Seminář software pro geofyziky Jednoocí slepým 10.4.2012

Fluid-structure interaction

Praktické ukázky

Program

- Obtékání elastické překážky Newtonovskou kapalinou (2D)

Elmer

Rozebereme příklad z http://www.nic.funet.fi/pub/sci/physics/elmer/doc/ElmerTutorials.pdf (str. 40) v jeho původní verzi bez použití GUI. Archiv tutoriál souborů stáhneme z http://www.nic.funet.fi/pub/sci/physics/elmer/doc/ElmerTutorialsFiles_nonGUI.tar.gz, otevřeme FluidStructureBeam

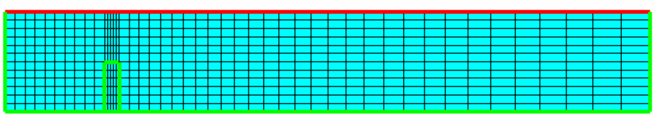
1.) Příprava sítě

fsi.grd:

```
***** ElmerGrid input file for structured grid generation *****
Version = 210903
Coordinate System = Cartesian 2D
Subcell Divisions in 2D = 5 4
Subcell Limits 1 = -1 0 10 11.5 65 100
Subcell Limits 2 = -1 0 5 10 100
Material Structure in 2D
  3    3    3    3    3
  6    1    1    1    4
  6    1    2    1    4
  5    5    5    5    5
End
Materials Interval = 1 2   !pouze pro materiál mezi 1 a 2 se vygeneruje mesh
Boundary Definitions
! type     out      int
  1        6        1        0
  2        4        1        0
  3        2        1        0
  4        3        1        0
  5        5        1        0
  6        5        2        0
End
Numbering = Vertical
Element Degree = 2
Element Innernodes = True
Triangles = False
Surface Elements = 500
Coordinate Ratios = 1
Minimum Element Divisions = 1 1
Element Ratios 1 = 1 1 1 3 1
Element Ratios 2 = 3 1 1 1
Element Densities 1 = 1 1 3 1 1
Element Densities 2 = 1 1 1 1
```

Příkazem

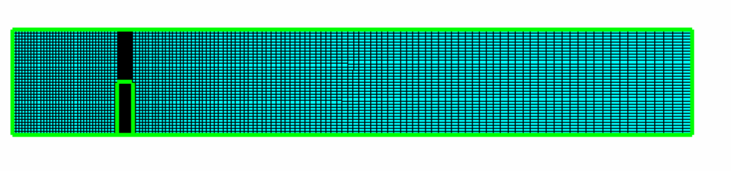
`ElmerGrid 1 2 fsi.grd`



vygenerujeme síť, kterou můžeme nahlédnout v pomocí ElmerGUI.

Chceme-li síť jemnější, změníme např. Surface Elements = **5000** a vygenerujeme jemnější síť

`ElmerGrid 1 2 fsi.grd -out fsi2`



2.) Řešení

Nyní si prohlédneme fsi.sif:

```
Header
  Mesh DB "." "fsi2"                       !cesta k síti - p•ípadn• zm•níme na fsi2
End

Constants
End

Simulation
  Coordinate System = Cartesian 2D
  Simulation Type = Steady State
  Steady State Max Iterations = 50
  Steady State Min Iterations = 2
  Output Intervals = 1
  Output File = "fsi.result"
  Post File = "fsi.ep"
End

Body 1
  Equation = 1
  Material = 1
End

Body 2
  Equation = 2
  Material = 2
End

Material 1
   Density = 1.0
   Viscosity = 0.5
   Poisson Ratio = 0.3

   Youngs Modulus = Variable time
         Real Procedure "FsiStuff" "Youngs" ! Young•v modul pružnosti pro sí• je p•edepsán
       End                                  ! v externí fci Youngs v souboru FsiStuff

Material 2
   Density = 1000
   Youngs Modulus = 1.0e3
   Poisson Ratio = 0.3
End

Solver 1
  Equation = Navier-Stokes
  Stabilize = True
  Linear System Solver = Iterative
  Linear System Iterative Method = BiCGStab
  Linear System Preconditioning = ILU1
  Linear System Max Iterations = 500
  Linear System Convergence Tolerance = 1.0e-8
  Nonlinear System Max Iterations = 10
  Nonlinear System Convergence Tolerance = 1.0e-5
  Nonlinear System Newton After Tolerance = 1.0e-5
  Nonlinear System Newton After Iterations = 20
  Nonlinear System Relaxation Factor = 1.0
  Steady State Convergence Tolerance = 1.0e-4
End

Solver 2
  Equation = Elasticity Solver
  Variable = Displacement
  Variable DOFs = 2
  Procedure = "ElasticSolve" "ElasticSolver"
  Linear System Solver = Direct
  Linear System Direct Method = Banded
  Nonlinear System Newton After Tolerance = 1.0e-3
  Nonlinear System Newton After Iterations = 20
  Nonlinear System Max Iterations = 100
  Nonlinear System Convergence Tolerance = 1.0e-5
  Nonlinear System Relaxation Factor = 1.0
```

```
  Steady State Convergence Tolerance = 1.0e-4
End

Solver 3
  Equation = Mesh Update
  Linear System Solver = Iterative
  Linear System Iterative Method = BiCGStab
  Linear System Preconditioning = ILU1
  Linear System Max Iterations = 500
  Linear System Convergence Tolerance = 1.0e-8
  Steady State Convergence Tolerance = 1.0e-4
End

Equation 1
  Active Solvers(2) = 1 3
End

Equation 2
  Active Solvers = 2
  Plane Stress = True
End

Boundary Condition 1
  Target Boundaries = 1

  Velocity 1 = Variable Time
      Real Procedure "FsiStuff" "InFlow"
  Velocity 2 = 0.0
  Mesh Update 1 = 0.0
End

Boundary Condition 2
  Target Boundaries = 2
  Velocity 2 = 0.0
  Pressure = 0.0
  Mesh Update 1 = 0.0
End

Boundary Condition 3
  Target Boundaries = 3
  Velocity 1 = 0.0
  Velocity 2 = 0.0
  FSI BC = True
  Mesh Update 1 = Equals Displacement 1
  Mesh Update 2 = Equals Displacement 2
End

Boundary Condition 4
  Target Boundaries(2) = 4 5
  Velocity 1 = 0.0
  Velocity 2 = 0.0
  Mesh Update 2 = 0.0
End

Boundary Condition 5
  Target Boundaries = 6
  Displacement 1 = 0.0
  Displacement 2 = 0.0
End

!End
```

Kód používá dvě externí funkce: FsiStuff.f90:

!Youngův modul pružnosti pro mesh update solver:

```
FUNCTION Youngs( Model, n, x ) RESULT( s )
  USE Types
  TYPE(Model_t) :: Model
  INTEGER :: n
  REAL(KIND=dp) :: x,s,s1,s2,s3,xx,yy

  xx = Model % Nodes % x(n)
  yy = Model % Nodes % y(n)

  s =  1.0d0 / SQRT( (xx-11.0)**2 + (yy-4.9)**2 )

END FUNCTION Youngs
```

!Parabolický profil na vstupu postupně lineárně zapínaný v čase

```
FUNCTION InFlow( Model, n, x ) RESULT( vin )
  USE Types
```

```
  TYPE(Model_t) :: Model
  INTEGER :: n
  REAL(KIND=dp) :: yy,x,vin,v0,vt

  xx = Model % Nodes % x(n)
  yy = Model % Nodes % y(n)

  v0 = (-(yy**2)+10*yy)/25

  IF(x < 8.0) THEN
    vt = x/8.0
  ELSE
    vt = 1.0
  END IF

  vin = v0*vt
END FUNCTION InFlow
```

FsiStuff.f90 zkompilujeme pomocí

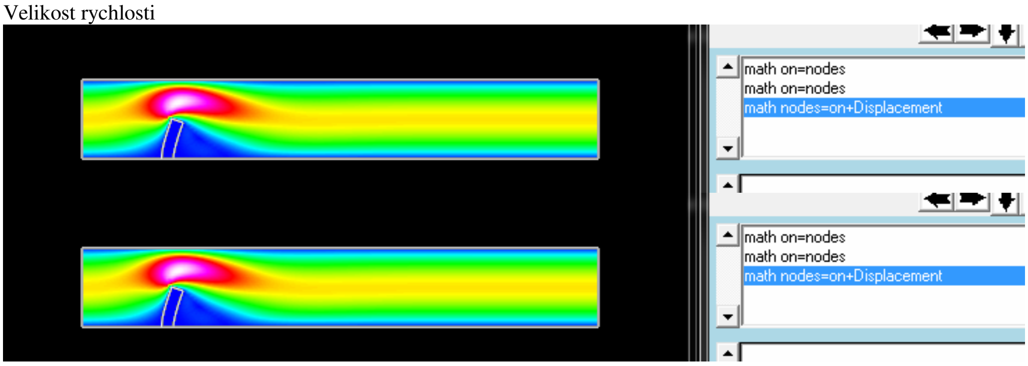
```
Elmerf90 FsiStuff.f90 -o FsiStuff
```

A spustíme výpočet:

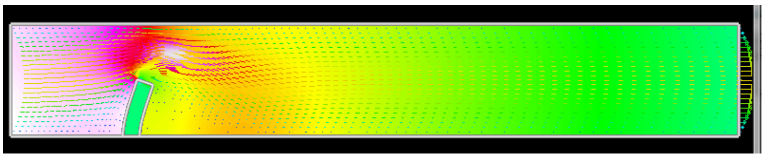
```
ElmerSolver fsi.sif
```

Výsledek vizualizujeme pomocí ElmerPost

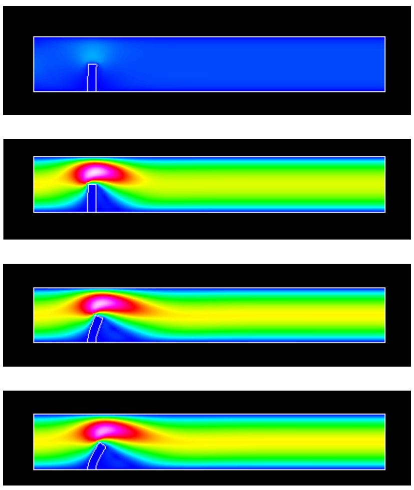
Velikost rychlosti

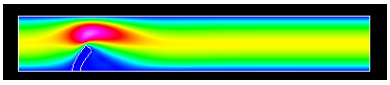


Vektorové pole rychlosti a pole tlaku



Doposud jsme se zabývali steady-state problémem. Zkusme se podívat na časovou evoluci.





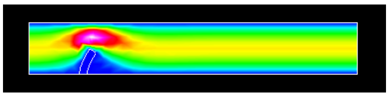
Proč nekonvergujeme k steady-state řešení? Evoluční problém pro velké časové kroky produkuje “periodické“ oscilace – elastický kvádřík se navrátil do původní polohy. Jedná se o přechodový stav a máme tedy steady-state řešení napočítané správně? Nepomohlo zvýšení viskozity, které mělo vést k rychlejší relaxaci neb jsme prestali konvergovat..

Zkusme tedy nastartovat evoluční řešič ze steady-state řešení – měli bychom v něm zůstat. V sekci Simulation doplníme

```
  Restart File = "fsi_s.result"
  Restart Position = 14
```

A v FsiStuff.f90 přidáme inflow fci už časově nezávislou (abychom si řešení opět hned nezkazili špatnou hraniční podmínkou), nahradíme v sif souboru v hraniční podmínce změníme

```
Velocity 1 = Variable Time
      Real Procedure "FsiStuff" "InFlow2"
```



Jedná se o řešení našeho problému, kvádřík přestane oscilovat a zůstane v daném (a tedy stacionárním) stavu.