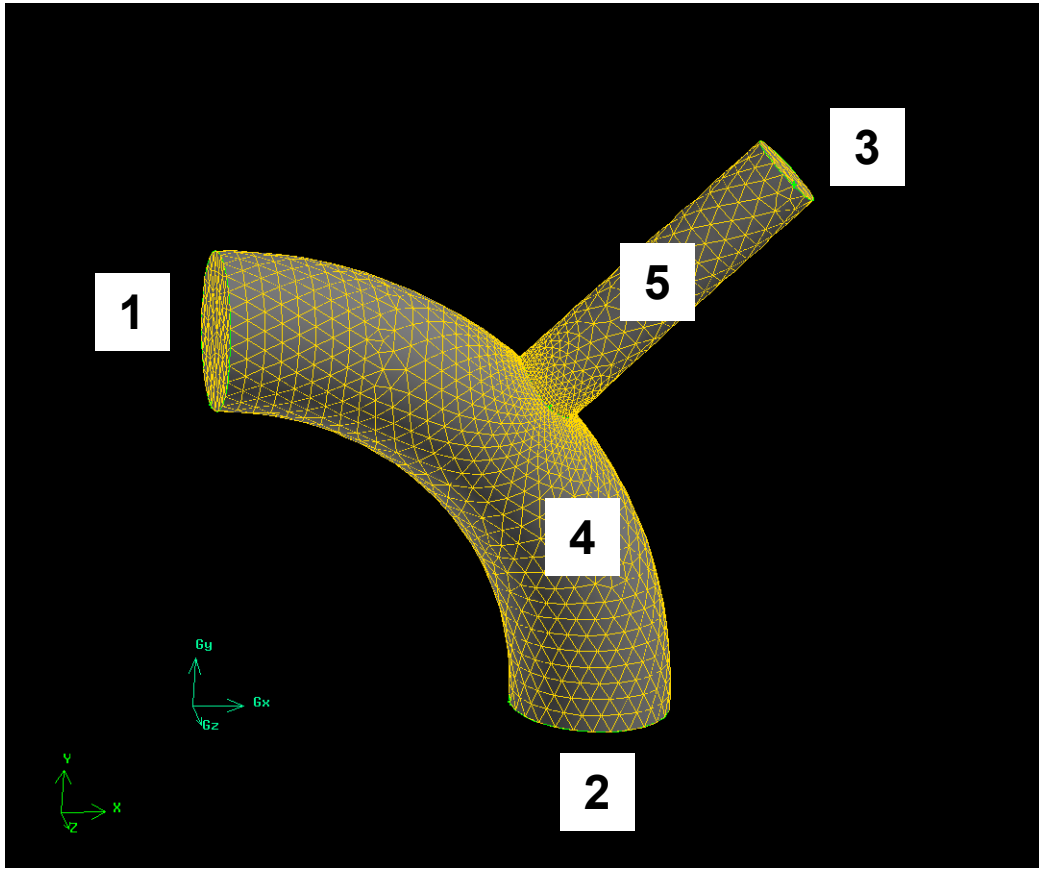
Elmer Parallel Introductory Example

T. Zwinger

CSC ï Tieteen tietotekniikan keskus Oy
CSC ï IT Center for Science Ltd.

Problem Outline

- Flow through pipe junction
- Boundary conditions:
  1. v_{in}=*1*cm/s
  2. None
  3. v_{in}=*1* cm/s
  4. + 5. noslip

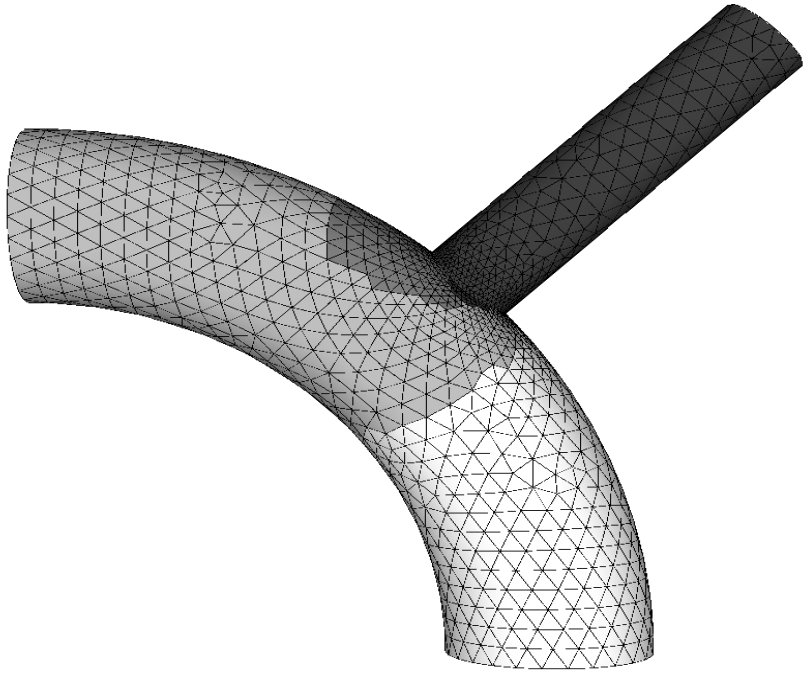


Pre-pocessing

- Splitting with ElmerGrid

```
ElmerGrid 2 2 flow
-metis 4 1
```

- Creates 4 partition mesh



Solver Input File (SIF)

```
Header
  Mesh DB "." "flow"
End

Simulation
  Coordinate System  = "Cartesian 3D"
  Simulation Type ="Steady"
  Output Intervals = 1
  Post File = "parallel_flow.ep"
  Output File = "parallel_flow.result"
  max output level = 4
End
```

the mesh directory

3D, steady-state

Solver Input File (SIF)


```
Solver 1
  Equation = "Navier-Stokes"
  Optimize Bandwidth = Logical True
  Linear System Solver = Iterative
  Linear System Iterative Method = "BiCGStab"
  Linear System Max Iterations = 500
  Linear System Convergence Tolerance = 1.0E-04
  Linear System Abort Not Converged = True
  Linear System Preconditioning = "ILU1"
  Linear System Residual Output = 1
```

Solver Input File (SIF)

```
  Steady State Convergence Tolerance = 3.0E-03

  Stabilization Method = Stabilized

  Nonlinear System Convergence Tolerance = 1.0E-03

  Nonlinear System Max Iterations = 30

  Nonlinear System Newton After Iterations = 1

  Nonlinear System Newton After Tolerance =  1.0E-03

  Nonlinear System Relaxation Factor = 1

End
```

Non-linearity from convection

Solver Input File (SIF)

```
Body 1
  Name = "fluid"
  Equation = 1
  Material = 1
  Body Force = 1
  Initial Condition = 1
End
Equation 1
 Active Solvers(1) = 1
 Convection = Computed
End
```

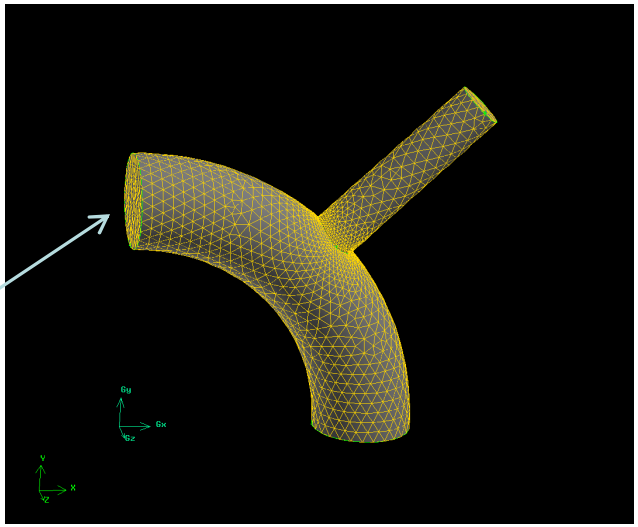
Solver Input File (SIF)

```
Initial Condition 1
  Velocity 1 = 0.0
  Velocity 2 = 0.0
  Velocity 3 = 0.0
  Pressure = 0.0
End
Body Force 1
  Flow BodyForce 1 = 0.0
  Flow BodyForce 2 = 0.0
  Flow BodyForce 3 = 0.0
End
```

Could be gravity, i.e., -9.81

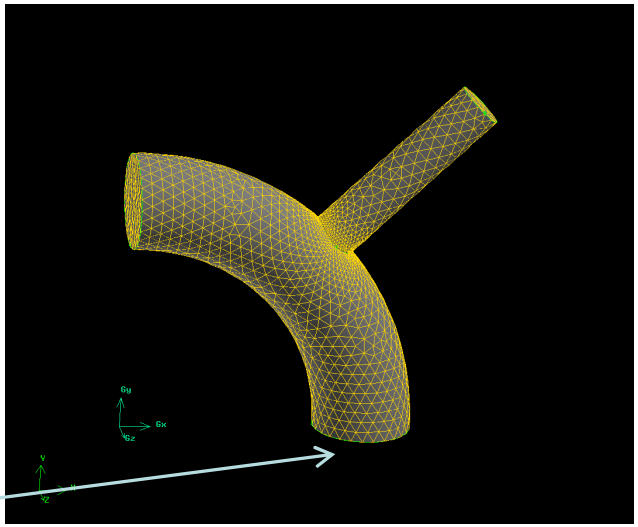
Solver Input File (SIF)

```
Material 1
  Density = 1000.0
  Viscosity = 1.0
End
Boundary Condition 1
  Name = "largeinflow"
  Target Boundaries  = 1
  Normal-Tangential Velocity = True
  Velocity 1 = -0.01
  Velocity 2 = 0.0
  Velocity 3 = 0.0
End
```



Outward pointing normal!

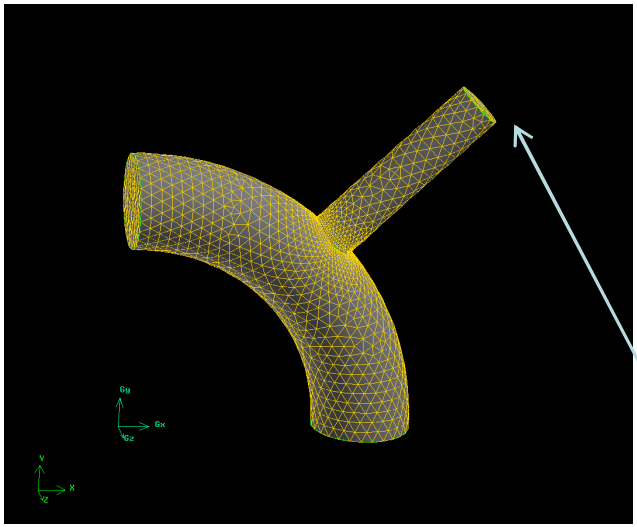
Solver Input File (SIF)



```
Boundary Condition 2
  Name = "largeoutflow"
  Target Boundaries  = 2
  Normal-Tangential Velocity = True
  Velocity 2 = 0.0
  Velocity 3 = 0.0
End
```

Only non-zero normal component

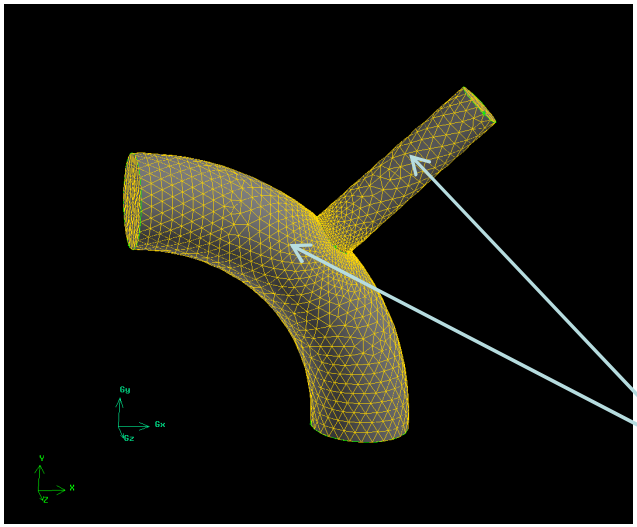
Solver Input File (SIF)



```
Boundary Condition 3
  Name = "smallinflow"
  Target Boundaries  = 3
  Normal-Tangential Velocity = True
  Velocity 1 = -0.01
  Velocity 2 = 0.0
  Velocity 3 = 0.0
End
```

Outward pointing normal

Solver Input File (SIF)



```
Boundary Condition 4
  Name = "pipewalls"
  Target Boundaries(2)  = 4 5
  Normal-Tangential Velocity = False
  Velocity 1 = 0.0
  Velocity 2 = 0.0
  Velocity 3 = 0.0
End
```

No-slip condition

The parallel run

- Save the SIF under `parallel_flow.sif`
- Write the SIF name into `ELMERSOLVER_STARTINFO`

```
module switch PrgEnv-pgi PrgEnv-gnu

module load elmer/latest

sallocate -n 4 --ntasks-per-node=4
   --mem-per-cpu=1000 -t 00:10:00 -p
   interactive

srun ElmerSolver_mpi
```

Special commands for `vuori.csc.fi`

```
mpirun -np 4 ElmerSolver_mpi
```

On an usual MPI platform

Combining the results

- Change into the mesh directory `cd flow`
- Run ElmerGrid to combine results

```
ElmerGrid 15 3
          parallel_flow
```

- Launch ElmerPost
- Load `parallel_flow.ep`