Seminář software pro geofyziky Jednoocí slepým 30.10.2008

Program

- Relaxace kráteru znovu a paralelně
- Zvon – formulace úlohy (Eliška Z.)
- Vlastní kmity zvonu
  - COMSOL
  - Elmer

Relaxace kráteru znovu a paralelně

Od minula máme připravenou hustou síť pro kráter
Mesh_crater_fine.mphtxt (COMSOL)
Převedeme do Elmer Mesh formátu:

```
ElmerGrid 9 2 mesh_crater_fine.mphtxt
```

Prohlédneme pomocí ElmerGUI

Provedeme rozdělení oblasti na 4 podoblasti pro paralelizaci
K dispozici buď knihovna metis:

```
ElmerGrid 9 2 mesh_crater_fine.mphtxt –metis 4
```

, nebo dělení podle souřadných os – to se osvědčilo lépe:

```
ElmerGrid 9 2 mesh_crater_fine.mphtxt –partition 4 1 1
```

Výsledek dělení si prohlédneme v ElmerPost:

```
ElmerGrid 9 3 mesh_crater_fine.mphtxt –partition 4 1 1 –out mesh_par
```

ElmerPost => Open mesh.par, Color Mesh Edit => Color Variable : Partition, Mesh style: Surface

Nyní k vlastní paralelizaci
Není potřeba jakkoli měnit sif file. Pouze musíme jinak volat – zapojit MP:
Vytvoříme spustitelný soubor run.mpi a vepíšeme

```
mpirun -np 4 -x LD_LIBRARY_PATH -hostfile hosts ElmerSolver_mpi
```

4 = počet threadů – vláken = počet podoblastí, na něž jsme rozdělili výpočetní oblast
dále vytvoříme soubor ELMERSOLVER_STARTINFO
do kterého vepíšeme název našeho sif filu: crater_par.sif

a konečně soubor hosts, který bude obsahovat názvy stanic pro MP, kam se má úloha distribuovat (k dispozici geof 30, geof 40, geof50) – na vepsané stroje musíme mít nastaven ssh přístup bez hesla – viz Františkův návod v jednookých

Spustíme paralelní výpočet
./run.mpi

Výsledek je nyní ve 4 output .ep souborech, do jednoho jej složíme příkazem

```
ElmerGrid 15 3 crater
```

Volaným v adresáři s našimi 4-mi výstupními soubory
Dílo si prohlédneme ElmerPost em

Vlastní kmity zvonu

- Comsol

Shrnutí modelu– viz report
Pracovní postup:

New => Axial symmetry (2D), Structural mechanics module => Axial symmetry, stress, strain
Import geometrie z CADu – soubor zvon.dxf
Import => CAD data from file
Umístit na osu symetrie, naškálovat, mesh, hraniční podmínky – free surface všude (=nekonečně tenká dírka nahoře v čepci zvonu), mesh, Solve

Pozn. – pro lepší rozlišení (menší tolerance parametr pro linear solver) je vhodné zadat v Solver Parameters => General => Search for eigenfrequencies around: zde první "normální" vlastní frekvenci, parazitická nultá je velmi malá a odpovídá kmitům celého zvonu bez deformace, kazí nám přesnost neb díky velmi malé hodnotě je relativní chyba pro tuto frekvenci velká, tímto zadáním je možné jít s tolerancí níž, výsledky se podstatně zestabilní

- Elmer

Importujeme geometrii z comsolu
`ElmerGrid 9 2 zvon_maly.mphtxt`
Otevřeme v ElmerGUI a pomocí Ctrl+poklik postupně označíme segmenty hranice a tlačítkem unify edge (na liště vedle S = Solveru) vztvoříme hranici z pouze dvou segmentů – plášť a stěna díry v čepci
Uložíme ! : File=> Save

Dále jsme vytvořili následující sif file (zvon.sif):

```
Header
  CHECK KEYWORDS Warn
  Mesh DB "." "zvon_maly"
  Include Path ""
  Results Directory ""
End

Simulation
  Coordinate System = "Axi Symmetric"
  Coordinate Mapping(3) = 1 2 3
  Simulation Type = "Steady State"
  Steady State Max Iterations = 1
  Solver Input File = "zvon.sif"
  Output File = "zvon.dat"
  Post File = "zvon.ep"
End

Body 1
  Equation = 1
  Material = 1
End

Material 1
  Youngs Modulus = 2e9
  Poisson Ratio = 0.33
  Density = real 7850.00
End

Equation 1
  Stress Analysis = True
End

Solver 1
  Equation = "Stress Analysis"
  Eigen Analysis = Logical True
  Eigen System Values = Integer 10
  Linear System Solver = "direct"
  Variable = "Displacement"
  Variable Dofs = 3
```

```
  Linear System Iterative Method = "BiCGStab"
  Linear System Max Iterations = 10000
  Linear System Convergence Tolerance = 1.0e-10
  Linear System Abort Not Converged = True
  Linear System Preconditioning = "ILU1"
  Linear System Residual Output = 1
  Steady State Convergence Tolerance = 1.0e-05
  Nonlinear System Convergence Tolerance = 1.0e-05
  Nonlinear System Max Iterations = 1
  Nonlinear System Newton After Iterations = 3
  Nonlinear System Newton After Tolerance = 1.0e-02
  Nonlinear System Relaxation Factor = 1
  Linear System Precondition Recompute = 1
End

Boundary Condition 1
  Target Boundaries(1) = 1
!  Displacement 1 = 0
!  Displacement 2 = 0
!  Displacement 3 = 0
End

Boundary Condition 2
  Target Boundaries(1) = 2
End
```

Pustíme : **`ElmerSolver zvon.sif`**
Výstupem:

```
EigenSolve: Computed Eigen Values:
EigenSolve: -------------------------------
EigenSolve:           1 (3.978088729624121E-008,0.000000000000000E+000)
EigenSolve:           2 (453436.545836348,0.000000000000000E+000)
EigenSolve:           3 (1227248.10079085,0.000000000000000E+000)
EigenSolve:           4 (2510611.30287463,0.000000000000000E+000)
EigenSolve:           5 (3110317.04321625,0.000000000000000E+000)
EigenSolve:           6 (4058468.76756696,0.000000000000000E+000)
EigenSolve:           7 (5451357.58613953,0.000000000000000E+000)
EigenSolve:           8 (5801964.95082970,0.000000000000000E+000)
EigenSolve:           9 (8372493.45659205,0.000000000000000E+000)
EigenSolve:          10 (12350743.7703284,0.000000000000000E+000)
EigenSolve: -------------------------------
```

Trvalo chvíli si uvědomit, že výstupem obecného EigenSolveru jsou vlastní čísla daného problému, nikoli vlastní frekvence, jak bychom si přáli, tato vlatní čísla jsou $(2*pi*f)^2$
Po přepočtení:

```
 1   3.174368584596E-005
 2   1.071712706540E+002
 3   1.763137965378E+002
 4   2.521795502221E+002
 5   2.806871108583E+002
 6   3.206278520359E+002
 7   3.715972225611E+002
 8   3.833607778348E+002
 9   4.605189854484E+002
 10  5.593281714630E+002
```

Vizualizace:
Pro vizualizaci pou6ijeme ElmerPost a skript anim.cmd:

```
# how many cycles to go trhough
set ncycles 50

# how many frames  in total
set frames  5000

# how to scale displacements
set scaleFactor 0.04

math nsave=nodes;
do i 0 [@ $frames-1] {
  math w= 2*pi*$i*$ncycles/$frames;
  math nodes=nsave+$scaleFactor*Displacement*sin(w);
  display;
 }
```

Skript pustíme z ElmerPostu po načtení modelu příkazem

```
source anim.cmd
```

v hlavním příkazovém okně.