A Fully Implicit Second Order Method for Viscous Free Surface Stokes Flow - Application to Glacier Simulations

Josefin Ahlkrona[1,2,3], A. Clara J. Henry[1,2], and André Löfgren[1,2]

[1]Department of Mathematics, Stockholm University, Stockholm, Sweden
[2]Bolin Centre for Climate Research, Stockholm University, Stockholm, Sweden
[3]Swedish e-Science Research Centre, Stockholm, Sweden

Correspondence: Josefin Ahlkrona (ahlkrona@math.su.se)

Abstract. We present a fully implicit finite element method for viscous free surface Stokes problems which are common in glaciology and geodynamics. Standard fully implicit iterative solvers diverge for such problems when using large time steps. We address this by incorporating a stabilization term that vanishes upon convergence. This approach enables the use of large time steps, thereby significantly improving the efficiency of simulations. Furthermore, we leverage the fully implicit solver to develop second-order time-stepping schemes that mitigate the high errors associated with first-order methods for large time steps. The methods are tested and evaluated on model domains relevant to geodynamics and glaciology.

1 Introduction

Viscous, gravity-driven, free surface problems are prevalent in geoscientific applications such as mantle convection and glaciology (Greve and Blatter, 2009; Rose et al., 2017). Such simulations involve solving the Stokes equations for the velocity and pressure, $(\mathbf{u}(x,y,z,t), p(x,y,z,t))$, and a free-surface equation for the upper surface height $h(x,y,t)$ that determines the geometry, see Fig. 1. A key computational challenge arises from the strong coupling between the evolving geometry and the velocity field, which can lead to numerical stability issues and imposes severe restrictions on the time-step size, ultimately reducing computational efficiency. Fully implicit methods would overcome these time-step limitations, but the strong coupling introduces convergence issues that prevent practical usage.

In ice-sheet and glacier modeling, the time-step size restriction is a prominent problem (Cheng et al., 2017; Löfgren et al., 2022; Löfgren et al., 2024). Ice-sheet and glacier models describe the movement of the ice covering Antarctica and Greenland, or that of smaller mountain glaciers and ice caps. Under the long time scales that are relevant to ice dynamics, ice deforms as a viscous fluid. It slowly creeps under its own weight from the interior of a continent to the coast, where it melts due to lower altitudes and meeting the warm ocean. Accurately calculating the geometry and the rate of flow towards the ocean is crucial in estimating the future rise in sea level in response to climate change. However, the computational challenges of ice-sheet modeling limits the accuracy of projections (Pörtner et al., 2019). Due to the coupling between geometry and ice velocity, the full problem can be likened to solving the heat equation, but the state-of-the-art numerical methods are explicit in time. To maintain stability, time-steps are therefore often limited to around $0.1 - 1.0$ years Leguy et al. (2021), which is a bottleneck. Implicit iterations would increase the largest stable time-steps but are challenging and have so far only been achieved for

25 approximate models (Bueler, 2016). For models based on the full Stokes equations there is, to our knowledge, no working implicit method. However, the great promise of such methods was recognized by (Bueler, 2022) who later conjectured that such schemes should be well-posed (Bueler, 2024).

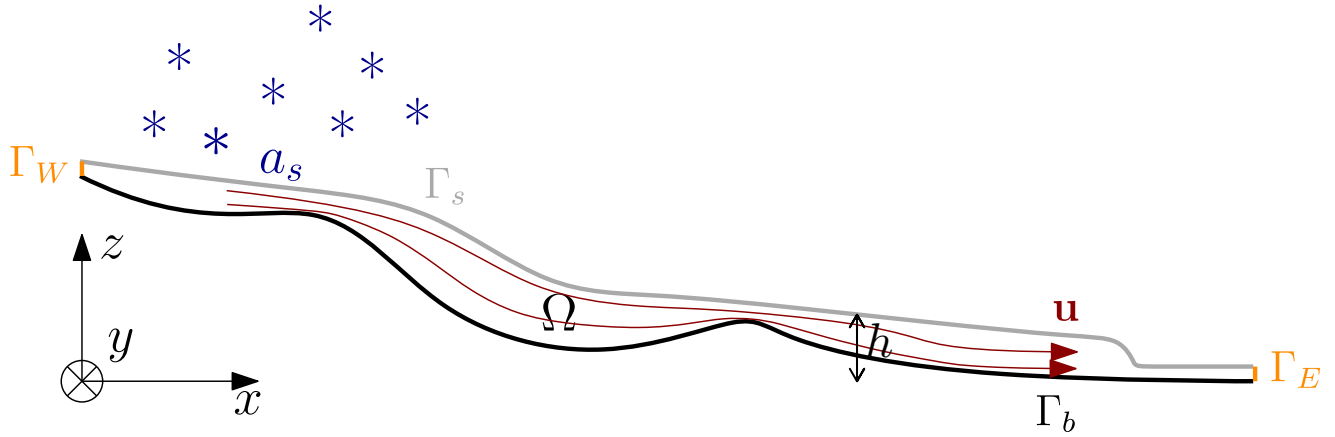



Figure 1. Illustration of a viscous free surface problem in the form of a mountain glacier. The ice is formed by accumulated snow and flows down the mountain due to the force of gravity. The surface height h indicates the distance from $z = 0$ to the surface Γ_s. Adapted from Löfgren et al. (2024), published in *The Cryosphere*, licensed under CC BY 4.0.

In this paper, we present a numerically stable and accurate fully implicit method by introducing boundary terms in the Stokes equations, which vanish during convergence. We argue that the convergence issues of standard implicit iterations are connected

30 to the stability issues that are seen for explicit methods (since the first iteration has to be explicit), and we introduce a method that circumvents these issues in a computationally efficient way. The method relies on the Free Surface Stabilization Algorithm (FSSA), which was originally developed for mantle convection problems (Kaus et al., 2010) and later adapted for ice sheet simulations (Löfgren et al., 2022; Löfgren et al., 2024). The FSSA relaxes the time-step restrictions for explicit solvers by adding additional boundary terms. We adapt this method to the context of fully-implicit solvers. The boundary term vanish as

35 convergence is reached. This method allows for very large time-steps and ensures consistency. We then utilize the new implicit method to construct second-order accurate time-stepping schemes and demonstrate efficiency and accuracy on setups relevant to geodynamics and glacier simulations.

2 Governing Equations and their Finite Element Discretization

The surface height position h, the velocity $\mathbf{u} = (u_x, u_y, u_z)$ and the pressure p are the solution to a system of coupled partial
differential equations, namely the $\mathfrak{p}$-Stokes-coupled free-surface equation:

$$\left.\begin{aligned} -\nabla \cdot \mathbf{S}(\mathbf{Du}) + \nabla p &= \rho \mathbf{g}, \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned}\right\} \quad \text{in } \Omega(t), \tag{1}$$

$$\frac{\partial h}{\partial t} + \mathbf{u}(x, y, z = h, t) \cdot \nabla h = a_s \quad \text{on } \omega, \tag{2}$$

Here, $\Omega \subset \mathbb{R}^d$ with $d \in \{2, 3\}$, is an open and bounded domain with a boundary Γ, of which one part is the upper free surface
Γ_s which is located at a height h, see Fig. 1 for an illustration from glaciology. The free-surface equation (2) is solved on
$\omega \in \mathbb{R}^{d-1}$, which represents a projection of the upper ice surface Γ_s onto a flat two-dimensional plane. The ice-flow dynamics
are driven by the gravitational term $\rho \mathbf{g}$. The free-surface equation can be forced by the term a_s, which in a glaciological context
represents accumulation and ablation. The deviatoric stress tensor $\mathbf{S}$ depends on the strain-rate tensor $2\mathbf{Du} = \nabla \mathbf{u} + (\nabla \mathbf{u})^T$
according to a constitutive law

$$\mathbf{S}(\mathbf{u}) = \nu(\mathbf{Du})\mathbf{Du}, \tag{3}$$

where

$$\nu := \nu(\mathbf{Du}) = \nu_0 (\varepsilon^2 + |\mathbf{Du}|^2)^{\frac{\mathfrak{p}-2}{2}} \quad \mathfrak{p} \in (1, 2], \tag{4}$$

is the viscosity function, dependent on the effective strain rate $|\mathbf{Du}|$; ε^2 is a regularisation term to avoid infinite viscosity at
zero shear strain rate. The relation is linear for $\mathfrak{p} = 2$, as for Newtonian dynamics and non-linear for non-Newtonian materials
such as glacial ice for which $\mathfrak{p} = 4/3$, where $\mathfrak{p}$ is usually stated in terms of the Glen's parameter $n = 1/(\mathfrak{p} - 1)$. We consider a
very viscous flow, so that ν_0 is large.

A key challenge in the time discretization of such problems arises from the fact that the computational domain Ω is coupled
to the time-dependent height of the ice surface $h(x, y, t)$ so that $\Omega(t) = \Omega(h(t))$. The dependence of the domain on the surface
elevation h means that the velocity field also implicitly depends on h. At the same time, these velocity components appear
as coefficients in the free surface evolution equation that governs the changes in h. The mutual dependence between h and $\mathbf{u}$
limits the numerical stability of explicit methods.

The Stokes equations are solved under the following boundary conditions: At the surface a stress-free boundary condition is
applied

$$((\mathbf{S}(\mathbf{u}) - \mathbf{I}p) \cdot \mathbf{n} = \mathbf{0}. \tag{5}$$

Here, $\mathbf{n} = (n_x, n_y, n_z)$ denotes the outward pointing normal and $\mathbf{I}$ the identity matrix. The lateral boundaries, Γ_W and Γ_E (see
Fig. 1), are assumed to be rigid in this study so that an impenetrability condition applies,

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad \mathbf{x} \in \Gamma_W \cup \Gamma_E. \tag{6}$$

Lastly, at the base or bedrock, Γ_b, we apply either a no-slip condition

$$\mathbf{u} = \mathbf{0}, \quad \mathbf{x} \in \Gamma_b, \tag{7}$$

or an impenetrability condition combined with a so-called Weertman sliding law

$$\mathbf{t}_i \cdot (\boldsymbol{\sigma}\mathbf{n}) = -C\mathbf{u} \cdot \mathbf{t}_i, \tag{8}$$

where $\boldsymbol{t}_i$ are tangent vectors that span the lower ice surface and C is the friction coefficient. Note that the methods in this article are not restricted to the specific boundary conditions on Γ_b, Γ_W and Γ_E.

We consider a finite element discretization in space, which is commonly used in both ice sheet modeling and geodynamics (Gagliardini et al., 2013; Larour et al., 2012; Kaus et al., 2010; Rose et al., 2017). The weak formulation of the Stokes problem is to find the velocity and pressure $(\mathbf{u}, p) \in [W_0^{1,\mathfrak{p}}(\Omega)]^d \times \{L^{\mathfrak{p}'}(\Omega)\}$ so that

$$(\mathbf{S}(\mathbf{u}), \mathbf{D}\mathbf{v})_\Omega - (\nabla \cdot \mathbf{v}, p)_\Omega - (\nabla \cdot \mathbf{u}, q)_\Omega = (\rho\mathbf{g}, \mathbf{v})_\Omega \quad \forall(\mathbf{v}, q) \in [W_0^{1,p}(\Omega)]^d \times \{L^{\mathfrak{p}'}(\Omega)\}, \tag{9}$$

where $(\cdot, \cdot)_\Omega$ denotes the L^2 scalar product, e.g. $(\mathbf{u}, \mathbf{v})_\Omega = \int_\Omega \mathbf{u} \cdot \mathbf{v} \, dx$. The appropriate spaces for the continuous problem are the Sobolev space $[W_0^{1,\mathfrak{p}}(\Omega)]^d$ and the Lebesgue space $\{L^{\mathfrak{p}'}(\Omega)\}$, where $\mathfrak{p}' = \mathfrak{p}/(\mathfrak{p}-1)$ (e.g., Hirn, 2013).

In this work, we discretize the weak form using Taylor-Hood (Taylor and Hood, 1974) elements, i.e. linear elements for the pressure and quadratic elements for the velocity. Such a discretization is inf-sup stable (Brezzi and Fortin, 1991).

The weak form of the free surface equation is to find the surface height $h \in W^{1,2}(\omega)$ such that

$$\left(\frac{\partial h}{\partial t}, q\right)_\omega = \left(\left(-u_x\frac{\partial h}{\partial x} + u_z\right), q\right)_\omega, \quad \forall q \in W^{1,2}(\omega). \tag{10}$$

In this paper, it is useful to note that the equation can be written equivalently as

$$\left(\frac{\partial h}{\partial t}, q\right)_\omega = \int_{\Gamma_s} \mathbf{u} \cdot \mathbf{n}q \, ds \tag{11}$$

The equivalence is easily seen by changing from integration on the surface to the two-dimensional footprint ω, and considering that the normal vector pointing out of a surface is $\mathbf{n} = \tilde{\mathbf{n}}/\|\mathbf{n}\|$, where $\tilde{\mathbf{n}} = (-\partial_x h, -\partial_y h, 1)$. This derivation takes the form

$$(\mathbf{u} \cdot \mathbf{n}, q)_{\Gamma_s} = \int_{\Gamma_s} \mathbf{u} \cdot \mathbf{n}q \, ds = \int_\omega \mathbf{u} \cdot \mathbf{n}q\sqrt{1 + (\partial_x h)^2 + (\partial_y h)^2} \, dx \tag{12}$$

$$= \int_\omega \mathbf{u} \cdot (-\partial_x h, -\partial_y h, 1)\frac{1}{\sqrt{1 + (\partial_x h)^2 + (\partial_y h)^2}}q\sqrt{1 + (\partial_x h)^2 + (\partial_y h)^2} \, dx \tag{13}$$

$$= \int_\omega \mathbf{u} \cdot (-\partial_x h, -\partial_y h, 1)q \, dx = \int_\omega \frac{\partial h}{\partial t}q \, dx. \tag{14}$$

The free surface equation is discretized using linear elements.

3 Low-order Time-Stepping Schemes and the Free Surface Stabilization Algorithm

The standard approach to solving the coupled system (1)-(2) is to solve the Stokes equations once per time step on a mesh that reflects the geometry from the preceding step, (Algorithm 1). An explicit Euler discretization of the free surface can thus be written as

$$
(h^{k+1}, q)_\omega = (h^k, q)_\omega + \Delta t \left(\left(-u_x^k \frac{\partial h^k}{\partial x} + u_z^k \right), q \right)_\omega = (h^k, q)_\omega + \Delta t \int_{\Gamma_s^k} \mathbf{u}^k \cdot \mathbf{n}^k q \, ds^k, \quad \forall q \in W^{1,1}(\omega),
\tag{15}
$$

where k is the current time step and $k+1$ is the next time step. In this explicit Euler discretization, the velocity, $\mathbf{u}$, is, by necessity, always evaluated on time step k. It is easy to change to an implicit treatment of the surface height, h, by evaluating it at step $k+1$ instead, but as was shown in Löfgren et al. (2022) and Löfgren et al. (2024), it is the treatment of the velocity $\mathbf{u}$ that determines the stability properties of the overall system. The explicit treatment of velocities is essentially analogous to solving the heat equation using an explicit Euler time-stepping scheme, and therefore, the time-step size is limited by stability constraints rather than by accuracy considerations.

Algorithm 1 Standard explicit ice sheet model algorithm

0: Create mesh

0: **for** each time-step k **do**

0:     Solve Stokes eq. for $(\mathbf{u}^k, p^k)$

0:     Solve free surface equation for h^{k+1}

0:     Update mesh to get Ω_h^{k+1}

0: **end for**=0

To avoid the restriction on the time-step, an implicit solver could in theory be used. A natural approach to such a solver is to iterate between the free surface solver and the Stokes solver until convergence. Unfortuntaely, the iterations does not converge due to the time step restriction and numerical instability. Here we adress this issue.

To relax the stability restriction without implicit iterations, the FSSA method was introduced in geodynamics in Kaus et al. (2010) and subsequently adapted to ice-sheet modeling in Löfgren et al. (2022); Löfgren et al. (2024). One way of understanding the FSSA stabilization method is by applying the Reynolds transport theorem to the gravitational force on weak form:

$$
\frac{\partial}{\partial t} (\rho \mathbf{g}, \mathbf{v})_\Omega = \frac{\partial}{\partial t} \int_{\Omega(t)} \rho \mathbf{g} \cdot \mathbf{v} \, d\mathbf{x} = \int_{\partial \Omega(t)} \rho (\mathbf{u}(t) \cdot \mathbf{n}(t))(\mathbf{g} \cdot \mathbf{v}) \, ds.
\tag{16}
$$

An explicit Euler discretization of the above relation gives an expression for how the driving stress on time-step $k+1$ relates to the driving stress on time-step k:

$$
\int_{\partial \Omega(t^{k+1})} \rho \mathbf{g} \cdot \mathbf{v} \, d\mathbf{x} \approx \int_{\Omega(t^k)} \rho \mathbf{g} \cdot \mathbf{v} \, d\mathbf{x} + \underbrace{\theta \Delta t \int_{\partial \Omega(t^k)} \rho (\mathbf{u} \cdot \mathbf{n})(\mathbf{g} \cdot \mathbf{v}) \, ds}_{\text{stabilization}},
\tag{17}
$$

where the last term is the FSSA stabilization term and $\theta \in [0,1]$ is a user-defined parameter. The stabilization is added to the Stokes equations (1) rather than the free-surface equation (2). Due to the impenetrability condition, the FSSA term vanishes on all surfaces except Γ_s. The FSSA-stabilized solutions mimic a fully implicit solution and thereby increase stability. Since the flow is gravity driven, it appears to be sufficient to represent the force of gravity implicitly (which FSSA does approximately) to yield an implicit solution. Note, however, that it is not a true fully implicit solver.

The FSSA method has proved remarkably successful in increasing the largest stable time step in marine ice sheet models, by up to a factor of approximately 100 (Henry et al., 2025).

An observation that will be useful for creating the fully implicit solver is that the FSSA term is similar to the right-hand side of the discretized free surface equation (15), the difference lies in the test function (which determines which equation the term is added to) and the scaling ρg which is only in the FSSA term.

4 A fully implicit solver using FSSA

A true fully implicit solver requires iterations between the free surface equation and Stokes equations in each time step, as in Algorithm 2.

Algorithm 2 Fully Implicit first order method

0: Create initial mesh Ω_0^0

0: **for** each time-step k **do**

0:    $h_0^{k+1} = h^k$

0:    $\Omega_0^{k+1} = \Omega^k$

0:    **while** $\|h_{r+1}^{k+1} - h_r^{k+1}\| >$tolerance and $r < r_{max}$ **do**

0:       Solve Stokes eq. for $(\mathbf{u}_r^{k+1}, p_r^{k+1})$

0:       Solve $h_{r+1}^{k+1} = h^k + \Delta t \left(-u_r^{x,k+1} \frac{\partial h_r^{k+1}}{\partial x} + u_r^{z,k+1} \right)$

0:       Update mesh to get Ω_r^{k+1}

0:       r++

0:    **end while**

0:    $h^k = h_r^{k+1}$

0:    k++

0: **end for**=0

Note that in step 6 of Algorithm 2 the time-discretized free surface equation is written on strong form for brevity of presentation, while in practice it is solved on weak form. Since for the first implicit iteration $r = 0$, the Stokes equations is solved on $\Omega_0^{k+1} = \Omega^k$, the first step is indead explicit, i.e. $(\mathbf{u}_0^{k+1}, p_0^{k+1}) = (\mathbf{u}^k, p^k)$.

In theory, a fully implicit solver should be unconditionally stable. However, in practice the implicit iterations (index r in Algorithm 2) do not converge for large time steps, making the fully implicit solver no more useful than an explicit one. This

is perhaps not surprising, as the first iteration in the implicit loop always consists of an explicit step in the velocity-pressure solution, which can induce instability for larger time steps and renders a poor first guess for the velocity coefficients in the free-surface equation (see the line labeled Explicit Euler in Fig. 4b).

The main goal of this paper is to stabilize the implicit iterations using two slightly different FSSA terms subtracted from each other. At the first non-linear iteration, $r = 0$, of each time step, an explicit FSSA-stabilized method is used just as for the purely explicit solver of Section 3. This renders a first guess that is close enough to the true implicit solution for the iterations to converge, this is the line labeled FSSA in Fig. 4b. For large time steps, the subsequent iterations must also be stabilized to avoid the growth of spurious oscillations, but applying the FSSA again would be erroneous since FSSA predicts the surface position at the next time step, not the next iteration. To create an appropriate stabilization term, we observe that the surface at implicit iteration $r + 1$, i. e. h_{r+1}^{k+1}, is computed as

$$(h_{r+1}^{k+1}, q)_\omega = (h^k, q)_\omega + \Delta t \left(\left(-u_{x,r}^{k+1} \frac{\partial h_r^{k+1}}{\partial x} + u_{z,r}^{k+1} \right), q \right)_\omega = (h^k, q)_\omega + \Delta t \int_{\Gamma_{s,r}^k} \mathbf{u}_r^{k+1} \cdot \mathbf{n}_r^{k+1} q \, ds_r^{k+1}, \tag{18}$$

while the surface at the previous implicit iteration, h_r^{k+1}, is computed as

$$(h_r^{k+1}, q)_\omega = (h^k, q)_\omega + \Delta t \left(\left(-u_{x,r-1}^{k+1} \frac{\partial h_{r-1}^{k+1}}{\partial x} + u_{z,r-1}^{k+1} \right), q \right)_\omega = (h^k, q)_\omega + \Delta t \int_{\Gamma_{s,r-1}^k} \mathbf{u}_{r-1}^{k+1} \cdot \mathbf{n}_{r-1}^{k+1} q \, ds_{r-1}^{k+1}. \tag{19}$$

Therefore, the distance between two subsequent surfaces in the implicit iterations is

$$(h_{r+1}^{k+1}, q) - (h_r^{k+1}, q) = \Delta t \int_{\Gamma_r^k} \mathbf{u}_r^{k+1} \cdot \mathbf{n}_r^{k+1} q \, ds_r^{k+1} - \Delta t \int_{\Gamma_{r-1}^k} \mathbf{u}_{r-1}^{k+1} \cdot \mathbf{n}_{r-1}^{k+1} q \, ds_{r-1}^{k+1}, \tag{20}$$

see Fig. 2 for a conceptual illustration. Recalling the similarity between the terms of the right-hand side of (15) we claim that the FSSA term must be augmented to be consistent with (20). The FSSA term of the previous iterations is thus subtracted from the FSSA term on the current iteration.

$$\theta_1 \Delta t \int_{\Gamma_r^k} \rho \mathbf{g} \cdot \mathbf{v} \, \mathbf{u}_r^{k+1} \cdot \mathbf{n}_r^{k+1} ds_r^{k+1} - \theta_2 \Delta t \int_{\Gamma_{r-1}^k} \rho \mathbf{g} \cdot \mathbf{v} \, \mathbf{u}_{r-1}^{k+1} \cdot \mathbf{n}_{r-1}^{k+1} ds_{r-1}^{k+1}. \tag{21}$$

We will call this method the *subtraction-FSSA*, and we will use (21) to stabilize the fully implicit discretization for $r > 0$. By the subtraction-FSSA, the term $\Delta t \mathbf{u} \cdot \mathbf{n}$ in (17), which is the distance traveled by the surface from one time step to another, is replaced by (20) to ensure that the correct surface displacement is used in the Reynolds transport theorem. Here, we have introduced the parameters θ_1 and θ_2 for later convenience. Setting $\theta_2 = 0$ results in the original FSSA term and, for the new subtraction-FSSA, $\theta_1 = \theta_2$. Exactly like for the original FSSA, the subtraction-FSSA of (21) is added to the Stokes equations. A key observation is that

$$(h_{r+1}^{k+1}, q) - (h_r^{k+1}, q) \to 0 \quad \text{as } r \to \infty, \tag{22}$$

i.e. as the implicit discretizations converge, the distance between the surfaces at consecutive iterations tend to zero and thereby the stabilization also approaches zero. The iterations should therefore converge to the unstabilized implicit solution.

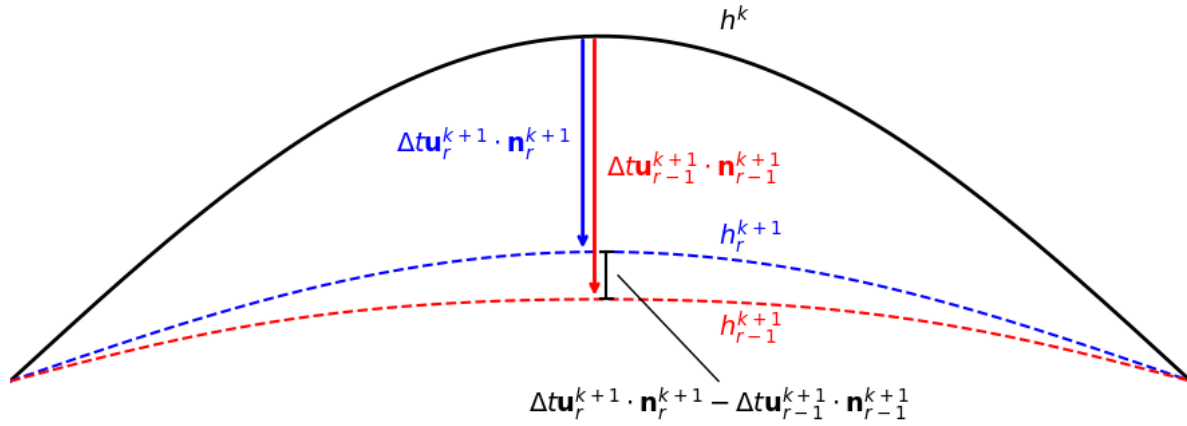



Figure 2. Illustration of the distance estimate that is the basis of the stabilization method used for implicit iterations. The arrows marks the distance traveled from h^k to h^{k+1} in to different implicit iterations. The vertical bar measures the difference between the two arrows, which is the distance traveled by the surface between two implicit iterations.

In some software programs, like Elmer/Ice, it is difficult to access the mesh and normal vector from iteration k, required in (21). For this purpose we introduce a simplified subtraction-FSSA, which only requires the velocity to be accessible from the previous iteration

$$\theta_1 \Delta t \int_{\Gamma_r^k} \rho \mathbf{g} \cdot \mathbf{v} \, \mathbf{u}_r^{k+1} \cdot \mathbf{n}_r^{k+1} ds_r^{k+1} - \theta_2 \Delta t \int_{\Gamma_r^k} \rho \mathbf{g} \cdot \mathbf{v} \, \mathbf{u}_{r-1}^{k+1} \cdot \mathbf{n}_r^{k+1} ds_r^{k+1}. \tag{23}$$

5 A second order implicit time-stepping scheme

For first order Euler discretizations of the free surface equation, the numerical error increases linearly with the time step. The increased stability of FSSA and fully implicit solutions allows for large time steps, which can introduce large errors. To reduce errors, we now construct higher order time-stepping schemes. It is of course trivial to exchange the discretization of the free surface equation with a higher-order scheme. However, the FSSA stabilization term was derived by a first-order discretization of the Reynold's transport theorem, and higher-order effects of the h-dependency of terms other than the body force term were ignored. The FSSA-stabilization will thus reduce the overall accuracy to first order. However, with the new implicit scheme introduced in Section 4 it is possible to achieve higher-order accuracy since the stabilization term of (21) tends to zero. Thus, a standard implicit higher-order scheme can be applied to the free surface. Here, we choose a second-order Crank-Nicolson scheme as well as a second-order Backward Differentiation (BDF2) scheme. We present these schemes on strong form and

drop the index r for brevity of presentation. The Crank-Nicolson scheme is

$$h^{k+1} = h^k + \Delta t \left(\frac{1}{2} \left(-u_x^k \frac{\partial h^k}{\partial x} + u_z^k \right) + \frac{1}{2} \left(-u_x^{k+1} \frac{\partial h^{k+1}}{\partial x} + u_z^{k+1} \right) \right) \tag{24}$$

175   where $\mathbf{u}^{k+1}$ is the solution retrieved from the implicit iterations and $\mathbf{u}^k$ is the solution of the previous time step. The BDF2-scheme is

$$\frac{3h^{k+1} - 4h^k + h^{k-1}}{2\Delta t} = -u_x^{k+1} \frac{\partial h^{k+1}}{\partial x} + u_z^{k+1}. \tag{25}$$

6   Idealized Numerical Experiments

6.1   Overview

180   To evaluate the error, the convergence order and the efficiency of the subtraction-FSSA in its full and simplified form, we run the same idealized experiments as in (Löfgren et al., 2022). The setup is relevant to mantle convection, and the experiments are inspired by the mantle convection experiments of, e.g. (Rose et al., 2017; Andrés-Martínez et al., 2015). The software used is Biceps (Löfgren, 2025), which is a C++ finite element library developed specifically to solve two-dimensional Stokes-coupled free-surface problems. It relies on the Eigen3 library (Guennebaud et al., 2010) for storing and solving linear systems. Biceps

185   includes its own suite of finite element utilities, providing users with fine-grained control over the assembly process.

6.2   Experimental Setup

The simulation domain consists of a two-dimensional horizontal slab of ice with an undulating upper surface, defined over the interval $x \in \omega = [0, L]\,\mathrm{m}$, where the length of the domain L is 100 km. The mean ice thickness is set to $H = 1.0\,\mathrm{km}$, and the surface elevation varies sinusoidally with amplitude $z_0 = 0.1\,\mathrm{km}$.

190   $$z_s(x) = H + z_0 \cos\left(\frac{\pi x}{L}\right).$$

The ice is treated as a Newtonian fluid ($\mathfrak{p} = 2$, $\varepsilon = 0$) with $\nu_0 = 10^{12}\,\mathrm{Pa \cdot s}$. Gravity is acting in the vertical direction with magnitude $g = 9.8\,\mathrm{m/s^2}$, and the ice density is set to $\rho = 910\,\mathrm{kg/m^3}$. The surface is a material surface, i.e. $a_s = 0$. The computational mesh is structured and consists of 50 elements in the horizontal direction and 5 elements in the vertical. In Löfgren et al. (2022); Löfgren et al. (2024) the mesh dependence of the stability properties was studied in detail. Triangular elements

195   are used with a polynomial degree of 2 for velocity, 1 for pressure, and 1 for surface height. The time integration spans a final simulation time of $T = 20.0\,\mathrm{years}$, which allows the surface to evolve significantly, yet remains far enough away from the steady-state configuration.

We perform experiments with first-order Euler schemes, Crank-Nicolson and BDF2, with and without stabilization, and with the maximum number of implicit iterations r_{max} being 1 (no implicitness), 2 (minimum implicitness) and 100 (allowing full

200   implicitness). The error tolerance for the implicit iterations is set to $1e-9$ unless otherwise stated. The stabilization parameter is

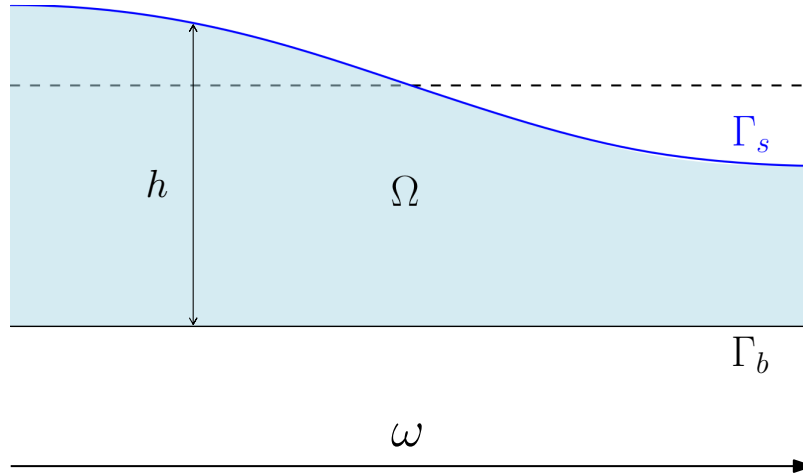



Figure 3. Experimental setup. Outlined is the bulk domain Ω, the free-surface domain ω, the surface Γ_s, the height of the surface h, and the bedrock Γ_b. The surface will relax to the equilibrium configuration (black dashed line) as time progresses.

set to $\theta = 1$ when applicable. Experiments are performed with time step sizes of $\Delta t = 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2.5, 5, 10,$ and 20 years. For each experiment, we measure the error and count the number of iterations, i.e. Stokes solves.

To measure the errors, we construct a reference solution by running a second-order Adams-Bashforth time-stepping scheme

$$h^{k+1} = h^k + \frac{\Delta t}{2}\left(3\left(-u_x^k\frac{\partial h^k}{\partial x} + u_z^k\right) - \left(-u_x^{k-1}\frac{\partial h^{k-1}}{\partial x} + u_z^{k-1}\right)\right) \tag{26}$$

for a very small time step of $\Delta t = 0.00001$ without FSSA-stabilization. Note that Adams-Bashforth is an explicit method, which is unstable for large time steps but stable for $\Delta t = 0.00001$. The accuracy of the surface elevation is monitored using a normalized L_2-like metric:

$$\|\Delta h\| = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{\left(h_r^{k+1,i} - h_{r+1}^{k+1,i}\right)^2}{\left(h_r^{k+1,i} - H\right)^2}},$$

where h_r^{k+1} and h_{r+1}^{k+1} denote the surface height at two consecutive implicit iterations and i is the surface mesh node.

6.3 Results

6.3.1 First order schemes

The results are shown in Fig. 4 and Fig. 5, where errors in surface position is plotted against time-step size. The largest time step for which even the unstabilized explicit Euler scheme is stable is $\Delta t = 0.01$ years, see Fig. 5a. The unstabilized implicit solver (i.e. the backward Euler or BDF1 solver) is also only stable for time steps up to $\Delta t = 0.01$ years; see Fig. 5c. For

$\Delta t = 0.01$, one can see that the surface of the FSSA-stabilized "explicit" Euler discretization lies between the unstabilized and the fully implicit BDF1 method Fig. 4a. This is expected since FSSA only modifies the integration domain of the right-hand side of the Stokes equations, while the left-hand-side integration domains are still treated explicitly, so the discretization is a hybrid between explicit and implicit approaches. However, the FSSA-stabilized solution is close to the implicit solution,

220   perhaps indicating that since the flow is gravity driven, the implicit treatment of the left-hand side is not as important as that of the right-hand side.

In Fig. 4b, all three solutions are shown for a large time step of $\Delta t = 20$. The unstabilized explicit Euler scheme is unstable, and the instability is a low-frequency sloshing or "drunken sailor" type of instability similar to those of e.g. (Kaus et al., 2010). The FSSA-stabilized solution and the implicit solution stabilized by equation (21) are stable and close to each other, differing

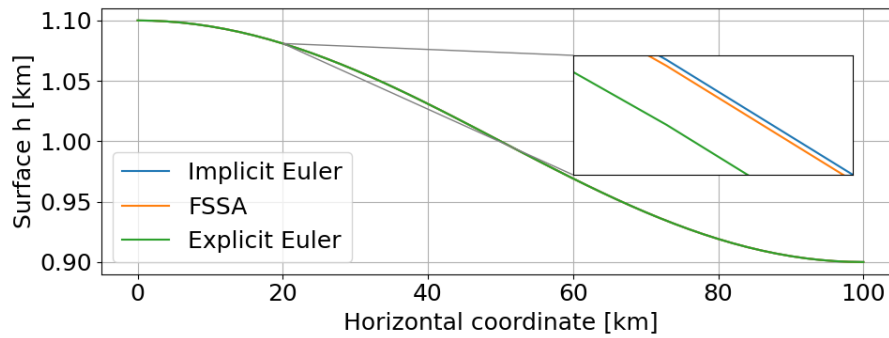
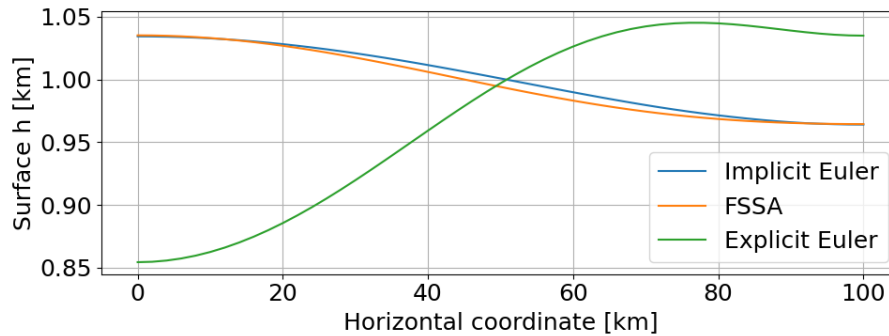
225   mostly at the inflection point. Interestingly, the overall error as shown in Fig. 5 is not lower for the fully implicit solver, indicating that the FSSA error is lower than the overall discretization error for first-order schemes. Note that the behavior of

(a) $\Delta t = 0.01$ years. The inserted box is a zoom in around $x = 20$, $y = 1.0808$.

(b) $\Delta t = 20$ years

Figure 4. Explicit Euler with (orange) and without FSSA (green), as well as the implicit solution (blue). The solutions are shown after one single small time-step for which all methods are stable in the top panel, and one single large stable time step for which the unstabilized solution is unstable.

the implicit solver depends on the error tolerance. If the error tolerance is lower than the overall discretization error, it will use up unnecessarily many iterations, and high-frequency errors will eventually start to grow. These high frequency oscillations can be mitigated by, for example, an artificial viscosity approach on the free surface equation, but we instead choose to terminate the iterative solver if the errors grow, as this is more efficient. Note also that even though the FSSA solution from the first iteration is close to the converged implicit solution, it is not possible to only use FSSA in the first time-step, i.e. θ_1 must be greater than 0 for all iterations.

6.3.2 Second order schemes

The results are shown in Fig. 6 and Fig. 6d. As for the first-order Euler method, the simulations need the stabilization to be stable. Simply adding the FSSA stabilization renders first-order convergence, while our idea of subtracting two FSSA terms from different iterations according to (21) renders second-order convergence for both Crank-Nicolson and BDF2. An interesting result of this is that only two iterations (Stokes solves) are needed in each time step to achieve second-order convergence. This is very advantageous for efficiency. For example, if a user requires an accuracy of 10^{-4}, a time step $\Delta t = 10^{-3}$ is needed for the first-order method, that is, $20/10^{-3} = 20000$ time steps are needed. The most efficient first-order method is to choose one implicit iteration, i.e., without implicitness, only FSSA stabilization. Then one Stokes solve is needed per time-step, which results in 20 000 Stokes solves. For Crank-Nicolson and BDF2, this accuracy is reached for $\Delta t = 0.1$, i.e., $20/0.1 = 200$ time steps are needed. With two Stokes solves per time step, this means that 400 Stokes solves are needed in total. The speedup should thus be about $20000/400 = 50$, which is remarkable.

6.3.3 The simplified subtraction-FSSA

The simplified scheme of equation (23) performs as well as the exact method of equation (21) for both first order and second order schemes, see Fig. 8. Just as for the full subtraction-FSSA, only two coupled iterations are needed to obtain second order convergence.

7 Application to glacier simulations

7.1 Overview

An important application of the methods in this paper is in the simulation of glaciers and ice sheets. These type of simulations exhibit two key characteristics, which the idealized mantle convection setup of the previous section does not:

- Non-linear rheology, with $\mathfrak{p} = 4/3$ in equation (4).

- A constraint on the free surface, h, ensuring that the surface stays above the bedrock.

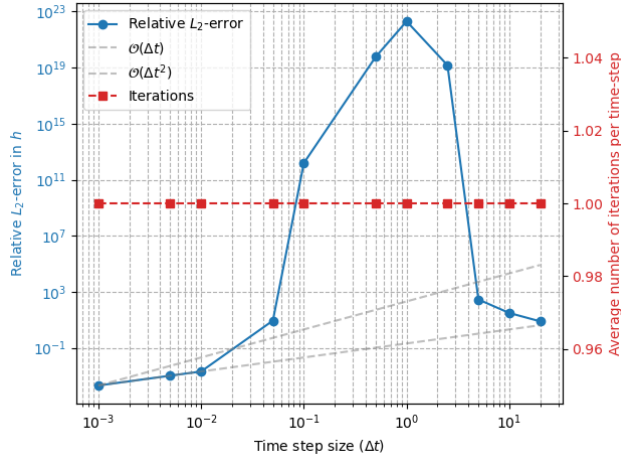
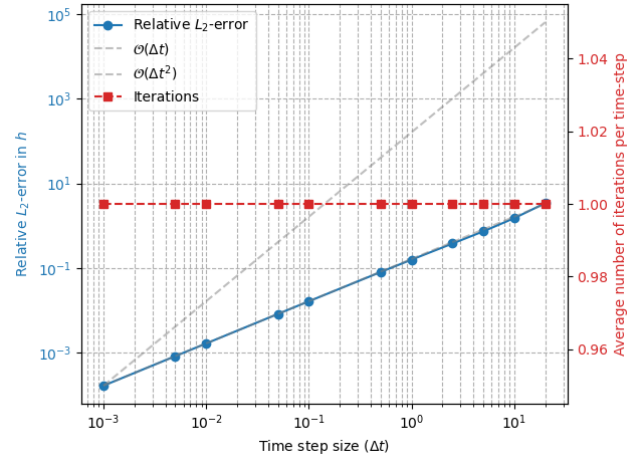
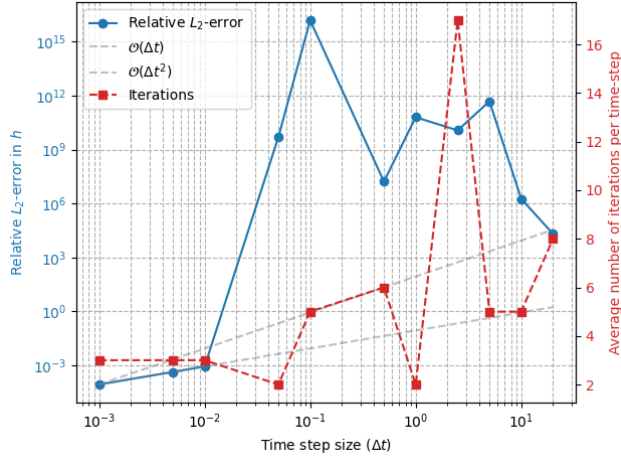
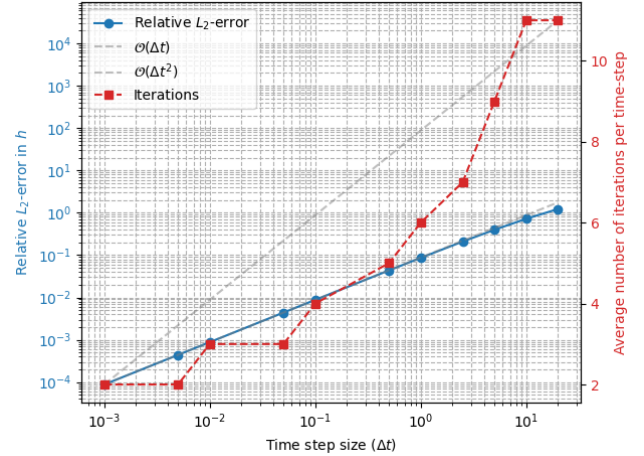
The non-linear rheology means that each Stokes solve is more expensive due to internal Picard fixed-point iterations that resolve the nonlinear viscosity. Reducing the number of time-steps would therefore be even more beneficial. The constraint on

(a) Euler forward for free surface, no FSSA. The very large errors for intermediate time-step sizes are due to high frequency oscillations. For the three largest time steps, the solution is smooth but erroneous.

(b) Euler forward for free surface, with FSSA ($\theta_1 = 1, \theta_2 = 1$).

(c) Fully implicit solver, no FSSA. A maximum of 100 implicit iterations is permitted per time step. Error tolerance $1e - 9$.

(d) Fully implicit solver. A maximum of 100 implicit iterations is permitted per time step. Error tolerance $1e - 9$.

Figure 5. Error in surface position h (in km) as computed with first order schemes for different time-steps Δt (in years), with and without FSSA, and with and without implicit iterations. The reference solution is with $\Delta t = 0.00001$ years. Final time = 20 years.

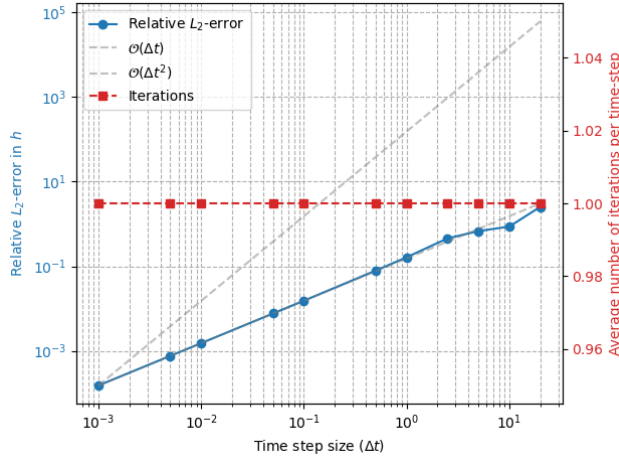
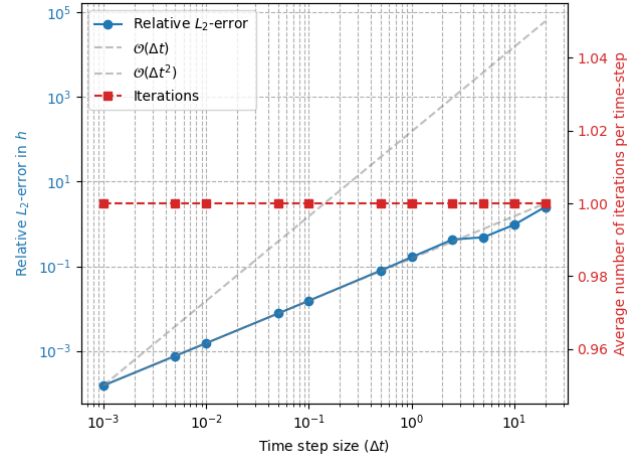
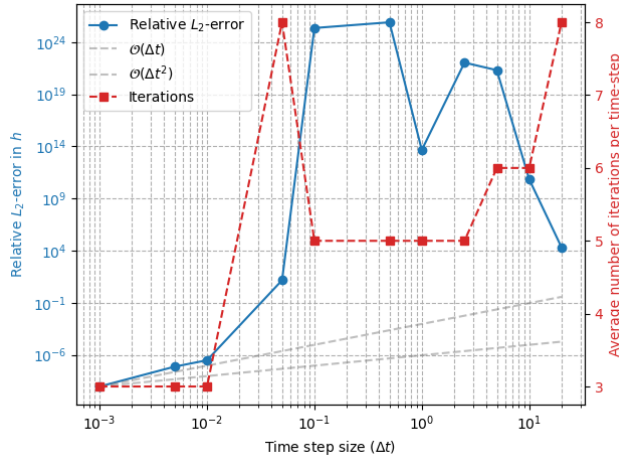
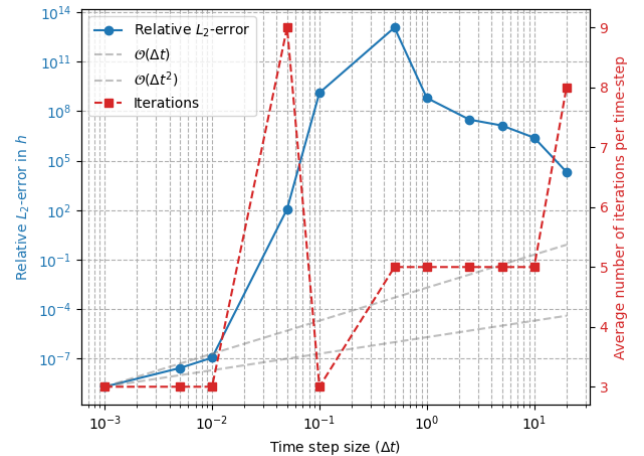
the free surface is implemented in such a way that it can impact the number of implicit iterations and possibly the accuracy of the methods, which is investigated in the experiments.

(a) Crank Nicolson solver without implicit iterations, i.e. FSSA is used to approximate $\mathbf{u}^{n+1}$.

(b) BDF2 solver without implicit iterations, i.e. FSSA is used to approximate $\mathbf{u}^{n+1}$.

(c) Crank Nicolson solver with implicit iterations, without FSSA.

(d) BDF2 solver with implicit iterations, without FSSA.

Figure 6. Error in surface position h (in km) as computed with non-working "second order" schemes for different time-steps Δt (in years). The simulations without FSSA or implicit iterations does yield second order accuracy. Reference solution is with $\Delta t = 0.00001$ years. Final time = 20 years.

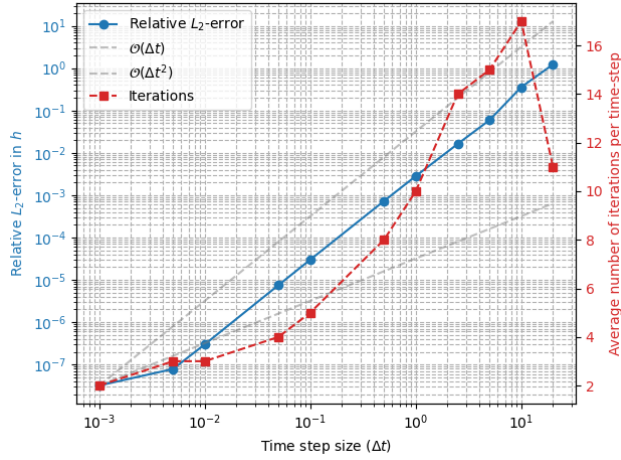
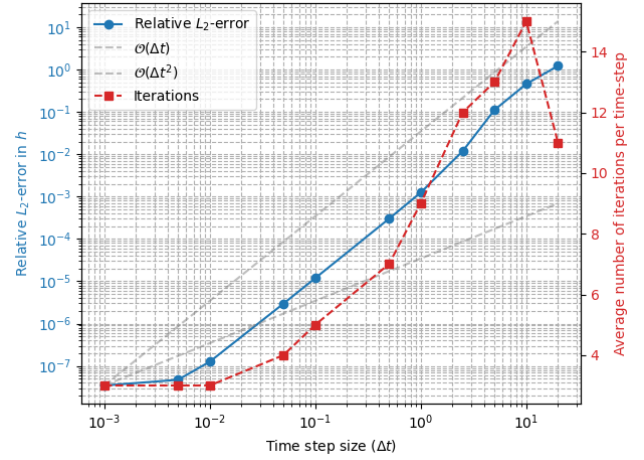
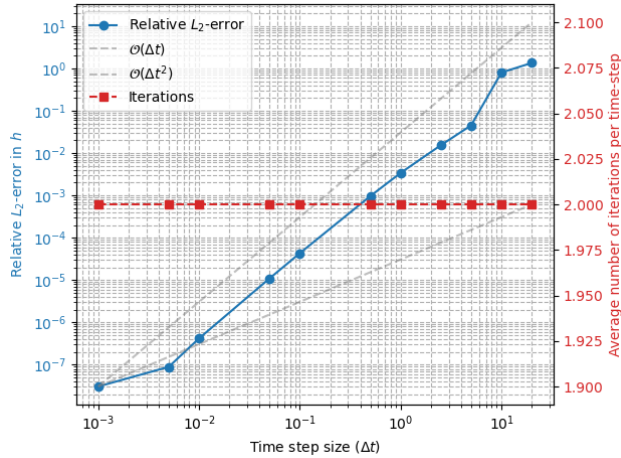
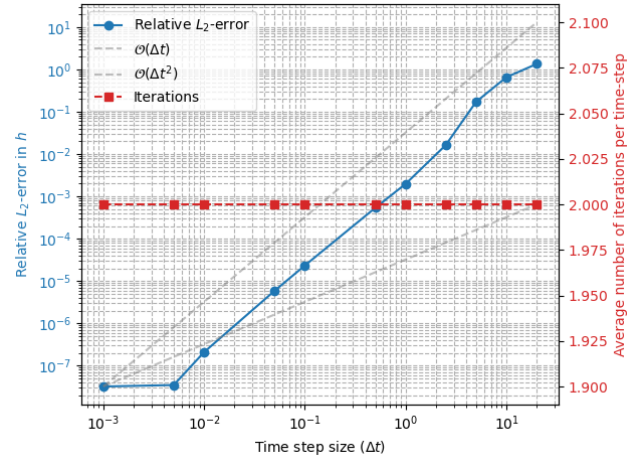
We evaluate the accuracy and efficiency of glacier simulations using the popular ice sheet simulation code Elmer/Ice (Gagliardini et al., 2013). Since accessing the mesh at a previous iteration in Elmer/Ice is nontrivial, we implement the sim-
260 plified version of the stabilization method as in (20). It should be noted that without stabilization, the free surface of glaciers and ice sheets is prone to instabilities, and therefore relaxation of this solution is routinely used in Elmer/ice, which might be unnecessary in simulations with FSSA.

(a) Fully implicit Crank-Nicolson solver. A maximum of 100 implicit iterations is permitted per time step. Error tolerance $1e-9$.

(b) Fully implicit BDF2-solver. A maximum of 100 implicit iterations is permitted per time step. Error tolerance $1e-9$.

(c) Crank Nicolson solver with only two implicit iterations.

(d) BDF2 solver with only two implicit iterations.

Figure 7. Error in surface position h (in km) as computed with working second order schemes for different time-steps Δt (in years). The reference solution is computed with $\Delta t = 0.00001$ years. Final time = 20 years.

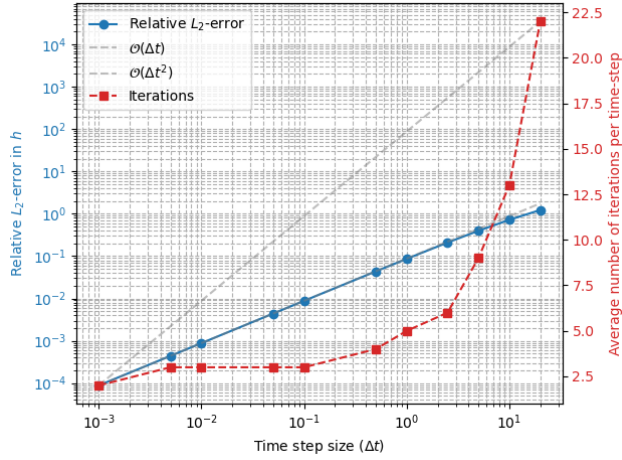
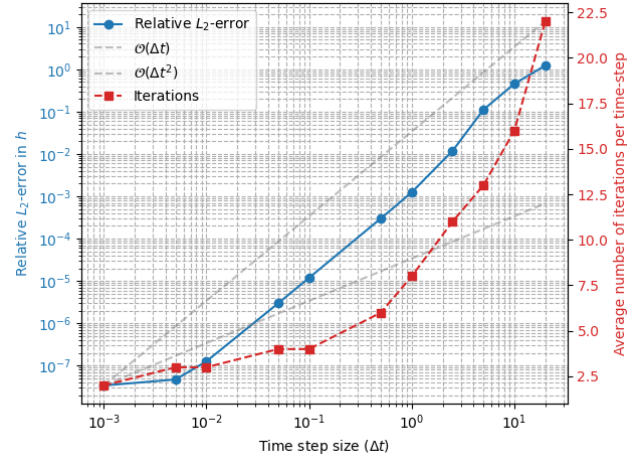
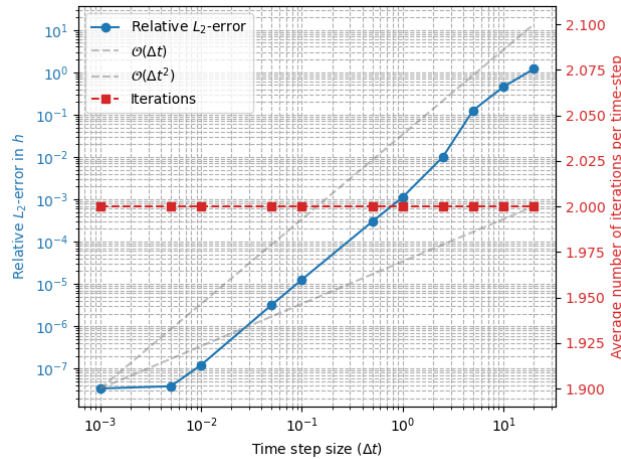
We choose to model a so-called Perlin glacier similar to that of (Löfgren et al., 2024), a setup that allows for relatively fast simulations and thus thorough experimentation with parameter choices. Note that this type of mountain glacier is more stable than an ice sheet (compare the results of (Löfgren et al., 2024) to (Löfgren et al., 2022)), but that it has the important feature of a moving glacier front, which activates the constraint on the surface h. In the following section, we outline the strategy used in Elmer/Ice to handle the constraint as described in (Durand et al., 2009; Gagliardini et al., 2013).

(a) Fully implicit BDF1 solver using the simplified stabilization of (23). A maximum of 100 implicit iterations is permitted per time step. Error tolerance $1e-9$.

(b) Fully implicit BDF2-solver using the simplified stabilization of (23). A maximum of 100 implicit iterations is permitted per time step. Error tolerance $1e-9$.

(c) Fully implicit BDF2-solver using the simplified stabilization of (23). A maximum of 100 implicit iterations is permitted per time step. Error tolerance $1e-9$.

Figure 8. The simplified stabilization method of (23) in combination with BDF1 and BDF2. As with the exact stabiliation of (21), 2 coupled iterations are enough. Reference solution is with $\Delta t = 0.00001$. Final time = 20 years.

7.2 The free surface constraint

To prevent the ice surface from falling below the bedrock elevation $\boldsymbol{b}$, a constraint is imposed on the free surface:

270 $$h(x,t) - H_{\min} \geq b(x,t), \tag{27}$$

where $H_{\min} > 0$ avoids remeshing near the glacier front. Discretizing the free surface equation (10) results in the linear system

$$\boldsymbol{Ah} = \boldsymbol{f}, \tag{28}$$

with $\boldsymbol{A}$ as the finite element matrix, $\boldsymbol{h}$ the surface elevation vector, and $\boldsymbol{f}$ the forcing vector that includes the vertical velocity
275 and surface mass balance.

The nodes on the surface boundary Γ_s are labeled as *inactive* if they satisfy the constraint and *active* if they do not. The system is solved iteratively: When a previously inactive node violates the constraint, it becomes active. For active nodes, a Dirichlet condition is enforced by modifying the system such that $A'_{ij} = \delta_{ij}$ and $f_i = H_{\min} + b$, setting the surface to the minimum allowable thickness. The residual of the system may now become larger.

280 To determine whether a previously active node would become inactive, the residual r_j is checked: If $r_j < 0$, the node is reclassified as inactive; if $r_j > 0$, it remains active. This iterative process continues until the solution $\boldsymbol{h}$ converges.

The standard approach in glaciology is to not use implicit iterations. This does not allow the velocity to update in the system (28) within one time step, which for large time steps could affect behavior close to the areas where the constraint is active, i.e., close to the glacier front.

285 ## 7.3 Experimental Setup

The Perlin glacier is a synthetic two-dimensional glacier flowing down a mountainside (Figure Fig. 9). The bedrock elevation is constructed using so-called Perlin noise and takes the form

$$b(x) = \frac{\alpha}{L_x}(x - L_x)^2 + C^1 \text{octave}_1(\Delta x^1, x) + C^2 \text{octave}_2(\Delta x^2, x) + C^3 \text{octave}_3(\Delta x^3, x), \tag{29}$$

where $L_x = 8\,000$ m is the horizontal extent of the domain and $\alpha = 0.1$ controls the slope of the bedrock. The octaves represent
290 noise of various frequencies and allow the construction of pseudo-random topography, which resembles natural bedrock (Perlin, 1985; Löfgren et al., 2024). The topography chosen corresponds to the $\alpha = 0.1$ simulation in Löfgren et al. (2024).

At the initial time $t = 0$, the glacier domain contains an artificial ice layer of thickness 10 m, corresponding to the minimum allowed ice thickness $H_{\min}$. The glacier is then built up by surface accumulation, prescribed at the rate

$$a_s = \max\left(1 - \frac{3x}{L_x}, , 0\right). \tag{30}$$

295 During the simulation, the glaciated area is defined as the region where the ice thickness exceeds $H_{\min}$. Areas with thickness equal to or less than $H_{\min}$ are considered ice-free, consistent with standard Elmer/Ice procedures. Simulations are run until 400 years.

The viscosity is defined by $\mathfrak{p} = 1/n + 1$, $\nu_0 = 1/2^{(1/n+1)/2} A^{1/n}$, $\varepsilon = 2 \cdot 10^{-10}$, where the Glen exponent $n = 3$ and the ice fluidity factor $A = 100$ years^{-1} MPa^{-3}, that is, the ice is very viscous and shear-thinning. The density of ice is 910 kg m^{-3} and

300   the acceleration of gravity is -9.8 m,s^{-2}. At the bottom Γ_b we use a linear Weertman sliding law (8) with friction coefficient $C = 0.1$ MPa m^{-1} a, which differs from the friction law used in Löfgren et al. (2024), but results in a similar ice geometry.

Triangular elements are used with a polynomial degree of 1 for velocity, 1 for pressure, and 1 for surface height. To circumvent the inf-sup condition, we employ the stabilization method of (Franca and Frey, 1992), a common choice in Elmer/Ice. The mesh is structured with 10 vertical layers and 1000 horizontal elements. During the simulation, the mesh deforms vertically to

305   fit the evolving free surface, and the minimum thickness $H_{\min}$ prevents the elements from degenerating to zero thickness.

The nonlinear viscosity is resolved using Picard fixed-point iterations with a convergence tolerance of 10^{-8} and a relaxation factor of $2/3$. The linearized system is solved at each nonlinear iteration with the MUMPS direct solver (Amestoy et al., 2001). The free surface equation is stabilized with streamline upwind Petrov-Galerkin stabilization (SUPG) (Gagliardini et al., 2013) and the resulting linear system is solved with the BiCGStab method, using a convergence tolerance of 10^{-9}. The error tolerance

310   for the implicit iterations is set to $1e - 9$ as in the idealized experiments, but no error growth check is needed to terminate the iterations, probably due to the use of SUPG.

The experiments aim to study how the geometry of the glacier front and CPU time depend on

- the time step sizes $\Delta t = 0.1, 5, 10$,

- the number of maximum allowed implicit iterations $r_{max} = 1, 3, 100$,

315   - FSSA stabilization ($\theta_1 = 0, 1, \theta_2 = 0, 1$),

- the order of discretization of the free surface (1 or 2), and the relaxation of the free surface (0,0.5).

The simulations with a time step of 0.1 are used to produce reference solutions. The time steps of 5 and 10 years are very large. The second-order free surface discretization of choice is BDF2, since this approach is common in Elmer/Ice and showed results, which performed as well as Crank-Nicolson in the idealized experiments.

320   **7.4   Results**

In Fig. 9 the glacier geometry is shown for all the simulations performed, while the efficiency is illustrated in Fig. 10. The stability limit for the unstabilized method lies between 1 and 5 years, which is why the case $\theta_1 = \theta_2 = 0$ is not shown for $\Delta t = 5, 10$. The largest stable time-step of similar simulations was studied at length in (Löfgren et al., 2024).

Panel (a) and the zoomed in version of panel (b) of Fig. 9 show the glacier front for first-order Euler schemes with implicit

325   iterations (i.e., BDF1). The maximum allowed implicit iterations is $r_{max} = 100$ and no surface relaxation is used. It can be seen that the glacier front is in approximately the same position for the stabilized simulations ($\theta_1 = 1$) with large time steps $\Delta t = 5, 10$ as it is for the reference solutions where $\Delta t = 0.1$, although the shape is smoother. The front is more accurate for the new subtracted-FSSA method ($\theta_2 = 1$) than it is for the original FSSA method ($\theta_2 = 0$), and as expected, $\Delta t = 5$ gives a better result than $\Delta t = 10$. The CPU time is dominated by the number of linear Stokes solves, which is the number of Picard

iterations per implicit iteration × number of implicit iterations × number of time steps. As expected, the number of nonlinear
Picard iterations decreases somewhat for small time steps and with an increased number of implicit iterations (panel (a) of
Fig. 10). This expected since the initial guess to the Picard solver is better in these cases (the initial is the solution from the
previous time step and implicit iteration). The decreased number of time steps still renders the large time-step simulations
much more efficient, as can be seen in panel (e).

Panel (c) of Fig. 9 shows the effect of using a higher-order BDF2 scheme instead of the first-order Euler (BDF1) schemes.
BDF2 is more accurate than BDF1, and notably it is more accurate to run a time step of $\Delta t = 10$ using BDF2 than a time-step
of $\Delta t = 5$ using BDF1. As can be seen in panel (b) and panel (f) of Fig. 10, there is a very small computational cost associated
with switching to BDF2. In panel (d) of Fig. 9, the effect of switching on the free surface relaxation is illustrated. The accuracy
is severely affected by doing so, while the efficiency is not impacted (see panel (c) and (g) of Fig. 10).

Finally, in panel (e) of Fig. 9, the impact of restricting the number of implicit iterations r_{max} is shown. For $r_{max} = 1$, i.e.
without implicit iterations, the front is delayed for the large time steps. This is likely due to the fact that the iterations used to
impose the constraint described in Section 7.2 benefit from the velocity updates that the implicit iterations offer. In the idealized
experiments, we showed that $r_{max} = 2$ is enough to yield accurate results. However, for the Perlin glacier, $r_{max} = 2$ still yields
a delayed front, while $r_{max} = 3$ yields as accurate results as $r_{max} = 100$. The extra iteration needed in the glacier experiments
compared to idealized experiments is probably due to the constraint of the free surface. The computational cost is the highest
for $r_{max} = 100$ (panel (h)) but is not 100 times higher, as expected, since the number of coupled iterations is terminated by
the error tolerance check rather than the maximum number of iterations, and since the number of Picard iterations per coupled
iteration is lower for $r_{max} = 100$ (panel (d)). Setting $r_{max} = 3$ slightly reduces the computational cost, whilst retaining good
accuracy.

8 Conclusion

We have developed the first fully implicit method that allows for very large time-steps as well as second-order accuracy
in simulations of viscous free-surface flow simulations. The method has important applications in e.g. ice sheet and glacier
modelling and geodynamics. The implicitness allows for large time-steps and thereby simulation speedup, while second order
schemes keeps accuracy high. The key is a new stabilization scheme, the **subtraction-FSSA**, which extends the original FSSA
approach of (Kaus et al., 2010; Löfgren et al., 2022; Löfgren et al., 2024; Henry et al., 2025). Compared to the original FSSA
stabilization, our new scheme increases accuracy, not only because it enables second-order time-stepping schemes but also
because it allows for a more accurate representation of the free-surface constraint preventing negative thickness at ice sheet
and glacier margins. We further find that a simplified, easy-to-implement variant of the implicit iterations performs equally
well as the full method and requires only access to the velocity of the previous iteration.

For first-order schemes, our implicit solver matches the accuracy of explicit FSSA in unconstrained geodynamic problems,
confirming that FSSA already approximates an implicit solution well in such cases. In contrast, for moving glacier fronts,

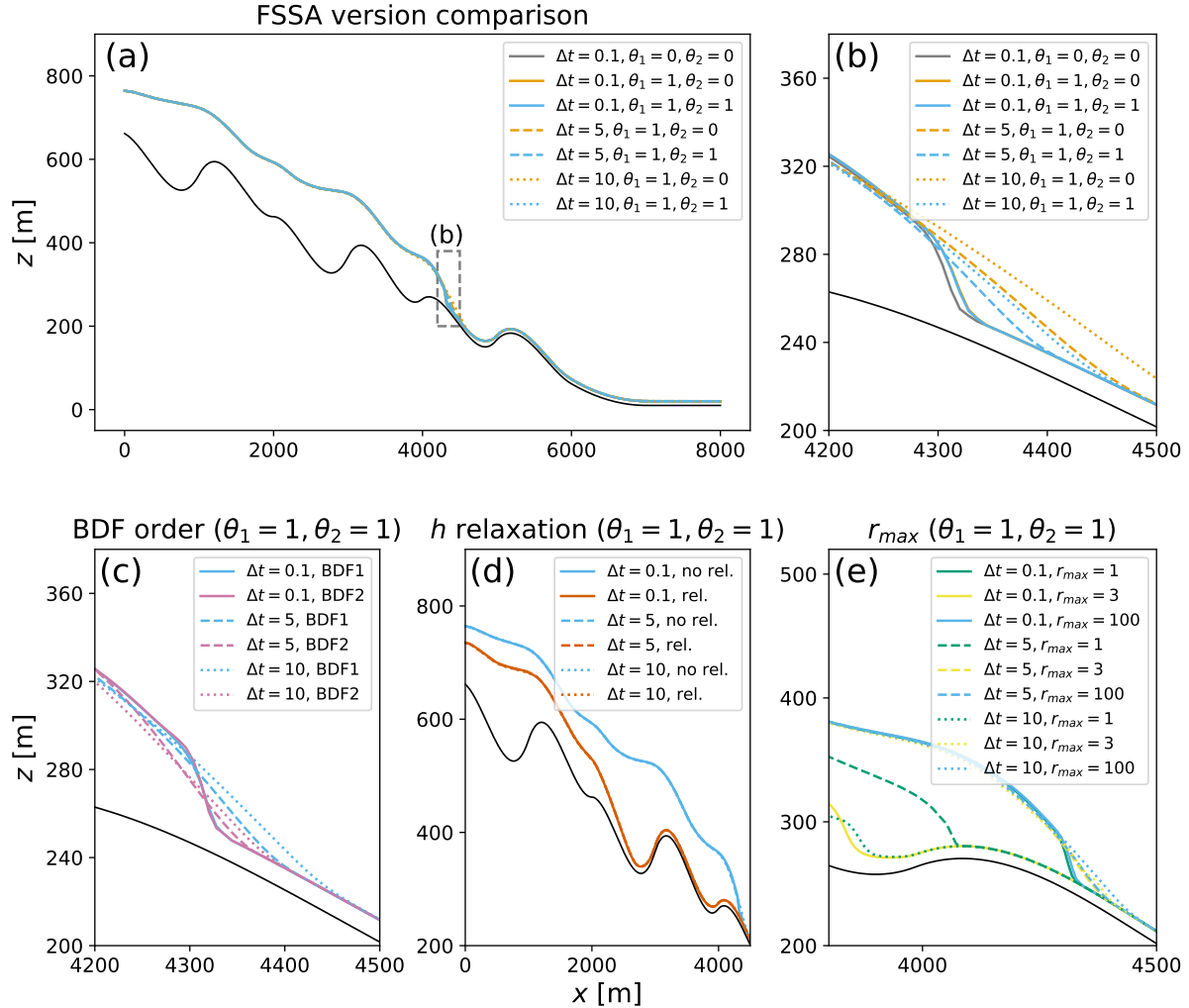



Figure 9. The surface elevation, h, of the Perlin glacier after a 400-year simulation, with varying parameters. Panels (a) and (b) show the surface elevation computed with BDF1 across the time step sizes 0.1, 5 and 10 years with subtraction-FSSA and the original FSSA formulation, along with the reference simulation with no FSSA stabilisation. Panel (c) shows a comparison between BDF order 1 and 2 across simulations with time step sizes of 0.1, 5 and 10 years, all of which include the subtraction-FSSA. Panel (d) shows simulations with time step sizes of 0.1, 5 and 10 years with and without relaxation of the free-surface solver. Lastly, panel (e) shows the surface elevation with and without coupled iterations for time step sizes of 0.1, 5 and 10 years. Across all figures, the black line indicates the bed elevation.

implicit iterations are substantially more accurate. It is overall more computationally efficient to use large time steps with three implicit iterations compared to using smaller time-steps with no implicit iterations.

Second-order schemes improve accuracy for both unconstrained and constrained (glacier) problems, requiring only two implicit iterations for the former and three for the latter, keeping computational costs low. The approach of using implicit

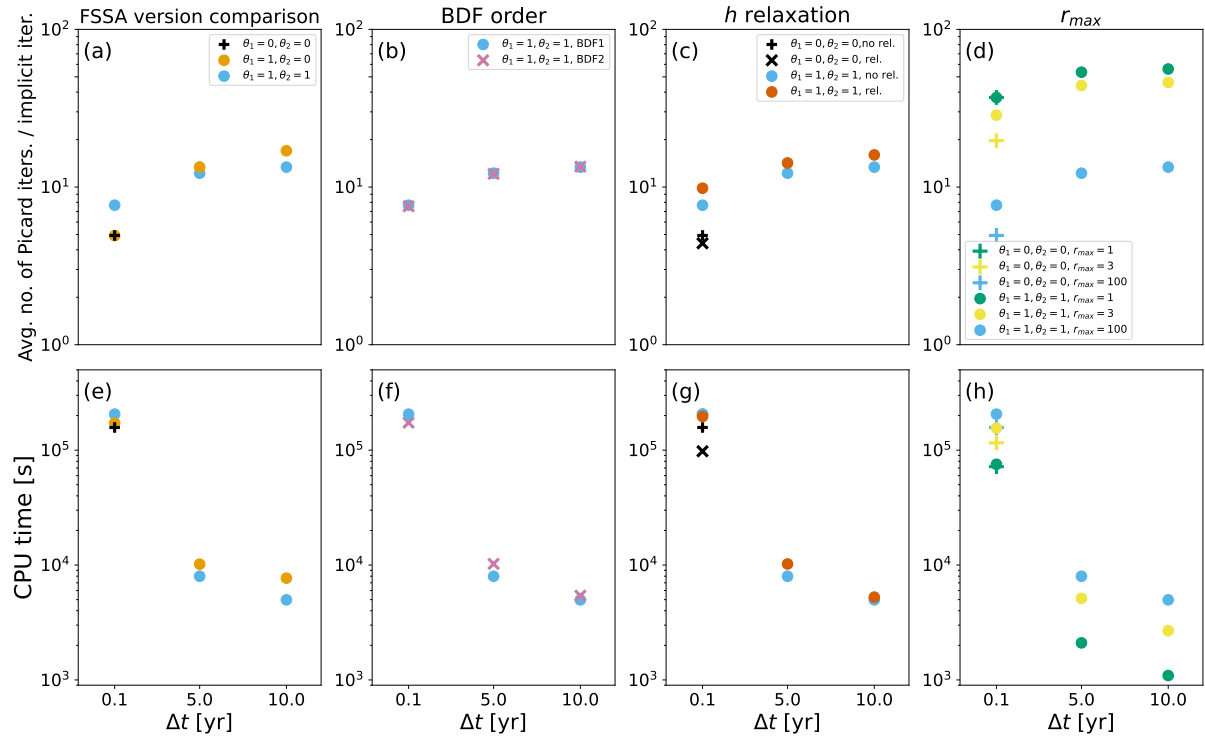



Figure 10. The top panels show the average number of Picard fixed point iterations per implicit iteration, and the bottom panels show the CPU time for the same simulations as in Fig. 9. Panels (a) and (e) illustrate the effect of the FSSA formulation (subtraction-FSSA, $\theta_1 = \theta_2 = 1$, the original FSSA $\theta_1, \theta_2 = 0$ and no FSSA, $\theta_1 = \theta_2 = 0$) across the time step sizes 0.1, 5 and 10 years. These simulations are performed with BDF1 and $r_{max} = 100$, and with no surface relaxation. Panels (b) and (f) illustrate the effect of the BDF order in simulations with $r_{max} = 100$, no surface relaxation and subtraction-FSSA stabilization applied. Panels (c) and (g) illustrate the effect of surface relaxation in simulations with BDF1, $r_{max} = 100$, with and without subtraction-FSSA. Lastly, panels (d) and (h) show the effect of the maximum allowed implicit iterations r_{max}, in simulations with BDF1, $r_{max} = 100$, with and without subtraction-FSSA

iterations to construct higher order schemes appears especially appealing for glacier and ice sheet simulations, where the implicit iterations needed regardless for resolving the minimum thickness constraint with large time-steps. Implementations such as BDF2 add minimal overhead, needing only a small modification to the left-hand side of the free-surface equation (25). Overall, second-order implicit time-stepping with large time-steps is generally more efficient than first-order explicit schemes.

370   Future work will refine stopping criteria for implicit iterations and explore how to further increase accuracy near the glacier front, where a minimum thickness constraint is needed. None-the-less, subtraction-FSSA represents a major advance in accuracy and efficiency. Its integration into Elmer/Ice makes it readily available to the ice sheet community, paving the way for faster, more precise glacier and ice sheet simulations.

Code and data availability. The Perlin glacier experiments were carried out using *Elmer* version 9.0 (Ruokolainen et al., 2023) (https: //doi.org/10.5281/zenodo.7892181). The idealized experiments were performed with *Biceps*, an in-house code available at https://github. com/andrelofgrenSU/biceps (Löfgren, 2025). Additional details and the specific code modifications required to run the simulations are provided at https://zenodo.org/records/17225311 .

.

Author contributions. JA conceptualized the subtraction-FSSA in discussion with AL. ACJH conceptualized the simplified version of the subtraction-FSSA in discussion with all authors. AL and JA implemented the full and simplified method in Biceps, while ACJH implemented the simplified method in Elmer in discussion with JA. AL designed the Biceps experimental set up while AL and AJCH designed the experimental setup in Elmer. JA performed the Biceps-simulations and ACJH performed the Elmer-simulations. All authors contributed to creating figures. JA wrote the original draft, which was edited by all authors

Competing interests. The authors declare that there are no conflicts of interest.

9 Acknowlegements

This research was funded by the Swedish Research Council, grant number 2021-04001, as well as the Swedish e-Science Research Centre (SeRC) and the Wallenberg Foundation (KAW 2021.0275). The paper was written with the assistance of ChatGPT (4 and 5), which was used to clarify the text, summarize the parameter settings in the experiment sections, write bash scripts for running experiments, and write scripts to generate the Python script that was used to create figure Fig. 2. The AI-powered tool Writefull was used to polish the text.

References

Amestoy, P., Duff, I. S., Koster, J., and L'Excellent, J.-Y.: A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling, SIAM Journal on Matrix Analysis and Applications, 23, 15–41, 2001.

Andrés-Martínez, M., Morgan, J. P., Pérez-Gussinyé, M., and Rüpke, L.: A new free-surface stabilization algorithm for geodynamical modelling: Theory and numerical tests, Physics of the Earth and Planetary Interiors, 246, 41 – 51, 2015.

Brezzi, F. and Fortin, M.: Mixed and Hybrid Finite Element Methods, Springer-Verlag New York, Inc., New York, NY, USA, 1991.

Bueler, E.: Stable finite volume element schemes for the shallow-ice Approximation, Journal of Glaciology, 62, 230–242, https://doi.org/10.1017/jog.2015.3, 2016.

Bueler, E.: Performance analysis of high-resolution ice-sheet simulations, Journal of Glaciology, p. 1–6, https://doi.org/10.1017/jog.2022.113, 2022.

Bueler, E.: Surface elevation errors in finite element Stokes models for glacier evolution, https://arxiv.org/abs/2408.06470, 2024.

Cheng, G., Lötstedt, P., and von Sydow, L.: Accurate and stable time stepping in ice sheet modeling, J. Comput. Phys., 329, 29 – 47, https://doi.org/https://doi.org/10.1016/j.jcp.2016.10.060, 2017.

Durand, G., Gagliardini, O., Zwinger, T., Le Meur, E., and Hindmarsh, R. C. A.: Full Stokes modeling of marine ice sheets: influence of the grid size, J. Glaciol., 50, 109–114, 2009.

Franca, L. P. and Frey, S. L.: Stabilized finite element methods: II. The incompressible Navier-Stokes equations, Computer Methods in Applied Mechanics and Engineering, 99, 209–233, https://doi.org/https://doi.org/10.1016/0045-7825(92)90041-H, 1992.

Gagliardini, O., Zwinger, T., Gillet-Chaulet, F., and et al.: Capabilities and performance of Elmer/Ice, a new generation ice-sheet model, Geosci. Model Dev., 6, 1299–1318, 2013.

Greve, R. and Blatter, H.: Dynamics of Ice Sheets and Glaciers, Advances in Geophysical and Environmental Mechanics and Mathematics (AGEM^2), Springer, Berlin, 2009.

Guennebaud, G., Jacob, B., et al.: Eigen v3, http://eigen.tuxfamily.org, 2010.

Henry, C., Zwinger, T., and Ahlkrona, J.: Grounding-line dynamics in a Stokes ice-flow model: Improved numerical stability allows larger time steps, submitted, 2025.

Hirn, A.: Finite Element Approximation Of Singular Power-Law Systems, Mathematics of Computation, 82, 1247–1268, http://www.jstor.org/stable/42002697, 2013.

Kaus, B. J., Mühlhaus, H., and May, D. A.: A stabilization algorithm for geodynamic numerical simulations with a free surface, Phys. Earth Planet. In., 181, 12 – 20, 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), J. Geophys. Res., 117, F01 022, 2012.

Leguy, G. R., Lipscomb, W. H., and Asay-Davis, X. S.: Marine ice sheet experiments with the Community Ice Sheet Model, The Cryosphere, 15, 3229–3253, https://doi.org/10.5194/tc-15-3229-2021, 2021.

Löfgren, A., Ahlkrona, J., and Helanow, C.: Increasing stable time-step sizes of the free-surface problem arising in ice-sheet simulations, Journal of Computational Physics: X, 16, 100 114, https://doi.org/https://doi.org/10.1016/j.jcpx.2022.100114, 2022.

Löfgren, A., Zwinger, T., Råback, P., Helanow, C., and Ahlkrona, J.: Increasing numerical stability of mountain valley glacier simulations: implementation and testing of free-surface stabilization in Elmer/Ice, The Cryosphere, 18, 3453–3470, https://doi.org/10.5194/tc-18-3453-2024, 2024.

Löfgren, A.: Biceps: A prognostic two-dimensional full-Stokes ice-sheet solver, https://github.com/andrelofgrenSU/biceps, accessed: 2025-04-23, 2025.

430 Perlin, K.: An image synthesizer, ACM Siggraph Computer Graphics, 19, 287–296, 1985.

Pörtner, H.-O., Roberts, D., and et al. (eds.): IPCC Special Report on the Ocean and Cryosphere in a Changing Climate, Cambridge University Press, 2019.

Rose, I., Buffett, B., and Heister, T.: Stability and accuracy of free surface time integration in viscous flows, Phys. Earth Plan. In., 262, 90 – 100, 2017.

435 Ruokolainen, J., Malinen, M., Råback, P., Zwinger, T., Takala, E., Kataja, J., Gillet-Chaulet, F., Ilvonen, S., Gladstone, R., Byckling, M., Chekki, M., Gong, C., Ponomarev, P., van Dongen, E., Robertsen, F., Wheel, I., Cook, S., t7saeki, luzpaz, and Rich_B: ElmerCSC/elmer-fem: Elmer 9.0, https://doi.org/10.5281/zenodo.7892181, 2023.

Taylor, C. and Hood, P.: Navier Stokes equations using mixed interpolation, Int. Symp. on Finite Element Methods in Flow Problems, pp. 121–132, 1974.