

Program

- Novinky
- Relaxace kráteru – připomenutí úlohy (gravitačně indukovaná relaxace topografie viskózního nebo viskoelastického materiálu, úloha na volný povrch)
- Viskózní tečení
 - COMSOL
 - Elmer
- Visko-elasticita
 - COMSOL

Novinky

ElmerGUI a ElmerPost nyní nově i na geof30 (doposud pouze geof40)

Relaxace kráteru**Viskózní tečení – Elmer**

Úloha nelze implementovat v ElmerGUI (zatím), budeme muset ručně připravit .sif file

1.) Mesh

- Buď ElmerGridem vytvoříme jednoduchou geometrii (čtverec) a ten namapovat na počáteční profil
- Import sítě z COMSOLu:
 - V COMSOLU vytvořit síť
 - File=> Export => Mesh to file , uložit jako textový nikoli binární soubor (.mphtxt)
 - Konverze do elmer mesh formátu:
ElmerGrid 9 2 název.mphtxt
 - Vytvořenou síť můžeme zkontrolovat v ElmerGUI (File=>Load Mesh)

2.) Tvorba Solver Input File (sif)

- Použijeme sif file z podobného problému (volný povrch 2D ledovce), který vhodně modifikujeme
- Crater.sif :

```
check keywords warn
echo on
```

```
Header
```

```
  Mesh DB "." "mesh_crater"  !název sítě
End
```

```
Constants
```

```
! No constant Needed
End
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
Simulation
```

```
  Coordinate System = Cartesian 2D
  Simulation Type = Transient
  Timestepping Method = "bdf"
  BDF Order = 1
  Output Intervals = 1
  Timestep Intervals = 100
  Timestep Sizes = 0.01
  Steady State Min Iterations = 1
  Steady State Max Iterations = 10
  Post File = "crater.ep"
  max output level = 3
End
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
! The flowing material
Body 1
```

```
  Equation = 1
  Body Force = 1
  Material = 1
```

```
Initial Condition = 1
End
! The Free surface      !volný povrch je třeba zadat jako samostatné těleso
Body 2
Equation = 2
Body Force = 2
Material = 1
Initial Condition = 2
End
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Initial Condition 1
Pressure = Real 0.0
Velocity 1 = Real 0.0
Velocity 2 = Real 0.0
End
```

```
Initial Condition 2
Zs = Variable Coordinate 1 !počáteční polohu
real Procedure "USF_Zs" "ZsIni"
End
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Body Force 1
Flow BodyForce 1 = Real 0.0
Flow BodyForce 2 = Real -9.7696 !MPa - a - m
Mesh Update 1 = real 0.0
End
```

```
!! accumulation flux in m/year
Body Force 2
Zs Accumulation Flux 1 = Real 0.0e0
Zs Accumulation Flux 2 = Real 0.0e0
End
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Material 1
Density = Real 1.
Viscosity = 1
Mesh Youngs Modulus = Real 1.0
Mesh Poisson Ratio = real 0.3
End
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Solver 1
Equation = "Navier-Stokes"
Stabilization Method = String Bubbles
Linear System Solver = Direct
Linear System Direct Method = umfpack
Nonlinear System Max Iterations = 100
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 = 1.00
Steady State Convergence Tolerance = Real 1.0e-3
End
```

```
Solver 2 !připojený solver pro výpočet volného povrchu
Equation = "Free Surface"
Variable = String Zs !deklarace proměnné nového solveru – vertikální komponenta povrchu
Variable DOFs = 1
Exported Variable 1 = String "Zs Residual"
Exported Variable 1 DOFs = 1
Procedure = ".FSurfSolverLimited" "FSurfSolverLimited" !název soboru a procedury, kterou má solver volat
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-5
Linear System Abort Not Converged = False
Linear System Residual Output = 1
Nonlinear System Max Iterations = 1
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 0.90
End
```

```
Solver 3 !solver pro remeshing
Equation = "Mesh Update"
Linear System Solver = "Direct"
Linear System Direct Method = umfpack
Steady State Convergence Tolerance = 1.0e-04
End
```

```
!Solver 4
! Exec Solver = After TimeStep ! For transient simualtion
! Procedure = File "/MySaveData" "SaveScalars"
! Filename = "ismip_scalars.dat"
! File Append = Logical True ! For transient simualtion
! Variable 1 = String "Time"
! Variable 2 = String "Flow Solution"
! Operator 2 = String "Volume"
! Variable 3 = String "Mesh Velocity 2"
! Operator 3 = String "Max Abs"
! Variable 4 = String "Flow Solution"
! Operator 4 = String "Convective flux"
! Variable 5 = String "cpu time"
! Variable 6 = String "cpu memory"
!End
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Equation 1 !rovnice pro Body 1 – tečící materiál
Active Solvers(2) = 1 3
NS Convect = Logical False
End
```

```
Equation 2 !rovnice pro Body 2 – volný povrch
Active Solvers(1) = 2
Flow Solution Name = String "Flow Solution" !keyword pro freesurface solver – odkud brát rychlost
Convection = String Computed
End
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
! Bedrock
Boundary Condition 1
Target Boundaries(1) = 2
! Target Boundaries(1) = 1
Velocity 1 = Real 0.0e0
Velocity 2 = Real 0.0e0
Mesh Update 1 = real 0.0
Mesh Update 2 = real 0.0
End
```

```
! Walls
Boundary Condition 2
Target Boundaries(4) = 1 3 6 7
Velocity 1 = Real 0.0e0
Velocity 2 = Real 0.0e0
Mesh Update 1 = real 0.0
Mesh Update 2 = real 0.0
End
```

```
! Upper Surface
Boundary Condition 3
!!! this BC is equal to body no. 2 !!!
Body Id = 2
Target Boundaries(4) = 4 5 8 9
Mesh Update 1 = Real 0.0
Mesh Update 2 = Variable Zs
Real Procedure "/USF_Zs" "ZsMZsIni"
End
```

V síř řílu jsou se objevují dvě procedury

```
Real Procedure "/USF_Zs" "ZsMZsIni"
a
Procedure = "/FSurfSolverLimited" "FSurfSolverLimited"
```

Obě máme k dispozici jako zdrojové kódy (.f90), které je potřeba přeložit:
elmerf90 FsurfSolverLimited.f90 -o FsurfSolverLimited
elmerf90 USF_Zs.f90 -o USF_Zs

a umístit do adresáře k sif filu

pozn. elmerf90 je fortranský wrapper skript na kompilování vlastních podprogramů pomocí kompilera, kterým byl kompilován Elmer při instalaci – výstupem jsou .so (shared object – dynamically linked library)

3.) Spuštění solveru

- **ElmerSolver crater.sif**
- Nebo vytvořit soubor ELMERSOLVER_STARTINFO v adresáři, kde máme sif soubor a do něj vepsat Crater.sif

1

a pak stačí v daném adresáři volat **ElmerSolver**

4.) Prohlížení výsledků – ElmerPost

- **ElmerPost**
- File=>Open=>crater.ep
- Select Timesteps=> All
- Color Mesh
 - Mesh Style:Surface
 - Color variable: Velocity_abs
- ElmerPost defaultně zobrazuje v referenční geometrii, chceme-li vidět, jak se těleso hýbe, musíme provést následující:
 - Do okna Elmer-Post vepsat: **math on=nodes**
 - Edit=>TimestepControl
Do „Do After Frame“:
math nodes(0:2,time(0)) = on(0:2,time(0)) +
Mesh.Update(0:2,time(\$t))
 - Pak Loop

Zatím jsme měli na hranicích vlevo a vpravo no-slip, raději bychom free-slip:

Modifikace crater.sif:

! Walls

Boundary Condition 2

Target Boundaries(4) = 1 3 6 7

Velocity 1 = Real 0.0e0

Mesh Update 1 = real 0.0

Mesh Update 2 = Variable Zs

Real Procedure "/USF_Zs" "ZsMZsIni"

End

To celé vypadá složitěji, než v COMSOLu, pojdme proto využít síly Elmeru, tkvící v možnosti hezky paralelizovat:

- V COMSOLu nejprve řádně zahustíme síť, aby se nám nasazení paralelizace vyplatilo
- Opět síť vyexportujeme a převedeme do elmermesh formátu pomocí ElmerGridu
- Použijeme metis knihovnu k rozdělení oblasti na 8 podoblastí:
ElmerGrid 2 2 mesh_crater_fine -metis 8
- Vizuální kontrola: **ElmerGrid 2 3 mesh_crater_fine -metis 8 -out mesh_par**

Zatím bez úspěchu – segmentation fault, čekáme na vyjádření s Helsinek

Zatím tedy zkusme něco jiného:

Depth-dependent viscosity