



OpenFOAM®

3rd Iberian Meeting

11 & 12 June, 2019 Porto - Portugal

Basic Courses (BII)

Post-processing

C. Fernandes / J.M. Nóbrega

cbpf@dep.uminho.pt / mnobrega@dep.uminho.pt

University of Minho



Load OpenFOAM

```
> . /opt/OpenFOAM/of6_envvars.sh
```

Copy tutorial files from the server

```
> cd $FOAM_RUN
```

```
> mkdir BII
```

```
> cd BII
```

```
> cp -r /FOAM-IBERIA-2019/Basic/  
Postprocessing$FOAM_RUN/BII
```

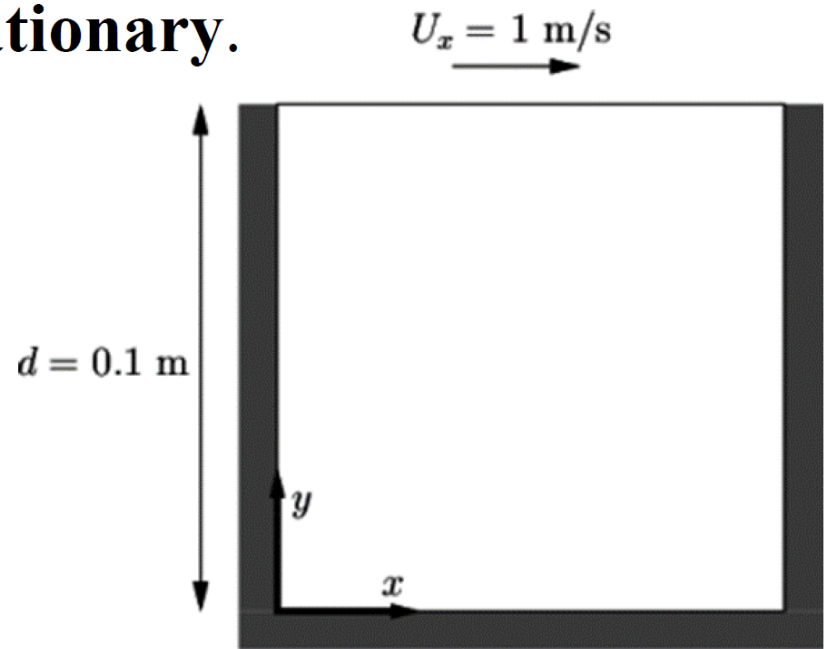
Tutorial

Lid Driven Cavity

Isothermal and Incompressible Fluid Flow

The geometry is a **two-dimensional** square domain which all the boundaries of the square are walls.

The **top wall moves** on the x-direction at a speed of 1 m/s while the **other three are stationary**.



Copy case files

```
> cd $FOAM_RUN/BII
> cp -r $FOAM_TUTORIALS/
incompressible/icoFOAM/cavity/cavity .

> cd cavity
```

Generate the mesh

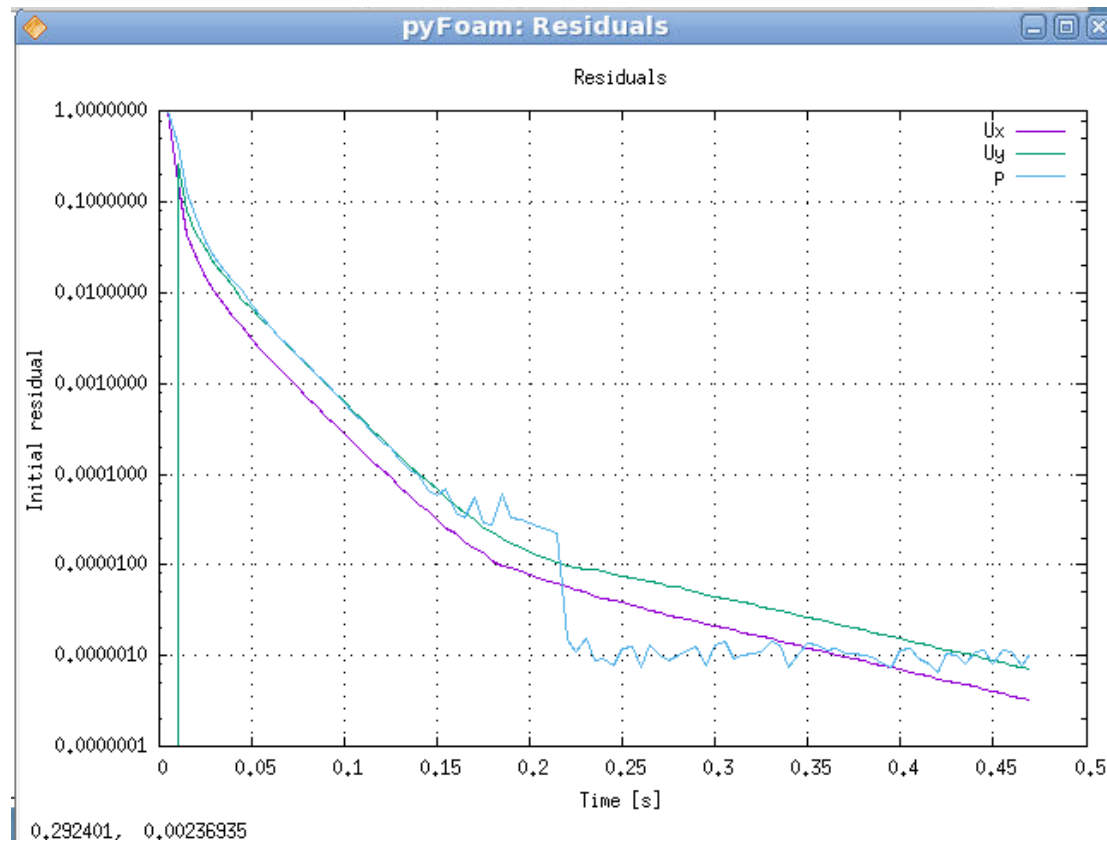
```
> blockMesh
```

Run the solver

```
> icoFoam > log.icoFoam
```

Visualize residuals evolution

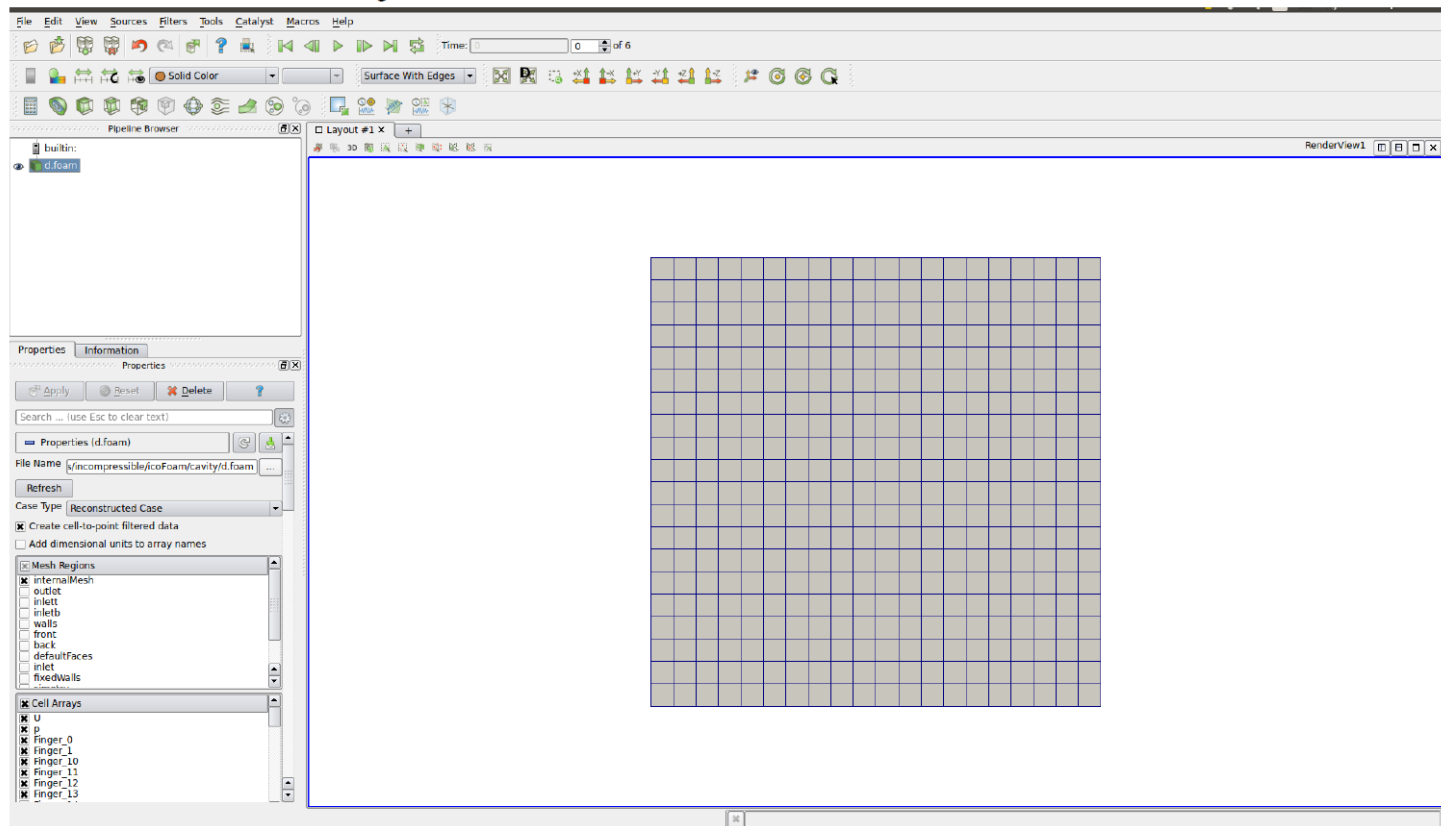
```
> pyFoamPlotWatcher.py log.icoFoam
```

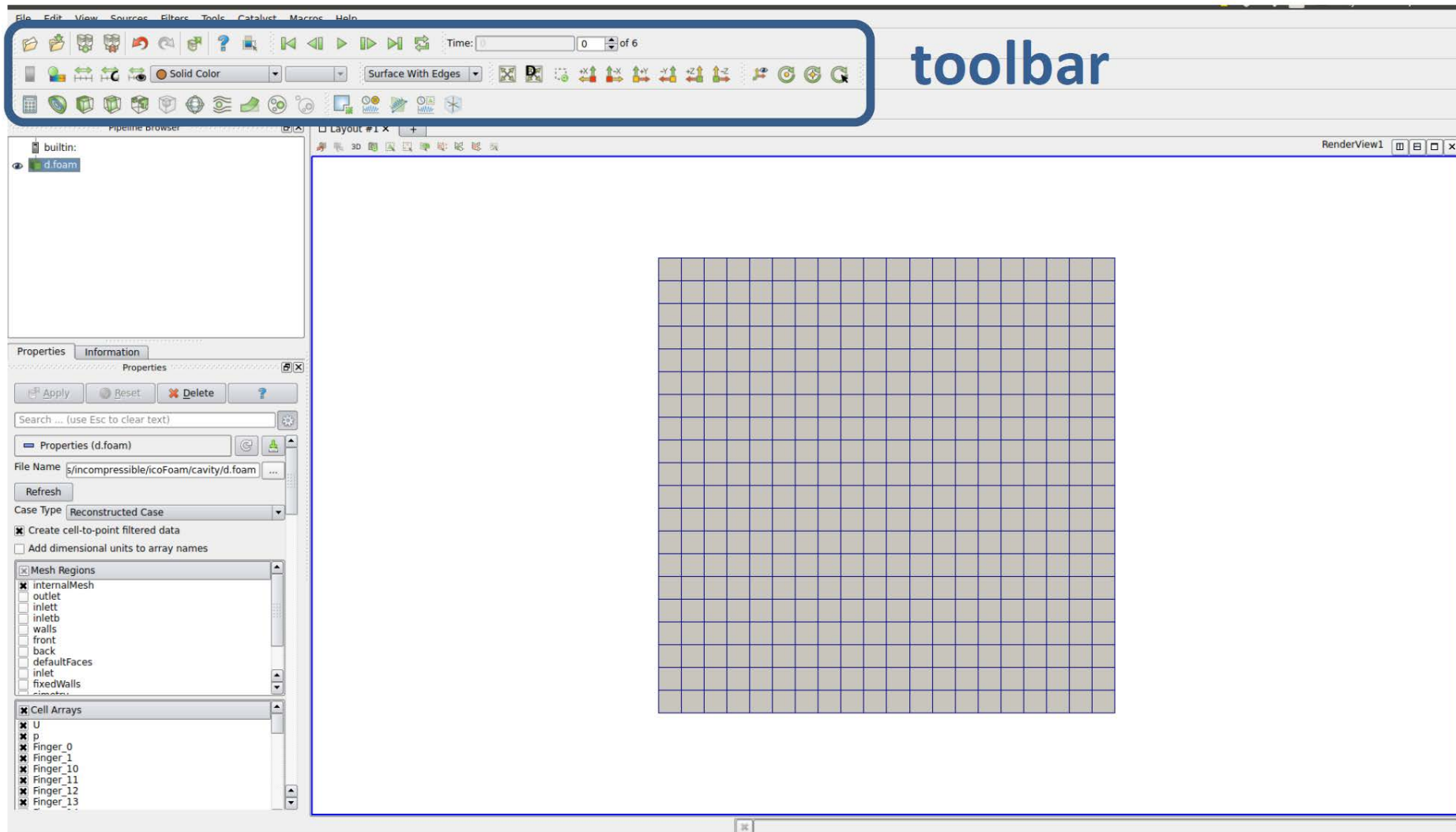


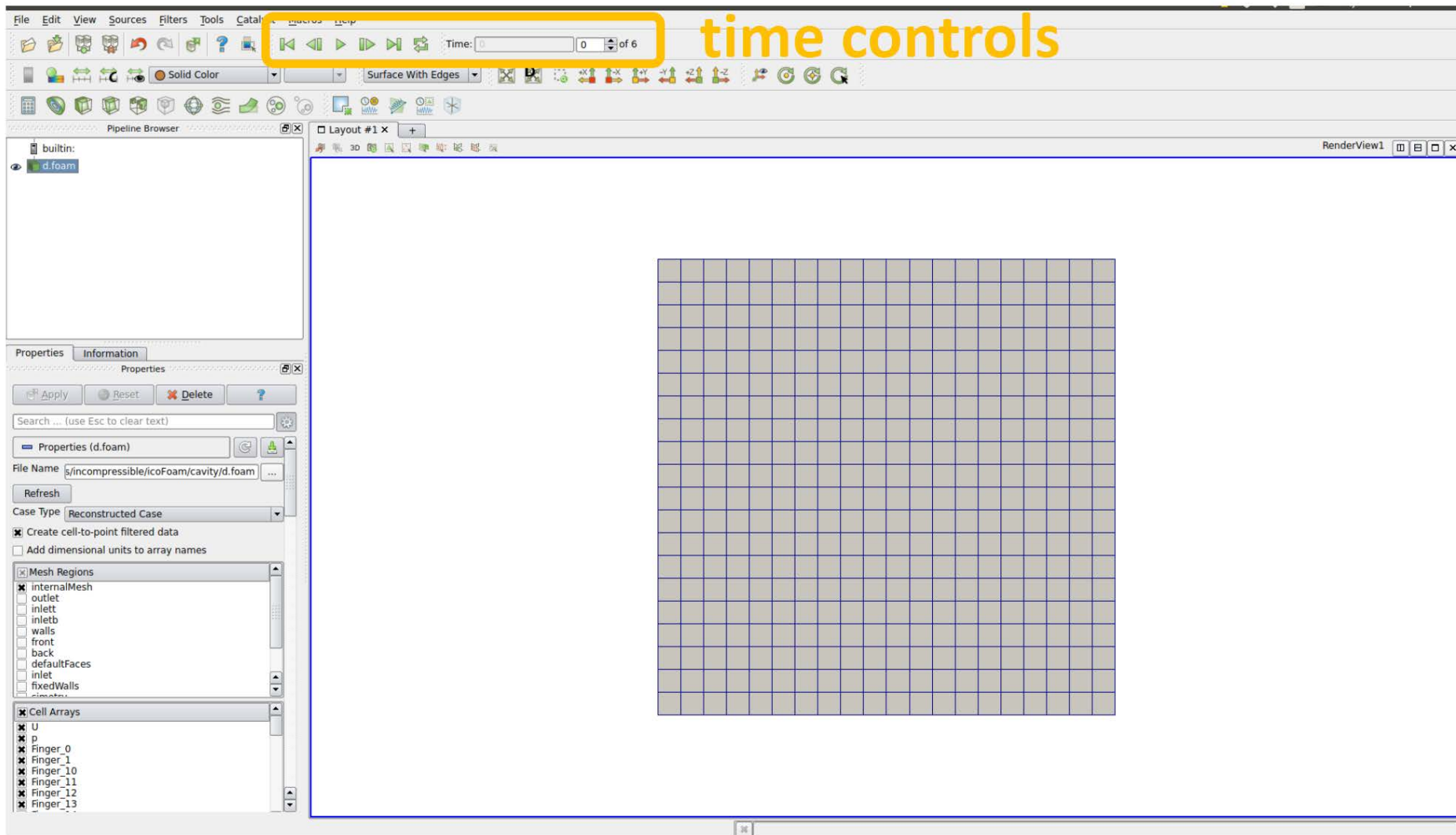
View results

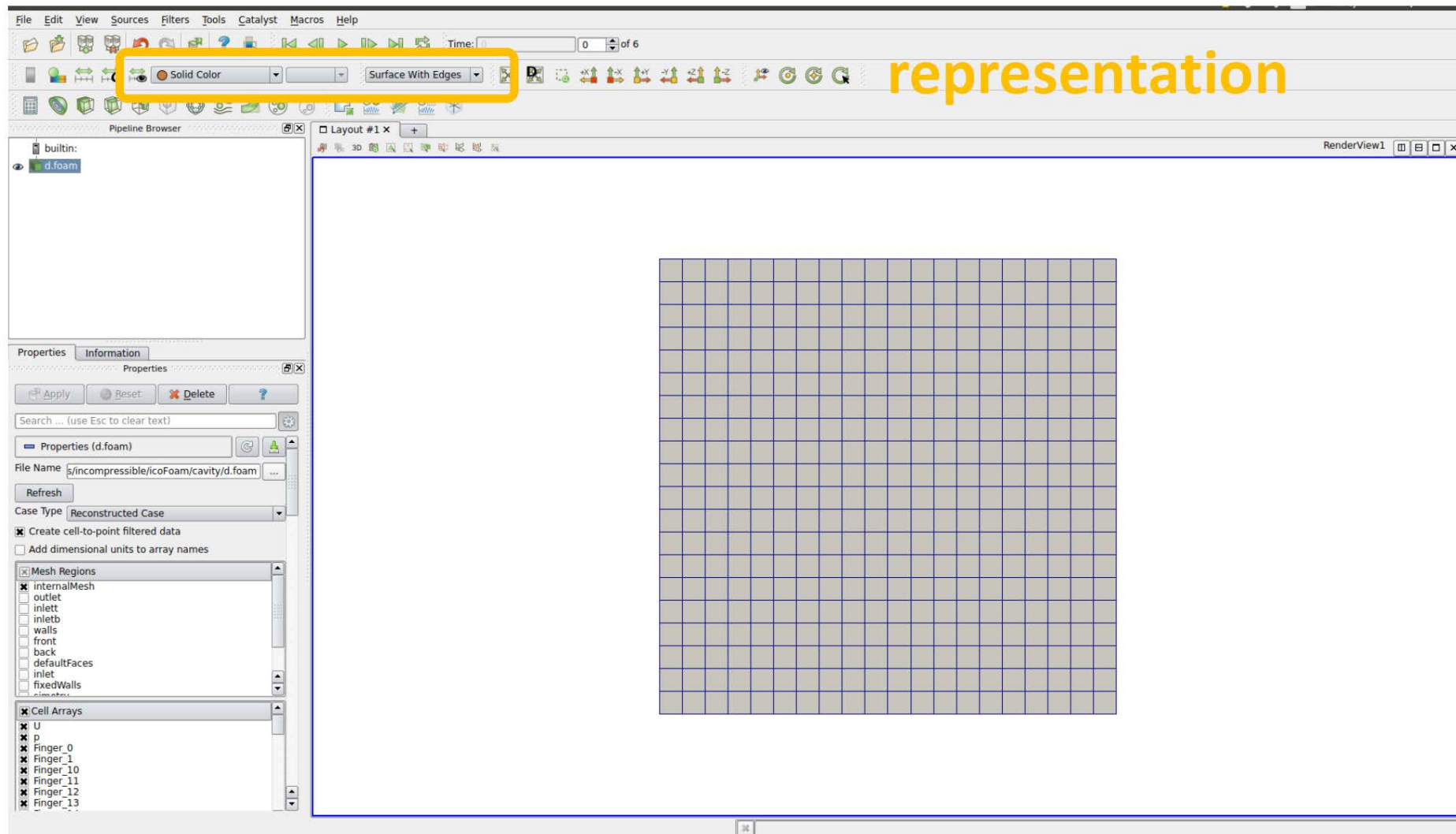
>> touch cavity.foam

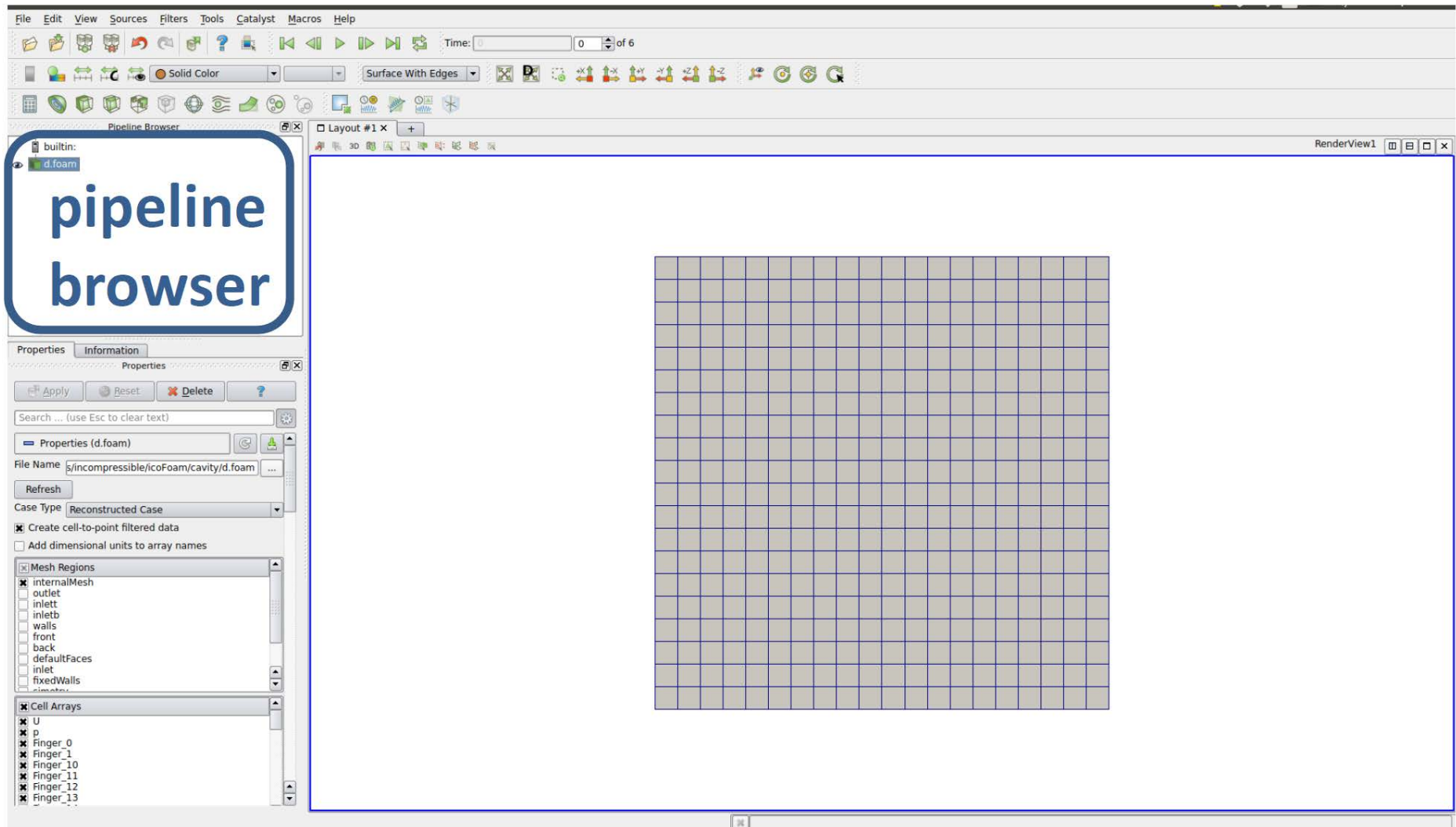
>> paraview cavity.foam

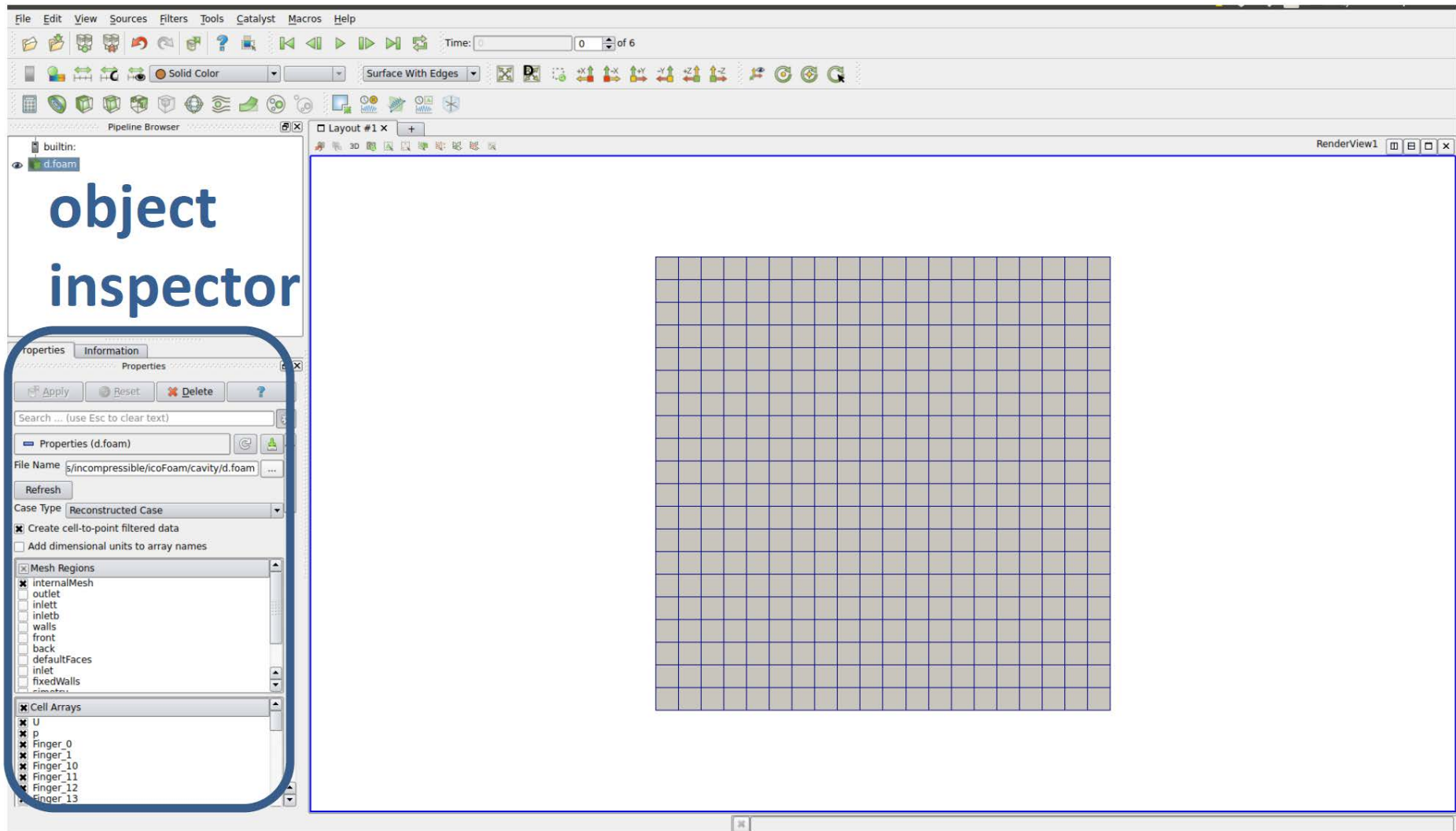


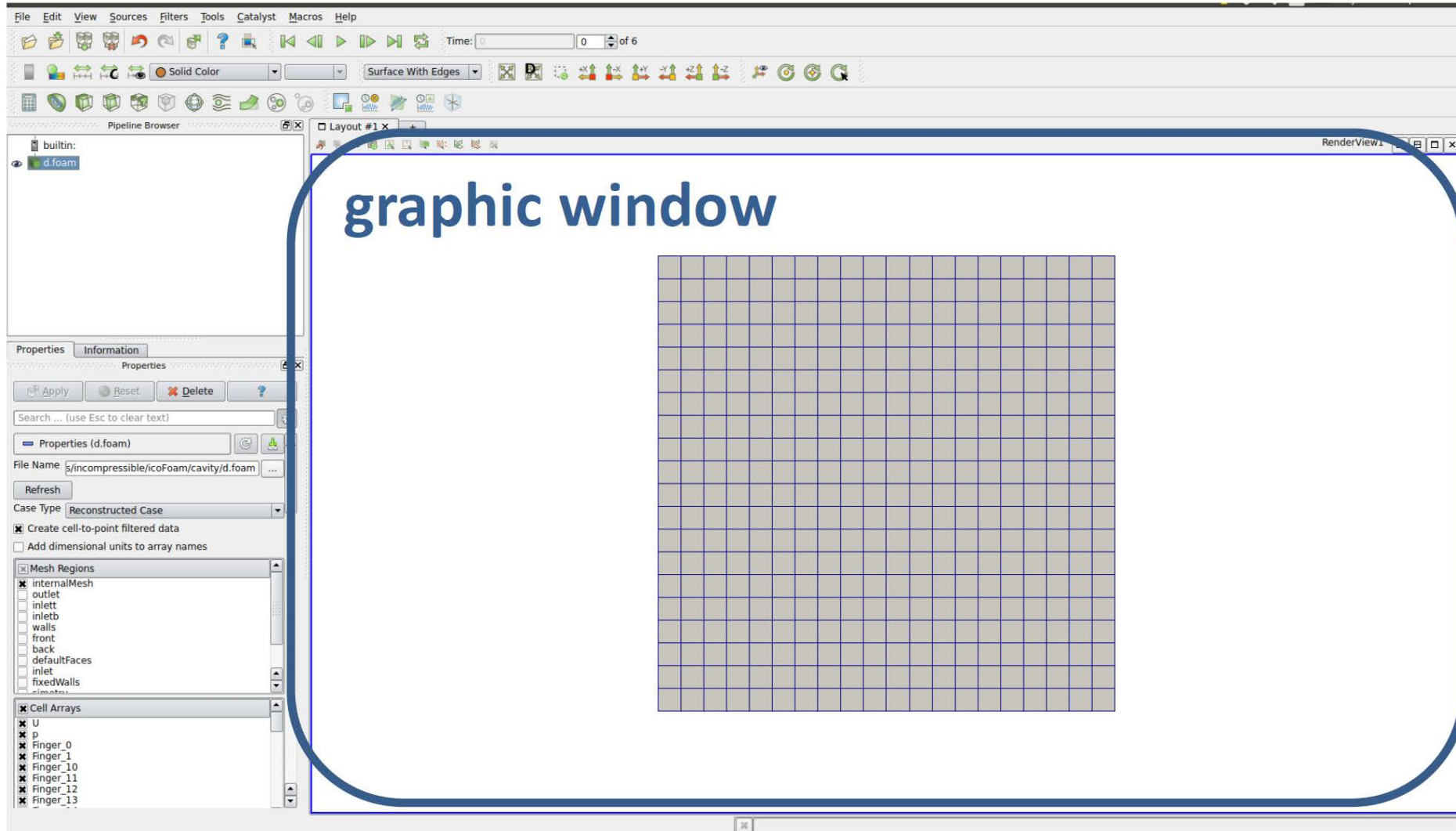




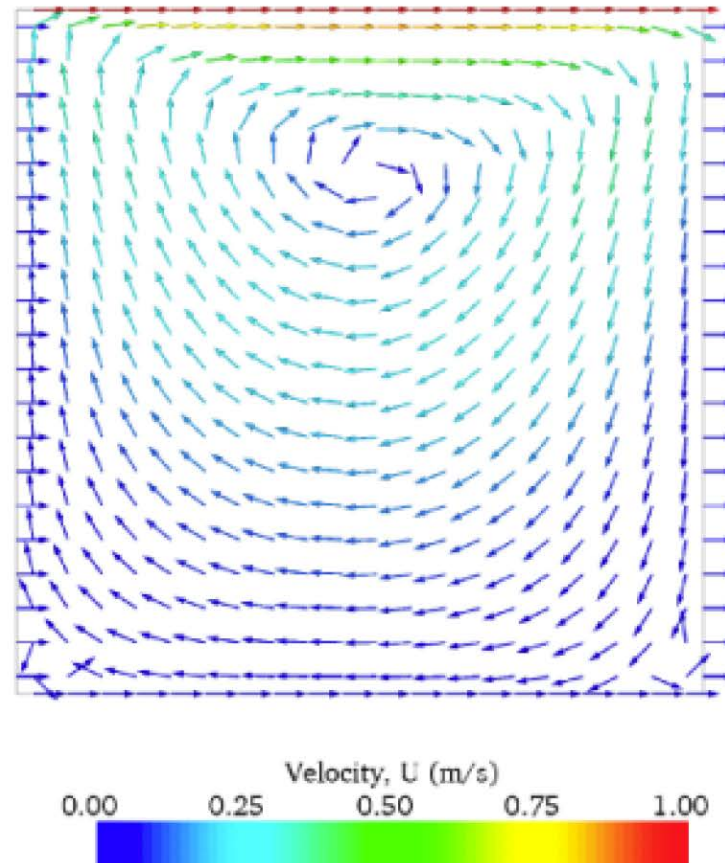




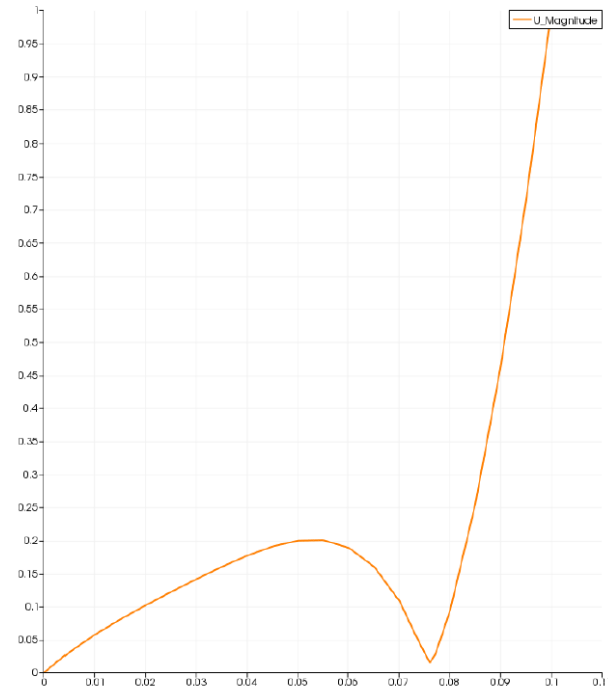
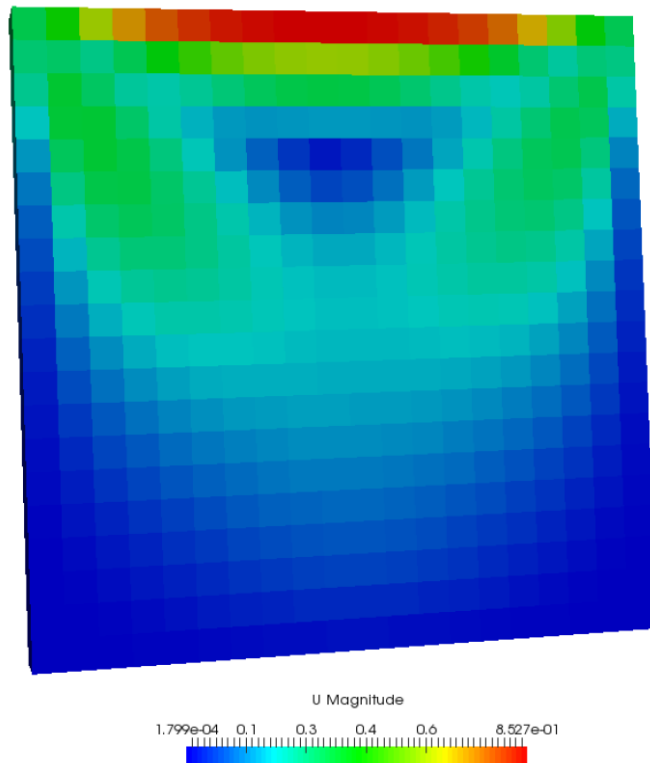




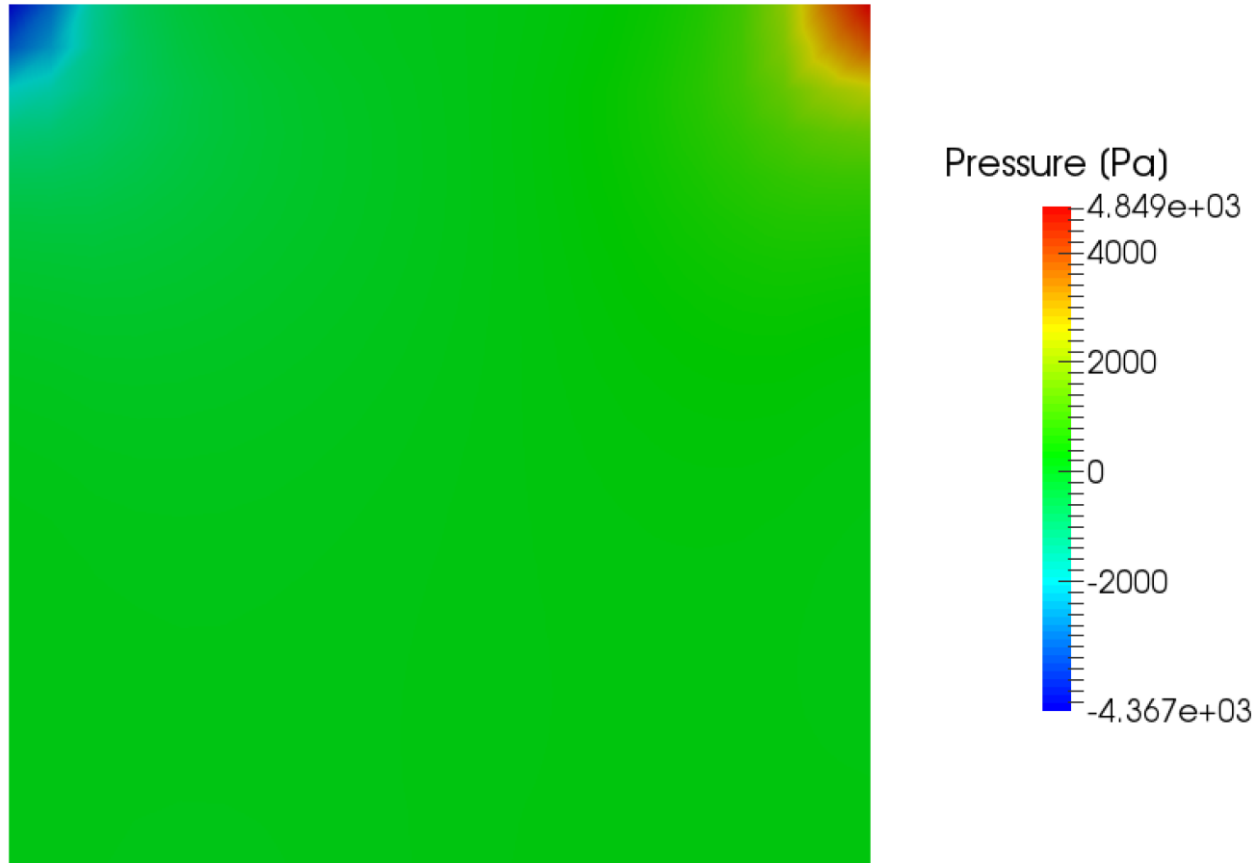
Use **Glyph** option to plot **velocity field**



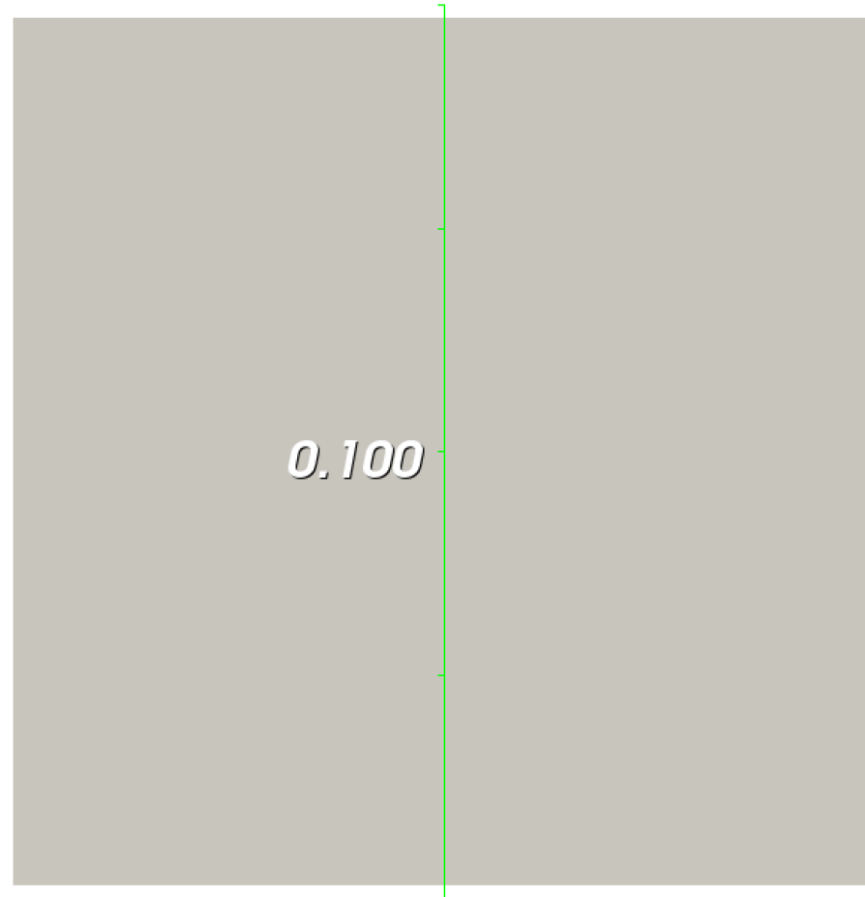
Plot Over Line the **velocity field** in the middle of the channel along y direction, $P1(0,05;0;0,005)$ and $P2(0,05;0,1;0,005)$.



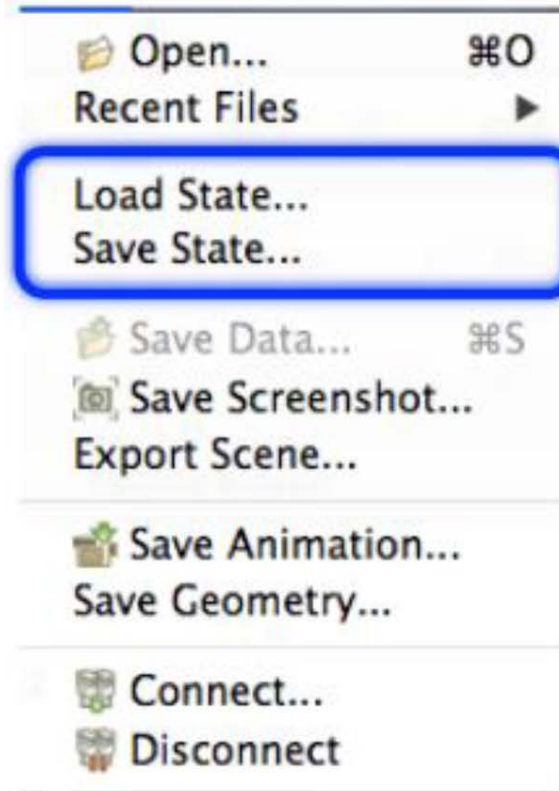
Use **Calculator** to plot **real pressure** (assume density equal 1000 kg/m^3)



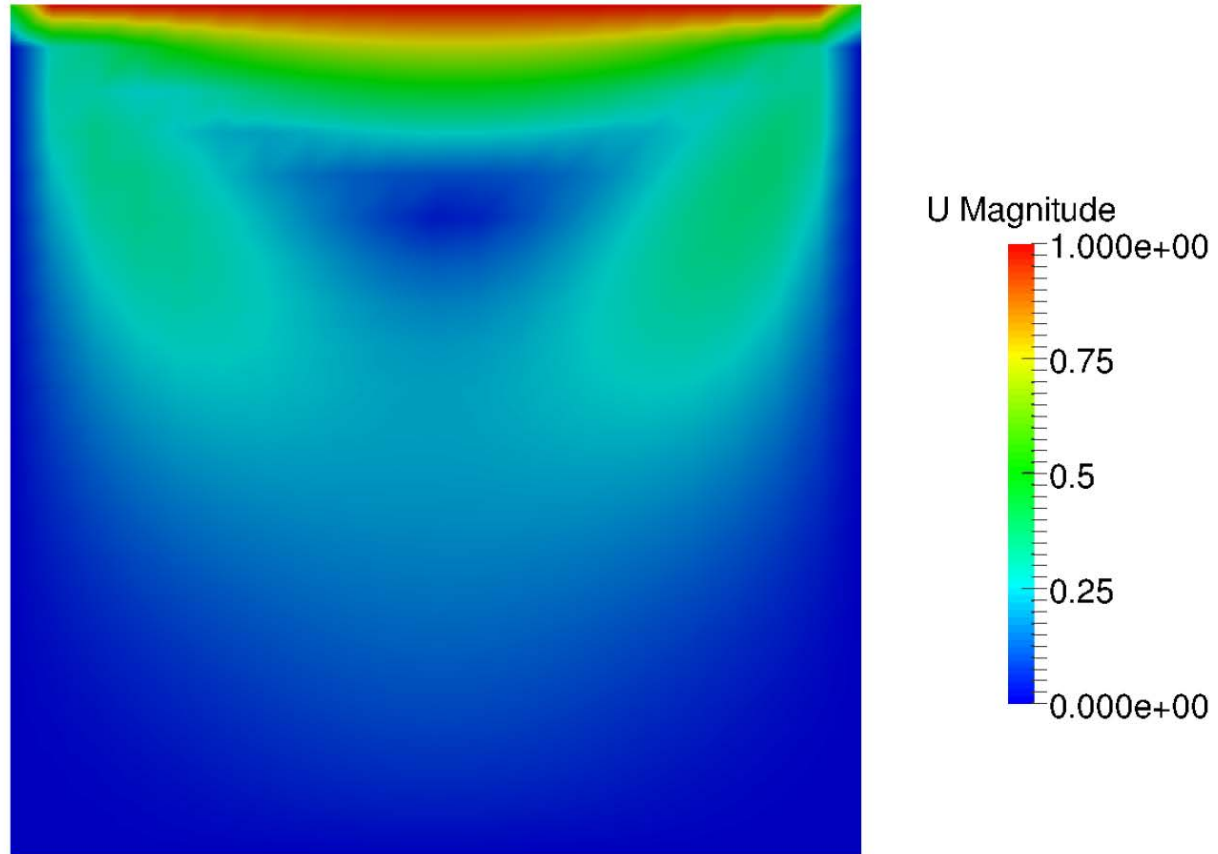
Sources



Load/Save States



Export Scenes



Tutorial

Goldschmidt

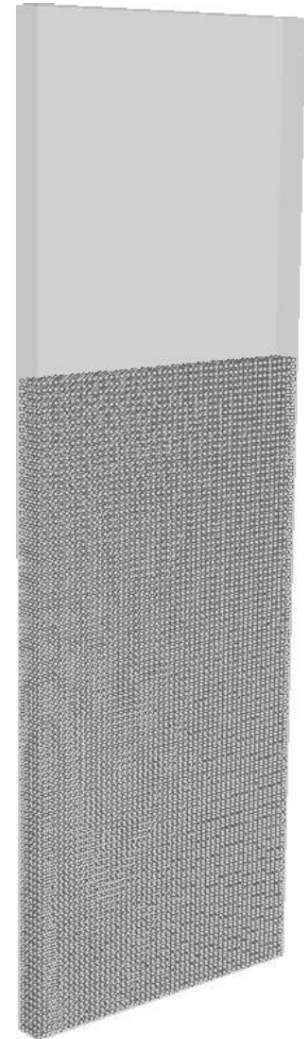
Fluidized bed flow

OpenFOAM includes a transient solver for the coupled transport of a single kinematic particle cloud including the effect of the particulate volume fraction on the continuous phase, suitable for **dense particle flow** simulation.

The solver name is **DPMFoam**.

A **bed of particles** (24750) is initially setup in a rectangular geometry.

For the **gas phase** a prescribed **influx condition** is applied at the bottom, **no-slip boundary conditions** are applied at the **side walls** and a **prescribed pressure condition** is applied at the **top of the bed**.



Decompress files

```
> cd $FOAM_RUN/BII
> tar -xzvf BII-goldschmidt.tar.gz

> cd Goldschmidt
```

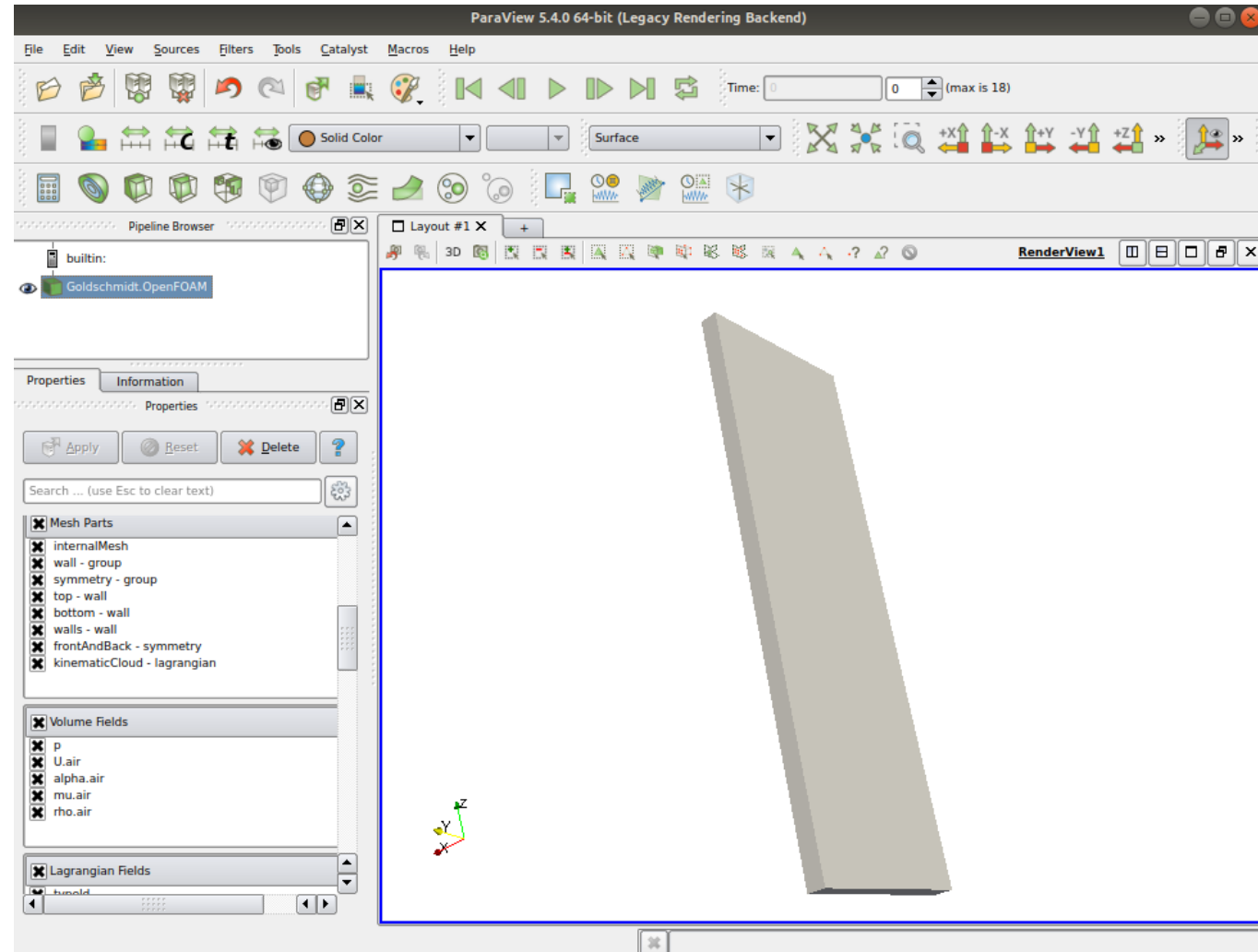
Open Paraview

```
> paraFoam    (Note: Do not use -builtin
               option which loads .foam
               extension and cannot read
               the particle baricentric
               coordinates)
```

1. Select:

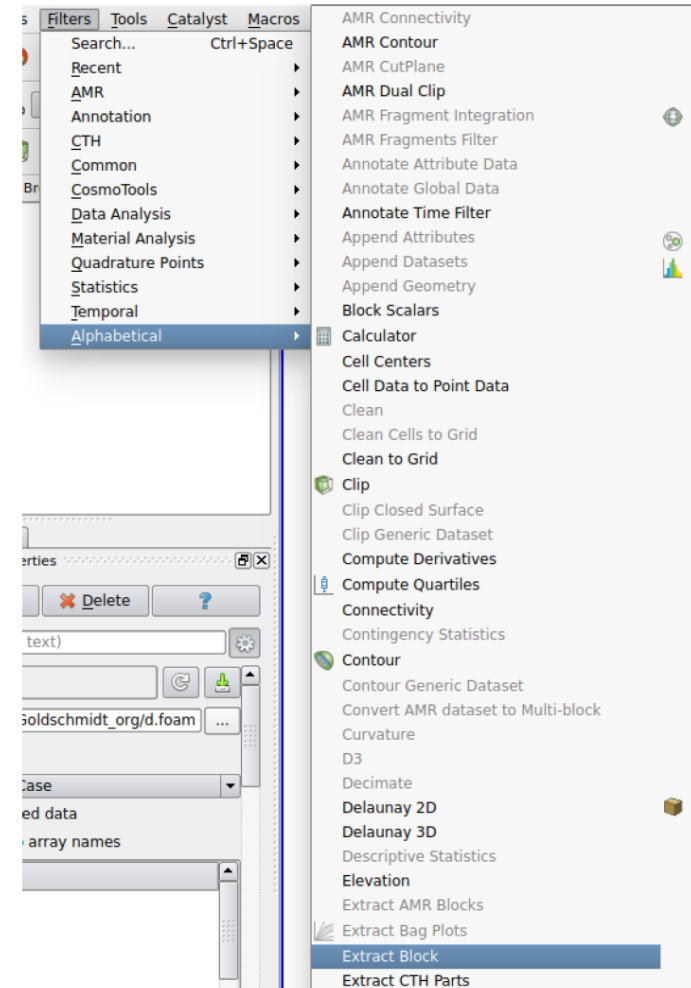
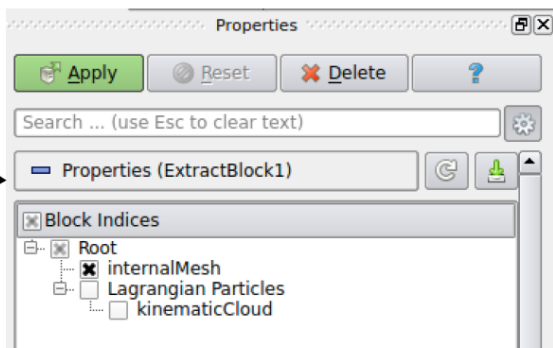
- Mesh Parts
- Volume Fields
- Lagrangian Fields

2. Click Apply



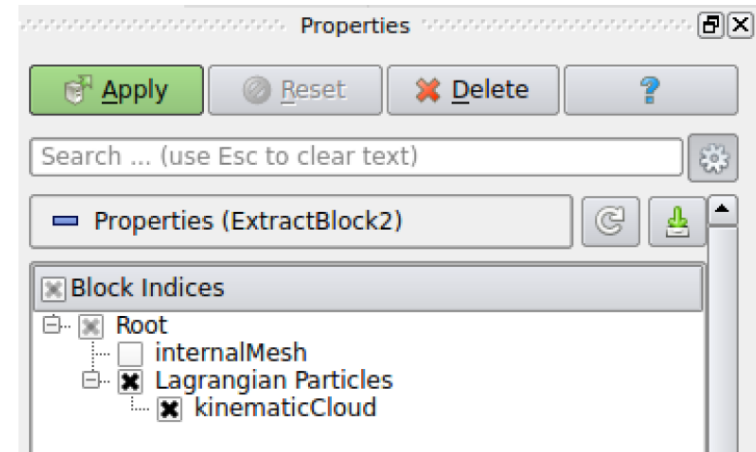
Next, in the **Filter** menu choose the **Alphabetical** submenu and click on the **Extract Block** option.

Choose the **internalMesh** field in the left side **Properties** panel and press **Apply**



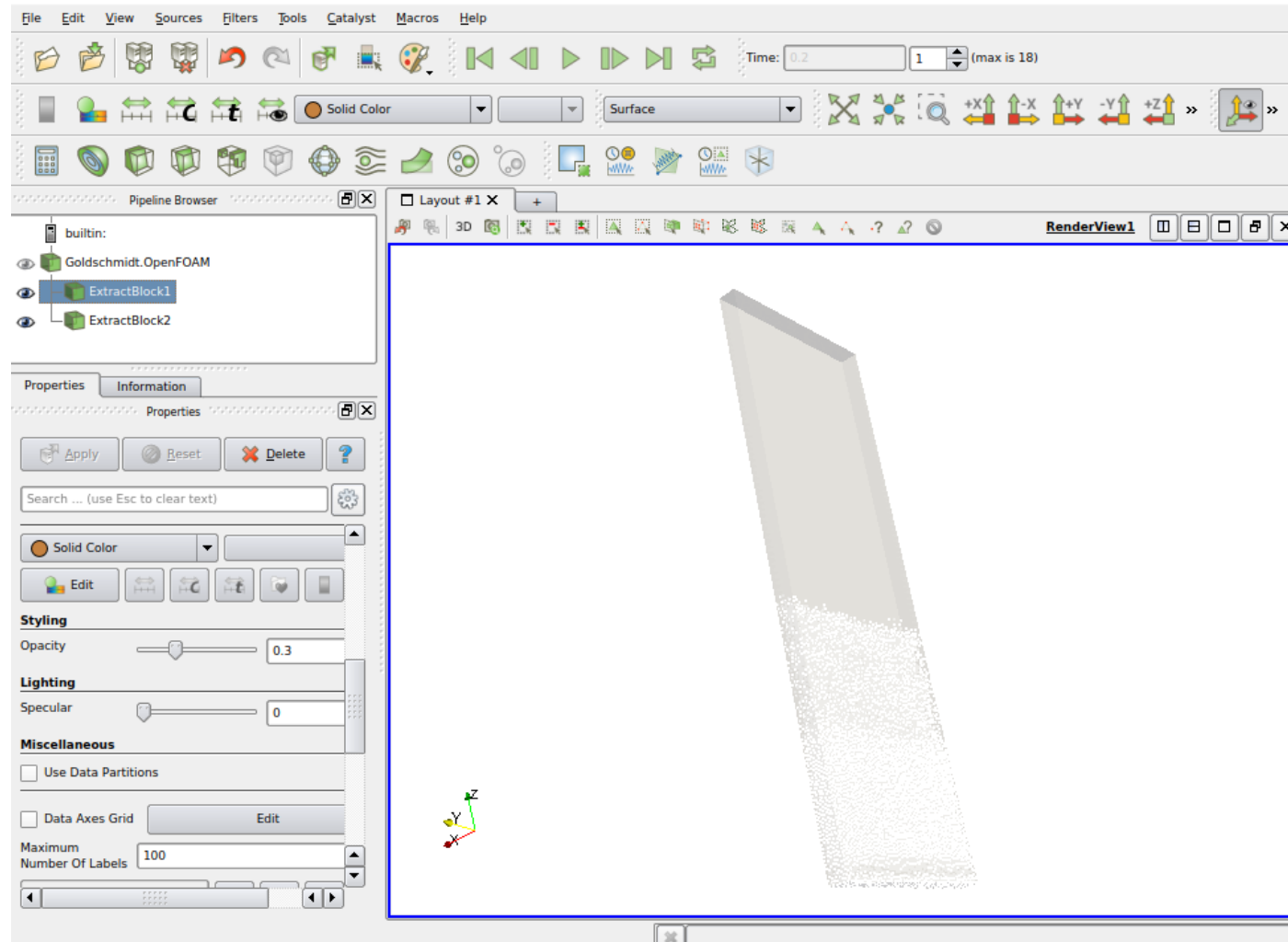
Repeat the same procedure but now choose the **Lagrangian Particles** field.

Now we can see separately the behavior of the continuous (gas) and discrete (particles) fields.



1. Select:
ExtractBlock1

2. Specify
Opacity to
0.3



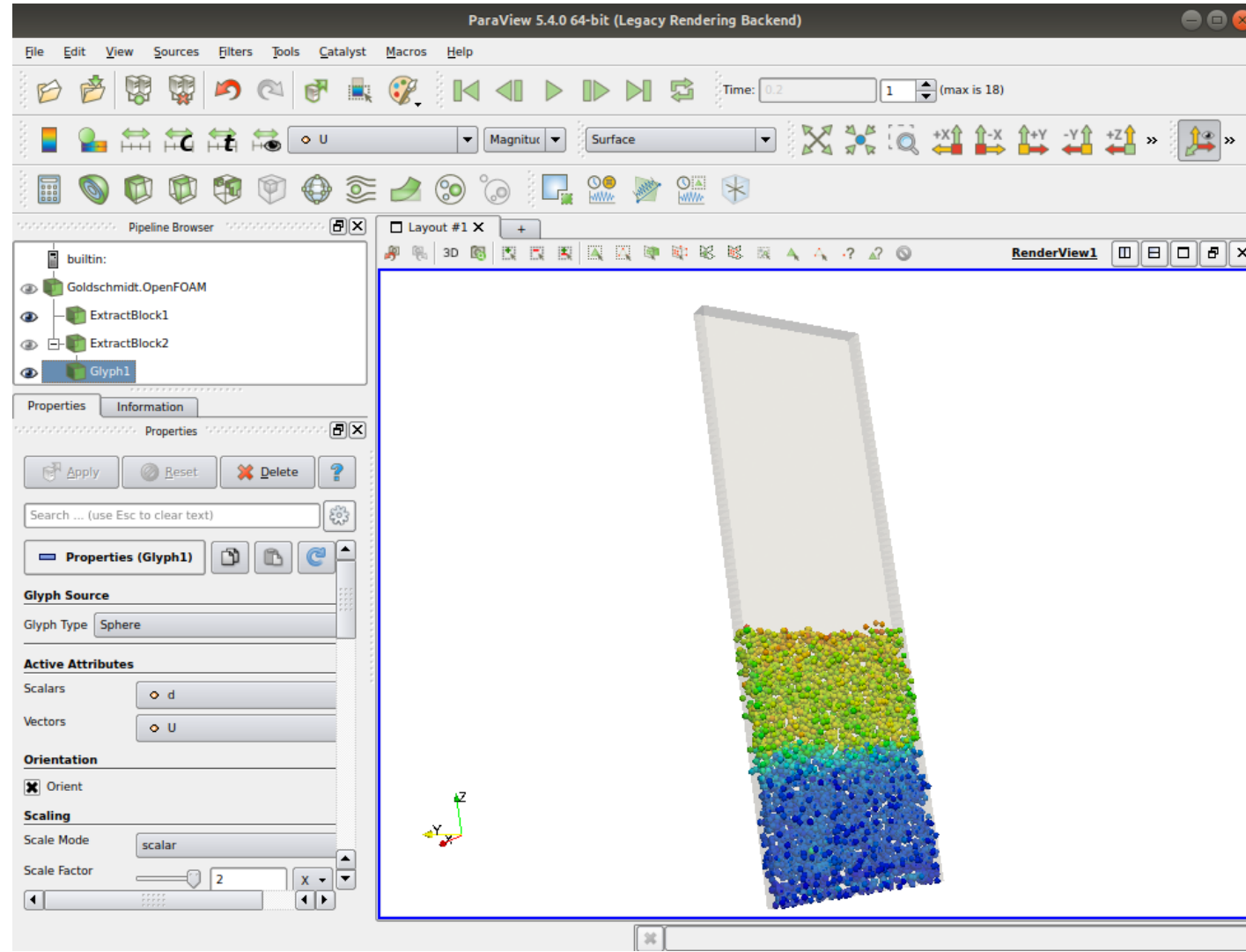
1. Select:
ExtractBlock2

2. Choose Glyph

3. Choose GlyphType
Sphere

4. Choose Scalars d;
Vectors U; Scale
Mode scalar and
Scale Factor 2

5. Click Apply



Tutorial

Propeller

Analysis of Flow around a Ship Propeller

OpenFOAM includes the **Arbitrary Mesh Interface** technique (AMI) for non-conformal patches.

AMI is a technique that allows simulation across **disconnected, but adjacent, mesh domains**. The domains can be stationary or move relative to one another.

The sliding interface capability has been tested on engineering geometries, including a **propeller**.

Flow was simulated using the **pimpleDyMFoam** solver.

Transient solver for incompressible flow of Newtonian fluids on a moving mesh using the PIMPLE (merged PISO-SIMPLE) algorithm.

The **propeller geometry** is represented in the figure.

**Inlet fixedValue
Velocity – 5 m/s**

**Inlet zeroGradient
Pressure**

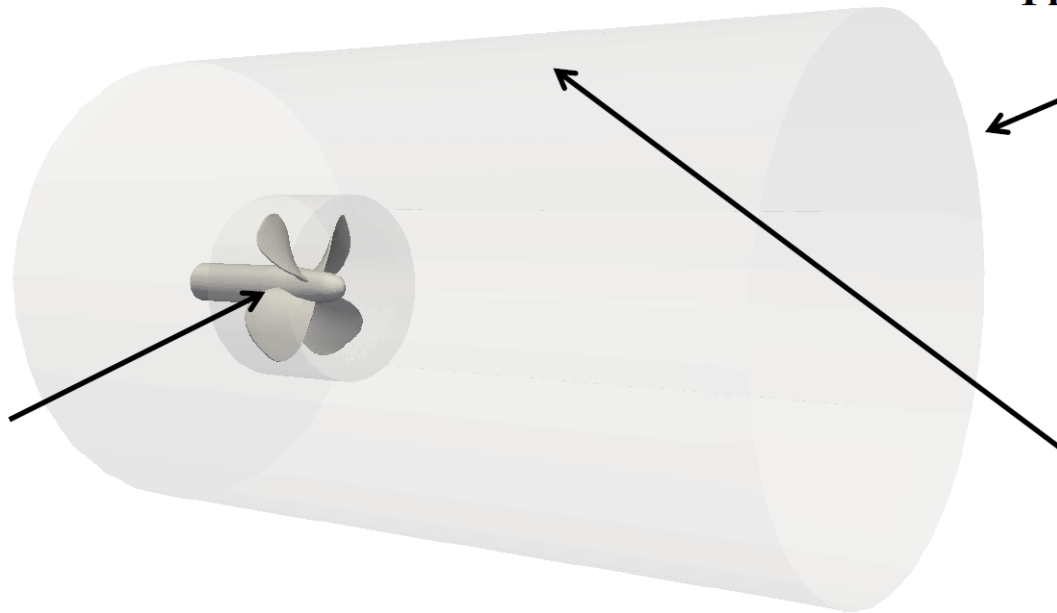
**Outlet inletOutlet
Velocity**

**Outlet fixedValue
Pressure – 0 Pa**

**propeller
movingWallVelocity**

**outerCylinder
fixedValue
Velocity – 0 m/s**

**outerCylinder
zeroGradient
Pressure**



Decompress files

```
> cd $FOAM_RUN/BII  
> tar -xzvf BII-propeller.tar.gz  
  
> cd propeller
```

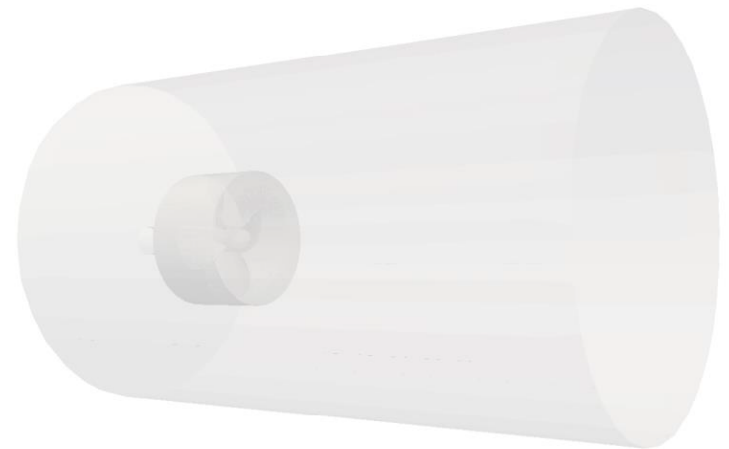
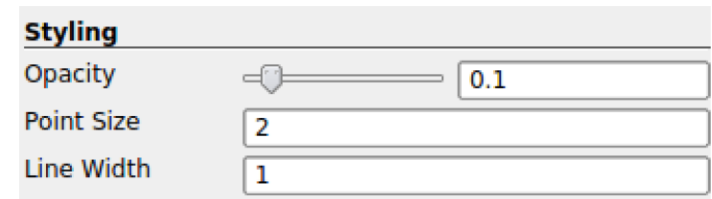
Open Paraview

```
> touch propeller.foam  
> paraview propeller.foam
```

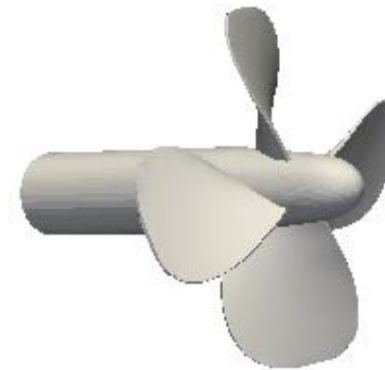
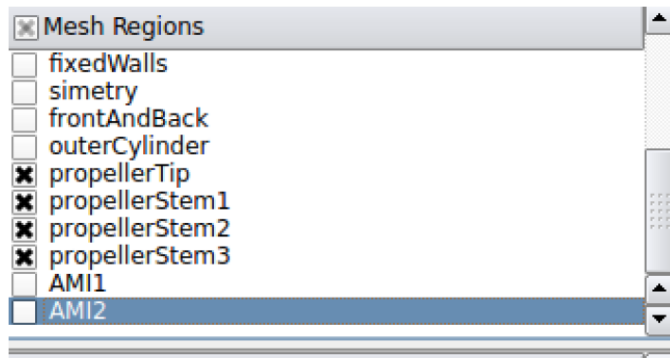
In Paraview try to **animate** the **propeller velocity** field and **export the video with streamlines**.

First, open **only the internalMesh** and press **apply**.

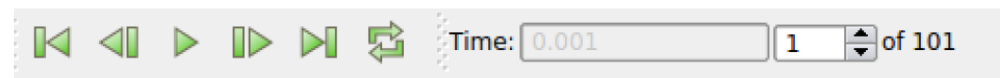
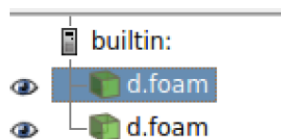
Change the **Opacity** value to 0.1.



Then, **open again the same case** and choose only the **propeller patches**.



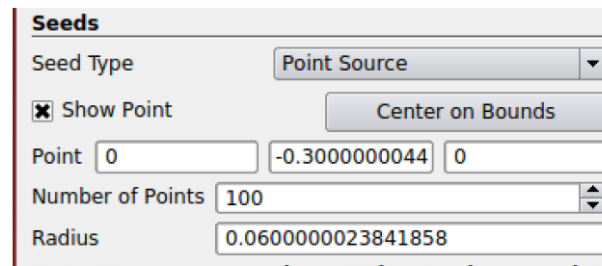
Now, have sure that the **two cases are selected** and **advance one time step**.



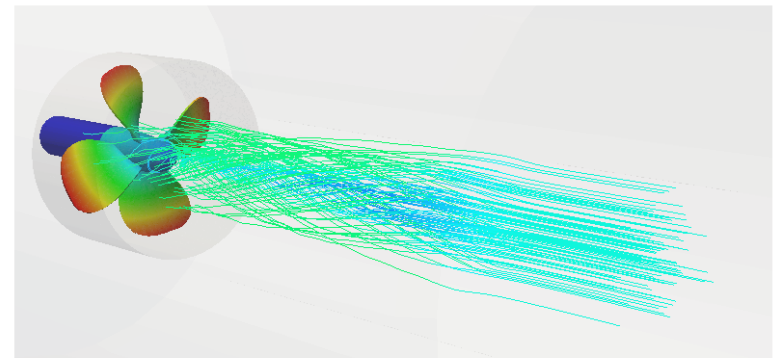
Choose the **streamTracer** option.



Press the **Center on Bounds** button. Finally, press **apply**.

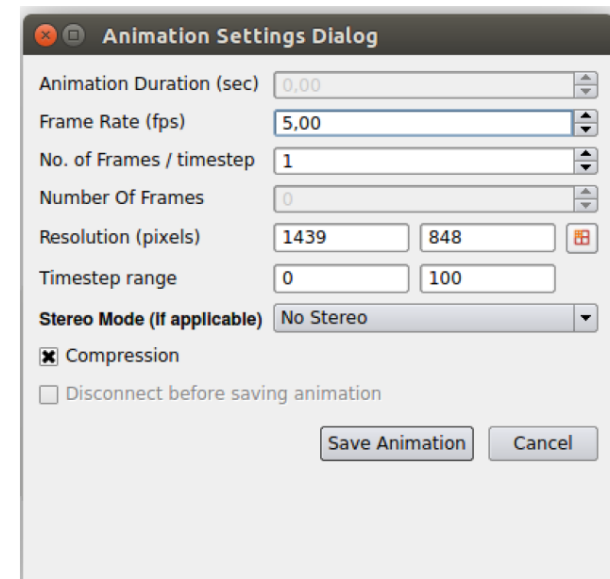
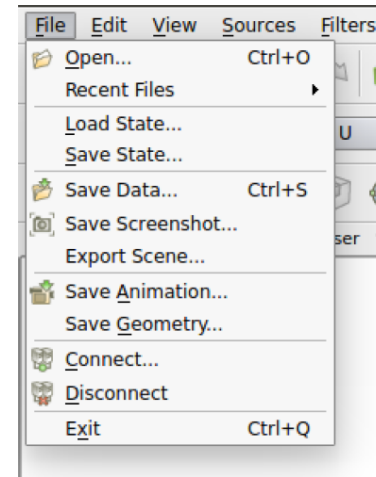


We should obtain an image similar to the one below, coloring by the velocity field.



To finalize we will **export a video** with the movement of the propeller.

For that in the **File** menu click on **Save Animation** and setup **Frame Rate** to 5 and finally click on **Save Animation**.



Tutorial

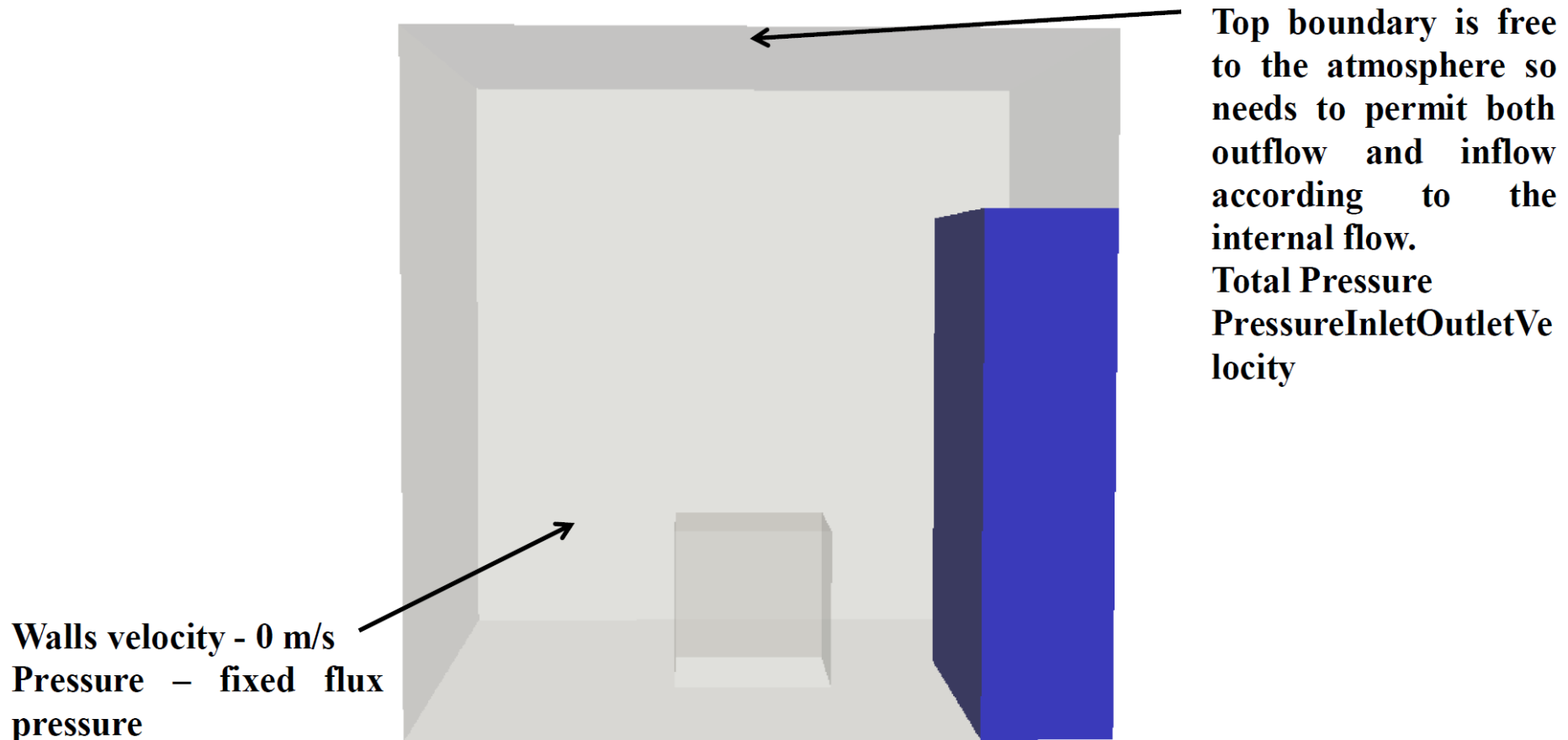
DamBreak3D

Analysis of a falling fluid column around an obstacle

Flow was simulated using the **interDyMFoam** solver.

Solver for **2 incompressible, isothermal immiscible fluids** using a **VOF (volume of fluid) phase-fraction** based interface capturing approach, with optional mesh motion and mesh topology changes including adaptive re-meshing.

The **damBreak** geometry is represented in the figure.



Decompress files

```
> cd $FOAM_RUN/BII
> tar -xzvf BII-damBreak3D.tar.gz

> cd damBreakWithObstacle
```

Open Paraview

```
> touch damBreakWithObstacle.foam
> paraview damBreakWithObstacle.foam
```

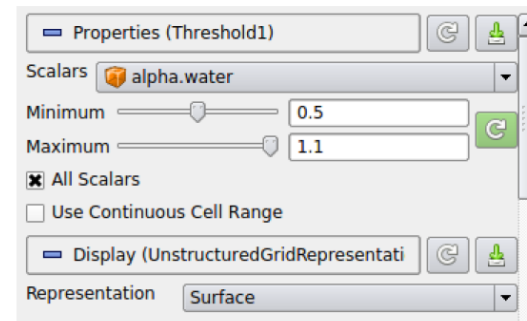
In Paraview try to **animate** the water fall and **export the respective video**.

First, select the internal mesh and atmosphere patch and choose **Opacity to 0.3**.

Choose the **Threshold** option.

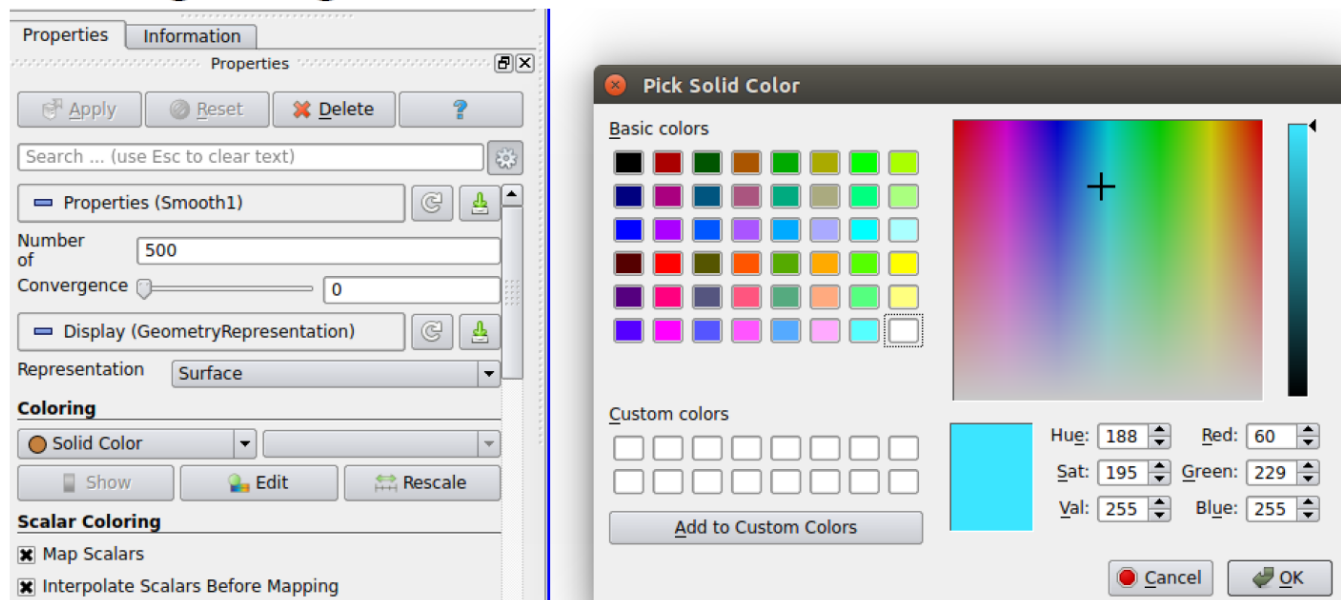


And apply the following values.



Next, apply the **Extract Surface** filter and the **Smooth** filter with **Number of Convergence** of 500.

Finally, color the surface with **Solid Color** and edit the color choosing a light blue color.



We should obtain



To finalize we will **export a video** with the movement of the water.

Tutorial

motorBike

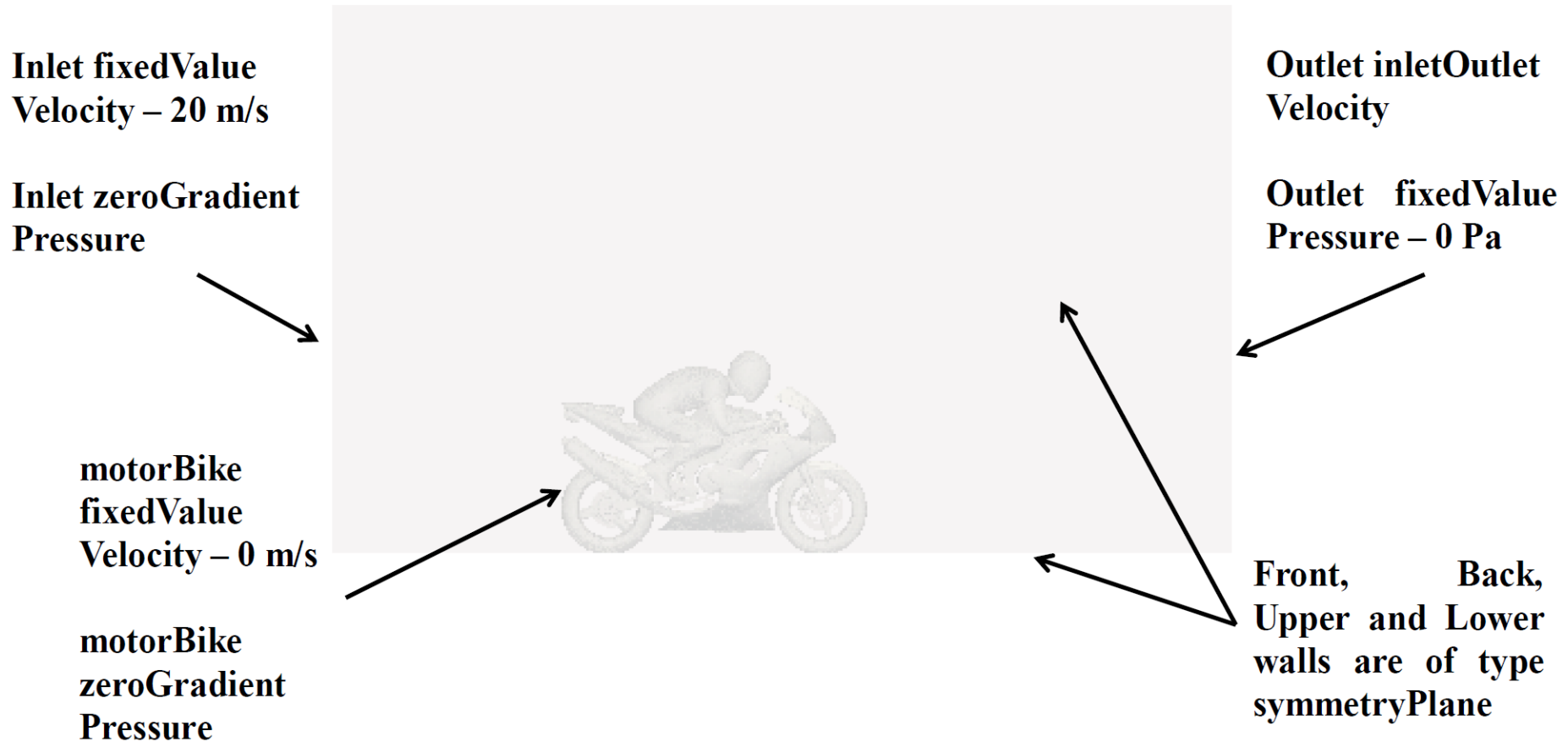
Flow around a Motor Bike

For this tutorial we will look at the simulation of the flow around a **motorbike** model.

Flow was simulated using the **pisoFoam** solver.

Transient solver for incompressible flow. Turbulence modeling is generic, i.e. laminar, RAS or LES.

The **motorBike** geometry is represented in the figure.



Decompress files

```
> cd $FOAM_RUN/BII  
> tar -xzf BII-motorBike.tar.gz  
  
> cd motorBike
```

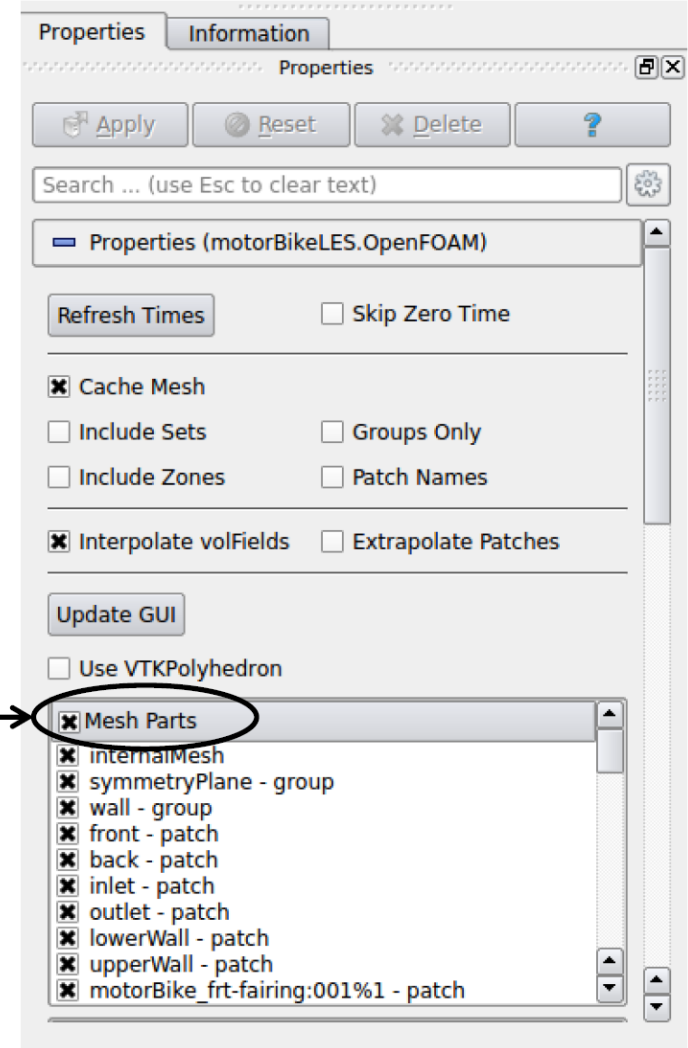
Open Paraview

```
> touch motorBike.foam  
> paraview motorBike.foam
```

In Paraview try to plot the **streamlines** that were exported from the computations and are saved on the **postProcessing** folder.

First in the Paraview **Properties** **painel** select all **Mesh parts**.

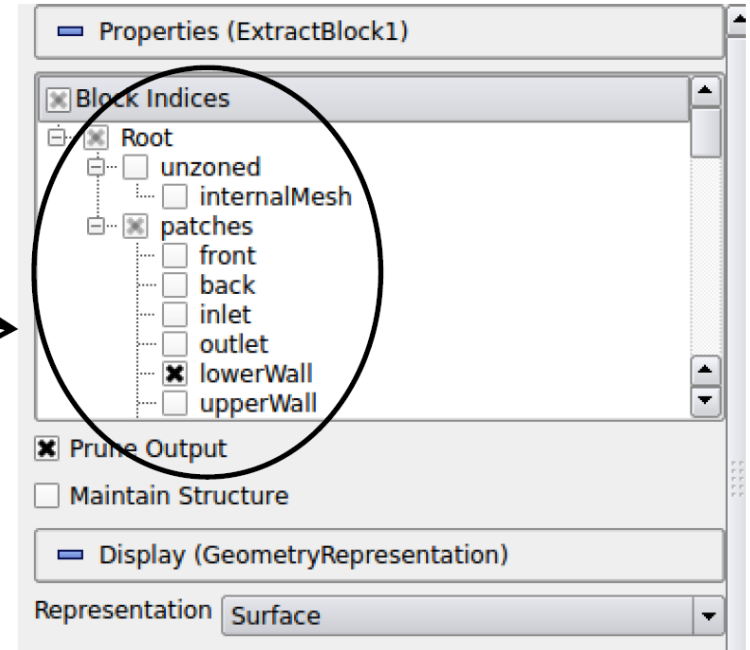
Press **Apply** to obtain all the mesh and patches.



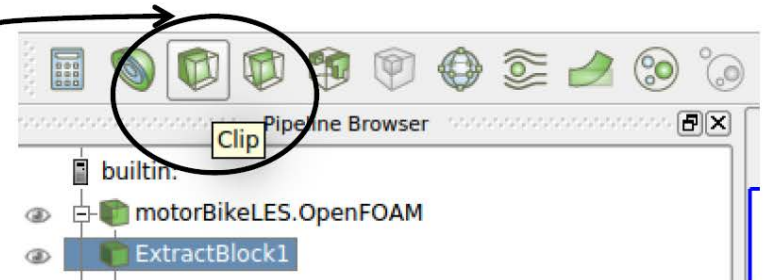
Next, as in the Goldschmidt tutorial select the **Extract block** filter.

Unselect the internalMesh and the front, back, inlet, outlet and upperWall patches.

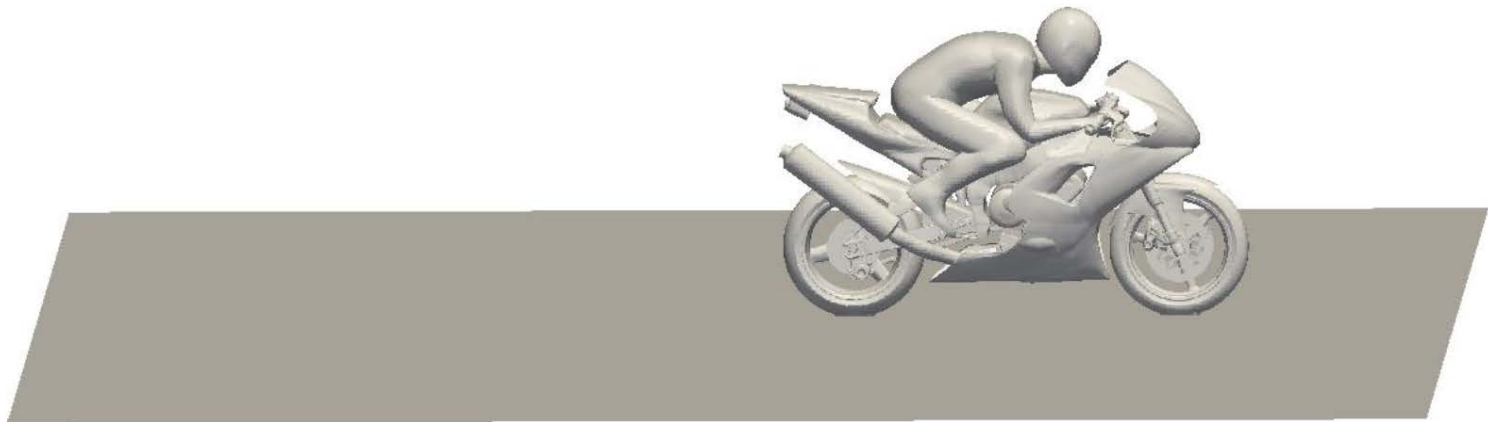
Press **Apply**.



Then, apply several **Clips** to reduce the geometry.

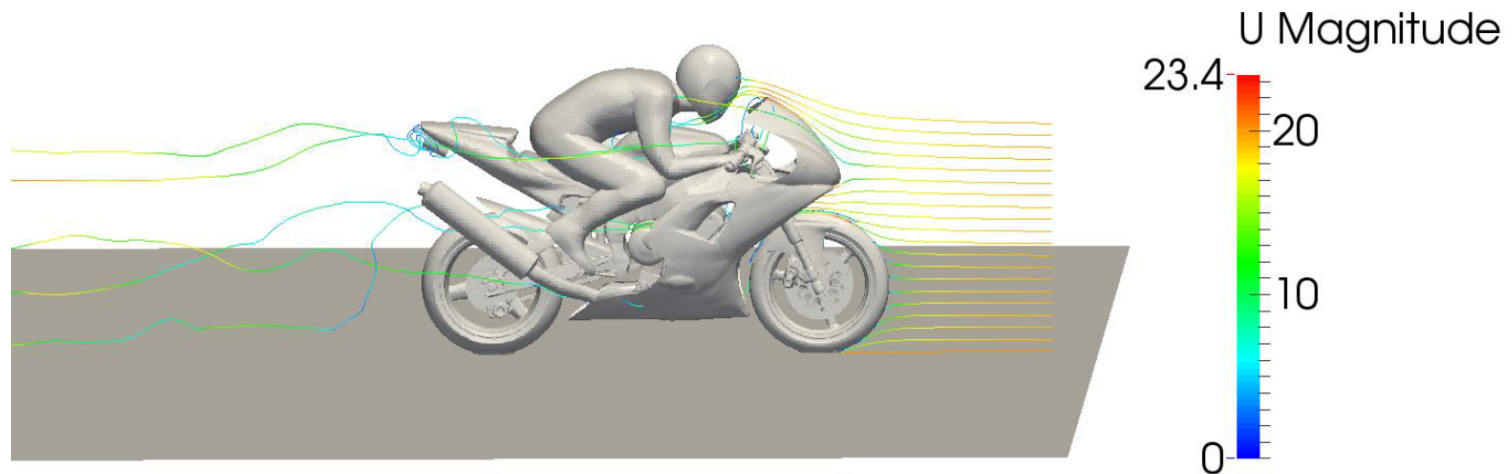


We should obtain one image similar to the one below.



Finally, open the streamline file for the last simulation time: `postProcessing/sets/streamLines/0.7`

And select **velocity field** to be seen. The result should be



Thank you for your attention!