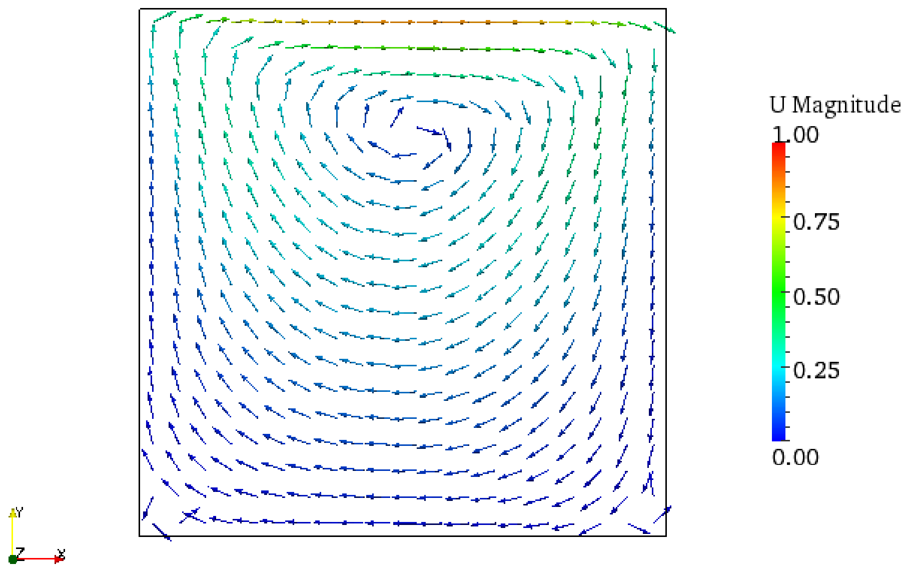
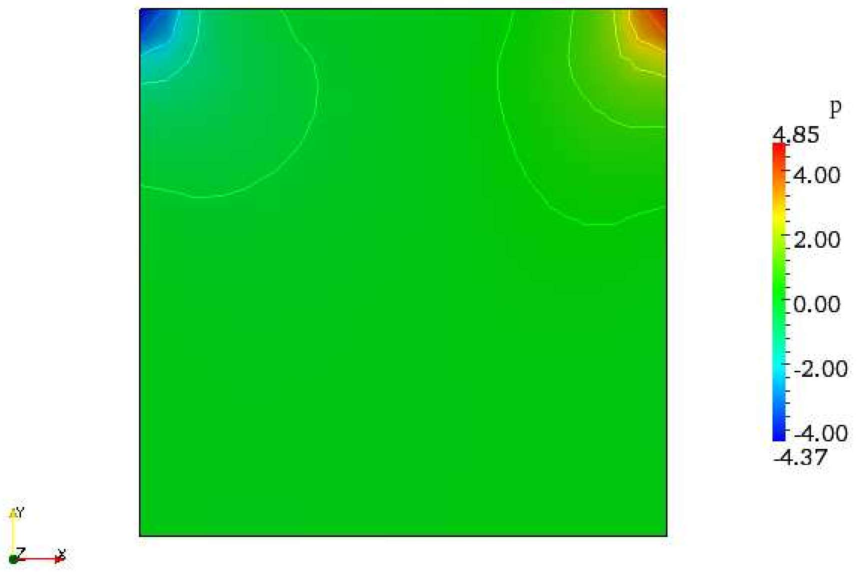
Assignment 1

- icoFoam: cavity, cavityClipped, cavityFine, cavityGrade, cavityHighRe
- solidDisplacementFoam: plateHole
- interFoam/laminar: damBreak, damBreakFine
- potentialFoam: cylinder
- simpleFoam: pitzDaily
- sonicFoam: forwardStep
- sonicLiquidFoam: decompressionTank, decompressionTankFine
- mhdFoam: hartmann

cavity



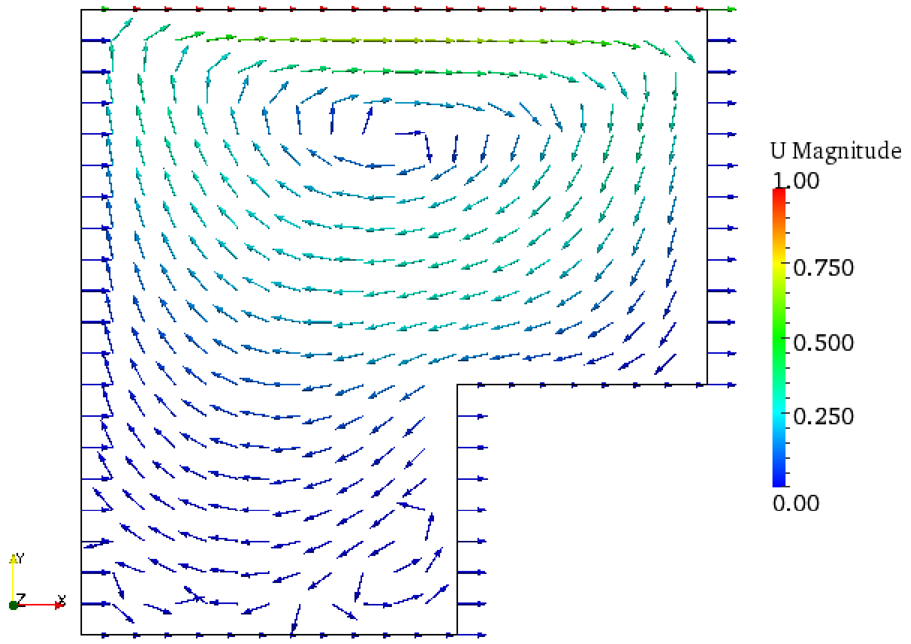


1. Create a *Slice* in z-plane
2. Select *Cell centers* and *Glyph* from the *Filter* menu

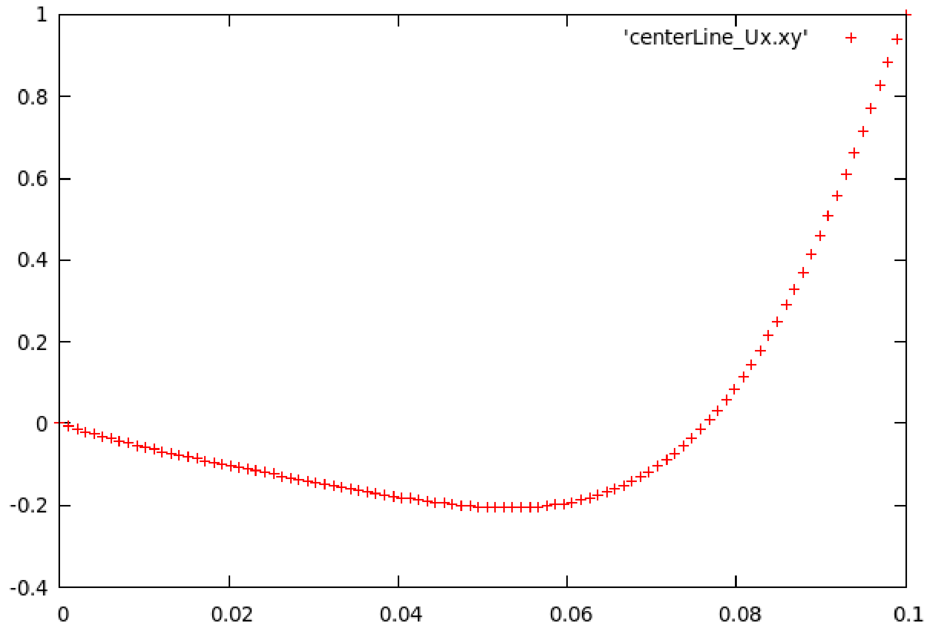
1. Create a *Slice* in z-plane showing the pressure
2. Select *Contour* filter to show isolines of pressure

cavityClipped

1. Highlight cavityClipped.OpenFOAM
2. Create a *Slice* in the z-plane
3. Select *Glyph* from the *Filter* menu
4. Open cavityClipped.OpenFOAM
5. Select only lid and fixed walls in Mesh parts
6. Display wireframe to get the outline of the clipped geometry



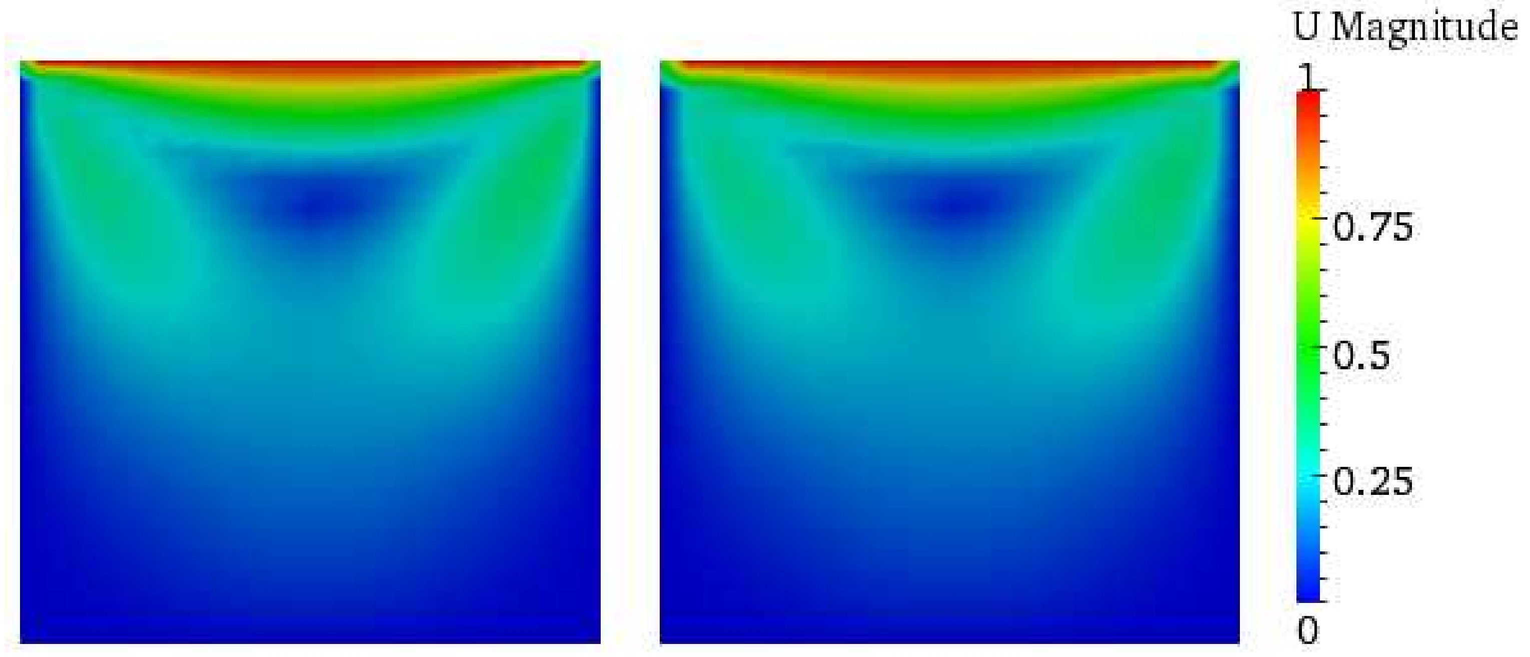
cavityFine



A sampleDict file is created to obtain Ux data along the y axis in the center of the domain. This data can be visualised by gnuplot with the command `plot 'sets/0.7/centerLine_Ux.xy'`

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      sampleDict;
}
// * * * * * * * * * * * * * * * * * //

interpolationScheme cellPoint;

setFormat       raw;

sets
(
    centerLine
    {
        type    uniform;
        axis    y;
        start   ( 0.05 0 0.005 );
        end     ( 0.05 0.1 0.005 );
        nPoints 100;
    }
);

surfaces        ();

fields          ( Ux );
```

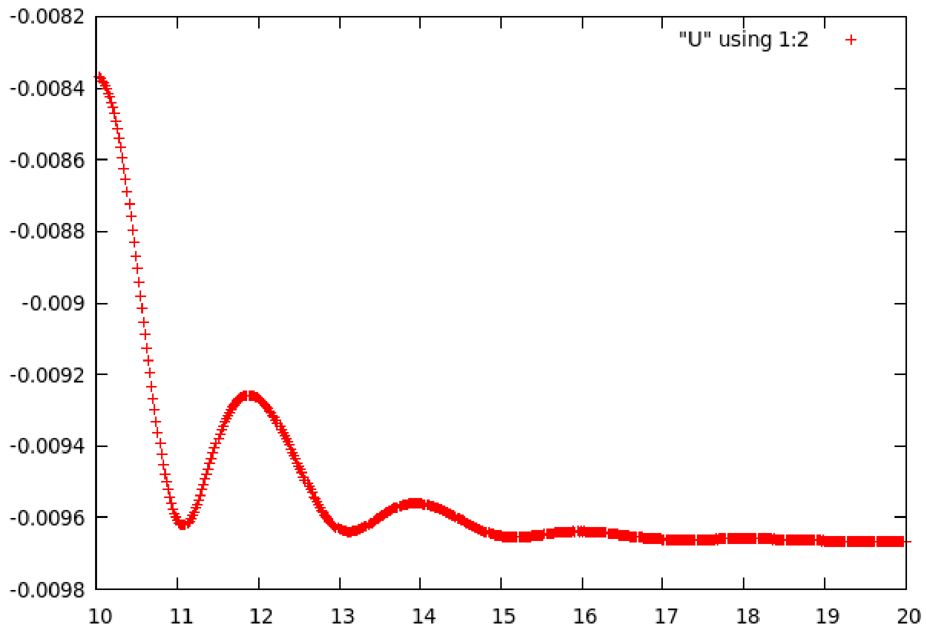
cavityGrade



Velocity in the cavityGrade case (left) and the cavity case (right)

Both the cavity and the cavityGrade cases have been opened in paraview. The cavityGrade.OpenFOAM has been tranlated -0.11 in the x-direction.

cavityHighRe

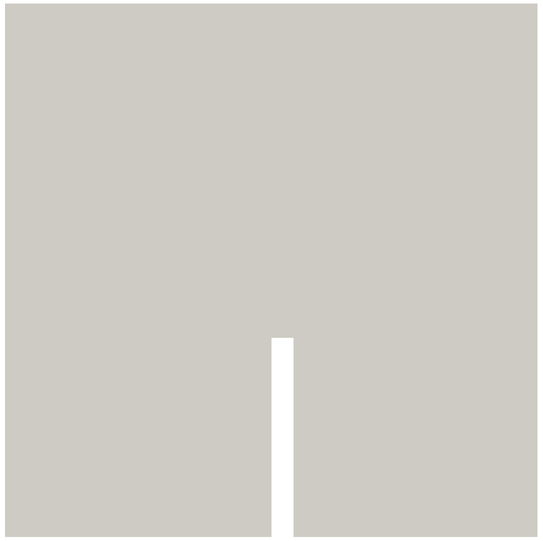


The objectFunction probes in inserted in the controlDict file. The convergence of Ux can now be monitored in gnuplot with the command `plot "probes/10/U" using 1:2`

```
functions
{
    probes
    {
        type                probes;
        functionObjectLibs ("libsampling.so");
        enabled             true;
        outputControl       timeStep;
        outputInterval      1;
        probeLocations
        (
            ( 0.05 0.05 0.005 )
        );

        fields
        (
            p
   U
        );
    }
}
```

cavityObstacle

The geometry of the cavity case has been changed and a new simulation has been performed. As you can see the new geometry reminds of the geometry in the damBreak case. I have copied the blockMeshDict file from the damBreak case into the cavityObstacle case and changed the height of the obstacle and the names of the patches, see the included blockMeshDict file.

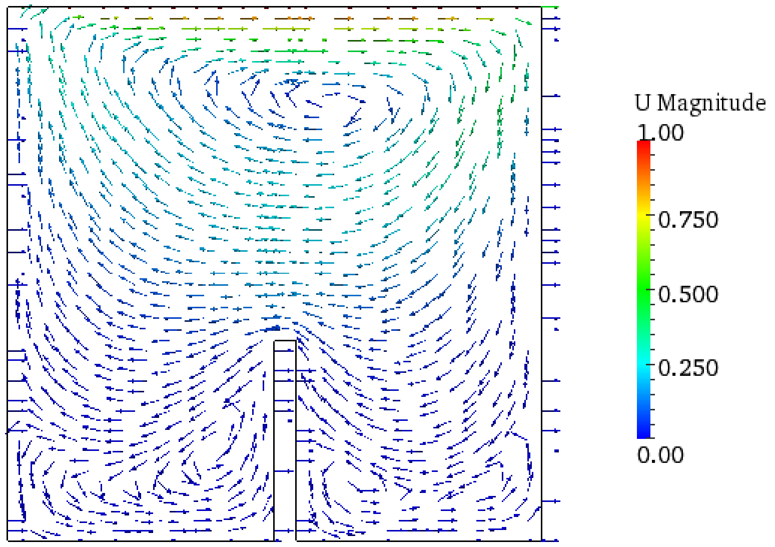


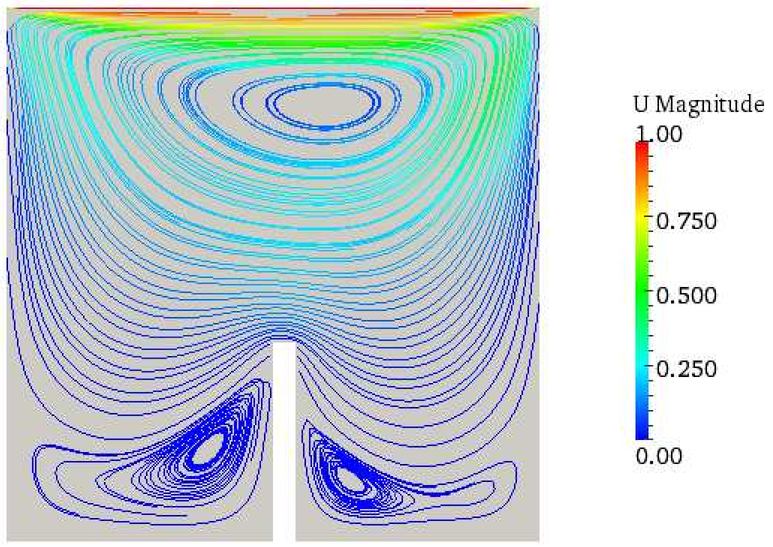
```
convertToMeters 0.146;

vertices
(
    (0 0 0)
    (2 0 0)
    (2.16438 0 0)
    (4 0 0)
    (0 1.5 0)
    (2 1.5 0)
    (2.16438 1.5 0)
    (4 1.5 0)
    (0 4 0)
    (2 4 0)
    (2.16438 4 0)
    (4 4 0)
    (0 0 0.1)
    (2 0 0.1)
    (2.16438 0 0.1)
    (4 0 0.1)
    (0 1.5 0.1)
    (2 1.5 0.1)
    (2.16438 1.5 0.1)
    (4 1.5 0.1)
    (0 4 0.1)
    (2 4 0.1)
    (2.16438 4 0.1)
    (4 4 0.1)
);
```

cavityObstacle

```
blocks
(
    hex (0 1 5 4 12 13 17 16) (23 20 1) simpleGrading (1 1 1)
    hex (2 3 7 6 14 15 19 18) (19 20 1) simpleGrading (1 1 1)
    hex (4 5 9 8 16 17 21 20) (23 30 1) simpleGrading (1 1 1)
    hex (5 6 10 9 17 18 22 21) (4 30 1) simpleGrading (1 1 1)
    hex (6 7 11 10 18 19 23 22) (19 30 1) simpleGrading (1 1 1)
);

edges
(
);

patches
(
    wall fixedWalls
    (
        (0 12 16 4)
        (4 16 20 8)
        (7 19 15 3)
        (11 23 19 7)
        (0 1 13 12)
        (1 5 17 13)
        (5 6 18 17)
        (2 14 18 6)
        (2 3 15 14)
    )
    patch movingWall
    (
        (8 20 21 9)
        (9 21 22 10)
        (10 22 23 11)
    )
);

mergePatchPairs
(
);
```

cavityObstacle

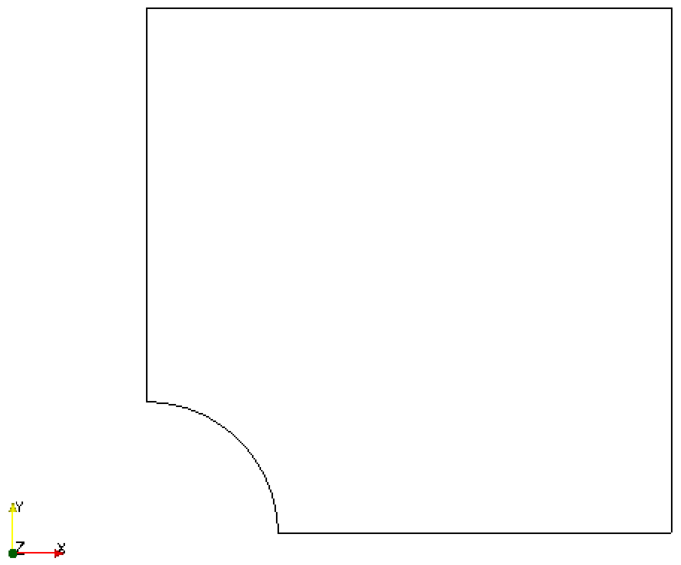


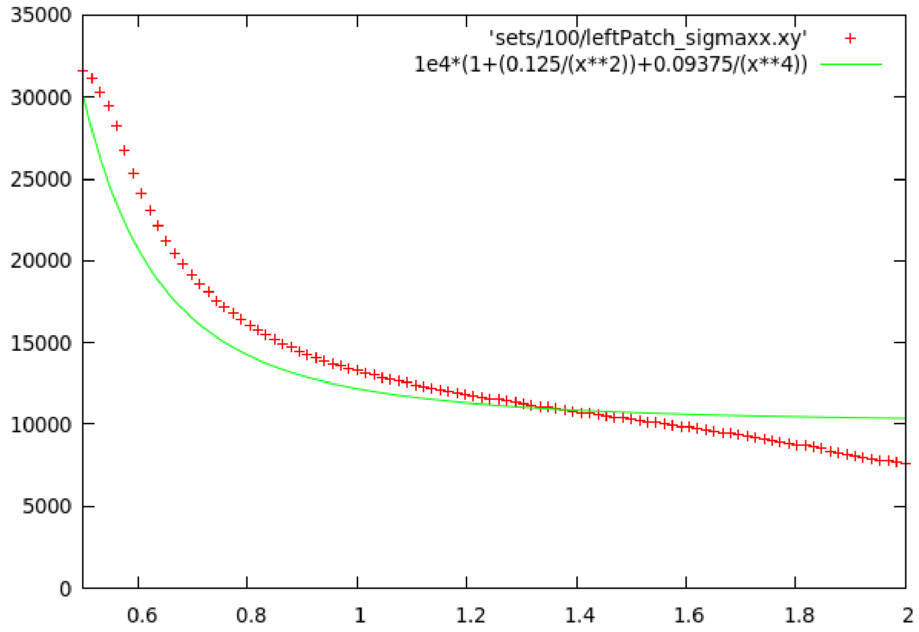


1. Create a *Slice* in z-plane
2. Select *Glyph* from the *Filter* menu.

1. Select the *Stream Tracer* to show the two developed vortices on each side of the obstacle.

plateHole

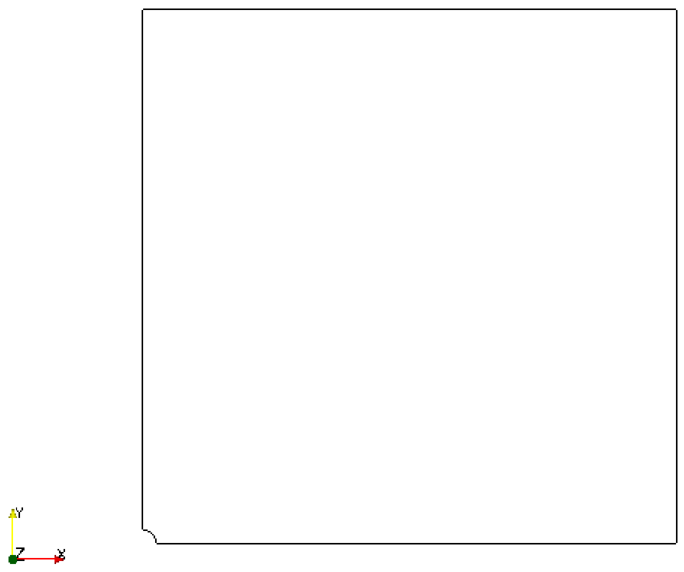


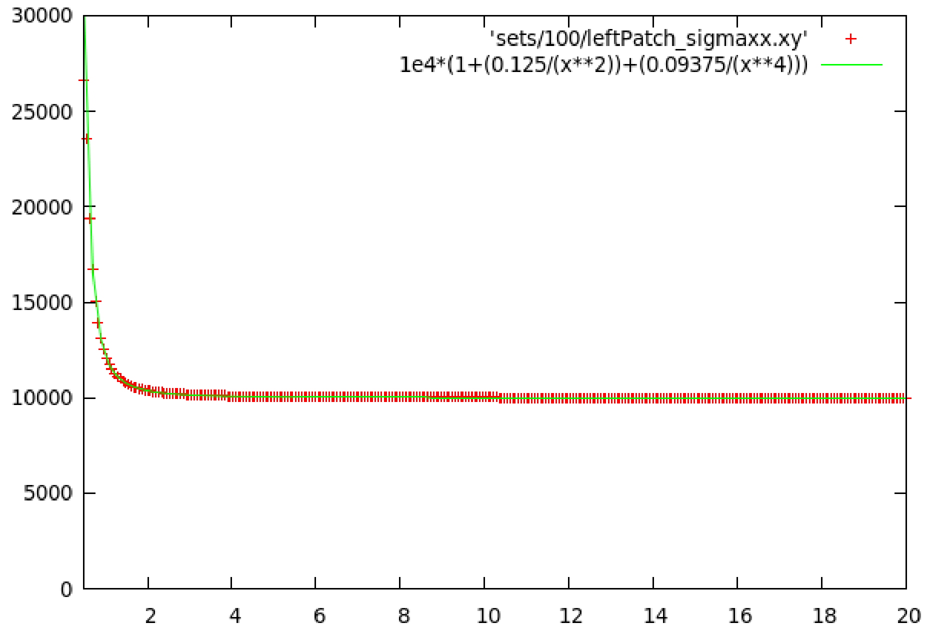


For an infinitely large, thin plate with a circular hole there exists an analytical solution for the stress normal to the vertical plane of symmetry (left wall of the geometry seen in the left figure). The numerical solution is compared to this analytical solution (right figure), and we see that the numerical solution is not completely accurate.

plateHole

In order to reduce the error of the numerical solution the size of the plate is increased. We now get a much better agreement between the numerical and the analytical solution.

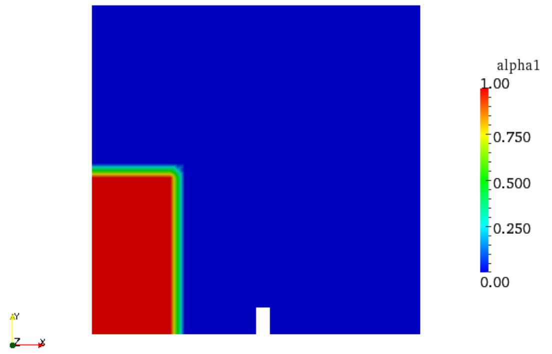




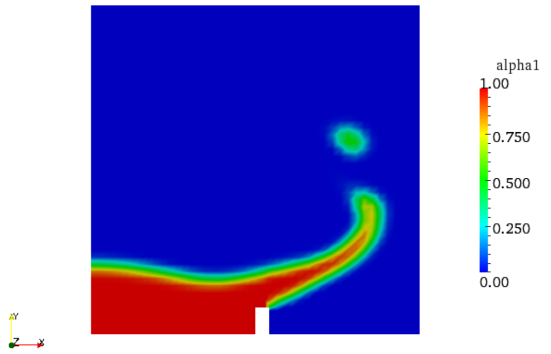
The plot is created in gnuplot by typing:

```
plot [0.5:2] [0:]  'sets/100/leftPatch_sigmaxx.xy', 1e4*(1+(0.125/(x**2))+(0.09375/(x**4)))
```

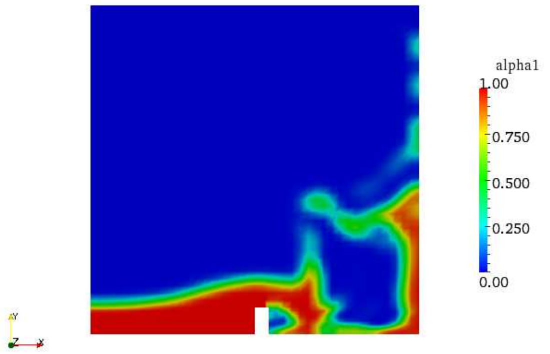
damBreak



(a) At t=0s



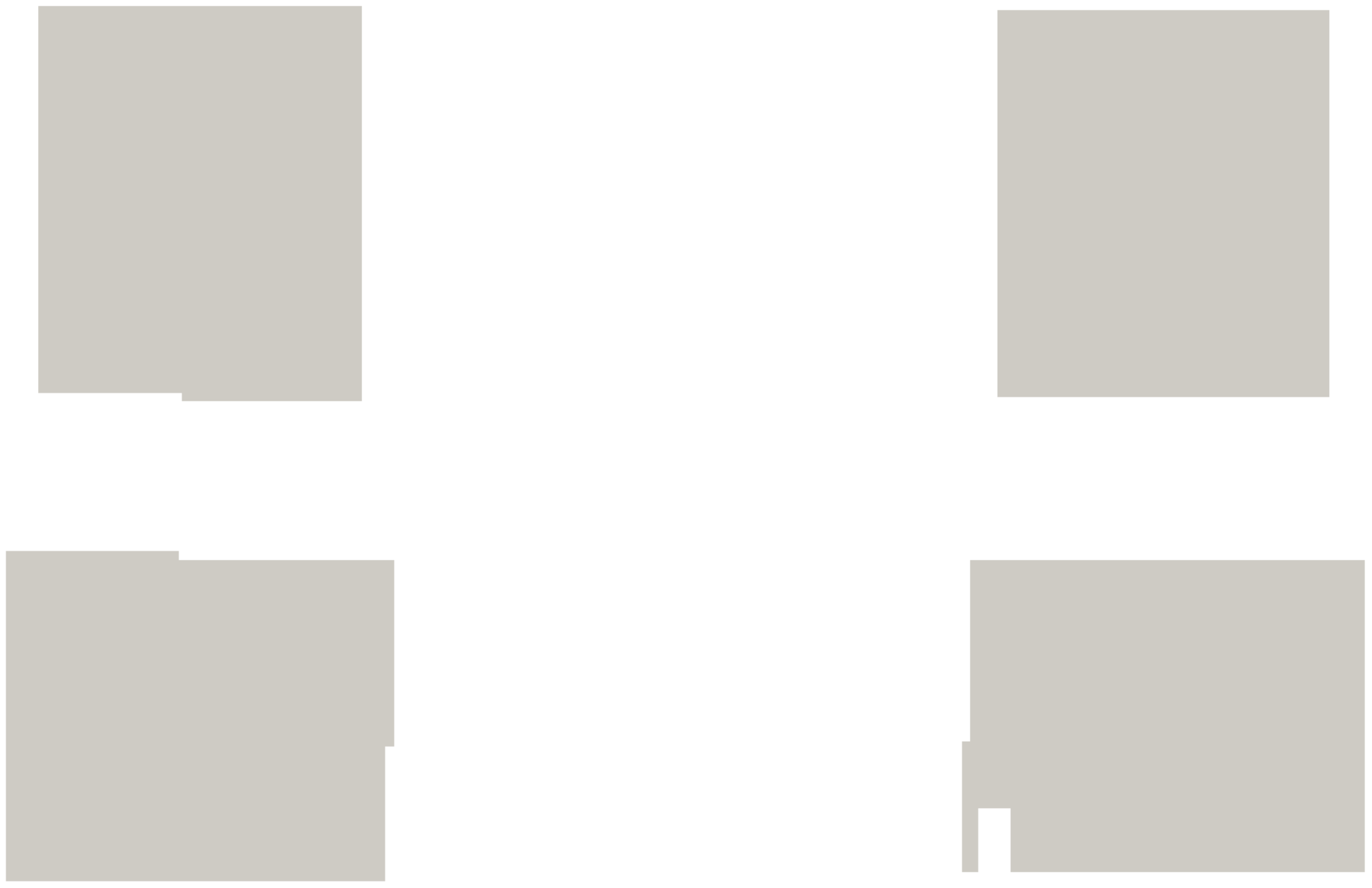
(b) At t=0.30s



(c) At t=0.60s

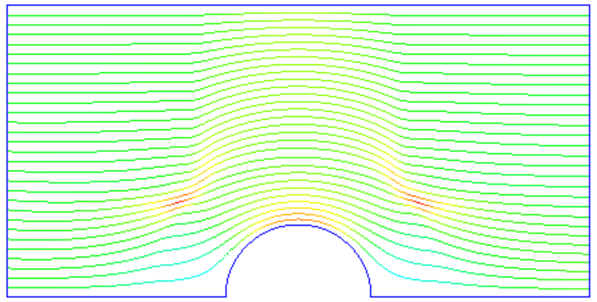
These pictures are obtained by toggeling alpha1 in Object Inspector Volume Fields and the displaying alpha1.

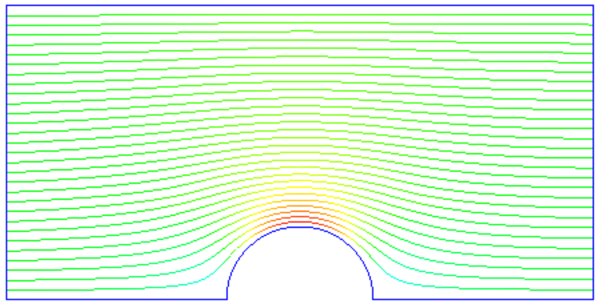
damBreakFine

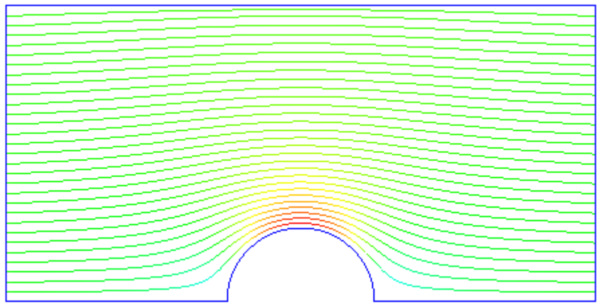


This is the damBreak geometry divided into four pieces in order to do parallel computing.

cylinder

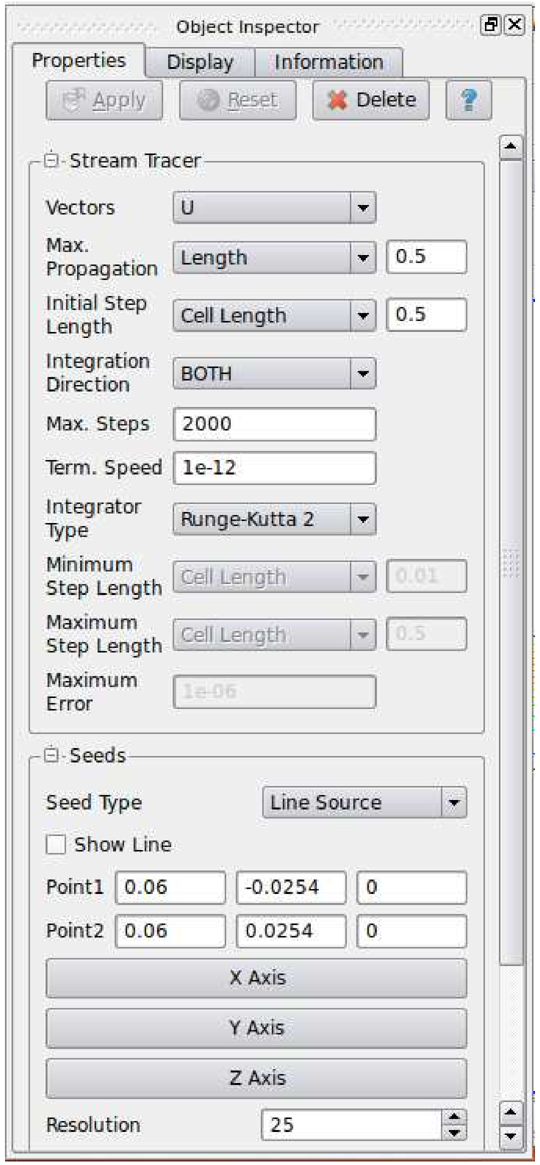

(a) No non-orthogonal correction


(b) Non-orthogonal correction = 3


(c) Analytical solution

1. Highlight cylinder.OpenFOAM
2. Select *Stream Tracer* from the *Filter* menu
3. Create a line source at (-2 2 0) and (-2 0 0) and change resolution to 30
4. Max propagation length is set to 5 and initial step length is set to 0.01
5. Color the streamlines by U

pitzDaily

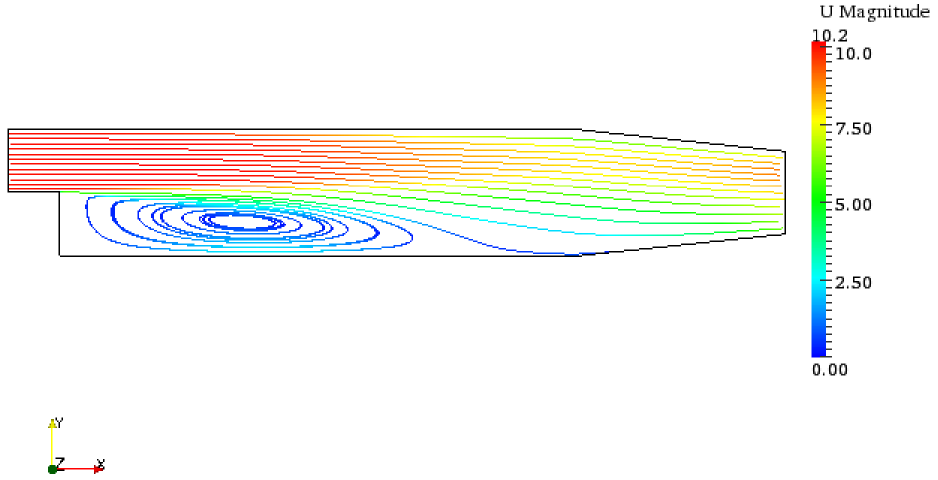


Stream Tracer

Vectors	U	
Max. Propagation	Length	0.5
Initial Step Length	Cell Length	0.5
Integration Direction	BOTH	
Max. Steps	2000	
Term. Speed	1e-12	
Integrator Type	Runge-Kutta 2	
Minimum Step Length	Cell Length	0.01
Maximum Step Length	Cell Length	0.5
Maximum Error	1e-06	

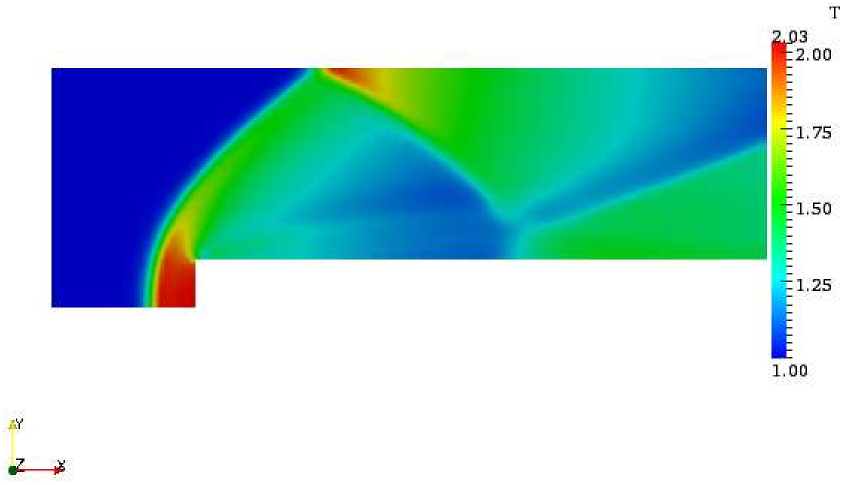
Seeds

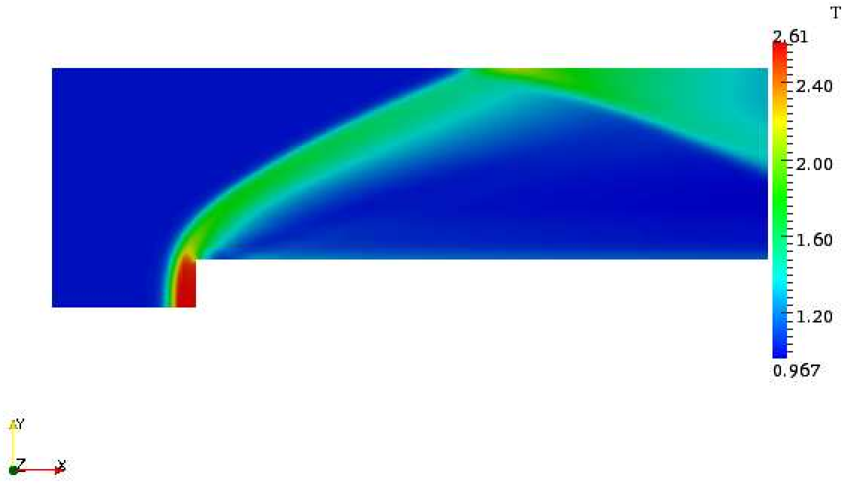
Seed Type	Line Source		
☐ Show Line			
Point1	0.06	-0.0254	0
Point2	0.06	0.0254	0
Resolution	25		



The streamlines is created with the settings shown in the Object Inspector window.

forwardStep - forwardStepFaster

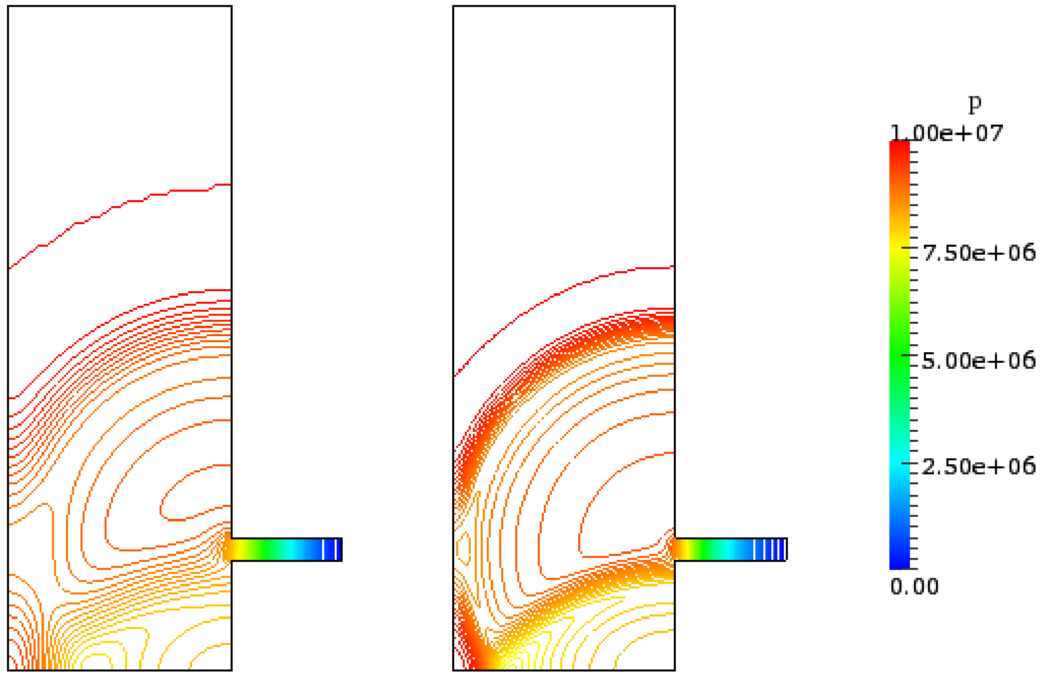




The left figure shows the temperature distribution of the converged case with an inlet velocity $U = Mach3$. The right figure shows the calculated temperature distribution where the inlet velocity has been increased to $U = Mach5$.

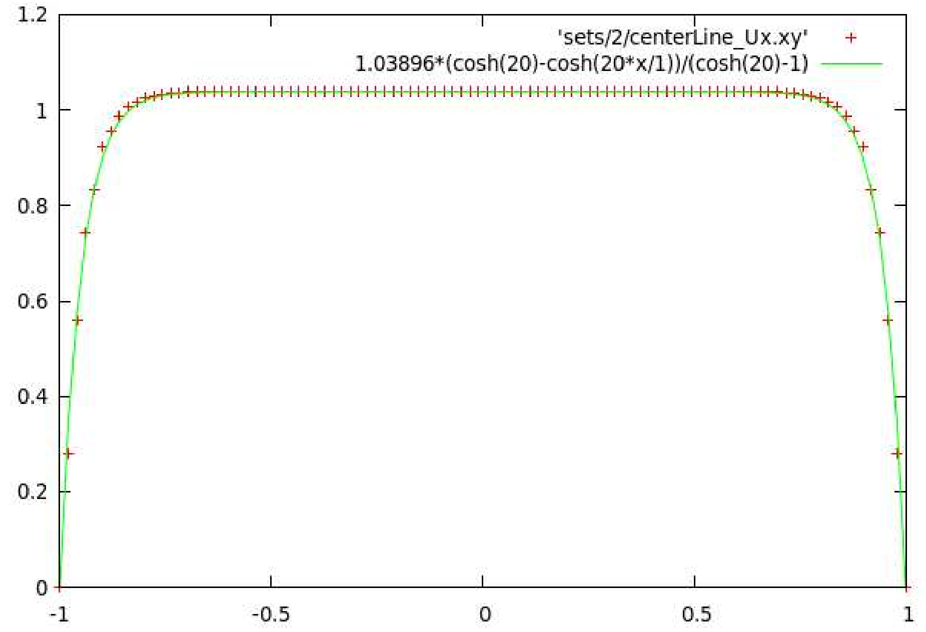
1. In order to visualize the temperature distribution the T has to be toggled in *Volume Fields*.
2. T can now be selected and displayed.

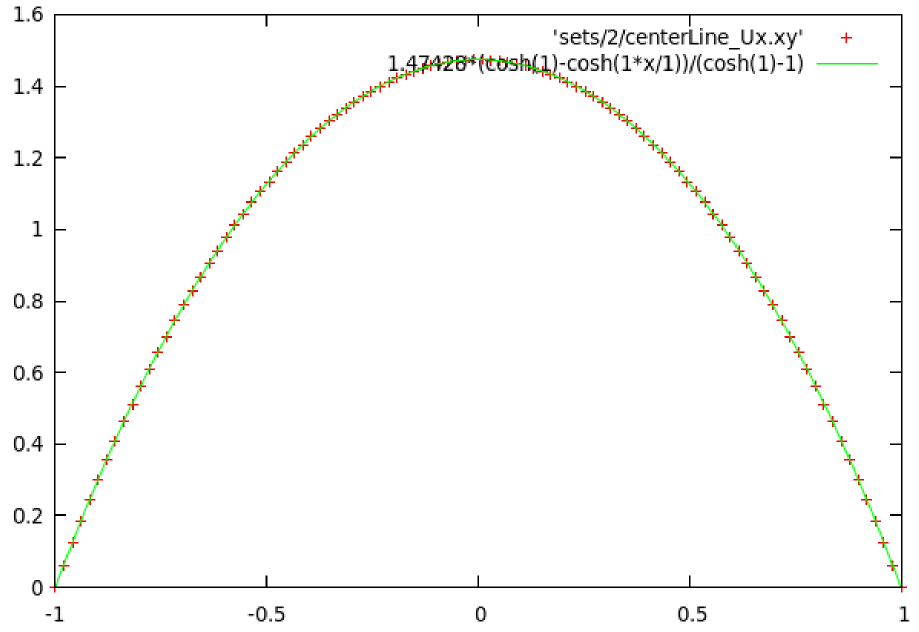
decompressionTank - decompressionTankFine



In order to view two different cases at the same time we first open decompressionTank with the command paraFoam. The file *decompressionTankFine.OpenFOAM* is then created using the latex command *touch*, and this case can now be opened in paraview. The *decopmressionTankFine.OpenFOAM* must be translated -0.2 in the x-direction.

hartmann





These figures show the velocity profile in the Hartmann problem for $B_y = 20T$ (left) and $B_y = 1T$ (right). The red crosses represent the numerical solution and the green line represents the analytical solution given by:

$$\frac{U_x(y)}{U_x(0)} = \frac{coshM - coshM(y/L)}{coshM - 1}$$

hartmann

$U_x(0)$ can be obtained by inserting the functionObject probes at the end of the controlDict file.

```
functions
{
    probes
    {
        type                probes;
        functionObjectLibs ("libsampling.so");
        enabled             true;
        outputControl       timeStep;
        outputInterval      1;
        probeLocations
        (
            ( 10 0 0.05 )
        );

        fields
        (
            U
        );
    }

}
```

In order to plot the U_x the following sampleDict file is needed.

```
interpolationScheme cellPoint;

setFormat       raw;

sets
(
    centerLine
    {
        type    uniform;
        axis    y;
        start   ( 10 -1 0.05 );
        end     ( 10 1 0.05 );
        nPoints 100;
    }
);

surfaces        ();

fields          ( Ux );
```

The plot for $B_y = 20T$ can now be created in gnuplot by typing:

```
    plot [-1:1] [0:]  'sets/2/centerLine_Ux.xy',
    1.03896*(cosh(20)-cosh(20*x/1))/(cosh(20)-1)
```