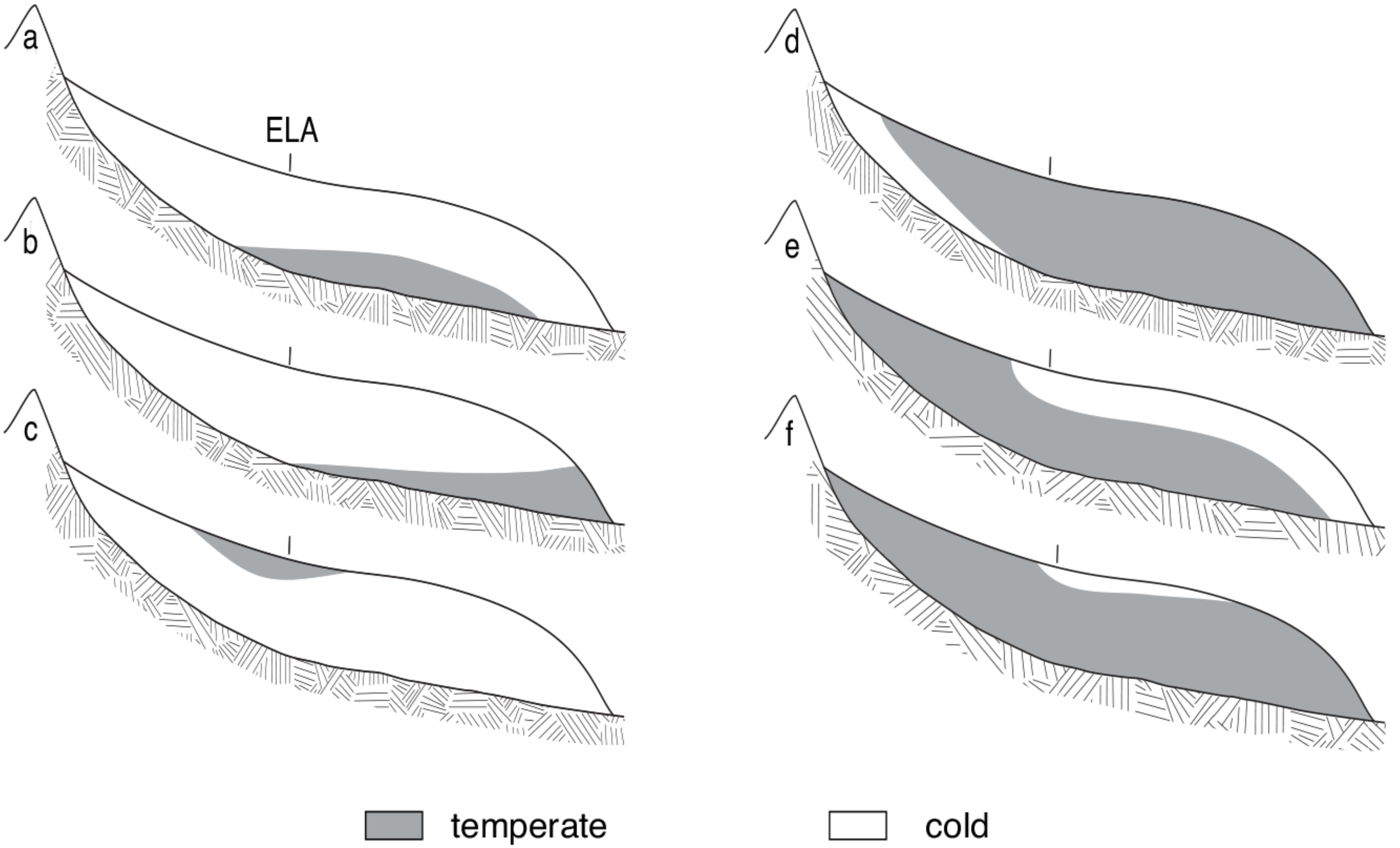
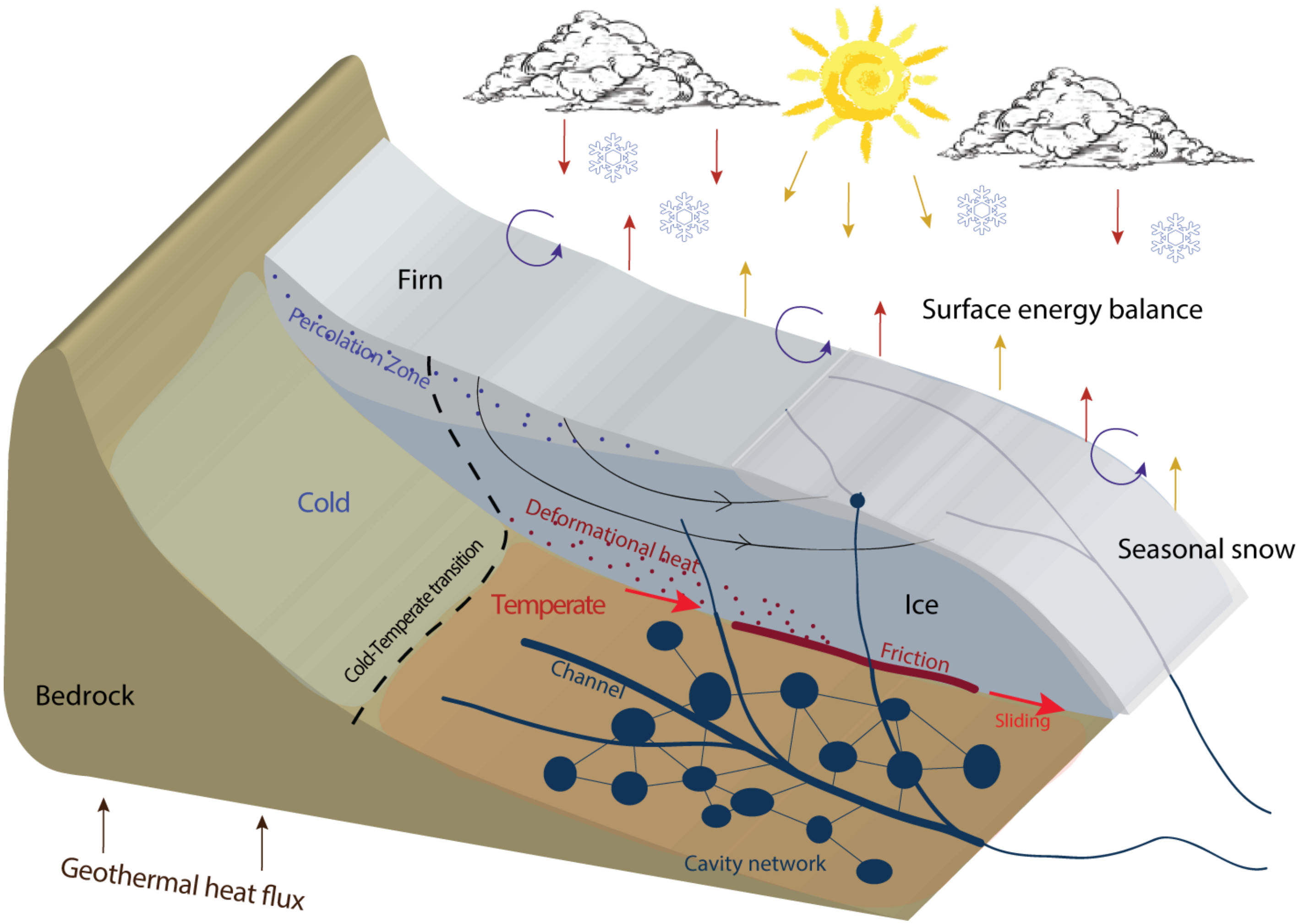
Modeling glacier thermal regime with Elmer/Ice

Adrien Gilbert – University of Oslo - Norway

Glacier thermal regime



Glacier thermal regime



Modeling approach

Enthalpy method :

$$\rho\left(\frac{\partial H}{\partial t} + \vec{v}\cdot\nabla H\right) = \nabla(\kappa\nabla H) + tr(\sigma\dot{\epsilon}) + Q_{lat}$$

$$H(T,\omega) = \begin{cases} \int_{T_0}^{T} C_p(T)\,dT, & H < Hf(p) \\ \int_{T_0}^{T_m(p)} C_p(T)\,dT + \omega L, & H \geq H_f(p) \end{cases}$$

Water Content (ω)

Temperature (T)

Heat capacity ($C_p(T)$)

Enthalpy of fusion ($Hf(p)$)

Latent heat of fusion (L)

Thermal conductivity : strongly dependent on density (k)

$$\kappa = \begin{cases} \frac{k(\rho,T)}{C_p(T)}, & H < Hf(p) \\ \kappa_0 & H \geq H_f(p) \end{cases}$$

Moisture diffusivity (κ_0)

Take into account water in temperate ice
No boundary condition for CTS

Modeling approach : Boundary condition

Surface :

- Surface temperature imposed by the surface energy balance
- Surface melting imposed by the surface energy balance
    - ➢ Deal with water percolation and refreezing

Bottom :

- Heat flux
- Frictional heating

UiO : University of Oslo

CSC

Modeling approach : Boundary condition

Higher level of complexity :
(full physical model)

Couple Elmer/Ice with a snow model

- Need every kind of meteorological data
- Very small time step

Gilbert et al., jgr 2012 (partially coupled)

Intermediate level of complexity :
(semi-physical model)

Compute density with the porous solver
Compute surface melting
Compute water percolation and refreezing
Surface temperature from air temperature

- Daily time step for the thermal part
- Need only daily air temperature

Gilbert et al., jgr 2014 and grl 2015

Simple model :

Simple firn thickness model
Compute released latent heat from annual melting
Surface temperature from seasonal temperature

- 6 month time step !

Not validate : built for this course !!

Simple model : Boundary condition

Mass balance = degree day model

Firn thickness :

$$H_{firn}(t+dt) = H_{firn}(t) + \left(m_b - H_{firn} \times a\right)dt$$

If $H_{firn}(t+dt) < 0$ *then* $H_{firn}(t+dt) = 0$

Surface enthalpy : 6 month timestep

$$H_s = H_s(T_s, \omega_s = 0)$$

$$T_s = T_{mean} + \frac{dT}{dz}\left(z - z_{ref}\right) \pm \Delta T/2 - constant$$

Percolation and refreezing :

Surface melting from mass balance model
Refreezing in summer as a function of temperature and density

Three climatic parameters :

Mean air temperature at z_{ref} : T_{mean}
Mean precipitation : *Precip*
Seasonal variability : ΔT

$$T_{day} = T_{mean} + \Delta T \sin\left(\frac{2\Pi day}{365}\right)$$

AruCo Lake twin glacier collapse

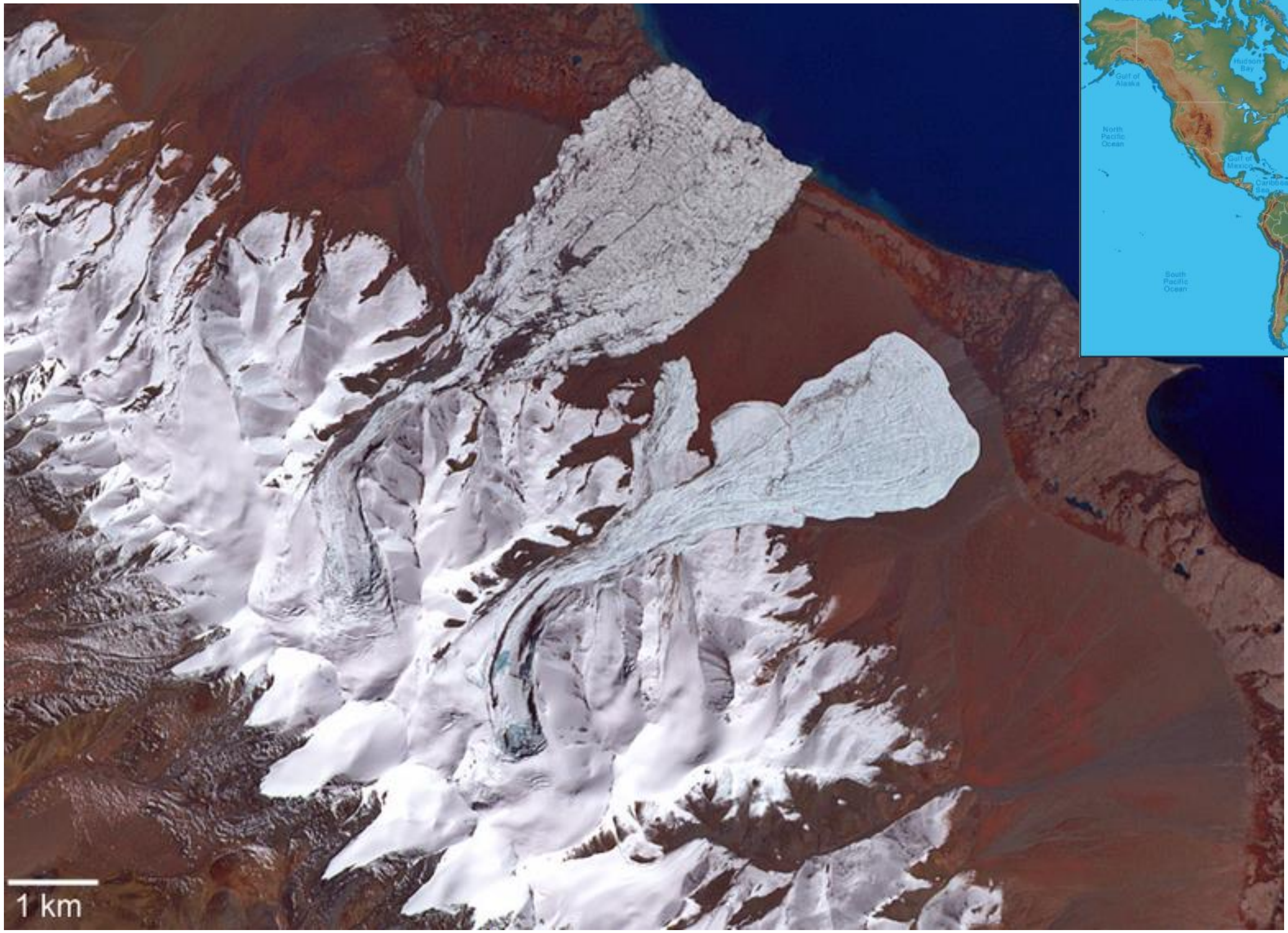


Image acquired by NASA's satellite ASTER on 4th October 2016.

Subglacial lake formation due to thermal regime ?

AruCo Lake twin glacier collapse

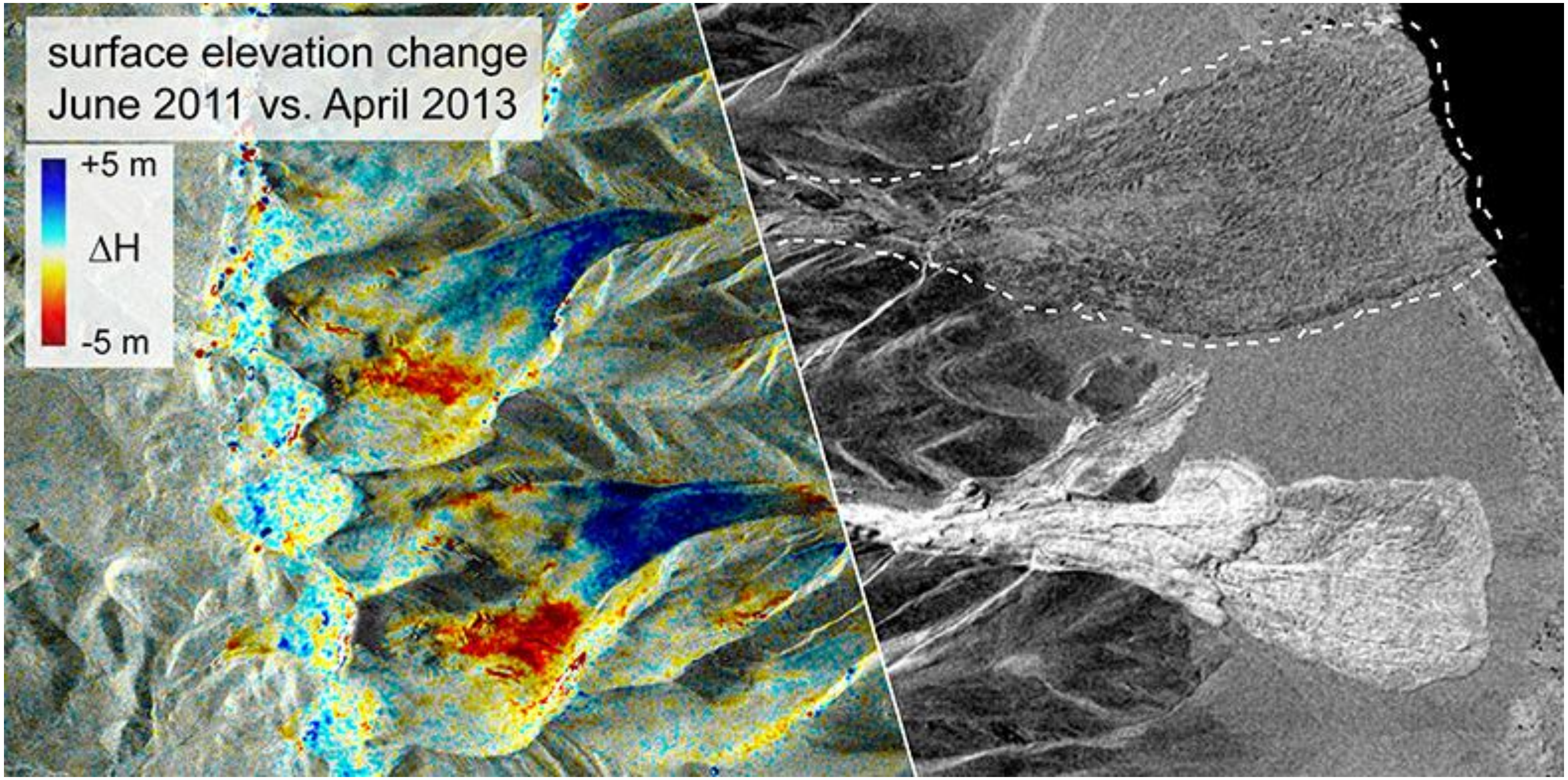


Image: Silvan Leinss / ETH Zurich; satellite data source: TanDEM-X / TerraSAR-X, DLR

AruCo Lake twin glacier collapse

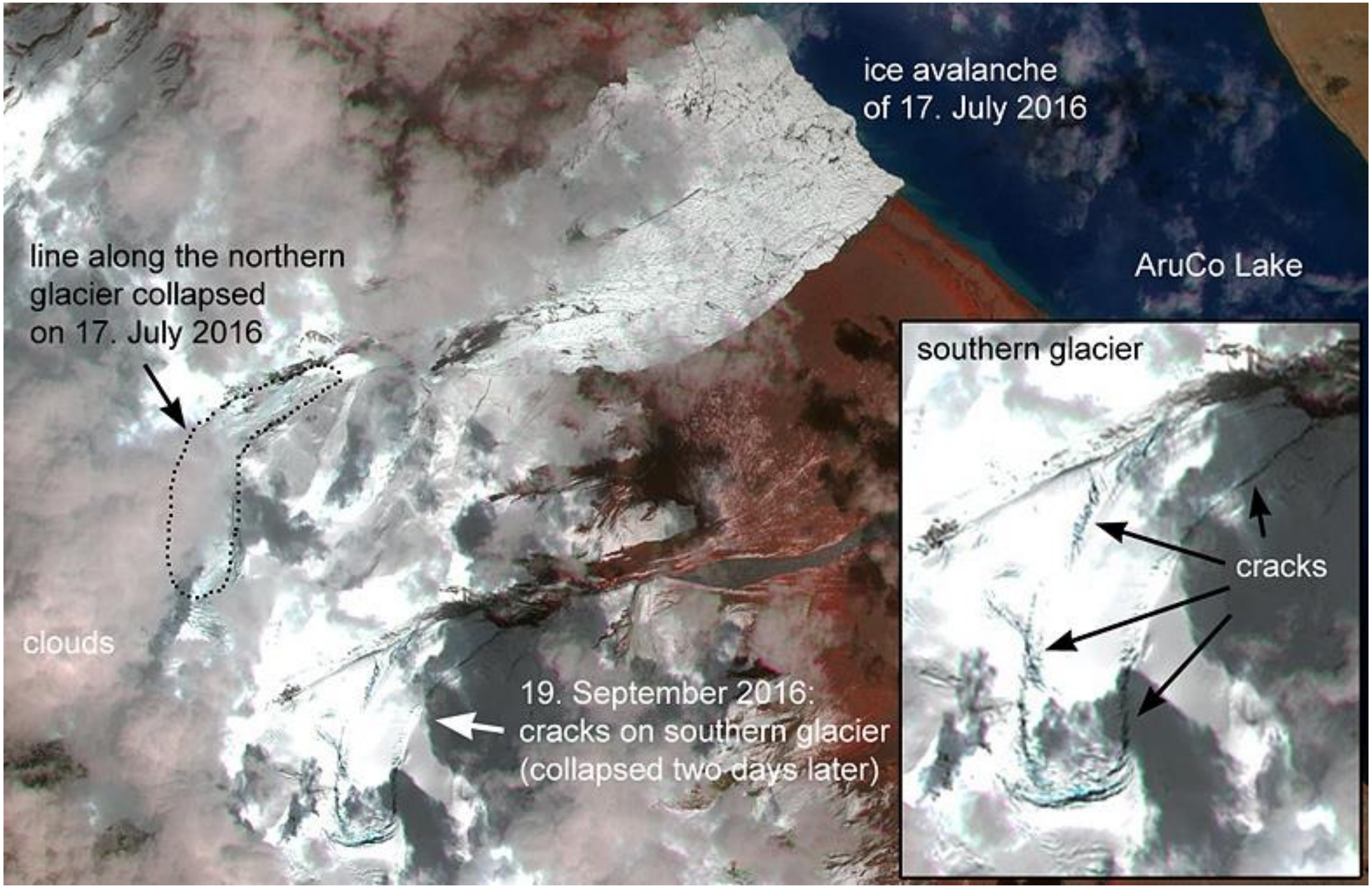


Image: Silvan Leinss / ETH Zurich; satellite data: Sentinel 2, ESA

UiO : University of Oslo

CSC

AruCo Lake twin glacier collapse

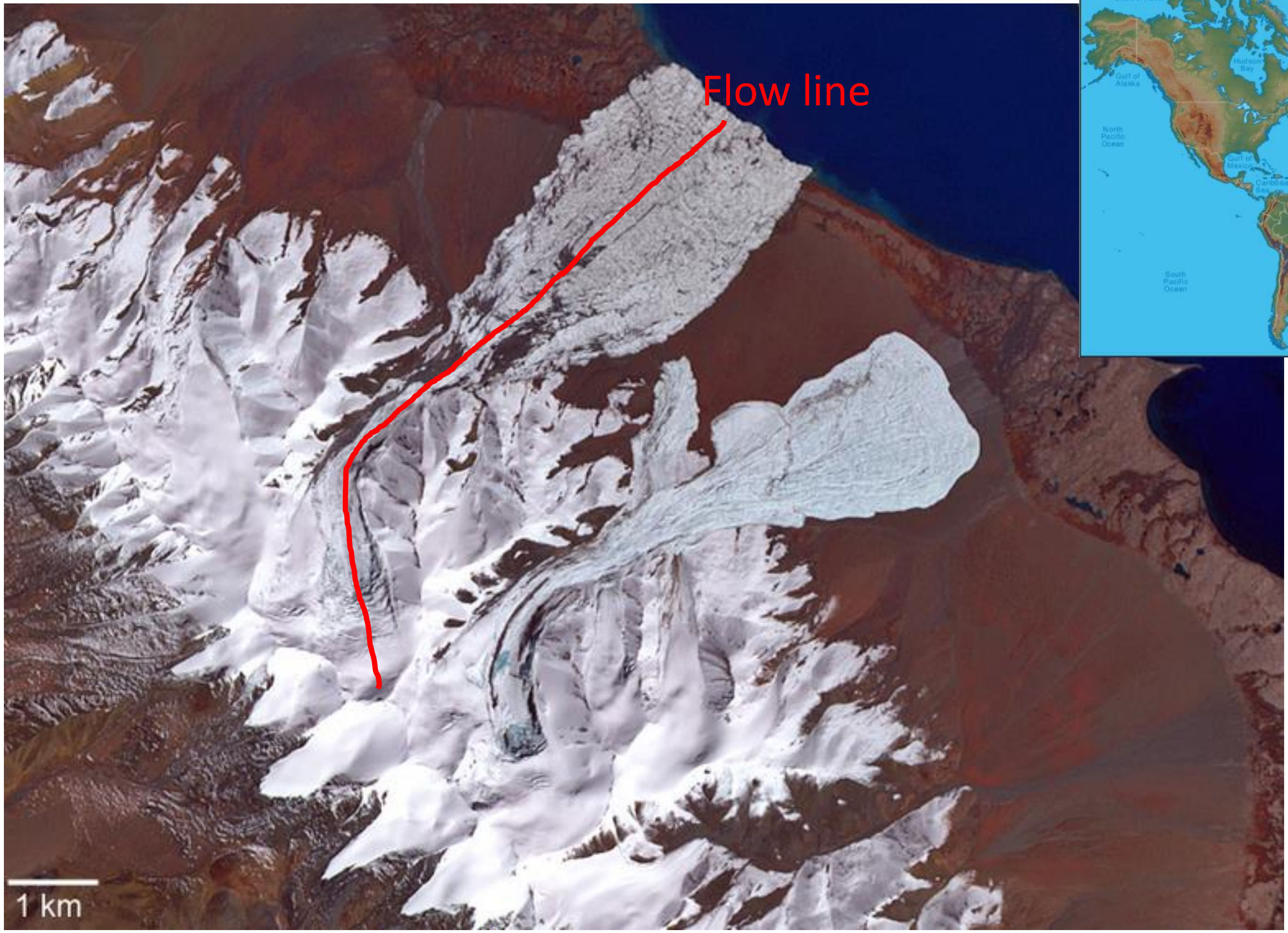


Image acquired by NASA's satellite ASTER on 4th October 2016.

Subglacial lake formation due to thermal regime ?

Application :
Compute steady and transient thermal regime of Aru Co glacier

Step 1 : Build the initial mesh (2D flow line)

Step 2 : Solve steady geometry for constant temperature

Step 3 : Add enthalpy solver with uniform boundary and deformational heating

Step 4 : Add percolation solver and set up non uniform boundary condition

Step 5 : Add sliding

Step 1 : Build the initial mesh (2D flow line)

1. Compile the different solver and user function :

```
elmerf90 -o bin/fct_aruco SRC/fct_aruco.f90
elmerf90 -o bin/SurfBoundary SRC/SurfBoundary.f90
elmerf90 -o bin/Percol_1D_solver SRC/percol_1D_solver.f90
elmerf90 -o bin/EnthalpySolver SRC/EnthalpySolver.f90
```

Step 1 : Build the initial mesh (2D flow line)

1. Compile the different solver and user function
2. Edit “ArucoGlacier.grd”

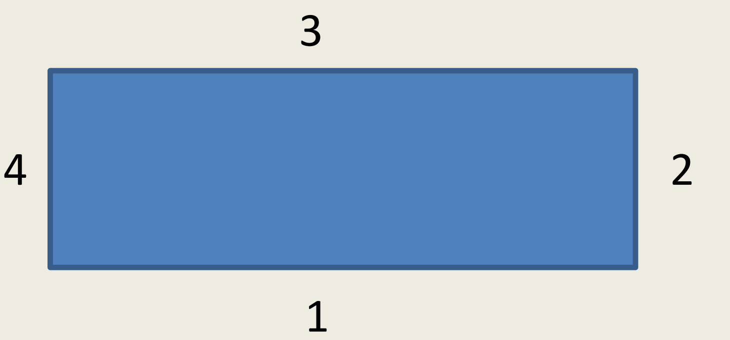
```
***** ElmerGrid input file for structured grid generation *****
Version = 210903
Coordinate System = Cartesian 2D
Subcell Divisions in 2D = 1 1
Subcell Limits 1 = 230.0 5000.0
Subcell Limits 2 = 0.0 1.0
Material Structure in 2D
  1
End
Materials Interval = 1 1
Boundary Definitions
! type     out      int
  1        -1        1        1
  2        -2        1        1
  3        -3        1        1
  4        -4        1        1
End
Numbering = Horizontal
Coordinate Ratios = 1
Element Innernodes = False
Element Degree = 1
Triangles = false
Element Divisions 1 = 300
Element Divisions 2 = 15
```

Subcell Limits 1 → xlim

Subcell Limits 2 → ylim

Element Divisions 1 → Number of element x-direction

Element Divisions 2 → Number of element y-direction



Step 1 : Build the initial mesh (2D flow line)

Edit Step1.sif

```
Solver 1
  Exec Solver = "Before Simulation"
  Equation = "MapCoordinate_ini"
  Procedure = "StructuredMeshMapper" "StructuredMeshMapper"
  Active Coordinate = Integer 2
  Dot Product Tolerance = real 0.01
End
```

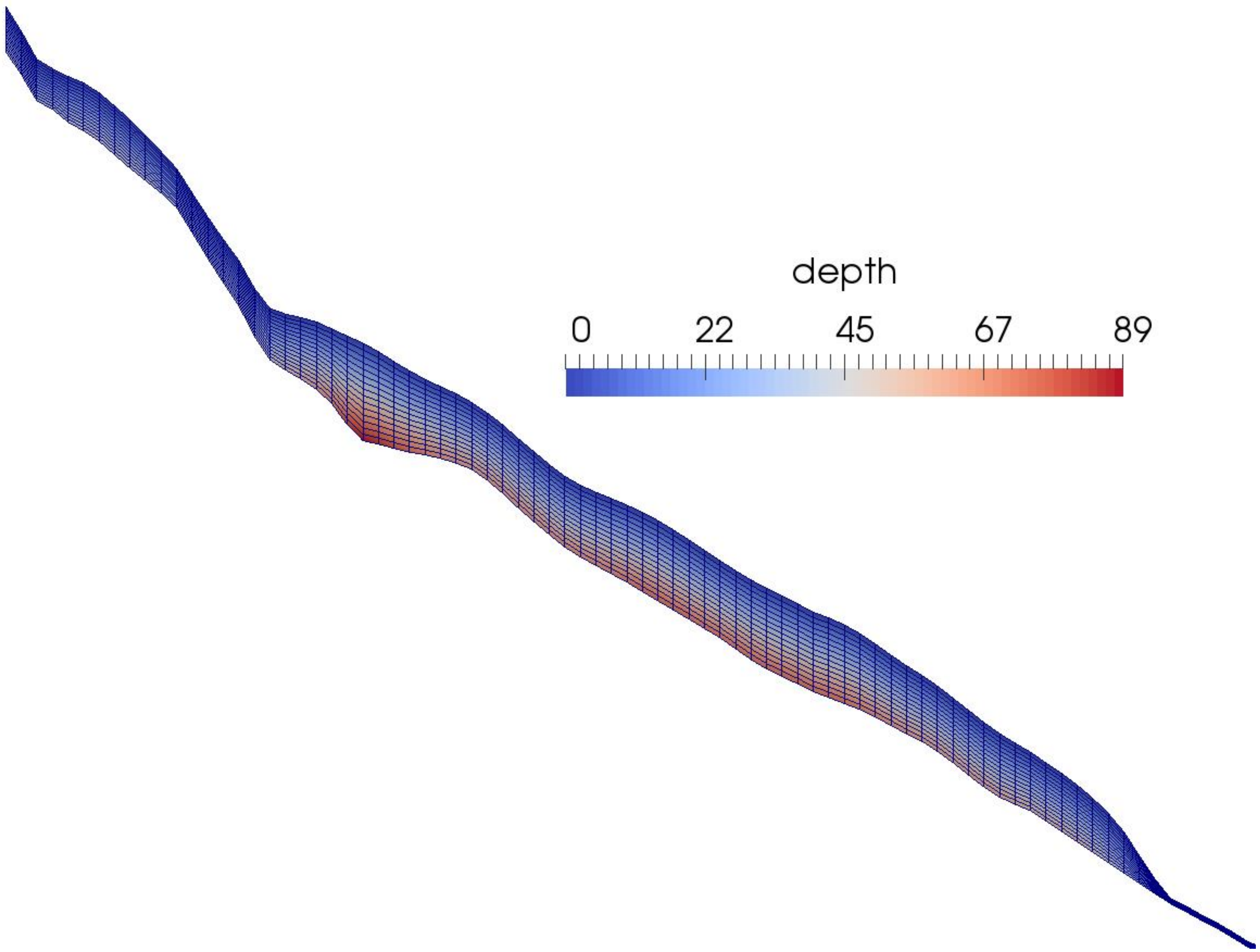
```
Solver 2
Equation = "Flowdepth"
Procedure = File "ElmerIceSolvers" "FlowDepthSolver"
Variable = String "Depth"
Variable DOFs = 1
Linear System Solver = "Direct"
Gradient = Real -1.0E00
End
```

Step 1 : Build the initial mesh (2D flow line)

Boundary :

```
! Bedrock
Boundary Condition 1
  Target Boundaries = 1
    Name = "bed"

Bottom Surface = Variable Coordinate 1
Real Procedure "bin/fct_aruco" "interp_bed"
End

! Upper Surface
Boundary Condition 3
  Target Boundaries = 3

Top Surface = Variable Coordinate 1
Real Procedure "bin/fct_aruco" "interp_surf"

Depth = real 0.0

End
```

Step 1 : Build the initial mesh (2D flow line)



Step 2 : Solve steady geometry for constant temperature

Add Stokes solver

```
Solver 3

Equation = "Navier-Stokes"

  Stabilization Method = String Stabilized
  Flow model = String "Stokes"

  Linear System Solver = Direct
  Linear System Direct Method = umfpack

  Nonlinear System Max Iterations = 50
  Nonlinear System Convergence Tolerance  = 1.0e-5
  Nonlinear System Newton After Iterations = 5
  Nonlinear System Newton After Tolerance = 1.0e-02
  !Nonlinear System Relaxation Factor = 1.0
  Nonlinear System Reset Newton = Logical True
  Steady State Convergence Tolerance = Real 1.0e-3

  Exported Variable 1 = String "Mass Balance"
  Exported Variable 1 DOFs = 1
  Exported Variable 2 = String "Surf Enth"
  Exported Variable 2 DOFs = 1
  Exported Variable 3 = String "Densi"
  Exported Variable 3 DOFs = 1
  Exported Variable 4 = String "Firn"
  Exported Variable 4 DOFs = 1

  Exported Variable 5 = String "Mesh Velocity"
  Exported Variable 5 DOFs = 2
End
```

New variables for boundary condition

Need by MeshMapper

Step 2 : Solve steady geometry for constant temperature

Material

```
Material 1
  Density = Real 9.150149e-19  ! MPa - a - m (910kg/m3)

  Viscosity = real 0.1
  Viscosity Model = String "power law"
  Viscosity Exponent = Real $1.0/3.0
  Critical Shear Rate = real $1.0E-03/31556926.0

End
```

Constant viscosity

Body Force

```
Body Force 1
  Flow BodyForce 1 = Real 0.0
  Flow BodyForce 2 = Real -9.7562e15  !MPa - a - m
End
```

Boundary condition

```
! Bedrock
Boundary Condition 1
  Target Boundaries = 1
    Name = "bed"
   Velocity 1 = Real 0.0
  Velocity 2 = Real 0.0
End

! Upper limit of the glacier
Boundary Condition 4
  Target Boundaries = 4
  Velocity 1 = real 0.0
 End
```

No sliding

Step 2 : Solve steady geometry for constant temperature

Add Solver for Surface Boundary

```
Solver 4
 Equation = SurfBoundary
  Variable = Mass Balance
  Variable DOFs = 1
  procedure =  "bin/SurfBoundary" "SurfBoundary"
  Exported Variable 1 = String "Surf Enth"
  Exported Variable 1 DOFs = 1
  Exported Variable 2 = String "Densi"
  Exported Variable 2 DOFs = 1
  Exported Variable 3 = String "Firn"
  Exported Variable 3 DOFs = 1
  Exported Variable 4 = String "Melting"
  Exported Variable 4 DOFs = 1
End
```

```
Constants
 T_ref_enthalpy = real 200.0

 delta_T= real 13.0
 Tmean= real -2.5
 Precip= real 1.1
End
```

Need some constants

Climatic parameters

Step 2 : Solve steady geometry for constant temperature

Add Free Surface solver

```
Solver 5
  Equation =  String "Free Surface Evolution"
  Variable = "Zs Top"
  Variable DOFs = 1
  Procedure = "FreeSurfaceSolver" "FreeSurfaceSolver"
  Before Linsolve = "EliminateDirichlet" "EliminateDirichlet"
  Apply Dirichlet = Logical true
  Linear System Solver = Iterative
  Linear System Iterative Method = BiCGStab
  Linear System Max Iterations  = 10000
  Linear System Preconditioning = ILU1
  Linear System Convergence Tolerance = 1.0e-08
  Nonlinear System Max Iterations = 100
  Nonlinear System Min Iterations = 2
  Nonlinear System Convergence Tolerance = 1.0e-10
  Steady State Convergence Tolerance = 1.0e-4
  Stabilization Method = Bubbles
  Flow Solution Name = String "Flow Solution"

  Exported Variable 1 =  Zs Top Residual
  Exported Variable 1 DOFS = 1
  Exported Variable 2 = Ref Zs Top
  Exported Variable 2 DOFS = 1

End
```

Step 2 : Solve steady geometry for constant temperature

```
Body 2
  Name= "surface"
  Equation = 2
  Material = 1
  Body Force = 2
  Initial Condition = 2
End

! Upper Surface
Boundary Condition 3
  Target Boundaries = 3

   Body Id = 2
```

```
Body Force 2
  Zs top Accumulation = Equals Mass Balance
End
```

```
Material 1
  Min Zs top = Variable Zbed
    Real MATC "tx(0)+10.0"
  Max Zs top = Variable Zbed
    Real MATC "tx(0)+10000.0"

End
```

Limit for the free surface

```
Solver 6
  Exec Solver = "Before Simulation"
  Equation = "ExportVertically"
  Procedure = File "ElmerIceSolvers" "ExportVertically"
  Variable = String "Zbed"
  Variable DOFs = 1
  Linear System Solver = Iterative
  Linear System Iterative Method = BiCGStab
  Linear System Max Iterations  = 500
  Linear System Preconditioning = ILU1
  Linear System Convergence Tolerance = 1.0e-06

  Nonlinear System Max Iterations = 1
  Nonlinear System Convergence Tolerance = 1.0e-06
End

! Bedrock
Boundary Condition 1
  Target Boundaries = 1
    Name = "bed"
    Zbed = Variable Coordinate 1
     Real Procedure "bin/fct_aruco" "interp_bed"
End
```

Need the variable Zbed = bed elevation

Step 2 : Solve steady geometry for constant temperature

Add MeshMapper

```
Solver 7
  Equation = "MapCoordinate"
  Procedure = "StructuredMeshMapper" "StructuredMeshMapper"
  Active Coordinate = Integer 2
  Mesh Velocity Variable = String "Mesh Velocity 2"
  Mesh Velocity First Zero = Logical True
  Dot Product Tolerance = real 0.01
End
```

```
! Upper Surface
Boundary Condition 3
  Target Boundaries = 3

   Body Id = 2

Top Surface = Equals Zs Top

End
```

Set Equations:

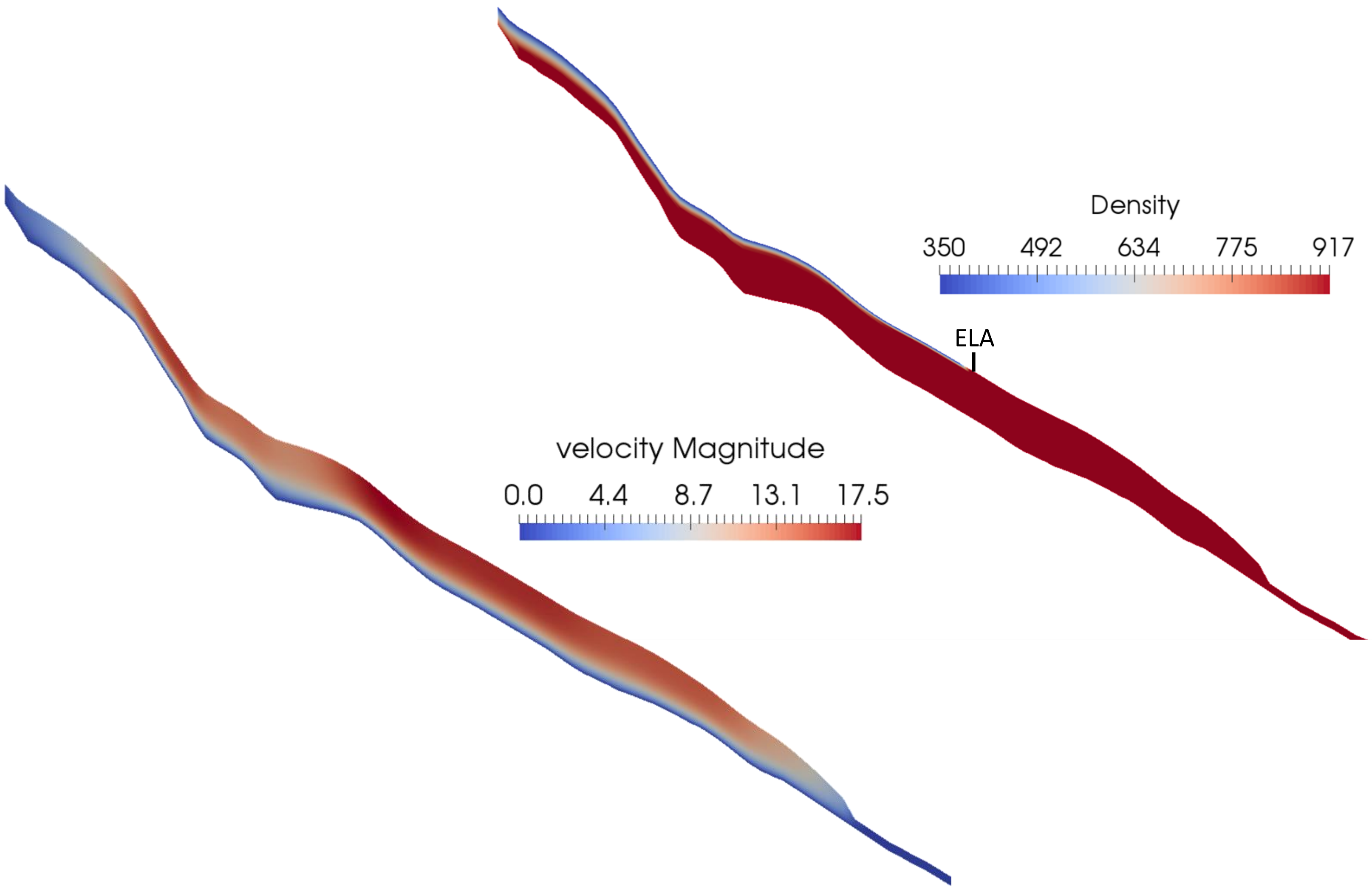
```
Equation 1
  Active Solvers(5) = 1 2 3 6 7
 Flow Solution Name = String "Flow Solution"
 Convection = Computed
End

Equation 2
 Active Solvers (2) =  4 5
 Flow Solution Name = String "Flow Solution"
 Convection = Computed
End
```

Step 2 : Solve steady geometry for constant temperature

Set transient simulation

```
Simulation
  Coordinate System  =  Cartesian 2D
  Simulation Type = Transient
  Timestepping Method = BDF
  BDF order = 2

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

  Timestep Intervals = 100!!!
  Timestep Sizes = 2.0
  Output Intervals = 2


  Output File = "$Step".result"
  Post File = "$Step".vtu"
  max output level = 3

End
```

Step 2 : Solve steady geometry for constant temperature



Step 3 : Add enthalpy solver and deformational heating

Add deformational heating solver:

```
Solver 8
  Equation = DeformationalHeat
  Variable = W
  Variable DOFs = 1
  procedure =  "ElmerIceSolvers" "DeformationalHeatSolver"

  Linear System Solver = "Iterative"
  Linear System Iterative Method = "BiCGStab"
  Linear System Max Iterations = 500
  Linear System Convergence Tolerance = 1.0E-07
  Linear System Abort Not Converged = True
  Linear System Preconditioning = "ILU0"
  Linear System Residual Output = 1
  Steady State Convergence Tolerance = 1.0E-02
  Nonlinear System Convergence Tolerance = 1.0E-03
  Nonlinear System Max Iterations = 10
  Nonlinear System Relaxation Factor = Real 1.0

End
```

Step 3 : Add enthalpy solver and deformational heating

Add enthalpy solver :

```
Solver 9

  Transient Simu = logical false
  Nb_steady_simu = integer 20

 Equation = String "Enthalpy Equation"
  Procedure = File "bin/EnthalpySolver" "EnthalpySolver"
  Variable = String "Enthalpy_h"
  Linear System Solver = "Iterative"
  Linear System Iterative Method = "BiCGStab"
  Linear System Max Iterations = 500
  Linear System Convergence Tolerance = 1.0E-07
  Linear System Abort Not Converged = True
  Linear System Preconditioning = "ILU0"
  Linear System Residual Output = 1
  Steady State Convergence Tolerance = 1.0E-04
  Nonlinear System Convergence Tolerance = 1.0E-07
  Nonlinear System Max Iterations = 3
  Nonlinear System Relaxation Factor = Real 1.0
  Apply Limiter = Logical true
  Apply Dirichlet = Logical True
  Stabilize = True

  Exported Variable 1 = String "Phase Change Enthalpy"
  Exported Variable 1 DOFs = 1
  Exported Variable 2 = String "water content"
  Exported Variable 2 DOFs = 1
  Exported Variable 3 = String "temperature"
  Exported Variable 3 DOFs = 1
End
```

Force steady state computation for 20 timesteps

Step 3 : Add enthalpy solver and deformational heating

```
Constants
 T_ref_enthalpy = real 200.0
 L_heat = real 334000.0
! Cp(T) = A*T + B
 Enthalpy Heat Capacity A = real 7.253
 Enthalpy Heat Capacity B = real 146.3
 P_triple = real 0.061173 !Triple point pressure for water (MPa)
 P_surf = real 0.1013 ! Surface atmospheric pressure(MPa)
 beta_clapeyron = real 0.0974 ! clausus clapeyron relationship (K MPa-1)

 delta_T= real 13.0
 Tmean= real -2.5
 Precip= real 1.1
End
```

```
Material 1

  Viscosity = Variable Temperature
           real procedure "bin/fct_aruco" "glen_law"


  Enthalpy Density = Equals Densi
  Enthalpy Heat Diffusivity = variable temperature
     Real Procedure "bin/fct_aruco" "diffusivity_calc"
  Enthalpy Water Diffusivity = real $1.045e-4*3600*24*365.25

End
```

Coupling with temperature

From simple firn model

Step 3 : Add enthalpy solver and deformational heating

```
Body Force 1
  Flow BodyForce 1 = Real 0.0
  Flow BodyForce 2 = Real -9.7562e15  !MPa - a - m

  Heat Source = Variable W
           real MATC "if (tx<0.0) (0.0); else (tx*1e6/917.0)" !Convert to J yr-1 kg-1

  Enthalpy_h Upper Limit = real $1.3620e+05+0.03*334000.0
  Enthalpy_h Lower Limit = real 0.0

End
```

Strain Heating

Water Content limited to 3%

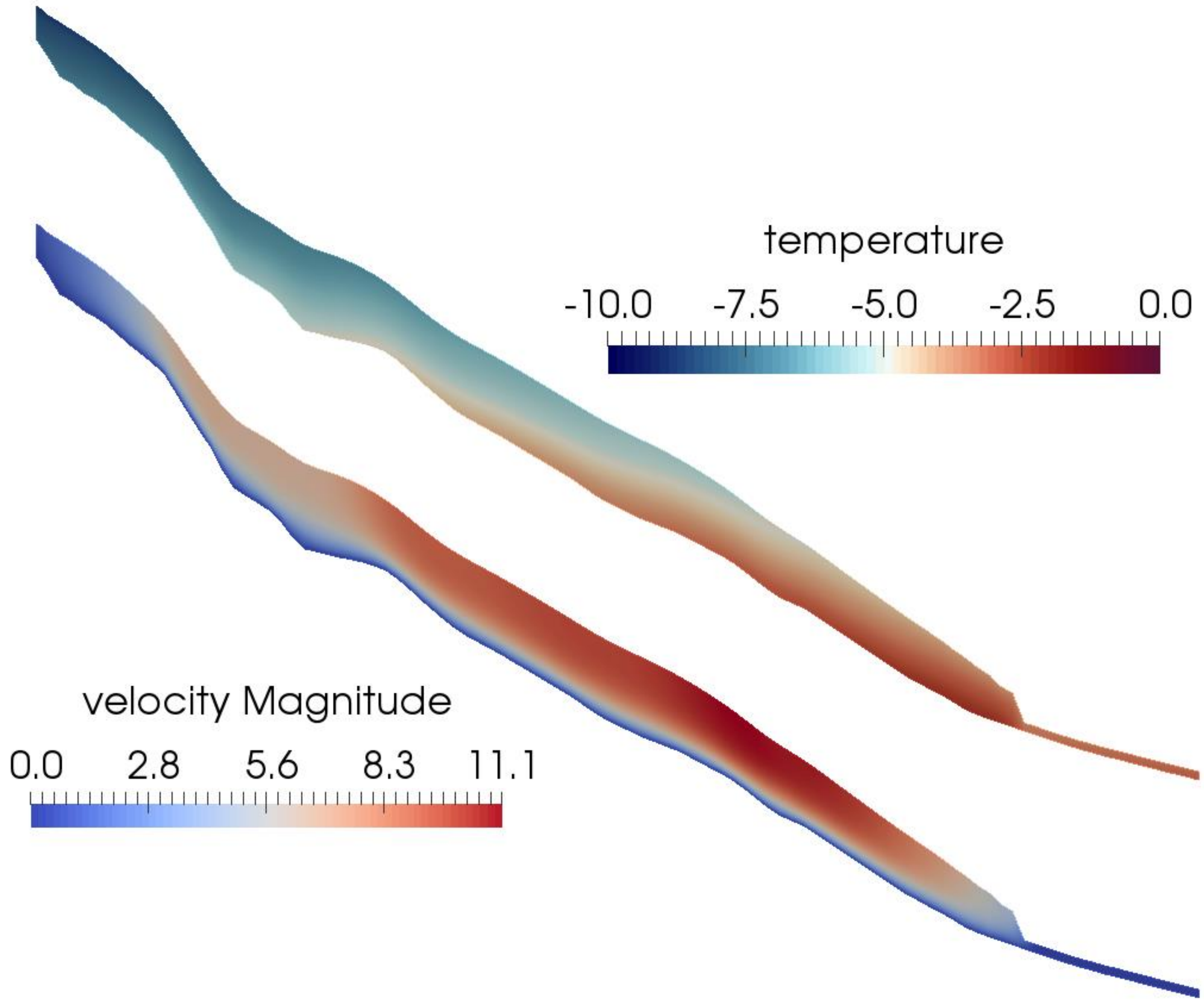
```
! Upper Surface
Boundary Condition 3
  Target Boundaries = 3

Enthalpy_h = Variable Coordinate 2
   real MATC "3.626*(-2.5-(tx-5000.0)*0.006+273.15)^2+146.3*(-2.5-(tx-
5000.0)*0.006+273.15)-200*(146.3+3.626*200)"

End

! Bedrock
Boundary Condition 1
  Target Boundaries = 1
    Name = "bed"


  Enthalpy Heat Flux BC = logical True
  Enthalpy Heat Flux = real $0.040*3600*24*365.25
End
```

$$H_s(T_s) = \frac{A}{2}Ts^2 + BT_s - T_0(B + \frac{A}{2}T_0)$$

$$C_p(T) = A \times T + B$$

Basal heat flux (J yr^{-1} m^{-2})

Step 3 : Add enthalpy solver and deformational heating



Step 4 : Add percolation solver

```
Solver 10

Equation = String "percol_1D"
Procedure = File "bin/Percol_1D_solver" "percol_1D_solver"

End
```

```
! Upper Surface
Boundary Condition 3
  Target Boundaries = 3

Surf_melt = Equals Melting

Enthalpy_h = Equals Surf Enth

End
```

Melting: Amount of water (m w.eq.) to refreeze each time the solver is called

Surf Enth: Computed in the boundary solver

```
Constants

rho_ice = real 917.0
rho_w = real 1000.0
Sr = real 0.01

End
```

0.01: % of the porosity able to retain liquid water by capillarity forces in the firn

Step 4 : Add percolation solver

Need to take into account seasonal variability and a 6 month timestep for enthalpy

- Multiple timestep depending on the solver

```
Timestep Intervals = 1200!!!
 Timestep Sizes = 0.5
 Output Intervals = 8
```

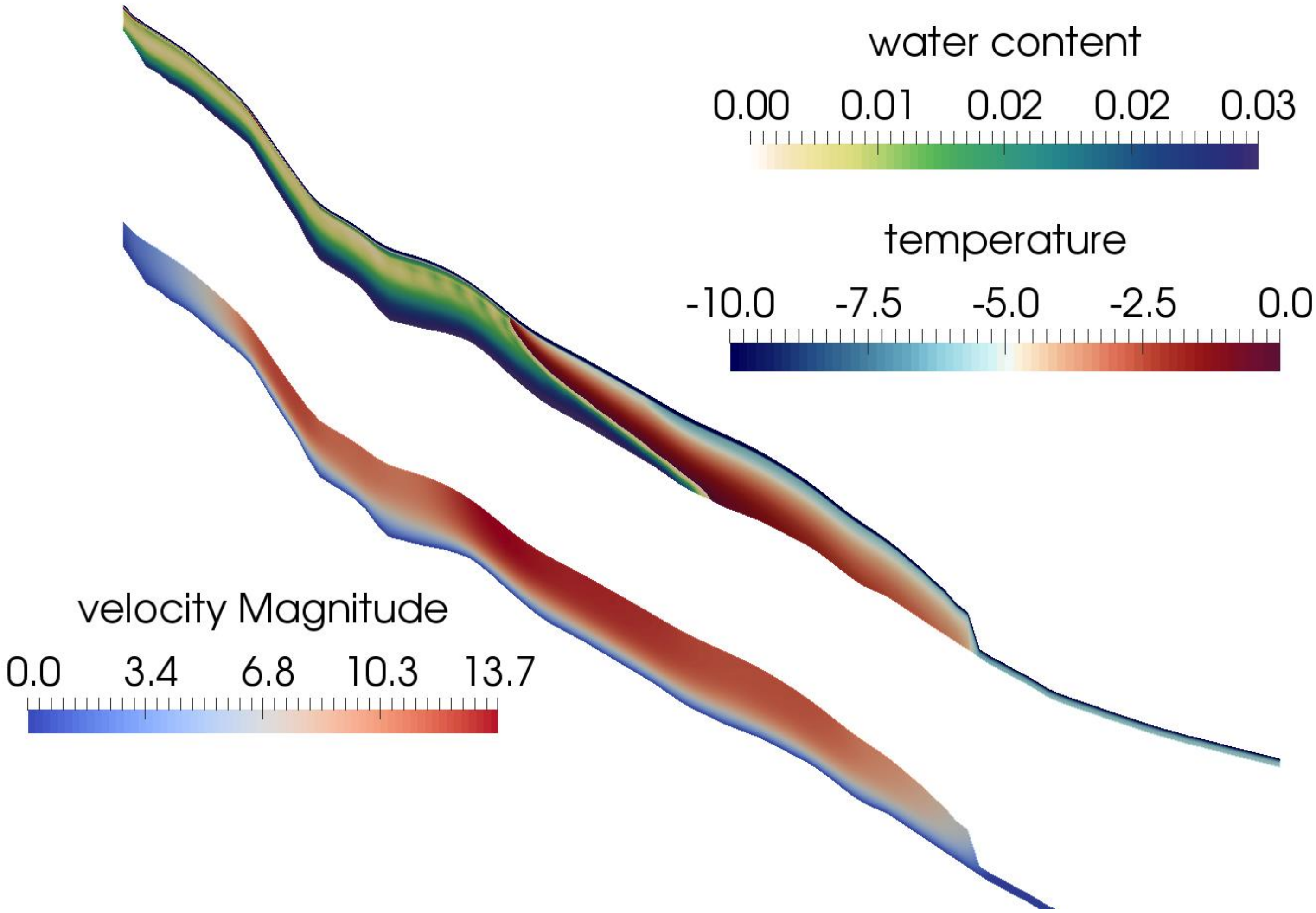
Set simulation to the smallest timestep = 6 month

```
exec interval = 4
Timestep Scale = Real 4.0
```

To add in the slover section to change the timestep .
Here 4*0.5 = 2yr

Enthalpy and boundary solvers = 6 month
Percolation solver = 1 year
Others solvers = 2 years

Step 4 : Add percolation solver

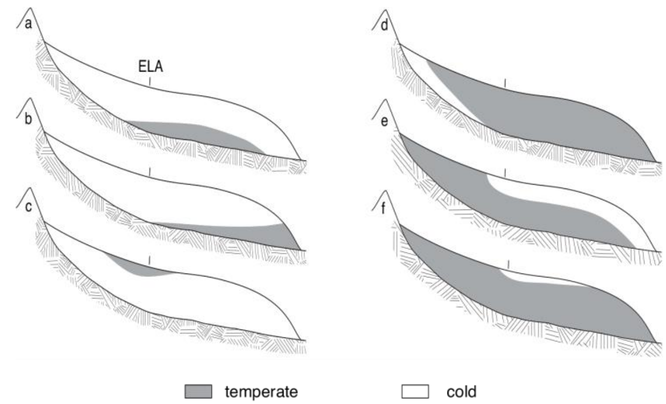


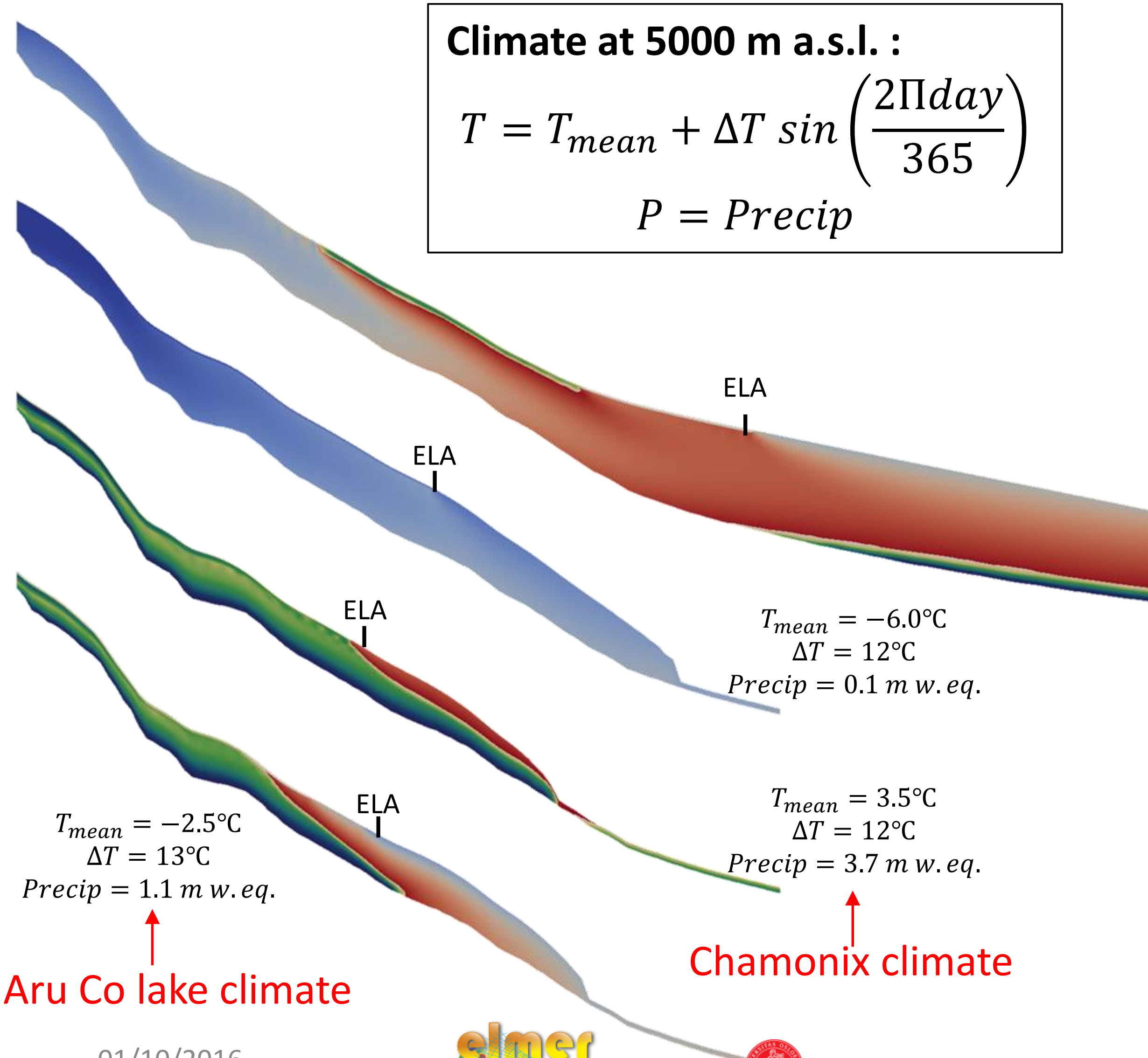
Examples : play with climatic condition

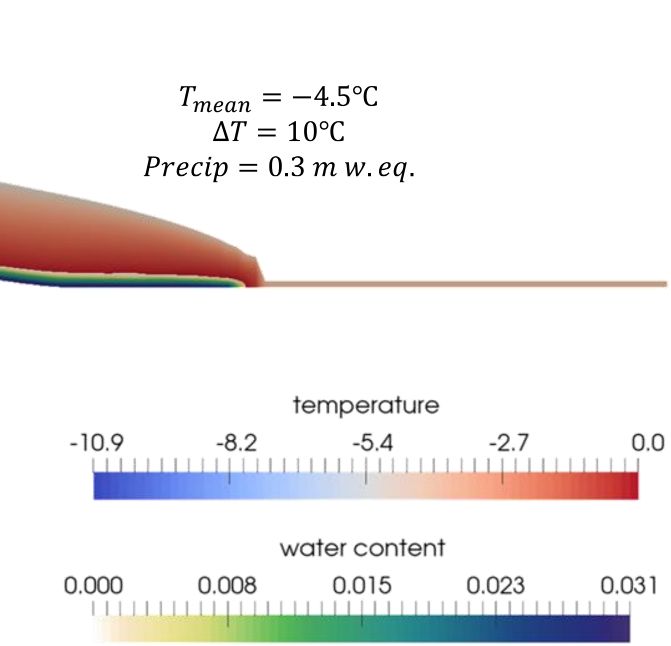
Climate at 5000 m a.s.l. :

$$T = T_{mean} + \Delta T \, sin\left(\frac{2\Pi day}{365}\right)$$

$$P = Precip$$







Step 5 : Add sliding and frictional heating

```
Boundary Condition 1
  Target Boundaries = 1
    Name = "bed"

 ComputeNormal = Logical True
 Mass Consistent Normals = Logical True

  Normal-Tangential Velocity = Logical True
  Velocity 1 = Real 0.0e0
  Slip Coefficient 2 = Variable Temperature
  REAL MATC "if (tx<-0.50) (1.0); else ((tx/-0.50)*0.99+0.01)"
```

Slip coefficient change from 1.0 to 0.01 when ice become temperate

```
 Enthalpy_h Load = Variable Velocity 1
        Real Procedure  "ElmericeUSF" "getFrictionLoads"
```

Adding frictional heating

```
End
```

```
 Exported Variable 6 = Flow Solution Loads[Fx:1 Fy:1 Fz:1 CEQ Residual:1 ]
 Calculate Loads = Logical True
```

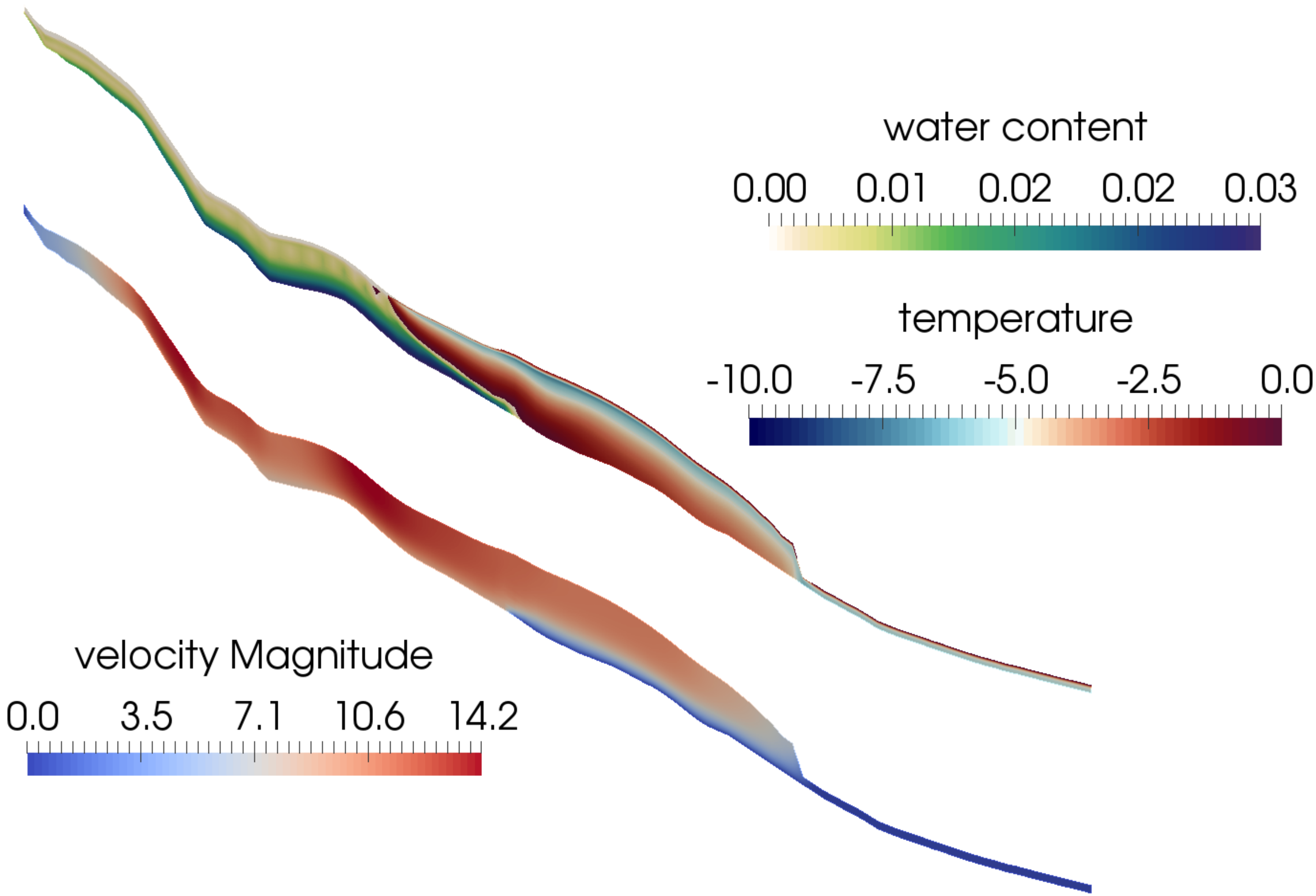
Stokes Solver
Needed for friction heating

Step 5 : Add sliding and frictional heating

```
Solver 11
  Equation = "NormalVector"
  Exec Solver = Before Simulation
  Procedure = "ElmerIceSolvers" "ComputeNormalSolver"
  Variable = String "Normal Vector"
  Variable DOFs = 2
  Optimize Bandwidth = Logical False
  ComputeAll = Logical False
End
```

Compute Normal
Needed for friction heating

Step 5 : Add sliding and frictional heating



Exercices

Model a surge !

- Restart the run from step 4 with sliding for some years, try different value of slip coefficient

Model air temperature warming !

- Restart from step 4 and modify SurfBoundary Solver to have time dependent Tmean

```
Simulation
  Restart File = "Aruco_step4_.result"
  Restart Position = 0
End
```