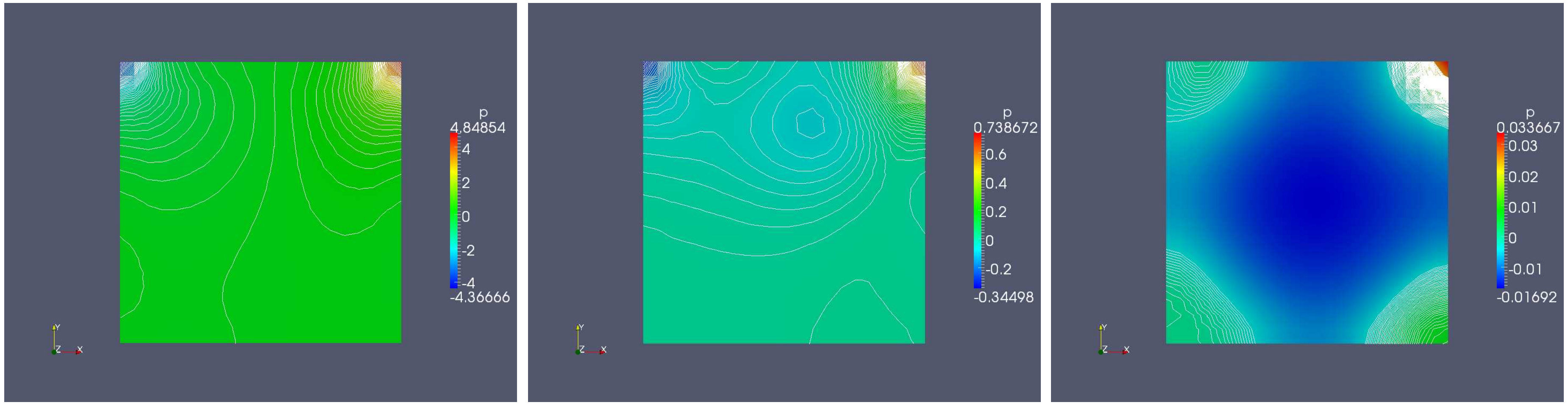
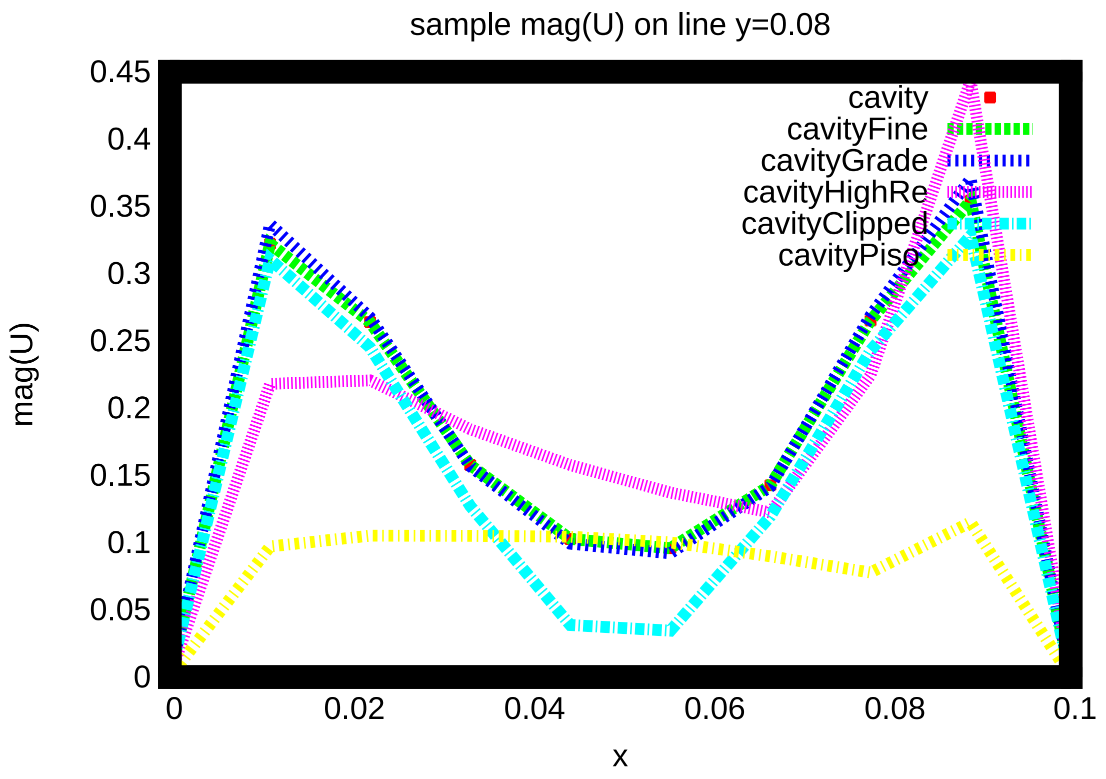
OpenFOAM Tutorials

Case: Cavity - Solver: icoFoam/pisoFoam



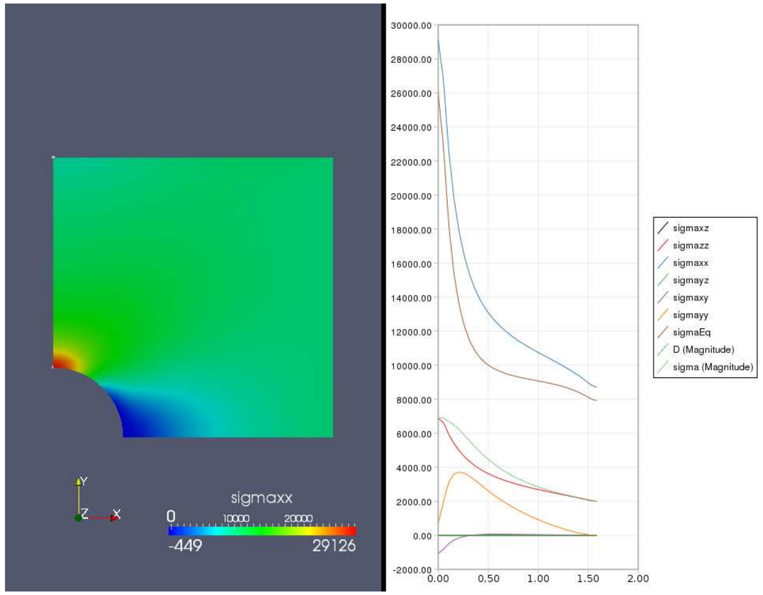
Pressure contour for the cases cavity and cavityHighRe solved with icoFoam and cavityHighRe solved with pisoFoam

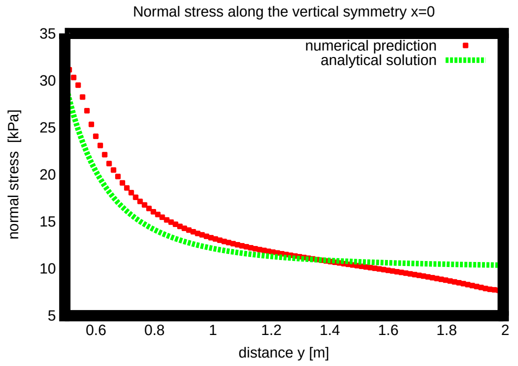
Case: Cavity*



Velocity magnitude along the line y=0.08 for all the cavity cases

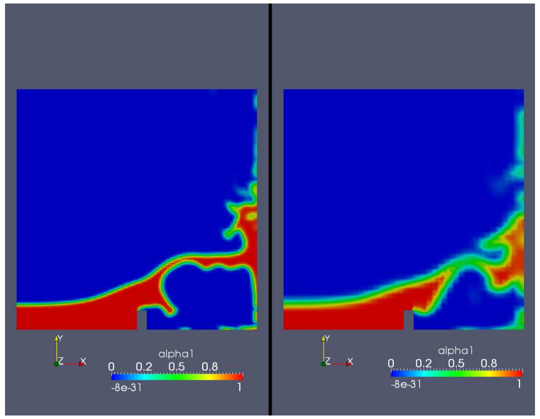
Case: PlateHole - Solver: solidDisplacementFoam



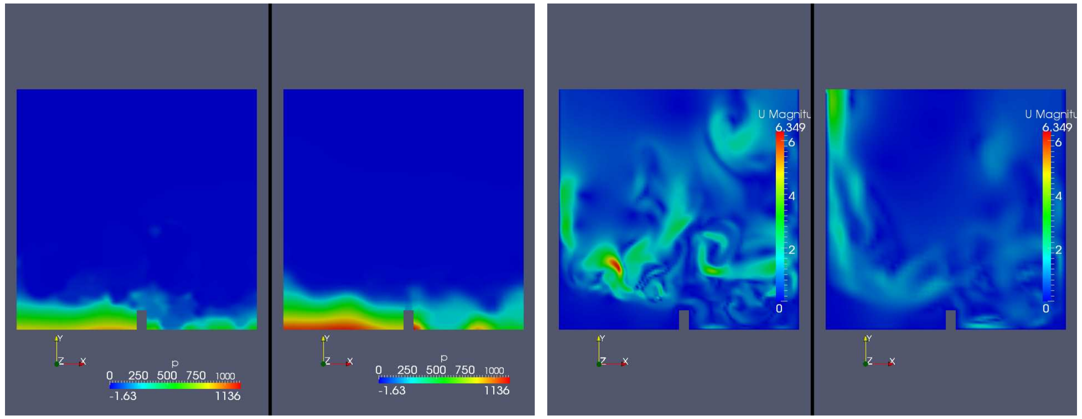


Stress along the line x=0. postprocessing with Paraview and Gnuplot

Case: damBreak - Solver :interFoam laminar



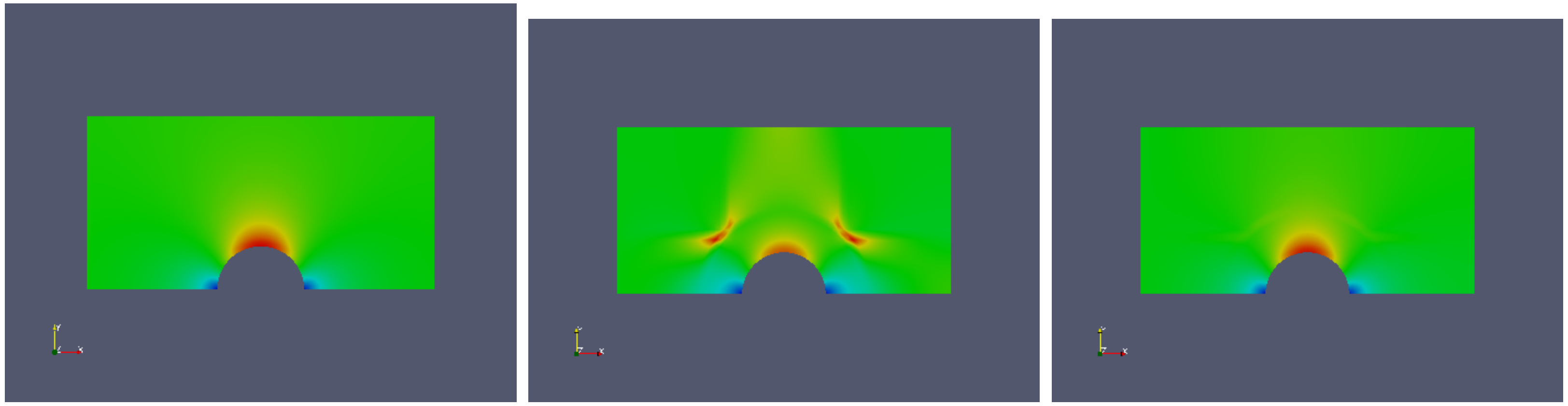
alpha at t=0.5 with a fine and a coarse mesh



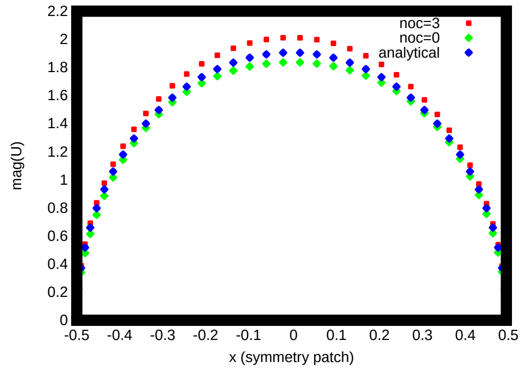
pressure and velocity at t=1 with a fine and a coarse mesh

Case: Cylinder - Solver: potentialFoam

Add UA.write() at the end of analyticalCylinder.C to write the vector UA in each time step diretory.

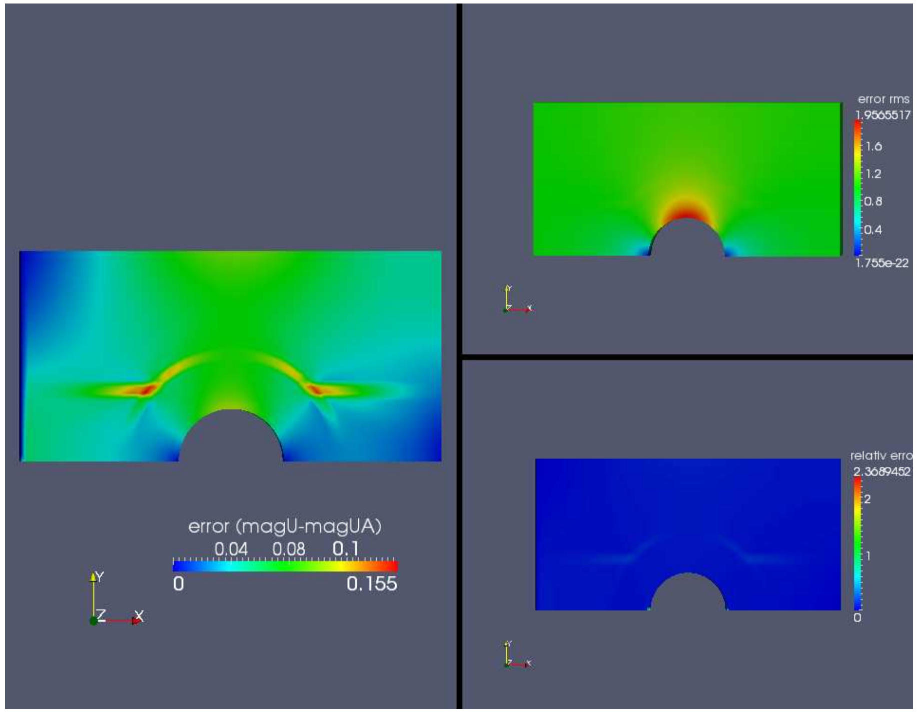


velocity profile of the analytical solution, the solution with 0 (resp 3) non orthogonal corrector



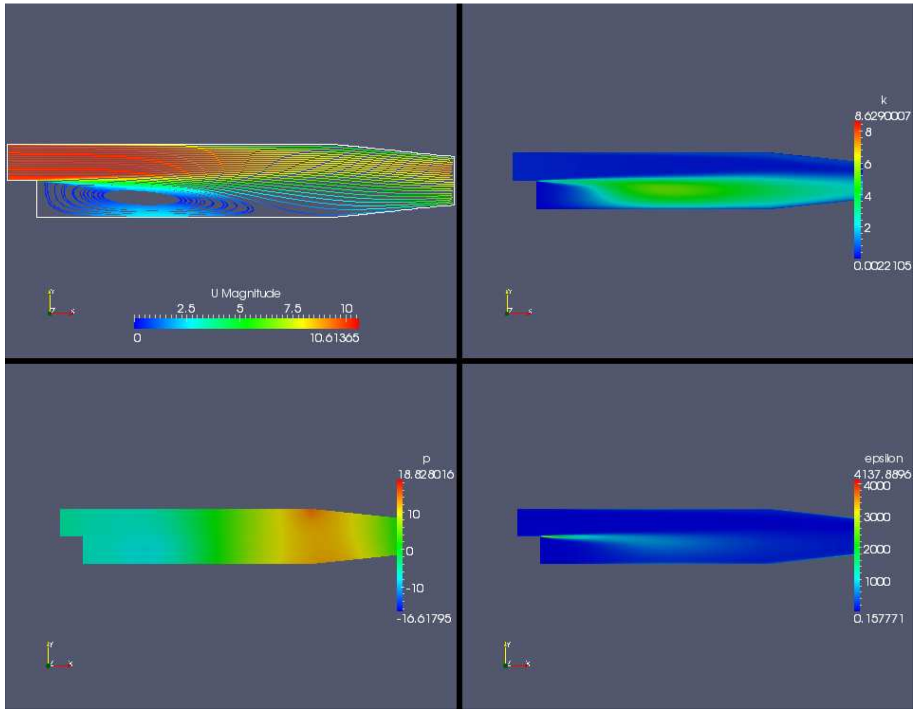
velocity profile on the symmetry patch. Postprocessing with gnuplot

Case: Cylinder - Solver: potentialFoam

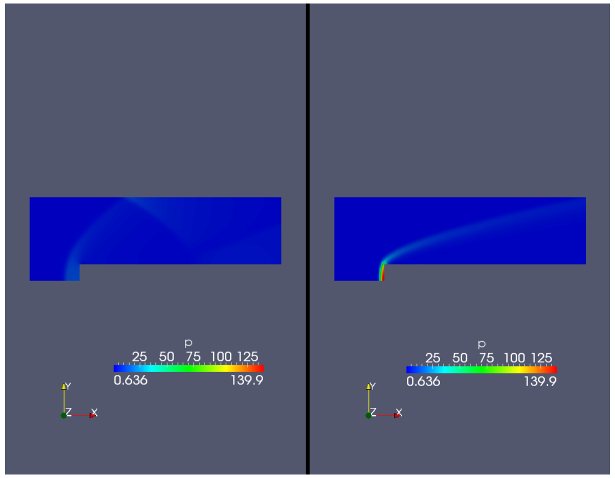


numerical error on the velocity field

Case: pitzDaily - Solver: simpleFoam

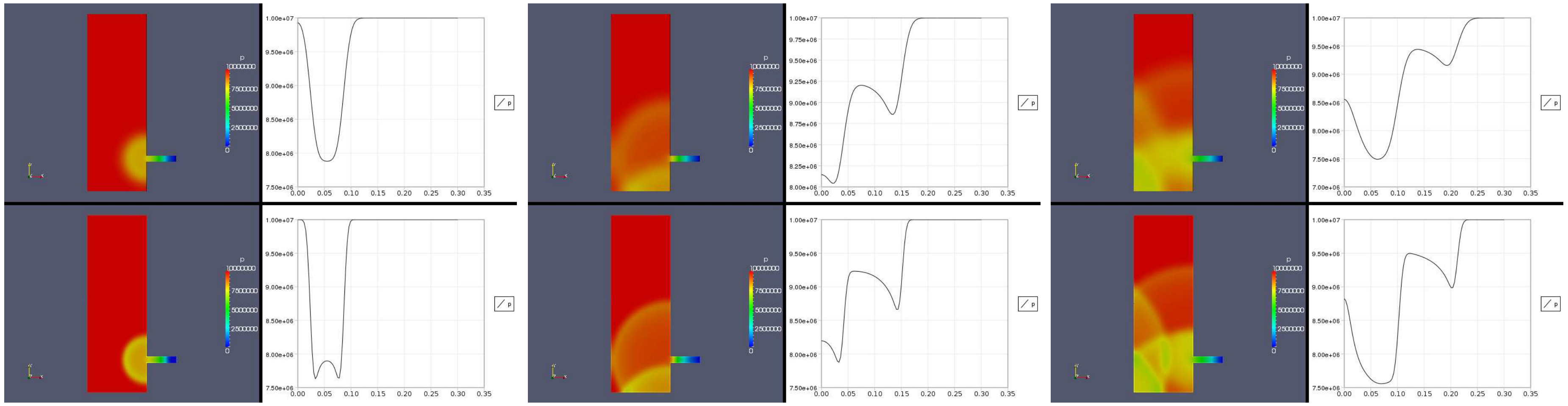


Case: forwardStep - Solver: sonicFoam



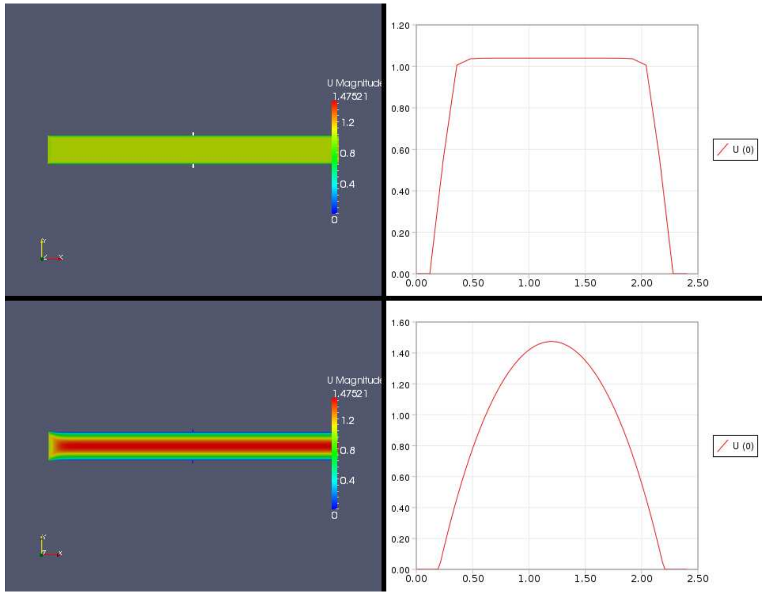
pressure field at t=10s with inlet velocity=3 and 10 m/s

Case: decompressionTank - Solver: sonicLiquidFoam



fine and coarse grid. t=6e-5, 1e-4 and 1.5e-4. Pressure along a line x=constant

Case: hartmann - Solver: mhdFoam



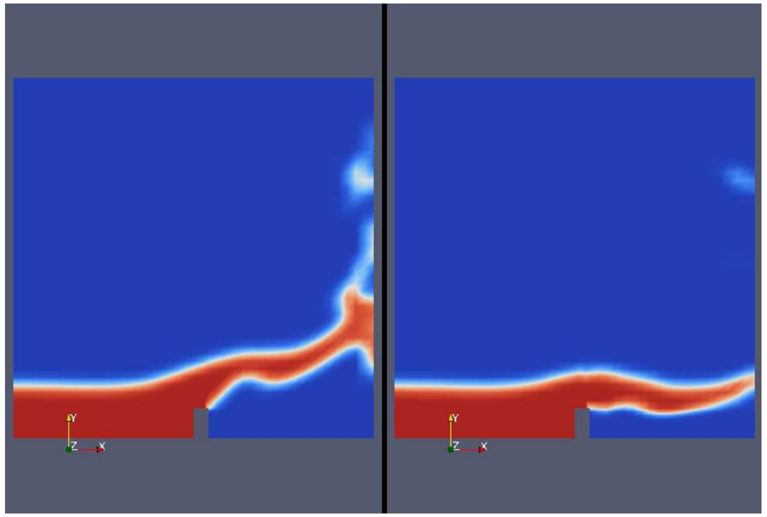
B=1 and 20. Ux along a line y=constant

Case: damBreak modified = damBreakOpen - Solver: interFoam

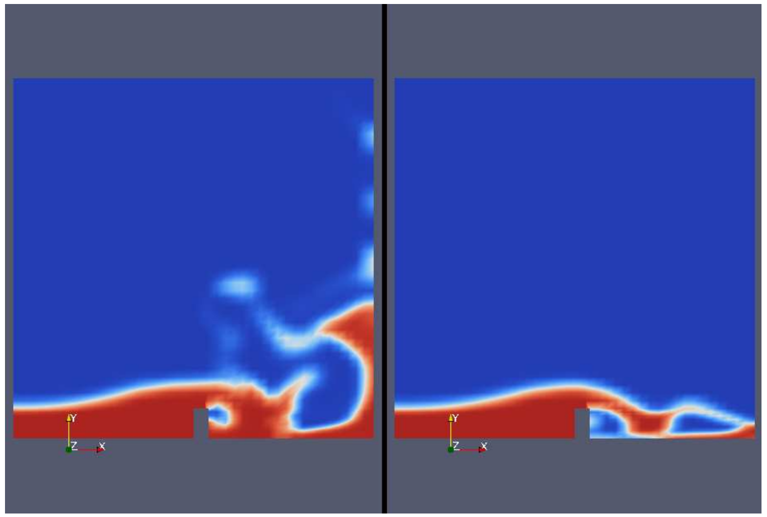
The boundary ”rightWall” with type wall is changed to type patch in polyMesh/boundary. The velocity boundary condition in 0/ is changed to zeroGradient, and for the pressure, totalPressure is used, similarly to the upper patch called ”atmosphere”. Now the liquid phase can escape the domain. The total volume fraction of the liquid phase in the domain is computed at each time step. We simply write an utility to print out alphaTot/volume

```
forAll(mesh.cells(),celli)
        {
          alphaTot=alphaTot+alpha1[celli]*mesh.V()[celli];
          volume=volume+mesh.V()[celli];
        }
```

The results are in the table. We clearly see that the liquid phase escapes the domain for the case damBreakOpen as the volume fraction of the liquid phase decreases. It should be constant for the intial case damBreak. It is actually constant untill $t = 0.4$, when the liquid phase hurts the boundary ”rightWall” which produces small bubbles and numerical errors. VOF method introduces som values of alpha < 0 and > 1 that are later bounded in $[0, 1]$.



Liquid phase at t= 0.4. Case damBreak and damBreakOpen



Liquid phase at t= 0.65 . Case damBreak and damBreakOpen

Time	damBreak	damBreakOpen
0	0.130194	0.130194
0.1	0.130194	0.130194
0.2	0.130194	0.130194
0.3	0.130194	0.130194
0.4	0.130185	0.113
0.5	0.129592	0.0986771
0.6	0.129562	0.085936
0.7	0.129557	0.0764507
0.8	0.129556	0.0717916
0.9	0.129556	0.0650988
1	0.129497	0.0600475

Liquid phase total volume fraction