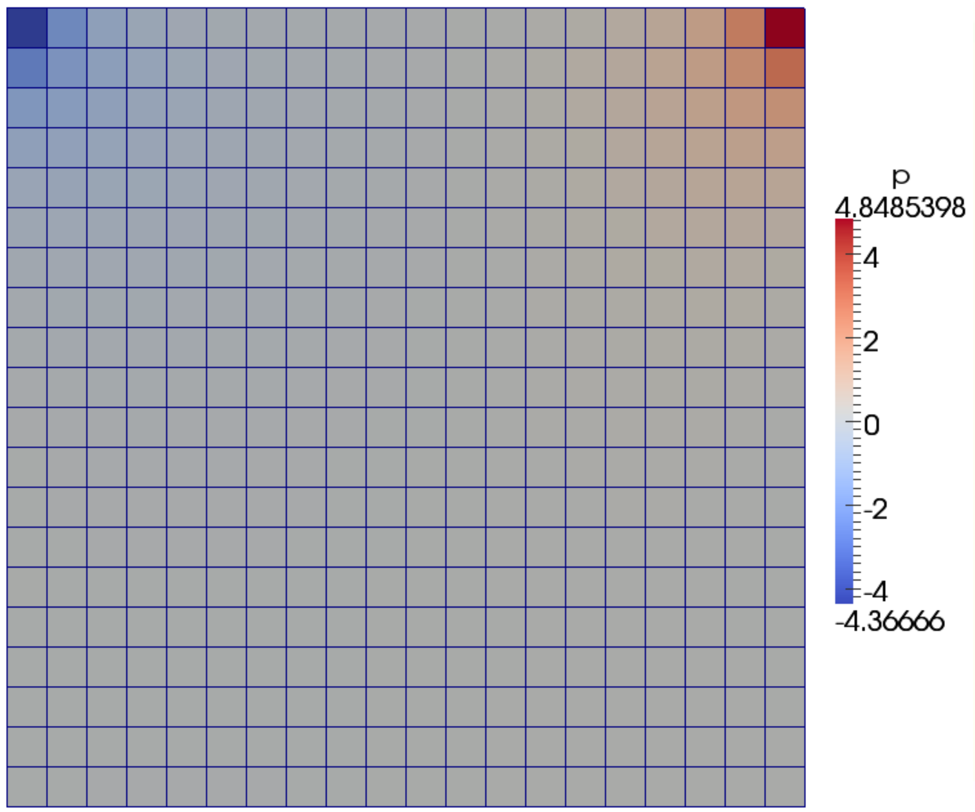
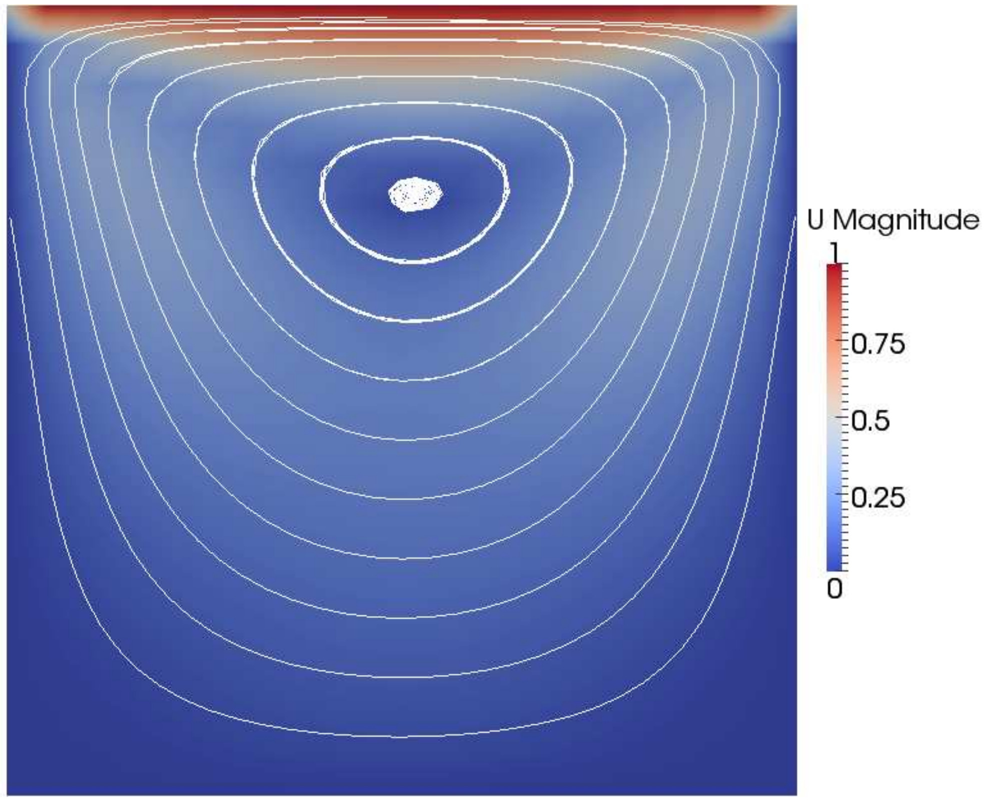
Assignment 1
CFD with open source software 2010

Erik Svenning

icoFoam: cavity

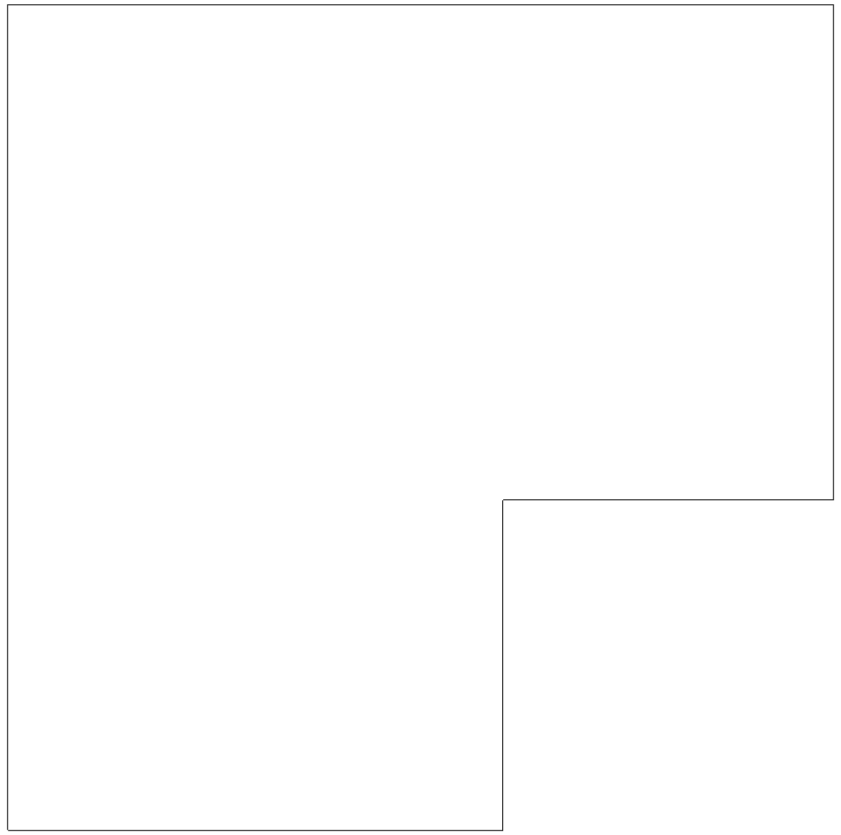


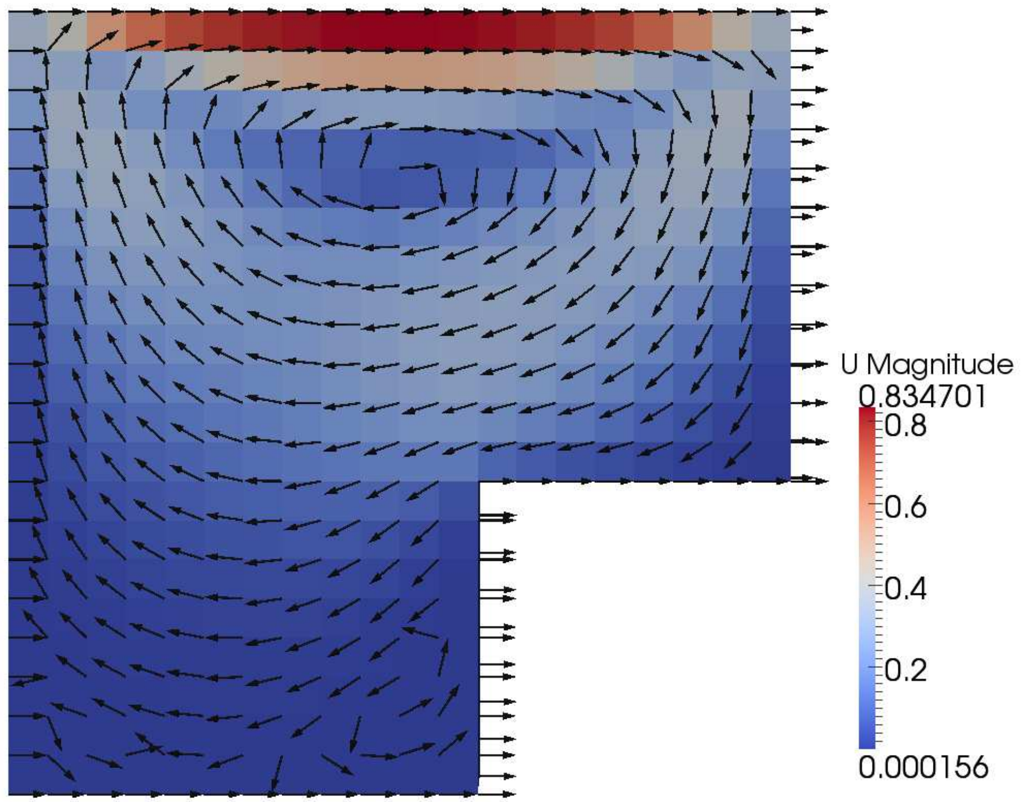
Pressure distribution and grid.



Velocity magnitude and streamlines. The streamlines were generated with a line source passing through the center of the vortex.

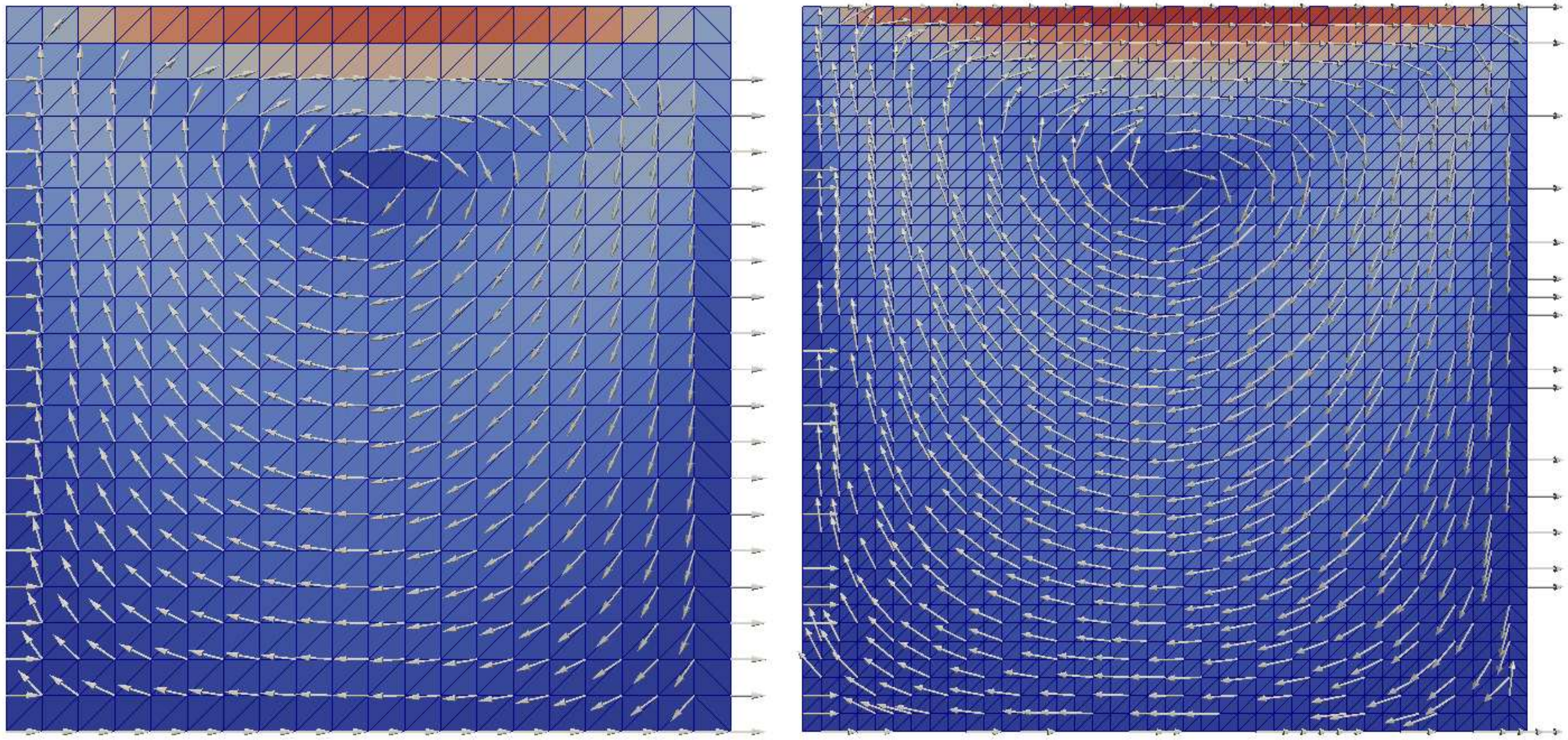
icoFoam: cavityClipped





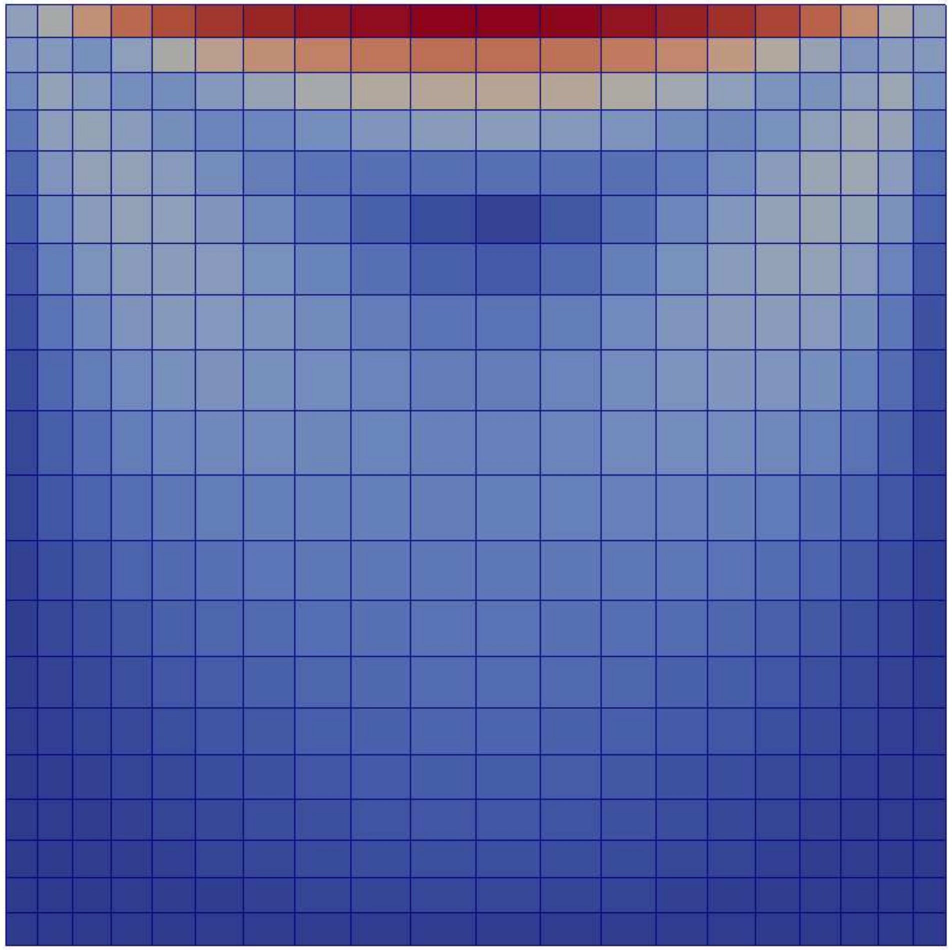
Geometry of a clipped cavity (left) and velocity magnitude and direction of velocity vectors (right). The arrows were generated with *Glyph* with scaling turned off.

icoFoam: cavityFine

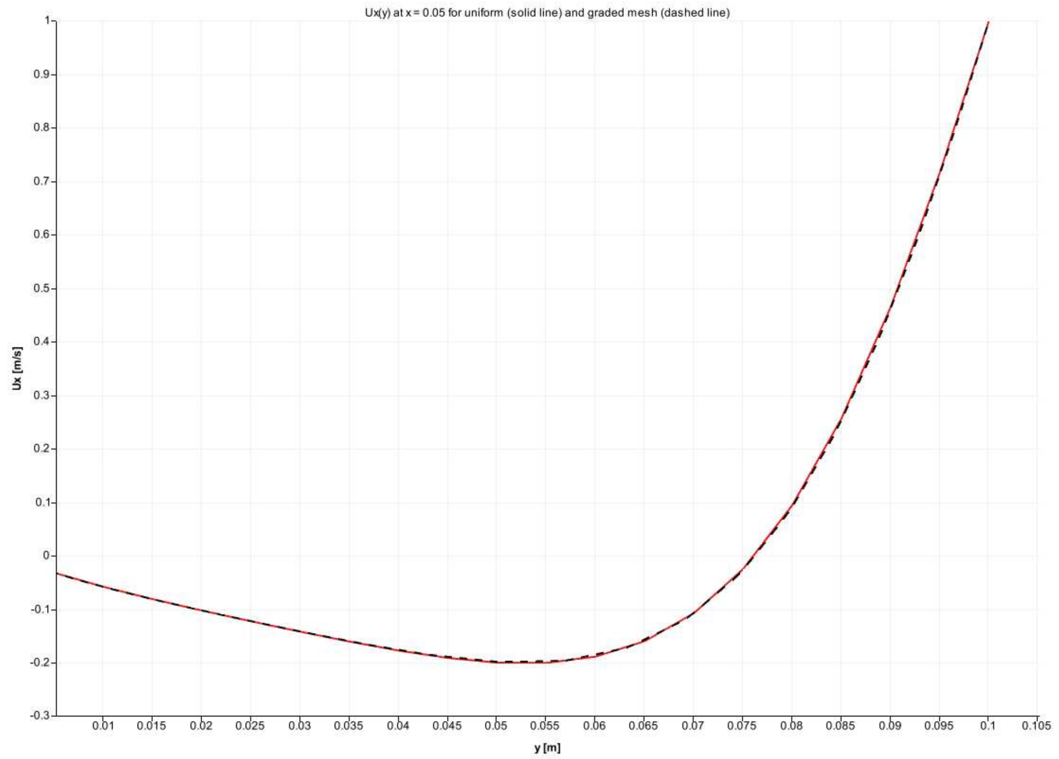


Velocity vectors predicted with a coarse grid (left) and a finer grid (right).

icoFoam: cavityGrade

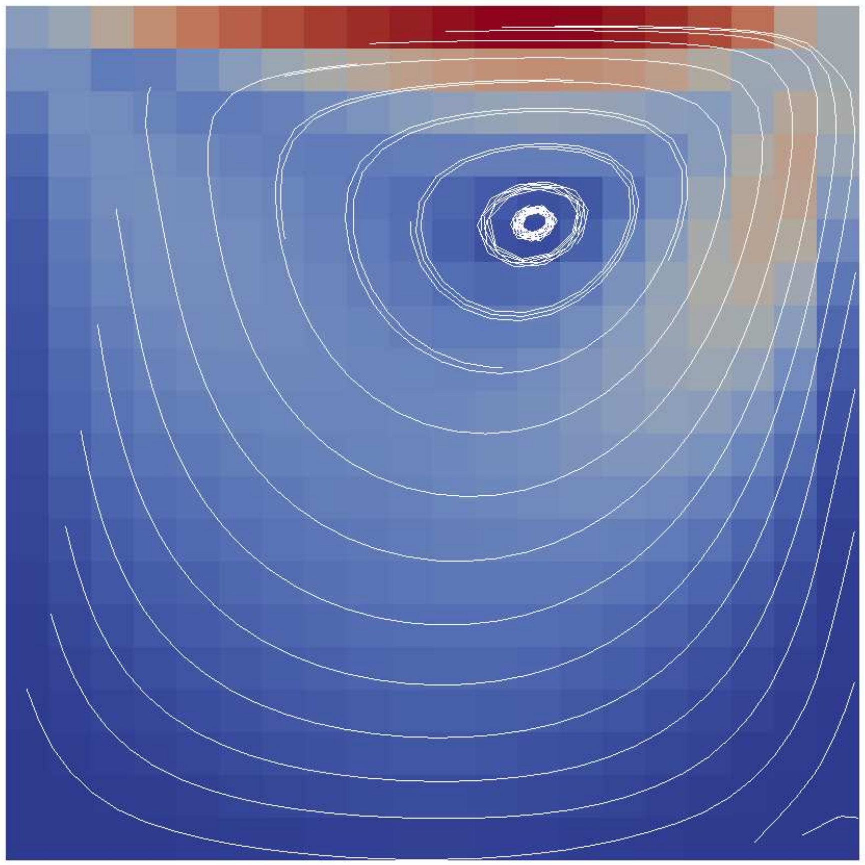


Graded mesh.

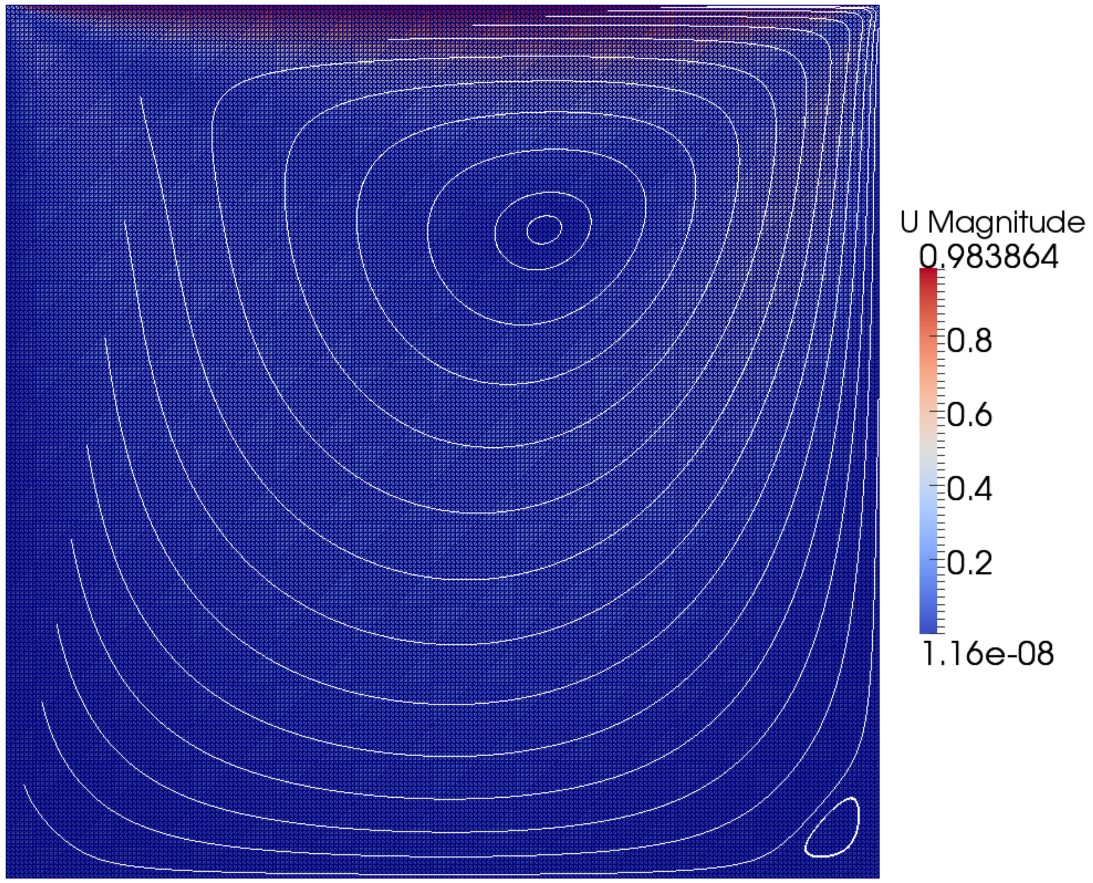


Horizontal velocity as a function of the vertical position. The horizontal position is in the center of the domain and the graph was generated with the *Plot over line* filter.

icoFoam: cavityHighRe

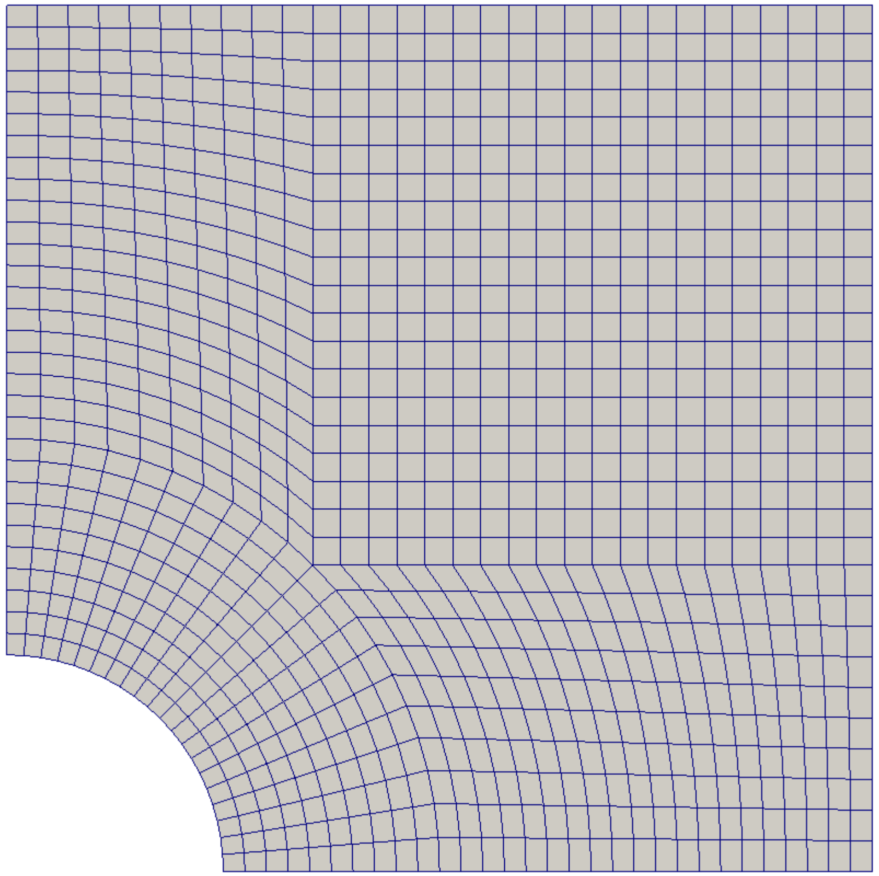
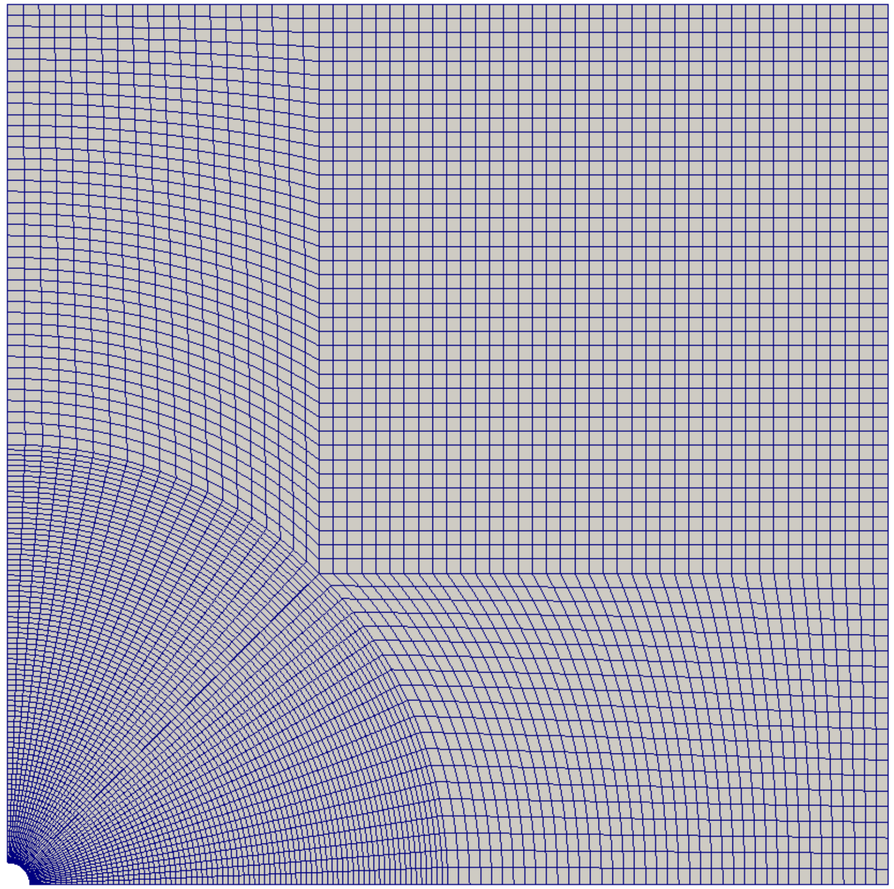


Velocity magnitude and streamlines for Re = 50. The grid consists of 400 cells.



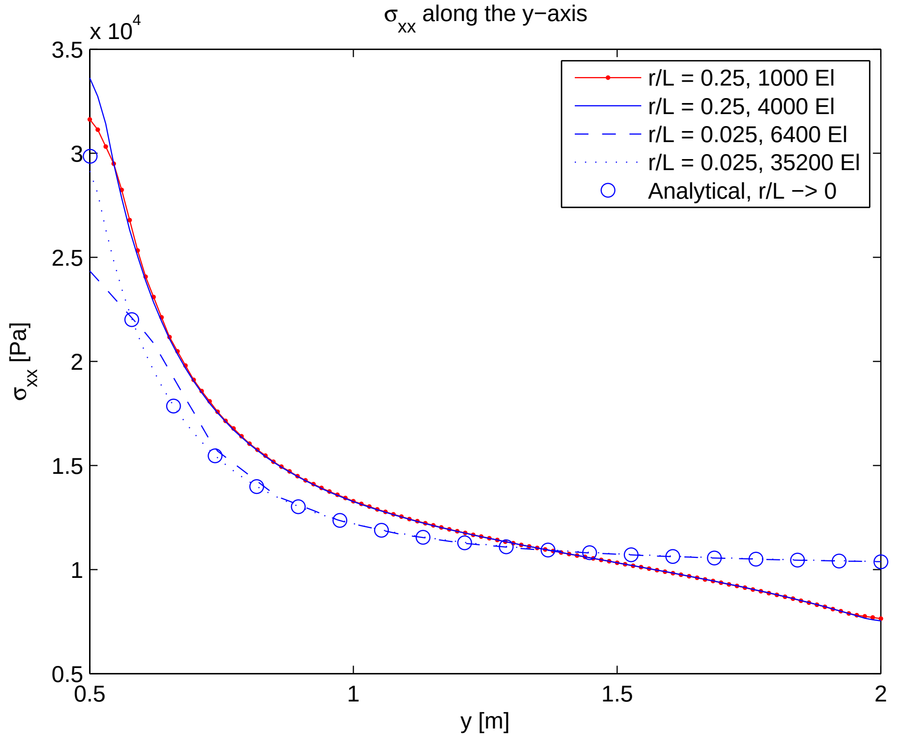
Velocity magnitude and streamlines for Re = 50. The grid consists of 40000 cells.

solidDisplacementFoam: plateHole

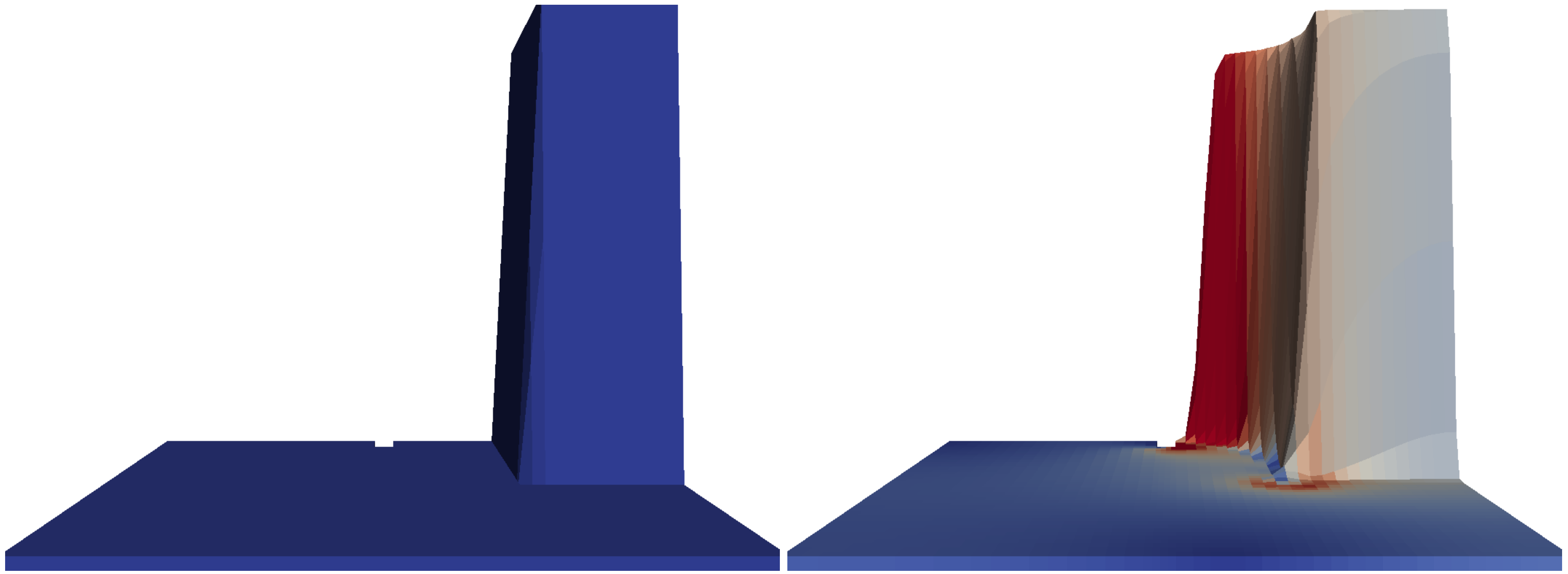
Two cases are considered: a plate with large hole compared to the plate length (left) and a small hole (right). The hole size was changed by manually editing some lines in *blockMeshDict*. For each case, two different grid sizes are considered.

solidDisplacementFoam: plateHole



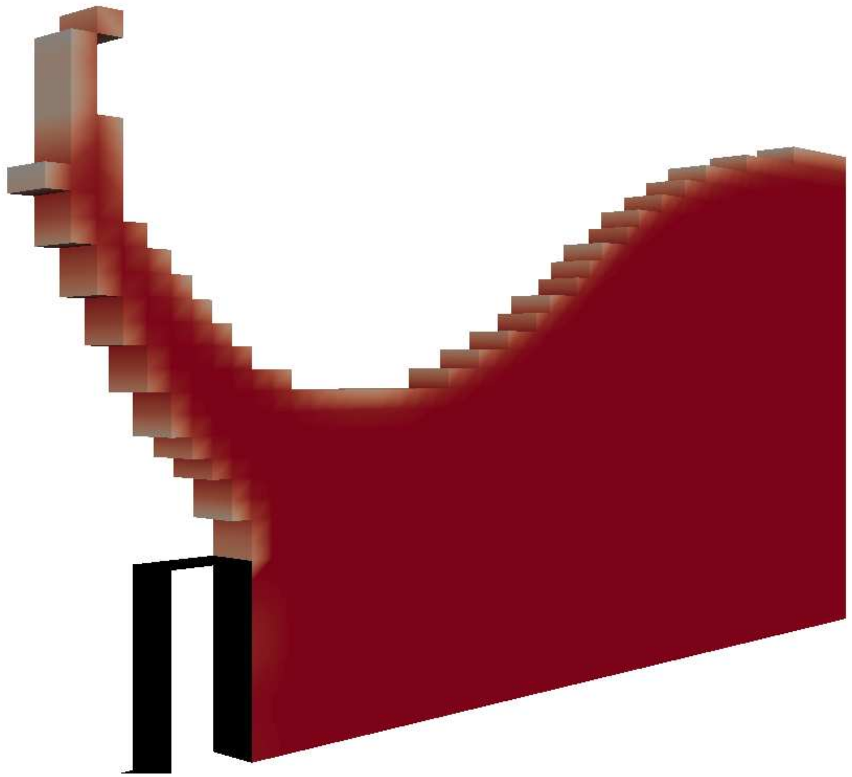
The figure shows the normal stress in the x-direction for a plate with a large hole compared to the plate length and plate with a smaller hole. Two different grid sizes for each case have been used. The agreement with the analytical solution is good for the case with a small hole and a fine grid. The raw data for the plot were extracted with *foamCalc components sigma* and the plot was made with Matlab.

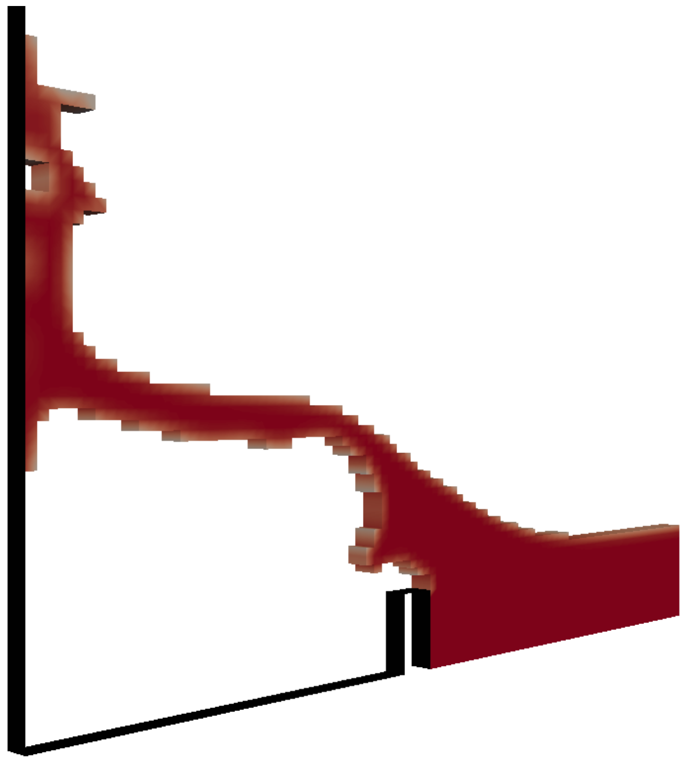
interFoam/laminar: damBreak



A column of water at rest (left) and a column of water approaching a dam wall (right). The plots were generated with *filters - Warp by scalar*. The volume fraction alpha was used for the warping and the z-direction was used as normal direction.

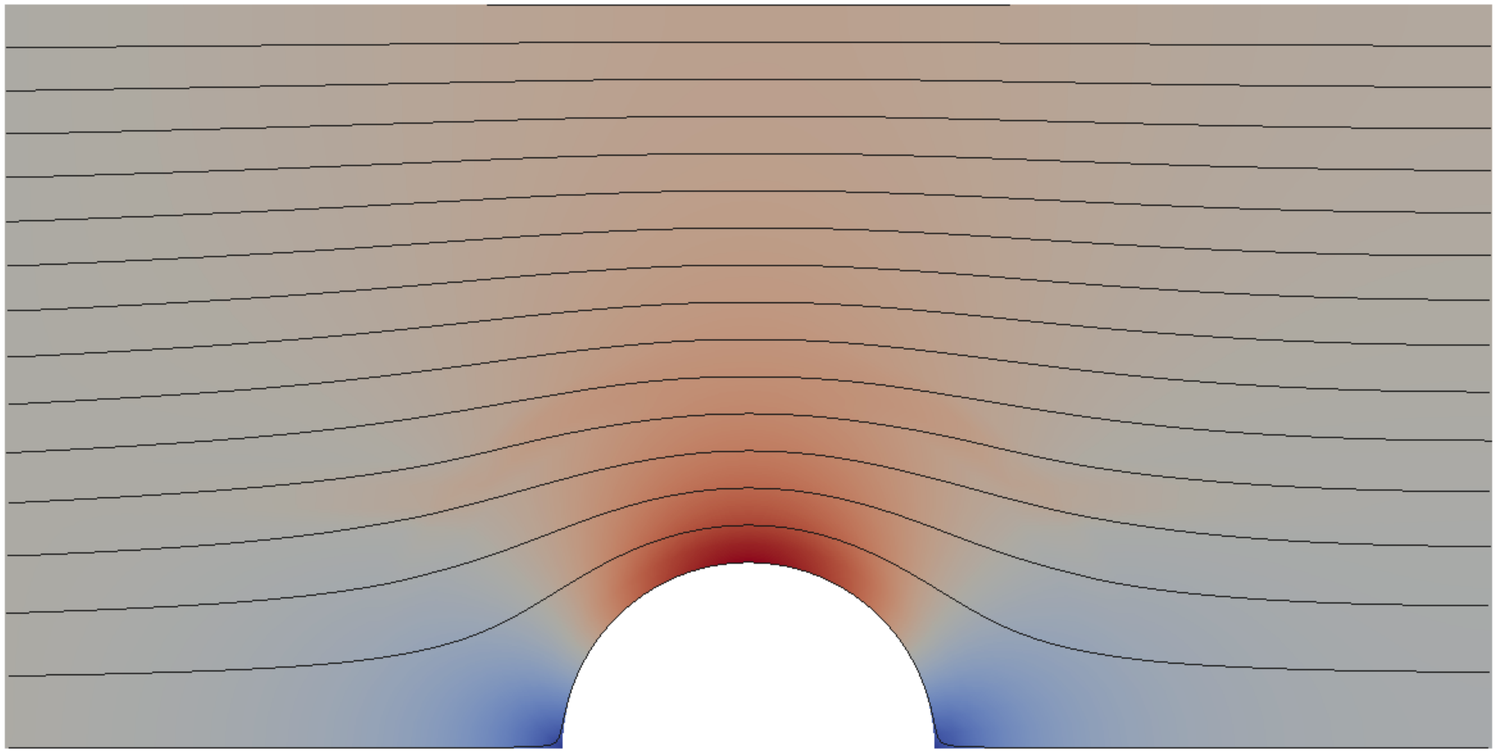
interFoam/laminar: damBreakFine





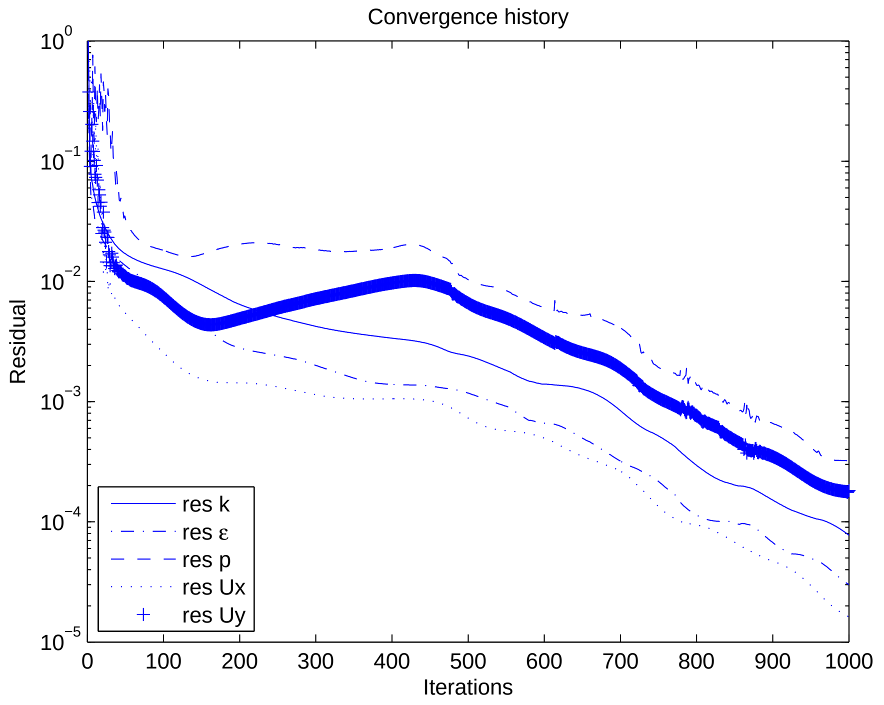
A column of water impacting on a small obstacle in the domain (left) and a column of water impacting on the domain wall (right). The plots were generated with *Threshold* with a threshold value of 0.5 on the volume fraction. The walls were visualized by showing the patches rightWall and *lowerWall*.

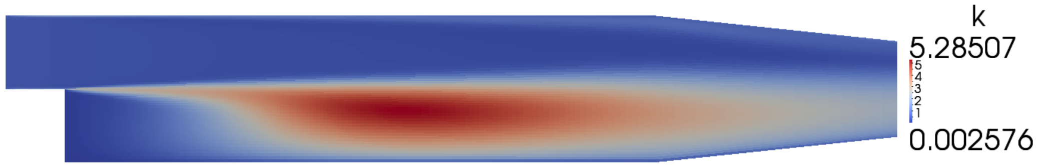
potentialFoam: cylinder



Velocity magnitude and streamlines for flow around a half cylinder. Translating the streamlines in the z-direction prevents the surface plot from hiding the streamlines.

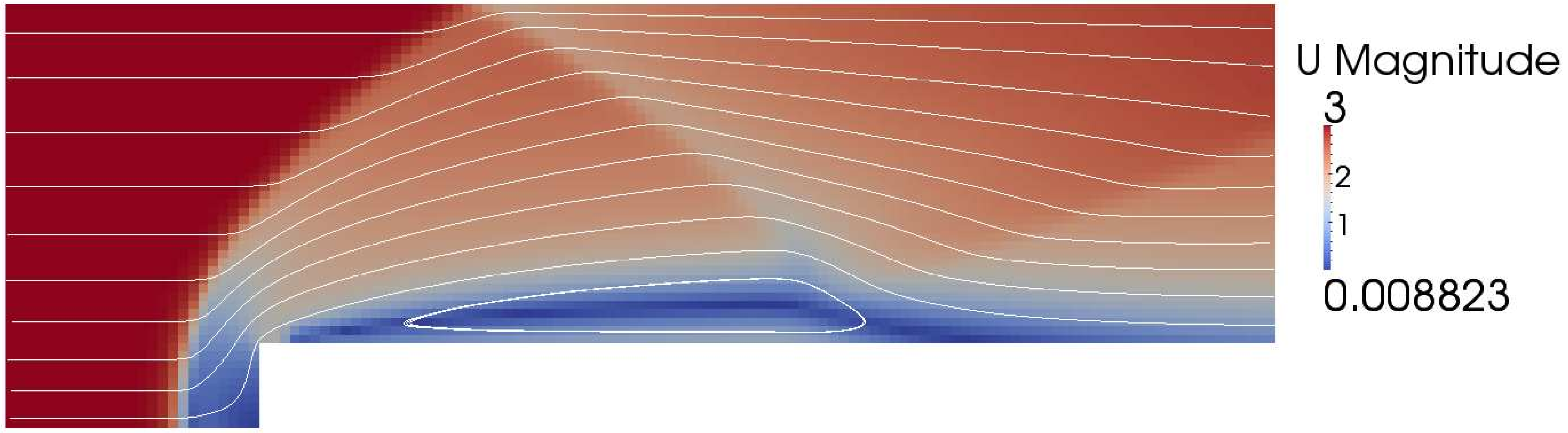
simpleFoam: pitzDaily





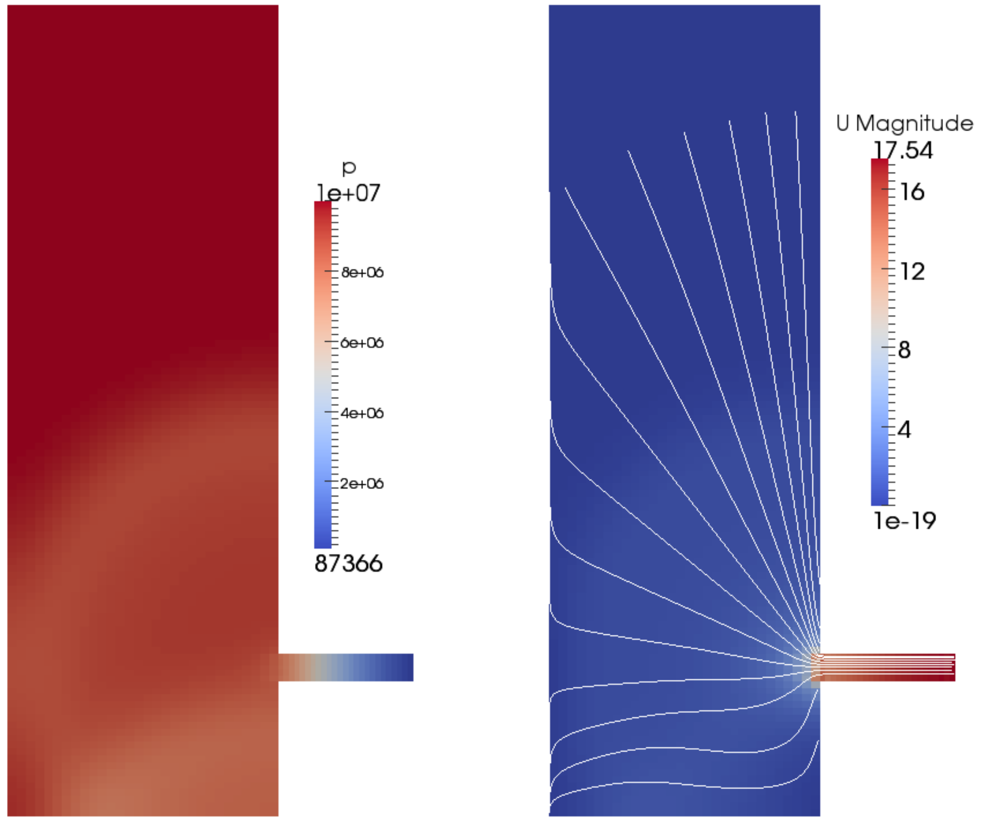
Convergence history (left) and turbulent kinetic energy (right). The convergence history was extracted from the logfile with *foamLog* and it was plotted with Matlab.

sonicFoam: forwardStep



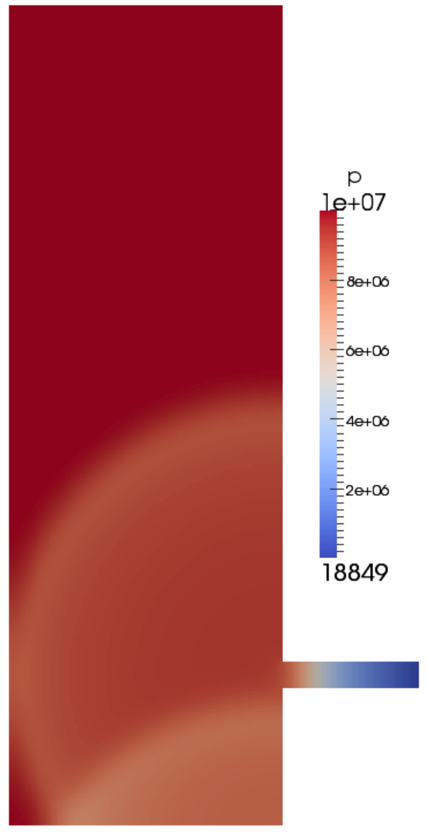
Velocity magnitude and streamlines. The streamlines were generated with line source. Decreasing the *Maximum Step Length* makes it easier to get streamlines that go through the whole domain.

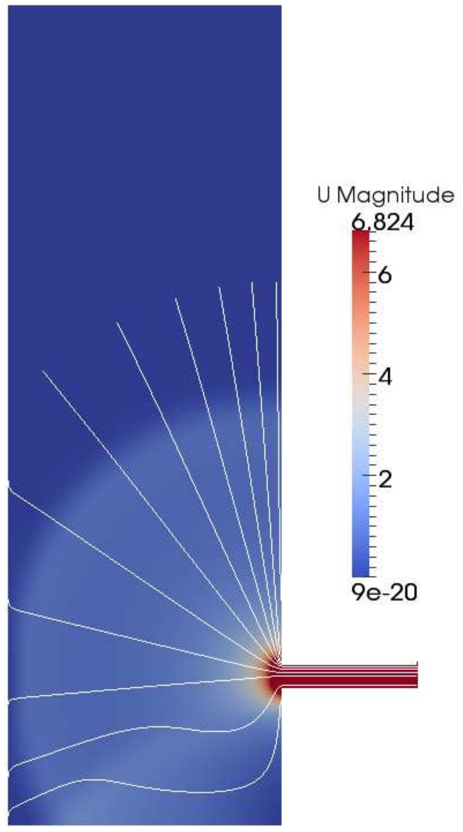
sonicLiquidFoam: decompressionTank



Pressure distribution and velocity field.

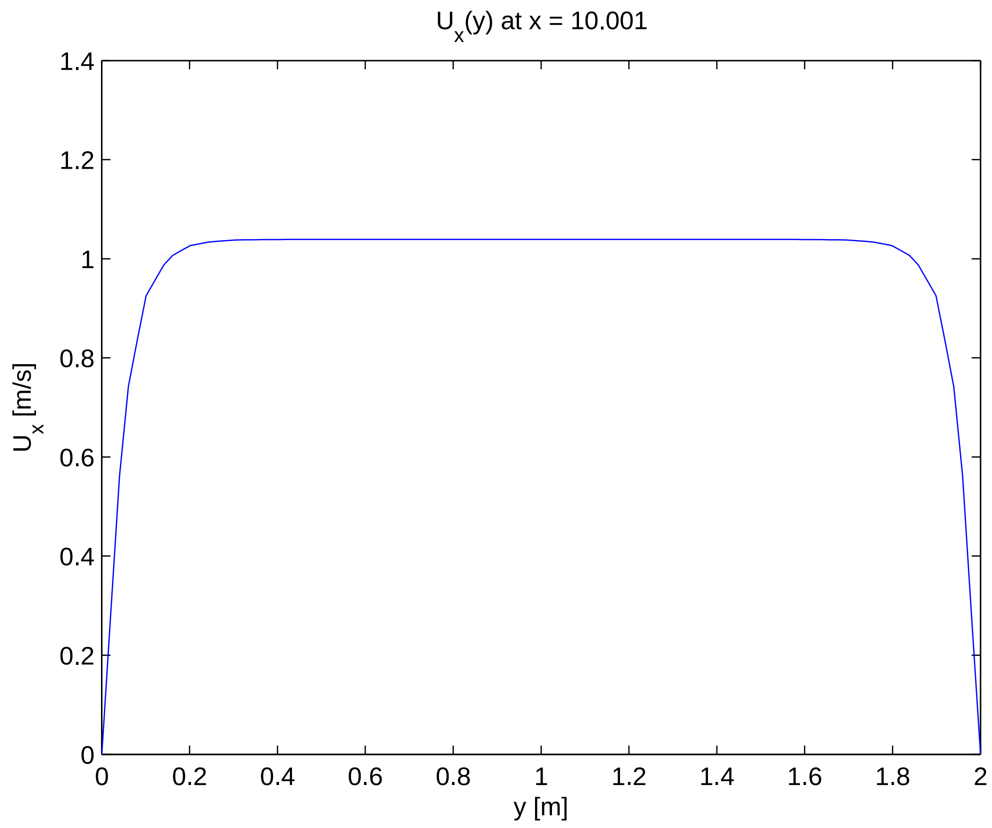
sonicLiquidFoam: decompressionTankFine





Pressure distribution and velocity field with finer mesh.

mhdFoam: hartmann



Horizontal velocity as a function of y at $x = 10.001$. The data was extracted with the *sample* utility.

Modification of interFoam/laminar: damBreakFine

In the original damBreakFine, the water is not allowed to pass the left and right domain walls. Now the boundary condition at the right wall is changed, so that water is allowed to leave the domain. Since outflow of water is possible now, we would like to know the amount of water that is still left in the domain at the end of the simulation.

Modification of interFoam/laminar: damBreakFine continued

To allow outflow of water, the boundary condition for U is modified as follows:

```
rightWall
{
    type            pressureInletOutletVelocity;
    value           uniform (0 0 0);
}
```

The boundary condition for the pressure is changed to:

```
rightWall
{
    type            totalPressure;
    p0              uniform 0;
    U               U;
    phi             phi;
    rho             rho;
    psi             none;
    gamma           1;
    value           uniform 0;
}
```

Modification of interFoam/laminar: damBreakFine continued

A measure of the water left in the domain is the volume fraction in a cell multiplied by the cell volume and the density. If we divide it by the amount present at time zero, we get a value between zero and one, where one means that all of the water is still in the domain and zero means that all the water has left the domain. If we assume that all the cells have the same volume (which is not completely true, but I am happy with a rough estimate), the relative amount of water left in the domain can be written as:

$$f = \frac{m}{m_0} = \frac{\sum \alpha_i \rho_i V_i}{\sum \alpha_{i,t=0} \rho_{i,t=0} V_{i,t=0}} = \frac{\sum \alpha_i}{\sum \alpha_{i,t=0}} \qquad (1)$$

where α is the volume fraction, ρ is the density of the water and V is the cell volume.

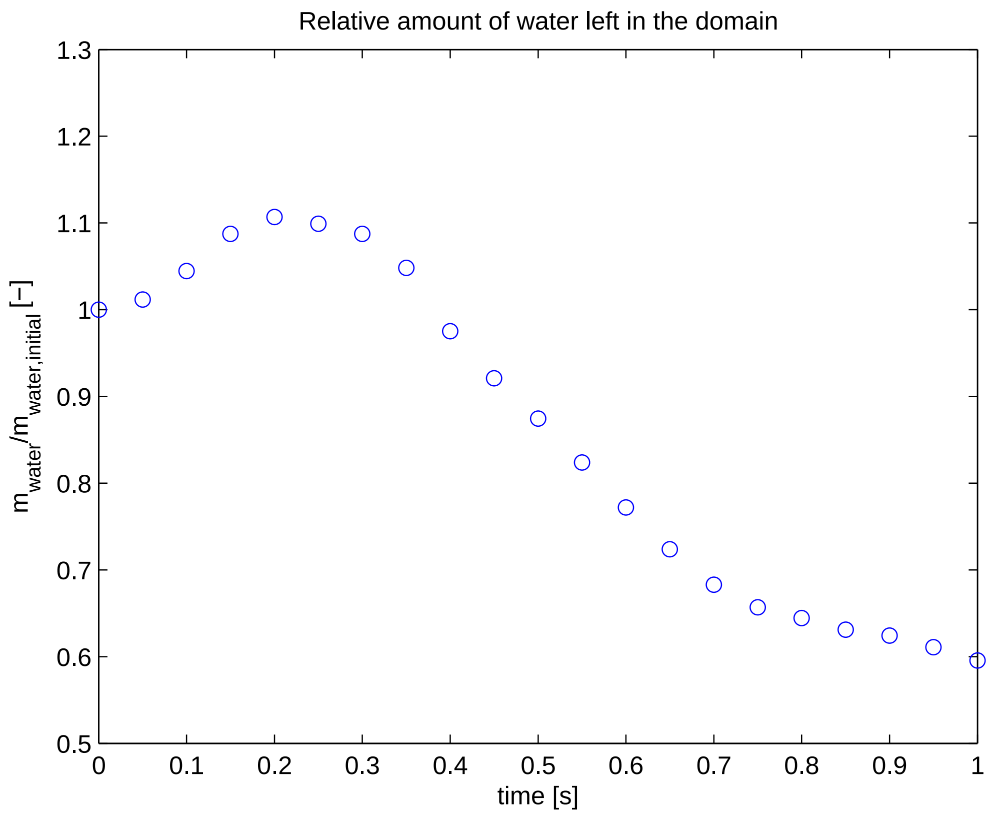
Modification of interFoam/laminar: damBreakFine continued

To be able to calculate the sum in (1), we need a convenient output of α that can be read by Matlab or a similar program. Such an output can be generated by letting the *sample* utility write the values of α on a plane that goes through the whole domain. To achieve this, copy the *sampleDict* file from *solidDisplacementFoam: plateHole* and change it to:

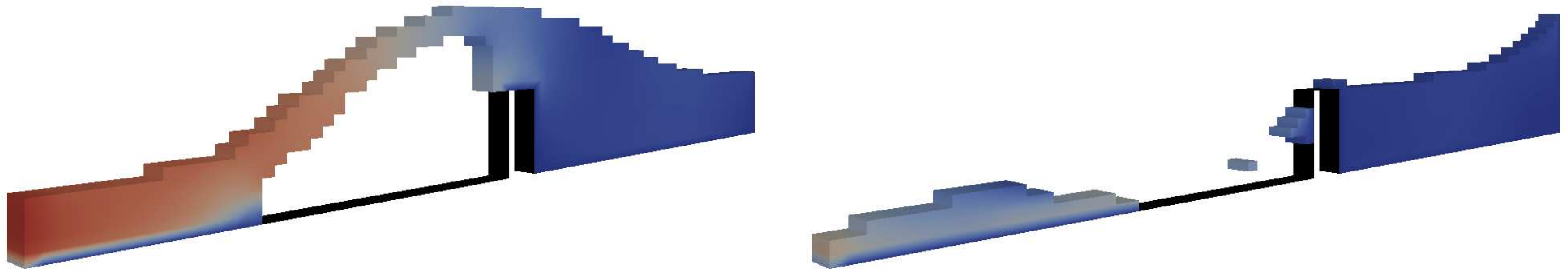
```
surfaceFormat raw;
   surfaces (
      plane1
      {
         type plane;
         basePoint ( 0 0 0.005 );
         normalVector ( 0 0 1 );
      }
   );

   fields ( alpha1 );
```

Modification of interFoam/laminar: damBreakFine continued

Now the data can be read into Matlab and summed over the whole domain to get the amount of water left. The result is shown in the figure below. It can be noted that the amount of water left is greater than unity at $t \approx 0.2$. This is probably an effect of the assumption of equally sized cells. We conclude however that roughly 60 % of the water is still in the domain at $t = 1$ and this is the answer to our initial question.



Modification of interFoam/laminar: damBreakFine continued



Water column at $t = 0.6\ s$ (left) and at $t = 1s$ (right). Compare these figures with the right plot on slide 10, where the water is not allowed to leave the domain.