Home Assignment : Numerical 1D Resolution of an Ice Flow Problem

2017

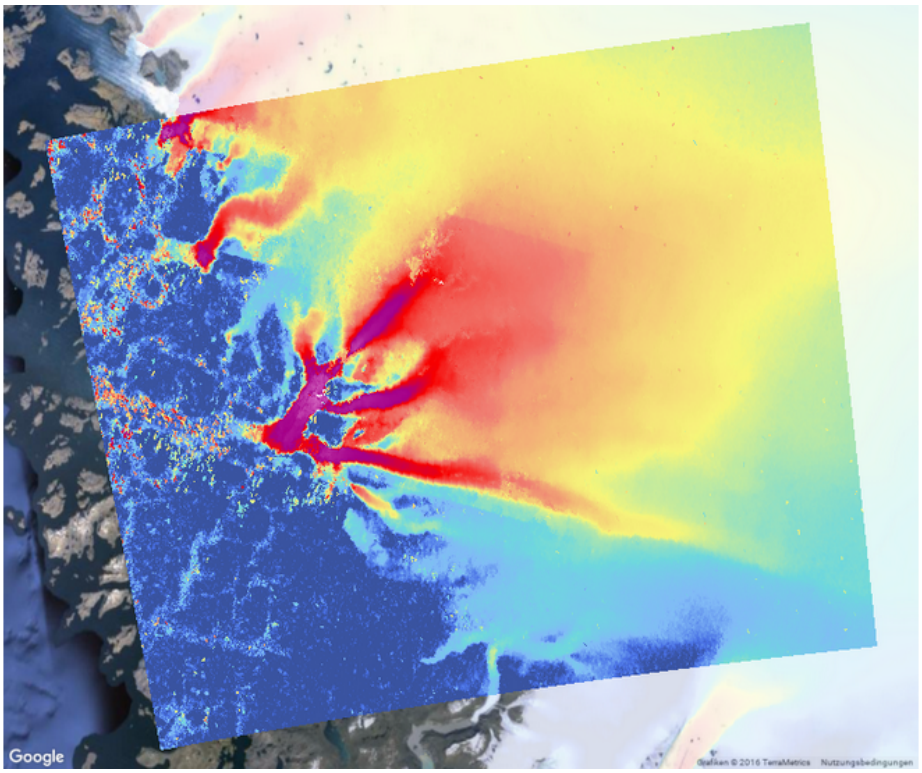

Figure 1: Satellite measurement of the ice velocity in the Uppernavik Isstroem from Sentinel-1 SAR data acquired from 2014-10 to 2016-06. Available at http://products.esa-icesheets-cci.org/

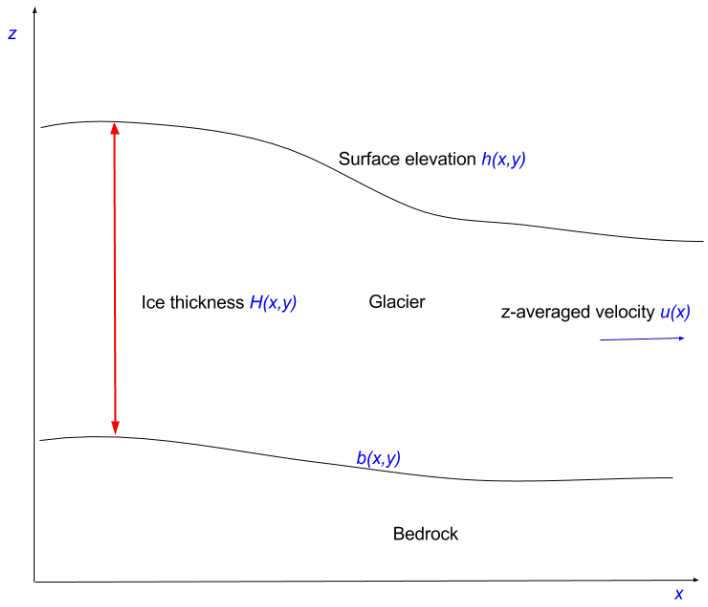

Figure 2: Cross section of a glacier

Motivation

One of the major challenges mankind will have to face in the forthcoming centuries, is the sea level rise. The sea level rise is due to mainly two factors : the water dilatation of the oceans due to the increase of the temperature, and the melting of the glaciers, and the ice sheets at the northern and southern poles. It is then of primal importance for knowing of how many meters the sea level will rise to be able to predict how the ice behave on the pole.

Glaciers and ice sheets can be interpreted as super viscous flows, and many models to describe them have been derived from the classical equations of fluid dynamics. Among them is the Shallow Shelf Approximation (SSA).

The SSA model is particularly well suited for high velocity streams of ice, near the coast, where the ice velocity at the surface can reach up to 5 km/year. This model considers that the ice thickness is really small compared to the domain's size (hence the term "shallow") and that the shear stress is more or less constant along the ice thickness.

Variables of interest are showed in the figure 2

The SSA model, in 1D is given by two equations : one linear transport equation, and a non-linear diffusion-reaction equation. The idea of the proposed work is, step by step, to solve these equations.

This project is divided in four parts: the two first parts can be treated

separately, to treat the third part, one will need the codes produced in the second, and to do the last part, the codes from the first and third parts will be required.

1 A transport problem.

1.1 Theoretical prerequisites.

In a first time we will focus on the equation modelling the transport of the ice thickness H by the velocity u, and where a source term is added (It models the effect of the wind on the ice thickness, the accumulation due to the snow, or the melting of the ice, ...).

$$\partial_t H = -\partial_x(u\ H) + M \tag{1}$$

a) • *Time discretization* - We want to apply the Euler explicit method :

$$\frac{H^{n+1} - H^n}{\Delta t} = -\partial_x(u^n H^n) + M \tag{2}$$

• *Spatial discretization* - Using a constant discretization ($\Delta x = cte$), write the discretization in space, using a centered scheme. Show that

$$H_i^{n+1} = H_i^n - \frac{\Delta t}{2\Delta x}(u_{i+1}^n H_{i+1}^n - u_{i-1}^n H_{i-1}^n) + \Delta t\, M_i \tag{3}$$

b) Let us assume that u is constant, and positive. Is there a condition on Δt and Δx that must be satisfied to ensure the stability of the scheme ? (One can use the results of the exercise, or the course).

c) Under the same assumptions ($u = cte > 0$), which simple spatial scheme would have been stable under a certain CFL condition ? (results of the course, exercises can be used).

d) An other technique to ensure stability is to add a small diffusion term in the equation. This corresponds to an uncentering of the scheme.

The equation that one wants to solve now is :

$$\partial_t H = -\partial_x(u\ H) + \epsilon\partial_{x^2}(u\ H) + M \tag{4}$$

Bonus Show that with u constant and $\epsilon = \frac{\Delta x}{2}$, this is exactly the uncentered scheme (backward).

e) Show that, using explicit Euler in time, centered scheme for both spatial first and second derivative, one has :

$$H_i^{n+1} = a\ H_{i-1}^n + b\ H_i^n + c\ H_{i+1}^n + \Delta t\, M_i \tag{5}$$

where a, b, c have to be explicited in function of u_i, ϵ, Δt, Δx . Write this equation in term of matrix and vectors.

f)
- *Dirichlet Boundary condition at $x = x_{min}$* - Having a Dirichlet boundary condition there means that

$$H_1^{n+1} = H_{dir} \tag{6}$$

Modify the first line of the matrix accordingly.

- *Neumann Boundary condition at $x = x_{max}$* - To create a homogeneous Neumann Boundary condition, for an explicit time scheme, one can consider a so called "ghost point", of index $N + 1$. Then numerical BC that has to be satified is then

$$\frac{H_{N+1}^n - H_N^n}{\Delta x} = 0 \tag{7}$$

Injecting this new equation in equation (5), show that the boundary condition that one has to implement is

$$H_N^{n+1} = a^\star H_{N-1}^n + b^\star H_N^n + \Delta t M_N$$

where $a^\star, b^\star$ have to be explicited.
Modify the last line of the matrix accordingly.

1.2 Numerical implementation.

a) Implement the code in Matlab/SciLab, using the skeletton script that is provided.

- In the file `transport_build_matrix.m` , implement the matrix whose coefficients are described in eq.(5).
- In the file `transport_apply_BC.m` modify the matrix, and vectors such that the boundary conditions are respected.
- In the file `solve_transport_time_step.m` solve eq.(5).

b) Describe what happens in the file `transport.m` .

2 A steady diffusion-reaction equation.

2.1 Theoretical prerequisites

In the SSA model, the velocity satisfies the equation :

$$\partial_x(\mu\, \partial_x u) - B\, u - f = 0 \tag{8}$$

In which μ is the viscosity of the ice sheet, averaged on the ice thickness, B is depending on properties of the bedrock on which the ice is lying, and f is a function of the ice thickness of the surface slopes. In this part, it is assumed that μ, B and f are known functions of x.

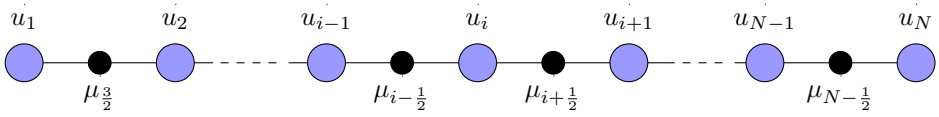

Figure 3: Scheme of the staggered grid

a) *Spatial discretization* - To solve this problem, one needs to introduce a so-called staggered grid for the parameter μ .

- Show, by applying two Taylor expansions of u at $x = x_i + \frac{\Delta x}{2}$ that the approximation (9) is of order 2.

$$\partial_x u(x_i + \frac{\Delta x}{2}) \approx \frac{u_{i+1} - u_i}{\Delta x} \tag{9}$$

- Show, using a similar technique that

$$\partial_x(\mu\, \partial(u)) = \frac{\mu_{i+\frac{1}{2}} \frac{u_{i+1}-u_i}{\Delta x} - \mu_{i-\frac{1}{2}} \frac{u_i - u_{i-1}}{\Delta x}}{\Delta x} + o(\Delta x) \tag{10}$$

b) *Linear problem formulation :* The discrete problem that one wants to solve is then :

$$\frac{\mu_{i+\frac{1}{2}}(u_{i+1} - u_i) - \mu_{i-\frac{1}{2}}(u_i - u_{i-1})}{\Delta x^2} - B\, u_i = f_i \tag{11}$$

- To have a Dirichlet BC at $x = x_{min}$, one must force $u_1 = u_{dir}$, with u_{dir} a prescribed velocity.
- To have a homogeneous Newton BC at $x = x_{max}$, one must force $u_N - u_{N-1} = 0$.
- write this problem with a matrix $A(\mu, B, \Delta x)$ and vectors F and U.

2.2 Numerical implementation

a) Implement the code to solve this equation. One may want to use the provided templates.

b) To check whether the code is valid or not, the following procedure will be done.

- Show that if $u(x) = \sin(\pi x)$, $\mu(x) = 1 + x$, $B(x) = x$, then to have equation (8) satisfied, f should be

$$f(x) = -\left[\pi^2(1 + x) + x\right] \cos(\pi x) - \pi\ \sin(\pi x) \tag{12}$$

- Compute the error between $u_{th} = \sin(\pi x)$ and $u_{computed}$ (the output of your program), using the function `norm` of Matlab.
- Compute the error for different Δx ($\Delta x = [0.1,\ 0.01,\ 0.001,\ 0.0001]$), and describe how it evolves with respect to Δx (plotting can help). Could you have expected this behaviour ?

3 A non-linear steady diffusion-reaction equation.

Warning : *To do this section, you need the codes from the previous section !*

In this section, one considers the equation

$$\partial_x(\mu(u)\,\partial_x u) - B(u)\,u - f = 0 \tag{13}$$

The viscosity μ is now depending on the velocity u (The fluid is non-newtonian), and the coefficient B that is due to ice/bedrock interactions is now also depending on the velocity. These term are defined as follows :

$$\mu(u) = \mu_0\ |\partial_x u|^p$$
$$B(u) = B_0\ |u|^q$$

To solve this non-linear problem, several approaches exist ; Among them are the Newton method, fixed-point algorithm. Here, the non-linear problem will be solved with a fixed-point algorithm (Newton's algorithm is left as a bonus).

3.1 Fixed-point algorithm.

Principle of the fixed point iteration. One need s to have a continuous function g such that $g(u) = u$. Then, starting from an initial guess u_0, one iterates $g(u_n) = u_{n+1}$ until one reaches a convergence criterion.

Usually a convergence criterion that one can take is

$$\delta = \| u_n - n_{n-1} \| < tolerance \tag{14}$$

Fixed point iteration applied to Eq (13). In our case, one will just solve the equation, with $\mu(u)$ and $B(u)$ computed with the velocity of the previous iteration *i.e.*

$$\partial_x(\mu(u^n)\,\partial_x u^{n+1}) - B(u^n)\,u^{n+1} - f = 0 \tag{15}$$

a) Without developping $\mu(u^n)$ and $B(u^n)$, discretize the problem using the same methods as in the previous part (use Eq:(11). Express the problem in terms of matrix and vectors.

b) Show that after discretisation, the problem can be rewritten in the following form :

$$U^{n+1} = G(U^n) \tag{16}$$

where U is the vector containing the u_i, and $G(U)$ has to be explicited.

Relaxation. In practice, one needs to add a so-called relaxation to have convergence of this method. This is done by doing the following steps :

$$\begin{aligned} U^* &= F(U^n) && \text{Compute an intermediate velocity} \\ U^{n+1} &= \theta U^* + (1-\theta)U^n && \text{Compute the new velocity} \end{aligned}$$

where θ has typically values of the order of the percent ($\theta \approx 0.05$).

3.2 Numerical implementation.

a) Implement a function `compute_reac_coeff` in which one computes B.

b) Implement a function `compute_diff_coeff` in which one computes μ (still on a staggered grid, cf. previous part, eq.(9)).

c) Implement the function `NL_get_velocity` in which one applies the fixed point algorithm, with relaxation. One will have to use the function implemented in the previous part.

d) Make θ vary and describe what happens if it is very small ($\approx 10^{-3}$) / large (≈ 0.7)

e) Show that if $u(x) = cos(\pi x)$, $p = q = 2$, $B_0 = \mu = 1$, then to have eq(13) satisfied, one needs to have

$$f = -3\pi^4\, cos(\pi x)\, sin(\pi x)^2 - cos(\pi x)^3 \tag{17}$$

f) Check if your algorithm is correct by using the above theoretical solution.

4 The SSA system of equation.

You have developped, in the previous part, all the tools that are required to solve the SSA system of equation that we recall here :

$$\begin{cases} \partial_t H = -\,\partial_x(H\,u) \\ \partial_x(\mu_0 H|\partial_x u|^p\ \partial_x u) - B_0|u|^q - \rho g H \partial_x h = 0 \end{cases} \tag{18}$$

The problem that one wants to solve is the set of equation (18), with Dirichlet boundary conditions for both H and u at $x = x_m in$, and homogeneous Neumann boundary conditions at $x = x_m ax$, for both H and u.

The method that will be used to solve this problem is called a splitting method, because each equation will be solved one after one other, and not both at the same time.

In the first section, one saw how to solve the first equation, if one has the velocity, and in the third part, one almost saw how to solve the second equation if one has the ice thickness H.

4.1 Treatment of the gravity driven term.

The only part of the second equation that has not been implemented yet is the gravity driven part $\rho g H \partial_x h$, where h is the surface elevation (cf. figure 2). The relation between h and H is naturally $h = b + H$, where b is the bedrock elevation.

Note that the diffusion term is now $\partial_x(\mu_0 H|\partial_x u|^p\, \partial_x u)$ and not $\partial_x(\mu_0|\partial_x u|^p\, \partial_x u)$ as in the non-linear case.

- Implement the function `build_source_term` that computes this gravity driven term $\rho g H \partial_x h$.

  One should estimate $\partial_x h$ as follow :

  - at $x = x_1$, $\quad \partial_x h \approx \frac{h_2 - h_1}{\Delta x}$
  - at $x = x_i$, $\quad \partial_x h \approx \frac{h_{i-1} - h_{i-1}}{2\Delta x}$, $\quad \forall i, 1 < i < N$
  - at $x = x_N$, $\quad \partial_x h \approx \frac{h_N - h_{N-1}}{\Delta x}$

  Building a matrix is not necessary, but can be done.

4.2 Numerical resolution of the SSA system of equation.

a) Describe what is the script `SSA.m` doing.

b) After how many year would the flow (whose initial settings are in the `SSA.m` file) would reach a steady state ?

c) On which criteria can you say whether a steady state has been reached or not ?

d) Play with the differents parameters and enjoy !