A (nearly) real-world application

Synth Glacier

Samuel Cook

IGE - Grenoble - France

Synth Glacier

✓ **Context**
- Glaciology: how, why, where?

✓ **Step 1**
- Model initialisation

✓ **Step 2**
- Growing Synth Glacier to a steady state

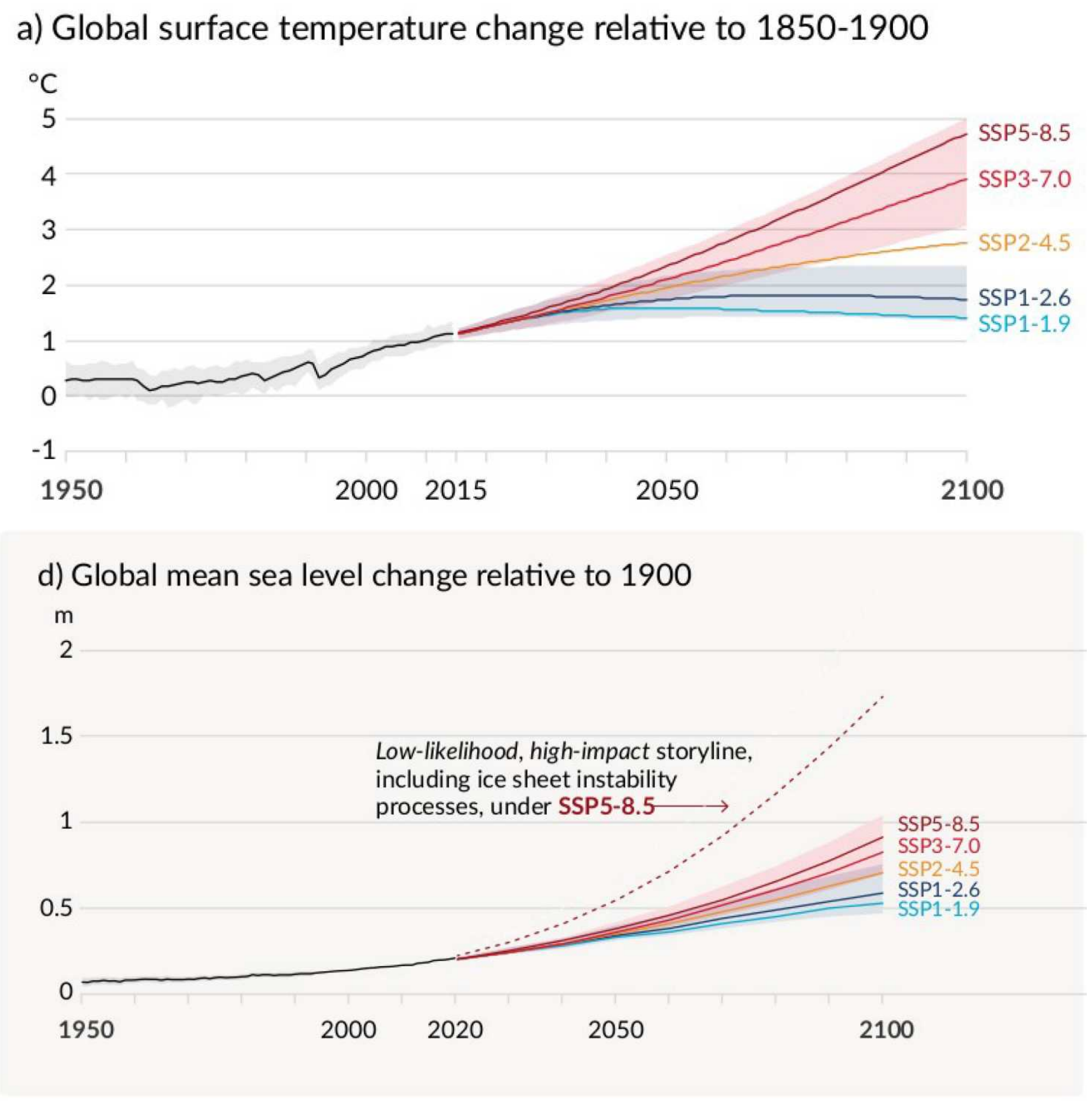
✓ **Step 3**
- Perturbing Synth Glacier

Why do we study glaciers?

- They contain lots of water
- They're changing fast
- Loss of water resources
- Sea-level rise

Why do we study glaciers?

- So we want to know how they'll change
- Also useful to know what they did in the past
- Can tell us where we're going



IPCC (2021)

How do we study glaciers?

How do we study glaciers?

- Fieldwork
- Remote sensing
- Modelling

Modelling ice

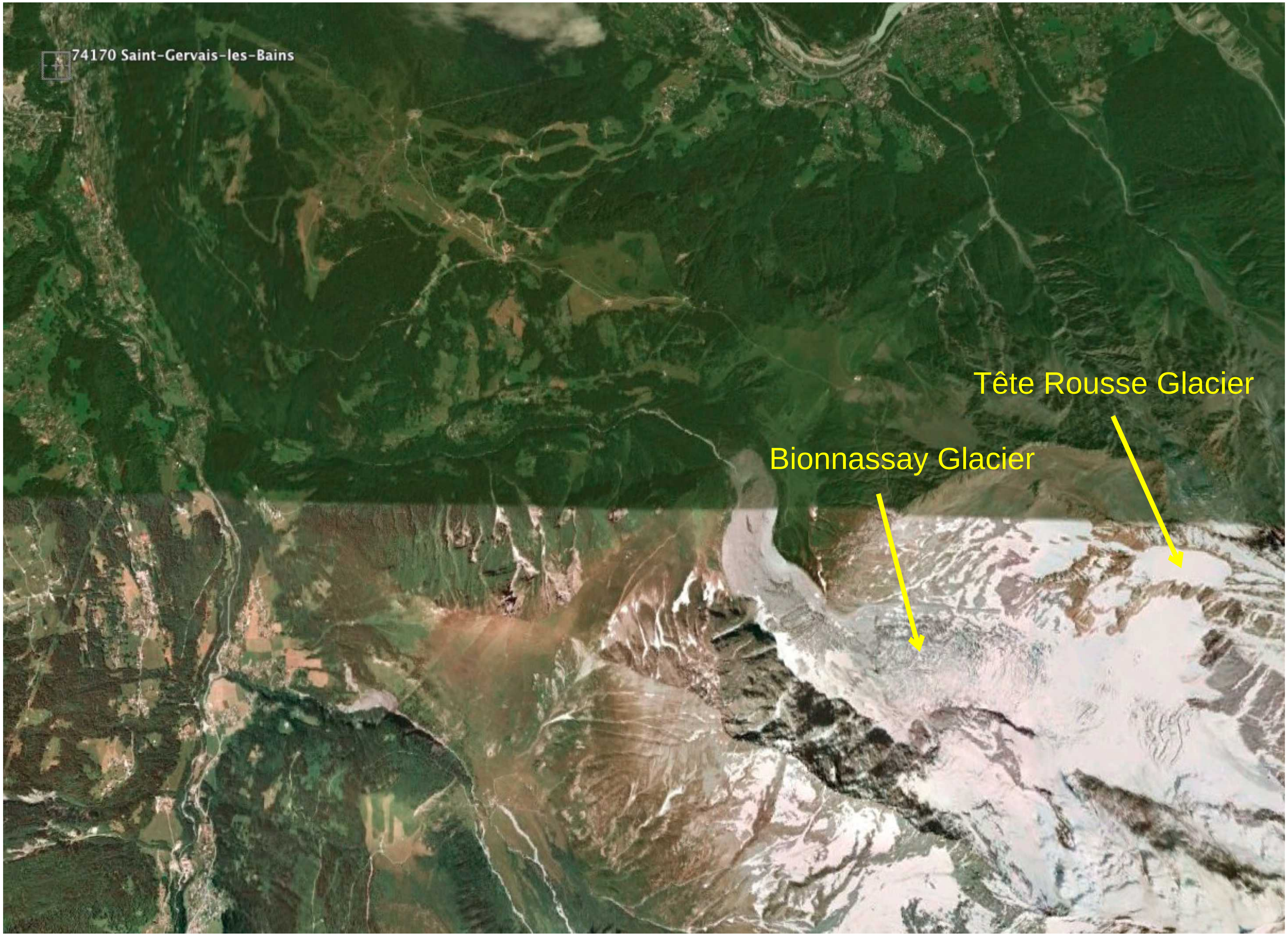
- Very useful
- Stuff we can’t observe
- Reaction to global warming
- Multiple scales

Tête Rousse Glacier

✓ **Context**

- The history of Tête Rousse Glacier
- The 2010 water filled-cavity
- Analysis of the cavity roof stability (Autumn 2010)

Location (Mont Blanc Area, France Alps)



Location (Mont Blanc Area, France Alps)

Tête Rousse glacier
3100 to 3300 m
0.08 km^2 (2007)

@B. Jourdain

Chronology

Past History – The 1892 catastrophe

Contemporary history:

2007-10 - Studies to answer the question about the necessity to maintain the tunnel

07/2010 - A water filled cavity under pressure is discovered

- Crisis – Artificial drainage

2011 - Small research program to understand the formation of the cavity

- New crisis – Artificial drainage

2012 - New Artificial drainage needed

2016 - Building of an artificial spillway

2020 - Still under surveillance…

The 1892 catastrophe

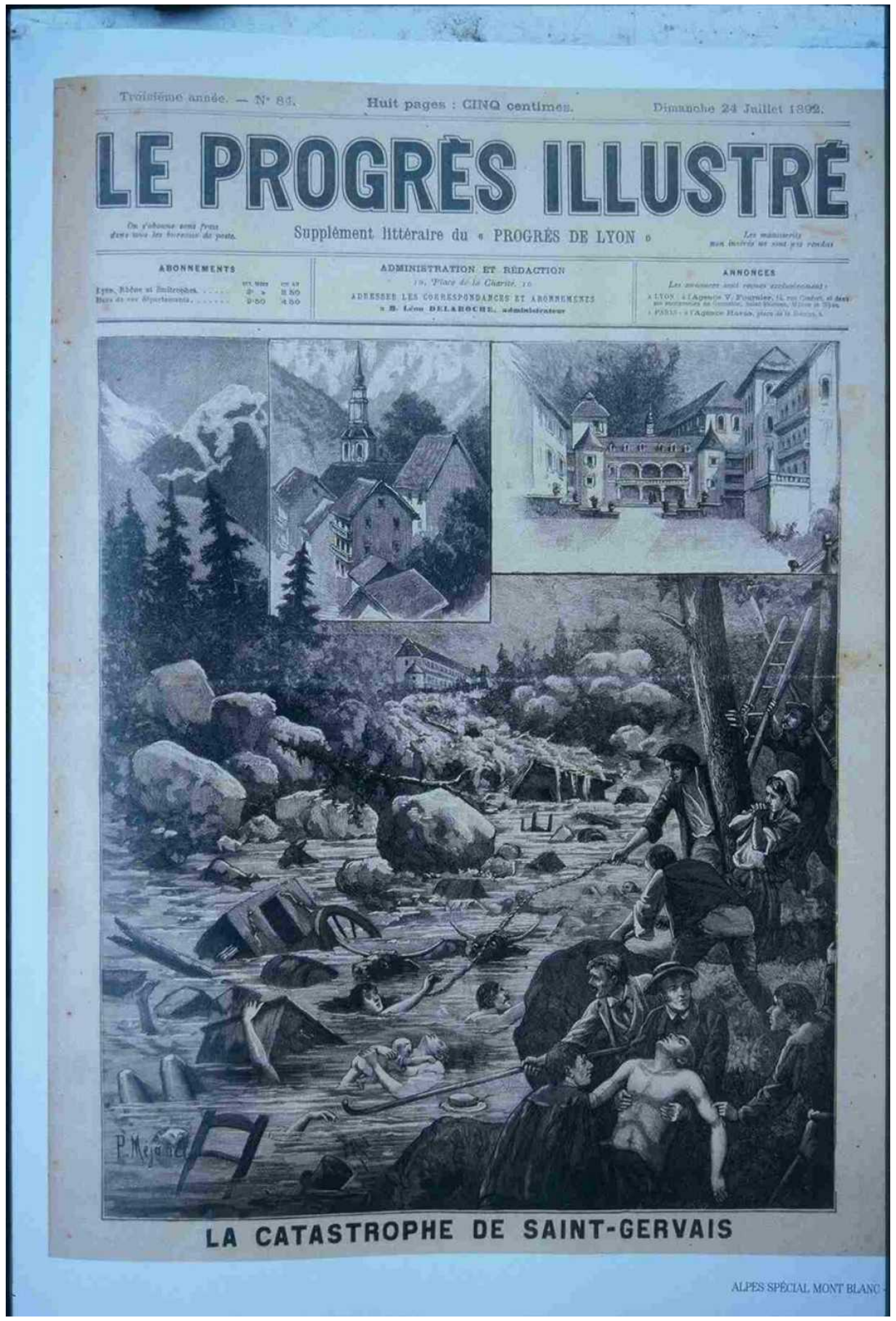
11 July 1892

175 fatalities

100 000 m^3 of water

Flood produced

800 000 m^3 of sediment



Troisième année. — N° 84. Huit pages : CINQ centimes. Dimanche 24 Juillet 1892.

LE PROGRÈS ILLUSTRÉ

Supplément littéraire du « PROGRÈS DE LYON »

ABONNEMENTS

ADMINISTRATION ET RÉDACTION

10, Place de la Charité, 10

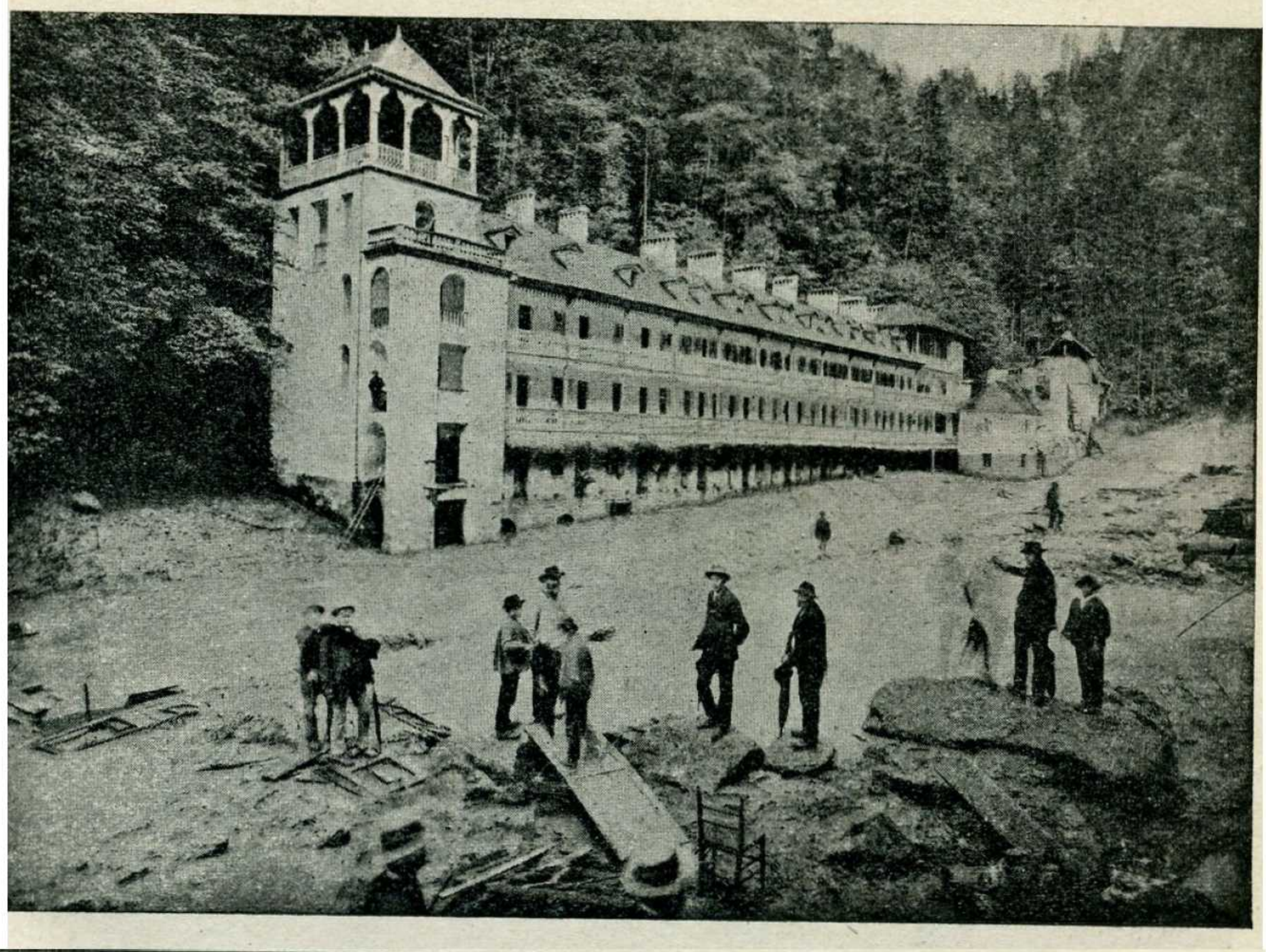
ADRESSER LES CORRESPONDANCES ET ABONNEMENTS

à M. Léon DELAROCHE, administrateur

ANNONCES

LA CATASTROPHE DE SAINT-GERVAIS

ALPES SPÉCIAL MONT BLANC



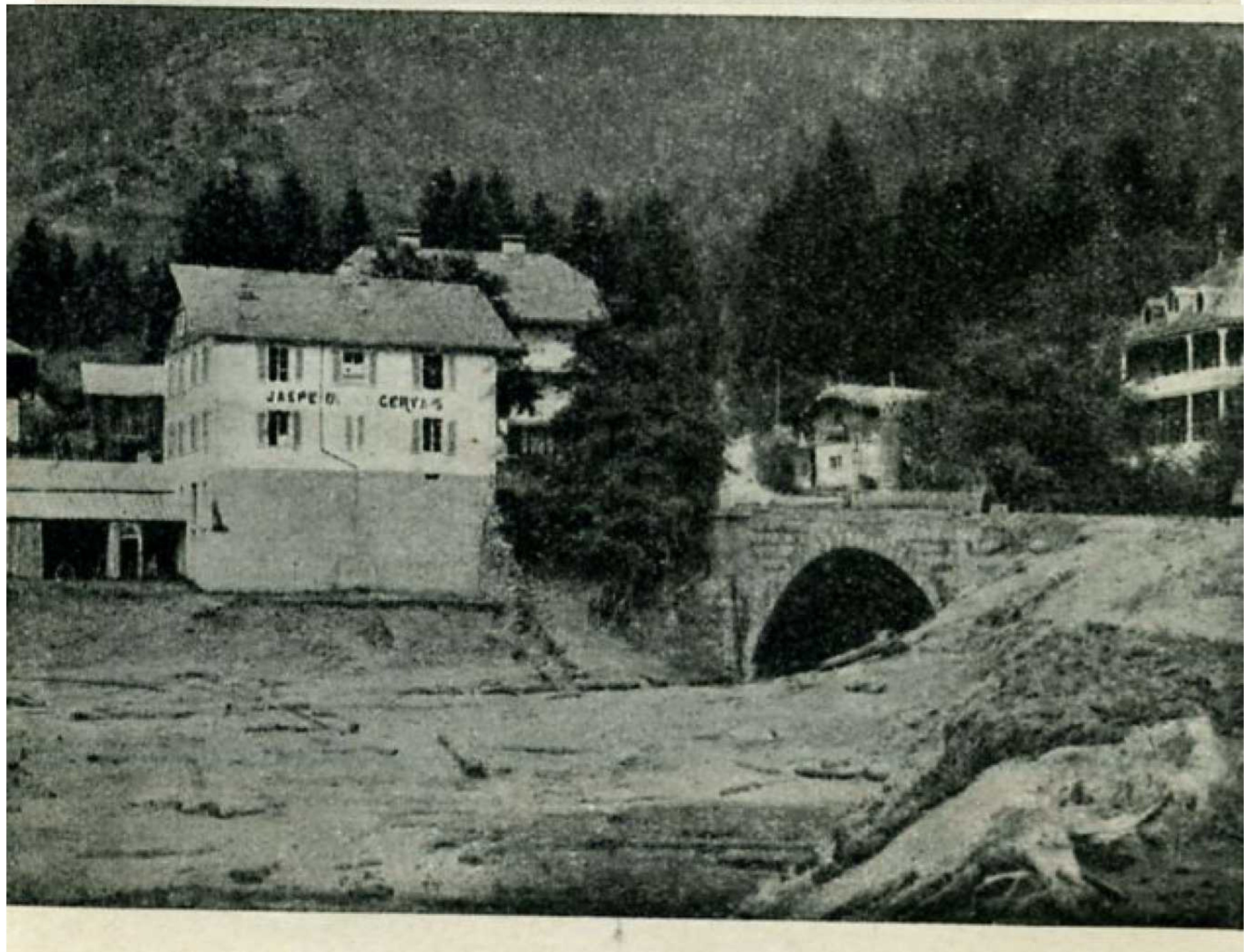
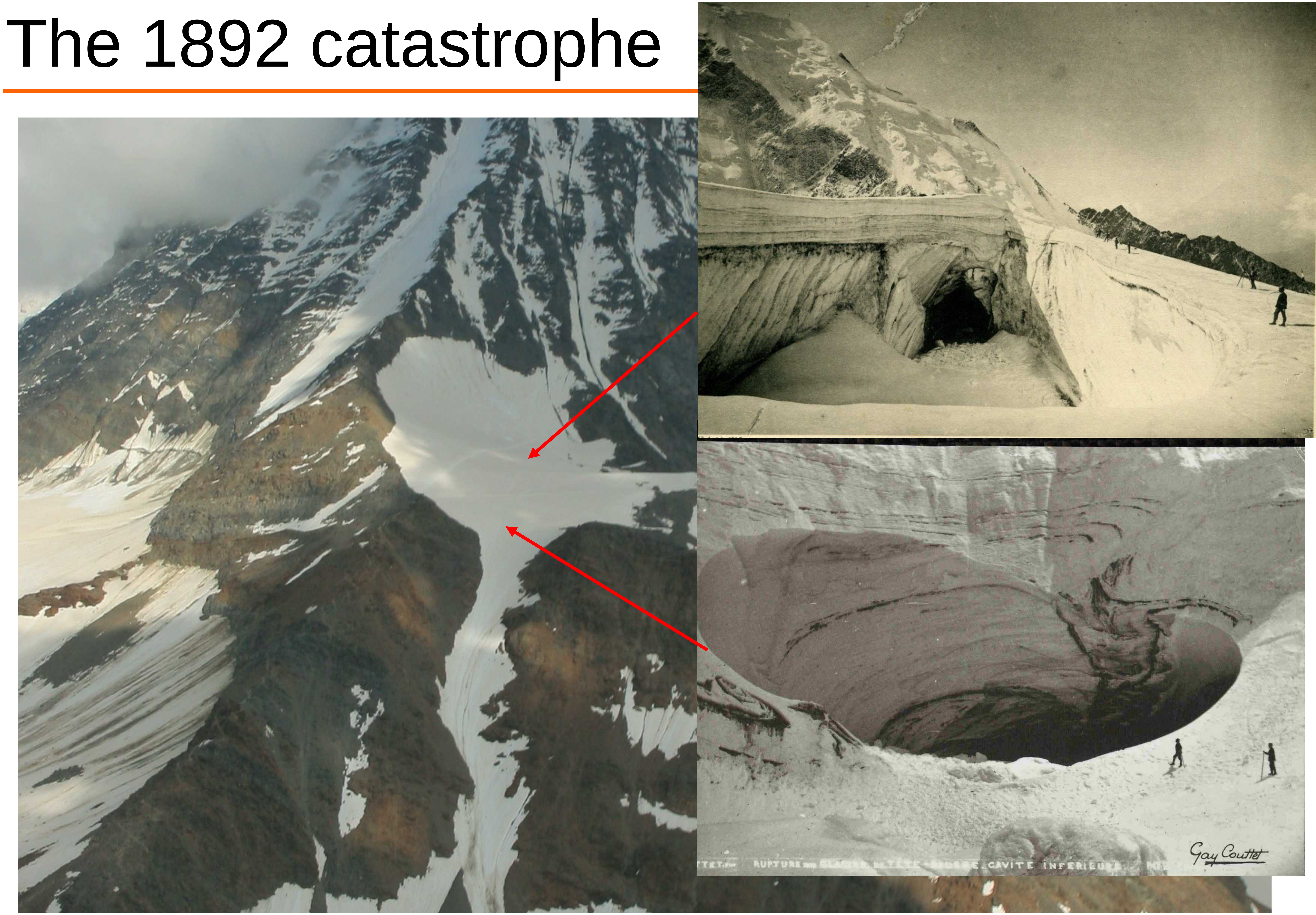


Fig. 22. — Le pont de la route départementale n° 4 tourné et submergé par la lave.
13 juillet 1892. — Cliché Kuss.

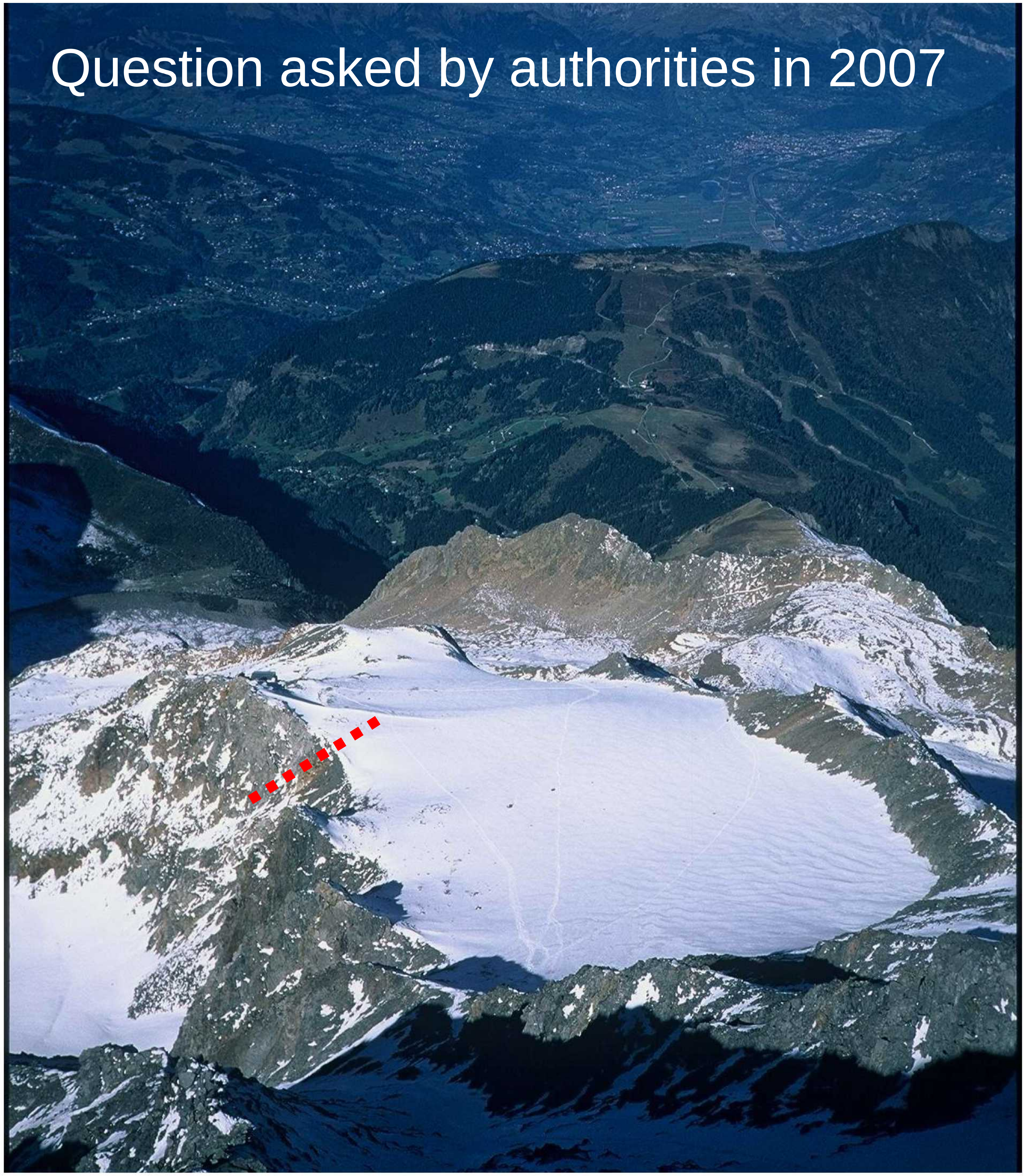
@Vincent, LGGE

The 1892 catastrophe



@Vincent, LGGE

Is there still a risk at Tête Rousse ?



Question asked by authorities in 2007

@Vincent, LGGE

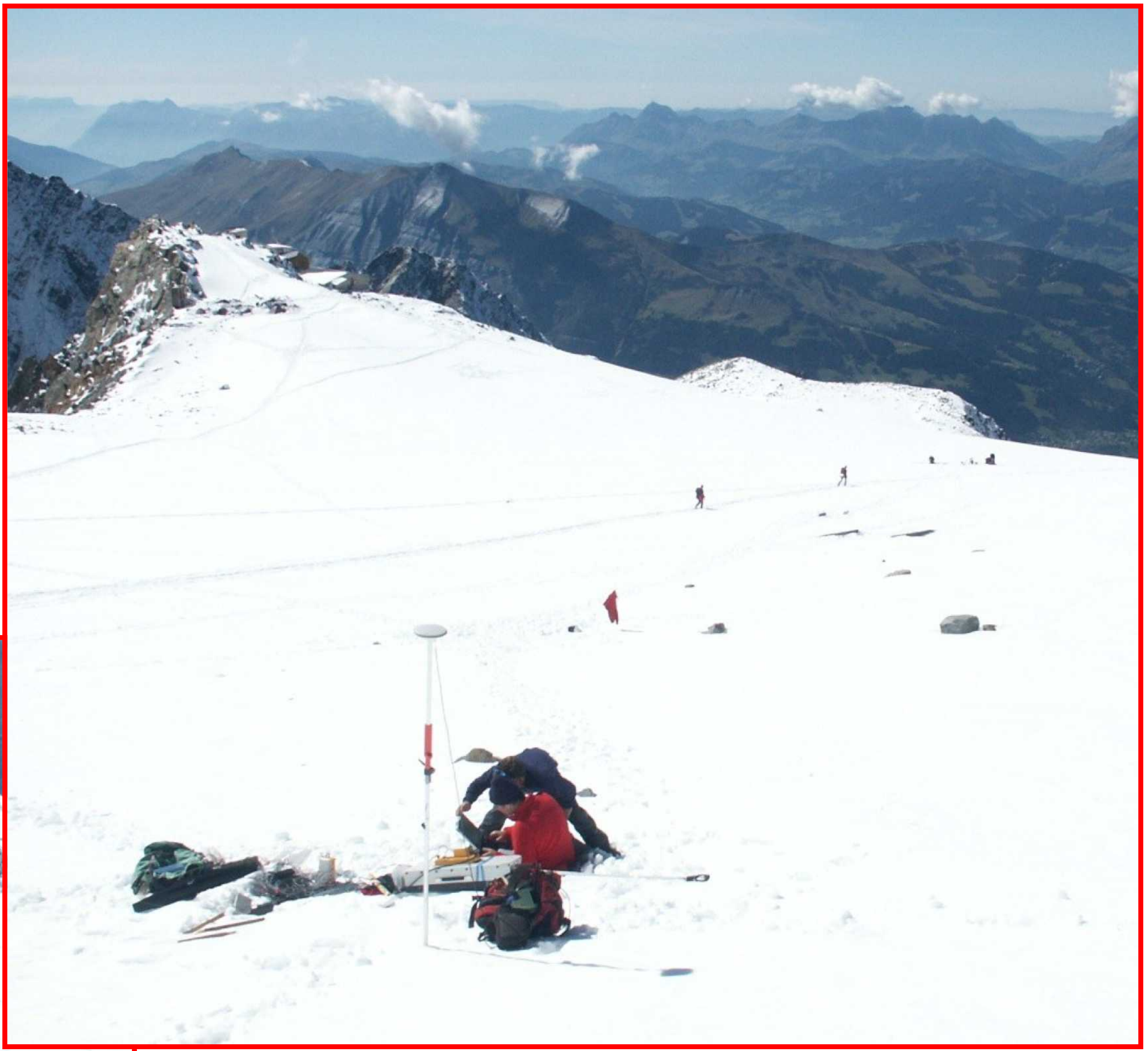
Glaciological studies

. Topographic measurements

. Radar measurements

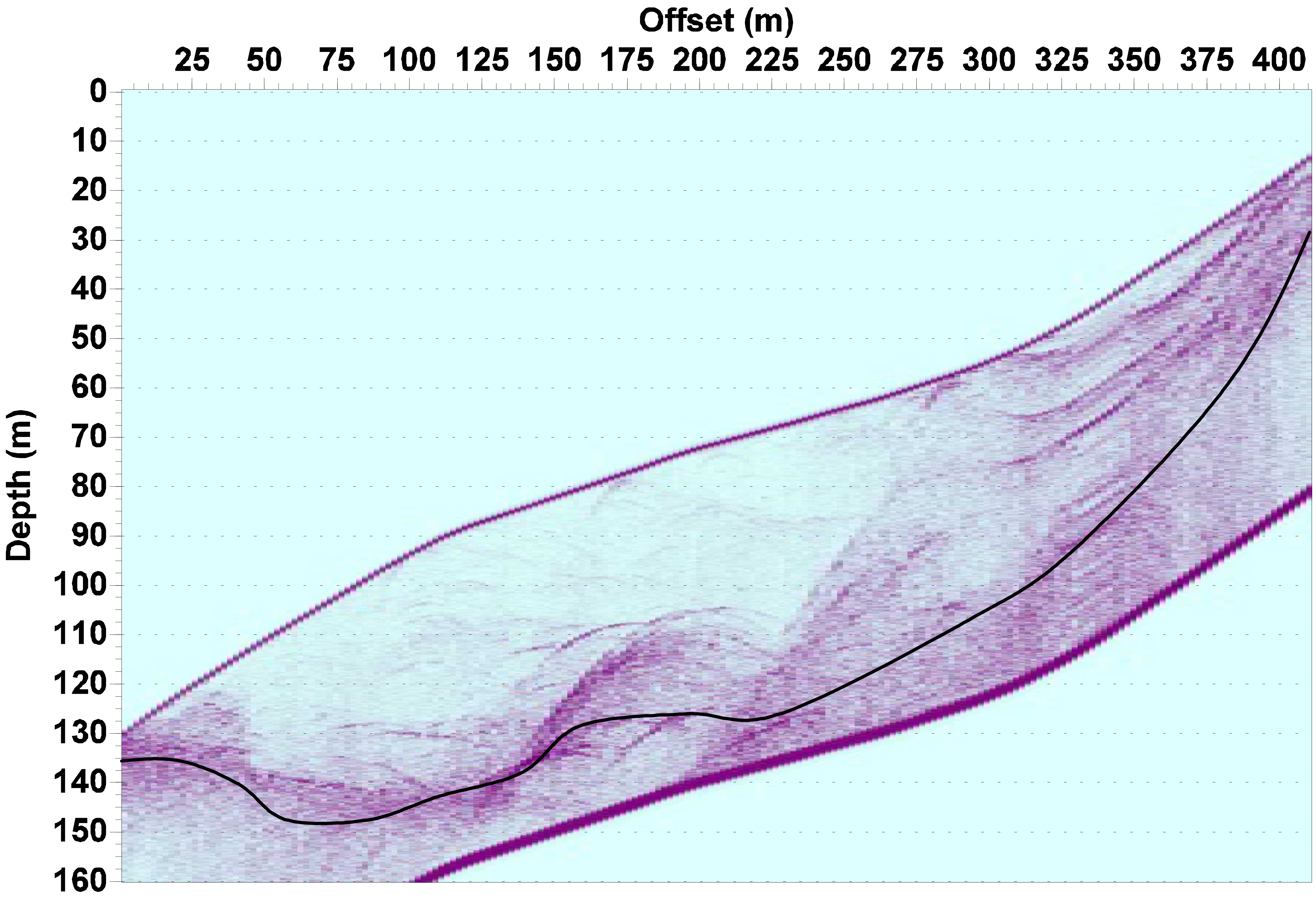
. Temperature measurements

. Mass balance measurements



@Vincent, LGGE

Glaciological studies



@Vincent, LGGE

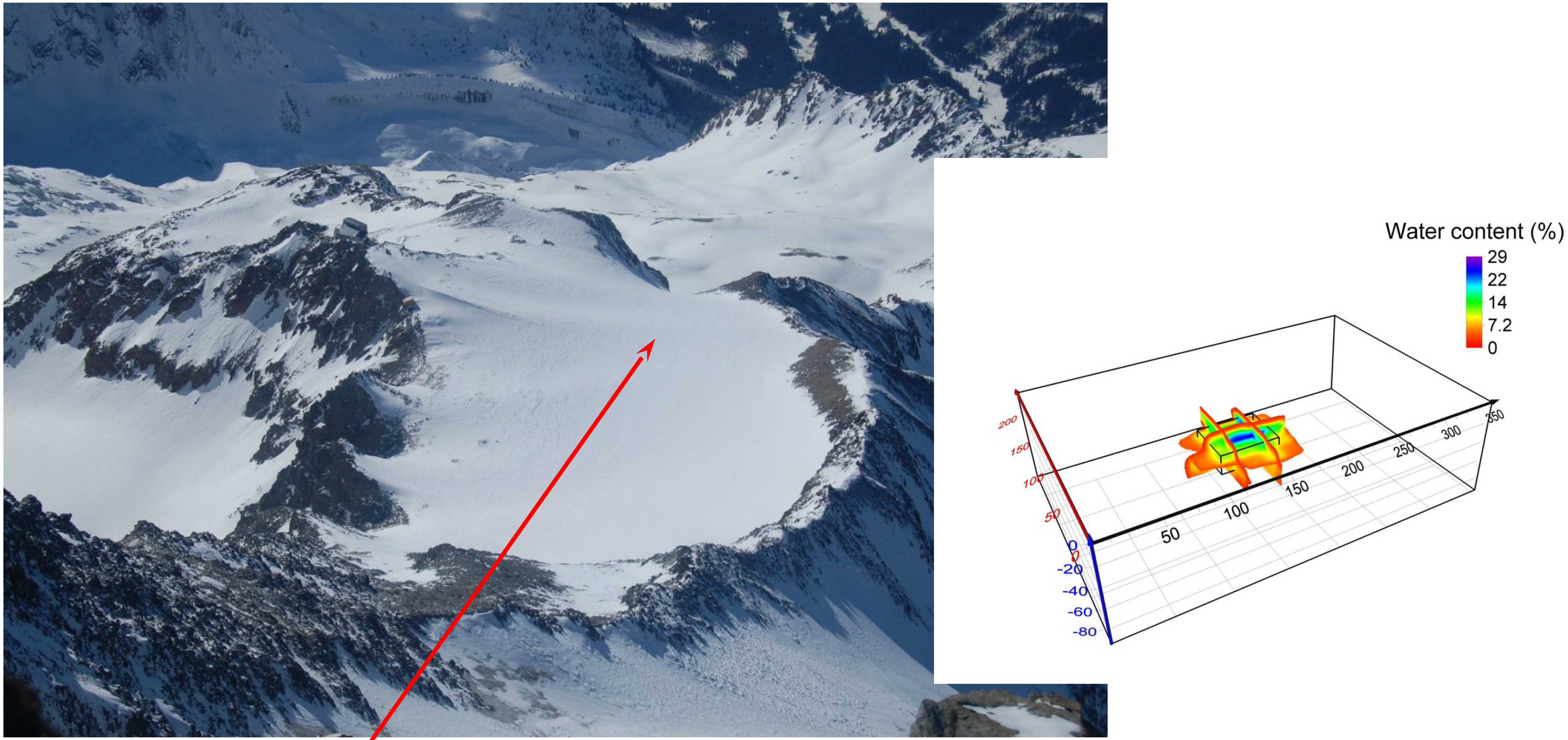
The radar measurements showed a zone (volume) with an anomaly.

Glaciological studies

In Sept 2009, geophysical survey using the Magnetic Resonance Imaging (LTHE, Grenoble)

@Vincent, LGGE

Glaciological studies



Water volume of 65 000 m³

Report presented to public authorities in March 2010

@Vincent, LGGE

Pressure measurements

20 hot-water drillings performed from 29 June to 8 July 2010

Confirm the presence of a cavity

and that

the cavity is under pressure!

Decisions

The hydrostatic pressure exceeded the ice pressure from the weight of the ice column

We could expect that the water contained in the glacier would be released suddenly

The public authorities were warned immediately (13 July, 2010)

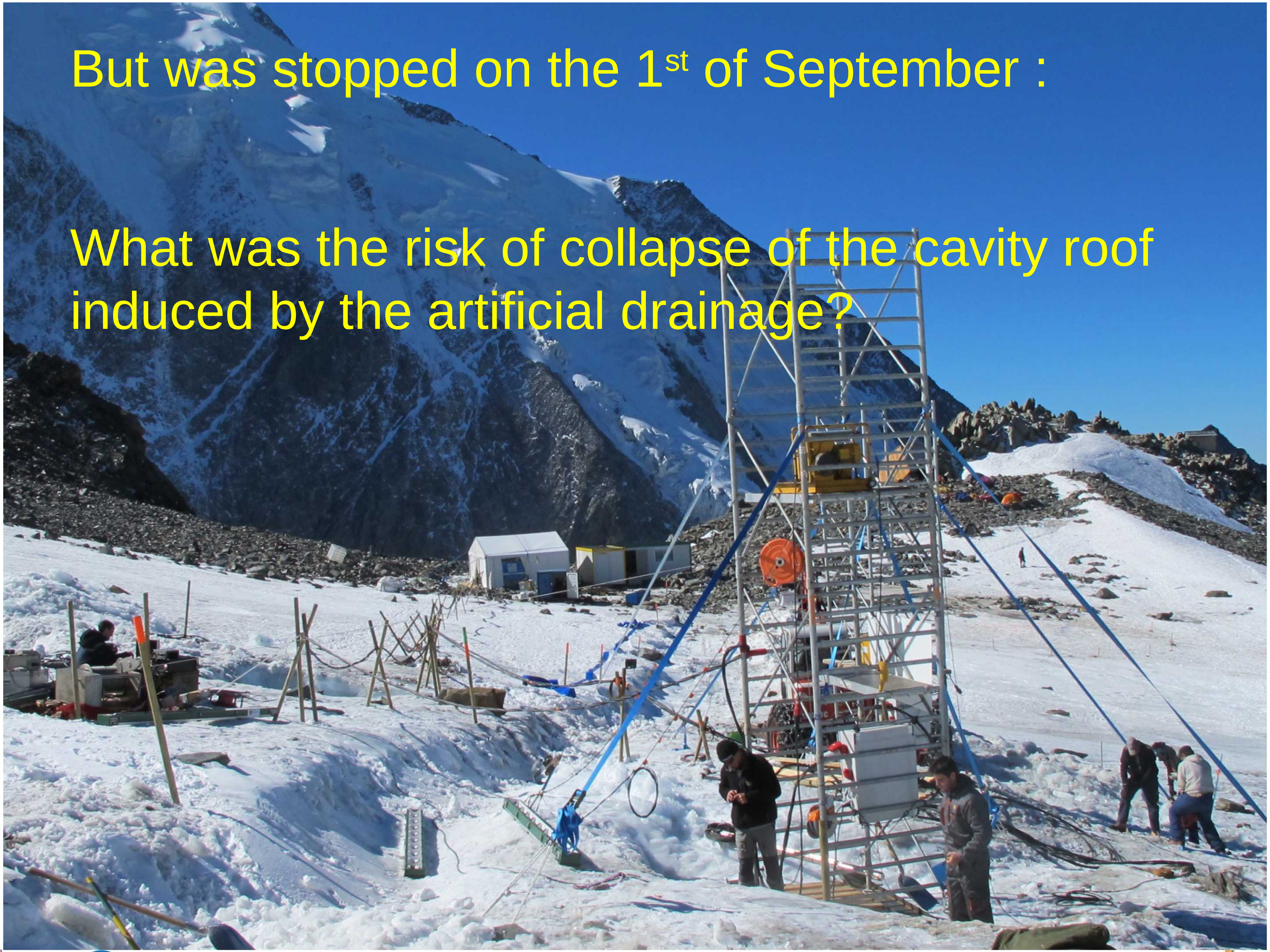
It was decided to drain the subglacial lake as soon as possible, because 3000 people were threatened in the valley.

A difficult field work

Drainage of the cavity

The artificial drainage started the 26th August 2010

A new risk ?



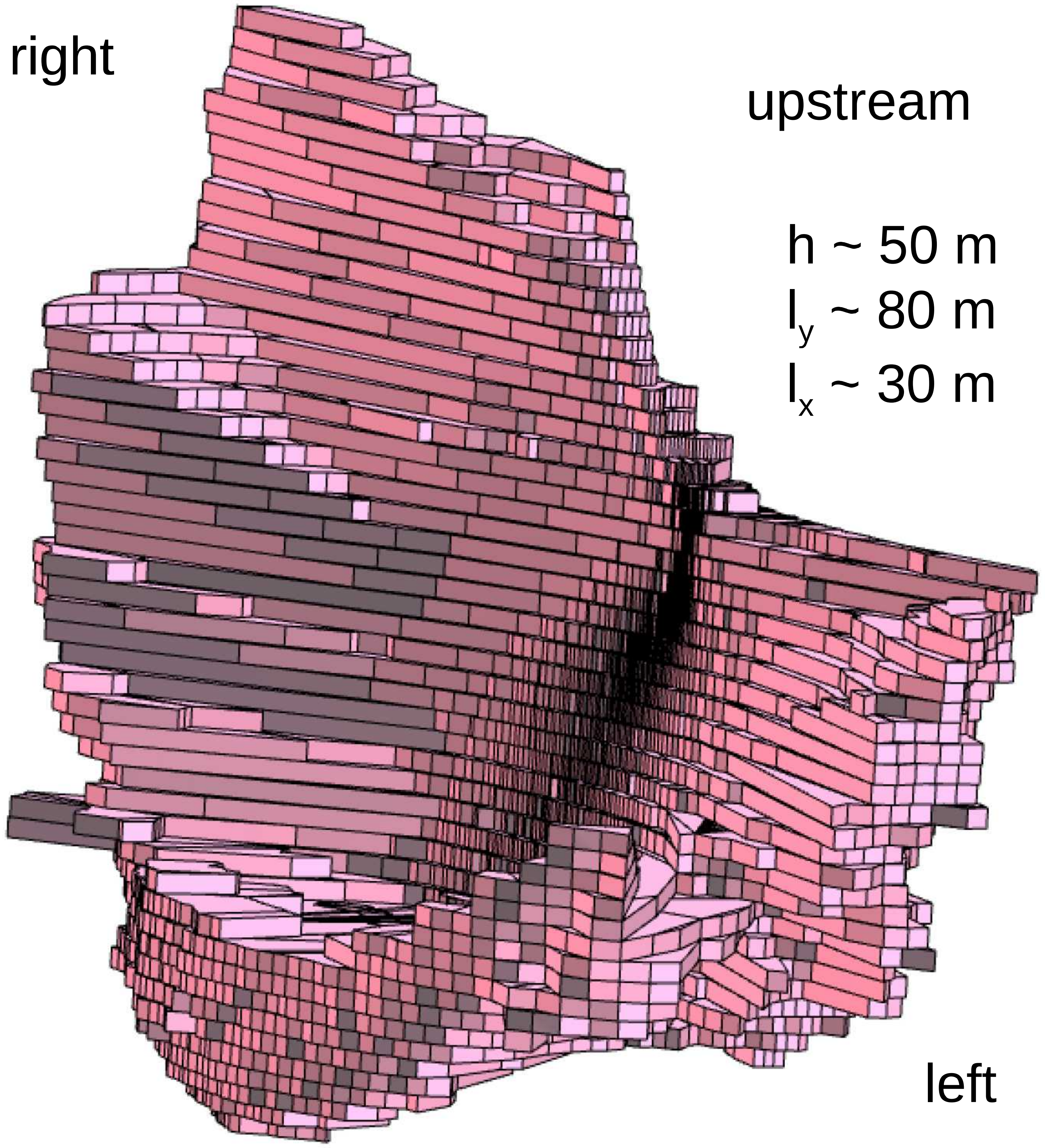
But was stopped on the 1st of September :

What was the risk of collapse of the cavity roof induced by the artificial drainage?

The 2010 cavity

Pumping of 47 700 m^3 from 25 August to 8 October 2010

Question (addressed end of August 2010):
What is the risk of break-up during the pumping phase?



$h \sim 50$ m
$l_y \sim 80$ m
$l_x \sim 30$ m

Time-line for investigations

Sonar data

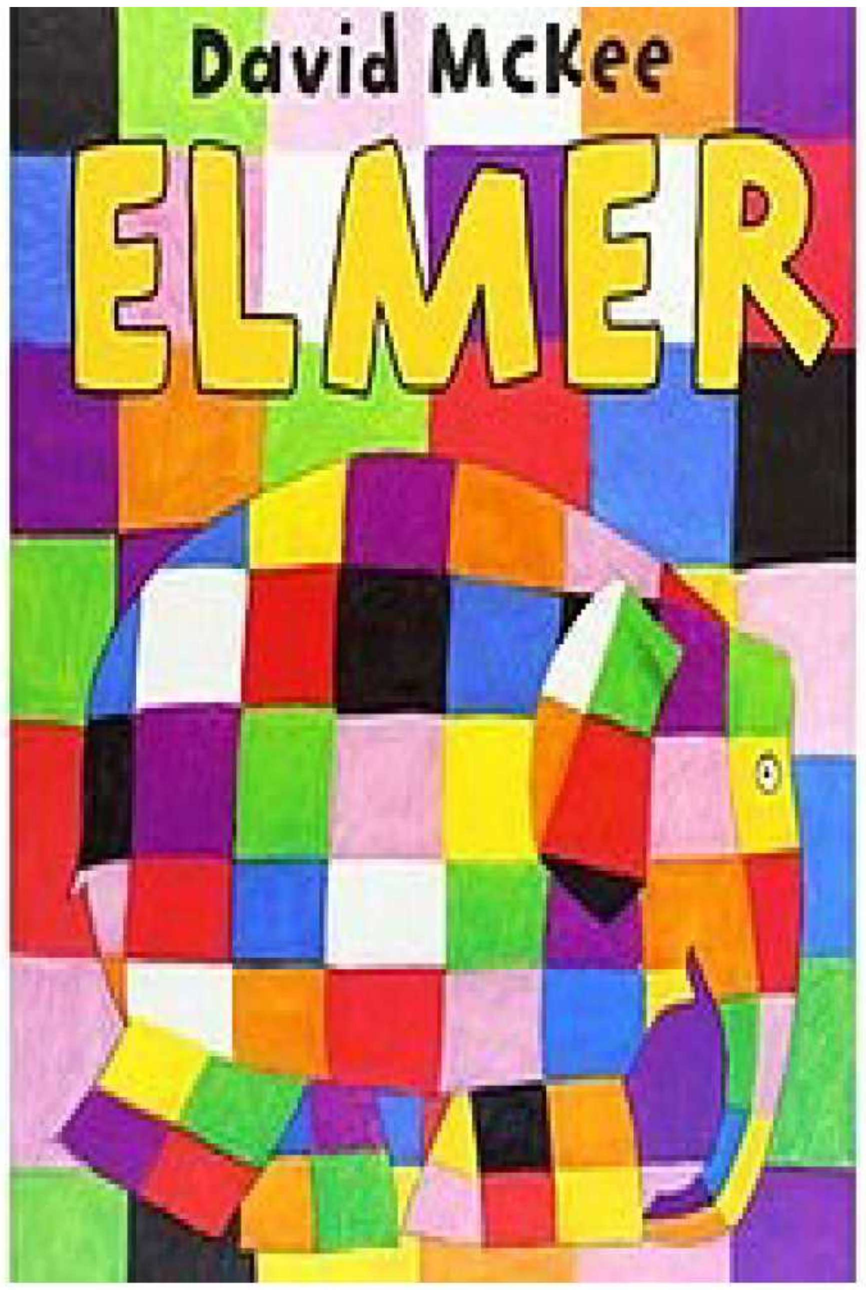
Septembre

D	L	M	E	J	V	S
			1	2	3	
				9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Meeting with the mayor of St Gervais

After all that, Elmer/Ice!

- Elmer
  - Elmer/Ice
- Open source
- Can do a lot, but….
- UNITS



Elmer Essentials

- Elmer install
- SIF
- Mesh
- Other data
- Results

Elmer Install

- Should be already installed on your computers
- If want to install on your personal computers (Mac/Linux only), go to http://elmerfem.org/elmerice/wiki/doku.php?id=compilation

Elmer Install

- Lots of folders
- Important ones:
  - DownloadDir/fem/src
  - DownloadDir/elmerice/Solvers
  - DownloadDir/elmerice/UserFunctions
  - DownloadDir/elmerice/Tests

SIF

- Solver Input File
- The instructions that tell Elmer what to do for a given simulation
- Can be long and complicated, but doesn’t need to be
- Mostly straightforward, but some syntax to learn

Mesh

- Elmer needs to be given a mesh to run on
  - i.e. a digital representation of the glacier
- Various ways of doing this
- Need to think about resolution, dimensions….

Other Data

- Elmer can load all kinds of other data
- Generally, easiest is to provide it as a text file or raster (netCDF or an ASCII format)
- Make sure in right co-ordinate system and units

Results

- Elmer outputs (useful) results in vtu format
- Best viewed and analysed in Paraview
  - Also open-source
  - Built on python

After all that, an example!

Construct a model of a realistic synthetic glacier (Synth Glacier)

- Step 1: Initialise the model
- Step 2: Grow Synth Glacier to a steady state
- Step 3: Perturb Synth Glacier

Data for ice-flow modelling

- Basics:
    - Bedrock Digital Elevation Model (DEM)
    - Surface DEM
    - A mesh domain
- More complex model = more inputs needed (usually)

Material:

To be listed when I know what I'm doing

Data: All generated internally !

SRC: Compute2DNodalGradient.F90, SyntSMB.F90

Step1: Synthetical_Glacier_BedDEM.py, Contour2geo.py, initialise_DEM.sif

Step2: steady_climat.sif, steady_climat_Stokes.sif

Step3: All.sif, Slip.sif, SMB.sif, TempDiagnostic.sif

Modelling Synth Glacier

- ✓ **Step 1** – Model initialisation

- ✓ **Step 2**
  - Growing Synth Glacier to a steady state

- ✓ **Step 3**
  - Perturbing Synth Glacier

Step 1: Work to do

- Create the mesh
- Import DEMs

Step 1: Create the mesh

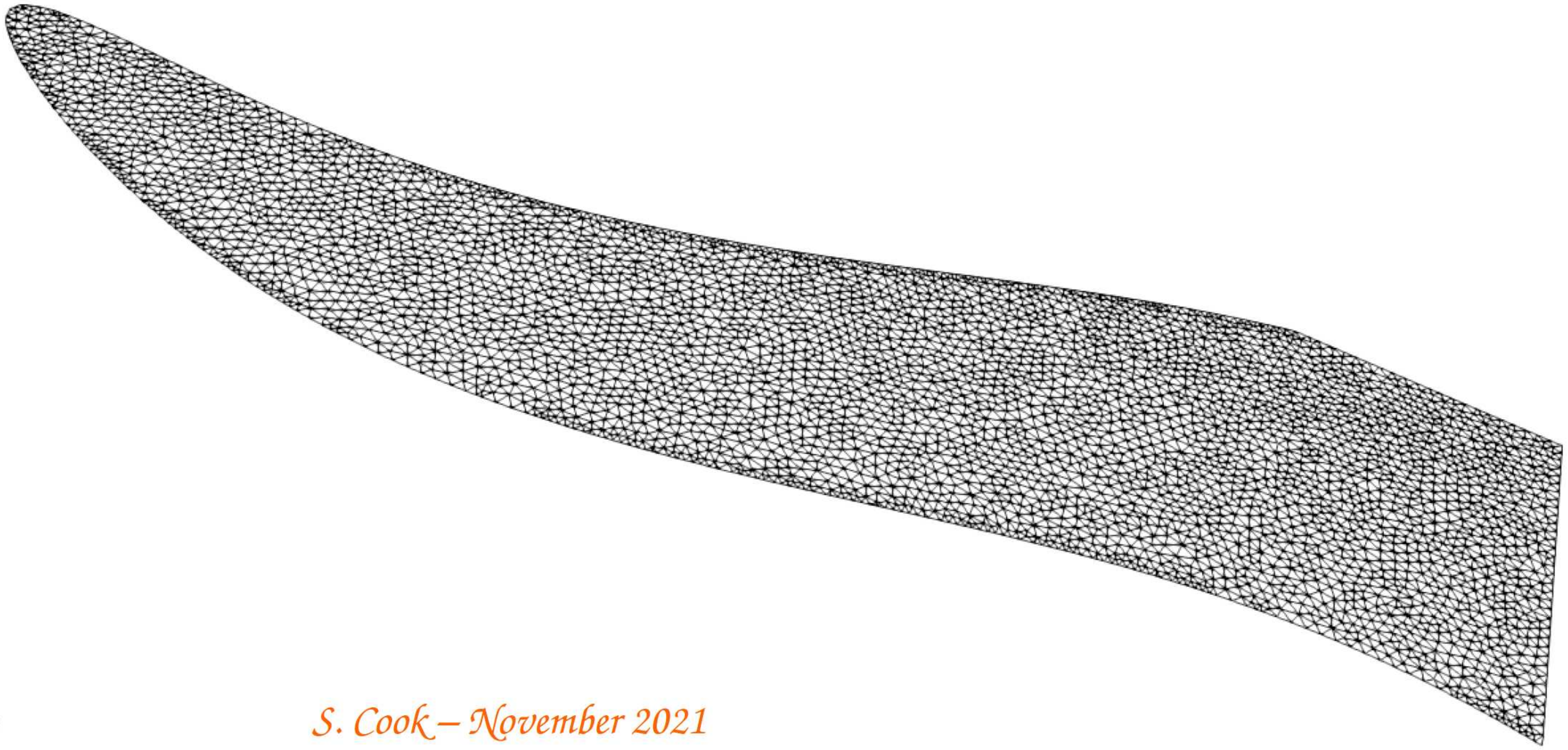
1/ Use python to run **Synthetical_Glacier_BedDEM.py** to create a contour outline

2/ Run **Contour2geo.py** to turn this into a .geo file (glacier footprint)

3/ **`gmsh`** to turn this into a .msh file (still the footprint)

4/ **ElmerGrid** to convert to Elmer format mesh

5/ This 2D footprint is then extruded by the model



Step 1: **Synthetical_Glacier_BedDEM.py**

The python script **Synthetical_Glacier_BedDEM.py** :

- Takes a load of modifiable parameters and uses them to construct a realistic bed DEM for a synthetic glacier
- Generates:
  1. A contour outline of the glacier footprint (SyntBed_Contour_bed1.dat)
  2. A bed DEM (SyntBed_DEM_bed1.nc)

Options :

- Open the script – lines 32-37 and 44-53 are all parameters you could modify
- We're not going to now, but you could

Run: **`python Synthetical_Glacier_BedDEM.py`**

Step 1: **Contour2geo.py**

The python script **Contour2geo.py** :

- Reads the point coordinates in the contour file
- Creates the **Mesh2d.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 (the mesh resolution)

Run:

```
python Contour2geo.py -r 100.0 -i SyntBed_Contour_bed1.dat -o Mesh2d.geo
```

Step 1: **GMSH** and **ElmerGrid**

```
> gmsh Mesh2d -1 -2
```

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

line commands:
`″-2"` performs 1D and 2D mesh generation and then exits

```
> ElmerGrid 14 2 Mesh2d.msh -autoclean
```

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

```
> ElmerGrid 14 5 Mesh2d.msh -autoclean
```

Converts a GMSH mesh (14) into a vtu file (5)
Use Paraview to visualise the mesh

Step 1: Create the mesh

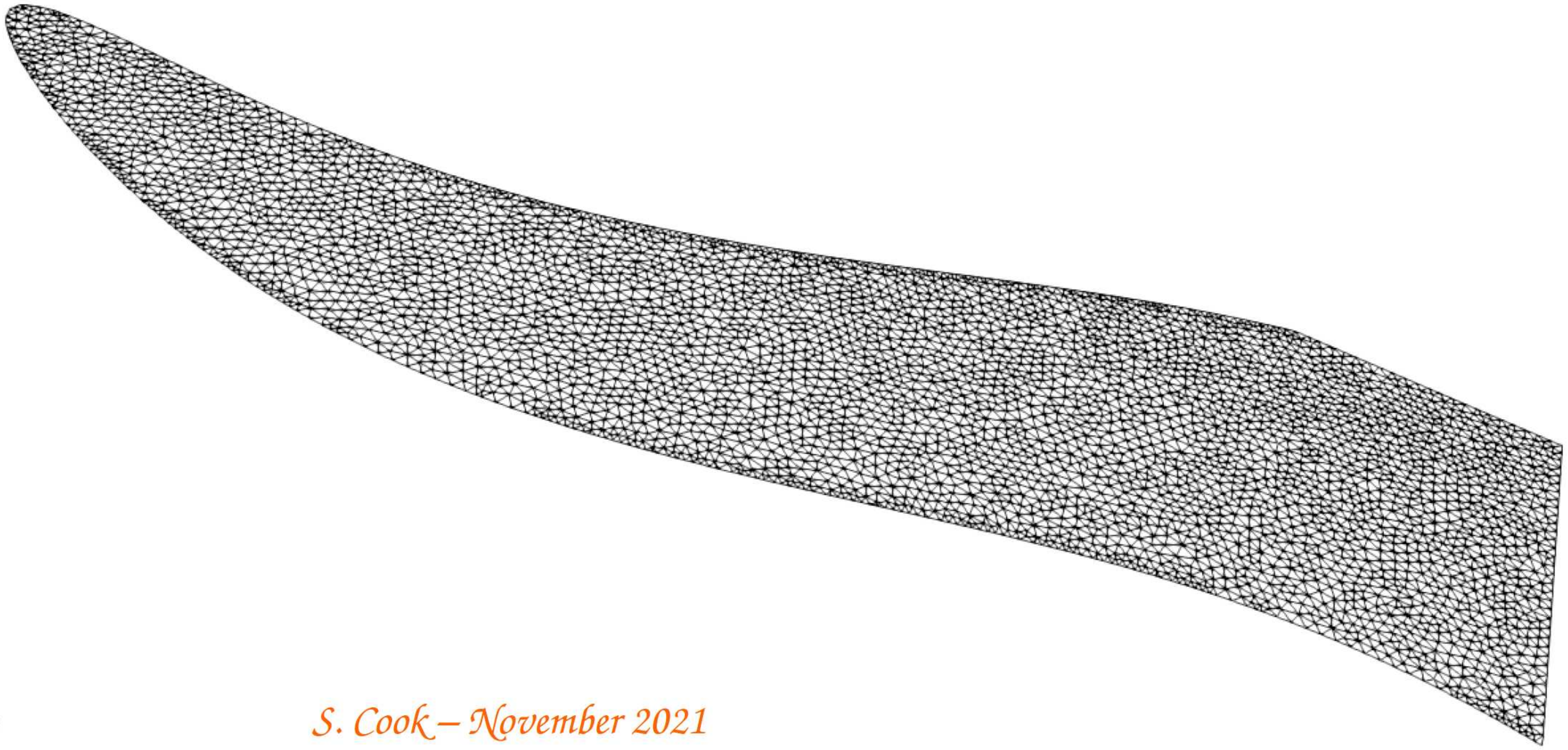
1/ Use python to run **Synthetical_Glacier_BedDEM.py** to create a contour outline

2/ Run **Contour2geo.py** to turn this into a .geo file (glacier footprint)

3/ **gmsh** to turn this into a .msh file (still the footprint)

4/ **ElmerGrid** to convert to Elmer format mesh

5/ This 2D footprint is then extruded by the model



Step 1: Work to do

- Create the mesh
- Import DEMs

Step 1: Import DEMs

- A very simple SIF: **initialise_DEM.sif**
- Internal extrusion

Step 1: SIFs

- Header and constants
    - Include files
- Simulation
- Bodies
- Solvers
- Equations
- Boundary conditions

Step 1: SIFs

- Not in this SIF:
  - Material
  - Initial conditions
  - Body forces

Step 1: Internal extrusion

Define the number of vertical layers to make your mesh 3D (Simulation section; see **Extrusion.sif**):

```
Simulation
  Coordinate System  =  Cartesian 3D
  Simulation Type = Steady

  Extruded Mesh Levels = Integer 5
  …
End
```

The second solver to be executed is the StructuredMeshMapper

```
Solver 2
  Equation = "MapCoordinate"
  Procedure = "StructuredMeshMapper" "StructuredMeshMapper"
  Active Coordinate = Integer 3
  Displacement Mode = Logical False
  Correct Surface = Logical True
  Minimum Height = Real #MinH
  ! Top and bottom surfaces defined from variables
  Top Surface Variable Name = String "surfDEM"
  Bottom Surface Variable Name = String "bedDEM"
End
```

3D problem, so the mesh moves in the z direction

zs = min(zs, bed+#MinH)

See next slide

Step 1: Internal extrusion

bedDEM and surfDEM (variable) must be declared in a solver (GridDataReader for example)

```
Exported Variable 1 = -dofs 1 "bedDEM”
Exported Variable 2 = -dofs 1 "surfDEM”
```

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

```
!Bed rock BC
Boundary Condition 2
  Bottom Surface = Equals bedDEM

  ...

End

! Upper Surface BC
Boundary Condition 3
  Top Surface = Equals surfDEM
End
```

Modelling Synth Glacier

- ✓ **Step 1** – Model initialisation

- ✓ **Step 2**
  - Growing Synth Glacier to a steady state

- ✓ **Step 3**
  - Perturbing Synth Glacier

Step 2: Work to do

- Grow the glacier
- Introduce Stokes
- Steady state?
- Parallelisation

Step 2: Grow the glacier

- Initial glacier geometry set up
- Now we need some ice
- And to do something with it….
- SIF gets a bit more complicated
  - New sections:
    - Initial Condition
    - Body Force
    - Material

Step 2: Grow the glacier

- Initial Condition

```
Initial Condition 1

End
```

Step 2: Grow the glacier

- Body Force

```
Body Force 1
  Flow BodyForce 1 = Real 0.0
  Flow BodyForce 2 = Real 0.0
  Flow BodyForce 3 = Real #gravity

! This should be in Body Force 2 but not working
! for solver executed on a boundary
  Zs = Variable bedDEM
    Real LUA "tx[0]+ MinH"
  ! should make it also dependent on SMB, i.e. allow zs to change if SMB>0
   Zs Condition = Variable "IcyMask","Mass Balance"
    Real LUA "IfThenElse((tx[0]< -0.5) and (tx[1] <= 0.0), 1.0, -1.0)"
End
```

Step 2: Grow the glacier

- Material

```
Material 1

! For the ice flow

  Density = Real #rhoi

  Viscosity Model = String "Glen"
  Viscosity = Real 1.0
  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

  ! or provide A

  !Set Arrhenius Factor = Logical
True
  !Arrhenius Factor = Real #A
```

```
  Min Zs = Variable bedDEM
    Real LUA "tx[0]+ MinH"
End
```

```
! Prefactor from Cuffey&Paterson
(2010) in MPa^{-3} a^{-1}
#A1 = 2.89165e-13*yearinsec*1.0e18
#A2 = 2.42736e-2*yearinsec*1.0e18

#Q1 = 60.0e3
#Q2 = 115.0e3
```

Step 2 – Grow the glacier

1/ Need to restart...

Initial conditions are set before the first solver is executed

Impossible then to initialize with another variable

This is then done by using a restart and specifying:

```
Restart Before Initial Conditions = Logical True
```

2/ (As an aside) Problem when a solver is called two time in the same sif...

Need to make a copy of the object file to avoid mixing the saved variables in the solver from two different calls:

```
cp $ELMER_HOME/share/elmersolver/lib/FreeSurfaceSolver.so
MyFreeSurfaceSolver
```

Use a different call in the sif file for Zb:

```
Procedure = "./MyFreeSurfaceSolver" "FreeSurfaceSolver"
```

Step 2 – Grow the glacier

- We currently have no ice
  - So cannot do a **diagnostic** simulation (unchanging geometry and boundary conditions)
  - We will run a **prognostic** simulation (changing geometry and boundary conditions)

In a **prognostic** simulation, need to:

  - Add the FreeSurface solver
  - Add one body per FS (new Initial Condition and Equation Sections)
  - Modifications in the Simulation and Boundary Condition Sections

Step 2 – Grow the glacier

The simulation Section has to be modified:

```
Simulation Type = Transient

Timestepping Method = “bdf”
BDF Order = 1
Output Intervals = 1
Timestep Intervals = 50
Timestep Sizes = #10.0/365.25

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

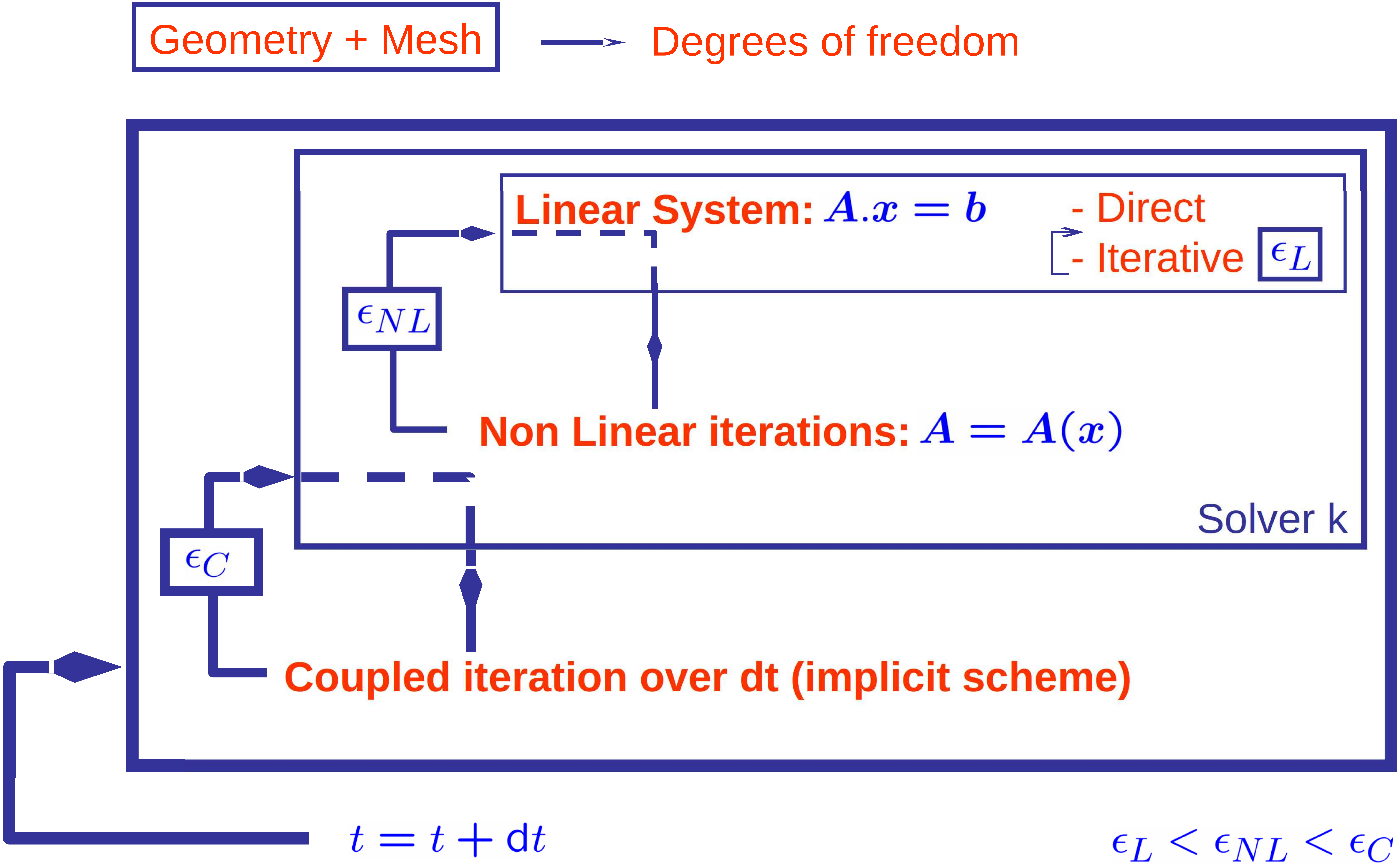
→ Backward Differences Formulae

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

```
Restart File = "synt_DEM.result”
Restart Position = 0
Restart Time = Real 0.0
Restart Before Initial Conditions = Logical True
```

→ We need a restart to have the Zs and bedDEM variables for the initial condition of Zs (and Zb, if we were setting one)

Step 2 – Sketch of a transient simulation



Geometry + Mesh → Degrees of freedom

Linear System: $\boldsymbol{A}.\boldsymbol{x} = \boldsymbol{b}$

- Direct
- Iterative ϵ_L

ϵ_{NL}

Non Linear iterations: $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{x})$

Solver k

ϵ_C

Coupled iteration over dt (implicit scheme)

$t = t + \mathrm{d}t$

$\epsilon_L < \epsilon_{NL} < \epsilon_C$

Step 2 – Grow the glacier

The free surface solver only applies to boundary 3 (upper surface)

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

```
Body 2
  name = “surface”
  Equation = 2
  Body Force = 2
  Material = 1
  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.

Need to say in BC3 that this is the location of Body 2:

```
Boundary Condition 3
      Body Id = 2
    …
End
```

Step 2 – Grow the glacier

Equation 2:

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

Initial Condition 2: (say that $z_s(x,0)$ is given by the surface DEM)

```
Initial Condition 2
  Zs = Equals surfDEM
  RefZs = Equals surfDEM

  IcyMask = Real 1.0
End
```

Step 2 – Grow the glacier

Body Force 2:

```
Body Force 2
  Zs Accumulation Flux 1 = Real 0.0e0
  Zs Accumulation Flux 2 = Real 0.0e0
  Zs Accumulation Flux 3 = Equals “Mass Balance“

  ! surface slope norm
  slope = Variable "dzs 1", "dzs 2"
  REAL LUA "math.sqrt(tx[0]*tx[0]+tx[1]*tx[1])"

  ! mask mass balance with surface slope
  Mass Balance = Variable "Mass Balance Ini","slope"
  Real LUA "IfThenElse(tx[0]>0,IfThenElse(tx[1]< 1.2, tx[0], 0.0),tx[0])"
End
```

Step 2 – Grow the glacier

```
Solver 7
Equation =  String "Free Surface Evolution"
  Procedure = "FreeSurfaceSolver" "FreeSurfaceSolver"

  Variable = "Zs"
  Variable DOFs = 1

 ! calculate dz/dt (better than from mesh velocity in case of steady-state iterations)
  Calculate Velocity = Logical True

 ! Apply internal limiters
  Apply Dirichlet = Logical true

 ! Steb method
  Stabilization Method = Stabilized

 ! linear settings
  Linear System Solver = Iterative
  Linear System Iterative Method = BiCGStab
  Linear System Max Iterations  = 1000
  Linear System Preconditioning = ILU0
  Linear System Convergence Tolerance = 1.0e-10

 ! non-linear settings
  Nonlinear System Max Iterations = 20 ! variational inequality needs more than one round
  Nonlinear System Min Iterations = 2
  Nonlinear System Convergence Tolerance = 1.0e-8

  Steady State Convergence Tolerance = 1.0e-6

 ! loads also takes into account dirichlet conditions
 ! to compute residual flux
  calculate loads = Logical True

  Exported Variable 1 = -nooutput "Zs Residual"
  Exported Variable 2 =  "Ref Zs"
End
```

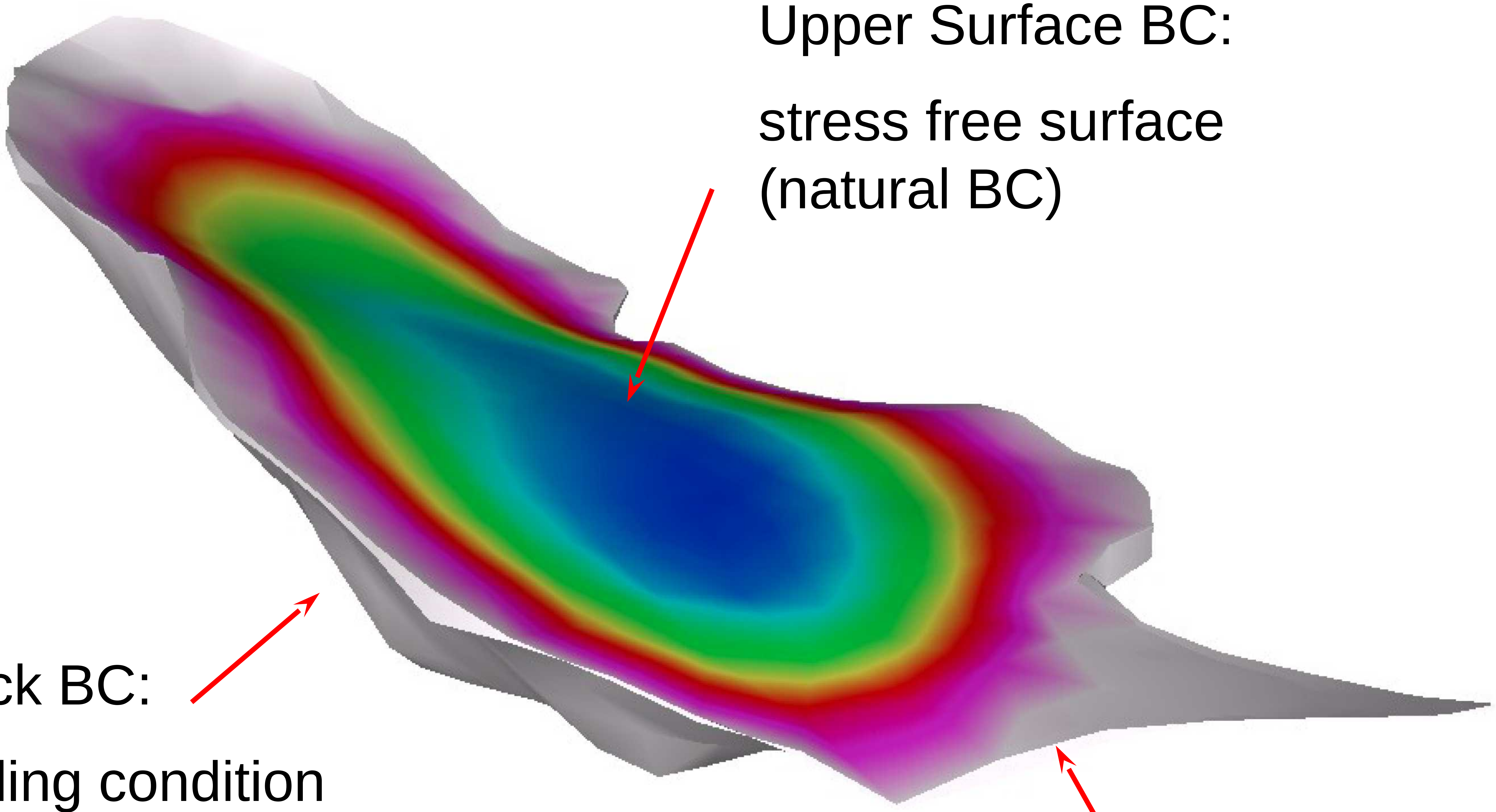
Step 2 – Introduce Stokes

- Next, we want to get that ice moving
- This means solving the Stokes equations

Step 2 – Introduce Stokes

Solve only the Stokes equation in a diagnostic way

3 boundary conditions



Upper Surface BC:

stress free surface (natural BC)

Bedrock BC:

No sliding condition

Lateral BC:

zero horizontal velocities

Step 2 – Introduce Stokes

```
! lateral side of the glacier
Boundary Condition 1
  Target Boundaries = 1
  Velocity 1 = real 0.0
  Velocity 2 = real 0.0
End

! Bedrock
Boundary Condition 2
  Bottom Surface = Equals BedDEM
  Velocity 1 = Real 0.0
  Velocity 2 = Real 0.0
  Velocity 3 = Real 0.0
End

! Upper Surface
Boundary Condition 3
  Top Surface = Equals ZsDEM
End
```

Null horizontal velocities

No sliding

Natural BC,
nothing to do!

Step 2 – Introduce Stokes

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

Natural BC

```
! lateral side of the glacier
Boundary Condition 1
  Target Boundaries = 1
  Velocity 1 = real 0.0
  Velocity 2 = real 0.0
  Velocity 3 = real 0.0
End
```

zero velocity

Can make velocities normal-tangential, however….

Step 2 – Introduce Stokes

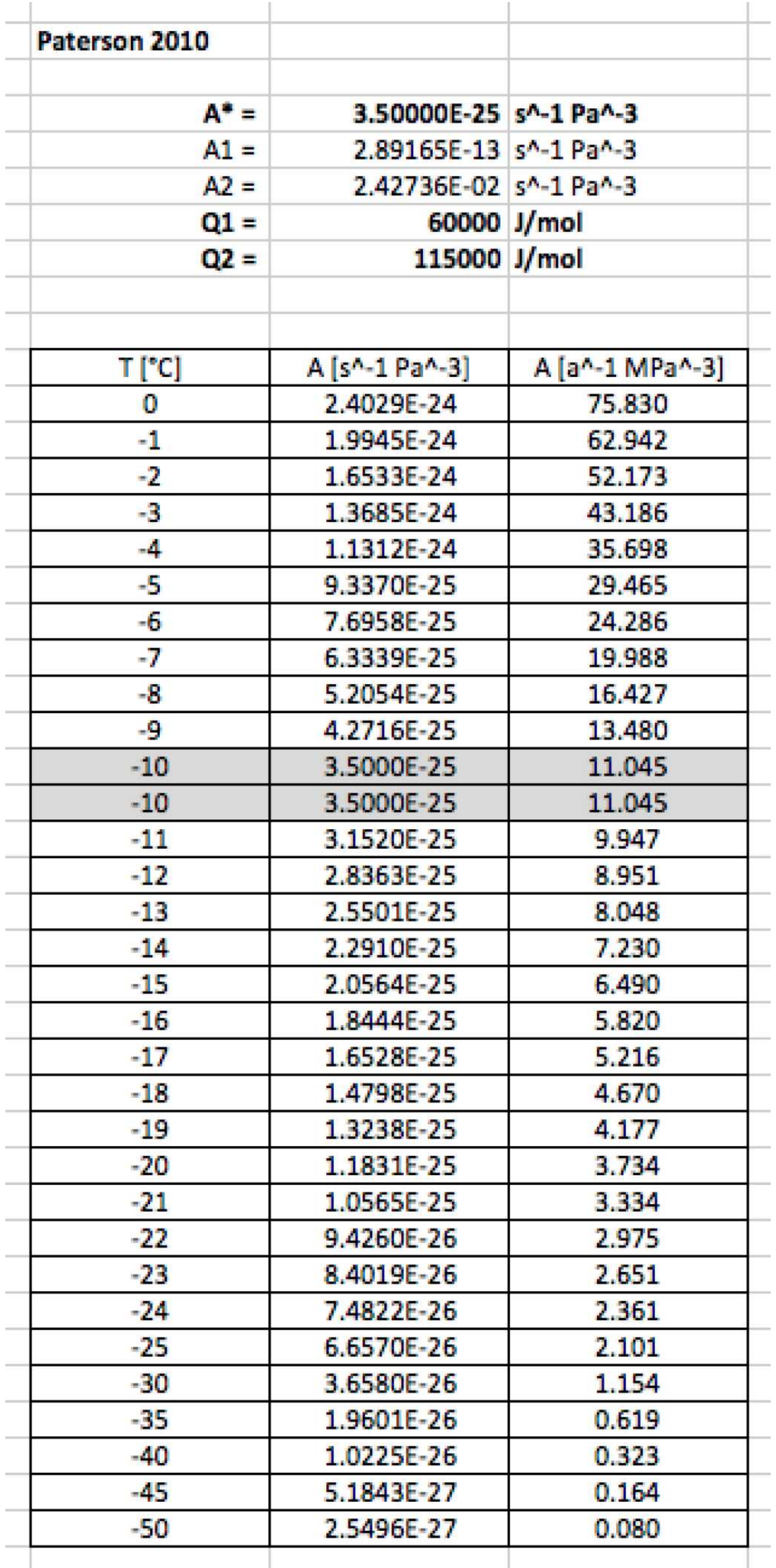
$$D_{ij} = A\tau_e^{n-1}S_{ij} \quad ; \quad S_{ij} = A^{-1/n}I_{D_2}^{(1-n)/n}D_{ij}$$

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

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

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

Cuffey and Paterson (2010)



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

assume a constant temperature of -1°C

Step 2 – Introduce Stokes

```
Material 1

! For the ice flow

  Density = Real #rhoi

  Viscosity Model = String "Glen"
  Viscosity = Real 1.0
  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

  ! or provide A

  !Set Arrhenius Factor = Logical
True
  !Arrhenius Factor = Real #A
```

```
  Min Zs = Variable bedDEM
    Real LUA "tx[0]+ MinH"
End
```

```
! Prefactor from Cuffey&Paterson
(2010) in MPa^{-3} a^{-1}
#A1 = 2.89165e-13*yearinsec*1.0e18
#A2 = 2.42736e-2*yearinsec*1.0e18
#Q1 = 60.0e3
#Q2 = 115.0e3
```

Step 2 – Introduce Stokes

```
Solver 4
  Equation = "Stokes-Vec"
  Procedure = "IncompressibleNSVec"
"IncompressibleNSSolver"
  Stokes Flow = logical true
  !Exec Solver = "Never"

  Div-Curl Discretization = Logical
False

  Stabilization Method = String
Stabilized

  !linear settings:

  !------------------------------

  include linsys/BiCGStab.sif

!Non-linear iteration settings:
  !------------------------------
  Nonlinear System Max Iterations =
20
  Nonlinear System Convergence
Tolerance  = 1.0e-5
  Nonlinear System Newton After
Iterations = 4
  Nonlinear System Newton After
Tolerance = 1.0e-4
```

```
  Nonlinear System Reset Newton = Logical True

  ! make it safe abort if non-linear diverge
  Nonlinear System Abort Not Converged =
Logical True

  ! Convergence on timelevel (not required
here)
  !---------------------------------------------
--
  Steady State Convergence Tolerance = Real
1.0e-3

  Relative Integration Order = -1
  !Number of Integration Points = Integer 44 !
21, 28, 44, 64, ...

  ! 1st iteration viscosity is constant

  Constant-Viscosity Start = Logical False
End
```

Step 2 – Introduce Stokes

```
BiCGSTab.sif:

  Linear System Solver = Iterative

  Linear System Iterative Method = BiCGStab

  Linear System Max Iterations  = 1000

  Linear System Preconditioning = ILU0

  Linear System Convergence Tolerance = 1.0e-08


  Linear System Residual Output = 100
```

Step 2 – Introduce Stokes

If you want to use Newton linearisation for the non-linear iterations, don't forget to reset the conditions used to move from Picard to Newton at each time step, by adding:

```
Solver 3
  Equation = "Navier-Stokes"
  Nonlinear System Reset Newton = Logical True

End
```

Step 2 – Steady state?

- Now we’ve got everything set up, we want to see what sort of glacier would exist in our conditions
    - Defined terrain
    - Defined SMB
    - But no ice yet
- This will be a pseudo-steady-state run, as the geometry will evolve (to start with), but the forcing and boundary conditions are fixed

Step 2 – Steady state?

- But, Stokes takes a **long** time to solve

- Run the two SIFs (steady_climat.sif and steady_climate_Stokes.sif) for a few minutes each

- Notice anything?

- We can help here with **parallelisation**

Step 2 – Parallelisation

Parallelisation is easy (hooray!).

You just need a partitioned mesh (here 2 partitions):

```
> ElmerGrid 2 2 Mesh2d –autoclean –partdual -metiskway 2
```

And to create a file (`ELMERSOLVER_STARTINFO)` which contains the name of the sif file on its first line,

and then

```
> mpirun –n 2 ElmerSolver_mpi
```

But, to reach steady state, will still take (11ish) hours of runtime…

And we also need to go back and run `initialise_DEM.sif` in parallel so we can actually restart from it.

Step 2 – Results

- Elmer produces several kinds of results files:
    - .result files – used for restarts
    - .vtu files – used for visualisation and post-processing – view in Paraview
    - Various output solvers can also output additional formats and files
- Have provided a steady-state .result set and a .vtu example – open it in Paraview!

Modelling Synth Glacier

✓ **Step 1** – Model initialisation

✓ **Step 2**
  - Growing Synth Glacier to a steady state

✓ **Step 3**
  - Perturbing Synth Glacier

Step 3 – Work to do

- Basal slip
- Changing SMB
- Temperature field
- All at once

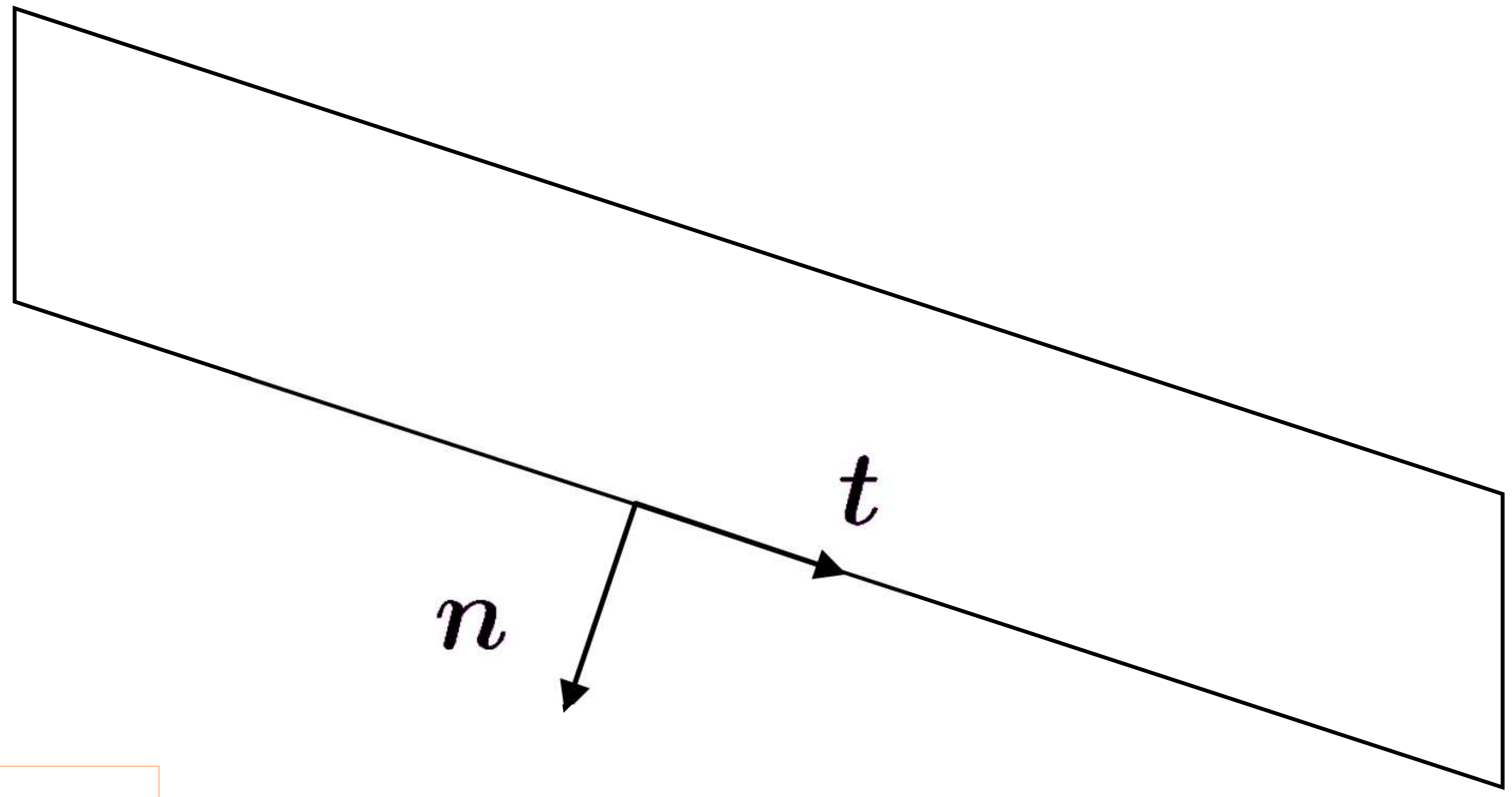
Step 3 – Basal slip

Friction law in Elmer (2d case illustrated):

$$C_i u_i = \sigma_{ij} n_j \;\; (i = 1, 2)$$

$$\rightarrow \quad C_t u_t = \sigma_{nt} \;;\; C_n u_n = \sigma_{nn}$$

where n is the surface normal vector



```
! Bedrock BC
Boundary Condition 2
  . . .
  Normal-Tangential Velocity = Logical True

  Velocity 1 = Real 0.0e0
  Slip Coefficient 2 = Real 0.1
  Slip Coefficient 3 = Real 0.1
End
```

How to evaluate the Slip Coefficient ?

Step 3 – Basal slip

- Need to change basal BC – replace ‘no_slip.sif’ with ‘slip_linear.sif’

```
  Normal-Tangential Velocity = Logical True
  Mass Consistent Normals = Logical True

  Velocity 1 = Real 0.0e0
  Slip Coefficient 2 =  Real #slc
  Slip Coefficient 3 =  Real #slc

 ! set no sliding if H< 2m
  Velocity 2 = Real 0.0e0
  Velocity 3 = Real 0.0e0
  Velocity 2 Condition = Variable "thickness"
    Real LUA "IfThenElse(tx[0]< 2.0, 1.0, -1.0)"
  Velocity 3 Condition = Variable "thickness"
    Real LUA "IfThenElse(tx[0]< 2.0, 1.0, -1.0)"
```

Step 3 – Basal slip

- This is a very simple linear slip law
- More complicated ones in `USF_Sliding.F90`
- Even here, glacier starts to speed up
- Sliding is important for glacier motion, but still not fully understood
- Easy to implement in models, though

Step 3 – Basal slip

- Which slip law to use?
    - Linear, Weertman, Budd, Tsai, Coulomb…?
    - Depends on what you're modelling
    - Scale, processes, timestep….

Step 3 – Changing SMB

- Need to change SMB parameters
- Physical_Parameters_New.IN
    - GlacierHead: SMB at top of glacier
    - GlacierHeadElevation: elevation of top of glacier
    - SMBExponent: how quickly SMB changes with altitude
    - SMBELA: the elevation of the equilibrium line altitude (ELA)

-Try increasing the ELA from 3700 to 3800 to start with

Step 3 – Changing SMB

- The glacier already retreats quite a long way

- Play around with the other parameters a bit to get a sense of how they matter

- **More generally, models often have several parameters – it’s important to get them right**

Step 3 – Temperature

- Need to add quite a lot – temperature is a bit complicated

- Temperature solver is also quite **numerically unstable** (especially in transient)

    - Can go back to square one to set up a converged Stokes-temperature field

    - Or can cheat a bit

```
Temperature Passive = Variable "Thickness","slope"
  Real LUA "IfThenElse((tx[0] <= 50) or (tx[1] > 1.2), 1.0, -1.0)"
```

    - Need to be careful with this, though

    - But best idea is to restart a real steady-state, diagnostic simulation from our transient glacier-growing run

Step 3 – Temperature

- New in the SIF:

```
Initial Condition 1
  !Temperature = Real 271.0
End

Body Force 1
  ...
  Temperature Volume Source = Equals W ! The volumetric heat source
End

Boundary Condition 2
  …
  !-------------------
  ! geothermal heatflux
  !--------------------
  Temperature Flux BC = Logical True
  Temperature Heat Flux = Real $56.05E-03*yearinsec*1.0E-6
  !-------------------
  ! frictional heat
  !--------------------
  Temperature Load = Variable Velocity 1
    Real Procedure  "ElmerIceUSF" "getFrictionLoads"
End
```

Setting an initial condition can help
Convergence (not needed here).

Step 3 – Temperature

```
Boundary Condition 3
  …
  Temperature = Real 273.0
End

Material 1
  ...
  ! the heat capacity as a MATC function of temperature itself
  !-----------------------------------------------------------
  Temperature Heat Capacity = Variable Temperature
    Real MATC "capacity(tx)*yearinsec^2"
  ! the heat conductivity as a MATC function of temperature itself
  !--------------------------------------------------------------
  Temperature Heat Conductivity = Variable Temperature
    Real MATC "conductivity(tx)*yearinsec*1.0E-06"
  ! Upper limit - pressure melting point
  !  as a MATC function of the pressure (what else?)
  !-----------------------------------------------
  Temperature Upper Limit = Variable Pressure
    Real MATC "pressuremeltingpoint(tx)"
  ! lower limit (to be save) as 0 K
  !--------------------------------
  Temperature Lower Limit = Real 0.0
End
```

Step 3 – Temperature

```
Solver 5
  Equation = DeformationalHeat
  Variable = W
  Variable DOFs = 1
  procedure =  "ElmerIceSolvers" "DeformationalHeatSolver"
  Linear System Solver = direct
  Linear System direct Method = umfpack
End
```

Step 3 – Temperature

```
Solver 6
  Equation = String "Homologous Temperature Equation"
  Procedure =  File "ElmerIceSolvers" "TemperateIceSolver"
  !------------------------------------------------------------
  Loop While Unconstrained Nodes = Logical True
  Variable = String "Temperature"
  Variable DOFs = 1
  Linear System Solver = "Iterative"
  Linear System Iterative Method = "BiCGStab"
  Linear System Max Iterations = 1000
  Linear System Convergence Tolerance = 1.0E-07
  Linear System Abort Not Converged = True
  Linear System Preconditioning = "ILU4"
  Linear System Residual Output = 1
  Steady State Convergence Tolerance = 1.0E-04
  Nonlinear System Convergence Tolerance = 1.0E-05
  Nonlinear System Max Iterations = 50
  Nonlinear System Relaxation Factor = Real 9.999E-01
  ! uses the contact algorithm (aka Dirichlet algorithm)
  !-----------------------------------------------------
  Apply Dirichlet = Logical True
  Stabilize = True
  ! those two variables are needed in order to store
  ! the relative or homologous temperature as well
  ! as the residual
  !-------------------------------------------------
  Exported Variable 1 = String "Temperature Homologous"
  Exported Variable 1 DOFs = 1
  Exported Variable 2 = String "Temperature Residual"
  Exported Variable 2 DOFs = 1
End
```

This is a really important option

Step 3 – Temperature

- Solving for temperature can also **greatly** increase model runtime (hooray!)
- Try running All.sif, which includes everything this step
    - It’s not fast….
    - Notice all the TemperateIceSolver output….
    - Trying to do much at once tends to break things
    - What do you actually need from the simulation?

Next steps

- There are loads of other solvers you can add
- Each should have an associated test in elmerice/Tests
    - The SIF for each test will show you how to set things up
    - There is also documentation for each Elmer/Ice solver and user function
        - elmerice/Solvers/Documentation
        - elmerice/USFs/Documentation

Next steps

- If you're interested in modelling mountain glaciers, this workflow is a really nice starting point:

https://gricad-gitlab.univ-grenoble-alpes.fr/maured/mountain-glacier-tutorial

- This is the more detailed version if you're interested in the development side:

https://gitlab.com/damien.maure/internship-workflow

Next steps

- Any questions, you’ve got this course

- And the Elmer/Ice forums

  - https://elmerfem.org/forum/viewforum.php?f=21

- If your problem is with a specific solver, try emailing whoever wrote it

- Good luck!