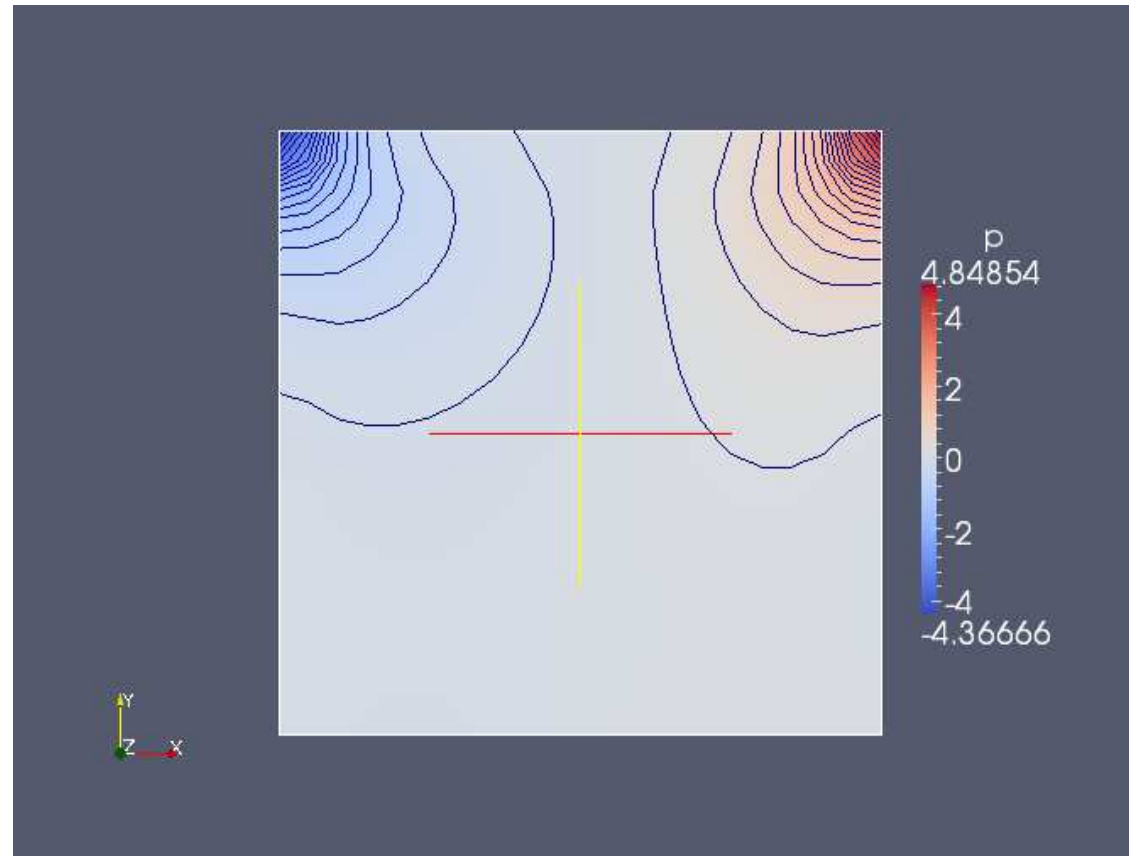
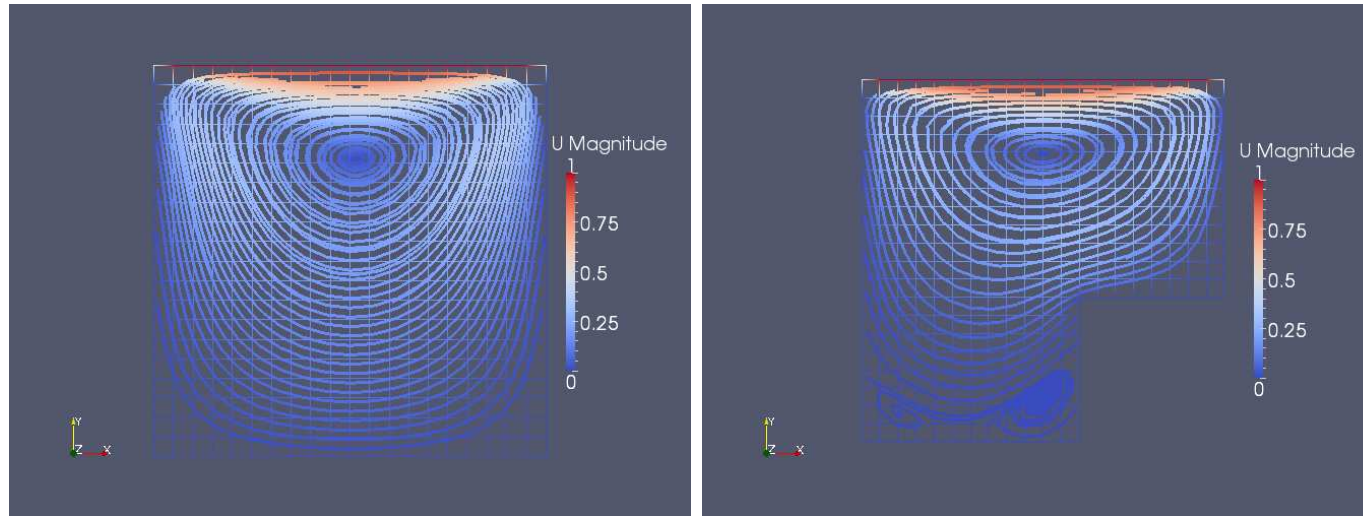


Cavity



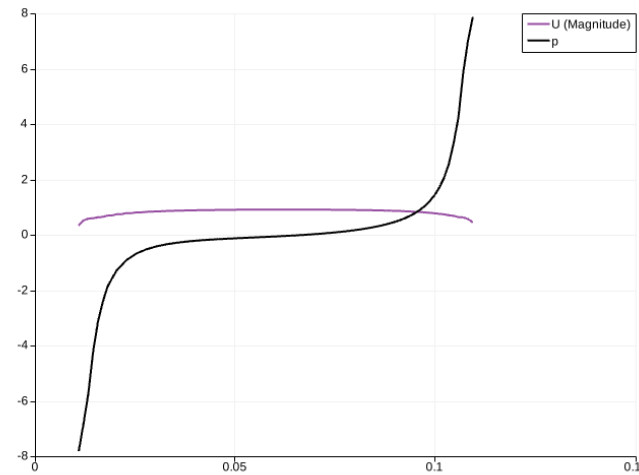
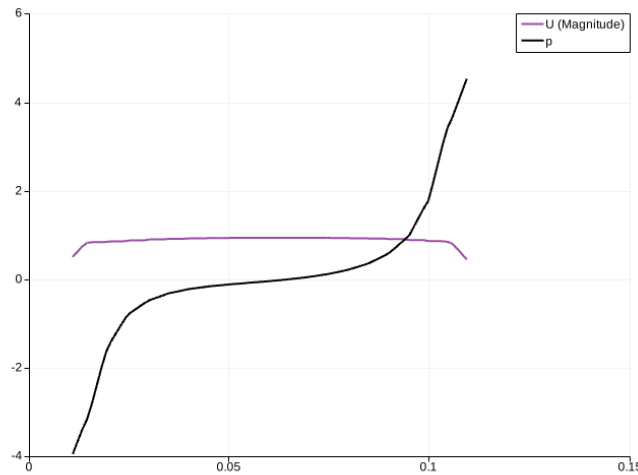
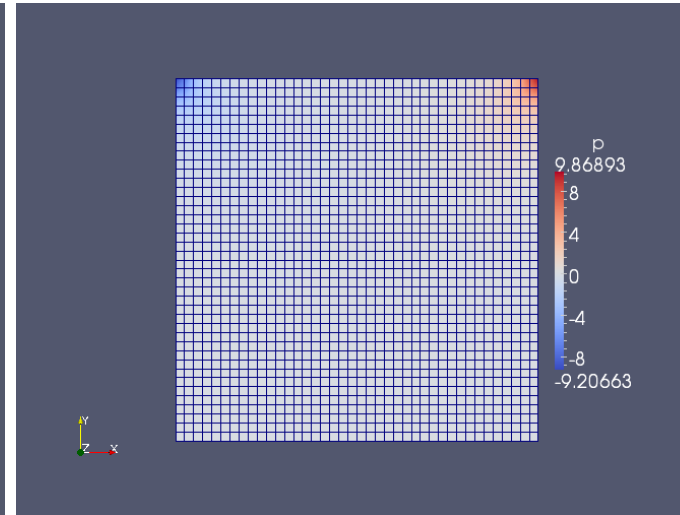
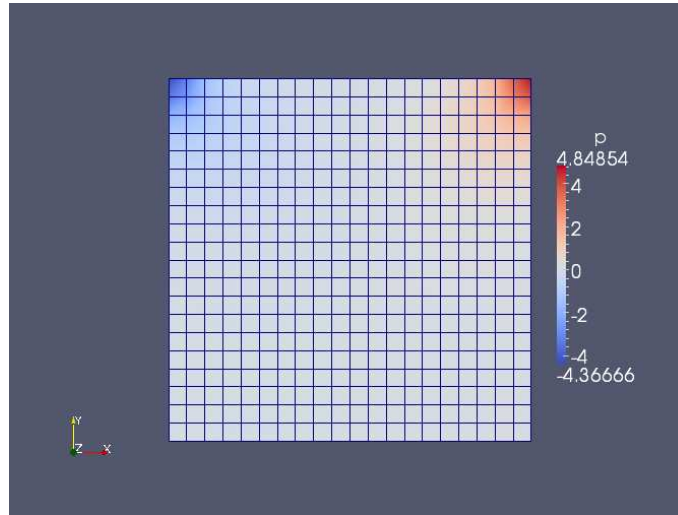
- Slicefilter in the z-normal direction, contour plots of pressure where number of isosurfaces is set to 40. Background is the pressure distribution displayed in "surface" option.

CavityClipped



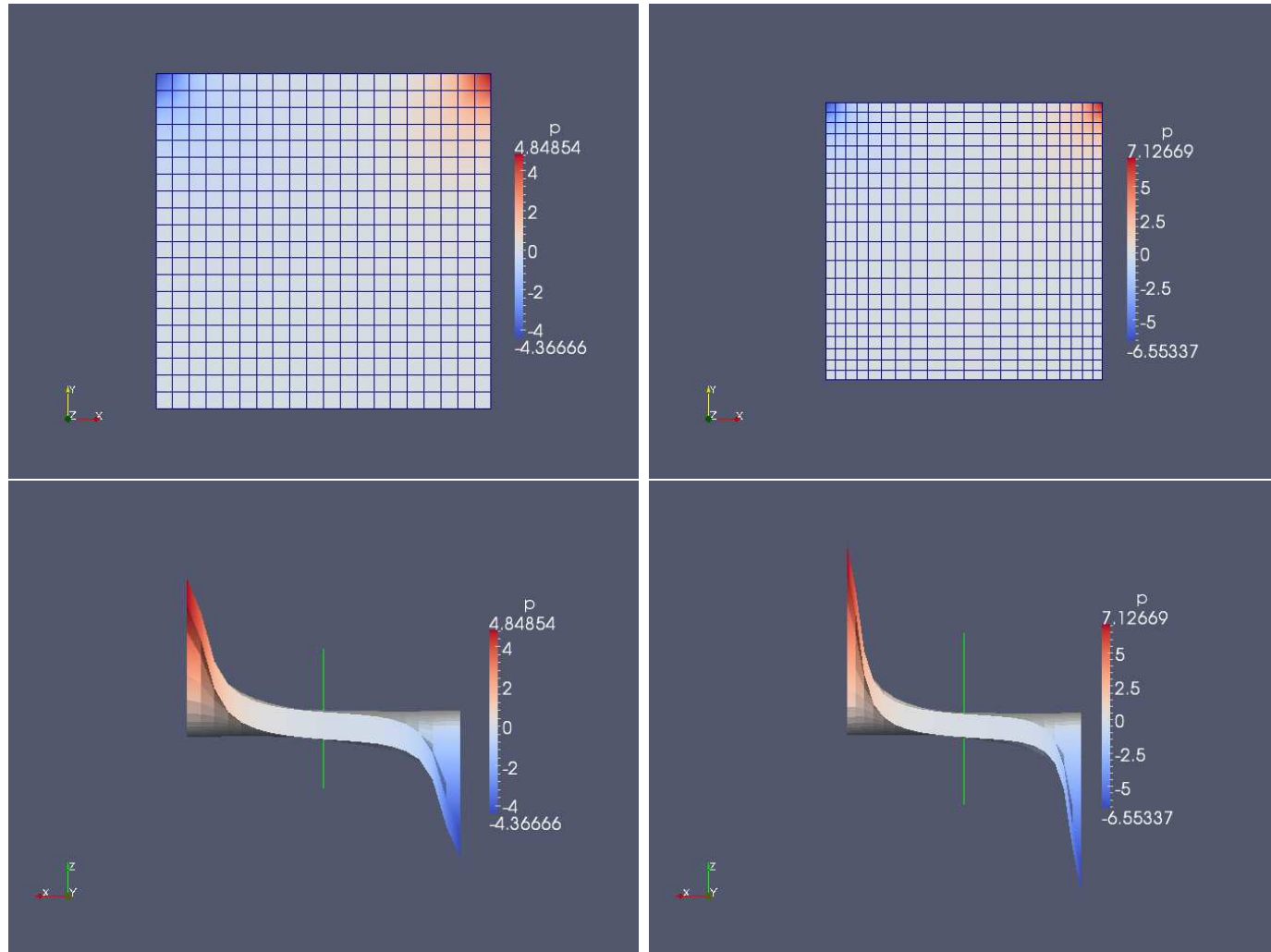
- Left: Cavity domain for comparison
- Right: CavityClipped domain
- Wireframe option is used and streamtracer filter with resolution 40.

CavityFine



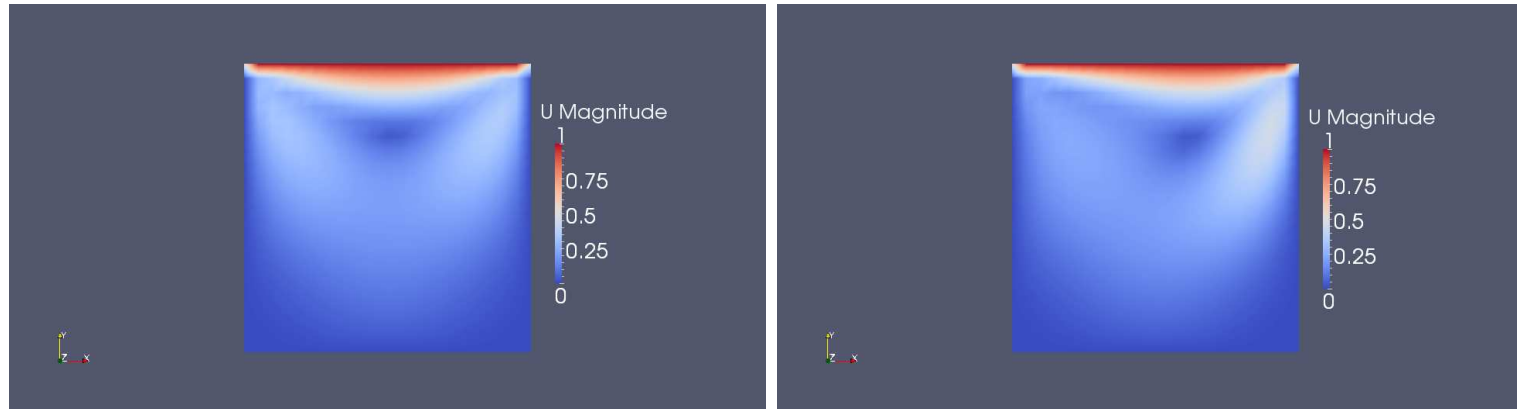
- Top: Pressure distribution and wireframe display.
- Bottom: Plot by line where the line is taken along the x-direction at the cavity top.
- As before the left figures are showing the original cavity case for comparison. To the right is the cavityFine case.

CavityGrade



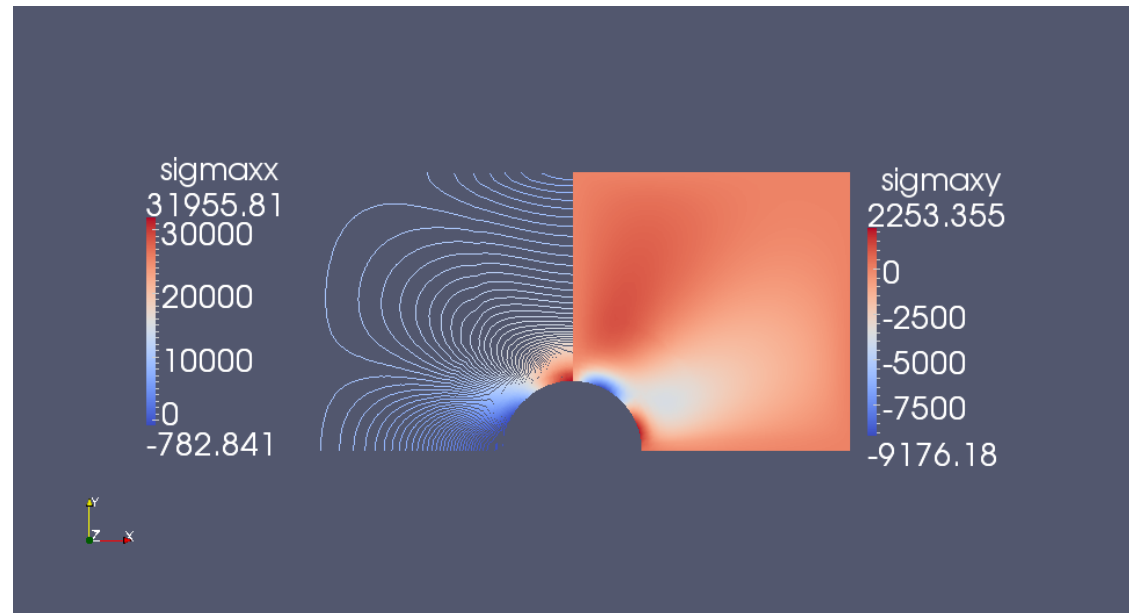
- Top: Pressure distribution and wireframe display of the coarse and fine mesh, respectively.
- Bottom: Filter "warp by scalar" with scale factor 0.01 set view direction to Y.

CavityHighRe



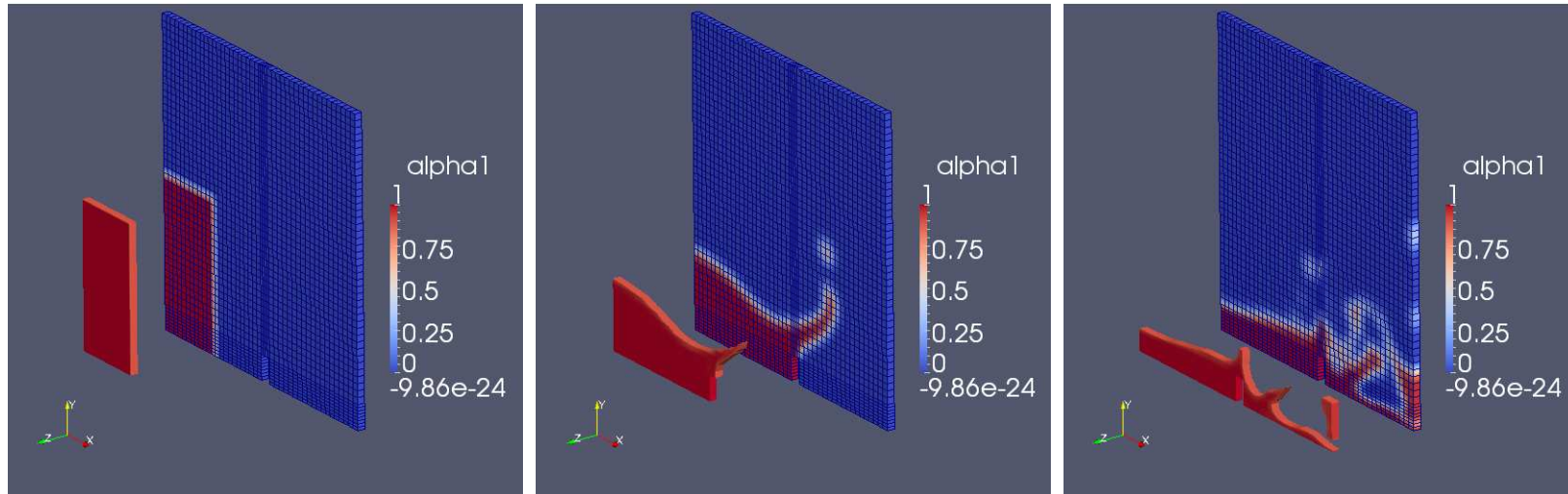
- Left: Velocity distribution for the cavity case.
- Right: Velocity distribution for the high Reynold cavity case.

plateHole



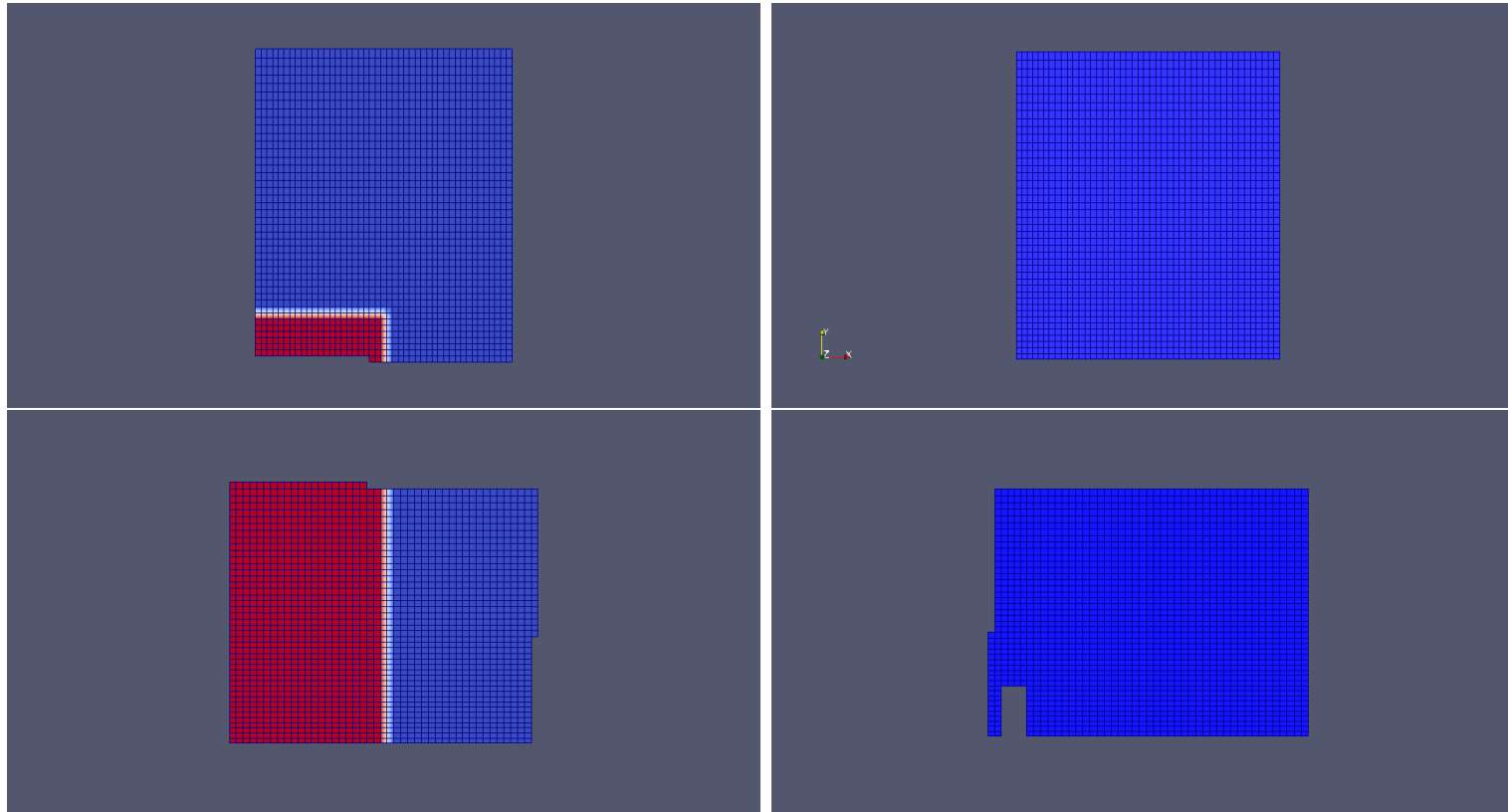
- Left: σ_{xx} using contourplot oriented -180 in the y-direction.
- Right: σ_{xy} displayed in an ordinary "surface" mode.

damBreak



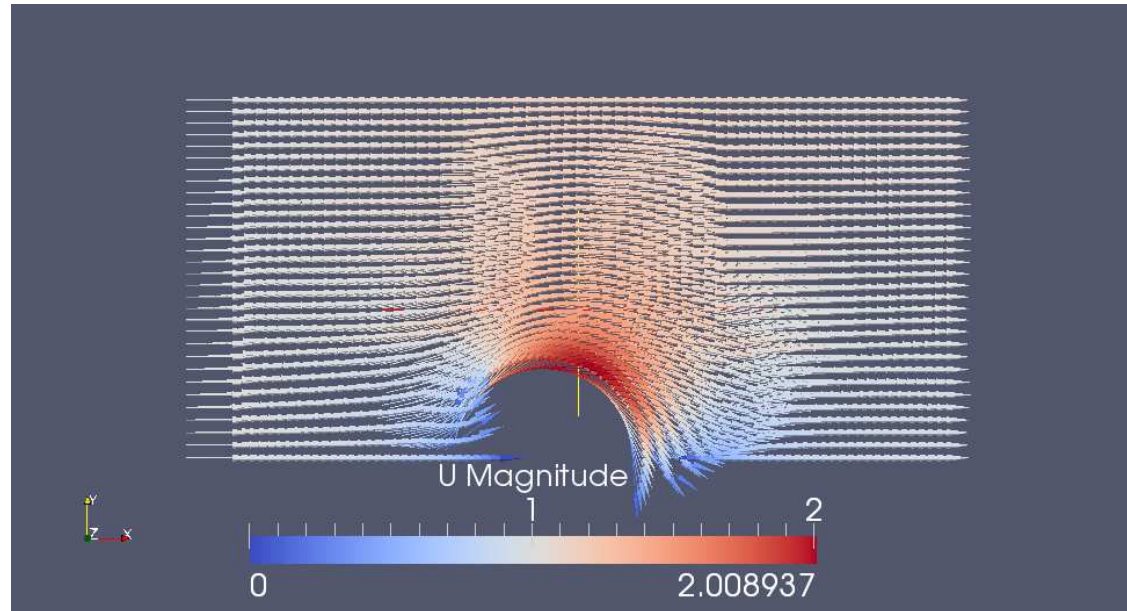
- "Clip" filter option is used where scalar is chosen as alpha1 and value is set to 0.90.
- The file damBreak.OpenFOAM is opened and translated -0.1 in the z-direction.
- Chosen times are 0,5 and 15, respectively.

damBreakFine



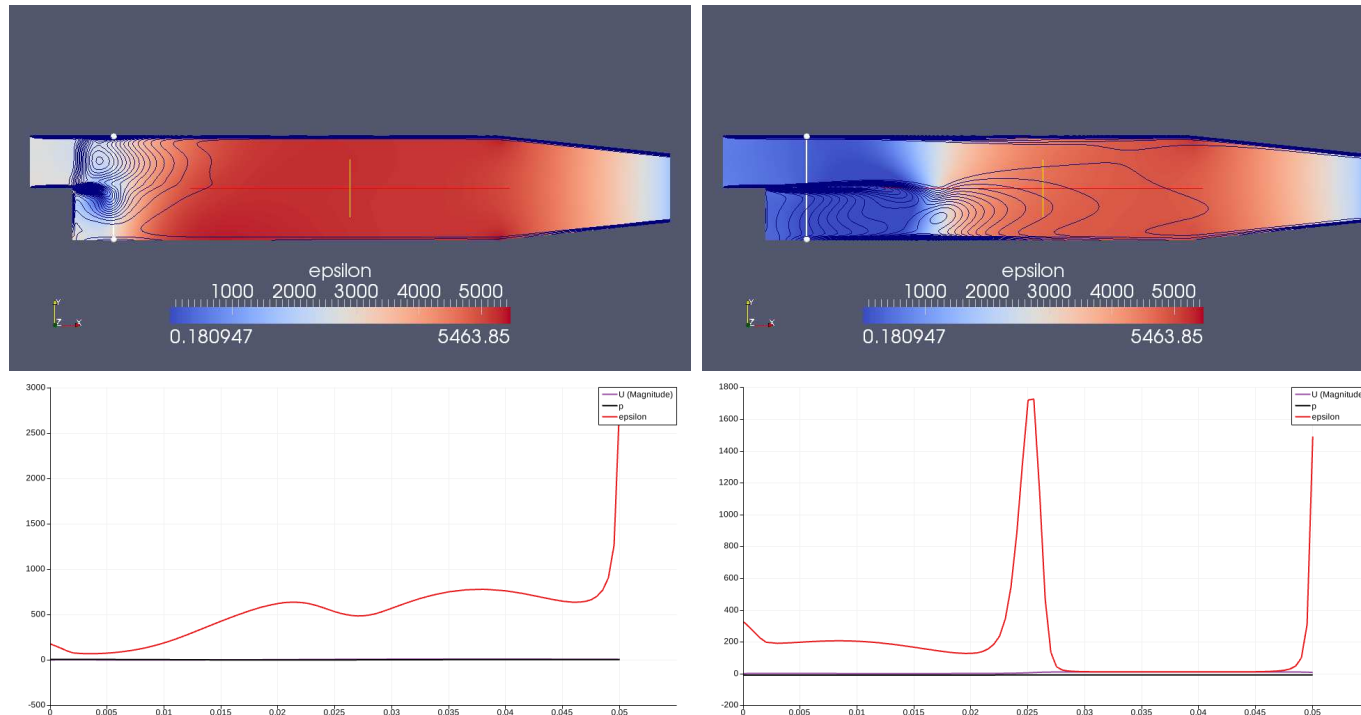
- The corresponding domains for each processor is illustrated and obtained using the command line "paraFoam -case processor0" and for processor1,processor2 and processor3.

cylinder



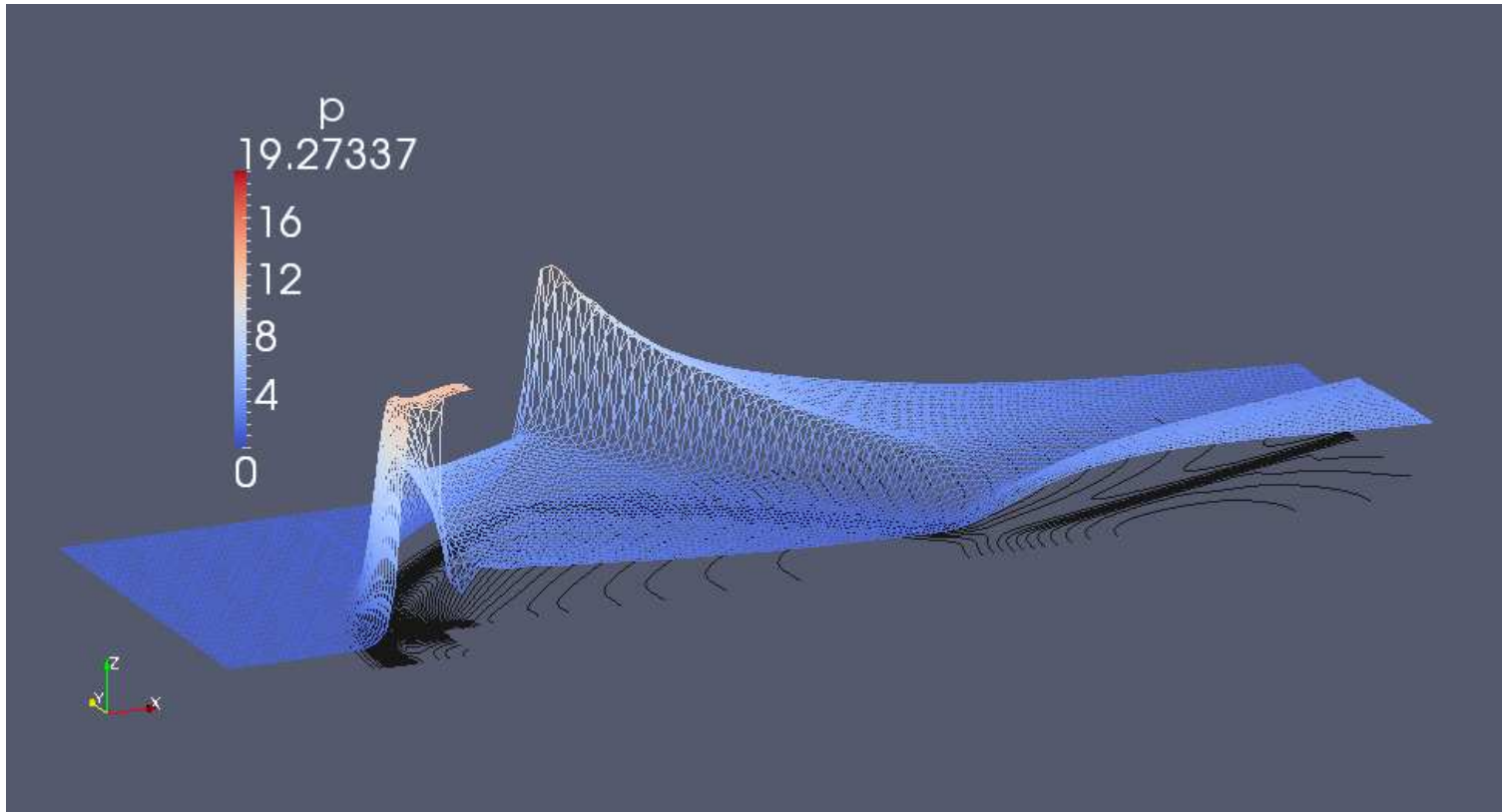
- Vectors of velocity is obtained using filter glyph. In order to be more clear, "maximum number of points" is set to 2500, "tip radius" to 0.05 and "shaft radius" to 0.0083.

pitzDaily



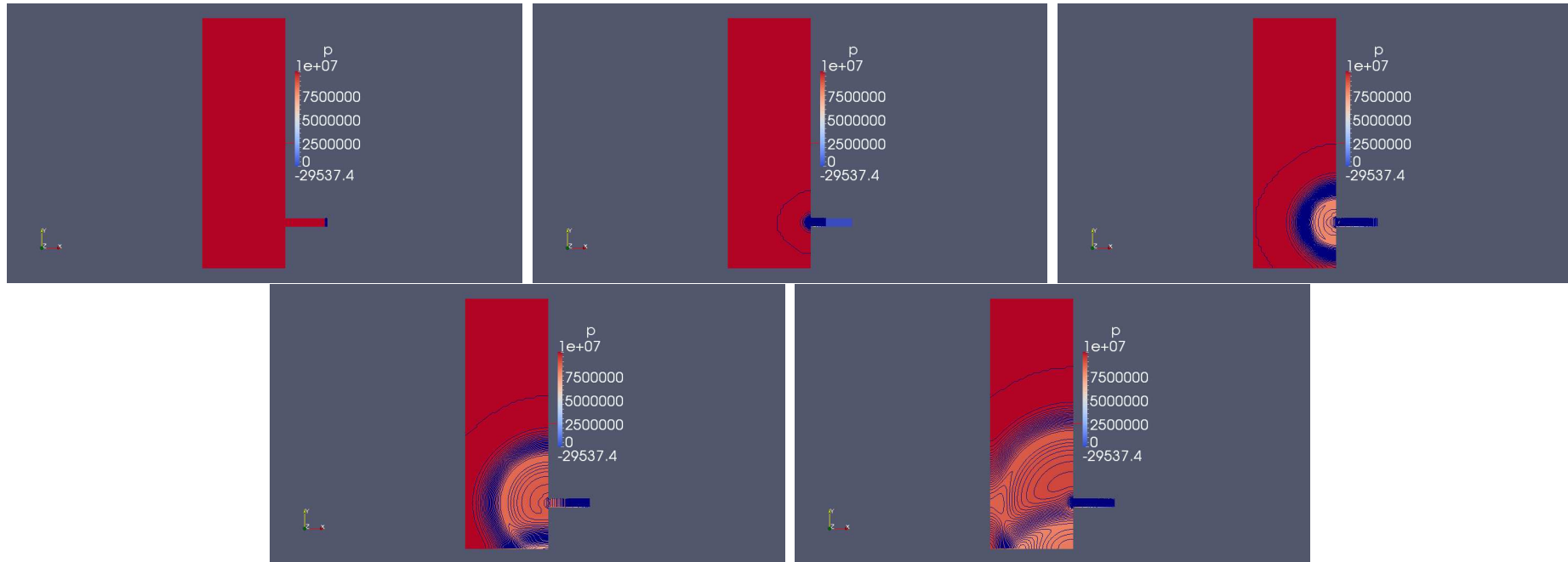
- Top: Dissipation for time 1 and 5 are illustrated using slice filter at $z=0$. Contour with a new range of 100 is also used.
- Bottom: "Plot over line" function between point1(0.02,-0.025,0) and point2(0.02,0.025,0), white line in top figures.

forwardStep



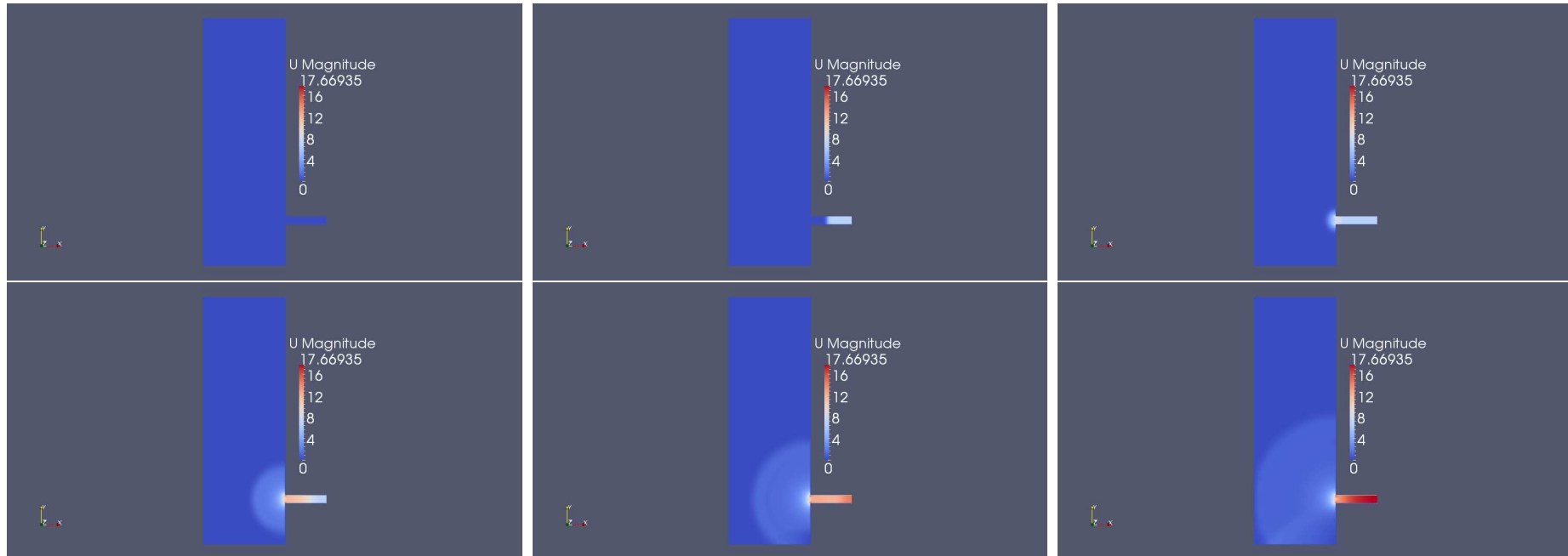
- Filter "warp by scalar" is used with a scalefactor set to 0.05. Isosurfaces of pressure is also illustrated in the figure at the bottom.

decompressionTank



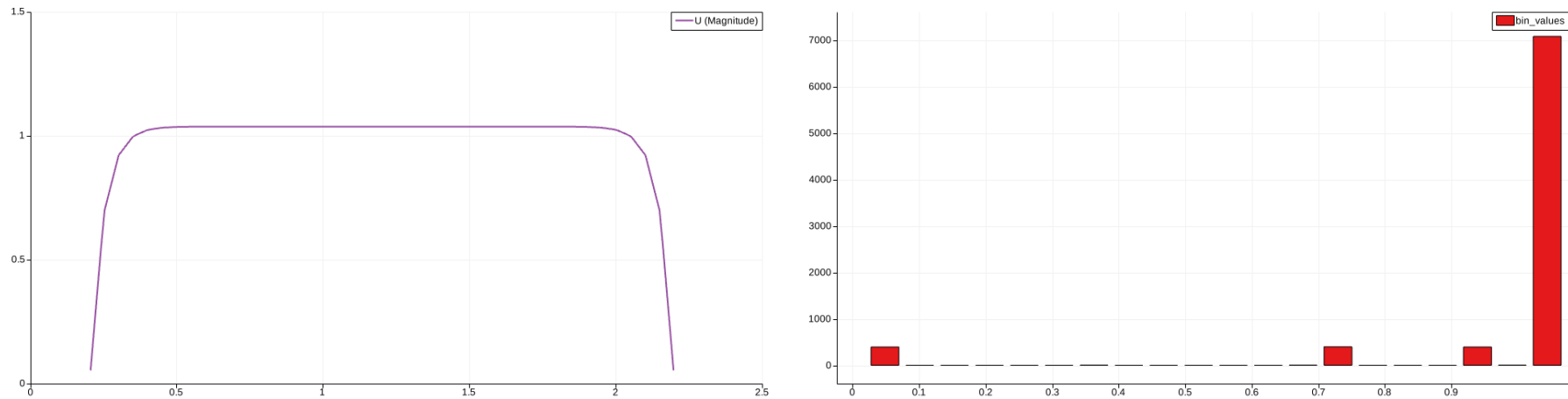
- Slice in the z normal direction and streamtracer function gives this look for times 10,20,30,40 and 50,respectively.

decompressionTankFine



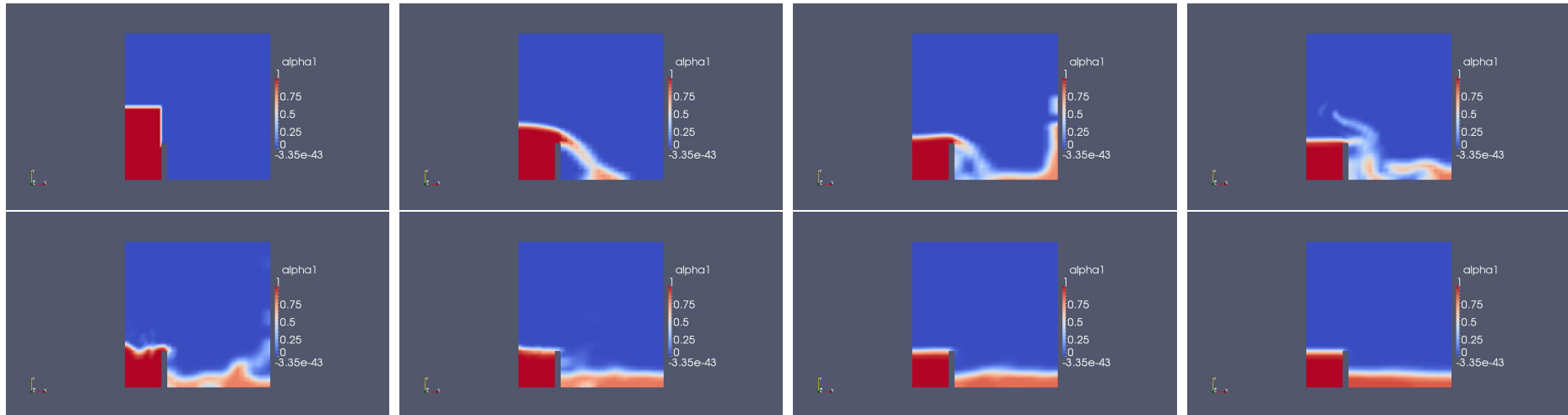
- Velocity distribution taken at time 0,10,20,30,40 and 50.

hartmann



- Left: Velocity profile in the domain along y .
- Right: Histogram of the velocity

Modified damBreak



- Figures corresponds to time 0,5,10,20,30,40,60 and 90, respectively.

The obstacle has been moved to a new location. It has also been elongated to further look at what happens when that space is filled. In order to see that the simulation time has been increased. The blockMeshDict file from my modified damBreak case is included.

```
/*-----*-- C++ -*-----
| =====
|  \ \      /  F i e l d      |  OpenFOAM: The Open Source CFD Toolbox
|  \ \      /  O p e r a t i o n  |  Version:  1.7.1
|   \ \    /   A n d           |  Web:       www.OpenFOAM.com
|    \ \ /    M a n i p u l a t i o n  |
\*-----
```

```
FoamFile
```

```
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       blockMeshDict;
}
```

```
// * * * * *
```

```
convertToMeters 0.146;
```

```
vertices
```

```
(
    (0 0 0)
```

(1 0 0)
(1.165068 0 0)
(4 0 0)
(0 1 0)
(1 1 0)
(1.165068 1 0)
(4 1 0)
(0 4 0)
(1 4 0)
(1.165068 4 0)
(4 4 0)
(0 0 0.1)
(1 0 0.1)
(1.165068 0 0.1)
(4 0 0.1)
(0 1 0.1)
(1 1 0.1)
(1.165068 1 0.1)
(4 1 0.1)
(0 4 0.1)
(1 4 0.1)

```
(1.165068 4 0.1)
(4 4 0.1)
);

blocks
(
  hex (0 1 5 4 12 13 17 16) (23 8 1) simpleGrading (1 1 1)
  hex (2 3 7 6 14 15 19 18) (19 8 1) simpleGrading (1 1 1)
  hex (4 5 9 8 16 17 21 20) (23 42 1) simpleGrading (1 1 1)
  hex (5 6 10 9 17 18 22 21) (4 42 1) simpleGrading (1 1 1)
  hex (6 7 11 10 18 19 23 22) (19 42 1) simpleGrading (1 1 1)
);

edges
(
);

patches
(
  wall leftWall
  (
```

```
        (0 12 16 4)
        (4 16 20 8)
)
wall rightWall
(
    (7 19 15 3)
    (11 23 19 7)
)
wall lowerWall
(
    (0 1 13 12)
    (1 5 17 13)
    (5 6 18 17)
    (2 14 18 6)
    (2 3 15 14)
)
patch atmosphere
(
    (8 20 21 9)
    (9 21 22 10)
    (10 22 23 11)
```

```
)  
);
```

```
mergePatchPairs
```

```
(  
);
```

```
// *****
```