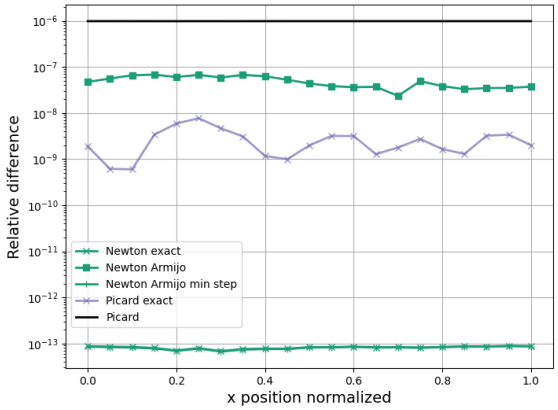

Figure 10. Relative difference of $|v - v_{\mathrm{ref}}|/v_{\mathrm{ref}}$ for each grid point at the surface at $y = L/4$ for ISMIP-HOM A. The reference solution is the solution from 80 Picard iterations. The relative difference for Newton with exact step sizes and Newton with Armijo step sizes and a minimal step of 0.5 are nearly identical.

difference, see Fig. 11, and the relative local difference, see Fig. 13, further than the Armijo step sizes. Also, exact step sizes improve the Picard iteration. Again it is interesting to consider the relative local difference for a few iterations, see Fig. 14. This figure emphasizes that the Picard iteration converges slowly compared to the other methods. As this experiment is more

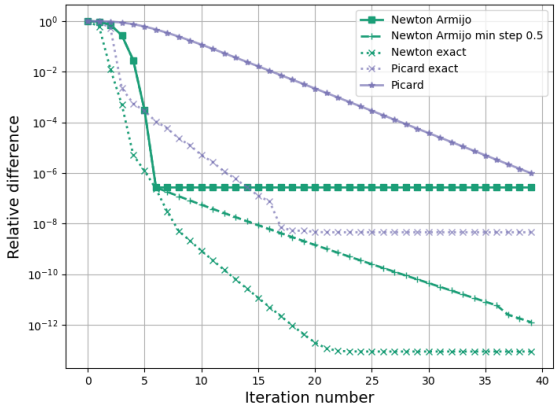

Figure 11. Relative difference compared to the reference solution for ISMIP-HOM A.

realistic regarding the number of grid points, we calculated the computation time for each iteration in experiment ISMIP-HOM A, see Table 1. There are two key findings: The computation time for Newton's method is about 20 % higher than for the Picard iteration. Additionally, the step size control is computationally cheap compared to Newton's method or the Picard iteration.

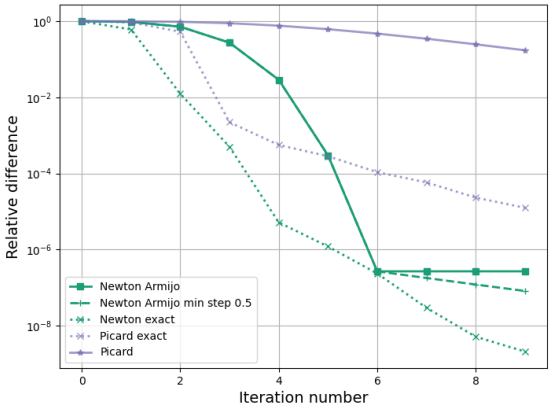

Figure 12. Relative difference compared to the reference solution for ISMIP-HOM A for the first 9 iterations.

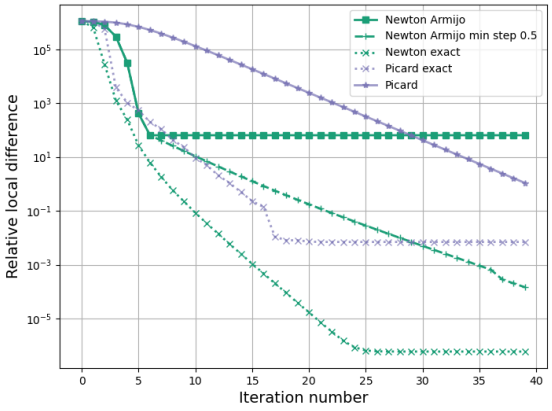

Figure 13. Relative local difference compared to the reference solution for ISMIP-HOM A.

However, the three-dimensional experiment has the additional uncertainty in precise computation times that we used the same processor type but a different processor.

6 Instationary experiments

In this section, we simulate a time-dependent version of the Haut Glacier d'Arolla without and with sliding. In a first step, we verify that our model produces similar results as in the experiments ISMIP-HOM $E1$ and $E2$, see Pattyn et al. (2008). The top and the bottom of the glacier are given by an input file. At the bottom, we have Dirichlet boundary conditions, and at the top, $\boldsymbol{\sigma} \cdot \boldsymbol{n} = \mathbf{0}$. The domain is represented in Fig. 15. In contrast to the stationary problems in Sect. 5, we do not have a reference

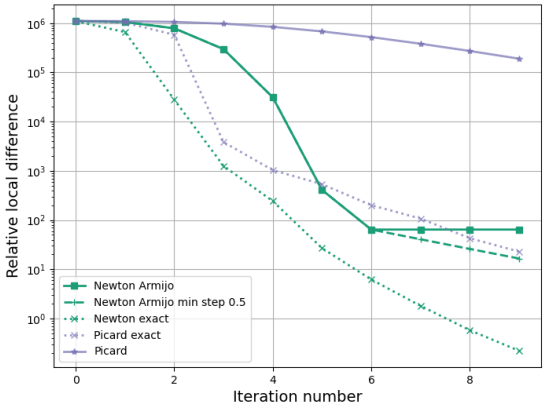

Figure 14. Relative local difference compared to the reference solution for ISMIP-HOM A for the first 9 iterations.

Table 1. Computation time in seconds without diagnostic calculations like the residual norm for the complete iteration.

	Complete iteration		Step size calculation	
	Mean	Standard deviation	Mean	Standard deviation
Picard	2226	20.3	-	-
Picard with exact step sizes	2286	11.0	60.4	0.18
Newton with Armijo step sizes and minimum step size=0.5	2706	6.77	4.60	0.45
Newton with exact step sizes	2757	18.5	60.1	0.24

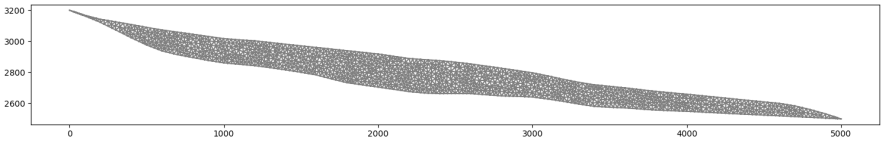

Figure 15. Domain of the Haut Glacier d'Arolla without sliding.

solution. A stopping criteria in (Gagliardini and Zwinger, 2008, section 2.4.2) is

$$2\frac{|\boldsymbol{u_{k+1}} - \boldsymbol{u_k}|}{|\boldsymbol{u_{k+1}}| + |\boldsymbol{u_k}|} < 10^{-5}. \tag{14}$$

To include the step size, we would multiply the right-hand side with α. This criterion is suitable to stop the iteration in a real simulation if the velocity field only has small changes. However, checking the relative difference let the Newton variants with step size control stop earlier in our test simulation without decreasing the error $\|G(\boldsymbol{v}, p)\|$ as much as the Picard variants did. As we compare solvers, we need to check if $G(\boldsymbol{v_k}, p_k)$ is close enough to zero. Thus, we check if we are close enough to our solution compared to the initial guess

$$\|G(\boldsymbol{v_k}, p_k)\| / \|G(\boldsymbol{v_0}, p_0)\| < \epsilon$$

with $\epsilon := 10^{-3}$. A relative stopping criteria seems necessary to reduce dependence on the domain and the absolute velocities. Thus, the calculated velocity field should have an error of 0.1 % compared to the initial guess after each time step. Our initial guess for time-dependent problems is the solution of a Stokes problem before the first step. After calculating a velocity field, the surface velocity determines the new surface. The grid points are moved according to the new surface. The initial guess for the velocity on the new domain is our velocity field shifted to the new domain. The stopping criterion has the advantage that we count the number of iterations needed to reduce the error by a certain factor. Therefore, the wanted error reduction is the same for all our solvers.

We know $G(\boldsymbol{v_k}, p_k) \in (H \times L)^*$. The Riesz isomorphism yields the existence of $(\tilde{\boldsymbol{v}}_{\boldsymbol{k}}, \tilde{p}_k) \in H \times L$ with

$$\int_\Omega \nabla \tilde{\boldsymbol{v}}_{\boldsymbol{k}} : \nabla \boldsymbol{\phi}\, dx + \int_\Omega \operatorname{div}(\boldsymbol{\phi}) \tilde{p}_k\, dx + \int_\Omega q \operatorname{div}(\tilde{\boldsymbol{v}}_{\boldsymbol{k}})\, dx = \langle G(\boldsymbol{v_k}, p_k), (\boldsymbol{\phi}, q) \rangle_{V_2^*, V_2} \quad \text{for all } (\boldsymbol{\phi}, q) \in H \times L. \tag{15}$$

Thus, we have to solve another Stokes problem in each iteration. Note that this Stokes problem is only diagnostic, and we do not need to solve it in practice. As the numerical analysis focuses on the velocity field, we calculate our error by

$$\|\tilde{\boldsymbol{v}}_{\boldsymbol{k}}\|_{V_2} = \sqrt{\int_\Omega |\nabla \tilde{\boldsymbol{v}}_{\boldsymbol{k}}|^2\, dx}. \tag{16}$$

For the experiment with sliding, we have to handle a difficulty arising in *FEniCS*: We can only force the boundary condition $\boldsymbol{v} \cdot \boldsymbol{n} = \boldsymbol{0}$ on horizontal and vertical boundaries. Thus, we use a lot of small stairs at the bottom instead of the slope in the original problem, see Fig. 16. On the bottom boundary with $2200 < x < 2500$, we employ the boundary condition $v_z = 0$ for $\boldsymbol{v} = (v_x, v_z)$.

6.1 Stationary solutions

We only discuss the accuracy of our model in simulating the experiment ISMIP-HOM $E1$ and $E2$ without considering convergence speed. We discuss the convergence speed for the time-dependent problems. The simulation of the velocity field is quite similar to Pattyn et al. (2008), see Fig. 17. Our velocity field at the surface is mostly within the mean with the standard deviation of the reference solutions, see Fig. 17. In some small parts, the velocity is slightly lower.

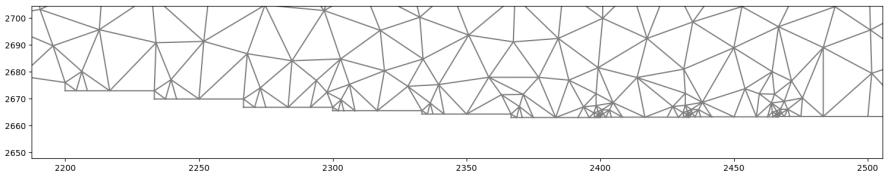

Figure 16. Stair-shaped domain at the bottom.

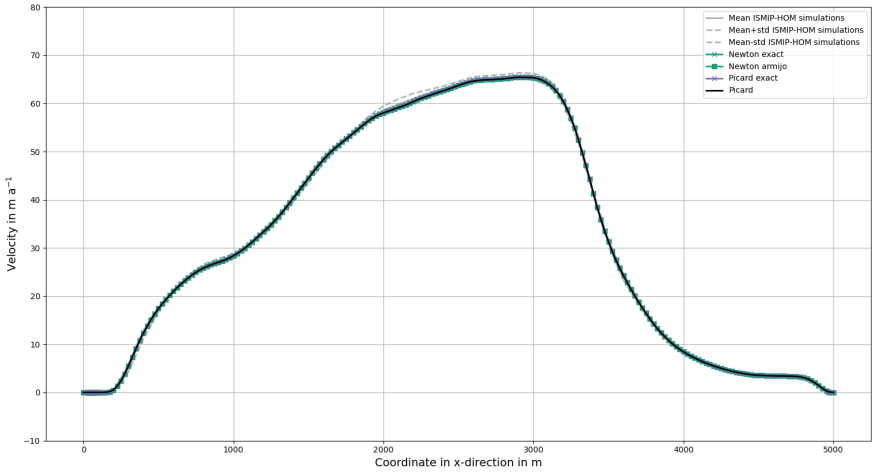

Figure 17. Surface velocity field of the Haut Glacier d'Arolla without sliding.

The calculated velocity field is near the mean minus the standard deviation of the reference solutions in Pattyn et al. (2008), see Fig. 18. Often, it is even a bit less. However, it is still a suitable approximation, and we use both problems for the time-dependent simulation.

6.2 Time dependent problem - mass transport

For the instationary problem, the surface develops dependent on the velocity field. In our case, we describe the height of the glacier by

$$\frac{\partial z(x)}{\partial t} + v_x(x)\frac{\partial z(x)}{\partial x} - v_z(x) = 0 \quad \text{for } x \in (0, 5000] \tag{17}$$

see Pattyn et al. (2008). The height is fixed at $x = 0$. Let $(x_i)_{i=0}^N$ be the discretization with $x_0 = 0$ and $x_N = 5000$. We approximate the spatial differential quotient by an upwinding scheme:

$$v_x(x_i)\frac{\partial z(x_i)}{\partial x} \approx \begin{cases} v_x(x_i)\frac{z(x_i)-z(x_{i-1})}{x_i - x_{i-1}} & \text{for } v_x(x_i) > 0 \text{ and } i > 0, \\ v_x(x_i)\frac{z(x_{i+1})-z(x_i))}{x_{i+1} - x_i} & \text{for } v_x(x_i) \leq 0 \text{ and } i < N. \end{cases} \tag{18}$$

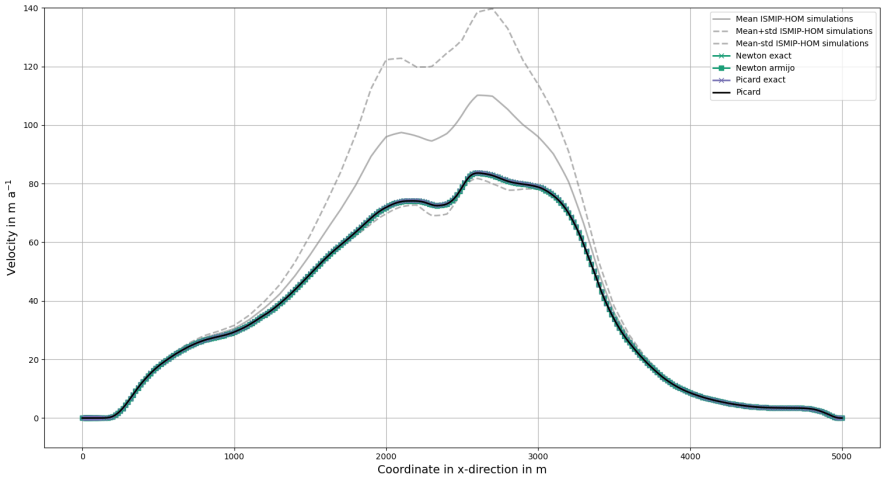

Figure 18. Surface velocity field of the Haut Glacier d'Arolla with sliding.

The upwinding scheme stabilizes the solution of the discretized Eq. (17). Moreover, it helped in our experiments for the conversation of mass compared to the forward difference quotient and the central difference quotient. We use an explicit Euler method in time and conclude for the $k+1$-th time step:

$$\frac{z^{k+1}-z^k}{\Delta t} + v_x^k(x)\frac{\partial z^k(x)}{\partial x} - v_z^k(x) = 0. \tag{19}$$

Together, we obtain for the $k+1$-th time step and the i-th grid point at the surface

$$z_i^{k+1} = z_i^k + \Delta t \left(v_z^k(x_i) - \begin{cases} v_x^k(x_i)\frac{z^k(x_i)-z^k(x_{i-1})}{x_i-x_{i-1}} & \text{for } v_x(x_i) > 0 \text{ and } i > 0, \\ v_x^k(x_i)\frac{z^k(x_{i+1})-z^k(x_i)}{x_{i+1}-x_i} & \text{for } v_x(x_i) \leq 0 \text{ and } i < N, \\ 0 & \text{for } i = 0. \end{cases} \right). \tag{20}$$

In our problem, the value $z(x_0)$ is fixed. Mathematically, we are not allowed to fix $z(x_N)$ because this value is determined by Eq. (17). Therefore, we add a grid point at $(5000, 2505)$, slightly above the bottom $(5000, 2500)$. We impose $\boldsymbol{\sigma} \cdot \boldsymbol{n} = 0$ on the newly generated right boundary. Hence, the mass can flow outside the glacier or physically interpreted ice is melting.

We calculate over 30 years to simulate a changing velocity field with the highest surface velocity at the right edge of the domain. We choose a time step size of 0.25 years to fulfill the CFL condition for the experiment $E1$. In experiment $E2$, we choose the same time step sizes to have a comparable experiment.

6.3 Time-dependent simulation without friction

In this subsection, we visualize the velocity field of the glacier at the surface over the time simulation and discuss the computational effort for the experiment without friction.

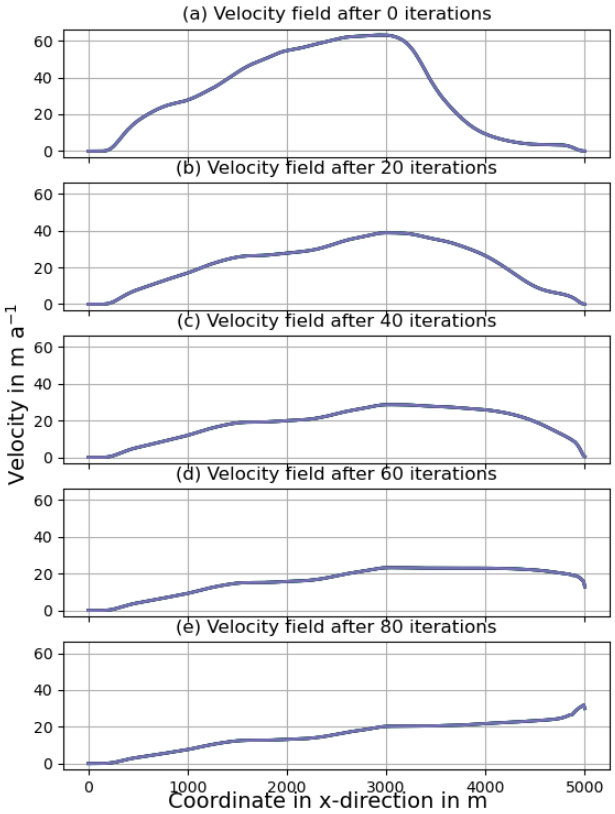

Figure 19. Surface velocity fields for the Picard iteration, the Picard iteration with exact step sizes, Newton's method with Armijo step sizes, and Newton's method with exact step sizes. The velocity fields for the different methods are nearly identical.

All simulations produce similar surface velocities over time, see Fig. 19. Thus, all methods seem to calculate the solution appropriately. Now, we discuss the computational effort. The number of iterations needed is shown in Fig. 20. We set the maximum number of iterations to 50. Newton's method with Armijo step sizes and a minimum step size of 0.5 has the problem that it does not always converge. Too small step sizes were chosen without the minimum step size, which yielded no convergence, too. The necessary number of Newton steps with exact step sizes varies a lot from time step to time step compared with both Picard variants. In Table 2, we see that Newton's method with exact step sizes performs best. However, it has a larger standard deviation than the Picard variants. The Picard iteration with exact step sizes needs only about two more iterations. Newton's method with Armijo steps has a really large standard deviation.

We measured the computation time for the time-dependent experiment, see Table 3. The computation of the step sizes takes 9 % or less of the total computation time for each iteration. The computation time for calculating diagnostics like the residual norm was not measured as it is unnecessary for the application.

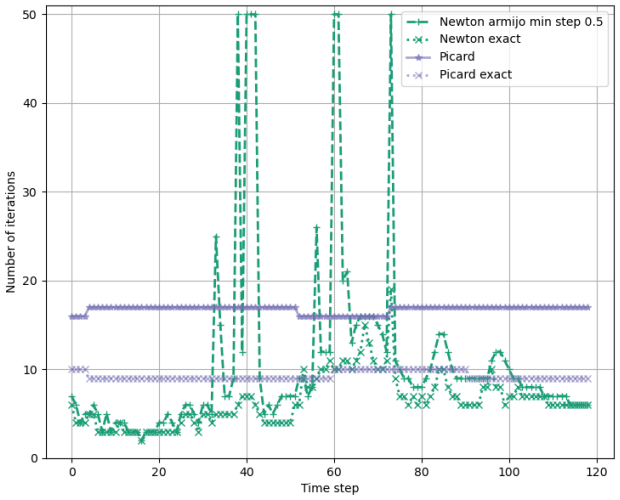

Figure 20. Number of iterations to solve the full-Stokes equations between each time step.

Table 2. Number of iterations for solving the full-Stokes equations in each time step.

	Time steps	Mean	Standard deviation
Picard	119	16.79	0.41
Picard with exact step sizes	119	9.29	0.46
Newton with Armijo step sizes and minimum step size=0.5	119	10.92	10.67
Newton with exact step sizes	119	6.45	2.83

Table 3. Computation time in seconds for the complete iteration without diagnostic calculations like the residual norm.

		Complete iteration		Step size calculation	
	Iterations	Mean	Standard deviation	Mean	Standard deviation
Picard	1998	5.61	0.61	-	-
Picard with exact step sizes	1106	6.00	0.59	0.48	0.03
Newton with Armijo step sizes and minimum step size=0.5	1299	5.73	0.60	0.02	0.01
Newton with exact step sizes	767	6.01	0.57	0.48	0.03

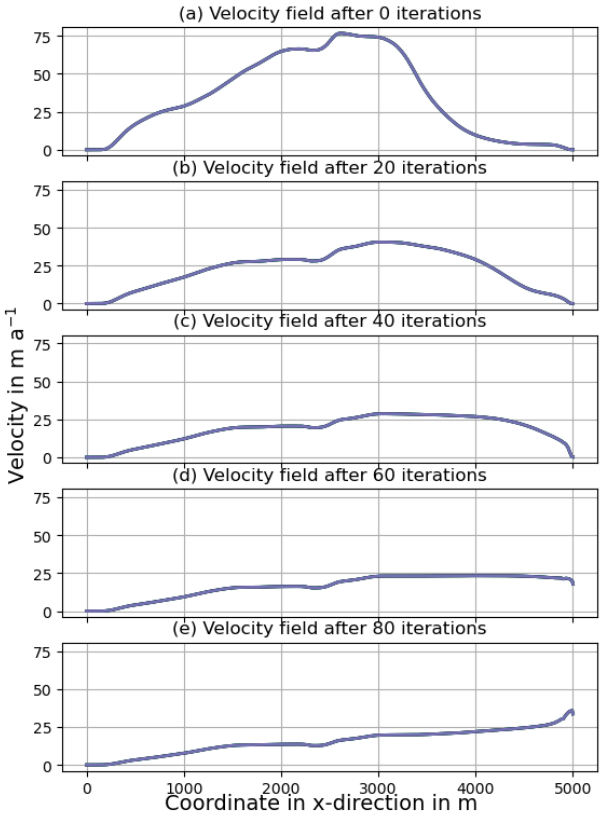

Figure 21. Surface velocity fields for the Picard iteration, the Picard iteration with exact step sizes, Newton's method with Armijo step sizes, and Newton's method with exact step sizes. The velocity fields for the different methods are nearly identical.

6.4 Time-dependent simulation with friction

In this subsection, we visualize the velocity field of the glacier at the surface over the time simulation and discuss the computational effort for the experiment with friction.

All simulations produce similar surface velocities over time, see Fig. 21. Thus, all methods seem to calculate the solution appropriately. Now, we discuss the computational effort. The number of iterations needed is shown in Fig. 22. We set the maximum number of iterations to 50. Newton's method with Armijo step sizes and a minimum step size of 0.5 has the problem that it does not always converge. Too small step sizes were chosen without the minimum step size, which yielded no convergence, too. The necessary number of Newton steps with exact step sizes varies a lot from time step to time step compared with both Picard variants. Interestingly, the Picard variants need nearly the same number of iterations with sliding as without sliding. There are also two time steps in which Newton's method with exact step sizes does not converge. In Table 4, we see that the Picard iteration with exact step sizes performes best. It also has the lowest standard deviation.

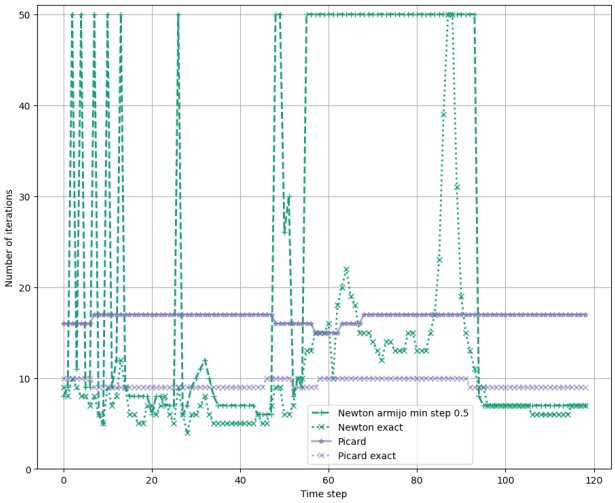

Figure 22. Number of iterations to solve the full-Stokes equations between each time step.

Table 4. Number of iterations for solving the full-Stokes equations in each time step.

	Time steps	Mean	Standard deviation
Picard	119	16.72	0.55
Picard with exact step sizes	119	9.39	0.49
Newton with Armijo step sizes and minimum step size=0.5	119	24.70	20.64
Newton with exact step sizes	119	10.42	7.48

We measured the computation time for the time-dependent simulation, see Table 5. The computation time for calculating diagnostics like the residual norm was not measured as it is unnecessary for the application. The computation of the step sizes takes 6.5 % or less of the total computation time for each iteration.

Table 5. Computation time in seconds for the complete iteration without diagnostic calculations like the residual norm.

		Computation time each iteration		Computation time step size	
	Iterations	Mean	Standard deviation	Mean	Standard deviation
Picard	1990	10.93	0.77	-	-
Picard with exact step sizes	1118	11.31	0.85	0.30	0.01
Newton with Armijo step sizes and minimum step size=0.5	2939	11.22	0.86	0.03	0.01
Newton with exact step sizes	1240	11.62	0.75	0.67	0.03

7 Summary and conclusion

Solving the full-Stokes equations is equivalent to minimizing a function. We use this function to introduce approximately exact step sizes. For a comparison, we also use Armijo step sizes. We test the algorithms for benchmark experiments with a sinusoidal bottom in two and three dimensions, for the Glacier d'Arolla benchmark experiment, and for a time-dependent variant of the Glacier d'Arolla experiment with and without sliding.

We observe that our calculated solutions are similar to those in Pattyn et al. (2008). However, the approximately exact step sizes greatly improve the convergence speed of the Picard iteration and ensure convergence of Newton's method for nearly all situations except two cases in the time-dependent simulation with friction, see Fig. 22. Thus, the approximately exact step sizes seem to be better than the Armijo step sizes.

The computation time of the step sizes is only a small part of the complete iteration. In our experiments, the time for calculating the step sizes took $9\ \%$, $6.5\ \%$, and $3\ \%$. The ratio is even smaller for higher resolutions. The concrete computation times in seconds are irrelevant as they depend on the hardware.

8 Outlook

The effort to implement the algorithms above is relatively low. For every additional boundary condition to those above, one has to check if a convex function exists. One only needs to implement these convex functions, the directional derivatives, and the Armijo and exact step sizes, respectively. The Picard iteration or Newton's method should already be implemented for solving the full-Stokes equations.

There are a few possible directions to work on: The computation of the step size could be done more efficiently by parallelizing the calculation of the integrals and testing how many bisections are necessary for calculating the exact step sizes.

Also, more realistic three-dimensional examples or different sliding laws could be tested. The mathematical theory for a nonlinear sliding boundary condition is discussed in Schmidt (2023).

The implementation in ice models is another way to check if the presented algorithms work in real-world applications. Lastly, the step size control might reduce the number of iterations for the Higher-Order equations. Solving those equations is also equivalent to finding the minimum of a convex function, (see Schoof, 2010).

Code and data availability. The model is available at https://doi.org/10.5281/zenodo.10618661. The latest version of the source code is available at https://github.com/Niko-ich/FEniCS-full-Stokes.

Appendix A: Mathematical derivations

A1 The variational formulation

For the well-posedness of the Picard iteration, see Algorithm 1,

$$B\big(|D\boldsymbol{v_k}|^2+\delta^2\big)^{\frac{1-n}{2n}} D\boldsymbol{v_{k+1}} : \nabla\boldsymbol{\phi} \tag{A1}$$

has to be integrable. Due to a bounded ice rheology B and $\delta > 0$, follows the boundedness with $c \in \mathbb{R}$ and

$$B\big(|D\boldsymbol{v_k}|^2+\delta^2\big)^{\frac{1-n}{2n}} < c. \tag{A2}$$

Thus, we need for integrability $D\boldsymbol{v_{k+1}} : \nabla\boldsymbol{\phi} \in L^1(\Omega)$. This is only fulfilled for $D\boldsymbol{v_{k+1}} \in L^p(\Omega)^{N\times N}$ and $\nabla\boldsymbol{\phi} \in L^q(\Omega)^{N\times N}$ with $1/p+1/q=1$. Hence, we can not use $\boldsymbol{v_{k+1}}, \boldsymbol{\phi} \in \{\boldsymbol{v} \in W^{1,1+1/n}(\Omega)^N; \boldsymbol{v}|_{\Gamma_b\cup\Gamma_\ell} = \boldsymbol{0}\}$, see (Belenki et al., 2012, section 2.3), which is the suitable space for $\mu_0 = 0$ as it allows the proof of existence and uniqueness of the solution.

However, expression $(A1)$ is well-defined for $\boldsymbol{v_{k+1}}, \boldsymbol{\phi} \in H = \{\boldsymbol{v} \in H^1(\Omega)^N; \boldsymbol{v}|_{\Gamma_b\cup\Gamma_\ell} = \boldsymbol{0}\}$. The additional diffusion term with $\mu_0 > 0$ verifies that the solution of the full-Stokes equations is in H. Similar reasons make the diffusion term necessary for Newton's method: The directional derivative, see equation (6) is only defined for $\boldsymbol{v}, \boldsymbol{w}, \boldsymbol{\phi} \in H$.

A2 The directional derivative of G

In this subsection, we compute the derivative of G at the velocity $\boldsymbol{v} \in H$ and pressure $p \in L$ in the direction $\boldsymbol{w} \in H$ and $q \in L$ with the diffusion $\mu_0 > 0$. Because we have a variational formulation, we can only interpret this derivative for test functions $\boldsymbol{\phi} \in H$ and $\psi \in L$. We calculate

$$\begin{aligned}
&\langle G'(\boldsymbol{v},p)(\boldsymbol{w},q),(\boldsymbol{\phi},\psi)\rangle \\
&= \lim_{t\to 0} \frac{\langle G(\boldsymbol{v}+t\boldsymbol{w},p+tq),(\boldsymbol{\phi},\psi)\rangle - \langle G(\boldsymbol{v},p),(\boldsymbol{\phi},\psi\rangle}{t} \\
&= \lim_{t\to 0} \int_\Omega \frac{B}{t}\Big(\big(|D(\boldsymbol{v}+t\boldsymbol{w})|^2+\delta^2\big)^{\frac{1-n}{2n}} D\boldsymbol{v} - \big(|D\boldsymbol{v}|^2+\delta^2\big)^{\frac{1-n}{2n}} D\boldsymbol{v}\Big) : \nabla\boldsymbol{\phi}\, dx \\
&\quad + \lim_{t\to 0} \int_\Omega \frac{B}{t}\Big(\big(|D(\boldsymbol{v}+t\boldsymbol{w})|^2+\delta^2\big)^{(1-n)/(2n)} tD\boldsymbol{w}\Big) : \nabla\boldsymbol{\phi}\, dx \\
&\quad + \lim_{t\to 0} \mu_0 \int_\Omega \nabla\left(\frac{\boldsymbol{v}+t\boldsymbol{w}-\boldsymbol{v}}{t}\right) : \nabla\boldsymbol{\phi}\, dx - \int_\Omega \frac{p+tq-p}{t}\mathrm{div}\boldsymbol{\phi}\, dx - \int_\Omega \mathrm{div}\left(\frac{\boldsymbol{v}+t\boldsymbol{w}-\boldsymbol{v}}{t}\right)\psi\, dx.
\end{aligned} \tag{A3}$$

The limits for the second and third lines on the right-hand side of the last equality are clear. For the first line, we use the Taylor expansion. Therefore, we define the function $f_x : [0,\infty) \to \mathbb{R}$,

$$f_x(t) = \big(|D(\boldsymbol{v}(x)+t\boldsymbol{w}(x))|^2+\delta^2\big)^{\frac{1-n}{2n}}. \tag{A4}$$

Its derivative is

$$f_x'(t) = \frac{1-n}{n}\big(|D(\boldsymbol{v}(x)+t\boldsymbol{w}(x))|^2+\delta^2\big)^{\frac{1-3n}{2n}} (D\boldsymbol{v}(x) : D\boldsymbol{w}(x) + t|D\boldsymbol{w}(x)|^2). \tag{A5}$$

We calculate the derivative by assuming we can draw the limes into the integral. A detailed explanation of why we can do this is in Schmidt (2023). We obtain with $\xi : \Omega \to [0,t]$ for the Taylor expansion

$$
\begin{aligned}
&\int_\Omega \lim_{t\to 0} \frac{B}{t} \Big(\big(|D(\boldsymbol{v}+t\boldsymbol{w})|^2+\delta^2\big)^{\frac{1-n}{2n}} D\boldsymbol{v} - \big(|D\boldsymbol{v}|^2+\delta^2\big)^{\frac{1-n}{2n}} D\boldsymbol{v} \Big) : \nabla\boldsymbol{\phi}\, dx \\
&= \int_\Omega \lim_{t\to 0} \frac{B}{t} \big(f_x(t) - f_x(0)\big) D\boldsymbol{v} : \nabla\boldsymbol{\phi}\, dx \\
&= \int_\Omega \lim_{t\to 0} \frac{B}{t} f_x'(\xi(x)) t D\boldsymbol{v} : \nabla\boldsymbol{\phi}\, dx \\
&= \int_\Omega \lim_{t\to 0} B\frac{1-n}{n} \big(|D\boldsymbol{v}(x)+\xi(x)\boldsymbol{w}(x)|^2+\delta^2\big)^{\frac{1-3n}{2n}} (D\boldsymbol{v}(x) : D\boldsymbol{w}(x) + \xi(x)|D\boldsymbol{w}(x)|^2) D\boldsymbol{v}(x) : \nabla\boldsymbol{\phi}(x)\, dx \\
&= \int_\Omega B\frac{1-n}{n} \big(|D\boldsymbol{v}|^2+\delta^2\big)^{\frac{1-3n}{2n}} (D\boldsymbol{v} : D\boldsymbol{w})\, (D\boldsymbol{v} : \nabla\boldsymbol{\phi})\, dx. \qquad \text{(A6)}
\end{aligned}
$$

Author contributions.

The manuscript is written by NS with contributions and suggestions from TS and AH. The code is implemented by NS with support from TS and AH.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. The authors appreciate helpful explanations of the ISMIP-HOM experiments from Dr. Martin Rückamp from the Bavarian Academy of Sciences and Humanities, and Dr. Thomas Kleiner from Alfred-Wegener-Institut in Bremerhaven. We also thank Prof. Mathieu Morlighem from the Dartmouth College in Hanover, New Hampshire for answering questions about *ISSM* and Dr. Fabien Gillet-Chaulet from the Grenoble Alpes University for answering questions about *Elmer/Ice*. Lastly, we thank the reviewers for suggesting experiments to increase the relevance of the manuscript and other helpful ideas to improve the manuscript.

References

COMSOL Multiphysics Reference Manual, https://doc.comsol.com/5.4/doc/com.comsol.help.comsol/COMSOL_ReferenceManual.pdf, 2018.

Belenki, L., Berselli, L. C., Diening, L., and Růžička, M.: ON THE FINITE ELEMENT APPROXIMATION OF p-STOKES SYSTEMS, SIAM Journal on Numerical Analysis, 50, 373–397, http://www.jstor.org/stable/41582741, 2012.

Chen, Q., Gunzburger, M., and Perego, M.: Well-Posedness Results for a Nonlinear Stokes Problem Arising in Glaciology, SIAM Journal on Mathematical Analysis, 45, 2710–2733, https://doi.org/10.1137/110848694, 2013.

Colinge, J. and Rappaz, J.: A strongly nonlinear problem arising in glaciology, ESAIM: Mathematical Modelling and Numerical Analysis, 33, 395–406, https://doi.org/10.1051/m2an:1999122, 1999.

Dennis, J. and Schnabel, R.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Society for Industrial and Applied Mathematics, Philadelphia, https://doi.org/10.1137/1.9781611971200, 1996.

Fraters, M. R. T., Bangerth, W., Thieulot, C., Glerum, A. C., and Spakman, W.: Efficient and practical Newton solvers for non-linear Stokes systems in geodynamic problems, Geophysical Journal International, 218, 873–894, https://doi.org/10.1093/gji/ggz183, 2019.

Gagliardini, O. and Zwinger, T.: The ISMIP-HOM benchmark experiments performed using the Finite-Element code Elmer, The Cryosphere, 2, 67–76, https://doi.org/10.5194/tc-2-67-2008, 2008.

Gagliardini, O., Zwinger, T., Gillet-Chaulet, F., Durand, G., Favier, L., de Fleurian, B., Greve, R., Malinen, M., Martín, C., Råback, P., Ruokolainen, J., Sacchettini, M., Schäfer, M., Seddik, H., and Thies, J.: Capabilities and performance of Elmer/Ice, a new-generation ice sheet model, Geoscientific Model Development, 6, 1299–1318, https://doi.org/10.5194/gmd-6-1299-2013, 2013.

Habbal, F., Larour, E., Morlighem, M., Seroussi, H., Borstad, C. P., and Rignot, E.: Optimal numerical solvers for transient simulations of ice flow using the Ice Sheet System Model (ISSM versions 4.2.5 and 4.11), Geoscientific Model Development, 10, 155–168, https://doi.org/10.5194/gmd-10-155-2017, 2017.

Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S.: Optimization with PDE Constraints, Springer Netherlands, https://doi.org/10.1007/978-1-4020-8839-1, 2009.

Hirn, A.: Finite element approximation of singular power-law systems, Mathematics of Computation, 82, 1247–1268, http://www.jstor.org/stable/42002697, 2013.

Joughin, I., Smith, B. E., Howat, I. M., Scambos, T., and Moon, T.: Greenland flow variability from ice-sheet-wide velocity mapping, Journal of Glaciology, 56, 415–430, https://doi.org/10.3189/002214310792447734, 2010.

Larour, E., Seroussi, H., Morlighem, M., and Rignot, E.: Continental scale, high order, high spatial resolution, ice sheet modeling using the Ice Sheet System Model (ISSM), Journal of Geophysical Research: Earth Surface, 117, n/a–n/a, https://doi.org/10.1029/2011jf002140, 2012.

Leng, W., Ju, L., Gunzburger, M., Price, S., and Ringler, T.: A parallel high-order accurate finite element nonlinear Stokes ice sheet model and benchmark experiments, Journal of Geophysical Research: Earth Surface, 117, n/a–n/a, https://doi.org/10.1029/2011jf001962, 2012.

Logg, A., Mardal, K.-A., and Wells, G., eds.: Automated Solution of Differential Equations by the Finite Element Method, Springer Berlin Heidelberg, https://doi.org/10.1007/978-3-642-23099-8, 2012.

Nocedal, J. and Wright, S.: Numerical Optimization, Springer series in operations research and financial engineering, Springer, New York, second edn., 2006.

Pattyn, F., Perichon, L., Aschwanden, A., Breuer, B., de Smedt, B., Gagliardini, O., Gudmundsson, G. H., Hindmarsh, R. C. A., Hubbard, A., Johnson, J. V., Kleiner, T., Konovalov, Y., Martin, C., Payne, A. J., Pollard, D., Price, S., Rückamp, M., Saito, F., Souček, O., Sugiyama, S., and Zwinger, T.: Benchmark experiments for higher-order and full-Stokes ice sheet models (ISMIP–HOM), The Cryosphere, 2, 95–108, https://doi.org/10.5194/tc-2-95-2008, 2008.

Rückamp, M., Kleiner, T., and Humbert, A.: Comparison of ice dynamics using full-Stokes and Blatter–Pattyn approximation: application to the Northeast Greenland Ice Stream, The Cryosphere, 16, 1675–1696, https://doi.org/10.5194/tc-16-1675-2022, 2022.

Schmidt, N.: Global Convergence of the Infinite-Dimensional Newton's Method for the Regularized P-Stokes Equations, https://doi.org/10.21203/rs.3.rs-3354498/v1, 2023.

Schoof, C.: Coloumb friction and other sliding laws in a higher-order glacier flow model, Mathematical Models and Methods in Applied Sciences, 20, 157–189, https://doi.org/10.1142/s0218202510004180, 2010.

Zhang, T., Price, S., Ju, L., Leng, W., Brondex, J., Durand, G., and Gagliardini, O.: A comparison of two Stokes ice sheet models applied to the Marine Ice Sheet Model Intercomparison Project for plan view models (MISMIP3d), The Cryosphere, 11, 179–190, https://doi.org/10.5194/tc-11-179-2017, 2017.