J. Inverse Ill-Posed Probl. 22 (2014), 787–818
DOI 10.1515/jip-2013-0016

Estimating the ice thickness of mountain glaciers with a shape optimization algorithm using surface topography and mass-balance

Laurent Michel, Marco Picasso, Daniel Farinotti, Andreas Bauder, Martin Funk and Heinz Blatter

Abstract. We present a shape optimization algorithm to estimate the ice thickness distribution within a two-dimensional, non-sliding mountain glacier, given a transient surface geometry and a mass-balance distribution. The approach is based on the minimization of the surface topography misfit at the end of the glacier's evolution in the shallow ice approximation of ice flow. Neither filtering of the surface topography where its gradient vanishes nor interpolation of the basal shear stress is involved. Novelty of the presented shape optimization algorithm is the use of surface topography and mass-balance only within a time-dependent Lagrangian approach for moving-boundary glaciers. On real-world inspired geometries, it is shown to produce estimations of even better quality in smaller time than the recently proposed steady and transient inverse methods. A sensitivity analysis completes the study and evinces the method's higher susceptibility to perturbations in the surface topography than in surface mass-balance or rate factor.

Keywords. Inverse free-surface problems, iterative methods, nonlinear parabolic equations, bedrock topography, glacier, mass-balance.

2010 Mathematics Subject Classification. 49M29, 90C20, 90C30, 90C46, 90C53, 90C90, 35K15, 35K55, 65M06, 65M22, 65M32.

1 Introduction

Recently, inverse approaches that derive the glacier bed topography from surface topography and mass-balance only in both steady and transient situations were proposed [34]. These so-called Steady (SIM) and Transient Inverse Methods (TIM) are based on the Shallow Ice Approximation (SIA) [18] and represent an efficient way to reconstruct the subglacial topography in two dimensions. In particular, neither surface filtering with a lower slope limit nor assumptions on the basal shear stress are applied, contrary to most of the other existing methods [13,20,30,38]. The SIM was inspired by a procedure used to reconstruct river beds in the Shallow Water context [15]. While this method is more reliable and easily implementable than a Shape Optimization Algorithm (SOA) [42] in the Shallow Water Approximation, we will see that this is not the case anymore in the SIA. The

TIM in turn found his basement in a work on the reconstruction of basal topographies of viscous, gravity-driven, steady flows [21].

Many scientists have already adopted optimal control approaches in glaciology to determine basal sliding law and rheology coefficients [2,3,19,27,29,32,39,44] that aim at minimizing the misfit between the computed and exact, or measured, surface velocities with the help of two and three-dimensional flow models of arbitrary order. On subglacial topography estimation with an SOA, the oldest contribution we can think of is an interpolation method that seeks the best solution fitting the measured data under the constraint that it also minimizes potential and curvature [23]. Subsequent work on SOA that infer the ice thickness distribution of a glacier is rather sparse but literature has lately become more abundant on those that minimize the surface velocity misfit on steady geometries [33,35]. The most recent attempt we are currently aware of is a procedure that minimizes the discrepancy between the actual ice thickness and the quasi-stationary SIA corresponding value for a three-dimensional geometry, without direct consideration of surface velocity data [11]. Basically, all existing SOA that solve this problem deal with stationary or quasi-stationary geometries exclusively and use surface velocity data most of the time.

The purpose of this paper is to demonstrate that the SIA can be inverted for the glacier's bedrock topography with an SOA in both steady and transient situations, even if moving boundaries are taken into account, with only surface topography and mass-balance data. Instead of locally reporting to the bedrock topography the discrepancy between the computed and exact final surfaces after each iteration in a fixed-point manner, as in TIM, we now propose to apply a more sophisticated minimization algorithm to it. In practical applications, the exact is replaced with the measured surface topography. The advantage of the new method is its flexibility, as the involved Lagrangian can be easily modified to take additional constraints into account. Moreover, it can incorporate sliding and be extended to three space dimensions as well as higher order models. Last, it handles with the same ease both steady and transient situations. While the validity of the SIA in the case of Alpine glaciers is questionable, we, however, consider useful to test it for ice thickness estimations since it is much easier to implement and less computer memory demanding than the full-Stokes equations. Technically, this is because the ice velocity can be differentiated directly with respect to the bedrock topography in the SIA, while an additional dual problem for the velocity and pressure fields has to be solved in the case of the Stokes equations.

The outline of the paper is as follows: first, we present the governing equation of the two-dimensional SIA. Then, we describe our SOA in the continuous setting, followed by its detailed space and time discretizations. The performance of the method is compared with the results obtained with the SIM and TIM. Finally,

we expose a sensitivity analysis where the high susceptibility to perturbations in surface topography and the low susceptibility to perturbations in surface mass-balance are brought forward, as was the case with SIM and TIM. Additionally, we show that the method is only slightly sensitive to the rate factor, which is important in the Shallow Ice model and usually subjected to errors in practical applications.

2 Forward model

The notations are those introduced in [34]. We consider a two-dimensional longitudinal section of a non-sliding glacier along one of its flow lines, defined in a Cartesian system of axes (x, z), over a spatial domain $\Omega = [x_1, x_2]$ (see Figure 1 (a)), from initial time t_i to final time t_f. Its bed and surface topographies are denoted by b and s respectively, the surface mass-balance by $\mathcal{B}$, and the ice thickness by $\mathcal{H} = s - b$. To simplify the presentation, $\mathcal{B}$ will be a function of space only in what follows. At initial time t_i, the glacier surface topography is s_i and evolves towards topography s_f during the time interval $[t_i, t_f]$ (see Figure 1 (b)).

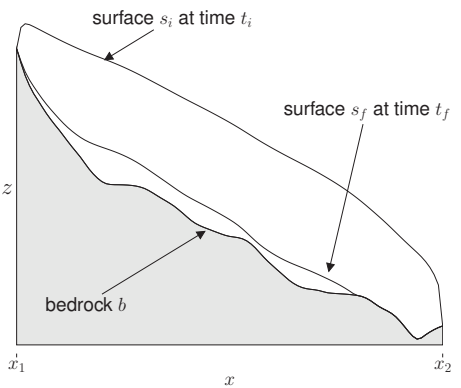

Figure 1. Glacier in the Cartesian system of axes (x, z). The surface s_i at time t_i evolves towards surface s_f during the time interval $[t_i, t_f]$.

We will focus on the SIA of ice dynamics, that is an approximation of order zero of the flow, where the transient forward problem can be cast into: *given a bedrock b, an initial surface s_i, and a surface mass-balance $\mathcal{B}$, find the ice thickness $\mathcal{H} : [x_1, x_2] \times [t_i, t_f] \to \mathbb{R}$ such that*

$$\begin{cases} \dfrac{\partial \mathcal{H}}{\partial t} = \dfrac{\partial}{\partial x}\left(\mathcal{D}(b, \mathcal{H}) \dfrac{\partial (b + \mathcal{H})}{\partial x} \right) + \mathcal{B}, & x_1 < x < x_2, \quad t_i < t < t_f, \\ \mathcal{H} = s_i - b, & x_1 < x < x_2, \quad t = t_i, \\ \mathcal{H} = 0, & x = x_1, x_2, \quad t_i \le t \le t_f, \end{cases} \tag{2.1}$$

where $\mathcal{D}$ is the diffusivity

$$\mathcal{D}(b, \mathcal{H}) = \Gamma \mathcal{H}^{n+2} \left| \frac{\partial (b + \mathcal{H})}{\partial x} \right|^{n-1}, \tag{2.2}$$

n is Glen's flow law exponent [16], and Γ the ice diffusion parameter, defined with

$$\Gamma = 2 \frac{A(\rho g)^n}{n+2}, \tag{2.3}$$

where ρ is the ice density, A the rate factor, and g the acceleration due to gravitation. From the computed thickness $\mathcal{H}$ at time t_f, we deduce the position s_f of the glacier surface at final time from

$$s_f(x) = s(x, t_f) = b(x) + \mathcal{H}(x, t_f). \tag{2.4}$$

Note that we did not specify the time dependence of x_1 and x_2 in $(2.1)_3$. Although the boundaries of the glacier actually move with time, it is not necessary to add this specification into the evolution equation. The mass-balance is the only responsible factor for the boundaries' motion. Equation (2.1) can handle vanishing thicknesses inside the domain $[x_1, x_2]$, as long as one enforces that the ice thickness must remain positive at all times.

3 Shape optimization problem

Consider problem (2.1) when the initial and final surfaces s_i and s_f and the surface mass-balance $\mathcal{B}$ are known but the bedrock geometry is unknown. Our purpose is to design an SOA that reconstructs the subglacial topography. On our synthetical test cases, we know that the glacier surface topography evolves from s_i to s_f over the time frame $[t_i, t_f]$. Therefore, we want to minimize the misfit between the computed and exact surfaces at time t_f. To this end, we set the bedrock topography b as the control variable and define the cost functional

$$\mathcal{J}(\mathcal{H}, b) = \frac{1}{2} \int_{x_1}^{x_2} \left(b + \mathcal{H}|_{t=t_f} - s_f \right)^2 \mathrm{d}x + \mathcal{R}(b), \tag{3.1}$$

where the first term represents the misfit between the computed surface topography, $b + \mathcal{H}|_{t=t_f}$, and the exact surface topography, s_f, and $\mathcal{R}$ is a Tikhonov regularization term introduced to cure the problem's ill-posedness [43]. Following [23] and [35], we choose

$$\mathcal{R}(b) = \frac{\alpha}{2} \int_{x_1}^{x_2} |b'|^2 \mathrm{d}x, \tag{3.2}$$

so as to minimize oscillations in the subglacial topography. The inverse problem

consists in minimizing (3.1) under the constraint

$$\begin{cases} \dfrac{\partial \mathcal{H}}{\partial t} = \dfrac{\partial}{\partial x}\left(\mathcal{D}\dfrac{\partial}{\partial x}(b+\mathcal{H})\right) + \mathcal{B}, & x_1 < x < x_2, \quad t_i \le t \le t_f, \\ \mathcal{H} = s_i - b, & x_1 < x < x_2, \quad t = t_i, \\ \mathcal{H} = 0, & x = x_1, x_2, \quad t_i \le t \le t_f. \end{cases} \tag{3.3}$$

To solve it, we use a BFGS [7, 14, 17, 37, 41] method, where the gradient of functional $\mathcal{J}$ must be calculated. To this end, we consider the Lagrangian associated to our problem, namely

$$\begin{aligned} \mathcal{L}(\mathcal{H}, \lambda, \mu, b) = \int_{t_i}^{t_f} \int_{x_1}^{x_2} \left[\frac{\partial \mathcal{H}}{\partial t}\lambda + \mathcal{D}\frac{\partial}{\partial x}(b+\mathcal{H})\frac{\partial \lambda}{\partial x} - \mathcal{B}\lambda\right] \mathrm{d}x\mathrm{d}t \\ + \int_{x_1}^{x_2} \big(\mathcal{H}|_{t=t_i} - s_i + b\big)\mu \mathrm{d}x + \mathcal{J}(\mathcal{H}, b), \end{aligned} \tag{3.4}$$

where λ is the costate variable, i.e. the Lagrange multiplier corresponding to constraint $(3.3)_1$ and μ the Lagrange multiplier corresponding to the initial state equation $(3.3)_2$. The two integrals augmenting the cost functional $\mathcal{J}$ to our Lagrangian are the weak form of constraint (3.3) where the test functions are replaced with the costate variables. Note that it is not necessary to take the boundary condition into account, as it is implicitly understood in the evolution equation. The solutions of the minimization problem are among the stationary points of $\mathcal{L}$ that satisfy [6]

$$\nabla \mathcal{L}(\mathcal{H}, \lambda, \mu, b) = 0 \tag{3.5}$$

which corresponds to seeking $\mathcal{H}$, λ, μ, and b such that

$$\begin{cases} 0 = \displaystyle\int_{t_i}^{t_f} \int_{x_1}^{x_2} \left[\frac{\partial \hat{\mathcal{H}}}{\partial t}\lambda + (n+2)\mathcal{F}\frac{\partial}{\partial x}(b+\mathcal{H})\frac{\partial \lambda}{\partial x}\hat{\mathcal{H}} + n\mathcal{D}\frac{\partial \hat{\mathcal{H}}}{\partial x}\frac{\partial \lambda}{\partial x}\right] \mathrm{d}x\mathrm{d}t \\ \qquad + \displaystyle\int_{x_1}^{x_2} \left[\mu\hat{\mathcal{H}}|_{t=t_i} + \big(b + \mathcal{H}|_{t=t_f} - s_f\big)\hat{\mathcal{H}}|_{t=t_f}\right] \mathrm{d}x, \\ 0 = \displaystyle\int_{t_i}^{t_f} \int_{x_1}^{x_2} \left[\frac{\partial \mathcal{H}}{\partial t}\hat{\lambda} + \mathcal{D}\frac{\partial}{\partial x}(b+\mathcal{H})\frac{\partial \hat{\lambda}}{\partial x} - \mathcal{B}\hat{\lambda}\right] \mathrm{d}x\mathrm{d}t, \\ 0 = \displaystyle\int_{x_1}^{x_2} \big(\mathcal{H}|_{t=t_i} - s_i + b\big)\hat{\mu}\mathrm{d}x, \\ 0 = -n\displaystyle\int_{t_i}^{t_f} \int_{x_1}^{x_2} \frac{\partial}{\partial x}\left(\mathcal{D}\frac{\partial \lambda}{\partial x}\right)\hat{b}\ \mathrm{d}x\mathrm{d}t \\ \qquad + \displaystyle\int_{x_1}^{x_2} \big(b + \mathcal{H}|_{t=t_f} - s_f + \mu\big)\hat{b}\ \mathrm{d}x + \frac{\partial \mathcal{R}}{\partial b}\hat{b}, \end{cases} \tag{3.6}$$

for all functions $\hat{\mathcal{H}}$ and $\hat{\lambda}$ that vanish at the boundaries x_1 and x_2 of domain Ω and all functions $\hat{\mu}$ and $\hat{b}$ defined on Ω, where

$$\mathcal{F}(b, \mathcal{H}) = \Gamma \mathcal{H}^{n+1} \left| \frac{\partial}{\partial x}(b + \mathcal{H}) \right|^{n-1}. \tag{3.7}$$

The first equation of (3.6) is the weak formulation of the backward diffusion-transport equation

$$\begin{cases} -\dfrac{\partial \lambda}{\partial t} + (n+2)\mathcal{F}\dfrac{\partial}{\partial x}(b + \mathcal{H})\dfrac{\partial \lambda}{\partial x} \\ \qquad - n\dfrac{\partial}{\partial x}\left(\mathcal{D}\dfrac{\partial \lambda}{\partial x}\right) = 0, & x_1 < x < x_2, \quad t_i \le t \le t_f, \\ \qquad \lambda = s_f - b - \mathcal{H}, & x_1 < x < x_2, \quad t = t_f, \\ \qquad \lambda = 0, & x = x_1, x_2, \quad t_i \le t \le t_f, \end{cases} \tag{3.8}$$

as well as

$$\mu = \lambda, \quad x_1 < x < x_2, \; t = t_i. \tag{3.9}$$

The second and third equations of (3.6) are the weak formulation of (3.3). Understanding variable $\mathcal{H}$ as a function of the bedrock topography, we define

$$J(b) = \mathcal{J}(\mathcal{H}(b), b). \tag{3.10}$$

Then, equation $(3.6)_4$ is the Fréchet derivative of functional (3.10), i.e.

$$\begin{aligned} J'(b)\hat{b} = -n \int_{t_i}^{t_f} \int_{x_1}^{x_2} \frac{\partial}{\partial x}\left(\mathcal{D}\frac{\partial \lambda}{\partial x}\right)\hat{b} \, \mathrm{d}x\mathrm{d}t \\ + \int_{x_1}^{x_2} \big(b + \mathcal{H}|_{t=t_f} - s_f + \mu\big)\hat{b} \, \mathrm{d}x + \frac{\partial \mathcal{R}}{\partial b}\hat{b}, \end{aligned} \tag{3.11}$$

where

$$\frac{\partial \mathcal{R}}{\partial b}\hat{b} = \alpha \int_{x_1}^{x_2} b'\hat{b}' \mathrm{d}x. \tag{3.12}$$

Remark that the gradient of (3.10) can be computed directly without consideration of Lagrangian (3.4). In this analogous procedure, a forward, linear diffusion problem replaces the dual problem (3.8). However, such a method cannot be extended to higher order models, like a First Order Approximation (FOA) or a full-Stokes Approximation of ice flow, as it needs an explicit expression for the ice velocity.

In general, snow may accumulate up-glacier and the glacier's tongue retreat or advance, i.e. the glacier's boundaries may not be the same at all times. In particular,

it may be that part of the bedrock topography inside domain Ω is known. Let $\tilde{\Omega} \subset \Omega$ be the domain where the bedrock topography is unknown. Then, to take moving boundaries into account, we enforce that

$$b = b^*, \quad \text{for } x \in \Omega \setminus \tilde{\Omega}, \tag{3.13}$$

where b^* is the given topography where it is known. The consideration of this constraint is chosen for its ease of implementation and reliability. Instead, we could incorporate a term of the form

$$\frac{1}{2} \int_{\Omega \setminus \tilde{\Omega}} (b - b^*)^2 \mathrm{d}x \tag{3.14}$$

into the cost functional (3.1) and restrict the original integral to the space domain $\tilde{\Omega}$ instead of Ω or even incorporate (3.14) and keep the original integral as it is. Such modifications of the original model for fixed boundary problems are delicate especially because of the needed adaption of the regularization term (3.12). All this leads to unnecessary complications, hence we decide to minimize the cost functional $\mathcal{J}$ over the whole domain Ω, with a regularization also applied to the entire Ω, with the consideration of constraint (3.13).

Additionally, in the case where $\mathcal{B}$ has negative values on part of Ω, constraint (3.3) may lead to negative thicknesses. Here, this problem is avoided by considering $\max\{0, \mathcal{H}\}$ instead of $\mathcal{H}$. Such a choice makes the algorithm conceptually simpler, at the expense of losing efficiency to determine the glacier's time-dependent boundaries compared to a formulation as an obstacle problem [24–26], which is translated by the need of small time steps in the discretization described below. Since the projection of thicknesses onto positive functions is not differentiable and was not taken into account in the dual problem (3.8), instabilities may arise in the algorithm.

However, numerical experiments show that we can circumvent this difficulty by seeking a control b that satisfies

$$b < \min_{t_i \le t \le t_f} s(x,t), \quad x \in \tilde{\Omega}. \tag{3.15}$$

In practice, we now discretize (3.3), (3.8), (3.9), and (3.11) in space and time. Then, we apply Algorithm 1 until stopping criterion

$$\|b^{m+1} - b^m\| = \max_{x_1 < x < x_2} |b^{m+1}(x) - b^m(x)| < \delta \tag{3.16}$$

is met, for a real constant tolerance $\delta > 0$, where b^m is the bedrock estimation at iteration m. We solve the corresponding linear systems with the PETSc Library [4,5]

and realize the line search as well as the BFGS algorithm with the "Bounded Limited Memory Variable Metric" (BLMVM) method available through the Toolkit for Advanced Optimization (TAO), with the default values for its parameters [36]. The choice of the bounded version of LMVM is due to bounds (3.13) and (3.15) because it allows us to avoid the consideration of the KKT conditions [28].

Algorithm 1. Shape optimization algorithm.

set glacier geometry with initial bedrock b^0 and surface s_i
while $\|b^{m+1} - b^m\| > \delta$ **do**
  solve the primal problem (3.3)
  solve the dual problem (3.8)–(3.9)
  compute the gradient (3.11)
  compute

$$b^{m+1} = b^m + \left(\frac{\mathrm{d}^2 J}{\mathrm{d}b^2}(b^m)\right)^{-1} \frac{\mathrm{d}J}{\mathrm{d}b}(b^m),$$

  with BLMVM that takes care of the bounds (3.13) and (3.15), where the Hessian is approximated with the BFGS method.
end while

4 Space and time discretization

Let $\Delta x > 0$ be some space step and $\xi_j = x_1 + j\Delta x$, $0 \le j \le N+1$, vertices in $[x_1, x_2]$ such that $\xi_0 = x_1$ and $\xi_{N+1} = x_2$. Let $\Delta t > 0$ be some time step and $t^\ell = t_i + \ell \Delta t$, $0 \le \ell \le M+1$, such that $t^0 = t_i$ and $t^{M+1} = t_f$. Furthermore, let us denote the approximations of the thickness $\mathcal{H}(\xi_j, t^\ell)$, bedrock topography $b(\xi_j)$, surface topography $s(\xi_j, t^\ell)$, and costate variables $\lambda(\xi_j, t^\ell)$ and $\mu(\xi_j)$ at location ξ_j and time t^ℓ with $\mathcal{H}_j^\ell$, b_j, s_j^ℓ, λ_j^ℓ, and μ_j respectively. Let us denote with $\boldsymbol{\mathcal{H}}^\ell$, $\boldsymbol{b}$, $\boldsymbol{s}^\ell$, $\boldsymbol{\lambda}^\ell$, $\boldsymbol{\mu}$, $\boldsymbol{s}_i$, and $\boldsymbol{s}_f$ the vectors of components $\mathcal{H}_j^\ell$, b_j, s_j^ℓ, λ_j^ℓ, μ_j, $s_i(\xi_j)$, and $s_f(\xi_j)$. In these notations, we can write

$$\boldsymbol{s}^\ell = \boldsymbol{b} + \boldsymbol{\mathcal{H}}^\ell.$$

Let us approximate (2.2) and (3.7) with

$$\mathcal{D}_{j-\frac{1}{2}}^\ell = \Gamma\left(\frac{\mathcal{H}_j^\ell + \mathcal{H}_{j-1}^\ell}{2}\right)^{n+2} \left|\frac{s_j^\ell - s_{j-1}^\ell}{\Delta x}\right|^{n-1} \tag{4.1}$$

and

$$\mathcal{F}_{j-\frac{1}{2}}^\ell = \Gamma\left(\frac{\mathcal{H}_j^\ell + \mathcal{H}_{j-1}^\ell}{2}\right)^{n+1} \left|\frac{s_j^\ell - s_{j-1}^\ell}{\Delta x}\right|^{n-1}. \tag{4.2}$$

From now on, the space index j always satisfies $1 \le j \le N$ unless stated otherwise. We approach the solution of (3.3) with the following finite-difference scheme, centered in space, Crank–Nicolson in time [1,8,9,31,34]:

$$\begin{cases} \dfrac{\boldsymbol{\mathcal{H}}^{\ell+1} - \boldsymbol{\mathcal{H}}^{\ell}}{\Delta t} = \boldsymbol{\mathcal{T}}(\boldsymbol{\mathcal{H}}^{\ell}, \boldsymbol{\mathcal{H}}^{\ell+1}, \boldsymbol{b}), & 0 \le \ell \le M, \\ \boldsymbol{\mathcal{H}}^{0} = \boldsymbol{s}_i - \boldsymbol{b}, & \\ \mathcal{H}_0^{\ell} = \mathcal{H}_{N+1}^{\ell} = 0, & 0 \le \ell \le M+1, \end{cases} \tag{4.3}$$

where

$$\mathcal{T}_j^{\ell,\ell+1} = \frac{1}{2\Delta x}\left(\mathcal{D}_{j+\frac{1}{2}}^{\ell}\frac{s_{j+1}^{\ell} - s_j^{\ell}}{\Delta x} - \mathcal{D}_{j-\frac{1}{2}}^{\ell}\frac{s_j^{\ell} - s_{j-1}^{\ell}}{\Delta x}\right) + \frac{1}{2\Delta x}\left(\mathcal{D}_{j+\frac{1}{2}}^{\ell+1}\frac{s_{j+1}^{\ell+1} - s_j^{\ell+1}}{\Delta x} - \mathcal{D}_{j-\frac{1}{2}}^{\ell+1}\frac{s_j^{\ell+1} - s_{j-1}^{\ell+1}}{\Delta x}\right) + \mathcal{B}_j. \tag{4.4}$$

We now deduce the discretizations of the dual problem (3.8)–(3.9) as well as the gradient (3.11). The corresponding equations in the semi-implicit case are given in Appendix A. The discretized objective function reads

$$\mathcal{J}_h(\boldsymbol{\mathcal{H}}, \boldsymbol{b}) = \frac{1}{2}\|\boldsymbol{b} + \boldsymbol{\mathcal{H}}^{M+1} - \boldsymbol{s}_f\|_2^2 + \mathcal{R}_h(\boldsymbol{b}), \tag{4.5}$$

where $\|\cdot\|_2$ stands for the usual Euclidean norm, $\boldsymbol{\mathcal{H}}$ is the vector of components $\mathcal{H}_j^{\ell}$, $0 \le \ell \le M+1$, $0 \le j \le N+1$, and

$$\mathcal{R}_h(\boldsymbol{b}) = \frac{\alpha}{2}\sum_{j=0}^{N}\left|\frac{b_{j+1} - b_j}{\Delta x}\right|^2. \tag{4.6}$$

We define the discretized Lagrangian with

$$\begin{aligned} \mathcal{L}_h(\boldsymbol{\mathcal{H}}, \boldsymbol{\lambda}, \boldsymbol{b}) = {} & \left[\boldsymbol{\mathcal{H}}^{0} - \boldsymbol{s}_i + \boldsymbol{b}\right]^T \boldsymbol{\lambda}^{0} \\ & + \sum_{\ell=0}^{M} \Delta t\left[\frac{\boldsymbol{\mathcal{H}}^{\ell+1} - \boldsymbol{\mathcal{H}}^{\ell}}{\Delta t} - \boldsymbol{\mathcal{T}}(\boldsymbol{\mathcal{H}}^{\ell}, \boldsymbol{\mathcal{H}}^{\ell+1}, \boldsymbol{b})\right]^T \boldsymbol{\lambda}^{\ell+1} \\ & + \sum_{\ell=0}^{M+1}\left(\mathcal{H}_0^{\ell}\lambda_0^{\ell} + \mathcal{H}_{N+1}^{\ell}\lambda_{N+1}^{\ell}\right) + \mathcal{J}_h(\boldsymbol{\mathcal{H}}, \boldsymbol{b}), \end{aligned} \tag{4.7}$$

where $\boldsymbol{\lambda}^{0}$ plays the role of μ in (3.4) and $\boldsymbol{\lambda}$ the vector of components λ_j^{ℓ}, $0 \le \ell \le M+1$, $0 \le j \le N+1$. As in the continuous case, the solutions of the minimization problem are among the stationary points of $\mathcal{L}_h$ that satisfy

$$\nabla \mathcal{L}_h(\boldsymbol{\mathcal{H}}, \boldsymbol{\lambda}, \boldsymbol{b}) = 0. \tag{4.8}$$

The resulting discretized dual problem reads

$$\begin{cases} -\dfrac{\boldsymbol{\lambda}^{\ell+1}-\boldsymbol{\lambda}^{\ell}}{\Delta t} = \tilde{\boldsymbol{\mathcal{T}}}(\boldsymbol{\lambda}^{\ell},\boldsymbol{\lambda}^{\ell+1}), & 1 \le \ell \le M, \\ \lambda_0^{\ell} = \lambda_{N+1}^{\ell} = 0, & 0 \le \ell \le M+1, \end{cases} \tag{4.9}$$

where

$$\begin{aligned} \tilde{\mathcal{T}}_j^{\ell,\ell+1} = -\frac{(n+2)}{4}\Bigg[& \mathcal{F}_{j+\frac{1}{2}}^{\ell}\frac{s_{j+1}^{\ell}-s_j^{\ell}}{\Delta x}\left(\frac{\lambda_{j+1}^{\ell}-\lambda_j^{\ell}}{\Delta x}+\frac{\lambda_{j+1}^{\ell+1}-\lambda_j^{\ell+1}}{\Delta x}\right) \\ & + \mathcal{F}_{j-\frac{1}{2}}^{\ell}\frac{s_j^{\ell}-s_{j-1}^{\ell}}{\Delta x}\left(\frac{\lambda_j^{\ell}-\lambda_{j-1}^{\ell}}{\Delta x}+\frac{\lambda_j^{\ell+1}-\lambda_{j-1}^{\ell+1}}{\Delta x}\right)\Bigg] \\ & + \frac{n}{2\Delta x}\Bigg[\mathcal{D}_{j+\frac{1}{2}}^{\ell}\left(\frac{\lambda_{j+1}^{\ell}-\lambda_j^{\ell}}{\Delta x}+\frac{\lambda_{j+1}^{\ell+1}-\lambda_j^{\ell+1}}{\Delta x}\right) \\ & - \mathcal{D}_{j-\frac{1}{2}}^{\ell}\left(\frac{\lambda_j^{\ell}-\lambda_{j-1}^{\ell}}{\Delta x}+\frac{\lambda_j^{\ell+1}-\lambda_{j-1}^{\ell+1}}{\Delta x}\right)\Bigg], \end{aligned} \tag{4.10}$$

whose final and initial conditions corresponding to $(3.8)_2$ and (3.9) are

$$\begin{aligned} & \lambda_j^{M+1} + (n+2)\frac{\Delta t}{4}\Bigg(\mathcal{F}_{j+\frac{1}{2}}^{M+1}\frac{s_{j+1}^{M+1}-s_j^{M+1}}{\Delta x}\frac{\lambda_{j+1}^{M+1}-\lambda_j^{M+1}}{\Delta x} \\ & \qquad + \mathcal{F}_{j-\frac{1}{2}}^{M+1}\frac{s_j^{M+1}-s_{j-1}^{M+1}}{\Delta x}\frac{\lambda_j^{M+1}-\lambda_{j-1}^{M+1}}{\Delta x}\Bigg) \\ & \quad - n\frac{\Delta t}{2\Delta x}\Bigg(\mathcal{D}_{j+\frac{1}{2}}^{M+1}\frac{\lambda_{j+1}^{M+1}-\lambda_j^{M+1}}{\Delta x} - \mathcal{D}_{j-\frac{1}{2}}^{M+1}\frac{\lambda_j^{M+1}-\lambda_{j-1}^{M+1}}{\Delta x}\Bigg) \\ & = s_f(\xi_j) - b_j - \mathcal{H}_j^{M+1}, \end{aligned} \tag{4.11}$$

and

$$\begin{aligned} \lambda_j^0 = \lambda_j^1 - (n+2)\frac{\Delta t}{4}\Bigg(& \mathcal{F}_{j+\frac{1}{2}}^{0}\frac{s_{j+1}^{0}-s_j^{0}}{\Delta x}\frac{\lambda_{j+1}^{1}-\lambda_j^{1}}{\Delta x} \\ & + \mathcal{F}_{j-\frac{1}{2}}^{0}\frac{s_j^{0}-s_{j-1}^{0}}{\Delta x}\frac{\lambda_j^{1}-\lambda_{j-1}^{1}}{\Delta x}\Bigg) \\ & + n\frac{\Delta t}{2\Delta x}\Bigg(\mathcal{D}_{j+\frac{1}{2}}^{0}\frac{\lambda_{j+1}^{1}-\lambda_j^{1}}{\Delta x} - \mathcal{D}_{j-\frac{1}{2}}^{0}\frac{\lambda_j^{1}-\lambda_{j-1}^{1}}{\Delta x}\Bigg). \end{aligned} \tag{4.12}$$

Moreover, the derivative of

$$J_h(\boldsymbol{b}) = \mathcal{J}_h(\mathcal{H}(\boldsymbol{b}), \boldsymbol{b}) \tag{4.13}$$

can be shown to be

$$\begin{aligned}\frac{\partial J_h}{\partial b_j}(\boldsymbol{b}) = -\frac{n\Delta t}{2\Delta x}\sum_{\ell=0}^{M}\Bigg[&\Big(\mathcal{D}^{\ell}_{j+\frac{1}{2}} + \mathcal{D}^{\ell+1}_{j+\frac{1}{2}}\Big)\frac{\lambda^{\ell+1}_{j+1} - \lambda^{\ell+1}_{j}}{\Delta x} \\ &- \Big(\mathcal{D}^{\ell}_{j-\frac{1}{2}} + \mathcal{D}^{\ell+1}_{j-\frac{1}{2}}\Big)\frac{\lambda^{\ell+1}_{j} - \lambda^{\ell+1}_{j-1}}{\Delta x}\Bigg] \\ &+ b_j + \mathcal{H}^{M+1}_j - s_f(\xi_j) + \lambda^0_j + \frac{\partial \mathcal{R}_h}{\partial b_j}(\boldsymbol{b}),\end{aligned} \tag{4.14}$$

where

$$\frac{\partial \mathcal{R}_h}{\partial b_j}(\boldsymbol{b}) = \frac{\alpha}{\Delta x^2}\big(2b_j - b_{j-1} - b_{j+1}\big). \tag{4.15}$$

Note that the Lagrangian (4.7) reduces to

$$\mathcal{L}_h(\boldsymbol{\mathcal{H}}, \boldsymbol{\lambda}, \boldsymbol{b}) = \big[\boldsymbol{\mathcal{T}}(\boldsymbol{\mathcal{H}}, \boldsymbol{b})\big]^T\boldsymbol{\lambda} + \big(\mathcal{H}_0\lambda_0 + \mathcal{H}_{N+1}\lambda_{N+1}\big) + \mathcal{J}_h(\boldsymbol{\mathcal{H}}, \boldsymbol{b}), \tag{4.16}$$

when the superscript ℓ is dropped. One may want to deduce the steady counterparts of (4.10), (4.11), (4.12), and (4.14). We do not advise such a procedure for steady geometries because it is time-consuming and lacks robustness. Indeed, it is not clear whether a stationary state exists for both the primal and dual problems, starting from any bedrock topography b^m and initial surface s_i. Moreover, its computation may take a large amount of time, especially during the first BFGS iterations. In practice, we rather solve the steady problem time-dependently, by setting $s_i = s_f = s^*$ and choosing a large enough time frame $[t_i, t_f]$. Instead of computing the stationary state at every iteration, we only compute the evolution of the surface topography over the time frame $[t_i, t_f]$ and study its deviation from the true stationary geometry. Because of stationarity, such a deviation must be zero when the algorithm has converged. This procedure saves significant computational effort compared to a purely stationary algorithm.

The precision of the gradient (4.14) is crucial for our computations. To check it, we consider Taylor's expansion of J_h,

$$J_h(\boldsymbol{b} + \varepsilon\hat{\boldsymbol{b}}) = J_h(\boldsymbol{b}) + \varepsilon\frac{\mathrm{d}J_h}{\mathrm{d}\boldsymbol{b}}(\boldsymbol{b})\cdot\hat{\boldsymbol{b}} + \mathcal{O}(\varepsilon^2), \tag{4.17}$$

and verify that [10]

$$\lim_{\varepsilon\to 0}\frac{J_h(\boldsymbol{b} + \varepsilon\hat{\boldsymbol{b}}) - J_h(\boldsymbol{b})}{\varepsilon\frac{\mathrm{d}J_h}{\mathrm{d}\boldsymbol{b}}(\boldsymbol{b})\cdot\hat{\boldsymbol{b}}} = 1. \tag{4.18}$$

To this end, we choose some reasonable vector $\boldsymbol{b}$ and for example the normalized vector

$$\hat{\boldsymbol{b}} = \frac{\frac{\mathrm{d}J_h}{\mathrm{d}\boldsymbol{b}}(\boldsymbol{b})}{\|\frac{\mathrm{d}J_h}{\mathrm{d}\boldsymbol{b}}(\boldsymbol{b})\|}. \tag{4.19}$$

Our numerical experiments verify relation (4.18). Hence our implementation of the discretizations of (3.3), (3.8), (3.9), and (3.11) can be trusted.

In the next section, we present numerical results obtained with this method applied to the same geometries as [34], namely the Flask outlet glacier in Antarctica as well as the Muragl, Silvretta, and Rhône mountain glaciers in the Swiss Alps. These glaciers were chosen because their subglacial geometry either could be measured [12, 13] or is visible as ice has completely vanished [40]. The geometry of Flask glacier is adapted to produce an additional test case, so that we have zero thickness at its boundaries and no sliding. A confrontation with the results obtained with SIM and TIM is carried out. Note that the TIM can be viewed as a special case of the SOA, where we set

$$\begin{aligned} &\lambda = 0, \quad x_1 \leq x \leq x_2,\ t_i \leq t \leq t_f, \\ &\mu = 0, \quad x_1 \leq x \leq x_2. \end{aligned} \tag{4.20}$$

We will see how convergence rate, number of iterations, computational time, and accuracy are improved with the SOA. We first investigate the steady case that we compare to the SIM and TIM. We then switch to transient geometries and examine the differences with the TIM in both the fixed and moving-boundary situations.

5 Numerical results

For the sake of numerical validation, we use available data on two-dimensional glacier evolution according to equation (2.1). In other words, this means that we first choose a bedrock topography b^*, an initial surface topography s_i, and a surface mass-balance $\mathcal{B}$ from which we compute the final surface geometry s_f with equation (2.1). Topography b^* is termed "solution bedrock." Then, we perturb b^* and give it as an initial guess b^0 to Algorithm 1, whose aim will be to recover the solution bedrock b^*. In the tables below, we denote the computed bedrock topography by b^c and the final surface topography obtained with b^c by s_f^c. We focus on the inversion of the SIA equations on the aforementioned synthetical geometries inspired from real-world data. Every computation is done with $\Delta x = 25\,\mathrm{m}$, $\Delta t = 10^{-3}\,\mathrm{a}$, $\delta = 10^{-6}\,\mathrm{m}$, $\rho = 900\,\mathrm{kg}\cdot\mathrm{m}^{-3}$, and $g = 9.81\,\mathrm{m}\cdot\mathrm{s}^{-2}$. Furthermore, from now on, we set $n = 3$, $A = 0.1\,\mathrm{bar}^{-3}\mathrm{a}^{-1}$, and use the norm

$$\|\cdot\| = \max_{x_1<x<x_2} |\cdot|. \tag{5.1}$$

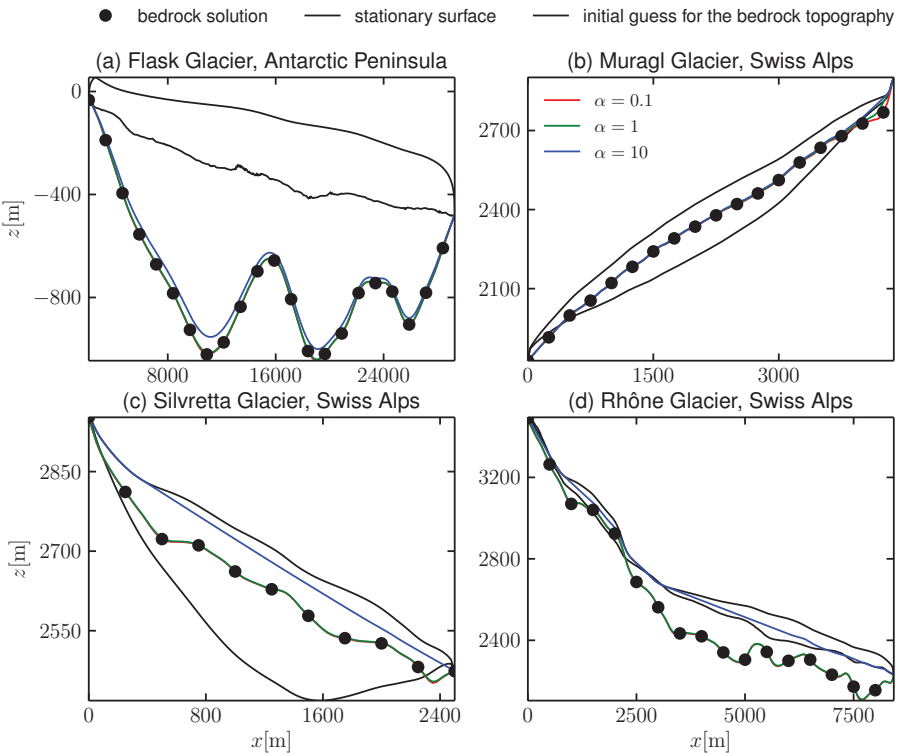

Figure 2. Various examples illustrating the application of Algorithm 1 to steady geometries, with $\Delta x = 25\,\text{m}$, $\Delta t = 10^{-3}\,\text{a}$, $\delta = 10^{-6}$, and $t_f - t_i = 1\,\text{a}$ for different values of the regularization parameter α (red, green, and blue lines). The continuous black line represents the stationary surface and the dashed line the initial guess b^0 for the bedrock topography. The white circles delineate the solution bedrock. When α is small, the computed and the exact solution are hard to distinguish. Errors in the bedrock and final surface topography estimations are listed in Table 1 for $\alpha = 1$ and various time frames. Table 2 lists the errors, computational times, and numbers of iterations when α is such that the error in the basal estimation is comparable to the one obtained with SIM for each geometry.

5.1 Steady geometries

We solve the stationary control problem with the SOA and compare the results to those obtained with the SIM and TIM. The purpose is to investigate the differences in accuracy and computational time. Let us first apply the SOA with various parameters α to the four previously mentioned glaciers. Figure 2 shows the resulting estimations when the time frame is 1 a. Most of the time, the large value $\alpha = 10$

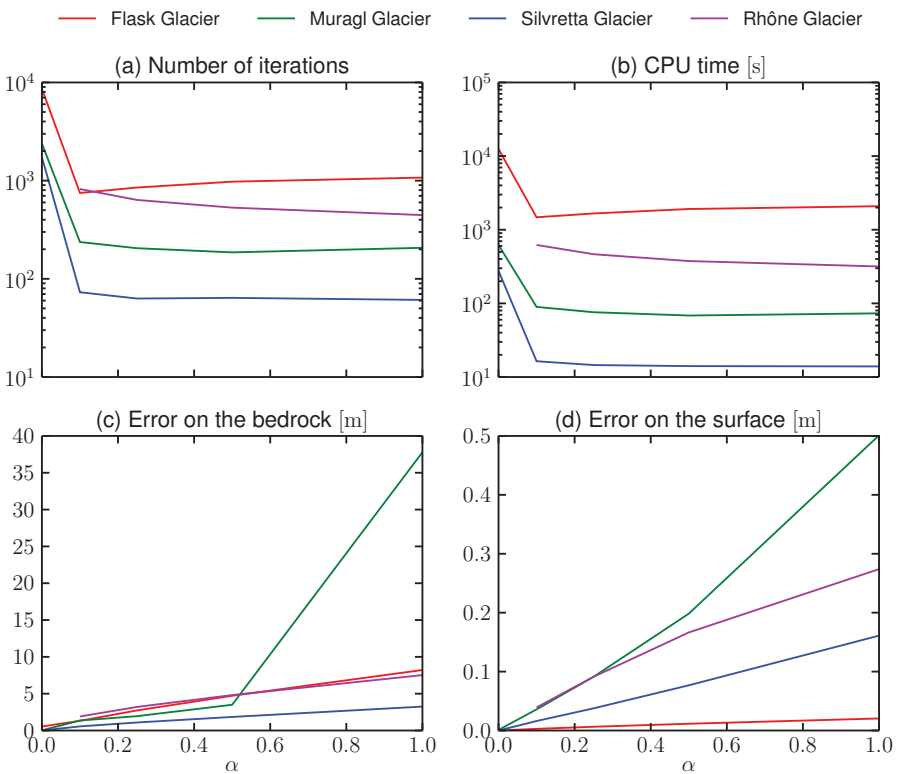

Figure 3. Sensitivity of the method to the regularization parameter when $\Delta x = 25\,\mathrm{m}$, $\Delta t = 10^{-3}\,\mathrm{a}$, $\delta = 10^{-6}\,\mathrm{m}$, and a time frame of 1 a for steady geometries. Variations in (a) number of iterations, (b) computational time, (c) error on the bedrock topography, and (d) error on the computed surface from the estimated bedrock topography are shown as a function of α. The number of iterations and the computational time are directly proportional. The errors grow with the regularization parameter. Differences between the various geometries can be large and, already in the range we chose α, the error in the subglacial estimation can be large (e.g. on Muragl glacier). Errors on the bedrock geometry are about one order of magnitude larger than the errors on the surface topography.

leads to estimations too far from the solution, while the other ones are close to it. When α is small enough, we cannot distinguish any large discrepancy between the estimated subglacial topography and the solution bedrock. Figure 3 shows how sensitive the SOA is to the regularization parameter α. Figure 3 (a) and (b) evince that it is not useful to increase α in order to decrease the computational time. Furthermore, the error in the subglacial topography due to the increase of α is so large that this parameter should be kept as small as possible, although the error in the

final surface remains one order of magnitude smaller (see Figures 3 (c) and (d)), indicating a low response on the surface topography to large fluctuations in the bedrock geometry. Table 1 gives the errors on the geometries displayed by Figure 2 for $\alpha = 1$ and several time frames. We notice that for fixed α, the larger the time frame, the more accurate the estimation. Not shown here, the computational time increases with the size of the time frame, given that the primal and dual problems require more time steps. On the other hand, if the time frame is too small, the algorithm has not enough information to deduce the subglacial topography, which explains the large errors in Table 1's first column. Although this error is large, the error on the final surface is small due to the problem's ill-posedness. Over this small time frame, even a very wrong bedrock can lead to a final surface close to the desired steady surface geometry.

	$t_f - t_i = 10^{-1}$ a		$t_f - t_i = 1$ a	
Figure 3	$\|b^c - b^*\|$ [m]	$\|s_f^c - s^*\|$ [m]	$\|b^c - b^*\|$ [m]	$\|s_f^c - s^*\|$ [m]
(a)	8.207e+02	6.325e−02	8.215e+00	2.043e−02
(b)	6.907e+01	5.410e−02	3.778e+01	5.010e−01
(c)	8.837e+01	6.176e−02	3.249e+00	1.610e−01
(d)	2.398e+02	1.784e−01	7.526e+00	2.739e−01

	$t_f - t_i = 10$ a	
Figure 3	$\|b^c - b^*\|$ [m]	$\|s_f^c - s^*\|$ [m]
(a)	3.353e+00	1.546e−02
(b)	1.554e+00	7.354e−02
(c)	–	–
(d)	4.630e+00	1.118e−01

Table 1. Discrepancy $\|b^c - b\|$ in the bedrock and discrepancy $\|s_f^c - s^*\|$ in the final surface topography as a function of the time frame on the four steady geometries represented in Figure 2 when the SOA with $\alpha = 1$ is applied. With the size of the time frame, the estimation improves most of the time. A hyphen is printed where the algorithm did not converge.

With smaller values of α, we can be as accurate as with the SIM. This is illustrated by Table 2, which shows the discrepancies, number of iterations, and

	$\|b^c - b^*\|$ [m]		$\|s_f^c - s^*\|$ [m]	
Figure 3	SIM	SOA	SIM	SOA
(a)	1.180e+00	1.180e+00	1.090e−02	2.472e−03
(b)	5.973e−02	5.491e−02	5.297e−02	2.849e−05
(c)	1.305e−01	1.305e−01	7.719e−02	2.079e−03
(d)	2.101e+01	1.374e+01	9.272e−01	3.810e−01

	n_{its}		T_{CPU} [s]	
Figure 3	SIM	SOA	SIM	SOA
(a)	1,527,130	756	3.580e+03	1.014e+03
(b)	96,089,700	2,119	4.383e+04	3.864e+02
(c)	9,613,360	198	3.093e+03	3.400e+01
(d)	13,132,500	546	9.974e+03	2.541e+02

Table 2. Discrepancies $\|b^c - b^*\|$ and $\|s_f^c - s^*\|$, number of iterations, and computational time needed to converge in both the SIM and SOA. In the SOA, the time frame was set to 1 a. Regularization parameters used (ε corresponds to SIM and α to SOA): (a) $\varepsilon = 10^{-6}$, $\alpha = 8.697 \cdot 10^{-2}$, (b) $\varepsilon = 10^{-4}$, $\alpha = 1.53 \cdot 10^{-5}$, (c) $\varepsilon = 10^{-4}$, $\alpha = 8.751 \cdot 10^{-3}$, (d) $\varepsilon = 10^{-3}$, $\alpha = 2.2258$. Parameter α was chosen in such a way that the subglacial discrepancy $\|b^c - b^*\|$ is the same in both cases in order to allow for a comparison. On geometry (d), this error is different, because no parameter α could be found to make both discrepancies match exactly. However, this does not affect the comparison significantly. Contrarily to the errors and computational times, the number of iterations of the two methods cannot be compared, since they are completely different. The solution provided by the SOA is of higher quality, because for the same subglacial discrepancy $\|b^c - b^*\|$ a smaller surface discrepancy $\|s_f^c - s^*\|$ is obtained. Computational times in the SOA are also smaller than in the SIM.

computational times with α chosen such that the same errors in the bedrock topography as with the SIM are obtained for each geometry. Note that columns 1 and 2 of the second part of Table 2, listing the number of iterations in each case, cannot be compared, because one iteration of the SIM corresponds to one single time step of the glacier evolution towards the steady state, while it is one whole simulation of the glacier evolution over the chosen time frame in the SOA. However, the computational time and the error on the final surface can be confronted.

		$\|b^c - b^*\|$ [m]		$\|s_f^c - s^*\|$ [m]	
Figure 3	α	TIM	SOA	TIM	SOA
(a)	2.480e−01	2.711e+00	2.711e+00	5.906e−01	6.333e−03
(b)	1.090e−01	1.390e+00	1.390e+00	5.080e−01	3.905e−02
(c)	3.270e+00	2.293e+01	2.293e+01	1.575e+01	5.001e−01
(d)	5.920e−01	5.358e+00	5.358e+00	2.340e+00	1.906e−01

		n_{its}		T_{CPU} [s]	
Figure 3	α	TIM	SOA	TIM	SOA
(a)	2.480e−01	17,473	854	1.893e+04	1.178e+03
(b)	1.090e−01	745	233	2.190e+02	9.110e+01
(c)	3.270e+00	2,980	160	6.315e+02	3.449e+01
(d)	5.920e−01	3,219	498	1.418e+03	2.581e+02

Table 3. Discrepancies $\|b^c - b^*\|$ and $\|s_f^c - s^*\|$ in bedrock and final surface topographies, number of iterations, and computational time needed to converge in both the TIM and SOA applied to a steady geometry, when the time frame is 1 a. In the TIM, we set the regularization and fixed-point parameters to $\varepsilon = 100$ and $\beta = 0.1$ respectively. As in the comparison with the SIM (see Table 2), the quality of the estimation is improved with the SOA. The number of iterations of the two methods can be compared and is much smaller with the SOA. Since we simulate fewer evolutions with SOA, its computational time is smaller. The listed regularization parameters α were chosen such that the discrepancy $\|b^c - b^*\|$ is the same in both cases.

We remark that the computational time needed to get an estimation with the same error on b is every time smaller when we use the SOA. Moreover, the computed final surface s_f^c is each time closer to the exact steady surface s^*. Hence, the SOA provides us with a bedrock topography of higher quality than the SIM.

Since one iteration of the SIM corresponds to an actual time step, whereas it is a complete simulation of the glacier evolution over some time frame in the SOA, we cannot compare their convergence rates. Therefore, we want to compare the convergence rates of the TIM to those of the SOA. The TIM applied to steady geometries consists in setting $s_i = s_f = s^*$ and choose the same time frame as in the SOA. Here, we list the values obtained with the TIM with fixed-point parameter $\beta = 0.1$, because the errors are smaller in this case and the convergence rates are not very different from those with larger β. Furthermore, we set the regularization

$\varepsilon = 100$ to make it fast. Table 3 compares the results obtained with the SOA to those with the TIM. As before, we have sought the regularization parameter α that makes the SOA converge towards a bedrock topography whose discrepancy with the solution bedrock is the same as with the TIM. Again, the computed final surface s_f^c with the estimated bedrock topography is closer to the exact one if we apply the SOA. In Table 3, the number of iterations of the two methods can be compared, since, this time, every iteration consists in simulating the glacier over the chosen time frame. We notice that both the number of iterations and the computational time are much smaller with the SOA. Figure 4 superimposes the convergence rates of the two methods and evinces the faster rate of the SOA.

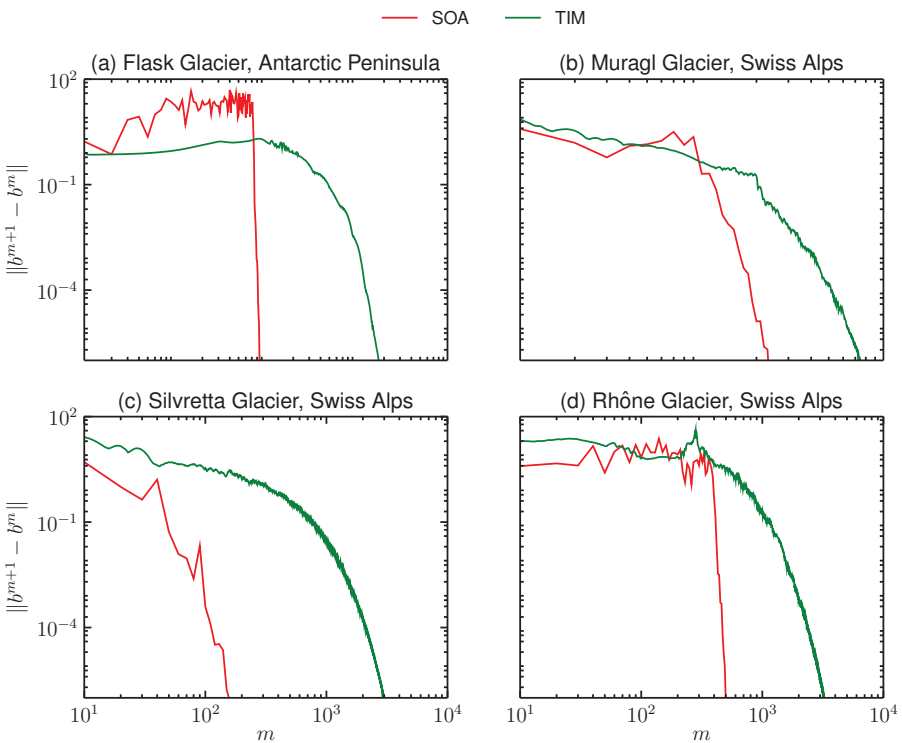

Figure 4. Comparison of the convergence rates of the TIM and the SOA in the case of the steady geometries shown in Figure 2, for parameters that lead to about the same error in the estimation of the bedrock topography in both cases (see first two columns of Table 3), when the time frame is 1 a. The TIM regularization and fixed-point parameters were set to $\varepsilon = 100$ and $\beta = 0.1$. The applied SOA regularizations are listed in Table 3. Convergence is faster with SOA than with TIM.

5.2 Transient geometries

We now want to compare the efficiency of the TIM to that of the SOA in the case of transient geometries. To make this comparison possible, the same time frame must be chosen in both methods and, as above, the regularization parameter α of the SOA must be selected in such a way that the discrepancy $\|b^c - b^*\|$ is about the same in both cases. Tables 4–6 report the discrepancies between the computed and solution bedrock topographies as well as between the computed and exact final surfaces, the number of iterations, and the computational time needed to converge for time frames of 1 a, 5 a, and 10 a when the glaciers' boundaries are fixed.

		$\|b^c - b^*\|$ [m]		$\|s_f^c - s_f\|$ [m]	
Figure 3	α	TIM	SOA	TIM	SOA
(a)	2.502e−01	2.726e+00	2.726e+00	5.920e−01	6.379e−03
(b)	6.032e−01	2.262e+00	2.262e+00	6.524e−01	9.772e−02
(c)	5.506e+00	6.951e+01	6.951e+01	8.587e−01	6.064e−01
(d)	5.549e−01	4.060e+00	4.060e+00	6.493e−01	1.021e−01

		n_{its}		T_{CPU} [s]	
Figure 3	α	TIM	SOA	TIM	SOA
(a)	2.502e−01	17,493	846	1.651e+04	1.146e+03
(b)	6.032e−01	925	159	2.190e+02	6.628e+01
(c)	5.506e+00	950	177	2.040e+02	2.874e+01
(d)	5.549e−01	2,500	334	9.263e+02	2.008e+02

Table 4. Discrepancy $\|b^c - b^*\|$ between the computed and the solution bedrock topographies b^c and b^*, discrepancy $\|s_f^c - s_f\|$ between the computed and the solution final surface geometries s_f^c and s_f, number of iterations, and computational time needed to converge in both the TIM and SOA applied to a fixed-boundary, transient geometry, when the time frame is 1 a. In the TIM, we set the regularization and fixed-point parameters to $\varepsilon = 100$ and $\beta = 0.1$ respectively. As in the previous results, we have chosen the regularization parameters α in such a way that the discrepancy $\|b^c - b^*\|$ is the same in both TIM and SOA. As in the steady case, the quality of the estimation with SOA is better than with TIM.

		$\|b^c - b^*\|$ [m]		$\|s_f^c - s_f\|$ [m]	
Figure 3	α	TIM	SOA	TIM	SOA
(a)	6.502e−01	2.765e+00	2.765e+00	5.996e−01	1.029e−02
(b)	4.248e+00	2.331e+00	2.331e+00	6.589e−01	3.107e−01
(c)	1.000e+02	1.420e+01	1.410e+01	3.024e+00	2.860e+00
(d)	2.462e−01	2.930e+00	2.772e+00	6.245e−01	4.483e−02

		n_{its}		T_{CPU} [s]	
Figure 3	α	TIM	SOA	TIM	SOA
(a)	6.502e−01	17,309	420	6.996e+04	2.972e+03
(b)	4.248e+00	853	80	7.076e+02	1.334e+02
(c)	1.000e+02	611	93	3.774e+02	7.212e+01
(d)	2.462e−01	2,459	449	3.458e+03	1.142e+03

Table 5. Same comparison as in Table 4 but over a time frame of 5 a.

The results are essentially the same as in the steady case. The resulting final surface discrepancy is smaller with the SOA than the TIM. The high values of α applied to geometry of Figure 2 (c) are due to the fact that a Gaussian filtering was applied to the bedrock topography after each TIM iteration, allowing for convergence but also increasing the error in the estimation. To get these errors with the SOA, a large value of α was used. Because the algorithm is the same as in the steady case, the behavior of the convergence rate is the same.

Figures 5(a) and (b) depict the estimations in the case of retreating glaciers, while Figures 5(c) and (d) those of growing glaciers. Technically, the estimation for expanding glaciers is very easy to do. However, the retreating case may be unstable and harder to deal with. This is because of the negativity of the surface mass-balance at locations where the ice thickness is very small or zero. In this case, much care has to be taken about bound (3.15) so that accurate estimations can be carried out. Table 7 reports the errors, numbers of iterations and computational times done in both the TIM and SOA. The same conclusions as before can be reached, i.e. the estimated topography is of higher quality with SOA than with TIM, with this set of parameters.

		$\|b^c - b^*\|$ [m]		$\|s_f^c - s_f\|$ [m]	
Figure 3	α	TIM	SOA	TIM	SOA
(a)	7.054e−01	2.795e+00	2.795e+00	6.067e−01	1.128e−02
(b)	3.489e+00	2.183e+00	2.183e+00	6.237e−01	2.406e−01
(c)	1.563e+01	5.646e+00	5.646e+00	9.010e−01	4.149e−01
(d)	2.691e−01	2.836e+00	2.836e+00	6.029e−01	4.742e−02

		n_{its}		T_{CPU} [s]	
Figure 3	α	TIM	SOA	TIM	SOA
(a)	7.054e−01	17,286	381	1.483e+05	5.369e+03
(b)	3.489e+00	797	82	1.242e+03	2.163e+02
(c)	1.563e+01	374	64	3.969e+02	1.035e+02
(d)	2.691e−01	2,444	401	6.540e+03	1.854e+03

Table 6. Same comparison as in Tables 4 and 5 but over a time frame of 10 a.

5.3 Sensitivity analysis

We present a sensitivity analysis of our method and restrict ourselves to perturbations of the form $a\chi(x)$, where the perturbation's amplitude $a > 0$ is of the order of magnitude of typical measurement errors and $\chi(x)$ is a uniformly distributed random variable on the interval $[-1, 1]$. We first add $a\chi(x)$ to the surface topography and then to the surface mass-balance. When $a = 1$ m, Figure 6 shows how sensitive the estimation is to the small perturbation of the initial and final surfaces and how the increase of the regularization parameter affects it. None of the estimations converged. Instead, they oscillated around the solution. However, the smoothness of the estimated topography is improved with increasing α, even though the estimation still differs from the solution by more than 1 m. To make the algorithm converge, one has to smooth the perturbed data. On the other hand, the SOA is not sensitive to the mass-balance, just like the SIM and TIM.

Finally, we show how susceptible the method is to changes in the diffusion parameter Γ (see equation (2.2)). Of particular interest is the sensitivity of the method to variations in the rate factor A that is related to Γ according to equation (2.3). Throughout this paper, we used the value $A = 0.1\,\text{bar}^{-3}\text{a}^{-1}$ and now aim at illus-

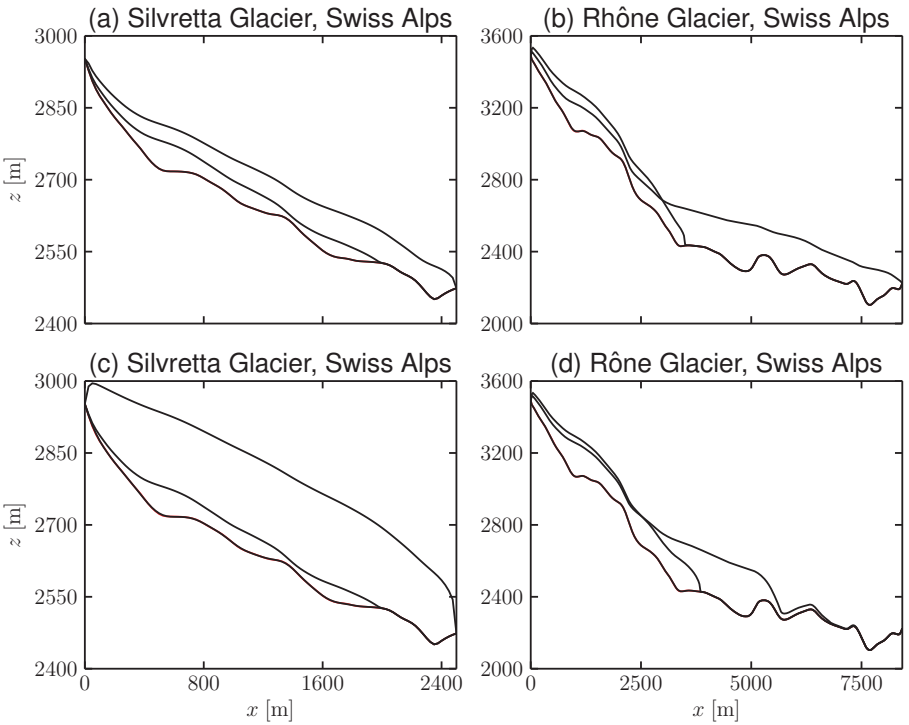

Figure 5. Bedrock topography estimation in the transient case when the glaciers are retreating ((a) and (b)) or expanding ((c) and (d)). The thick continuous and dashed lines represent the initial and final surfaces respectively, while the thick red and thin black lines show the solution and estimated subglacial topographies. For these simulations, we set $\Delta x = 25\,\mathrm{m}$, $\Delta t = 10^{-3}\,\mathrm{a}$, and $\delta = 10^{-6}\,\mathrm{m}$. The time frames are different: (a) $t_f - t_i = 15\,\mathrm{a}$, (b) $t_f - t_i = 5\,\mathrm{a}$, (c) $t_f - t_i = 5\,\mathrm{a}$, (d) $t_f - t_i = 10\,\mathrm{a}$. Each time, the mass-balance was chosen in such a way that a large retreat or advance could be generated. As before, the estimation is very close to the solution topography. Corresponding errors, number of iterations, and computational times are listed in Table 7.

trating the consequences of an uncertainty on this parameter, which is very likely in practical applications. Figure 7 evinces the obtained estimations when the solution bedrock topography is ruled by a rate factor amounting to $A = 0.1\,\mathrm{bar}^{-3}\mathrm{a}^{-1}$ and we try to estimate the bedrock with a different value, ranging from $A = 0.07$ to $0.13\,\mathrm{bar}^{-3}\mathrm{a}^{-1}$. In the SIA, the precise value of this factor does not seem to be important. The influence is largest on the geometry depicted by Figure 7 (a), due to the large thicknesses involved or small surface slope.

		$\|b^c - b^*\|$ [m]		$\|s_f^c - s^*\|$ [m]	
Figure 5	α	TIM	SOA	TIM	SOA
(a)	5.716e−01	2.042e−01	2.042e−01	3.196e−02	1.858e−02
(b)	5.000e+00	6.629e+00	1.718e+00	1.355e+01	1.335e−01
(c)	1.565e+00	1.805e+00	1.805e+00	4.157e−01	1.001e−01
(d)	9.409e−01	1.672e+00	1.672e+00	5.931e−01	5.732e−02

		n_{its}		T_{CPU}	
Figure 5	α	TIM	SOA	TIM	SOA
(a)	5.716e−01	2,548	49	3.574e+03	1.630e+02
(b)	5.000e+00	732	107	1.154e+03	3.530e+02
(c)	1.565e+00	1,468	85	8.220e+02	7.000e+01
(d)	9.409e−01	1,146	178	4.085e+03	1.022e+03

Table 7. Discrepancies $\|b^c - b^*\|$ and $\|s_f^c - s_f\|$ between the computed and solution bedrock topographies b^c and b^* as well as between the computed and solution final surface geometries s_f^c and s_f, number of iterations, and computational times needed to converge in both the TIM and SOA applied to a moving-boundary, transient geometry, over various time frames (listed in the caption of Figure 5). In the TIM, we set the regularization and fixed-point parameters to $\varepsilon = 100$ and $\beta = 0.1$ respectively. The regularization parameter α is chosen in such a way that the discrepancy $\|b^c - b^*\|$ is the same in both the TIM and SOA. The quality of the estimation with SOA is better than with TIM.

6 Conclusion

Existing SOA that aim at reconstructing glaciers' subglacial topographies are usually based on the minimization of the surface velocity mismatch and only deal with steady or quasi-stationary geometries. The recently introduced TIM, which is a particular case of SOA where the costate variables are set to zero, is an exception, since it minimizes the surface topography misfit in both steady and transient situations with a simple fixed-point procedure. Described as unreliable in the shallow water framework [15], the SOA applied to the shallow ice equation appears to be accurate, reliable, and fast. In particular, even if the initial guess for the bedrock

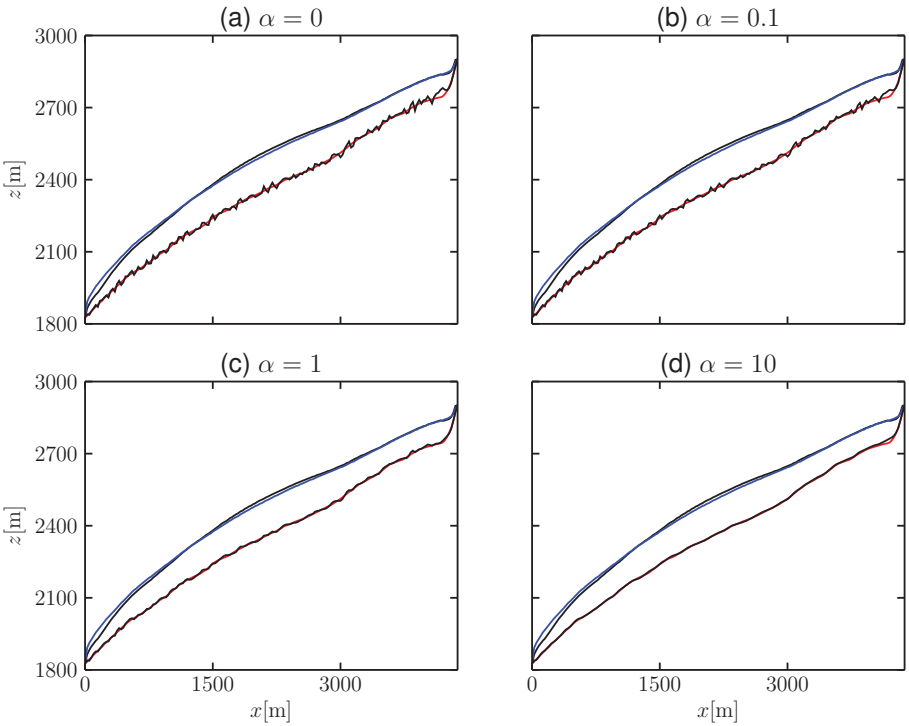

Figure 6. Sensitivity to the surface of Muragl glacier, on a transient, fixed-boundary problem over a time frame of 1 a. We applied the perturbation with $a = 1$ m to both the initial and final surfaces, represented by the thick black and thin blue lines respectively. The perturbation can hardly be noticed on the two surface geometries. The solution and estimated bedrock topographies are delineated by the thick red and thin black lines. Without smoothing of the perturbed data, the algorithm does not converge, but oscillates around the displayed estimation. With larger regularization parameters, the estimation becomes smoother and closer to the solution topography.

is far from the solution, the resulting subglacial estimation is close to the solution bedrock on all the tested geometries, even when moving boundaries are taken into account. The method is as sensitive to surface data as TIM.

The SOA not only deals with transient evolutions but is also faster and more accurate than the SIM and TIM, as far as our comparisons can tell. As the latter procedures, the SOA needs neither filtering of the surface topography with a lower slope limit nor further assumptions on the model itself. Moreover, in this case, it remains simple as it only consists in the minimization of the misfit between the computed and exact (or measured) surface topographies at the end of the glaciers'

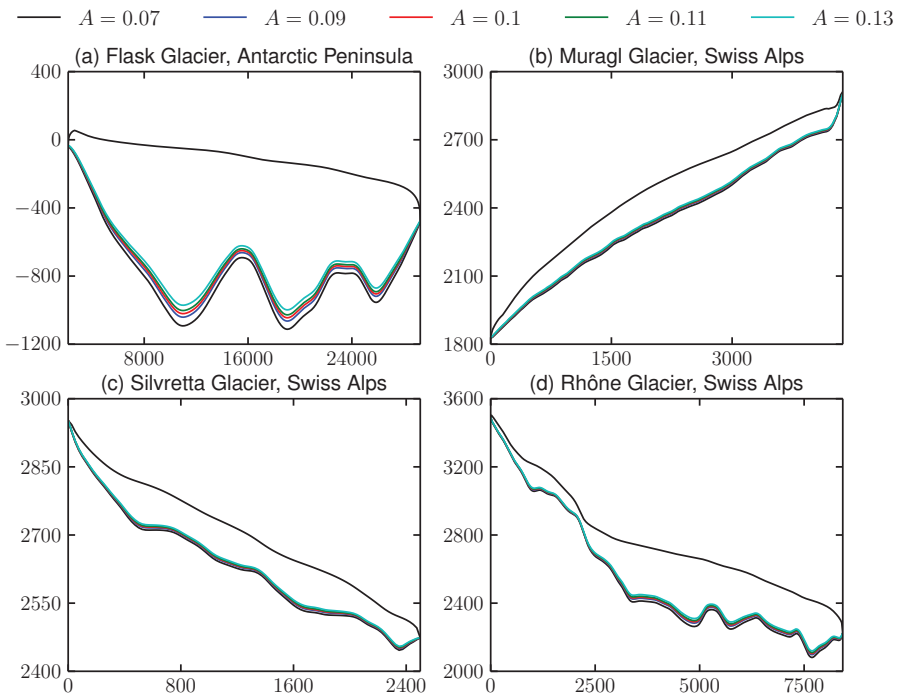

Figure 7. Sensitivity to the rate factor A on fixed-boundary problems. Comparison of the subglacial estimation with $A = 0.1\,\mathrm{bar}^{-3}\mathrm{a}^{-1}$ to estimations where A ranges from 0.07 to $0.13\,\mathrm{bar}^{-3}\mathrm{a}^{-1}$ in the transient case over a time frame of 1 a. The thick black and red lines delineate the initial surface s_i and the solution bedrock topographies respectively, corresponding to $A = 0.1\,\mathrm{bar}^{-3}\mathrm{a}^{-1}$. No large discrepancy with the solution bedrock can be noted, except in the extreme case (a) where the ice thickness is the largest and surface slope is the smallest.

time evolution and the PDE-constraint is cast into one single evolution equation. The applied regularization that minimizes the gradient of the bedrock topography is well known among glaciologists. A large quantity of minimization algorithms have been efficiently implemented to solve this type of PDE-constrained optimization problems [22, 36].

Convergence of the TIM is still an open question in three space dimensions. Whatever the model order is, the Laplacian regularization does not always smooth the topography sufficiently so that an additional Gaussian filtering may have to be applied after each iteration. Such a filtering usually prevents the three-dimensional TIM from converging or being accurate. Moreover, since it assumes the locality of bedrock to surface perturbation transfers, the method may not work well in all

situations. The SOA in turn seems to be a method that circumvents these issues. It is also extensible to three space dimensions and higher order models, at the expense of making the problem's formulation more complex. Furthermore, sliding can be easily embodied.

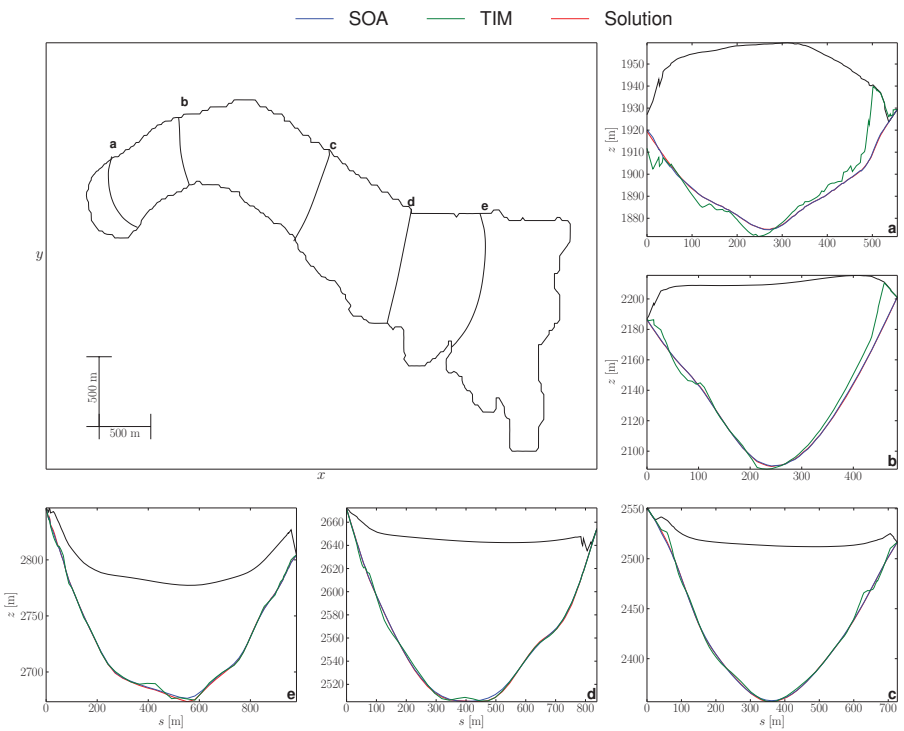

Figure 8. Application to the geometry of Muragl glacier, Swiss Alps, of the extended SOA and TIM to three space dimensions. The grid spacing, time step, and tolerance were set to $\Delta x = 25\,\mathrm{m}$, $\Delta t = 10^{-3}$ a, and $\delta = 10^{-6}$ m. SOA and TIM parameters were set to $\alpha = 1$, $\beta = 0.1$, and $\varepsilon = 1$. The thick black and red lines represent the final surface and the solution bedrock topographies respectively, while the thin blue and green lines delineate the bedrock topographies obtained with the SOA and TIM respectively. In the subplots (a) – (e), the abscissa s and ordinate z represent the coordinate following the profile and the altitude of the depicted topographies respectively. On each of the shown profiles, the error committed is small. The accuracy of the SOA is higher than that of TIM, especially at the glacier's tongue (profile (a)).

To conclude, let us show a preliminary result on the three-dimensional geometry of Muragl glacier, in the Swiss Alps (see Figure 8). It comes from the straightforward extension of the algorithm presented in this paper to three-space dimensions,

where the constraint reads

$$\begin{cases} \dfrac{\partial \mathcal{H}}{\partial t} = \nabla \cdot \left(\mathcal{H}^{n+2} \|\nabla(b+\mathcal{H})\|^{n-1} \nabla(b+\mathcal{H}) \right) + \mathcal{B} & \text{in } \Omega_\perp \times [t_i, t_f], \\ \mathcal{H} = s_i - b & \text{in } \Omega_\perp \times \{t_i\}, \end{cases} \tag{6.1}$$

where $\Omega_\perp$ is the glacier map domain (i.e. the whole (x, y)-plane shown in the contour plot depicted by Figure 8) and ∇ the (x, y)-gradient operator. We can follow the very same procedure as that exposed in this contribution in the three-dimensional case described by equation (6.1). Its application to Muragl glacier results in the profiles shown in Figure 8 (a)–(e) which are transversal lines across the glacier. For the sake of comparison, we also include the estimation obtained with the TIM. The SOA regularization parameter was set to $\alpha = 1$, while the TIM parameters to $\varepsilon = 1$ and $\beta = 0.1$. Again, on each of these profiles, the difference between the solution and reconstructed bedrock topographies is very small, of the order of 10 m with the SOA. The accuracy of the SOA is higher than that of the TIM, especially at the glacier's tongue. In fact, the TIM struggles converging on that part of the glacier, whereas the SOA convergence is straight and fast, which makes it very interesting in terms of computational times. The detailed extension of the two-dimensional method to three dimensions will be the subject of a forthcoming publication.

A Equations of the semi-implicit scheme

Define

$$\mathcal{G}^\ell_{j-\frac{1}{2}} = \left(\frac{\mathcal{H}^\ell_j + \mathcal{H}^\ell_{j-1}}{\Delta x} \right)^{n+2} \left| \frac{s^\ell_j - s^\ell_{j-1}}{\Delta x} \right|^{n-2} \operatorname{sign}(s^\ell_j - s^\ell_{j-1}).$$

In the semi-implicit scheme, we have

$$\mathcal{T}_j = \frac{1}{\Delta x} \left[\mathcal{D}^\ell_{j+\frac{1}{2}} \frac{s^{\ell+1}_{j+1} - s^{\ell+1}_j}{\Delta x} - \mathcal{D}^\ell_{j-\frac{1}{2}} \frac{s^{\ell+1}_j - s^{\ell+1}_{j-1}}{\Delta x} \right] + \mathcal{B}_j.$$

The dual problem is discretized with

$$\begin{aligned} \tilde{\mathcal{T}}^{\ell,\ell+1}_j = {} & \frac{1}{\Delta x} \left[\mathcal{D}^{\ell-1}_{j+\frac{1}{2}} \frac{\lambda^\ell_{j+1} - \lambda^\ell_j}{\Delta x} - \mathcal{D}^{\ell-1}_{j-\frac{1}{2}} \frac{\lambda^\ell_j - \lambda^\ell_{j-1}}{\Delta x} \right] \\ & + \frac{n-1}{\Delta x} \left[\mathcal{G}^\ell_{j+\frac{1}{2}} \frac{s^{\ell+1}_{j+1} - s^{\ell+1}_j}{\Delta x} \frac{\lambda^{\ell+1}_{j+1} - \lambda^{\ell+1}_j}{\Delta x} \right. \\ & \left. - \mathcal{G}^\ell_{j-\frac{1}{2}} \frac{s^{\ell+1}_j - s^{\ell+1}_{j-1}}{\Delta x} \frac{\lambda^{\ell+1}_j - \lambda^{\ell+1}_{j-1}}{\Delta x} \right] \end{aligned}$$

$$- \frac{n+2}{2}\left[\mathcal{F}^{\ell}_{j+\frac{1}{2}} \frac{s^{\ell+1}_{j+1} - s^{\ell+1}_{j}}{\Delta x} \frac{\lambda^{\ell+1}_{j+1} - \lambda^{\ell+1}_{j}}{\Delta x} + \mathcal{F}^{\ell}_{j-\frac{1}{2}} \frac{s^{\ell+1}_{j} - s^{\ell+1}_{j-1}}{\Delta x} \frac{\lambda^{\ell+1}_{j} - \lambda^{\ell+1}_{j-1}}{\Delta x}\right],$$

with final and initial conditions

$$\lambda^{M+1}_j - \frac{\Delta t}{\Delta x}\left(\mathcal{D}^{M}_{j+\frac{1}{2}} \frac{\lambda^{M+1}_{j+1} - \lambda^{M+1}_{j}}{\Delta x} - \mathcal{D}^{M}_{j-\frac{1}{2}} \frac{\lambda^{M+1}_{j} - \lambda^{M+1}_{j-1}}{\Delta x}\right) = s_f(\xi_j) - b_j - \mathcal{H}^{M+1}_j,$$

and

$$\lambda^0_j = \lambda^1_j - (n+2)\frac{\Delta t}{2}\left(\mathcal{F}^{0}_{j+\frac{1}{2}} \frac{s^{1}_{j+1} - s^{1}_{j}}{\Delta x} \frac{\lambda^{1}_{j+1} - \lambda^{1}_{j}}{\Delta x} + \mathcal{F}^{0}_{j-\frac{1}{2}} \frac{s^{1}_{j} - s^{1}_{j-1}}{\Delta x} \frac{\lambda^{1}_{j} - \lambda^{1}_{j-1}}{\Delta x}\right) + (n-1)\frac{\Delta t}{\Delta x}\left(\mathcal{G}^{0}_{j+\frac{1}{2}} \frac{s^{1}_{j+1} - s^{1}_{j}}{\Delta x} \frac{\lambda^{1}_{j+1} - \lambda^{1}_{j}}{\Delta x} - \mathcal{G}^{0}_{j-\frac{1}{2}} \frac{s^{1}_{j} - s^{1}_{j-1}}{\Delta x} \frac{\lambda^{1}_{j} - \lambda^{1}_{j-1}}{\Delta x}\right).$$

Finally, the gradient of the objective function is

$$\frac{\partial J_h}{\partial b_j} = -(n-1)\frac{\Delta t}{\Delta x}\sum_{\ell=0}^{M}\left(\mathcal{G}^{\ell}_{j+\frac{1}{2}} \frac{s^{\ell+1}_{j+1} - s^{\ell+1}_{j}}{\Delta x} \frac{\lambda^{\ell+1}_{j+1} - \lambda^{\ell+1}_{j}}{\Delta x} - \mathcal{G}^{\ell}_{j-\frac{1}{2}} \frac{s^{\ell+1}_{j} - s^{\ell+1}_{j-1}}{\Delta x} \frac{\lambda^{\ell+1}_{j} - \lambda^{\ell+1}_{j-1}}{\Delta x}\right) - \frac{\Delta t}{\Delta x}\sum_{\ell=0}^{M}\left(\mathcal{D}^{\ell}_{j+\frac{1}{2}} \frac{\lambda^{\ell+1}_{j+1} - \lambda^{\ell+1}_{j}}{\Delta x} - \mathcal{D}^{\ell}_{j-\frac{1}{2}} \frac{\lambda^{\ell+1}_{j} - \lambda^{\ell+1}_{j-1}}{\Delta x}\right) + b_j + \mathcal{H}^{M+1}_j - s_f(\xi_j) + \lambda^0_j + \frac{\partial \mathcal{R}_h}{\partial b_j}.$$

Acknowledgments. We would like to thank Gil Michel for English corrections as well as Jacques Rappaz and Alexandre Caboussat for fruitful discussions on the topic. We thank two anonymous reviewers for their helpful comments.

References

[1] G. Adalgeirsdottir, Flow dynamics of Vatnajökull ice cap, Iceland, Mitteilung 181, Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie der ETH Zürich, 2003.

[2] R. J. Arthern and G. H. Gudmundsson, Initialization of ice-sheet forecasts viewed as an inverse Robin problem, *Journal of Glaciology* **56** (2010), 527–533.

[3] S. Avdonin, V. Kozlov, D. Maxwell and M. Truffer, Iterative methods for solving a nonlinear boundary inverse problem in glaciology, *Journal of Inverse and Ill-Posed Problems* **17** (2009), 239–258.

[4] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik et al., B. F. Smith and H. Zhang, PETSc users manual, Report, ANL-95/11 – Revision 3.3, Argonne National Laboratory, 2012.

[5] S. Balay, J. Brown, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley et al., PETSc web page, 2011, `http://www.mcs.anl.gov/petsc`.

[6] R. Becker, H. Kapp and R. Rannacher, Adaptive finite element methods for optimal control of partial differential equations: Basic concept, *SIAM Journal on Control and Optimization* **39** (2000), 113–132.

[7] C. G. Broyden, The convergence of a class of double-rank minimization algorithms. I: General considerations, *Journal of Applied Mathematics* **6** (1970), 76–90.

[8] E. Bueler, Numerical approximation of a two-dimensional thermomechanical model for ice flow, Technical Report 02-02, University of Alaska, Fairbanks, 2002.

[9] E. Bueler, C. S. Lingle, J. A. Kallen-Brown, D. N. Covey and L. N. Bowman, Exact solutions and verification of numerical models for isothermal ice sheets, *Journal of Glaciology* **51** (2005), 291–306.

[10] H. Chen, C. Miao and X. Lv, A three-dimensional numerical internal tidal model involving adjoint method, *International Journal for Numerical Methods in Fluids* **69** (2012), 1584–1613.

[11] G. K. C. Clarke, F. S. Anslow, A. H. Jarosch, V. Radic, B. Menounos, T. Bolch and E. Berthier, Ice volume and subglacial topography for western Canadian glaciers from mass balance fields, thinning rates, and a bed stress model, *Journal of Climate* **26** (2013), 4282–4303.

[12] D. Farinotti, H. Corr and G. H. Gudmundsson, The ice thickness distribution of Flask Glacier, Antarctic Peninsula, determined by combining radio-echo soundings, surface velocity data, and flow modelling, *Annals of Glaciology* **54** (2012), 18–24.

[13] D. Farinotti, M. Huss, A. Bauder, M. Funk and M. Truffer, A method to estimate the ice volume and ice-thickness distribution of alpine glaciers, *Journal of Glaciology* **55** (2009), 422–430.

[14] R. Fletcher, A new approach to variable metric algorithms, *The Computer Journal* **13** (1970), 317–322.

[15] A. F. Gessese, M. Sellier, E. Van Houten and G. Smart, Reconstruction of river bed topography from free surface data using a direct numerical approach in one-dimensional shallow water flow, *Inverse Problems* **27** (2011), 025001.

[16] J. W. Glen, The flow law of ice, *IUGG/IAHS Symposium of Chamonix IAHS Publication* **47** (1958), 171–183.

[17] D. Goldfarb, A family of variable-metric methods derived by variational means, *Mathematics of Computation* **24** (1970), 23–26.

[18] R. Greve and H. Blatter, *Dynamics of Ice Sheets and Glaciers*, Springer-Verlag, Berlin, 2009.

[19] M. Habermann, D. Maxwell and M. Truffer, Reconstruction of basal properties in ice sheets using iterative inverse methods, *Journal of Glaciology* **58** (2012), 795–807.

[20] W. Haeberli and M. Hoelzle, Application of inventory data for estimating characteristics of and regional climate-change effects on mountain glaciers: A pilot study with the European Alps, *Annals of Glaciology* **21** (1995), 206–212.

[21] C. Heining, Velocity field reconstruction in gravity-driven flow over unknown topography, *Physics of Fluids* **23** (2011), 032101.

[22] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda et al., An overview of the Trilinos project, *ACM Transactions on Mathematical Software* **31** (2005), 397–423.

[23] M. F. Hutchinson, A new procedure for gridding elevation and stream line data with automatic removal of spurious pits, *Journal of Hydrology* **106** (1989), 211–232.

[24] G. Jouvet and E. Bueler, Steady, shallow ice sheets as obstacle problems: Well-posedness and finite element approximation, *SIAM Journal on Applied Mathematics* **72** (2012), 1292–1314.

[25] G. Jouvet, E. Bueler, C. Graeser and R. Kornhuber, A nonsmooth Newton multigrid method for a hybrid, shallow model of marine ice sheets, in: *Recent Advances in Scientific Computing and Applications*, Contemporary Mathematics 586, American Mathematical Society, Providence (2013), 197–205.

[26] G. Jouvet and C. Graeser, An adaptive Newton multigrid method for a model of marine ice sheets, *Journal of Computational Physics* **252** (2013), 419–437.

[27] A. Khazendar, E. Rignot and E. Larour, Larsen B Ice Shelf rheology preceding its disintegration inferred by a control method, *Geophysical Research Letters* **34** (2007), L19503.

[28] H. W. Kuhn and A. W. Tucker, Nonlinear programming, in: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability* (Berkeley 1950), University of California Press, Berkeley (1951), 481–492.

[29] E. Larour, E. Rignot, I. Joughin and D. Aubry, Rheology of the Ronne Ice Shelf, Antarctica, inferred from satellite radar interferometry data using an inverse control method, *Geophysical Research Letters* **32** (2005).

[30] H. L. Li, F. Ng, Z. Q. Li, D. H. Qin and G. D. Cheng, An extended "perfect-plasticity" method for estimating ice thickness along the flow line of mountain glaciers, *Journal of Geophysical Research* **117** (2012), F01020.

[31] M. W. Mahaffy, 3-dimensional numerical model of ice sheets – tests on barnes ice cap, *Journal of Geophysical Research – Oceans and Atmospheres* **81** (1976), 1059–1066.

[32] D. Maxwell, M. Truffer, S. Avdonin and M. Stuefer, An iterative scheme for determining glacier velocities and stresses, *Journal of Glaciology* **54** (2008), 888–898.

[33] R. W. McNabb, R. Hock, S. O'Neel, L. A. Rasmussen, Y. Ahn, M. Braun et al., Using surface velocities to calculate ice thickness and bed topography: A case study at Columbia Glacier, Alaska, USA, *Journal of Glaciology* **58** (2012), 1151–1164.

[34] L. Michel, M. Picasso, D. Farinotti, A. Bauder, M. Funk and H. Blatter, Estimating the ice thickness of mountain glaciers with an inverse approach using surface topography and mass-balance, *Inverse Problems* **29** (2013), 035002.

[35] M. Morlighem, E. Rignot, H. Seroussi, E. Larour, H. Ben Dhia and D. Aubry, A mass conservation approach for mapping glacier ice thickness, *Geophysical Research Letters* **38** (2011), L19503.

[36] T. Munson, J. Sarich, S. Wild, S. Benson and L. C. McInnes, TAO 2.0 users manual, Report, Mathematics and Computer Science Division, Argonne National Laboratory, 2012, `http://www.mcs.anl.gov/tao`.

[37] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., Springer Series in Operations Research and Financial Engineering, Springer-Verlag, New York, 2006.

[38] F. Paul and A. Linsbauer, Modeling of glacier bed topography from glacier outlines, central branch lines, and a DEM, *International Journal of Geographical Information Science* **26** (2012), 1173–1190.

[39] N. Petra, H. Zhu, G. Stadler, T. J. R. Hughes and O. Ghattas, An inexact Gauss–Newton method for inversion of basal sliding and rheology parameters in a nonlinear Stokes ice sheet model, *Journal of Glaciology* **58** (2012), 889–903.

[40] C. Rothenbuehler, Erfassung und Darstellung der Geomorphologie im Gebiet Bernina (GR) mit Hilfe von GIS, *Physische Geographie* **41** (2003), 117–126.

[41] D. F. Shanno, Conditioning of quasi-Newton methods for function minimization, *Mathematics of Computation* **24** (1970), 647–656.

[42] J. Sokolowski and J.-P. Zolesio, *Introduction to Shape Optimization. Shape Sensitivity Analysis*, Springer Series in Computational Mathematics, Springer-Verlag, 1992.

[43] A. Tikhonov, Solution of incorrectly formulated problems and the regularization method, *Soviet Mathematics Doklady* **4** (1963), 1035–1038.

[44] A. Vieli and A. J. Payne, Application of control methods for modelling the flow of Pine Island Glacier, West Antarctica, *Annals of Glaciology* **36** (2003), 197–204.

Received February 22, 2013.

Author information

Laurent Michel, EPFL SB MATHICSE, Station 8, 1015 Lausanne, Switzerland.
E-mail: laurent.michel@epfl.ch

Marco Picasso, EPFL SB MATHICSE, Station 8, 1015 Lausanne, Switzerland.
E-mail: marco.picasso@epfl.ch

Daniel Farinotti, German Research Centre for Geosciences (GFZ),
Telegrafenberg, 14473 Potsdam, Germany.
E-mail: farinotti@vaw.baug.ethz.ch

Andreas Bauder, VAW, ETH Zürich, 8093 Zürich, Switzerland.
E-mail: bauder@vaw.baug.ethz.ch

Martin Funk, VAW, ETH Zürich, 8093 Zürich, Switzerland.
E-mail: funk@vaw.baug.ethz.ch

Heinz Blatter, Institute for Atmospheric and Climate Science, ETH Zürich,
Universitätstrasse 16, 8092 Zürich, Switzerland.
E-mail: heinz.blatter@env.ethz.ch