Elmer/Ice Beginner course 2023
CSC, Espoo, Finland

Argentière Glacier application

Olivier GAGLIARDINI (1), Adrien GILBERT (1), Fabien Gillet-Chaulet(1), Léo CLAUZEL (1)

(1) University Grenoble Alpes / CNRS - IGE - Grenoble – France

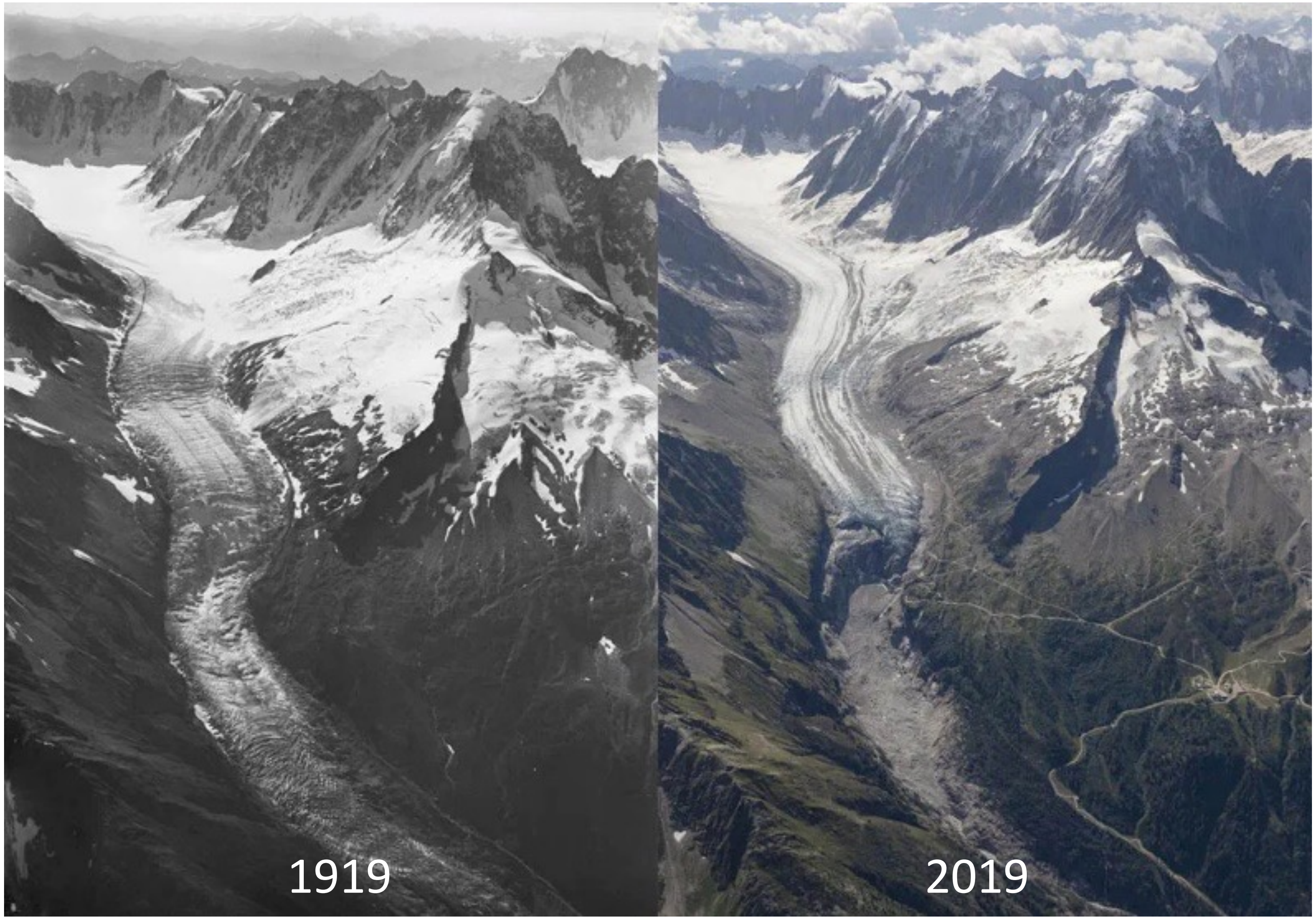
Argentière Glacier

- A bit of context – references
- Make the mesh
- Run diagnostic simulations
- Dating of ice
- Run prognostic simulations

Context - location

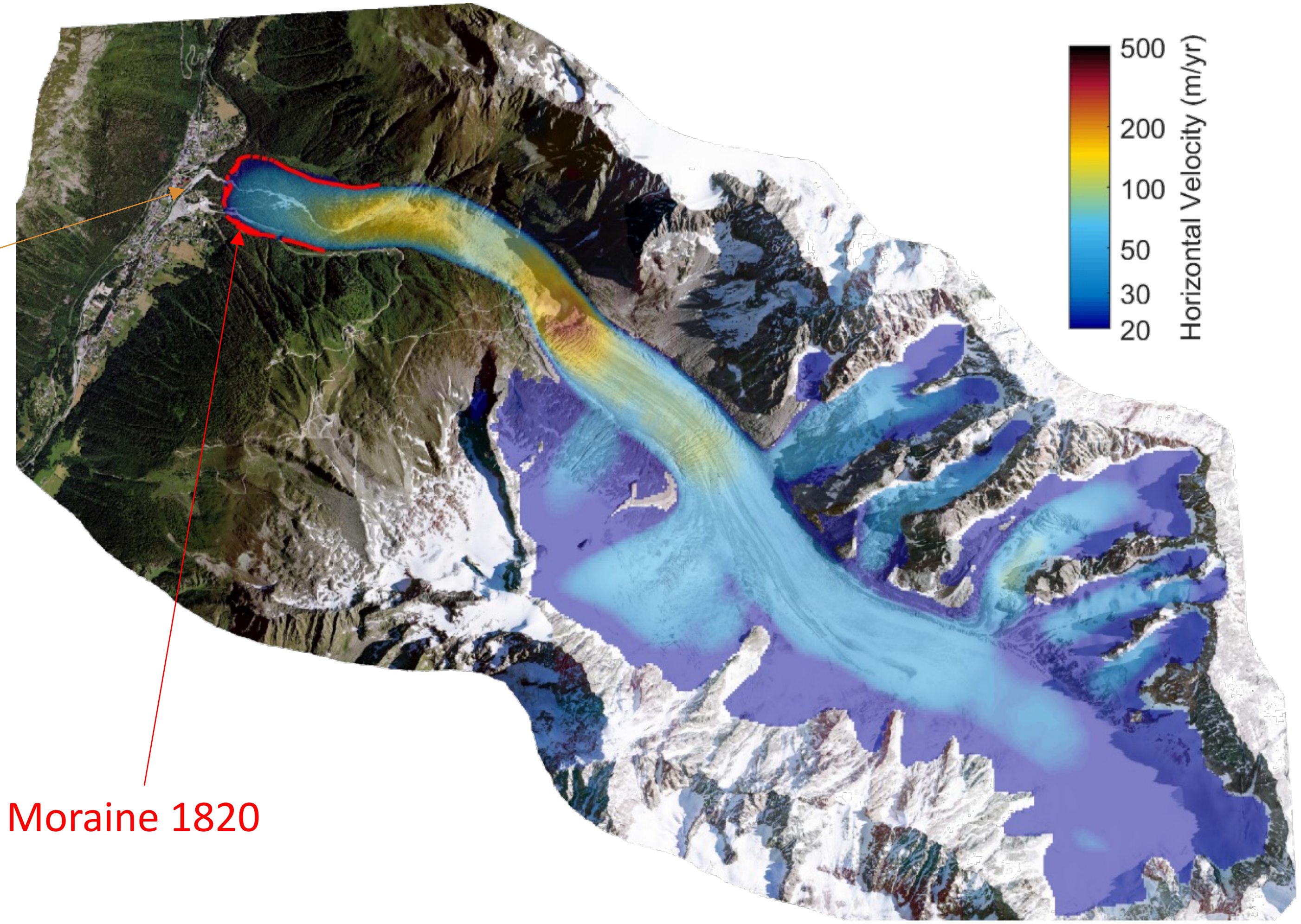
Context – Past evolution

Glacier d'Argentière, France



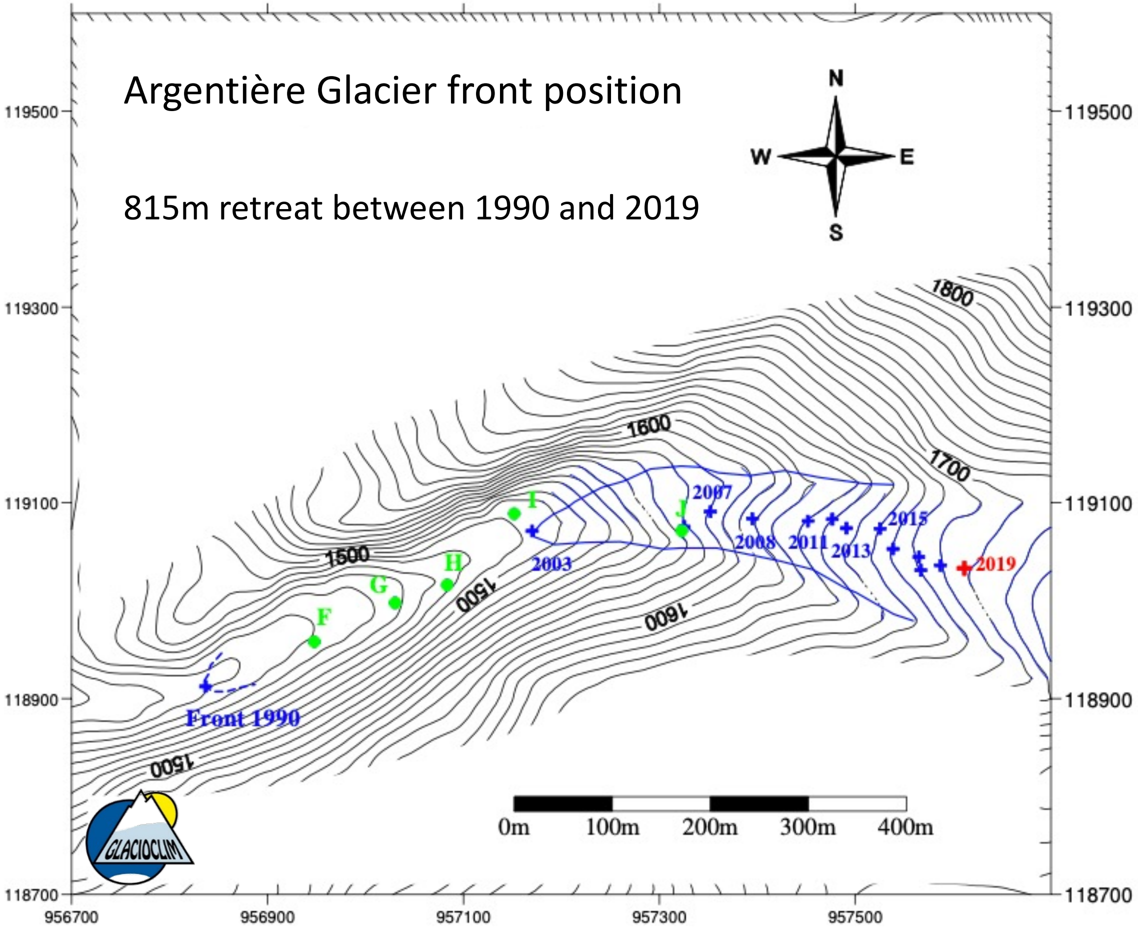
Context – Past evolution

Reconstruction of the Little Ice Age geometry (1820)

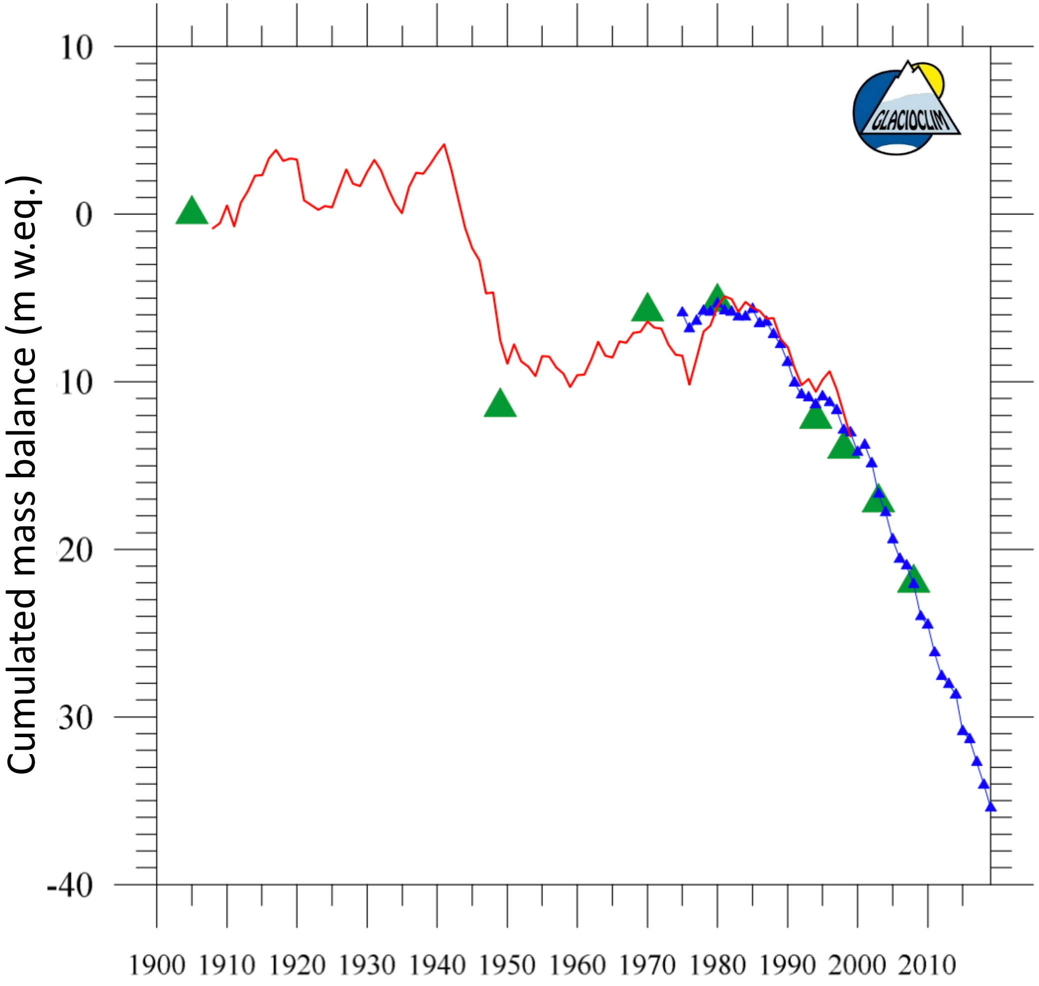


[Clauzel et al., 2023]

Context – Observed retreat

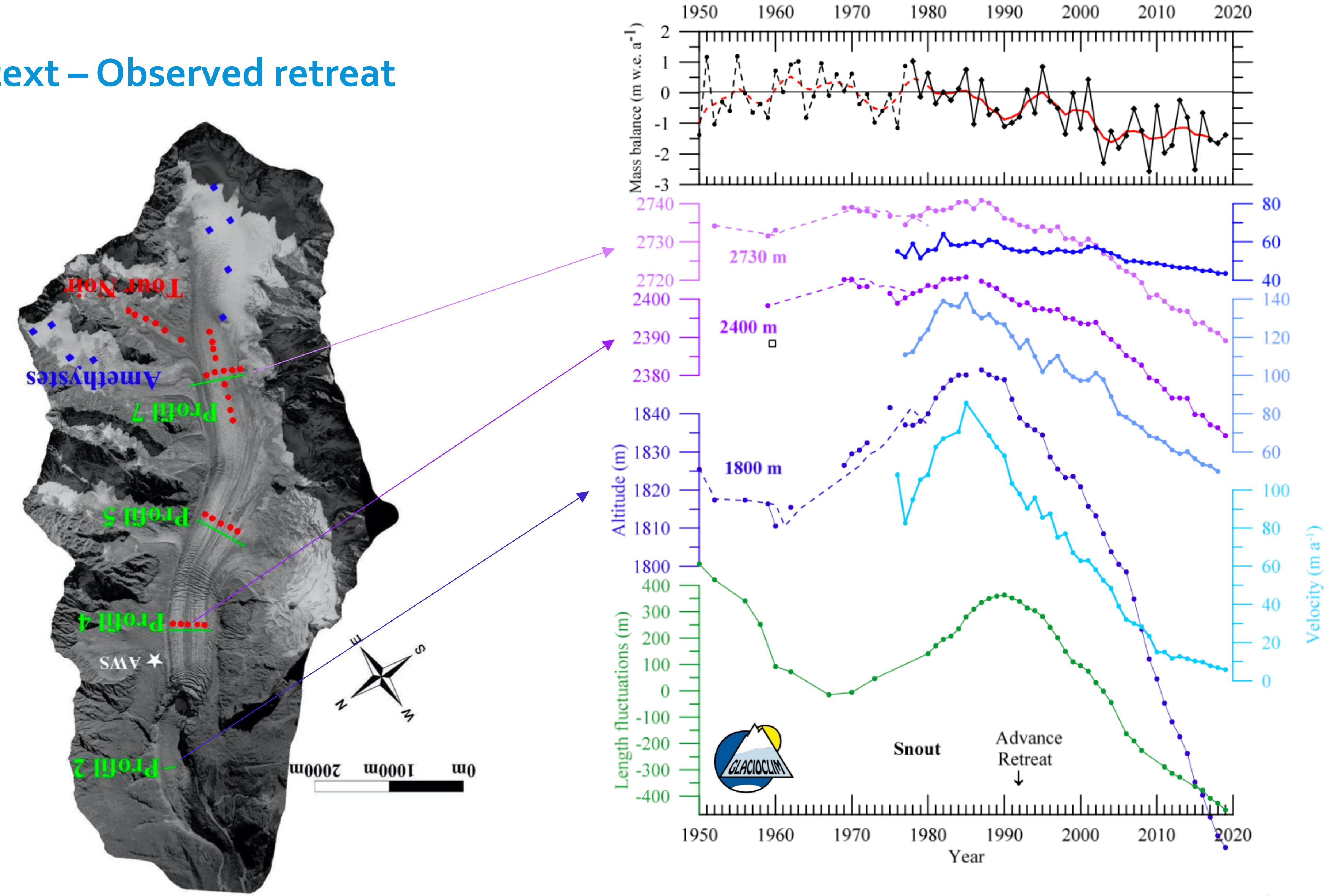


[GLACIOCLIM., 2018]



[GLACIOCLIM., 2018]

Context – Observed retreat

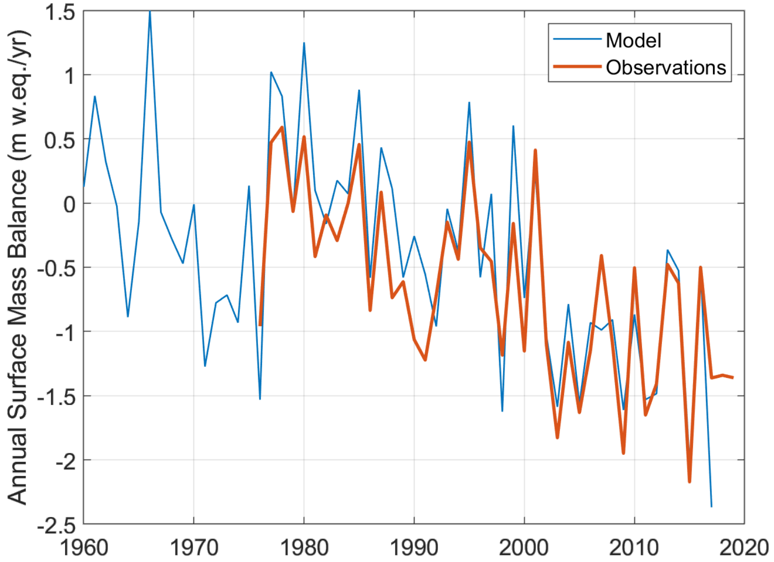


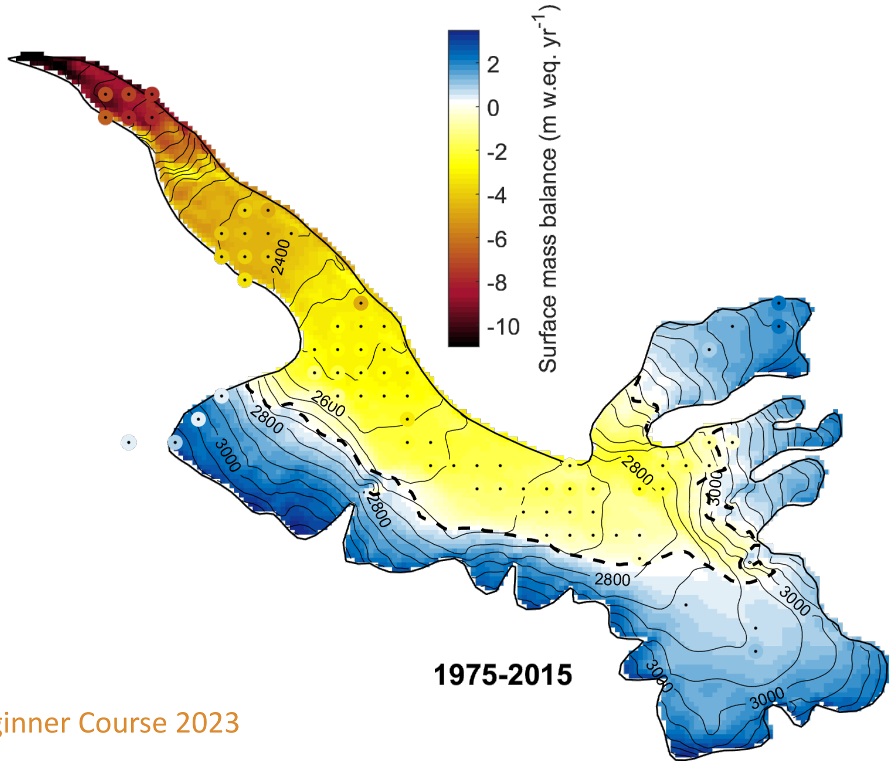
[GLACIOCLIM., 2018]

Context – Data

Argentière Glacier, Mont Blanc range, France

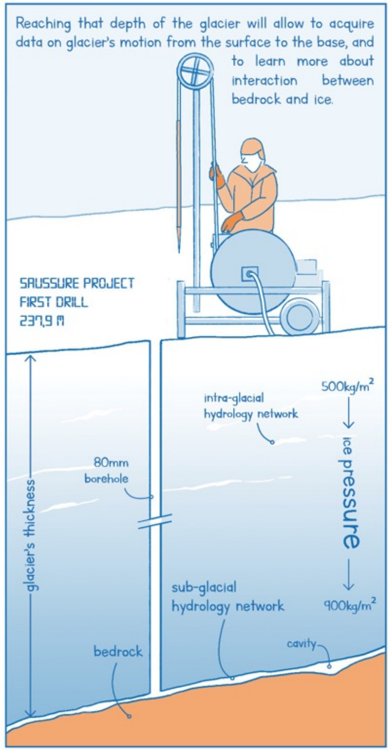
- 45 years of contineous mass balance observations (GLACIOCLIM)
- 60 years of reliable meteorological data (SAFRAN reanalizis) (Vernay et al., 2021)



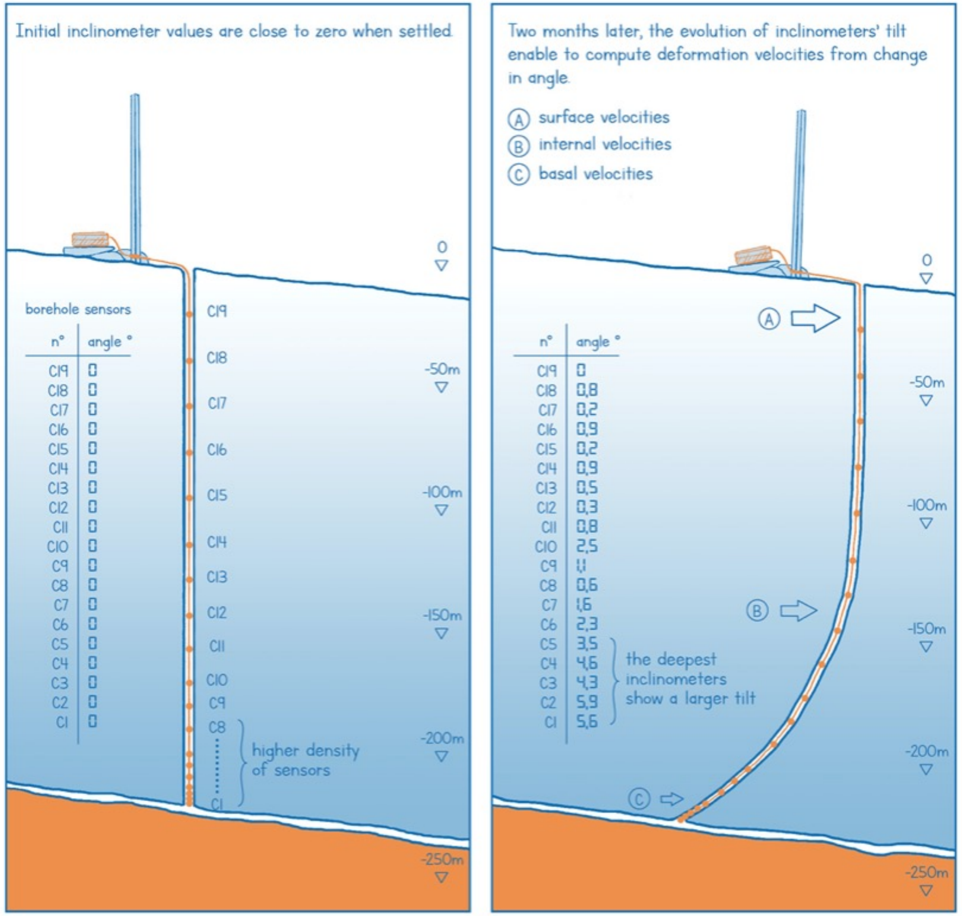


SAUSSURE
Sliding of glAciers and sUbglacial water preSSURE

SAUSSURE project



Evaluate, improve and validate various friction laws in a natural, geophysical scale configuration



https://saussure.osug.fr

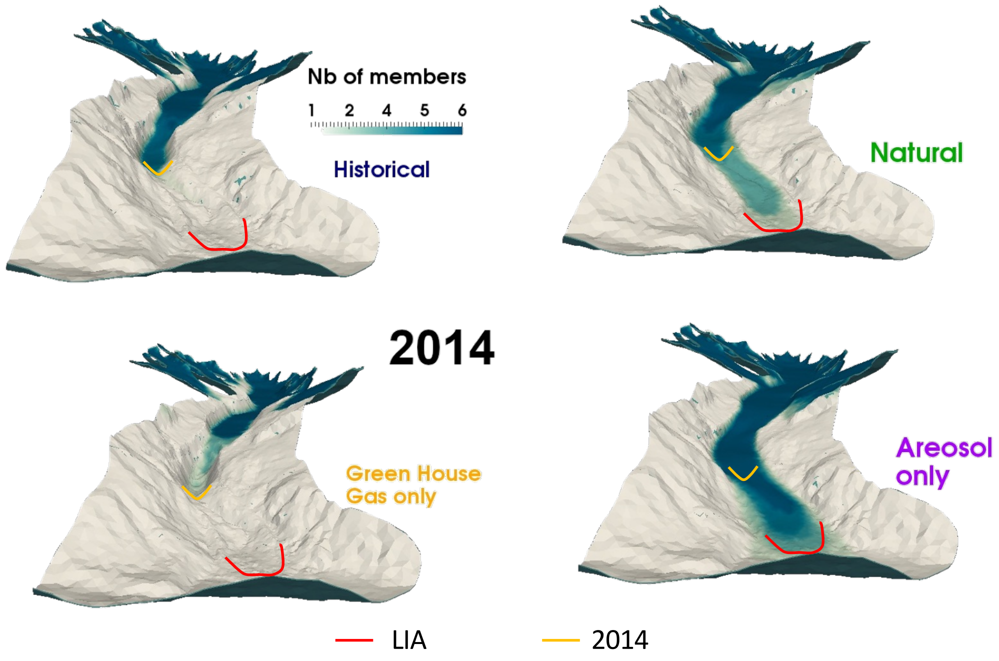
SAUSSURE – Related publications using Elmer/Ice

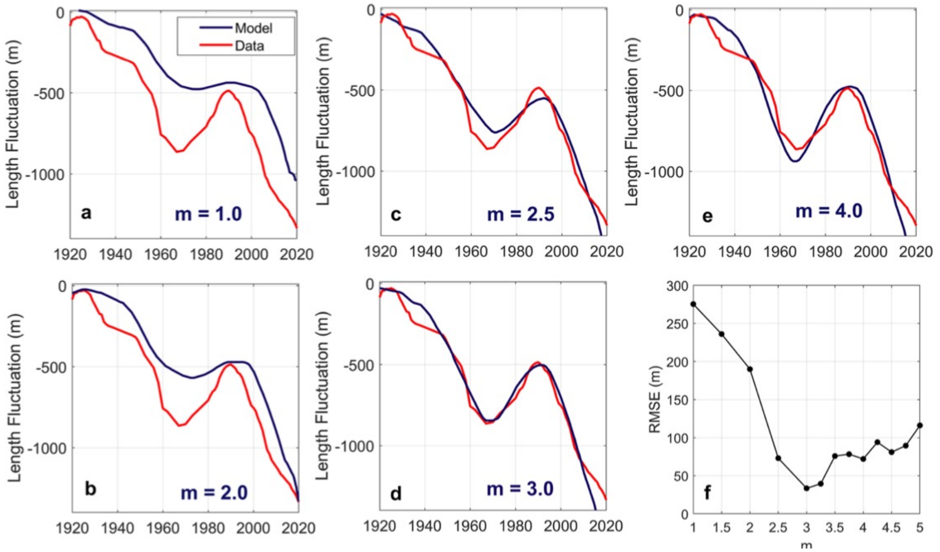
Gilbert A., F. Gimbert, K. Thøgersen, T. V. Schuler and A. Kääb, 2022. A Consistent Framework for Coupling Basal Friction With Subglacial Hydrology on Hard-Bedded Glaciers, Geophysical Research Letters, 49, e2021GL097507

Vincent C., A. Gilbert, A. Walpersdorf, F. Gimbert, O. Gagliardin, B. Jourdain, J.P. Roldan Blasco, O. Laarman, L. Piard, D. Six, L. Moreau, D. Cusicanqui and E. Thibert, 2022. *Evidence of seasonal uplift in the Argentière glacier (Mont Blanc area, France).* Journal of Geophysical Research: Earth Surface, 127, e2021JF006454

Clauzel L., M. Ménégoz, A. Gilbert, O. Gagliardini, D. Six, G. Gastineau and C. Vincent, 2023. *Sensitivity of glaciers in the European Alps to anthropogenic atmospheric forcings: Case study of the Argentière Glacier*. Geophysical Research Letters, 50, e2022GL100363

Gilbert A., O. Gagliardini, C. Vincent and F. Gimbert, 2023. Inferring the Basal Friction Law from long term changes of Glacier Length, Thickness and Velocity on an Alpine Glacier, Geophysical Research Letters 50, e2023GL104503





Proposed applications

Construct a model of Argentière Glacier

- Step 0 + 1: Mesh and Diagnostic for a given geometry
- Step 2: Add Particle solver (dating, ice trajectories)
- Step 3: Go for a Prognostic using realistic SMB

Data for ice flow modelling

- Bedrock DEM
- 1998 Surface DEM
- a contour of the glacier (2014 and large)
- Precipitation and temperature (SAFRAN 1975-2014) to run the SMB model
- Mask for the relief around the glacier (shadow) for the SMB model
- Mask for the accumulation (ad-hoc coefficient to correct from slope, avalanche,...) for the SMB model

Material

Data: *DEM_bedrock_ArgentiereLarge.dat, DEM_surface_Argentiere1998.dat, contour_2014_ARG.dat, contour_Argentiere_large.dat, safran_daily_tas.dat, safran_daily_pr.dat, MaskRelief.dat , MaskAccu.dat*

SRC: *Compute2DNodalGradient.F90, Scalar_OUTPUT_Glacier.F90, TransientMassBalance_MaskRelief.F90, README.md*

bin: *Compile.sh*

0_MakeMesh: *README.md, Contour2geo.py*

1_IMPORT_DEM: *README.md, initialise_DEM.sif*

2_ICE_AGE: *README.md, age_of_ice.sif*

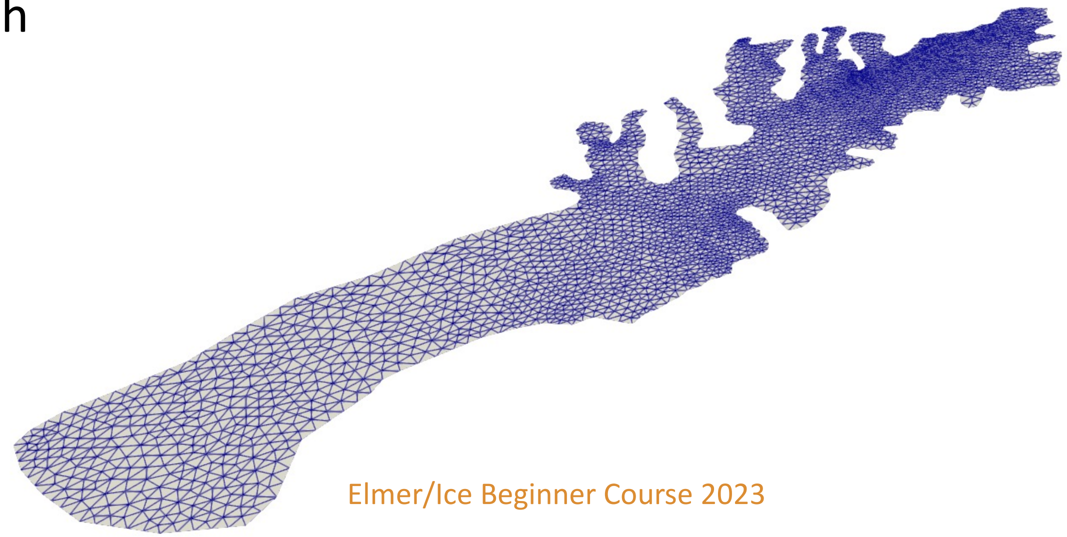
3_TRANSIENT: *README.md, transient.sif, GlacierOut (directory)*

0_MakeMsh

Objective: Create a 2d mesh (x, y) that will then be extruded in the next step within Elmer/Ice

Steps to make the mesh:

1. build the *argentiere_mesh.geo* file (input file for **GMSH**, footprint of the glacier)
2. use **GSMH** to get *argentiere_mesh.msh*
3. **ElmerGrid** to transform it into a Elmer mesh format (directory *argentiere_mesh*)
4. Still a footprint (2d) of the glacier. We will use the internal extrusion feature in Elmer to create a 3d mesh



0_MakeMsh: Makegeo.py

Create a .geo file

The python script Contour2geo.py :

- read the point coordinates in the contour file
- create the ARG_mesh.geo file (input file for GMSH)

Options :

- the contour can be made of one spline or many lines in between points
- one can choose the size of the elements around the contour

Execute:

```
> python Contour2geo.py -r 150.0 -i ../Data/contour_2014_ARG.dat -o ARG_mesh.geo
```

0_MakeMsh: GMSH

```
>gmsh -1 -2 ARG_mesh.geo -o ARG_mesh.msh
```

help: http://www.geuz.org/gmsh/

line commands:
`-2:` performs 1D and 2D mesh generation and then exit

```
serial: >ElmerGrid 14 2 ARG_mesh.msh -autoclean
parallel: >ElmerGrid 14 2 ARG_mesh.msh -autoclean -metis 4 0
```

transform a GMSH mesh (14) into an Elmer mesh (2)

```
>ElmerGrid 14 5 ARG_mesh.msh -autoclean
```

transform a GMSH mesh (14) into a vtu file (5)
use Paraview to visualize the mesh

1_IMPORT_DEM

Objective is to read the surface and bed DEM and make a first diagnostic Stokes simulation to compute the velocity

We use the internal extrusion

Define the number of vertical layers in the Simulation section:

```
Extruded Mesh Levels = Integer #Nz
```

#Nz is defined in Parameters/Physical_Parameters.IN using

```
include "../Parameters/Physical_Parameters.IN"
```

to be sure that all the different sif share the same parameters

1_IMPORT_DEM

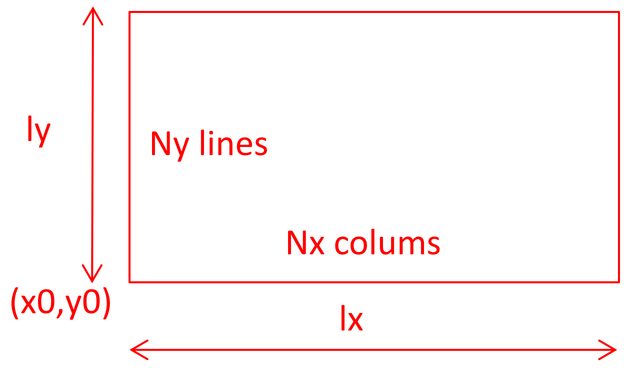
The bed and surface DEM are read using `Grid2DInterpolator`

```
! Bedrock DEM
  Variable 1 = String "bedDEM"
  Variable 1 data file = File "../Data/DEM_bedrock_ArgentiereLarge.dat"
  Variable 1 x0 = Real 949567.79d0
  Variable 1 y0 = Real 106682.32d0
  Variable 1 lx = Real 17900.0
  Variable 1 ly = Real 21640.0
  Variable 1 Nx = Integer 896
  Variable 1 Ny = Integer 1083
  Variable 1 Invert = Logical False
  Variable 1 Fill = Logical True
  Variable 1 Position Tol = Real 1.0
  Variable 1 No Data = Real -9999.0
  Variable 1 No Data Tol = Real 1.0
```

name of the DEM file

define the DEM file structure

$dx = lx / (Nx-1)$
$dy = ly / (Ny-1)$



Nx . Ny is the number of DEM points

(same for the surface DEM)

1_IMPORT_DEM

The `StructuredMeshMapper` solver allows to deform the mesh vertically according to the surface and bed DEMs

```
Solver 2
  Equation = "MapCoordinate"
  Procedure = "StructuredMeshMapper" "StructuredMeshMapper"

  Active Coordinate = Integer 3
  Mesh Velocity Variable = String "dSdt"
  Mesh Update Variable = String "dS"
  Mesh Velocity First Zero = Logical True

  Displacement Mode = Logical False
  Correct Surface = Logical True
  Minimum Height = Real #MinH
End
```

zs = min(zs, bed+MinH)

(MinH is defined in the Parameters)

1_IMPORT_DEM

The bed and surface variables BedDEM et SurfDEM have to be declared elsewhere (e.g. Stokes Solver)

```
  Exported Variable 3 = -dofs 1 "BedDEM"
  Exported Variable 4 = -dofs 1 "SurfDEM"
```

Keywords Bottom Surface and Top Surface (needed by the solver StructuredMeshMapper) are assigned the value of these two variables

```
Boundary Condition 2
  Name = "bed"
  Bottom Surface = Equals BedDEM
End


Boundary Condition 3
  Top Surface = Equals SurfDEM
End
```

Solve the Stokes equations (the minimum to know)

In the Solver section

Linear system keywords are relative to solving $A.x = b$
Both direct or iterative methods

Non linear keywords are relative to solving $A = A(x)$
Picard ($A(x_{i-1})\ x_i = b$) or Newton iterations

Navier-Stokes to Stokes: `Flow Model = Stokes`

Two available Navier-Stokes solvers in Elmer:
legacy: `Procedure File "FlowSolve" "FlowSolver"`
vectorized: `Procedure = "IncompressibleNSVec" "IncompressibleNSSolver"`

We will use the vectorized Stokes solver
See more in the sif file

Solve the Stokes equations (the minimum to know)

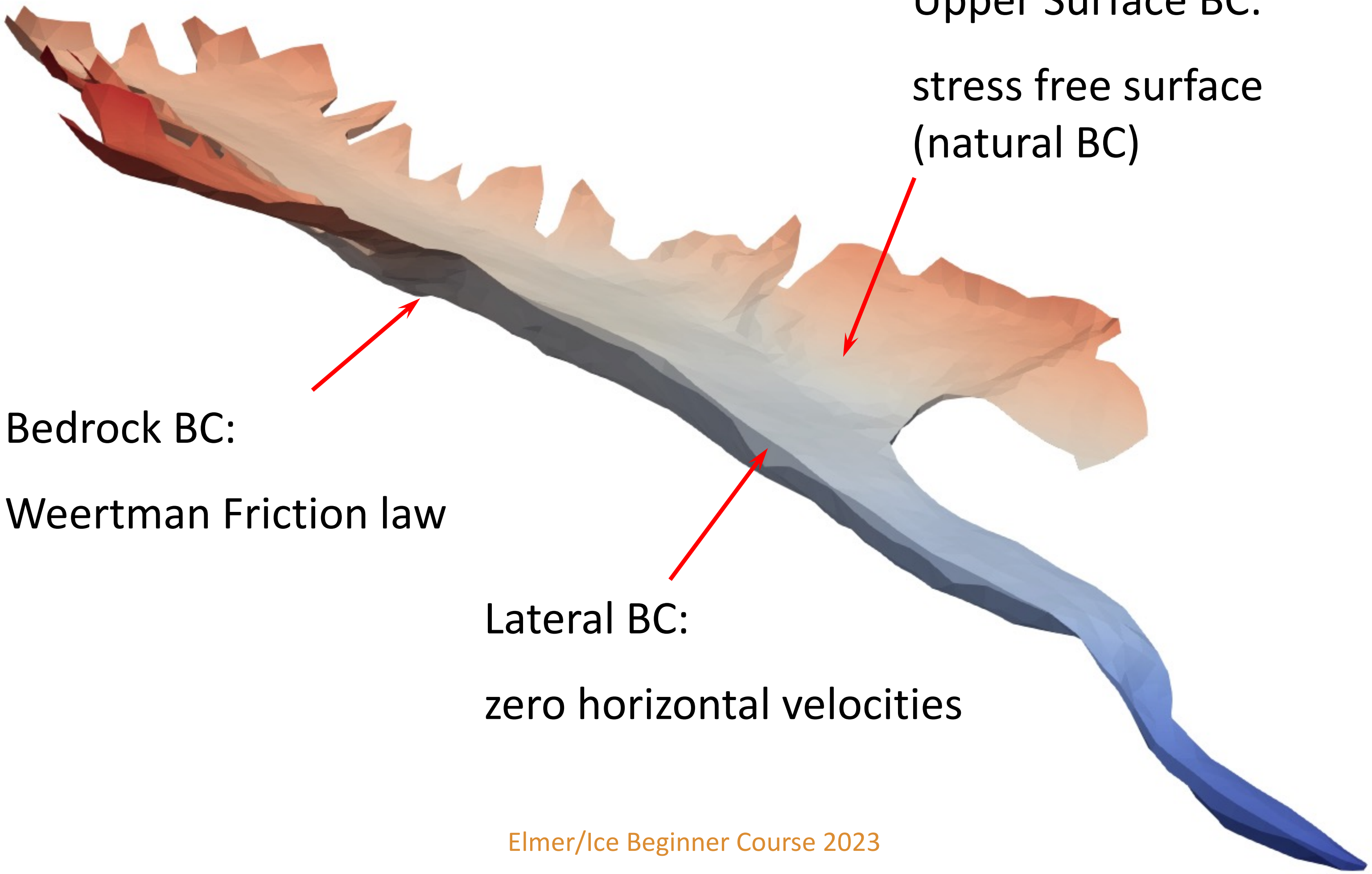
```
!Non-linear iteration settings:
  !------------------------------
  Nonlinear System Max Iterations = 50
  Nonlinear System Convergence Tolerance  = 1.0e-5
  Nonlinear System Newton After Iterations = 10
  Nonlinear System Newton After Tolerance = 1.0e-2
  Nonlinear System Reset Newton = Logical True
```

With the new vectorized Stokes solver, Newton linearisation should work.
Here we shift from Picard to Newton after 10 iterations or when the norm is lower than 10^{-2}.

Diagnostic modelling hypotheses

Solve only the Stokes equation in a diagnostic way

3 boundary conditions



Upper Surface BC:

stress free surface (natural BC)

Bedrock BC:

Weertman Friction law

Lateral BC:

zero horizontal velocities

Glen's flow law

$$D_{ij} = A\tau_e^{n-1} S_{ij} \quad ; \quad S_{ij} = A^{-1/n} I_{D_2}^{(1-n)/n} D_{ij}$$

$$A = A(T') = A_0 \exp^{-Q/RT'}$$

$$A = A_1 = 2.89 \times 10^{-13} \text{ s}^{-1}\text{Pa}^{-3} \text{ if } T \leq -10°\text{C}$$
$$A = A_2 = 2.43 \times 10^{-2} \text{ s}^{-1}\text{Pa}^{-3} \text{ if } T \geq -10°\text{C}$$

$$Q = Q_1 = 60 \text{ kJ mol}^{-1} \text{ if } T \leq -10°\text{C}$$
$$Q = Q_2 = 115 \text{ kJ mol}^{-1} \text{ if } T \geq -10°\text{C}$$

Cuffey and Paterson (2010)

assume a constant temperature of 0°C

Paterson 2010		
A* =	3.50000E-25	s^-1 Pa^-3
A1 =	2.89165E-13	s^-1 Pa^-3
A2 =	2.42736E-02	s^-1 Pa^-3
Q1 =	60000	J/mol
Q2 =	115000	J/mol

T [°C]	A [s^-1 Pa^-3]	A [a^-1 MPa^-3]
0	2.4029E-24	75.830
-1	1.9945E-24	62.942
-2	1.6533E-24	52.173
-3	1.3685E-24	43.186
-4	1.1312E-24	35.698
-5	9.3370E-25	29.465
-6	7.6958E-25	24.286
-7	6.3339E-25	19.988
-8	5.2054E-25	16.427
-9	4.2716E-25	13.480
-10	3.5000E-25	11.045
-10	3.5000E-25	11.045
-11	3.1520E-25	9.947
-12	2.8363E-25	8.951
-13	2.5501E-25	8.048
-14	2.2910E-25	7.230
-15	2.0564E-25	6.490
-16	1.8444E-25	5.820
-17	1.6528E-25	5.216
-18	1.4798E-25	4.670
-19	1.3238E-25	4.177
-20	1.1831E-25	3.734
-21	1.0565E-25	3.334
-22	9.4260E-26	2.975
-23	8.4019E-26	2.651
-24	7.4822E-26	2.361
-25	6.6570E-26	2.101
-30	3.6580E-26	1.154
-35	1.9601E-26	0.619
-40	1.0225E-26	0.323
-45	5.1843E-27	0.164
-50	2.5496E-27	0.080

Glen's flow law

```
Material 1
! For the ice flow
  Density = Real #rhoi
  Viscosity Model = String "Glen"
  Viscosity = Real 1.0 ! Dummy but avoid warning output
  Glen Exponent = Real 3.0
  Critical Shear Rate = Real 1.0e-10


! properties with T
  Rate Factor 1 = Real #A1
  Rate Factor 2 = Real #A2
  Activation Energy 1 = Real #Q1
  Activation Energy 2 = Real #Q2
  Glen Enhancement Factor = Real 1.0
  Limit Temperature = Real -10.0
  Relative Temperature = Real 0.0
End
```

!! For the vectorized Stokes solver used here
For the legacy Stokes solver:
`Constant Temperature = Real 0.0`

Parameters file

To be sure to use the same parameters in all the different sif files!
include "../Parameters/Physical_Parameters.IN »

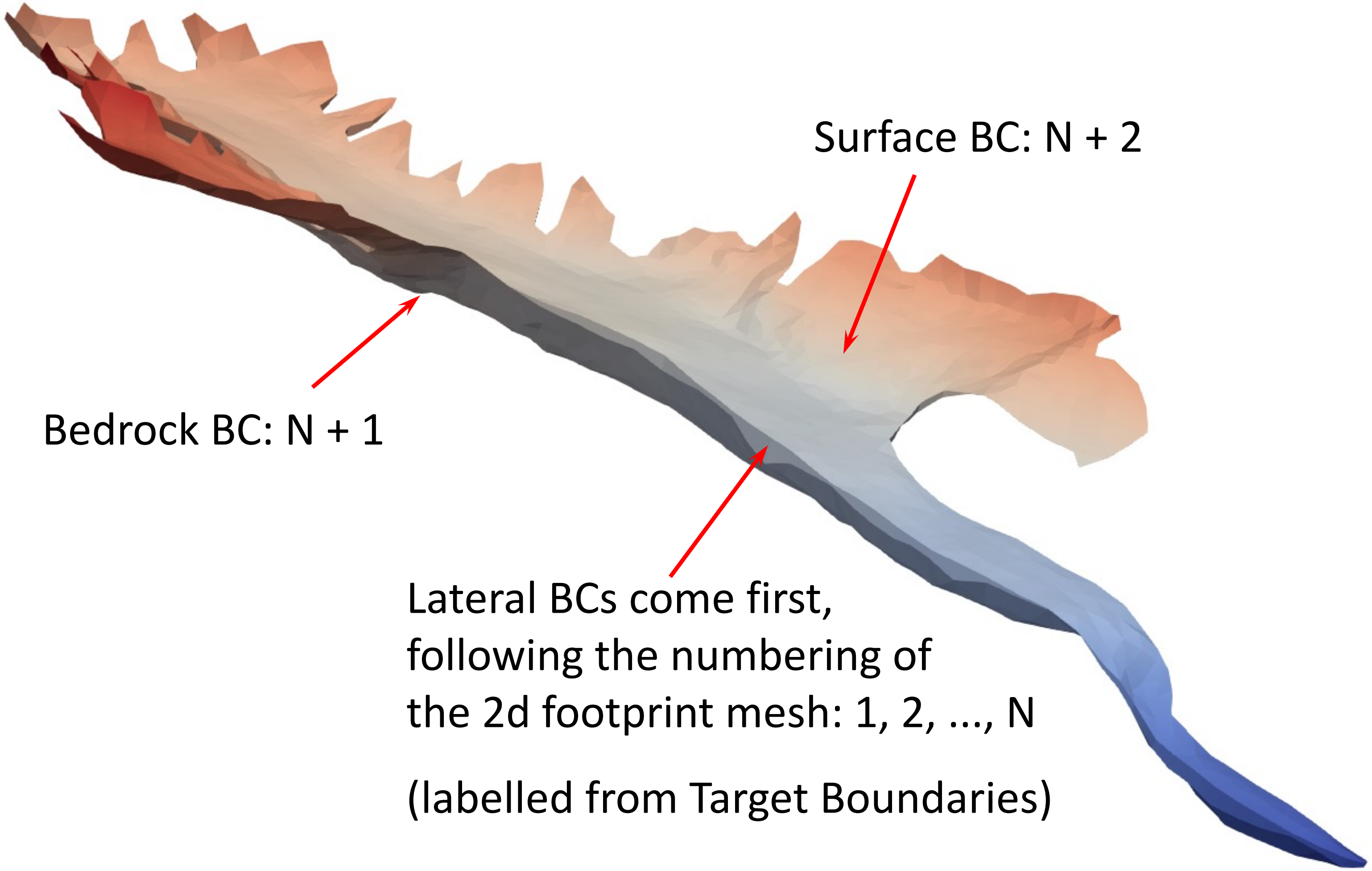
Parameters file:

```
! Define the parameter in MPa - a - m
#yearinsec = 365.25*24*60*60
#rhoikg = 917.0
#rhowkg = 1000.0
#rhoi = rhoikg/(1.0e6*yearinsec^2)
#rhow = rhowkg/(1.0e6*yearinsec^2)
#gravity = -9.81*yearinsec^2
#LHeat = 334000.0

! Prefactor from Cuffey and Paterson (2010) in MPa^{-3} a^{-1}
#A1 = 2.89165e-13*yearinsec*1.0e18
#A2 = 2.42736e-02*yearinsec*1.0e18
#Q1 = 60.0e3
#Q2 = 115.0e3

! Parameters for the Weertman friction law
#Cw = 0.074
#mw = 1.0/3.38

#Nz = 10
#MinH = 10.0
```

UNIL | Université de Lausanne

Numbering of the Boundary Conditions (BC) using Internal Extrusion

```
! lateral side of the glacier
Boundary Condition 1
  Target Boundaries = 1
  Name = "side"
  ...
End

! Bedrock
Boundary Condition 2
  Name = "bed »
  ...
End

! Upper Surface
Boundary Condition 3
  Name = "upper surface"
  ...
End
```



Surface BC: N + 2

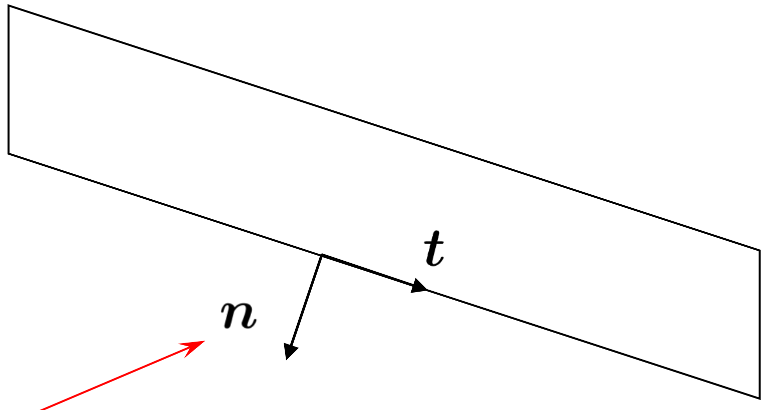
Bedrock BC: N + 1

Lateral BCs come first,
following the numbering of
the 2d footprint mesh: 1, 2, ..., N

(labelled from Target Boundaries)

Weertman friction at the base

```
! Bedrock
Boundary Condition 2
  Name = "bed"
  Bottom Surface = Equals BedDEM
  Mass Consistent Normals = Logical True
  Normal-Tangential Velocity = Logical True
  Velocity 1 = Real 0.0e0
  Weertman Friction Coefficient = Real #Cw
  Weertman Exponent = Real #mw
  Weertman Linear Velocity = Real 0.00001
End
```



$$\boldsymbol{u} \cdot \boldsymbol{n} = 0$$

$$\tau_{nt_i} = C \cdot u_b^m \cdot u_{t_i} \text{ for } i = 1, 2$$

$$u_b^2 = u_{t_1}^2 + u_{t_2}^2 \qquad m \approx 1/n$$

When using the legacy Stokes solver, one should add

```
Slip Coefficient 2 =  Variable Coordinate 1
  Real Procedure "ElmerIceUSF" "Sliding_Weertman"
Slip Coefficient 3 =  Variable Coordinate 1
  Real Procedure "ElmerIceUSF" "Sliding_Weertman"
```

Two others BCs

```
! lateral side of the glacier
Boundary Condition 1
  Target Boundaries = 1
  Name = "side"
  Mass Consistent Normals = Logical True
  Normal-Tangential Velocity = Logical
True
  Velocity 1 = Real 0.0
End
```

No flux on the side of the glacier

$$\boldsymbol{u} \cdot \boldsymbol{n} = 0$$

```
! Upper Surface
Boundary Condition 3
  Name = "upper surface"
  Top Surface = Equals SurfDEM
End
```

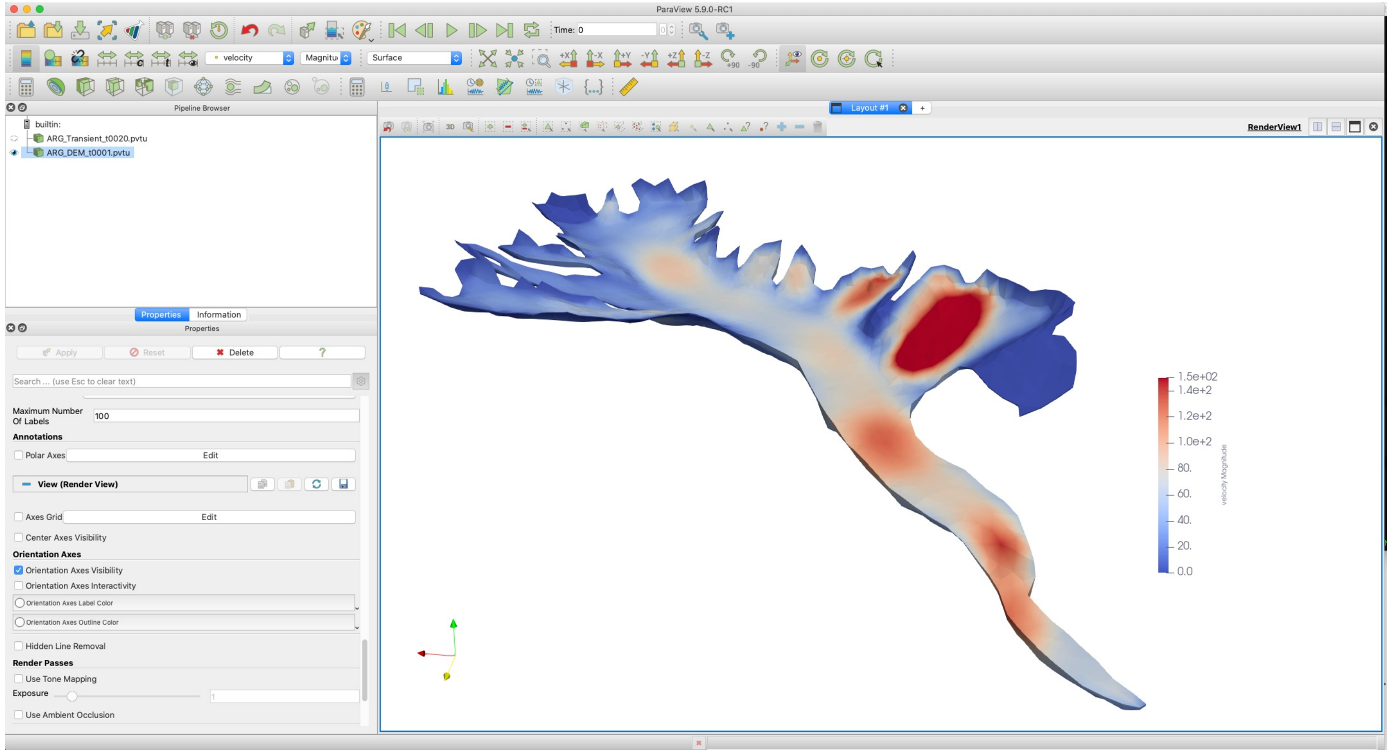
Stress free surface of the upper surface

Natural BC → Nothing to do!

Run the simulation

```
serial: ElmerSolver initialise_DEM.sif

parallel: mpirun –np 4 ElmerSolver initialise_DEM.sif
```

use Paraview to visualise the results (vtu files saved in the mesh directory)



2_ICE_AGE

The objective of this step is to add the semi-lagrangian (ParticleAdvector) solver to compute the ice age and trajectories

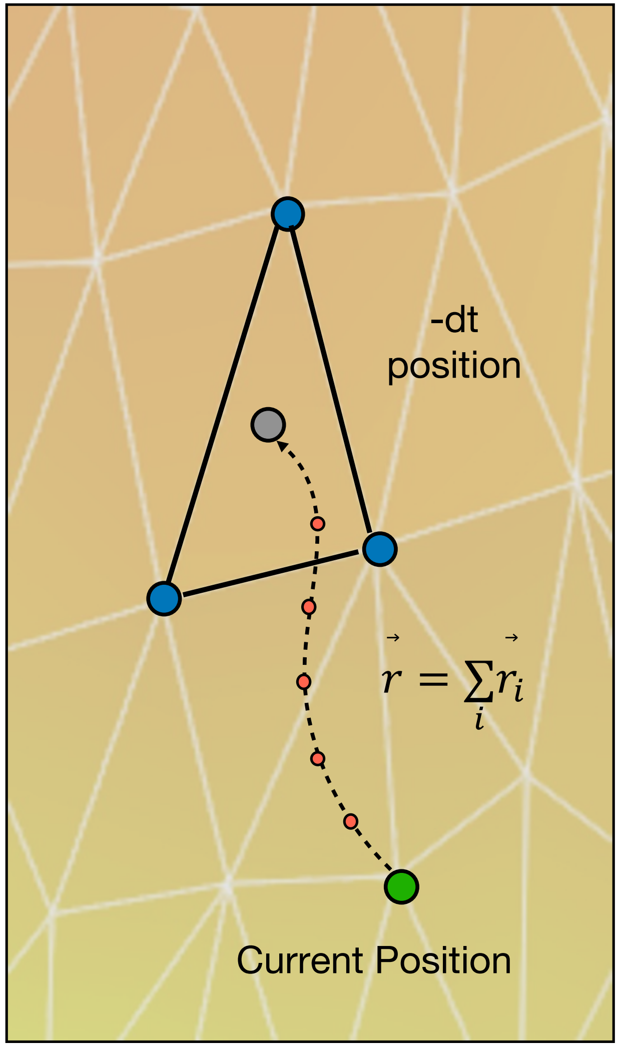
```
Solver 5
  Equation = ParticleAdvector
  Procedure = "ParticleAdvector" "ParticleAdvector"
  ...
! Initialize particles at center of elements (as opposed to nodes)
  Advect Elemental = Logical True
  ...
  Velocity Variable Name = String "Flow Solution"
  ...
! The internal variables for this solver
  Variable 1 = String "Particle Distance"
  Variable 2 = String "Particle Time"
  Operator 2 = String "Cumulative"
  ...
! The field variables being advected
  Variable 3 = String "Coordinate 1"
  Result Variable 3 = String "Advected X"
  Variable 4 = String "Coordinate 2"
  Result Variable 4 = String "Advected Y"
  Variable 5 = String "Coordinate 3"
  Result Variable 5 = String "Advected Z"
End
```

Add one solver

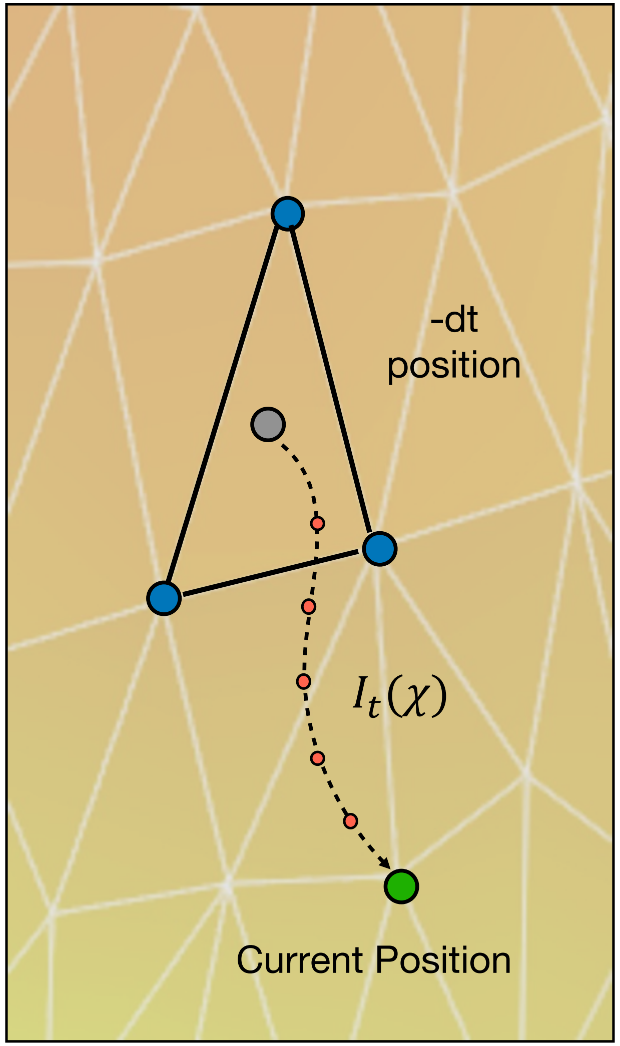
```
Equation 1
  Active Solvers(5) = 1 2 3 4 5
End
```

2_ICE_AGE: semi-Lagragian method

backward stage



forward stage

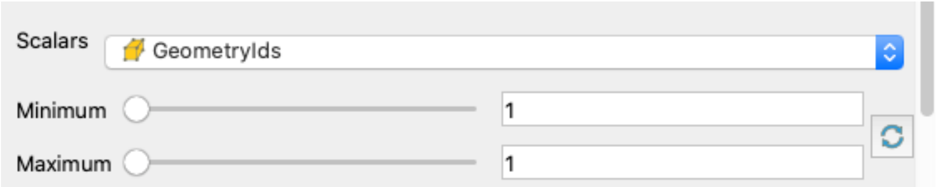


2_ICE_AGE: visualise results in Paraview

Visualise results in Paraview

Because of Elemental variables, results are not depicted on the surface → Nothing to see!

One have to select only the bulk elements using the Threshold filter in Paraview 



An other option is to output only on bulk elements (in the Simulation section)

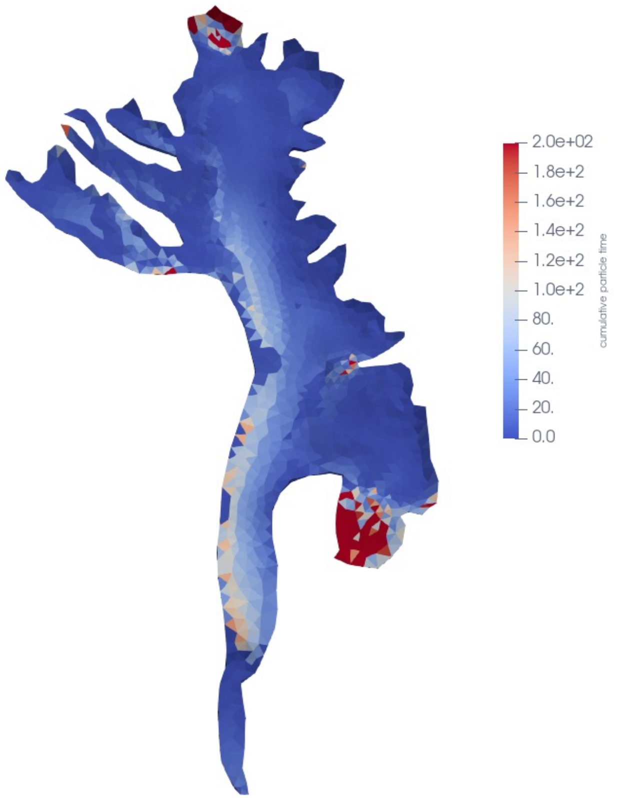
```
vtu: Save Bulk Only = Logical True
```

(might be problematic for the next step 3_TRANSIENT where we output a number of variables only on the surface)

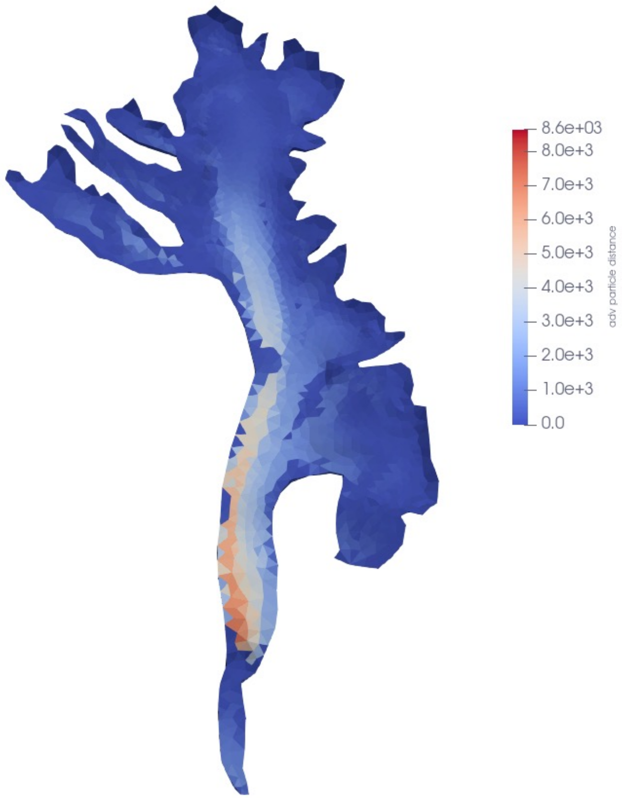
A third solution is to use the filter "Cell Data to Point Data" in Paraview

2_ICE_AGE: results

Age of the ice



Distance of the ice particles



Not so nice certainly because the upper surface is not relaxed

Try later: `Restart File = "../../3_TRANSIENT/ARG_mesh/ARG_Transient.result"`

3_TRANSIENT: Simulation section

A number of Solvers have to be added

- `StructuredProjectToPlane` to compute Depth and Thickness
- `IcyMaskSolver(*)` to calculate a mask for ice and land
- `Compute2DNodalGradient(**)` to compute the surface gradient needed by SMB solver
- `TransientMassBalance(***)` to compute the SMB to be applied at the surface of the glacier
- `FreeSurfaceSolver` to evolve the altitude of the upper free surface
- `Scalar_OUTPUT` to output some integrated variables (volume, area, front position)
- `SaveLine` to output some variables along a borehole

(*) In the distrib, but we use a slightly improved version that will be pushed soon
(**) In the distrib, but compiled locally in case mmg not working on your instal
(***) Not yet in the distrib but should be committed soon

3_TRANSIENT: Simulation section

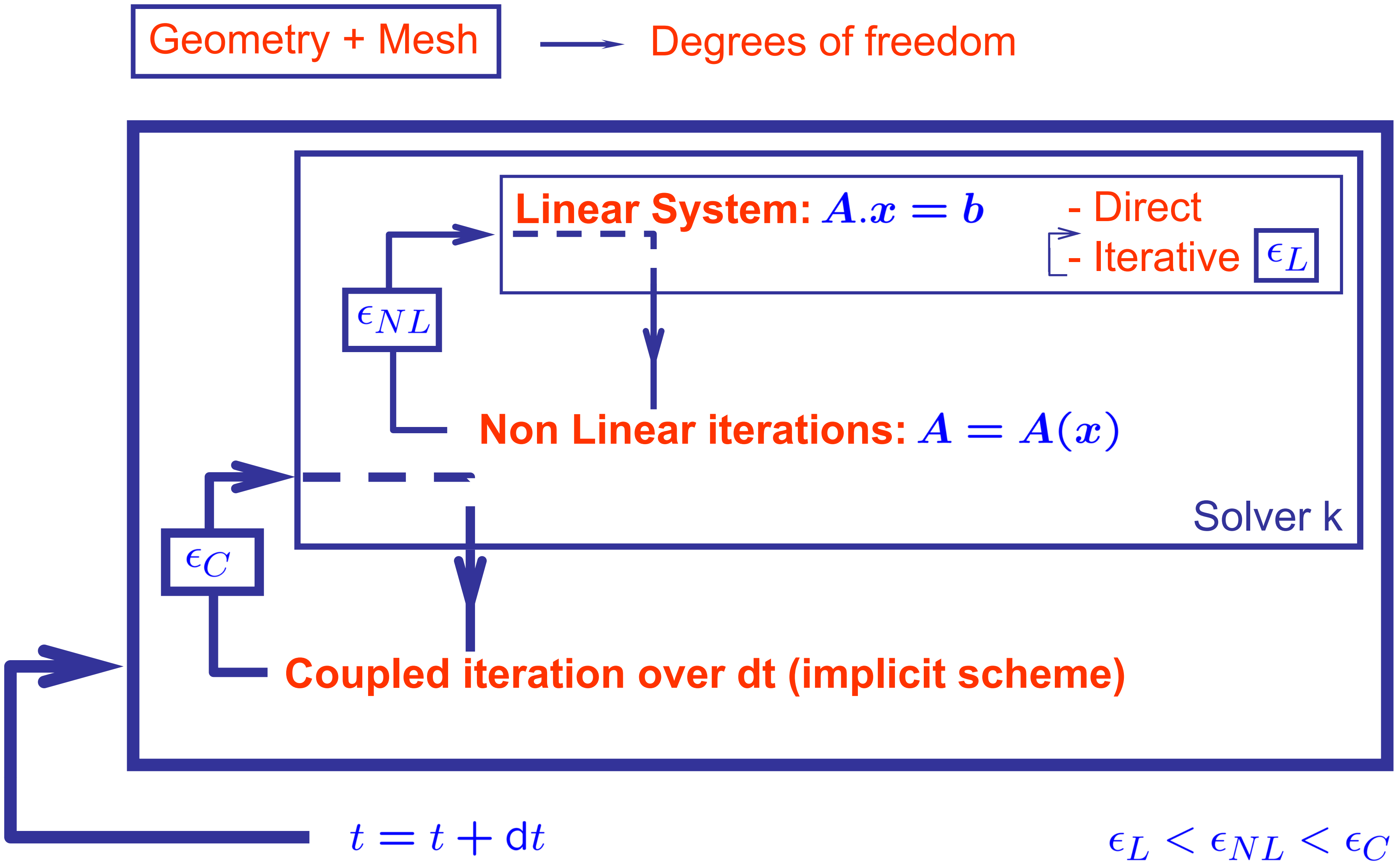
The Simulation Section has to be modified

```
Simulation
  Simulation Type = Transient
  ...
  Timestepping Method = "bdf"
  BDF Order = 2

  Timestep Intervals(1) = #duration * steps_per_year
  Output Intervals(1) = #steps_per_year
  Timestep Sizes(1) = #dt

  Steady State Max Iterations = 1
  Steady State Min Iterations = 1

  Output File = "$name_simu$.result"
  Post File = "$name_simu$.vtu"
  vtu:VTU Time Collection = Logical True

  Restart File = "../../1_IMPORT_DEM/ARG_mesh/ARG_DEM.result"
  Restart Position = 0
  ! Restart Time = Real #Startyear
  Restart Before Initial Conditions = Logical true

  max output level = 3
End
```

→ Backward Differences Formulae

→ To control the "implicity" of the solution over one time step (here 1 means explicit)

Sketch of a transient simulation



Free surface solver

The free surface solver only apply to the boundary 3 (upper surface)

→ Define a 2nd body which is on boundary 3.

```
! The upper free surface
Body 2
  Name = "surface"
  Equation = 2
  Material = 1
  Body Force = 2
  Initial Condition = 2
End
```

where `Equation 2`, `Body Force 2` and `Initial Condition 2` are defined for the free surface equation of the upper surface.

Tell in BC3 that this is the body 2:

```
! Upper Surface
Boundary Condition 3
  Name = "upper surface"
  Body Id = 2
  Top Surface = Equals Zs
End
```

Free surface solver

```
Initial Condition 2
  Zs = Equals SurfDEM
  Ref Zs = Equals SurfDEM
End
```

```
Body Force 2
  Zs Accumulation Flux 1 = real 0.0
  Zs Accumulation Flux 2 = real 0.0
  Zs Accumulation Flux 3 = Equals "Mass Balance"
  ...
End
```

The variable Mass Balance is computed by the TransientMassBalance Solver

```
Equation 2
  Active Solvers(5) = 3 5 6 7 8
  Flow Solution Name = String "Flow Solution"
  Convection = Computed
End
```

IcyMask (3), Compute2DNodalGradient (5), TransientMassBalance (6) and Scalar_OUTPUT (8) are also solved on the upper free surface only

Free surface solver

We add a limiter to ensure that $Z_s \geq b + h_{min}$

```
Material 1
  ...
  Min Zs= Variable BedDEM
    Real LUA "tx[0]+ MinH »

  Max Zs=  Variable Coordinate 3
    Real LUA "tx[0]+ 18.0"
End
```

and also that change over one time step are limited (avoid instability in the Free Surface Solver)

Need to let know the Free Surface Solver that Zs has limits

```
 Apply Dirichlet = Logical true
```

and make the Free Surface solver iterative

```
Nonlinear System Max Iterations = 100
Nonlinear System Min Iterations = 2
Nonlinear System Convergence Tolerance = 1.0e-6
```

StructuredMeshMapper

We say in StructuredMeshMapper that the top surface is defined by the variable Zs:

```
Solver 1
  Equation = "MapCoordinate"
  Procedure = "StructuredMeshMapper" "StructuredMeshMapper"

  Active Coordinate = Integer 3
  Mesh Velocity Variable = String "dSdt"
  Mesh Update Variable = String "dS"
  Mesh Velocity First Zero = Logical True

  Top Surface Variable Name = String "Zs"
  ...
End
```

And delete from the BC the initial definition of the top surface:

```
! Upper Surface
Boundary Condition 3
  Name = "upper surface"
  Body Id = 2
  Top Surface = Equals SurfDEM
End
```

Transient_MassBalance

This solver compute the surface mass balance using given

- daily precipitation and temperature over a given period
- a mask of the relief around the glacier
- a mask for accumulation factor (to account for sur-accumulation due to avalanche for example)

For each glacier, the model has to be calibrated against some data

The solver is based on a PDD method for the melt
It also compute a firn thickness to account for different albedo values for firn and ice

It can be run in transient or steady

```
Transient MB = Logical False
```

If False, this will compute the mean mass balance associated to the provided daily temperature and precipitation files
If True, the SMB is calculated dynamically for each timestep (which can be larger than 1 day, but smaller than a year if one want to see seasonal variation!)

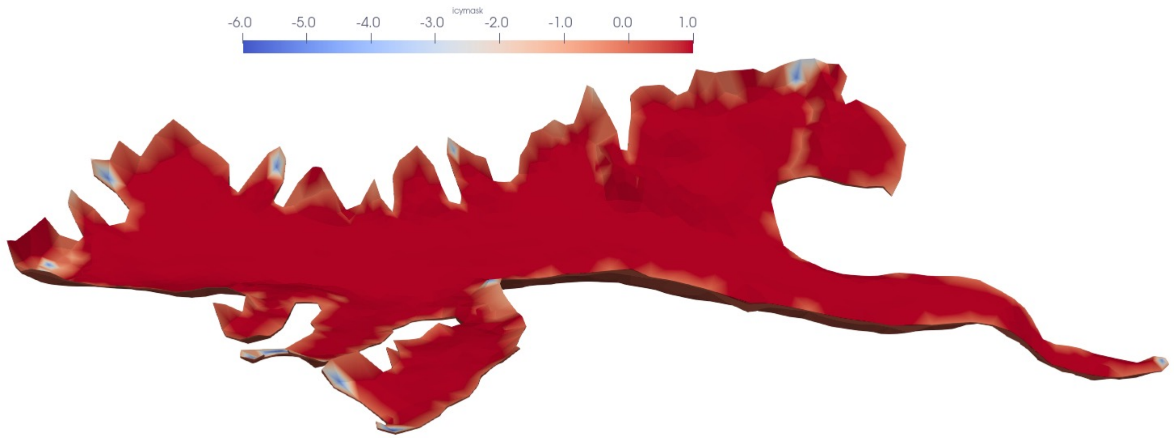
IcyMaskSolver

This solver compute a Mask which is

- +1 for glacier nodes (H > Hmin)
- 0 for the nodes belonging to the glacier contour (H=Hmin)
- -1 for Ice free nodes (H <= Hmin)
- <-1 for isolated nodes

Isolated nodes (and edges) are defined by nodes (edges) surrounded by only ice free nodes. They are tagged with a mask value < -1.

This allow to compute the glacier surface and volume if the mesh contour is larger than the glacier contour



Scalar_OUTPUT

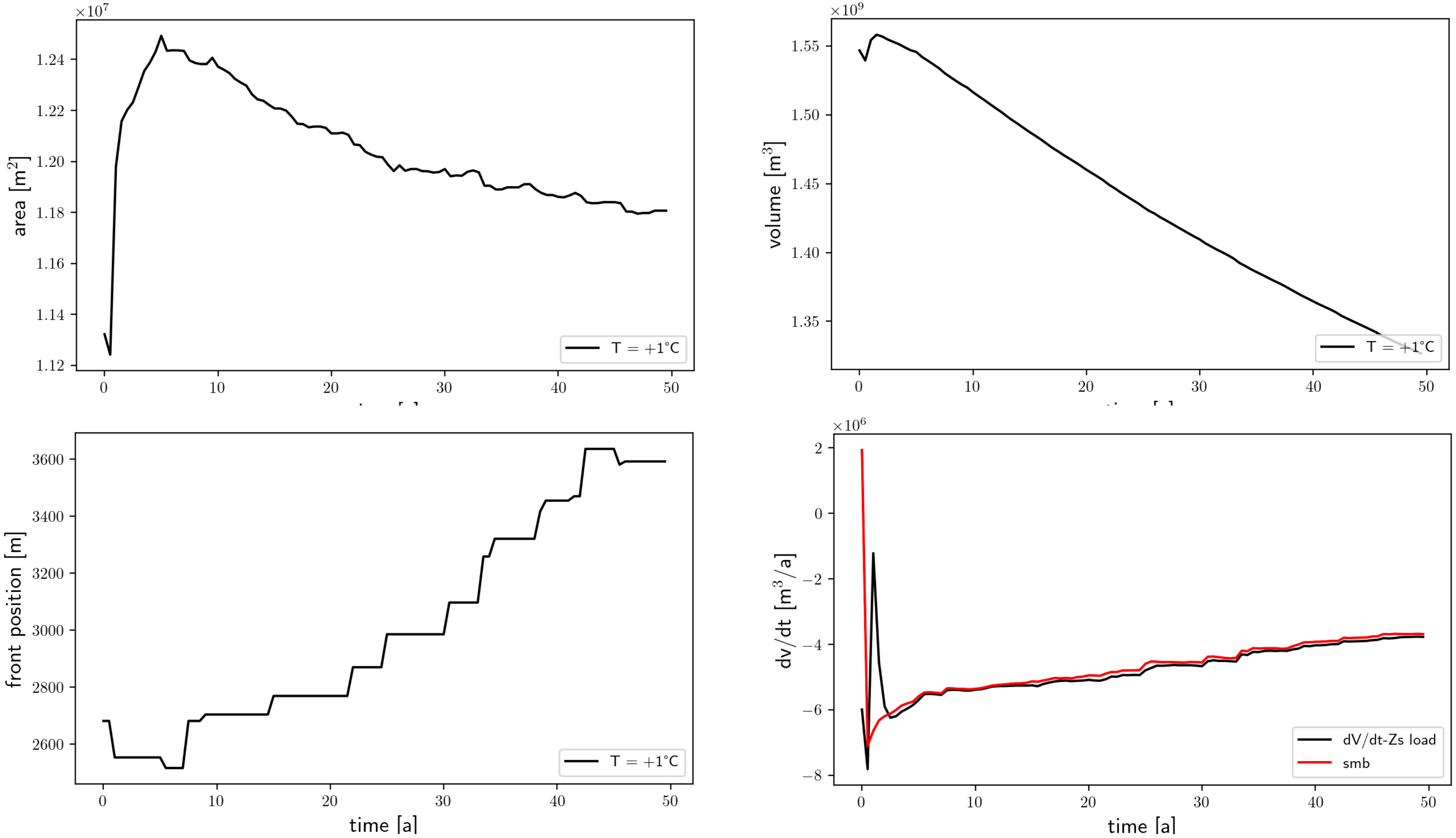
This solver allows to output some integrated scalar variables with time:

```
 1: Time
 2: Volume
 3: Volume Change
 4: Area
 5: Ablation Area
 6: Accumulation Area
 7: SMB Total
 8: SMB Ablation
 9: SMB Accumulation
10: Residual Flux
11: Altitude bin 1
12: SMBAlt Tot 1
13: AreaAlt 1
....
41: Altitude bin 11
42: SMBAlt Tot 11
43: AreaAlt 11
44: Front Elevation
45: X Front
46: Y Front
47: Front Distance
```

from 1600m to 3600m every 200m,
SMB and Area per bin

Scalar_OUTPUT

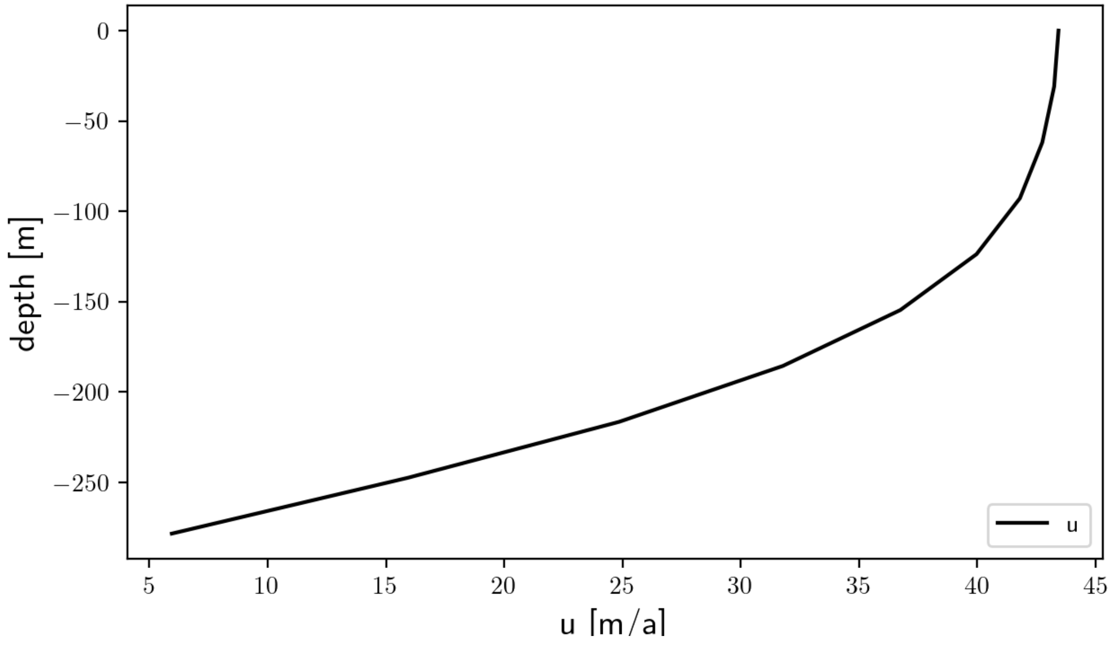
example for T = +1°C



Save Line (borehole)

Objective: save some variable along a given vertical profile (e.g. at a borehole location)

```
Solver 9
  Equation = String "Borehole"
  Exec solver = After timestep


  Procedure = "SaveData" "SaveLine"
  Filename = "GlacierOut/Profil_$name_simu$.dat"
  File append = false


  Variable 1 = Velocity 1
  Variable 2 = Velocity 2
  Variable 3 = Velocity 3
  Variable 4 = melting


  Polyline Coordinates(2,3) = Real  959323.0 117572.5
2000.0 959323.0 117572.5 2500.0
End
```



Compilation of local Solvers

in the bin directory, use the Compile.sh

```
> sh ./Compile.sh
```

```
elmerf90 -o MassBalance ../SRC/TransientMassBalance_MaskRelief.F90
elmerf90 -o Scalar_OUTPUT ../SRC/Scalar_OUTPUT_Glacier.F90
elmerf90 -o Compute2DNodalGradient ../SRC/Compute2DNodalGradient.F90
```

Some other potential tests

Re-run the 2_ICE_AGE starting from the relaxed velocity field of stage 3_TRANSIENT

Modify: `Restart File = "../../3_TRANSIENT/ARG_mesh/ARG_Transient_1C.result"`

Change `SurfDEM` by `Zs`

in age_of_ice.sif

See the effect of a +2°C in air temperature on Argentière Glacier

Modify: `TempCorrec= real 2.0`

in transient.sif

Compute the effect of a -2°C in air temperatire on Argentière Glacier

Modify: change the mesh contour to cover a larger domain (`contour_Argentiere_large.dat`)

Modify: `TempCorrec= real -2.0`

Re-run the whole setup (from 0_MakeMSH to 3_TRANSIENT) - change names of output!

Increase the mesh resolution to get nicer results...

Compute the emergence velocity (see MLB example from last year course)

Run your own glacier geometry...