A real world application
Tête Rousse Glacier

Olivier GAGLIARDINI
LGGE - Grenoble - France
Tête Rousse Glacier

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

✓ **Step 1**
- Tête Rousse Glacier flow without a water filled-cavity (diagnostic)

✓ **Step 2**
- Influence of an empty cavity below Tête Rousse Glacier (diagnostic)

✓ **Step 3**
- Rate of closure of the cavity for a given drainage scenario (prognostic)
Location (Mont Blanc Area, France Alps)
Location (Mont Blanc Area, France Alps)

Tête Rousse glacier
3100 to 3300 m
0.08 km² (2007)
Chronology

The 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
The 1892 catastrophe

11 July 1892

175 fatalities

100 000 m$^3$ of water

Flood produced

800 000 m$^3$ of sediment

@Vincent, LGGE
The 1892 catastrophe
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
The radar measurements showed a zone (volume) with an anomaly.
Glaciological studies

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

Water volume of 65,000 m$^3$

Report given to public authorities in March 2010
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 due to the weight of the ice column

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

The public authorities have been warned immediately (13 July, 2010)

It has been 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 26 of August
A new risk?

But was stopped the 1\textsuperscript{st} September:

What was the risk of breakout 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?

right

upstream

$h \sim 50$ m

$l_y \sim 80$ m

$l_x \sim 30$ m

left
Timing for answering

Sonar data

Meeting with the mayor of St Gervais
Proposed application

Construct a model of the flow of Tête Rousse Glacier

- Step 1: Without the cavity (normal state)
- Step 2: Add an empty cavity (stress analysis)
- Step 3: Rate of closure of the cavity (surface deformation analysis)
Data for ice flow modelling

- Bedrock DEM
- 2007 Surface DEM
- Cavity topography from sonar measurements
- Few surface velocities, without the cavity (0.6 m/a at the centre of the glacier)
- 27 Stakes to measure surface displacement during drainage
Modelling Tête Rousse Glacier

✓ Step 1
- 1 Tête Rousse Glacier flow without a water filled-cavity (diagnostic)
- 1.2 Add Stress Solver to get the stress
- 1.3 Compute the Eigenvalues of the stress tensor

✓ Step 2
- 2 Influence of an empty cavity below Tête Rousse Glacier (diagnostic)
- 2.1 Apply a water pressure in the cavity

✓ Step 3
- 3 Rate of closure of the cavity for a given drainage scenario (prognostic)
- 3.1 Add a drainage scenario
Step 1: Work to do

- create the mesh
- impose the boundary conditions in the SIF file
- test other BCs on the lateral boundary
- test sliding at the base of the glacier
Step 1: steps to make the mesh

1/ build the teterousse.geo file (input file of gmsh, footprint of the glacier)

2/ gmsh to get teterousse.msh (still footprint of the glacier)

3/ ElmerGrid to transform into Elmer format (still footprint of the glacier)

4/ ExtrudeMesh to extrude vertically the footprint (now we have a 3D mesh)

5/ MshGlacierDEM to deform vertically the mesh using the surface and bedrock DEMs by modifying the mesh.nodes Elmer file
Step 1: Makegeo.m (create a .geo file)

clear;
lc_out=18.0; (size of the element in the plane)

A=dlmread('Contour_TR_glider.dat'); (Read contour points)
fid1=fopen('teterousse.geo','w');
fprintf(fid1,'Mesh.Algorithm=5; 
'); (delaunay algorithm)

As=size(A,1);

np=0;
for ii=1:As
    np=np+1;
    fprintf(fid1,'Point(%g)={%14.7e,%14.7e,0.0,%g}; 
',np,A(ii,1),A(ii,2),lc_out);
end

fprintf(fid1,'Spline(1)=
');
for ii=1:As
    fprintf(fid1,'%g,',ii);
end
fprintf(fid1,'%g}; 
',1);

fprintf(fid1,'Line Loop(2)={1}; 
');
fprintf(fid1,'Plane Surface(3) = {2}; 
');
fprintf(fid1,'Physical Line(4) = {1}; 
');
fprintf(fid1,'Physical Surface(5) = {3}; 
');

fclose(fid1)
Step 1: Makegeo.m (Extrude)

% create teterousse.msh using gmsh
!gmsh teterousse.geo -l -2

% convert teterousse.gmsh in an Elmer type mesh
!ElmerGrid 14 2 teterousse.msh -autoclean

% Extrude vertically the mesh (1m thick)
!ExtrudeMesh teterousse WithOutCavity 14 1 1 0 0 0 0

% Deform vertically using the surface and bedrock DEM
% Input data are in mesh_input.dat
!./MshGlacierDEM

% Make a .ep to visualize in ElmerPost the mesh
!ElmerGrid 2 3 WithOutCavity
Step 1: gmsh (create a .msh file)

```
gmsh teterousse.geo -1 -2
```

help: [http://www.geuz.org/gmsh/](http://www.geuz.org/gmsh/)

line commands:
"-1 -2" performs 1D and 2D mesh generation and then exit
Step 1: ExtrudeMesh

Get ExtrudeMesh in elmerice/Meshers/

Compile it:
> cc ExtrudeMesh.c -o ExtrudeMesh -lm

Execute
ExtrudeMesh teterousse WithOutCavity 14 1 1 0 0 0 0 0

14 layers, 1m thick in total, 1 partition
Step 1: MshGlacierDEM

From surface and bedrock DEMs, deforms vertically the initial 1m height mesh.

Here surface and bedrock are given on a regular grid

\[ \begin{align*}
N_x & \text{ points along } x \\
N_y & \text{ points along } y \\
(x_0, y_0) & \text{ where data are missing: } -9999
\end{align*} \]
Step 1: MSH_Glacier3DGrille

Compile MSH_Glacier3DGrille

elmerf90-nosh ../elmerice/Meshers/MshGlacierDEM.f90 -o MshGlacierDEM

MshGlacierDEM reads mesh_input.dat

! Name of the mesh directory (should exist)
WithOutCavity
! Surface DEM name file
DEM_TR_surf.dat
! Nsx, Nsy
268 118
! xs0, ys0
947700.0 2104850.0
! lsx, lsy
800.0 350.0
! Bedrock DEM name file
DEM_TR_bed.dat
! Nbx, Nby
301 176
! xb0, yb0
947700.0 2104850.0
! lbx, lby
600.0 350.0
! Minimum ice thickness
1.0
Step 1: Hypothesis of the modelling

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 1: use Glen’s law

\[ D_{ij} = A \tau_c^{n-1} S_{ij} \quad ; \quad S_{ij} = A^{-1/n} I_D^{(1-n)/n} D_{ij} \]

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

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

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

Cuffey and Paterson (2010)

assume a constant temperature of -1°C
Step 1: use Glen’s law

\[
\begin{align*}
\text{yearinsec} &= 365.25\times24\times60\times60 \\
\rho_i &= \frac{900.0}{(1.0\times10^6\text{yearinsec}^2)} \\
\rho_w &= \frac{1000.0}{(1.0\times10^6\text{yearinsec}^2)} \\
\text{Prefactor from Paterson (1994) in MPa}^{-3}\text{ a}^{-1} \\
A_1 &= 3.985\times10^{-13}\times\text{yearinsec}\times1.0\times10^8 \\
A_2 &= 1.916\times10^3\times\text{yearinsec}\times1.0\times10^8 \\
\text{gravity} &= -9.81\times\text{yearinsec}^2
\end{align*}
\]

Material 1
- Density = Real $\rho_i$
- Viscosity Model = String "glen"
- Viscosity = 1.0 ! Dummy but avoid warning output
- Glen Exponent = Real 3.0
- Limit Temperature = Real -10.0
- Rate Factor 1 = Real $A_1$
- Rate Factor 2 = Real $A_2$
- Activation Energy 1 = Real 60e3
- Activation Energy 2 = Real 139e3
- Glen Enhancement Factor = Real 1.0
- Critical Shear Rate = Real 1.0e-10
- Constant Temperature = Real -1.0
End
Step 1: Boundary Conditions

! Bedrock
Boundary Condition 1
  Target Boundaries = 1
  Velocity 1 = Real 0.0
  Velocity 2 = Real 0.0
  Velocity 3 = Real 0.0
End

! Upper Surface
Boundary Condition 2
  Target Boundaries = 2
End

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

No sliding
Natural BC, nothing to do!
Null horizontal velocities
Step 1: Other BCs for the lateral boundary

! lateral side of the glacier
Boundary Condition 3
  Target Boundaries = 3
End

Natural BC

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

zero velocity

Conclusion?
Step 1: Add sliding on the bedrock

Friction law in Elmer:
\[ C_i u_i = \sigma_{ij} n_j \quad (i = 1, 2) \]
\[ C_{it} u_t = \sigma_{nt} ; C_{nn} u_n = \sigma_{nn} \]

where \( n \) is the surface normal vector

How to evaluate the Slip Coefficient?

Friction law in Elmer:
\[ C_i u_i = \sigma_{ij} n_j \quad (i = 1, 2) \]
\[ C_{it} u_t = \sigma_{nt} ; C_{nn} u_n = \sigma_{nn} \]

where \( n \) is the surface normal vector

! Bedrock BC
Boundary Condition 1
Target Boundaries = 1
Flow Force BC = Logical True
Normal-Tangential Velocity = Logical True
Velocity 1 = Real 0.0e0
Slip Coefficient 2 = Real 0.1
Slip Coefficient 3 = Real 0.1
End
Modelling Tête Rousse Glacier

✓ Step 1
- 1 Tête Rousse Glacier flow without a water filled-cavity (diagnostic)
- 1.2 Add Stress Solver to get the stress
- 1.3 Compute the Eigenvalues of the stress tensor

✓ Step 2
- 2 Influence of an empty cavity below Tête Rousse Glacier (diagnostic)
- 2.1 Apply a water pressure in the cavity

✓ Step 3
- 3 Rate of closure of the cavity for a given drainage scenario (prognostic)
- 3.1 Add a drainage scenario
Step 1.2: Add the Stress Solver

Objective: compute the stress field as

\[ \int_V S_{ij} \Phi \, dV = 2 \int_V \eta D_{ij} \Phi \, dV \]

where \( D_{ij} \) and \( \eta \) are calculated from the nodal velocities using the derivative of the basis functions

- Add a Solver

Solver 2
- Equation = Sij
- Procedure = "ElmerIceSolvers" "ComputeDevStress"
- Variable = -nooutput "Sij"
- Variable DOFs = 1
- Exported Variable 1 = Stress
- Exported Variable 1 DOFs = 6

Flow Solver Name = String "Flow Solution"

- Linear System Solver = Direct
- Linear System Direct Method = umfpack

End
Step 1.2: Add the Stress Solver

- Add this solver in the Equation Section
  \[ \text{Active Solvers}(2) = 1 \ 2 \]

- Tell you want the Cauchy stress to be computed (Material Section)
  \[
  \text{Material 1} \\
  \quad \text{Cauchy Stress} = \text{Logical True} \\
  \quad \text{End}
  \]

- Output:
  negative stress = Compressive stress
  positive stress = Tensile stress

\[
\begin{align*}
\text{Stress.1} & \rightarrow S_{xx} & \text{Stress.4} & \rightarrow S_{xy} \\
\text{Stress.2} & \rightarrow S_{yy} & \text{Stress.5} & \rightarrow S_{yz} \\
\text{Stress.3} & \rightarrow S_{zz} & \text{Stress.6} & \rightarrow S_{xz}
\end{align*}
\]
Modelling Tête Rousse Glacier

✓ Step 1
- 1 Tête Rousse Glacier flow without a water filled-cavity (diagnostic)
- 1.2 Add Stress Solver to get the stress
- 1.3 Compute the Eigenvalues of the stress tensor

✓ Step 2
- 2 Influence of an empty cavity below Tête Rousse Glacier (diagnostic)
- 2.1 Apply a water pressure in the cavity

✓ Step 3
- 3 Rate of closure of the cavity for a given drainage scenario (prognostic)
- 3.1 Add a drainage scenario
Step 1.3: Add the Eigenvalues Solver

Objective: compute the eigenvalues of the Cauchy stress tensor

- Add a Solver

Solver 3

Equation = "EigenStresses"
Procedure = "ElmerIceSolvers" "ComputeEigenValues"
Variable = -nooutput dummy
Variable DOFs = 1

! The 3 eigenvalues
Exported Variable 1 = EigenStress
Exported Variable 1 DOFS = 3

! The 3 eigenvectors (Option)
Exported Variable 2 = EigenVector1
Exported Variable 2 DOFS = 3
Exported Variable 3 = EigenVector2
Exported Variable 3 DOFS = 3
Exported Variable 4 = EigenVector3
Exported Variable 4 DOFS = 3

End
Step 1.3: Add the Eigenvalues Solver

- Add this solver in the Equation Section
  \[ \text{Active Solvers}(3) = 1 \ 2 \ 3 \]

- Output:
  negative stress = Compressive stress
  positive stress = Tensile stress
  ordered \( \rightarrow \) \text{Eigenstress.3} gives the maximal tensile stress

\[ \text{Eigenstress.1} \rightarrow S_1 \]
\[ \text{Eigenstress.2} \rightarrow S_2 \]
\[ \text{Eigenstress.3} \rightarrow S_3 \]
Modelling Tête Rousse Glacier

✔ Step 1
- 1 Tête Rousse Glacier flow without a water filled-cavity (diagnostic)
- 1.2 Add Stress Solver to get the stress
- 1.3 Compute the Eigenvalues of the stress tensor

✔ Step 2
- 2 Influence of an empty cavity below Tête Rousse Glacier (diagnostic)
- 2.1 Apply a water pressure in the cavity

✔ Step 3
- 3 Rate of closure of the cavity for a given drainage scenario (prognostic)
- 3.1 Add a drainage scenario
Step 2: Add the cavity (empty)

The initial problem

The new problem

Two BCs for the base:

\[ u = 0 \text{ if } z_b = b \]
\[ \sigma_{nn} = p_w \text{ if } z_b > b \]
\[ p_w = 0 \text{ if the cavity is empty of water} \]
Step 2: Make a new mesh

The bottom surface is now given by the DEM DEM_TR_cavity.dat

We will use the cavity contour to have smaller size elements in the vicinity of the cavity

Work to do: modify the Makegeo.m file to create this new mesh. Don't forget to modify the mesh_input.dat

Visualize the mesh in ElmerPost, you should see the cavity! What does it change in term of velocity and stress?
Step 2: Make a new mesh

The new mesh_input.dat

! Name of the mesh directory (should exist)
WithCavity
! Surface DEM name file
DEM_TR_surf.dat
! Nsx, Nsy
268  118
! xs0, ys0
947700.0 2104850.0
! lsx, lsy
800.0 350.0
! Bedrock DEM name file
DEM_TR_cavity.dat
! Nbx, Nby
301  176
! xb0, yb0
947700.0 2104850.0
! lbx, lby
600.0 350.0
! Minimum Ice Thickness
1.0
Step 2: Make a new mesh (Makegeo_2.m) 1/2

```matlab
clear;
lc_out=18.0;
lc_in=6.0;
A=dlmread('Contour_TR_glacier.dat');
B=dlmread('Contour_TR_cavity.dat');
fid1=fopen('teterousse.geo','w');
As=size(A,1);
Bs=size(B,1);
np=0;
for ii=1:As
    np=np+1;
    fprintf(fid1,'Point(%g)={%14.7e,%14.7e,0.0,%g}; \n',np,A(ii,1),A(ii,2),lc_out);
end
for ii=1:Bs
    np=np+1;
    fprintf(fid1,'Point(%g)={%14.7e,%14.7e,0.0,%g}; \n',np,B(ii,1),B(ii,2),lc_in);
end
fprintf(fid1,'Spline(1)={
');
for ii=1:As
    fprintf(fid1,'%g,',ii);
end
fprintf(fid1,'%g}; \n',1);

fprintf(fid1,'Spline(2)={
');
for ii=1:Bs
    fprintf(fid1,'%g,',As+ii);
end
fprintf(fid1,'%g}; \n',As+1);
```

Step 2: Make a new mesh (Makegeo_2.m) 2/2

```matlab
fprintf(fid1,'Line Loop(3)={1}; \n');
fprintf(fid1,'Line Loop(4) = {2}; \n');
fprintf(fid1,'Plane Surface(5) = {3, 4}; \n');
fprintf(fid1,'Plane Surface(6) = {4}; \n');
fprintf(fid1,'Physical Line(7) = {1}; \n');
fprintf(fid1,'Physical Surface(8) = {5,6}; \n');
fclose(fid1)

% create teterousse.msh using gmsh
!gmsh teterousse.geo -1 -2

% convert teterousse.gmsh in an Elmer type mesh
!ElmerGrid 14 2 teterousse.msh -autoclean

% Extrude vertically the mesh (1m thick)
!ExtrudeMesh teterousse WithCavity 14 1 1 0 0 0 0

% Deform vertically using the surface and bedrock DEM
% Input data are in mesh_input.dat
!./MshGlacierDEM

% Make a .ep to visualize in ElmerPost the mesh
!ElmerGrid 2 3 WithCavity
```
Step 2: Change in the basal BC

The basal BC will be of the form:

\[
\text{Velocity 1} = \text{Real} \ 0.0 \\
\text{Velocity 1 Condition} = \text{Variable Coordinate 1} \\
\text{Real Procedure } "./\text{USF TR}" \ "MaskCavity"
\]

And the same for Velocity 2 and Velocity 3.

The user function MaskCavity returns +1 where \( z_b = b \), -1 where \( z_b > b \)
\( z_b \) is the node altitude, \( b \) is given by the bedrock DEM

**Work to do**: modify the SIF (Bed BC + mesh name + name)

write the user function USF_TR.f90 using the same interpolation as in MSH_Glacier3DGrille.f90

Compile it: elmerf90 USF_TR.f90 –o USF_TR

Visualize the results in ElmerPost.

What does it change in term of velocity and stress?
Step 2: MaskCavity

The user function has to do:

- Evaluate for the called node \((x,y,z)\), the bedrock altitude (Copy and paste from MshGlacierDEM.f90)

- Then, if \(z > \text{bed}\), return -1, else if \(z=\text{bed}\) returns +1

- To save time, the reading of the DEM is done only the first time and \(xb,yb,zb\) are saved.

\[
\begin{align*}
x &= \text{Model} \% \text{Nodes} \% x(\text{nodenumber}) \\
y &= \text{Model} \% \text{Nodes} \% y(\text{nodenumber}) \\
z_{\text{node}} &= \text{Model} \% \text{Nodes} \% z(\text{nodenumber}) \\
\text{IF} \ (z_{\text{node}} > \text{Zbed}+0.1) \ \text{THEN} \\
  &\quad \text{Mask} = -1.0 \\
\text{ELSE} \\
  &\quad \text{Mask} = 1.0 \\
\text{END IF}
\end{align*}
\]
Modelling Tête Rousse Glacier

✓ Step 1
- 1 Tête Rousse Glacier flow without a water filled-cavity (diagnostic)
- 1.2 Add Stress Solver to get the stress
- 1.3 Compute the Eigenvalues of the stress tensor

✓ Step 2
- 2 Influence of an empty cavity below Tête Rousse Glacier (diagnostic)
- 2.1 Apply a water pressure in the cavity

✓ Step 3
- 3 Rate of closure of the cavity for a given drainage scenario (prognostic)
- 3.1 Add a drainage scenario
Step 2.1: Add a water pressure

Modify the SIF to add a water pressure

\[ p_w = \rho_w g (h_w - z) \]

\[ \text{water load } h_w \]

\[ \text{the water load } \]

$hw = 3170.0$

In the bedrock BC

Flow Force BC = Logical True

External Pressure = Variable Coordinate 3

Real MATC \\
"-rhow*gravity*(hw-tx)"

will only apply where a Dirichlet BC is not applied
Modelling Tête Rousse Glacier

✓ Step 1
- 1 Tête Rousse Glacier flow without a water filled-cavity (diagnostic)
- 1.2 Add Stress Solver to get the stress
- 1.3 Compute the Eigenvalues of the stress tensor

✓ Step 2
- 2 Influence of an empty cavity below Tête Rousse Glacier (diagnostic)
- 2.1 Apply a water pressure in the cavity

✓ Step 3
- 3 Rate of closure of the cavity for a given drainage scenario (prognostic)
- 3.1 Add a drainage scenario
Step 3: Move to prognostic

Will do it in two steps

- Move to prognostic assuming the cavity is empty of water at $t=0$
  (big step, need 3 new solvers!)

- Prescribe the observed drainage scenario for the water pressure

To move from a diagnostic to a prognostic simulations:

- Add the FreeSurface solver (here 2 times, since we have 2 FS)
- Add the MeshUpdate solver to deform the mesh from the FS displacements
- Add one body per FS (new Initial Condition and Equation Sections)
- Modifications in the Simulation and Boundary Condition Sections

Only shown for the upper free surface here
Step 3 – Steady to transient

The simulation Section has to be modified:

Simulation Type = Transient
Timestepping Method = “bdf” —> Backward Differences Formulae
BDF Order = 1
Output Intervals = 1 —> Save in .ep file
Timestep Intervals = 200
Timestep Sizes = 1.0
Steady State Min Iterations = 1
Steady State Max Iterations = 1 —> To control the “implicity” of the solution over one time step (here 1 means explicite)
Step 3 – Sketch of a transient simulation

Geometry + Mesh → Degrees of freedom

Linear System: $A . x = b$
- Direct
- Iterative $\epsilon_L$

Non Linear iterations: $A = A(x)$

Coupled iteration over $dt$ (implicit scheme)

$t = t + dt$

$\epsilon_L < \epsilon_{NL} < \epsilon_C$
Step 3 – Free surface Solver

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

Define a 2nd body which is the boundary 3.

Body 2
   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.

Tell in BC2 that this is the body 2:

Boundary Condition 2
   Target Boundaries =2
   Body Id = 2
   ...
End
Step 3 – Add the Free surface Solver

Solver 4
Equation = "Free Surface Top"
Variable = String "Zs Top"
Variable DOFs = 1
Exported Variable 1 = String "Zs Top Residual"
Exported Variable 1 DOFs = 1

Procedure = "FreeSurfaceSolver" "FreeSurfaceSolver"
Before Linsolve = "EliminateDirichlet" "EliminateDirichlet"

Linear System Solver = Iterative
Linear System Max Iterations = 1500
Linear System Iterative Method = BiCGStab
Linear System Preconditioning = ILU0
Linear System Convergence Tolerance = Real 1.0e-9
Linear System Abort Not Converged = False
Linear System Residual Output = 1

Nonlinear System Max Iterations = 100
Nonlinear System Convergence Tolerance = 1.0e-6
Nonlinear System Relaxation Factor = 1.00

Steady State Convergence Tolerance = 1.0e-03

Stabilization Method = Bubbles
Apply Dirichlet = Logical False

! How much the free surface is relaxed
Relaxation Factor = Real 1.00

! Is there a maximum step-size for the displacement
! use/or not accumulation
Use Accumulation = Logical True

! take accumulation to be given normal to surface/as vector
Normal Flux = Logical False

End
Step 3 – Upper Surface

Body Force 2:

```plaintext
Body Force 2
  Zs Top Accumulation Flux 1 = Real 0.0e0
  Zs Top Accumulation Flux 2 = Real 0.0e0
  Zs Top Accumulation Flux 3 = Real 0.0e0
End
```

Equation 2:

```plaintext
Equation 2
  Active Solvers(1) = 2
  Flow Solution Name = String “Flow Solution”
  Convection = String Computed
End
```

Initial Condition 2: (tell that $z_s(x,0) = \text{ordinate of the initial top surface}$)

```plaintext
Initial Condition 2
  Zs Top = Variable Coordinate 1
  Real Procedure “ElmerIceUSF” “ZsTopIni”
End
```

Have a look to this User Function in `elmerice/UserFunctions`
Step 3 – Mesh Update Solver

Add the Mesh Update Solver:

Solver 6
  Equation = “Mesh Update”
  Linear System Solver = “Direct”
  Linear System Direct Method = umfpack
  Steady State Convergence Tolerance = 1.0e-04
End

Material parameter for this solver:

  Mesh Youngs Modulus = Real 1.0
  Mesh Poisson Ratio = real 0.3

Force that Mesh Update 1 and 2 =0 everywhere:

Body Force 1
  Mesh Update 1 = Real 0.0
  Mesh Update 2 = Real 0.0
End
Step 3 – Mesh Update Solver BC

In all Boundary conditions:

Mesh Update 1 = real 0.0
Mesh Update 2 = real 0.0

For the upper surface BC (2):

Mesh Update 3 = Variable Zs Top
Real Procedure "ElmerIceUSF" "ZsTopMzsIni"
Step 3 – Same for the bedrock

Name of the variable: Zs Bottom
Add solver: Solver 5
Add equation: Equation 3

For the Bottom surface BC (3):

Boundary Condition 1
Target Boundaries = 1
Body Id = 3
Mesh Update 3 = Variable Zs Top
    Real Procedure "ElmerIceUSF" "ZsBottomMzsIni"
End

Add a limiter to ensure that $z_b \geq b$
In the material section

Min Zs Bottom = Variable Coordinate 1
    Real Procedure "/USF_TR" "MinZsBottom"
Max Zs Bottom = Real +1.0e10

+ in the Free Surface solver: Apply Dirichlet = Logical True

Same as MaskCavity but return zbed
Step 3 – Newton linearization

If you want to use Newton linearization 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 1

Nonlinear System Reset Newton = Logical False

End
Modelling Tête Rousse Glacier

✓ Step 1
- 1 Tête Rousse Glacier flow without a water filled-cavity (diagnostic)
- 1.2 Add Stress Solver to get the stress
- 1.3 Compute the Eigenvalues of the stress tensor

✓ Step 2
- 2 Influence of an empty cavity below Tête Rousse Glacier (diagnostic)
- 2.1 Apply a water pressure in the cavity

✓ Step 3
- 3 Rate of closure of the cavity for a given drainage scenario (prognostic)
- 3.1 Add a drainage scenario
Step 3.1 – Add a drainage scenario

Add an evolution of the water load of the form:

\[ h_w = 3170.0 - t \times \Delta h_w / \Delta t \]

Work to do:
Write a MATC function \( hw \) to prescribe the water load evolution
Step 3.1 – Add a drainage scenario

MATC function $h_w$ to prescribe the water load evolution:

! Water load function of time (in year)
! Decrease by DH in DT
$ function \quad h_w(t) \{ \$
\quad DH = 70.0; \$
\quad DT = 20.0; \$
\quad _{hw} = 3170.0 - t*365.25*DH/DT ; \$
$ \}

Call in the bedrock BC

External Pressure = Variable time, Coordinate 3
Real MATC "rhow*gravity*(h_w(tx(0))-tx(1))"
More Steps ???

Some idea:

- go to parallel

- add the Savedata solver to get upper and lower surfaces output

- ...

- ...
References