ECS25
EUROPEAN COATING SYMPOSIUM A CORUÑA

Getting Started with OpenFOAM: Fundamentals and Practical Coating Applications

Session 2: Geometry, Mesh Generation, and Case Setup in OpenFOAM

This presentation was adapted from Wagner Galuppo's and Gabriel Wagner's Foam@Iberia 2023 Beginner Course C2
https://github.com/Computational-Rheology/Foam_Iberia_2023/blob/main/Beginner/C2/

J. Vidal and J. Miguel Nóbrega

jpovidal@gmail.com / mnobrega@dep.uminho.pt

University of Minho
School of Engineering

Computational Rheology @IPC

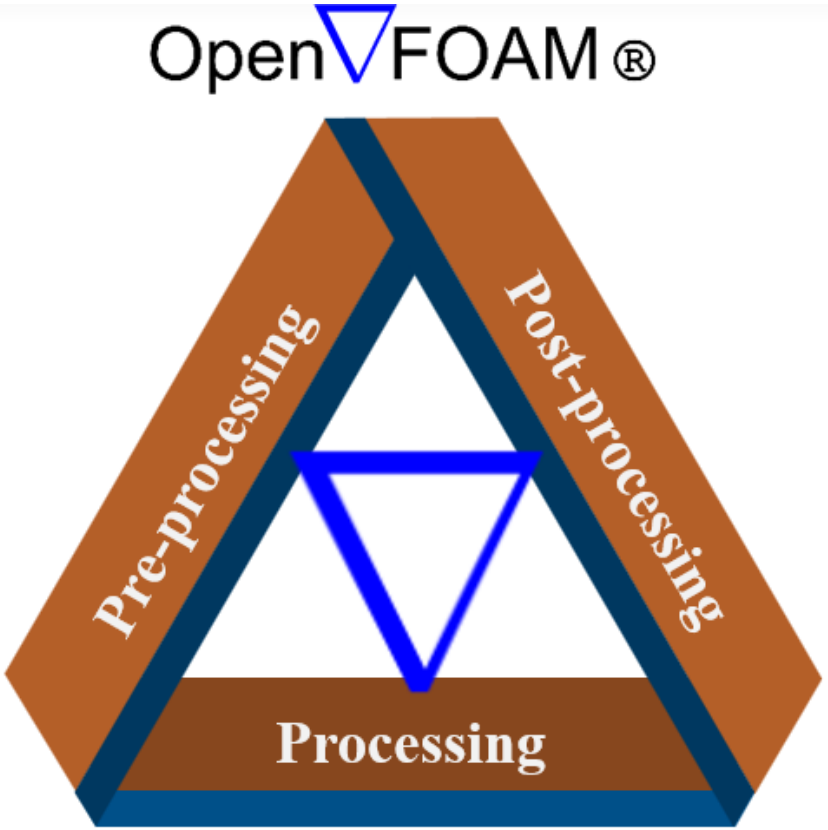
iPC
INSTITUTE FOR POLYMERS AND COMPOSITES

Outline

14:00 – 15:00	**Session 1**: Introduction to OpenFOAM and Simulation Fundamentals
15:00 – 15:30	**Session 2a**: Geometry, Mesh Generation, and Case Setup in OpenFOAM
15:30 – 16:00	Coffee-Break
16:00 – 16:30	**Session 2**: Geometry, Mesh Generation, and Case Setup in OpenFOAM
16:30 – 17:30	**Session 3**: Hands-On Coating Case Study

Introduction

- Geometry Creation
- Meshing Creation
- Meshing Tools
- Material Properties Setup
- Boundaries Definitions
- Convergence Checks
- …



- Data Retrieval and Visualization
- Derived Data Generation
- Trend Analysis
- Improving user's readability of calculated numerical data.
- A wise linkage that leads to enhance productive science for decision support and communication

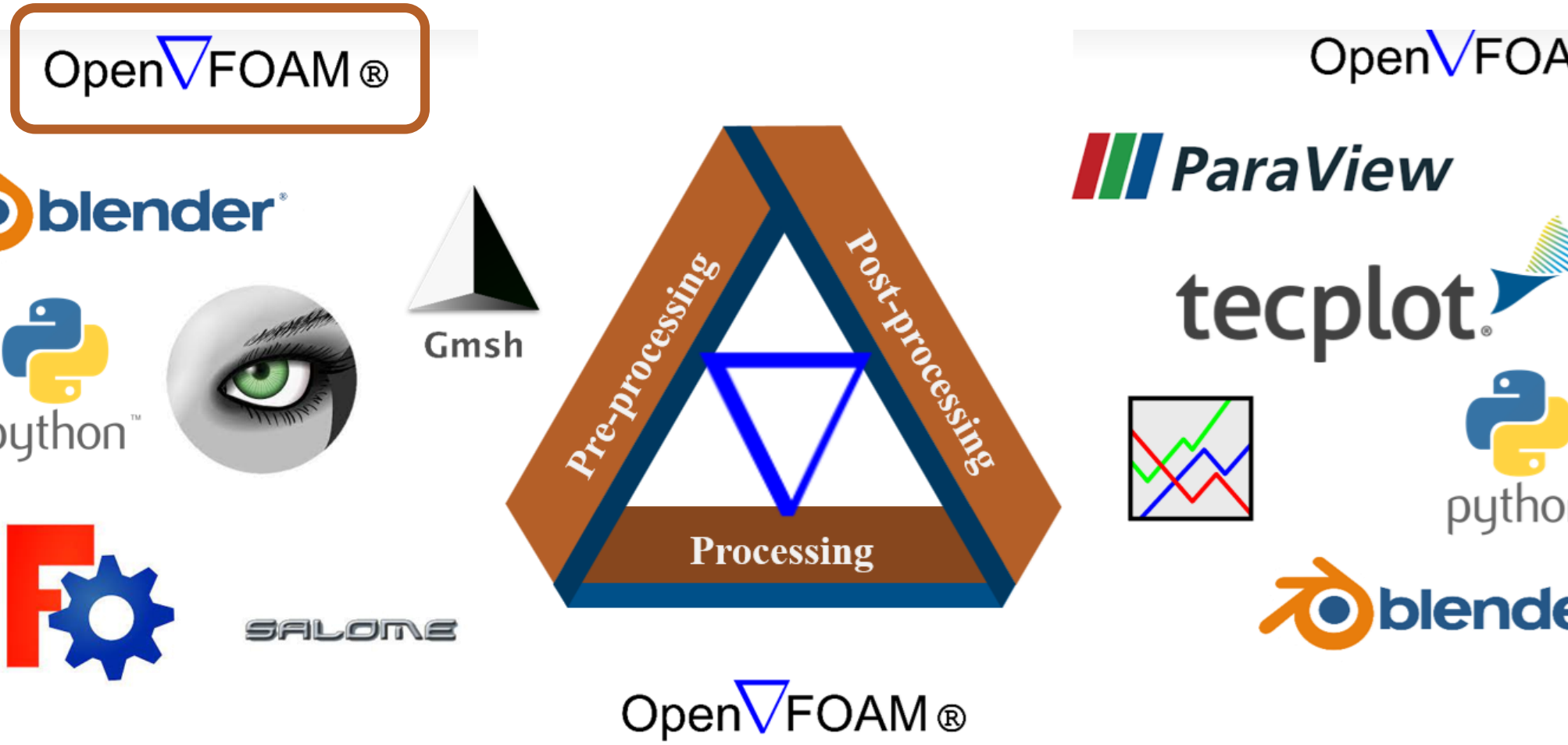
```
(1.0 2.0 3.0)
(4.0 5.0 6.0)
(7.0 8.0 9.0)
(10.0 11.0 12.0)
(13.0 14.0 15.0)
(16.0 17.0 18.0)
(19.0 20.0 21.0)
(22.0 23.0 24.0)
(25.0 26.0 27.0)
(28.0 29.0 30.0)
…
```

$$\frac{\partial}{\partial t}\int_{\Omega}\rho\vec{v}d\Omega = \oint_{\Gamma}\hat{\vec{n}}\cdot\boldsymbol{\sigma}^{T}d\Gamma + \int_{\Omega}\rho\vec{b}d\Omega$$

```
54686973206973206F6E
206578616D706C652066
726F6D205761676E6572
2047616C7570706F206F
662061206865786164 65
63696D616C20636F6465
20666F7220696C6C7573
74726174696F6E207075
72626F736573
…
```

```
(14.2 19.2 23.7)
(3.2 8.5 12.8)
(6.9 12.1 16.4)
(1.4 6.7 11.0)
(17.8 22.8 27.3)
(16.0 21.0 25.5)
(8.7 13.9 18.2)
(19.7 24.6 29.1)
(10.5 15.7 20.0)
(12.4 17.4 21.9)
…
```

Introduction

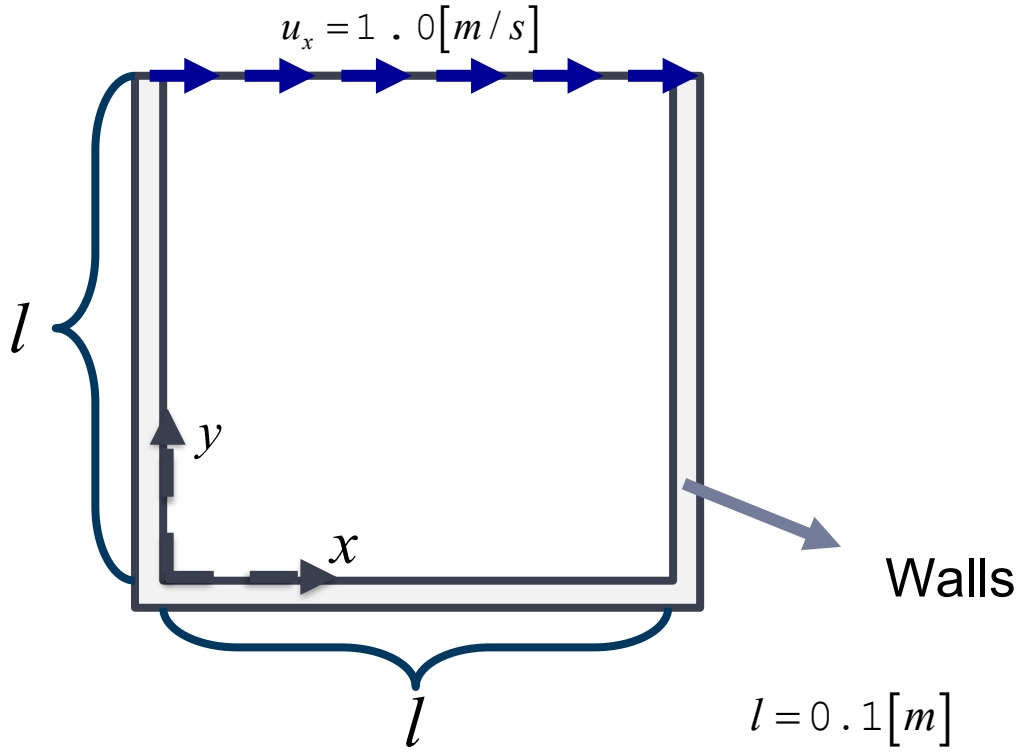


blockMesh – Case 2 (general)

1. Open WSL
2. >> openfoam2506
3. >> run
4. >> cd case2/general
5. >> code . ## Open Vscode
6. Visualize blockMeshDict file ##in the System folder

blockMesh – Case 2 (general)

- Physical Problem: (Cavity tutorial)



blockMesh – Case 2 (general)

BlockMesh Dictionary case2/general/system/blockMeshDict

../C2/tutorials/case01/system/blockMeshDict

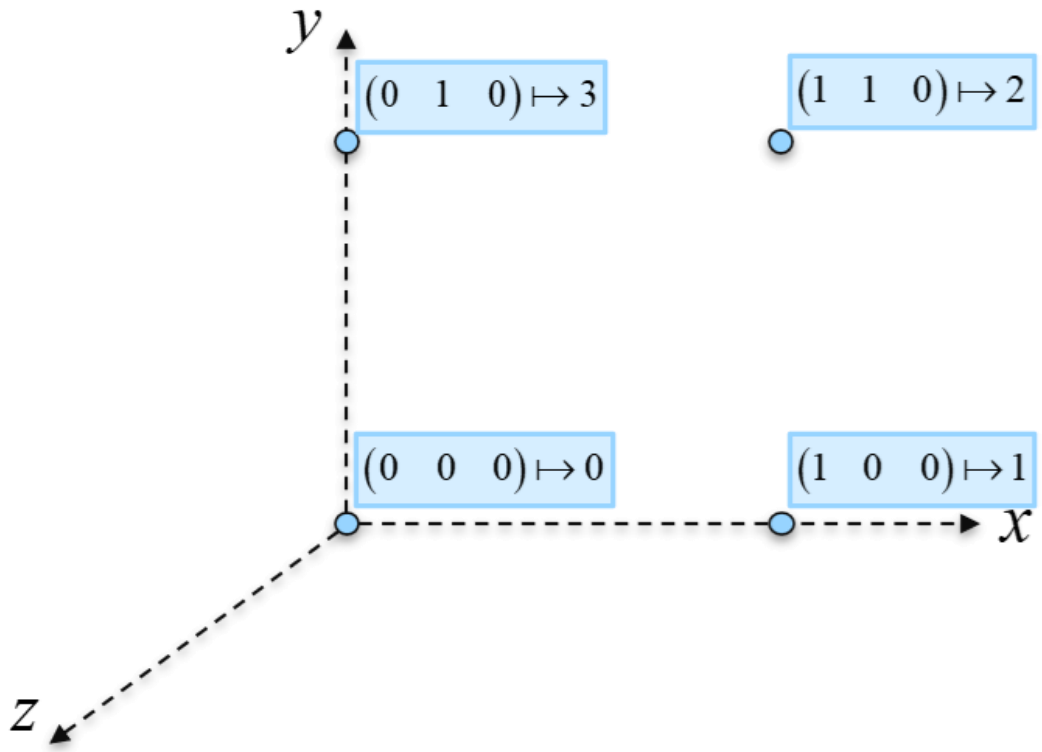
```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v2306                                 |
|   \\  /    A nd           | Website:  www.openfoam.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// convertToMeters 0.1;
scale 0.1;

vertices
(
    (0 0 0)
    (1 0 0)
    (1 1 0)
    (0 1 0)
    (0 0 0.1)
    (1 0 0.1)
    (1 1 0.1)
    (0 1 0.1)
);
```

blockMesh – Case 2 (general)

- Generating Vertices

../case01/system/blockMeshDict

```
17. scale 0.1;
18.
19. vertices
20. (
21.     (0 0 0)     // id: 0
22.     (1 0 0)     // id: 1
23.     (1 1 0)     // id: 2
24.     (0 1 0)     // id: 3
25.     (0 0 0.1)   // id: 4
26.     (1 0 0.1)   // id: 5
27.     (1 1 0.1)   // id: 6
28.     (0 1 0.1)   // id: 7
29. );
```

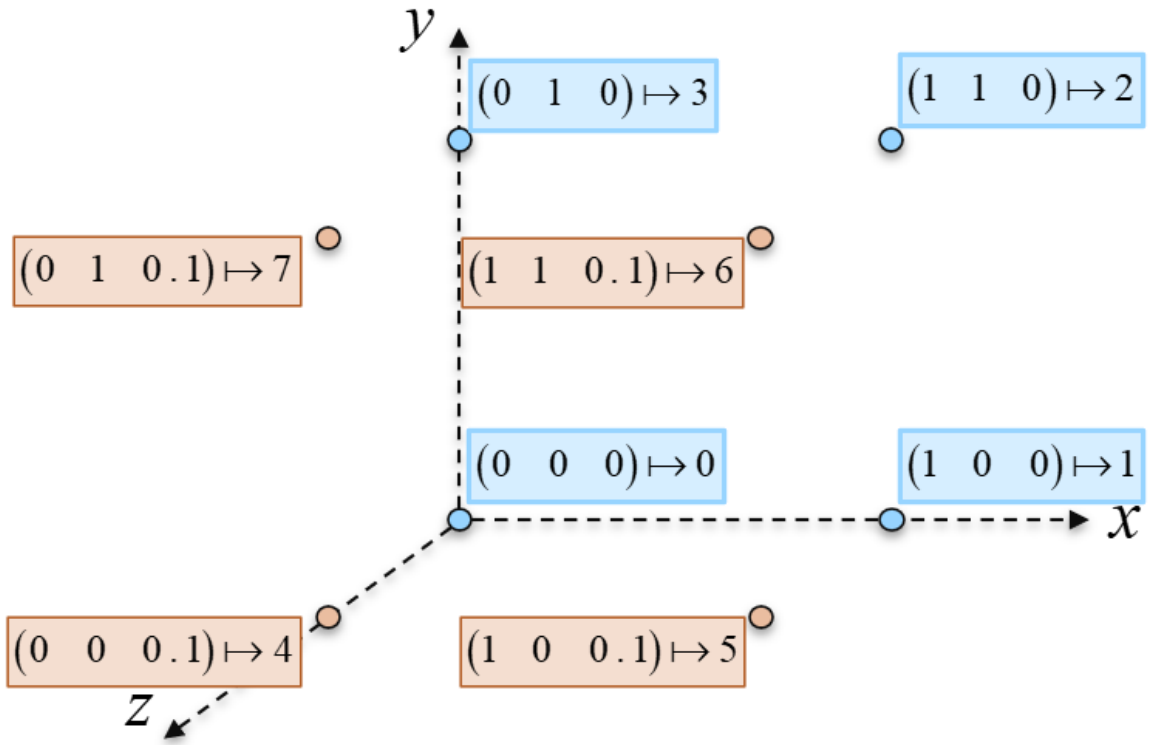


blockMesh – Case 2 (general)

- Generating Vertices

../case01/system/blockMeshDict

```
17. scale 0.1;
18.
19. vertices
20. (
21.     (0 0 0)      // id: 0
22.     (1 0 0)      // id: 1
23.     (1 1 0)      // id: 2
24.     (0 1 0)      // id: 3
25.     (0 0 0.1)    // id: 4
26.     (1 0 0.1)    // id: 5
27.     (1 1 0.1)    // id: 6
28.     (0 1 0.1)    // id: 7
29. );
```

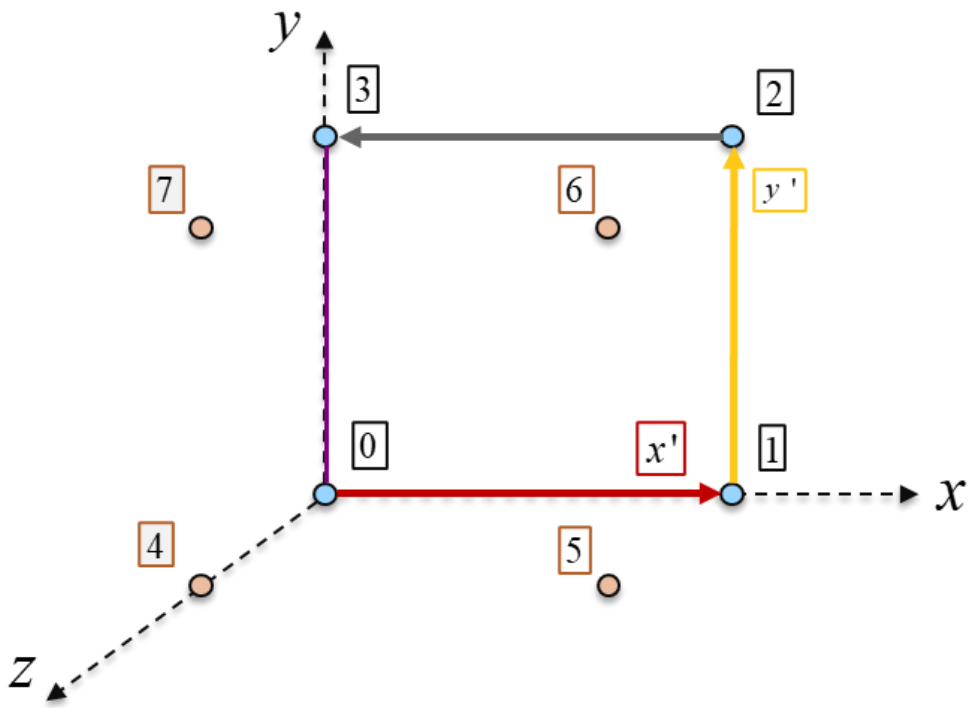


blockMesh – Case 2 (general)

- Generating Vertices

../case01/system/blockMeshDict

```
31.  blocks
32.  (
33.      hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
34.  );
```

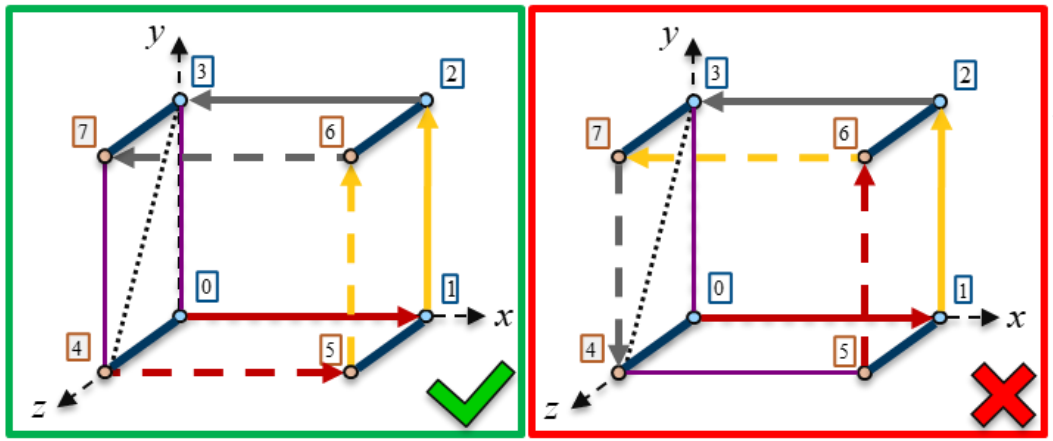


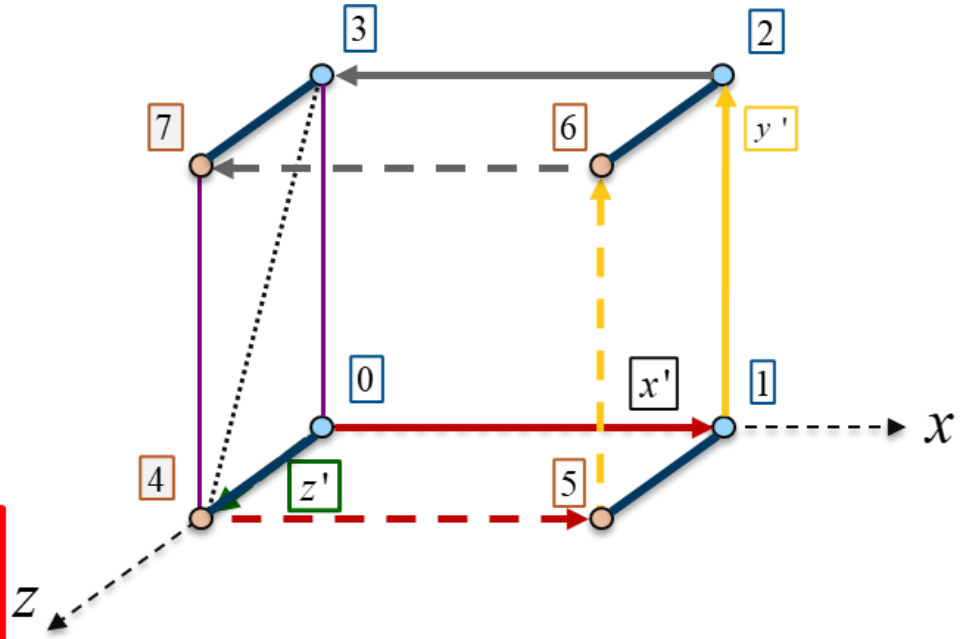
blockMesh – Case 2 (general)

- Generating Vertices

../case01/system/blockMeshDict

```
31.  blocks
32.  (
33.      hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
34.  );
```



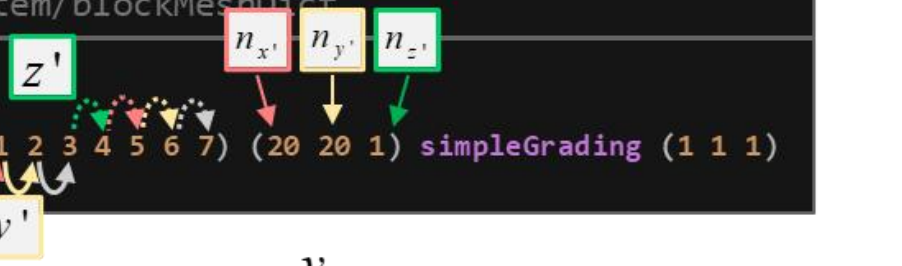


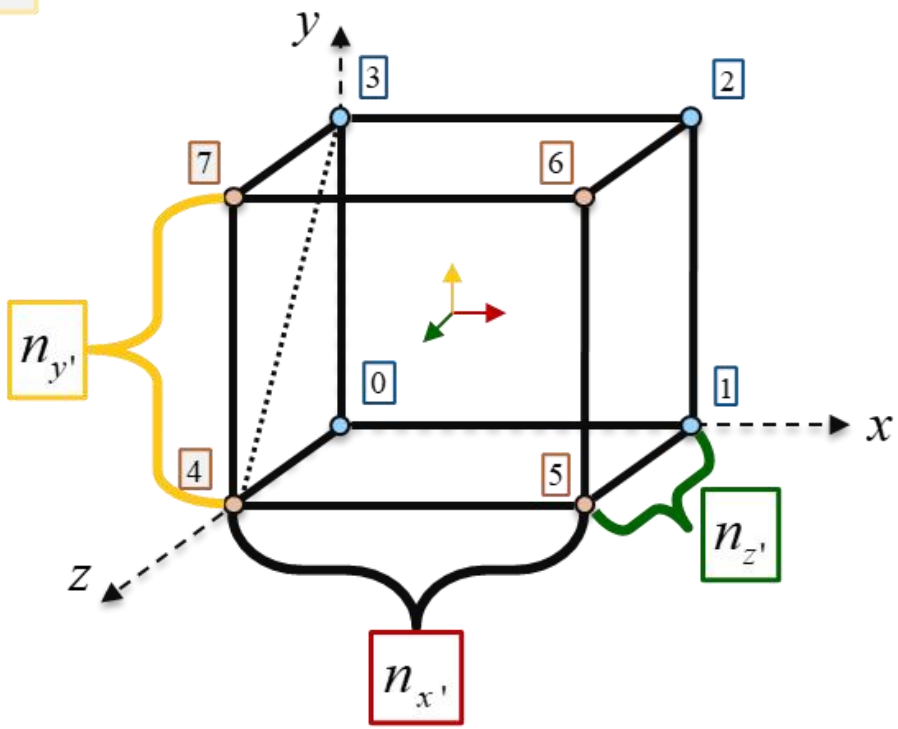
blockMesh – Case 21 (general)

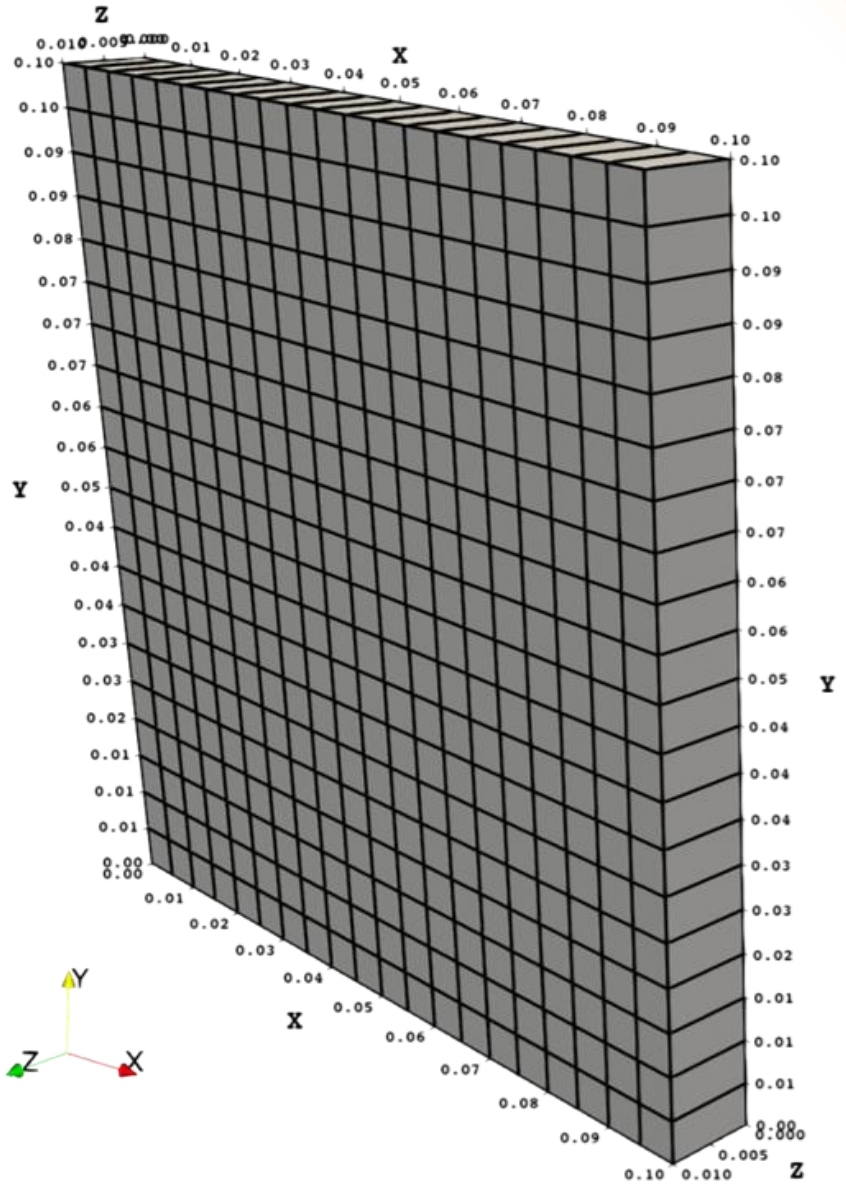
- Slicing block

../case01/system/blockMeshDict

```
31.  blocks
32.  (
33.      hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
34.  );
```







blockMesh – Case 2 (general)

- Extra Functionalities

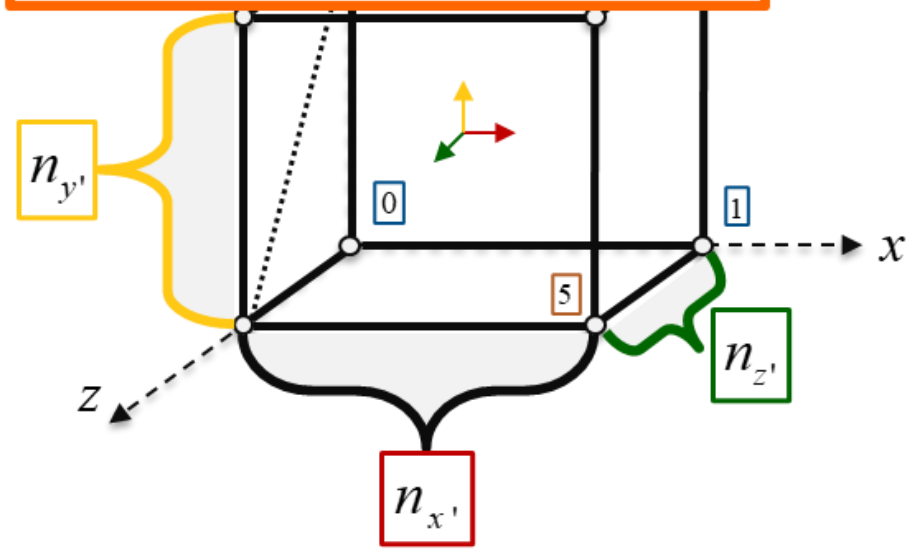
../case01/system/blockMeshDict

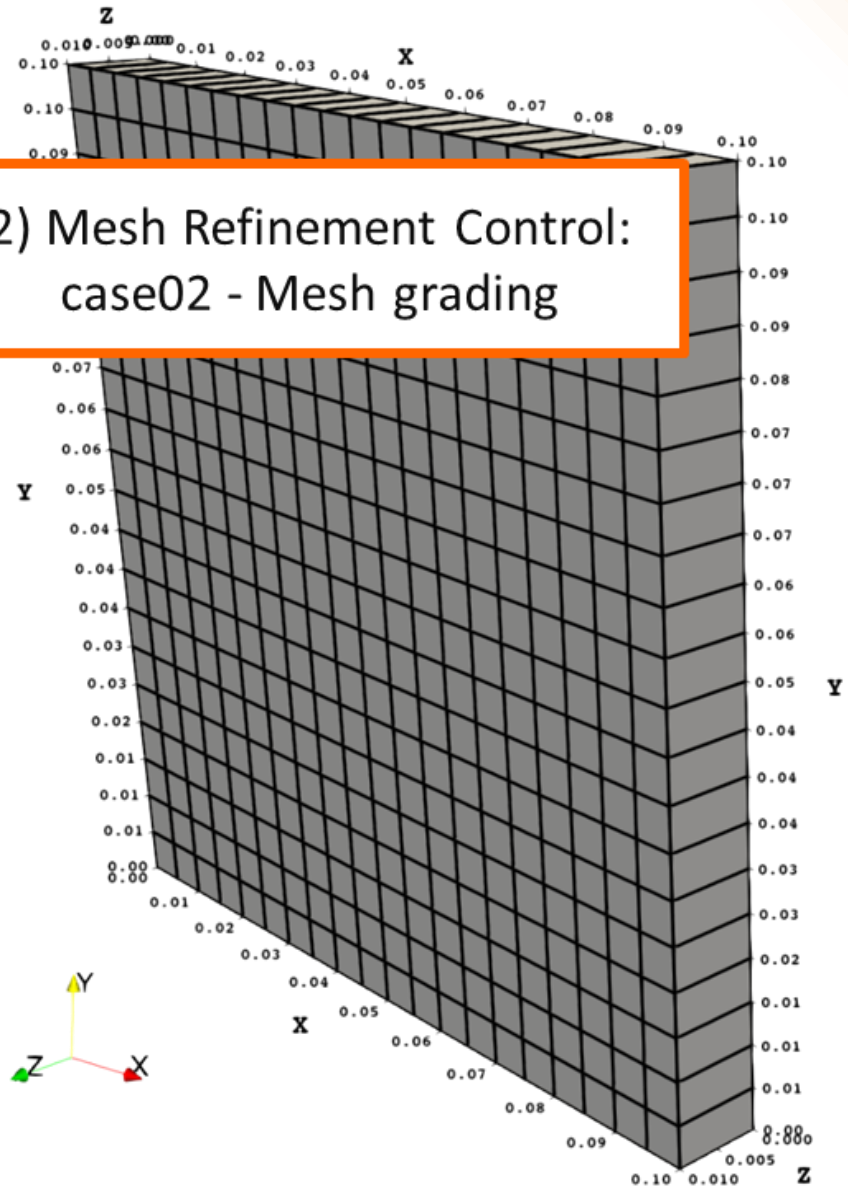
```
31.  blocks
32.  (
33.      hex (0 1 2 3 4 5 6 7) (20 20  1) simpleGrading (1 1 1)
34.  );
35.
36.  edges
37.  (
38.  );
```

z' $n_{x'}$ $n_{y'}$ $n_{z'}$ x' y'

2) Mesh Refinement Control: case02 - Mesh grading

4) Curved Surfaces: Designing curvilinear regions





blockMesh – Case 2 (general)

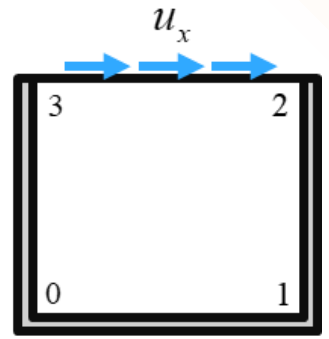
- Naming and Predefining Boundaries

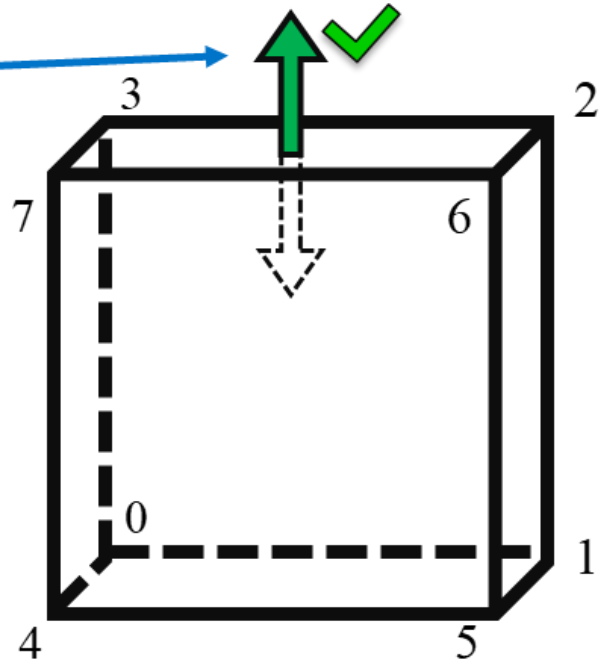
../case01/system/blockMeshDict

```
17.  boundary
18.  (
19.      movingWall
20.      {
21.          type wall;
22.          faces
23.          (
24.              (3 7 6 2)
25.          );
26.      }
27.      fixedWalls
28.      {
29.          type wall;
30.          faces
31.          (
32.              (0 4 7 3)
33.              (2 6 5 1)
34.              (1 5 4 0)
35.          );
36.      }
37.      frontAndBack
38.      {
39.          type empty;
40.          faces
41.          (
42.              (0 3 2 1)
43.              (4 5 6 7)
44.          );
45.      }
46.  );
```

User input name for the followed face(s)

Base type boundary condition





blockMesh – Case 2 (general)

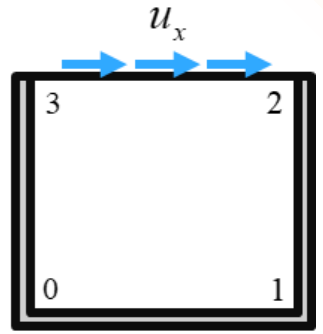
- Naming and Predefining Boundaries

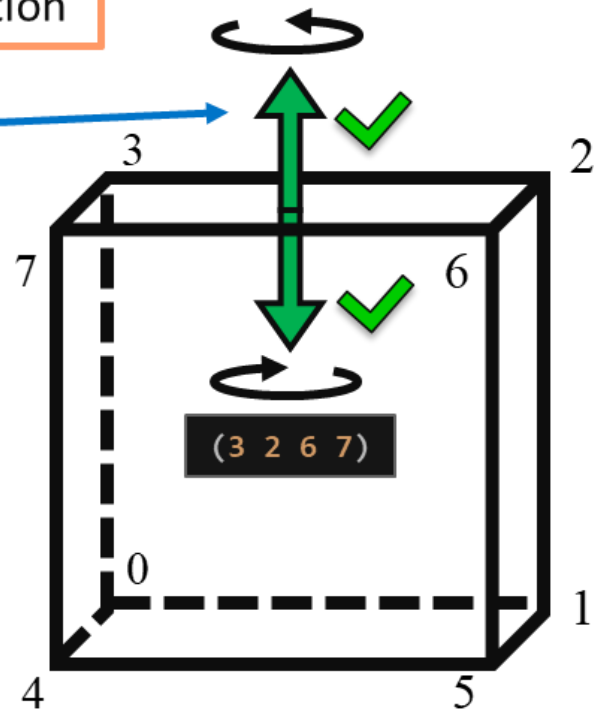
../case01/system/blockMeshDict

```
17.  boundary
18.  (
19.      movingWall
20.      {
21.          type wall;
22.          faces
23.          (
24.              (3 7 6 2)
25.          );
26.      }
27.      fixedWalls
28.      {
29.          type wall;
30.          faces
31.          (
32.              (0 4 7 3)
33.              (2 6 5 1)
34.              (1 5 4 0)
35.          );
36.      }
37.      frontAndBack
38.      {
39.          type empty;
40.          faces
41.          (
42.              (0 3 2 1)
43.              (4 5 6 7)
44.          );
45.      }
46.  );
```

User input name for the followed face(s)

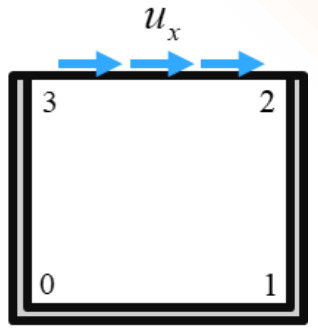
Base type boundary condition





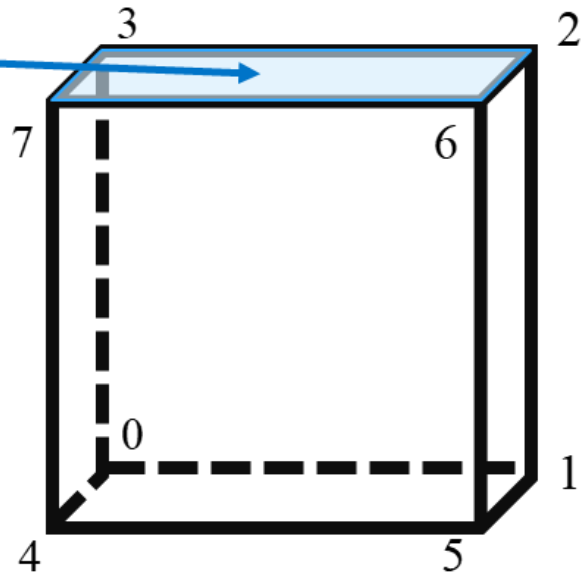
blockMesh – Case 2 (general)

- Naming and Predefining Boundaries



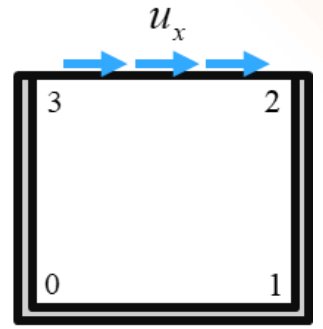
../case01/system/blockMeshDict

```
17.  boundary
18.  (
19.      movingWall
20.      {
21.          type wall;
22.          faces
23.          (
24.              (3 7 6 2)
25.          );
26.      }
27.      fixedWalls
28.      {
29.          type wall;
30.          faces
31.          (
32.              (0 4 7 3)
33.              (2 6 5 1)
34.              (1 5 4 0)
35.          );
36.      }
37.      frontAndBack
38.      {
39.          type empty;
40.          faces
41.          (
42.              (0 3 2 1)
43.              (4 5 6 7)
44.          );
45.      }
46.  );
```



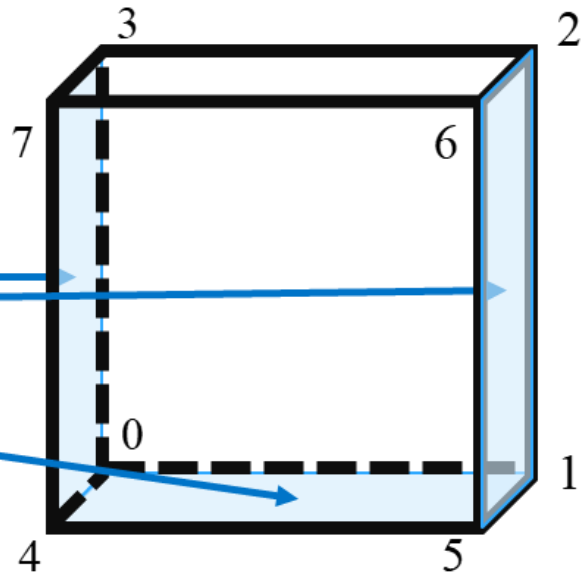
blockMesh – Case 2 (general)

- Naming and Predefining Boundaries



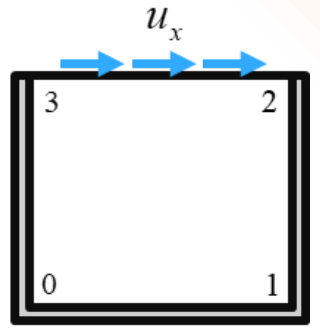
../case01/system/blockMeshDict

```
17.  boundary
18.  (
19.      movingWall
20.      {
21.          type wall;
22.          faces
23.          (
24.              (3 7 6 2)
25.          );
26.      }
27.      fixedWalls
28.      {
29.          type wall;
30.          faces
31.          (
32.              (0 4 7 3)
33.              (2 6 5 1)
34.              (1 5 4 0)
35.          );
36.      }
37.      frontAndBack
38.      {
39.          type empty;
40.          faces
41.          (
42.              (0 3 2 1)
43.              (4 5 6 7)
44.          );
45.      }
46.  );
```



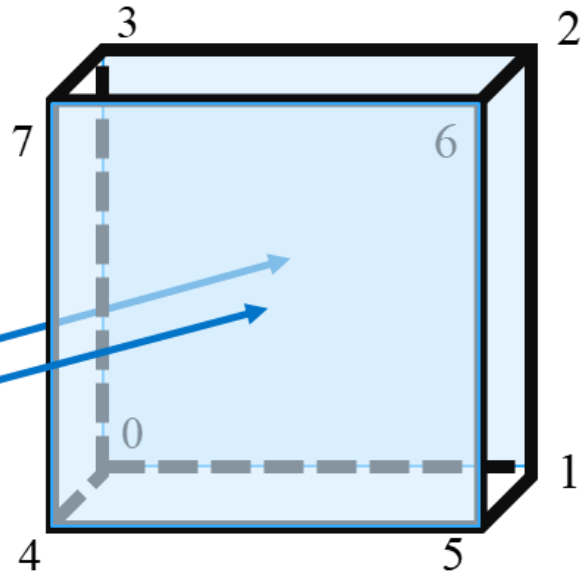
blockMesh – Case 2 (general)

- Naming and Predefining Boundaries



../case01/system/blockMeshDict

```
17.  boundary
18.  (
19.      movingWall
20.      {
21.          type wall;
22.          faces
23.          (
24.              (3 7 6 2)
25.          );
26.      }
27.      fixedWalls
28.      {
29.          type wall;
30.          faces
31.          (
32.              (0 4 7 3)
33.              (2 6 5 1)
34.              (1 5 4 0)
35.          );
36.      }
37.      frontAndBack
38.      {
39.          type empty;
40.          faces
41.          (
42.              (0 3 2 1)
43.              (4 5 6 7)
44.          );
45.      }
46.  );
```



blockMesh – Case 2 (general)

1. >> blockMesh
2. Check results in Paraview

Hands on (change and visualize)

1. Number of cells
2. Block height from 1 dm to 2 dm

blockMesh – Case 2 (grading)

1. >> run
2. >> cd case2/grading/
3. Visualize blockMeshDict file in VSCode

blockMesh – Case 2 (grading)

- Mesh grading for stretching the mesh towards one or more Planes.
- cavity Cavity case

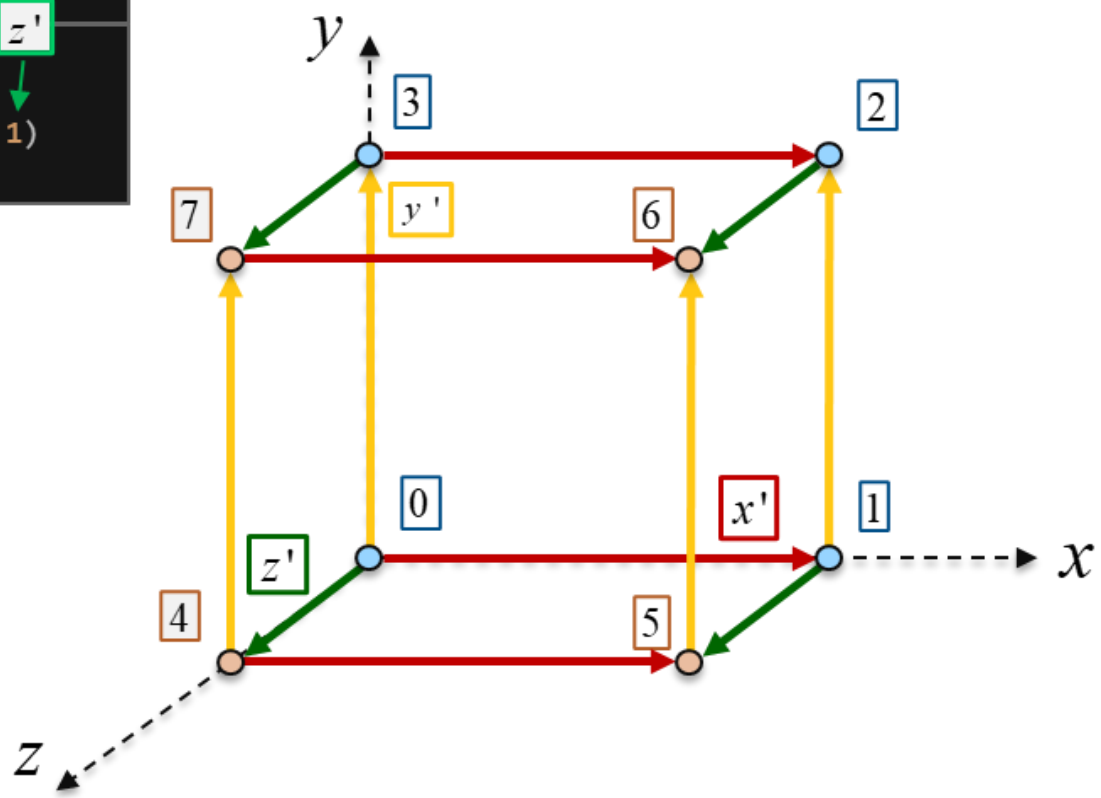
../case01/system/blockMeshDict

```
31.  blocks
32.  (
33.      hex (0 1 2 3 4 5 6 7) (20 20  1) simpleGrading ( 1  1  1)
34.  );
```

x' y' z' x' y' z'

simpleGrading = edges' cells expansion ratios

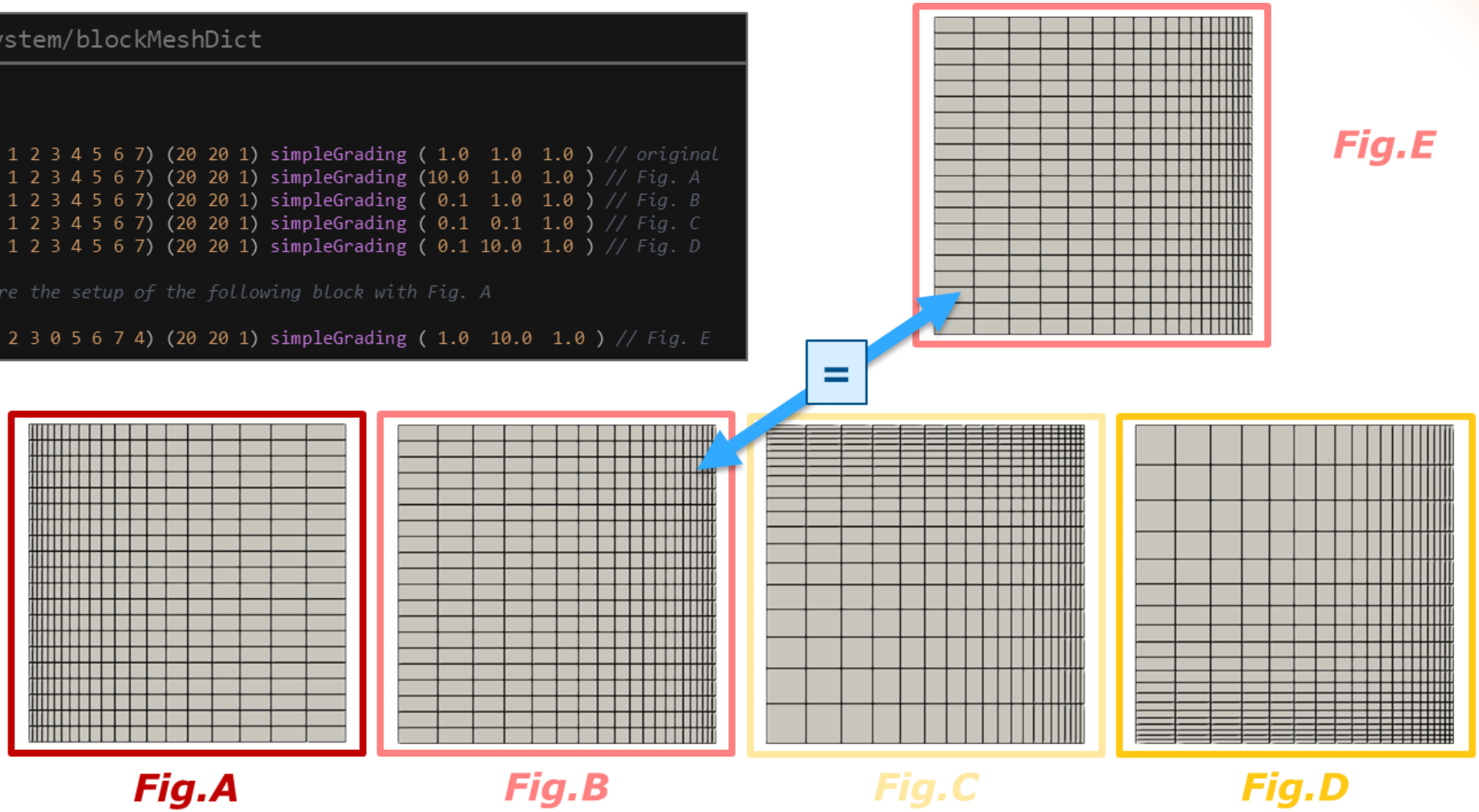
$$f_i = \frac{l_i^n}{l_i^0} \qquad f_{x'} = \frac{l_{x'}^n}{l_{x'}^0}$$



blockMesh – Case 2 (grading)

../case02/system/blockMeshDict

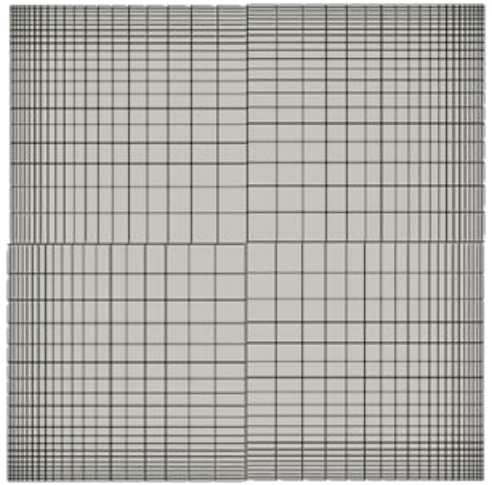
```
31.  blocks
32.  (
       …
44.     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 1.0  1.0  1.0 ) // original
45.     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (10.0  1.0  1.0 ) // Fig. A
46.     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 0.1  1.0  1.0 ) // Fig. B
47.     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 0.1  0.1  1.0 ) // Fig. C
48.     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 0.1 10.0  1.0 ) // Fig. D
49.
50.   // Compare the setup of the following block with Fig. A
51.
52.     hex (1 2 3 0 5 6 7 4) (20 20 1) simpleGrading ( 1.0  10.0  1.0 ) // Fig. E
```



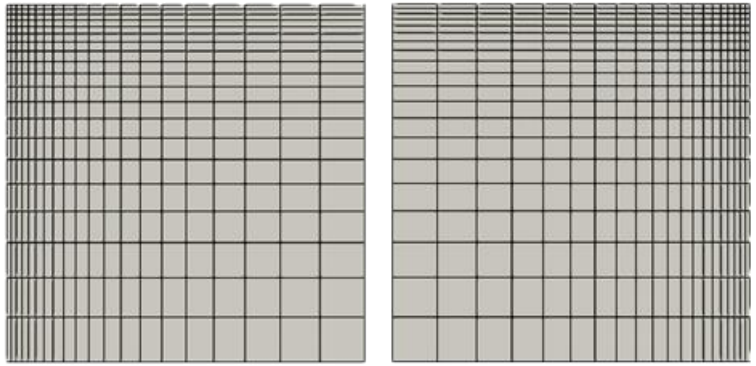
blockMesh – Case 2 (grading)

../case02/system/blockMeshDict

```
31.  blocks
32.  (
33.      hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (10.0  0.1  1.0 ) // Block 0
34.      hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 0.1  0.1  1.0 ) // Block 1
35.      hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (10.0 10.0  1.0 ) // Block 2
36.      hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 0.1 10.0  1.0 ) // Block 3
37.
```

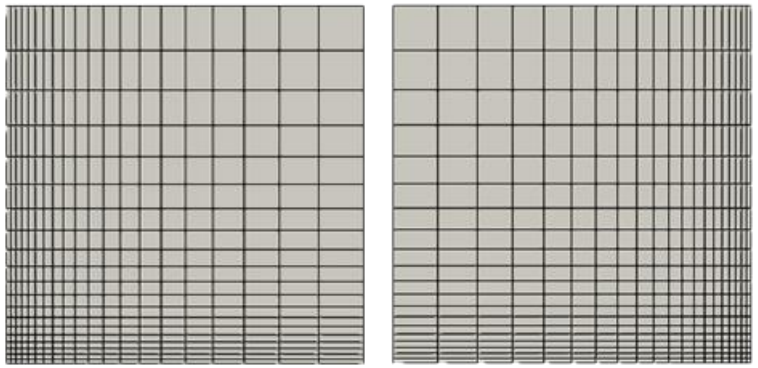


Block 0



Block 1

Block 2



Block 3

blockMesh – Case 2 (grading)

1. >> blockMesh
2. Check results in Paraview

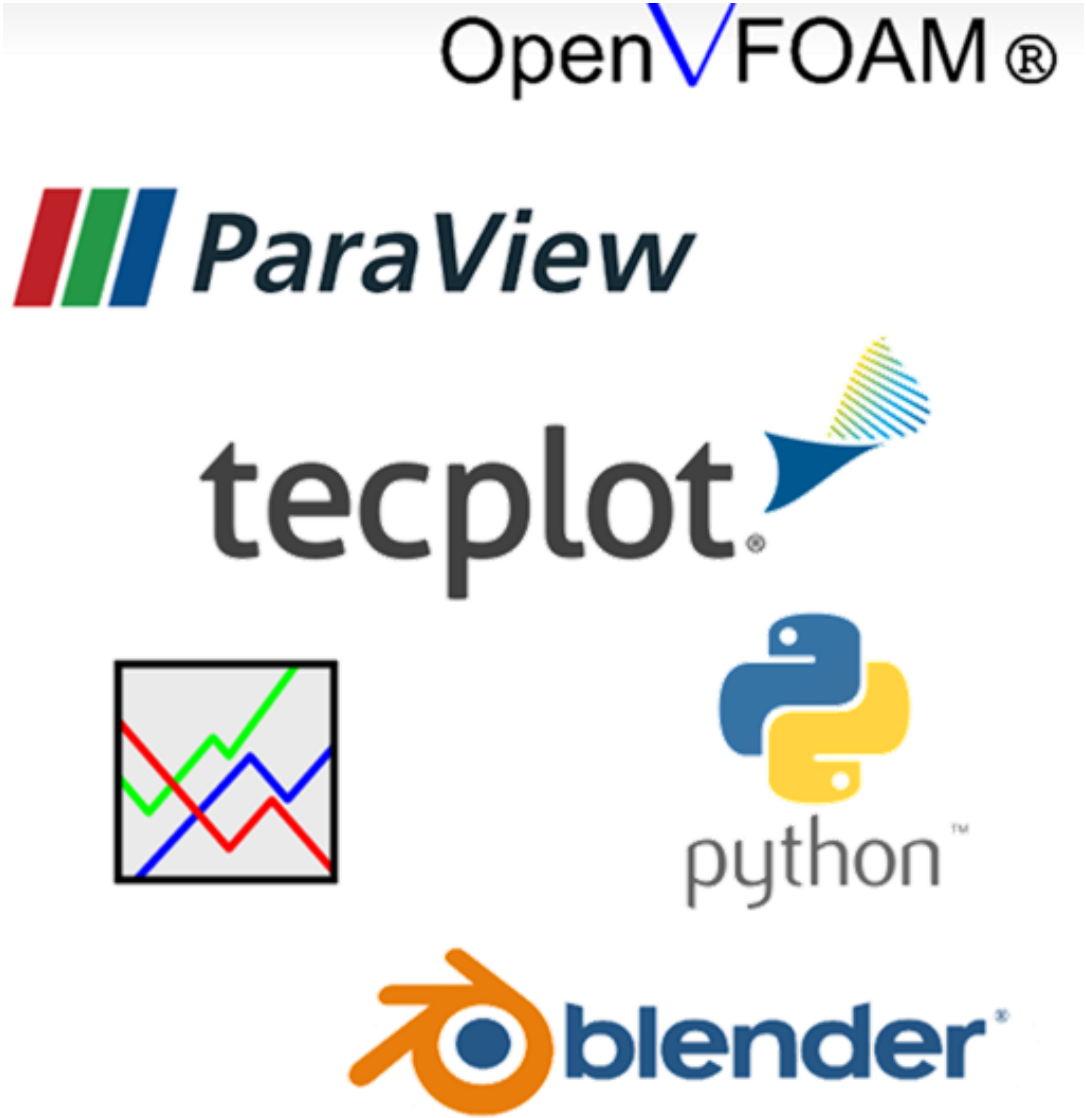
Hands on (change and visualize)

1. Test the different mesh grading configurations

Outline

14:00 – 15:00	**Session 1**: Introduction to OpenFOAM and Simulation Fundamentals
15:00 – 15:30	**Session 2a**: Geometry, Mesh Generation, and Case Setup in OpenFOAM
15:30 – 16:00	Coffee-Break
16:00 – 16:30	**Session 2**: Geometry, Mesh Generation, and Case Setup in OpenFOAM
16:30 – 17:30	**Session 3**: Hands-On Coating Case Study

Post-processing



Post-processing - Introduction

1. >> run
2. >> cd case2/post/
3. >> touch x.foam ## create file with “foam” extension to allow Paraview identify a OpenFOAM case
4. >> explorer.exe . ## Open the current folder in Windows Explorer
5. Open the x.foam file with paraview

Post-processing – User Interface | Over view

Menu bar

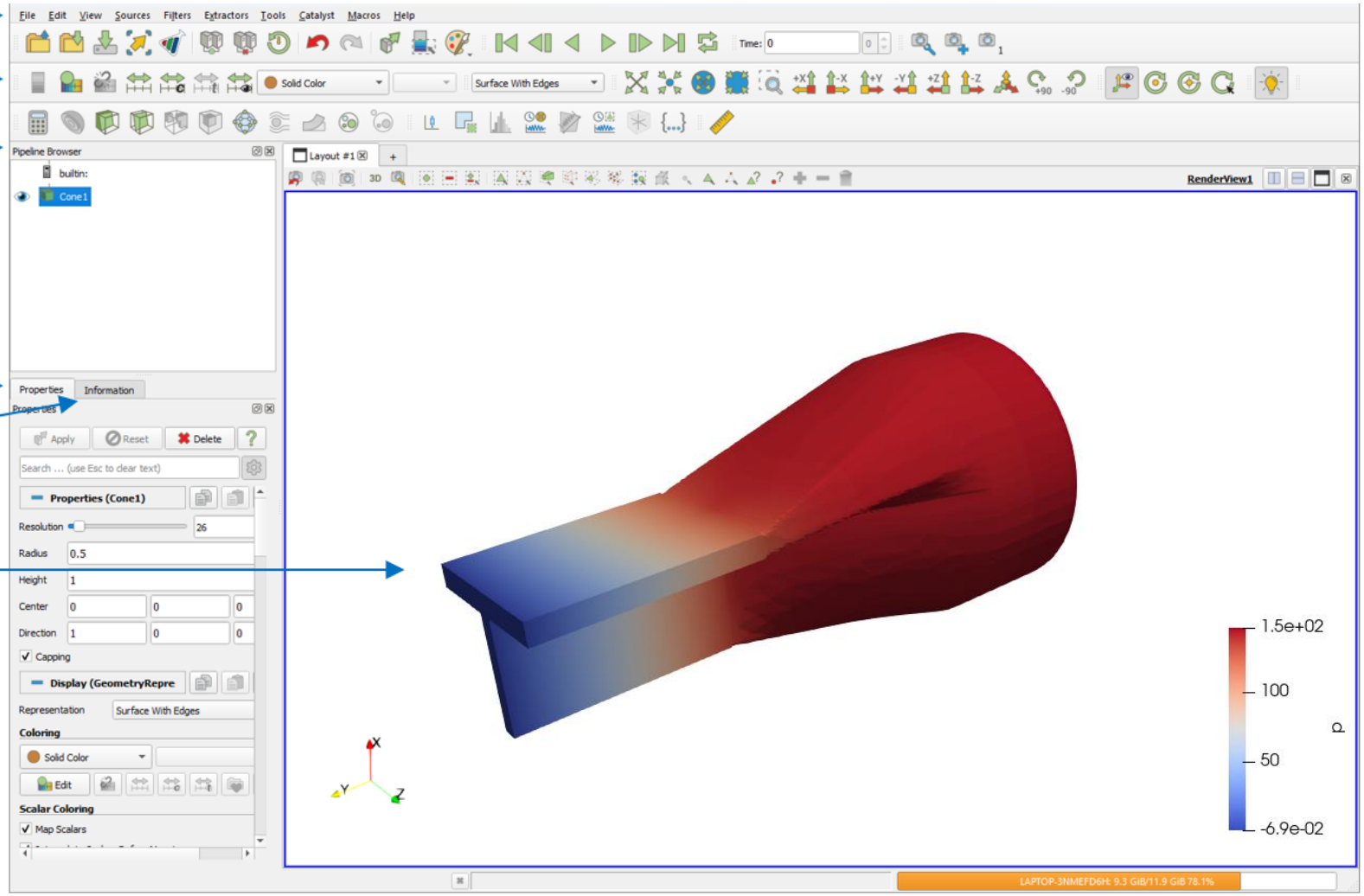
Toolbars

Pipeline Browser

Properties panel

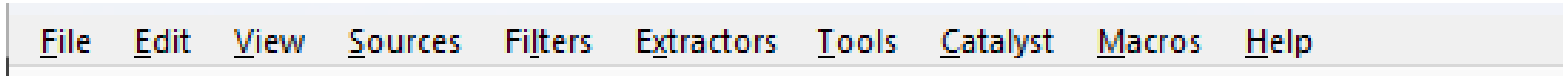
Information panel

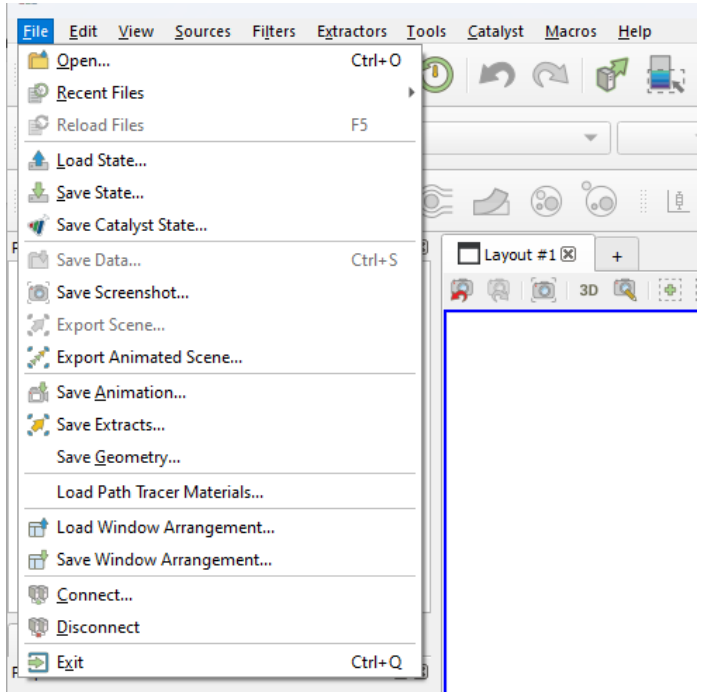
Canvas

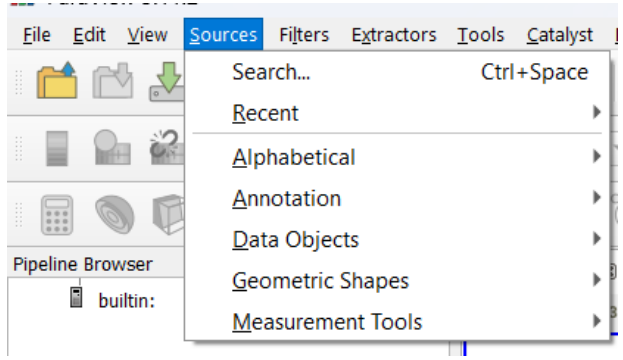


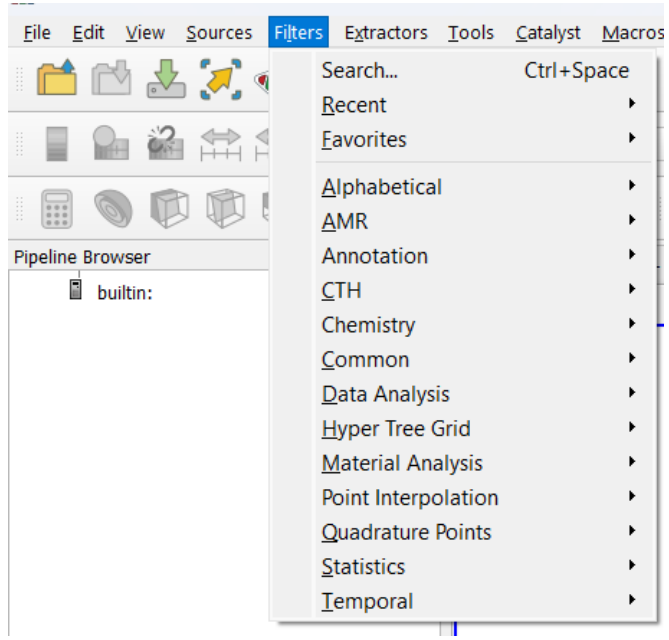
Post-processing – User Interface | Menu Bar

The menu bar allows you to access the majority of tools and features, dividing it into sections.







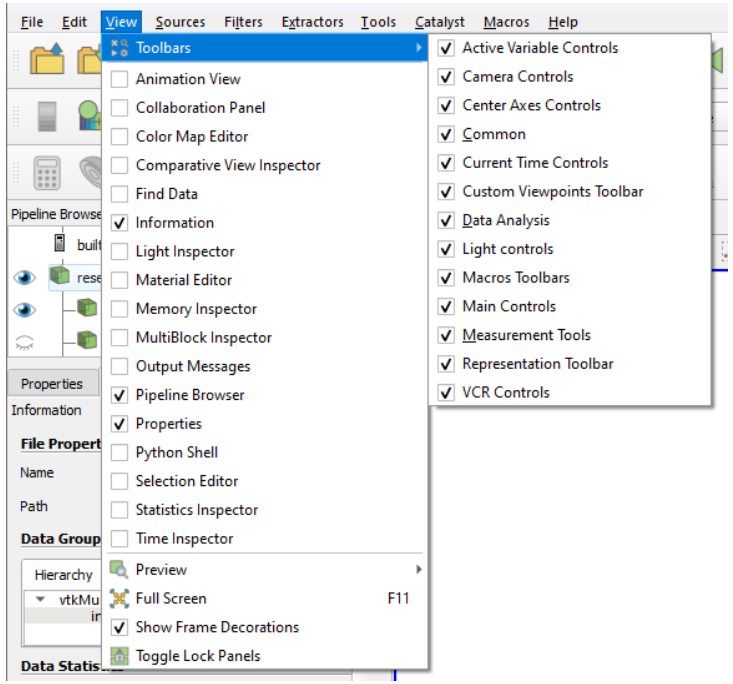


Post-processing – User Interface | Toolbars

The toolbars provide shortcuts to some features and tools.

Toolbars are customized by hiding/showing what you desire.

Menu bar View - Toolbars



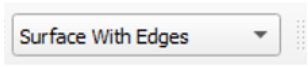
Post-processing – User Interface | Toolbars

Main Controls

VCR Controls

Current Time Controls

Custom Viewpoints Toolbar



Representation Toolbar

Active Variable Controls

Camera controls

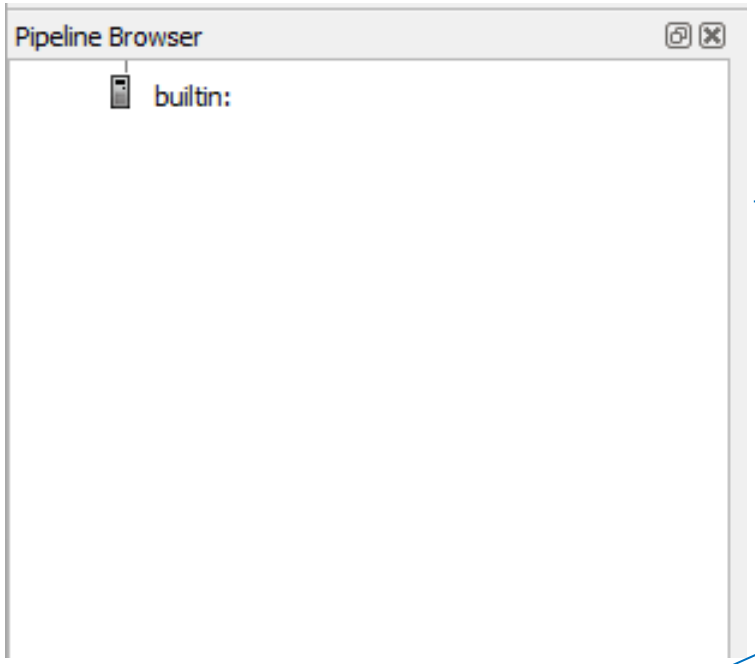
Center Axes Controls

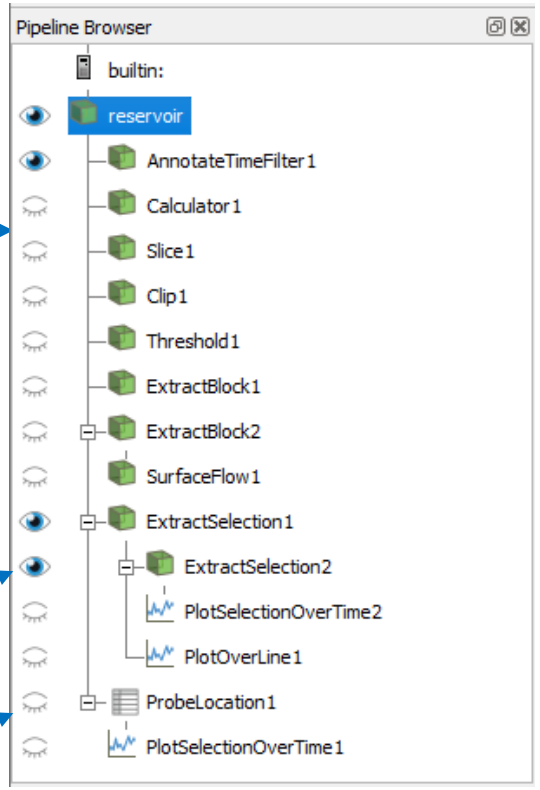
Common Filters

Data Analysis Toolbar

Post-processing – User Interface | Pipeline browser

The pipeline browser shows the pipeline structure, providing the objects being post processed and its correspondent applied filters.



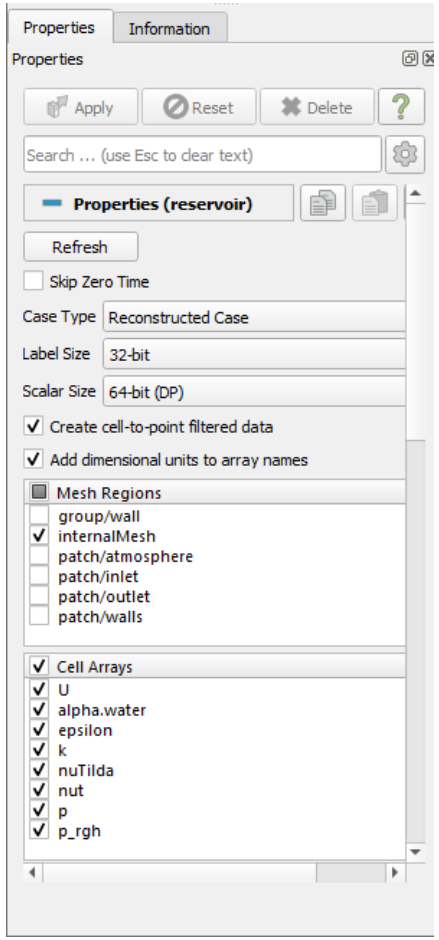


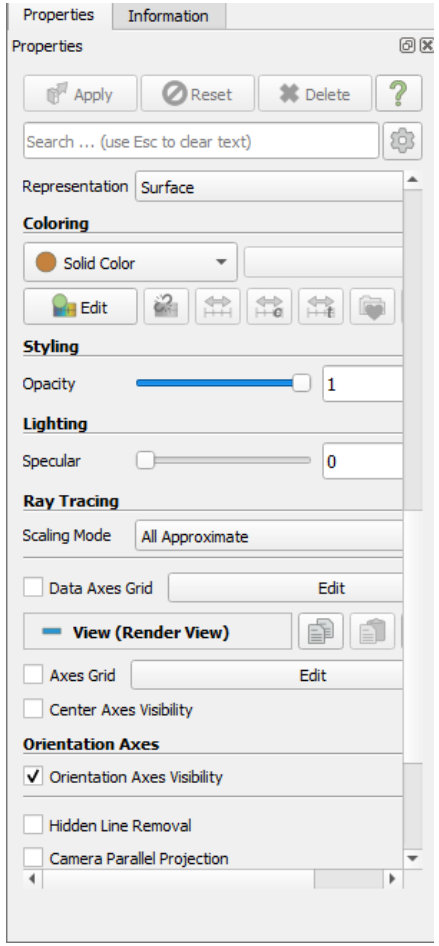
The objects/filters being shown in the canvas

The objects/filters being hidden in the canvas

Post-processing – User Interface | Properties panel

The properties panel allows you to view and modify some parameters of the selected pipeline object/filter.



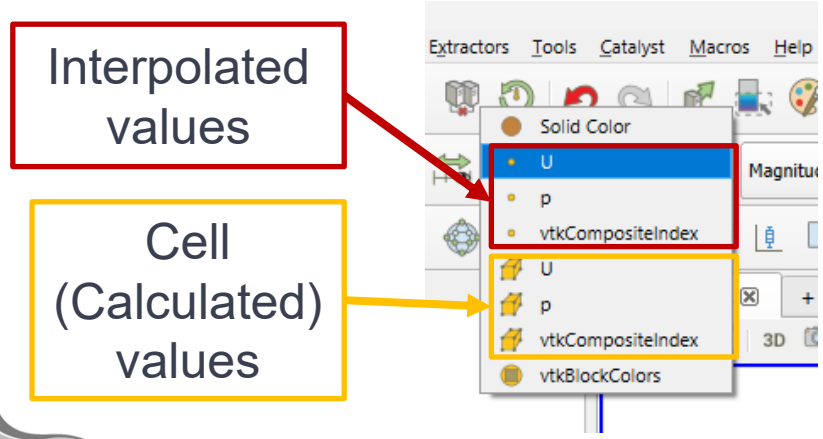


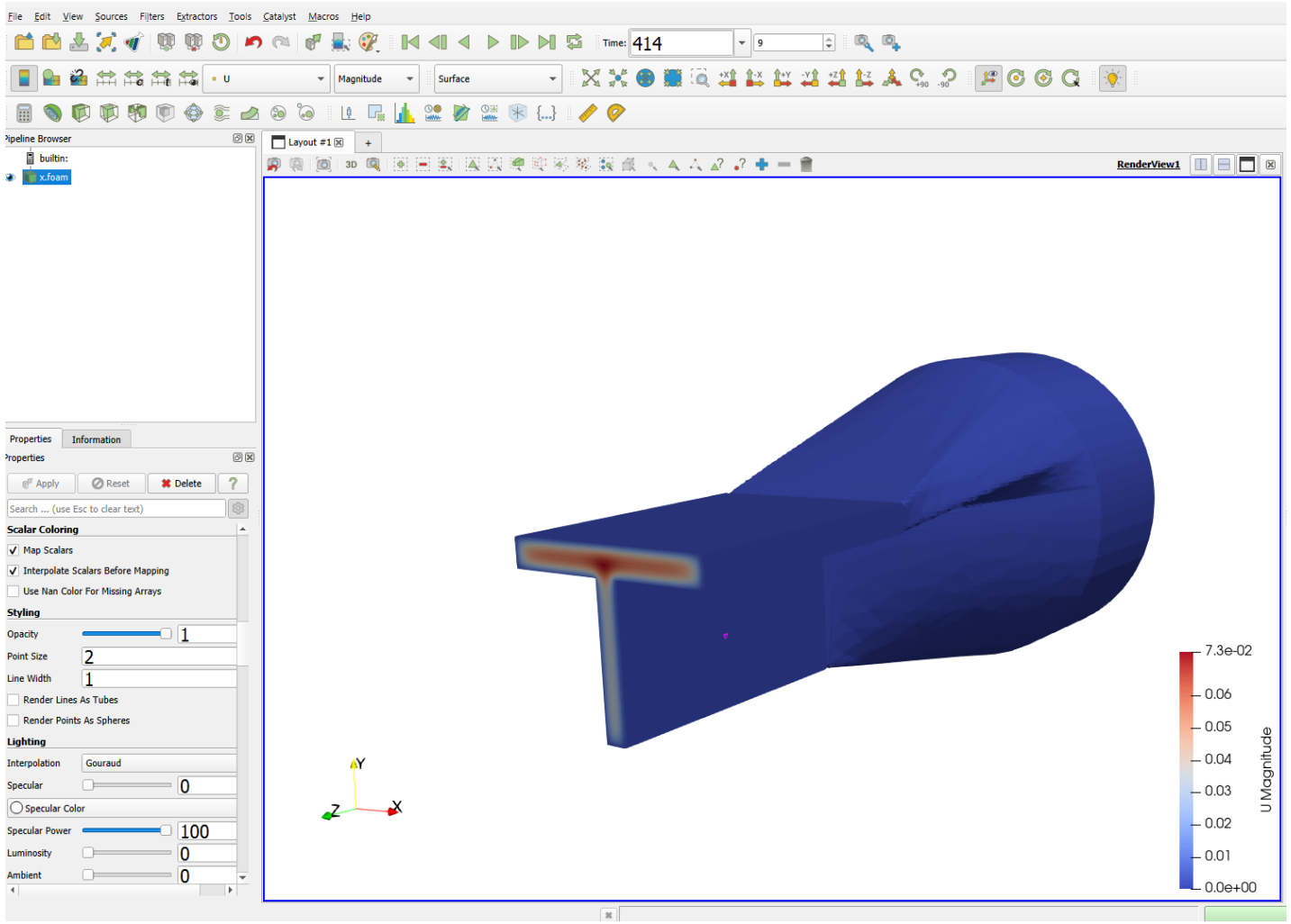
Post-processing – Data Visualizing

1. Displaying scalar and vector fields
2. Adjusting colors, data range and labels
3. Applying Common Filters
4. Select cells and show values

Post-processing – Data Visualizing

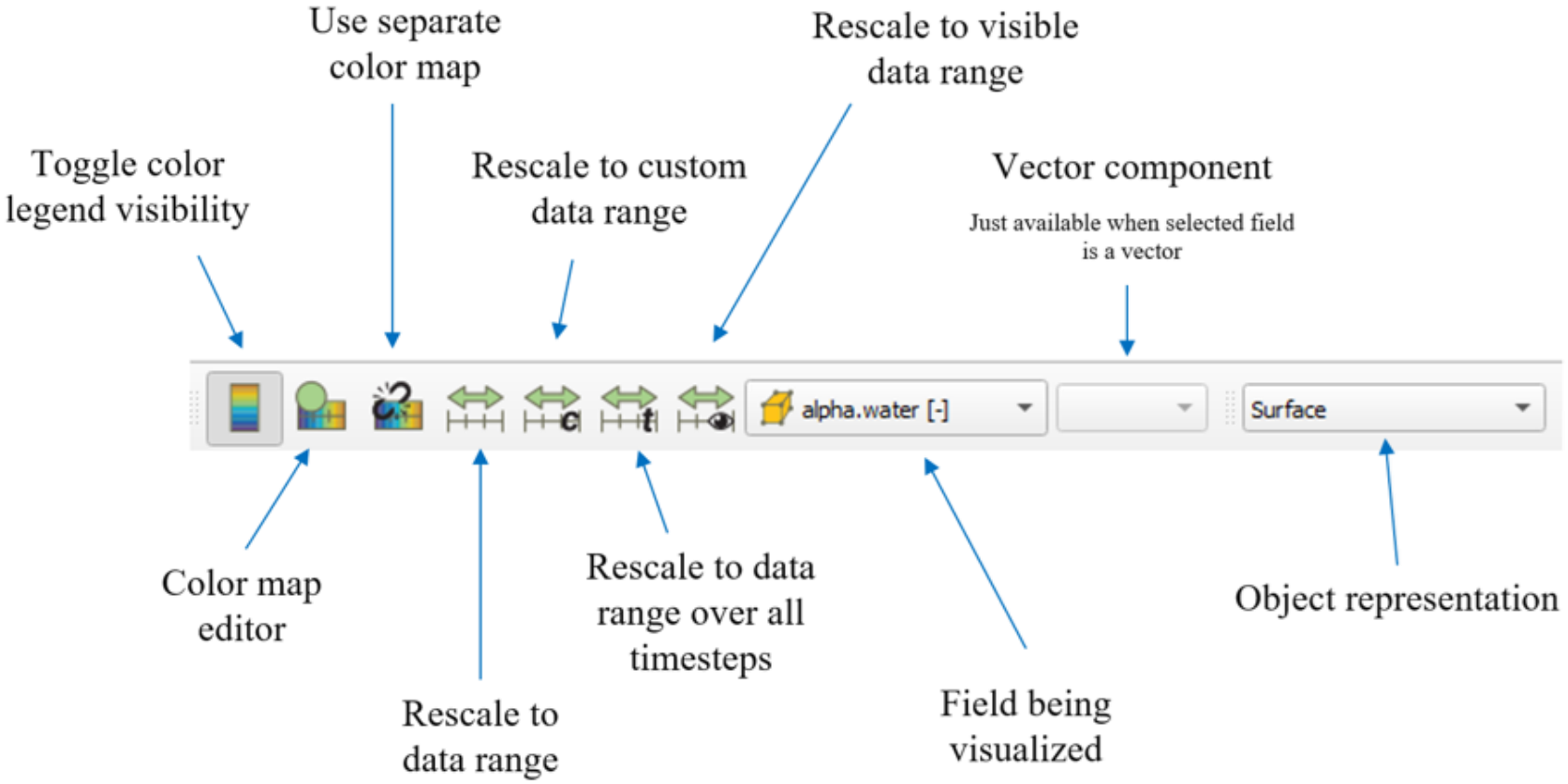
1. Displaying scalar and vector fields
2. Adjusting colors, data range and labels
3. Applying Common Filters
4. Select cells and show values





Post-processing – Data Visualizing

1. Displaying scalar and vector fields
2. Adjusting colors, data range and labels
3. Applying Common Filters
4. Select cells and show values

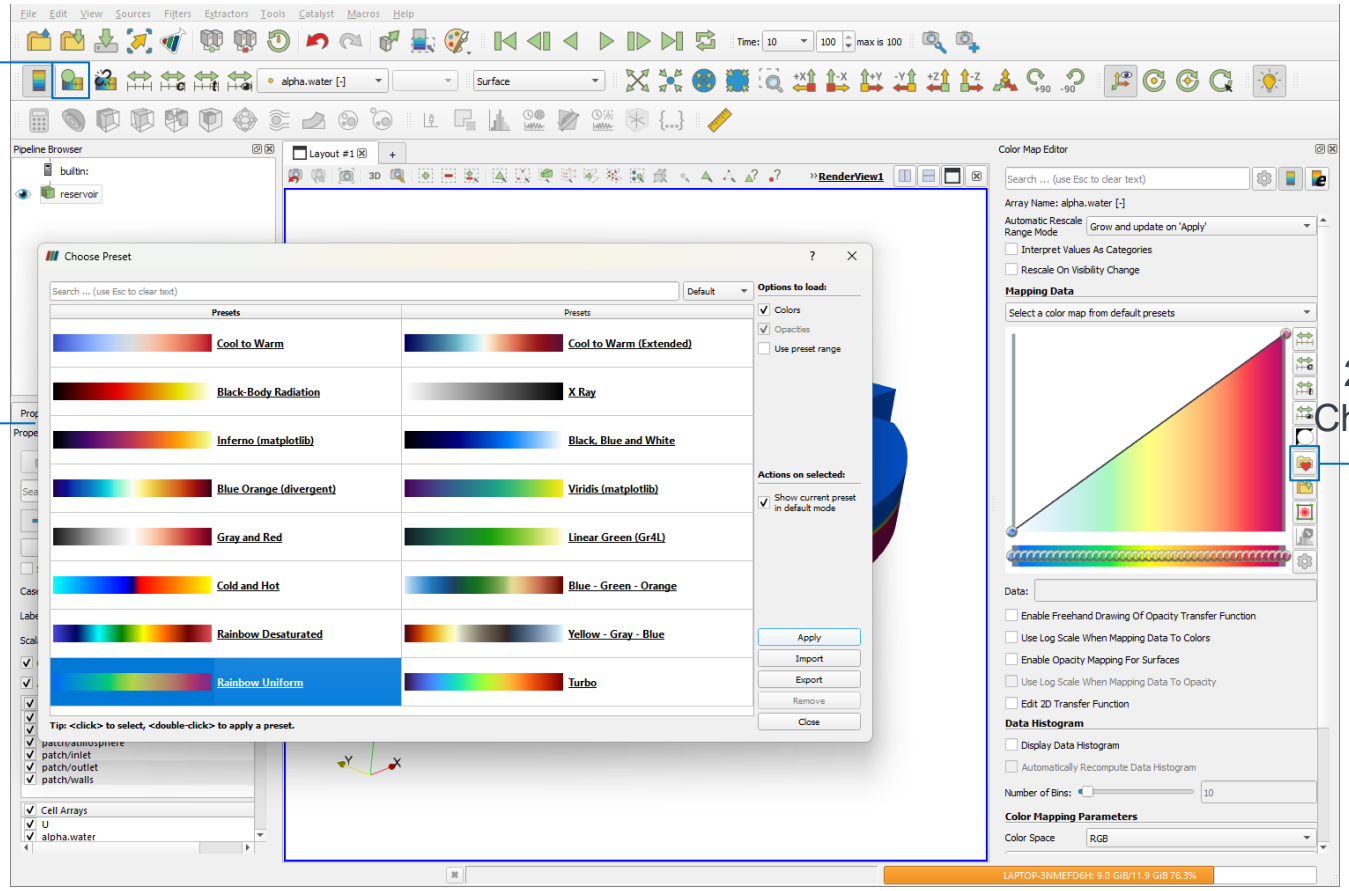


Post-processing – Data Visualizing

1. Displaying scalar and vector fields
2. Adjusting colors, data range and labels
3. Applying Common Filters
4. Select cells and show values

1 – click on
Edit color map

3 – select the
color map and
click Apply



2 – click on
Choose Preset

Post-processing – Data Visualizing

1. Displaying scalar and vector fields
2. Adjusting colors, data range and labels
3. Applying Common Filters
4. Select cells and show values

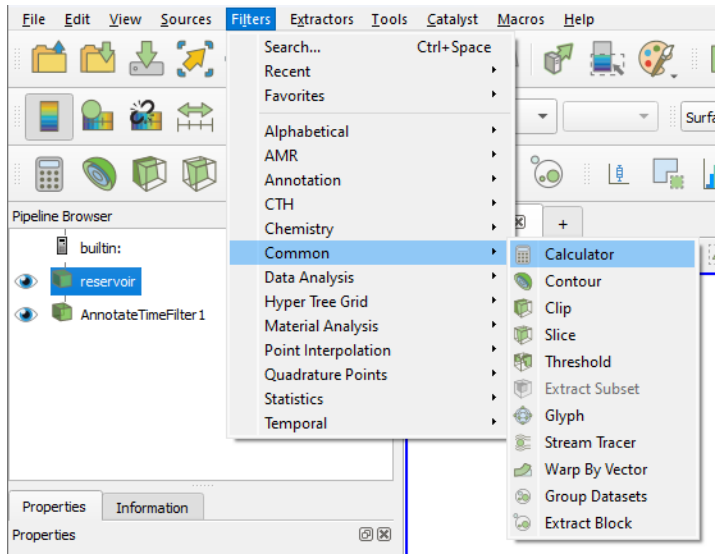
Calculator

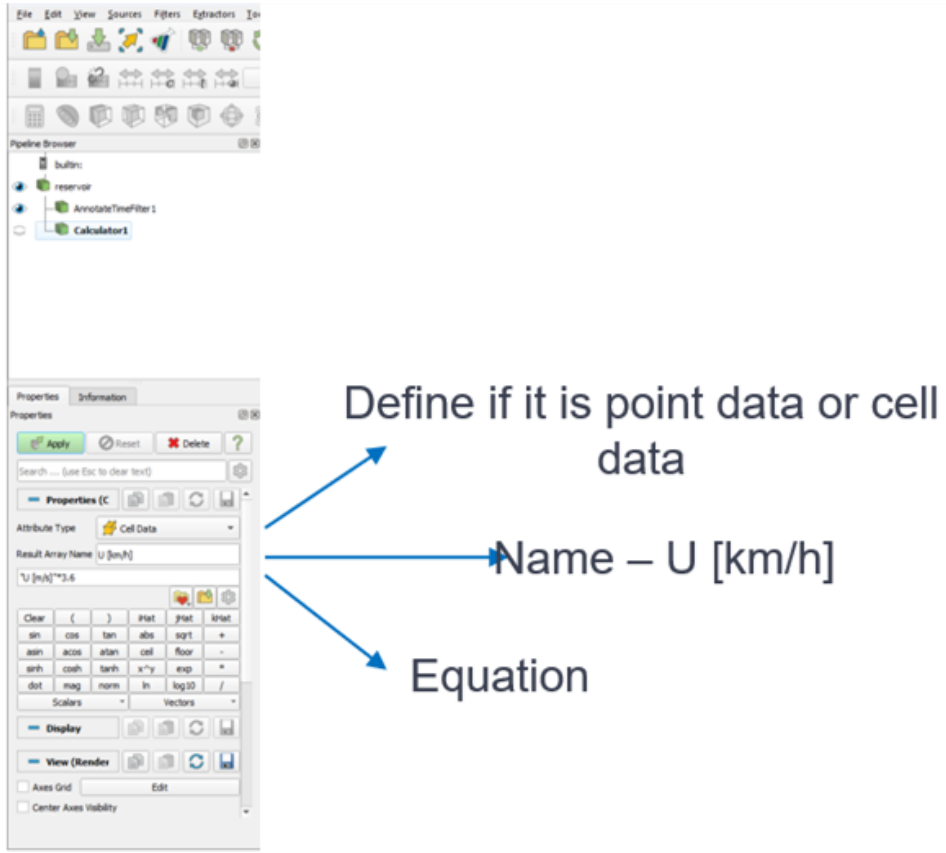
Creates an variable field according to an expression

- Select the reservoir object in the pipeline browser

- Select the Calculator filter in the menu bar

Filters > Common > Calculator



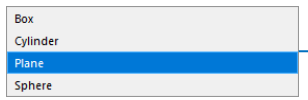


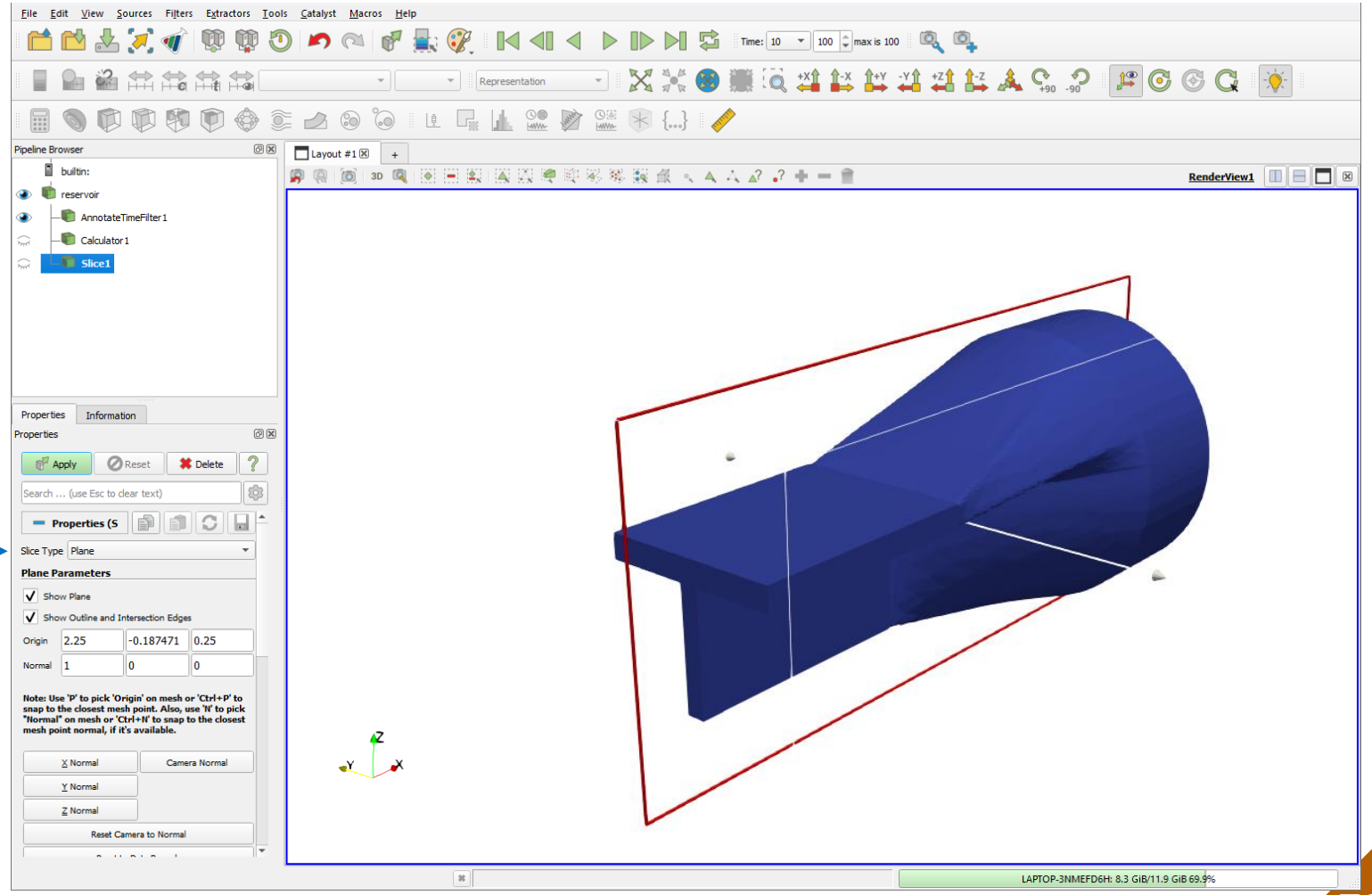
Post-processing – Data Visualizing

1. Displaying scalar and vector fields
2. Adjusting colors, data range and labels
3. Applying Common Filters
4. Select cells and show values

Slice

Slices the object with a Box, Cylinder, Plane or Sphere



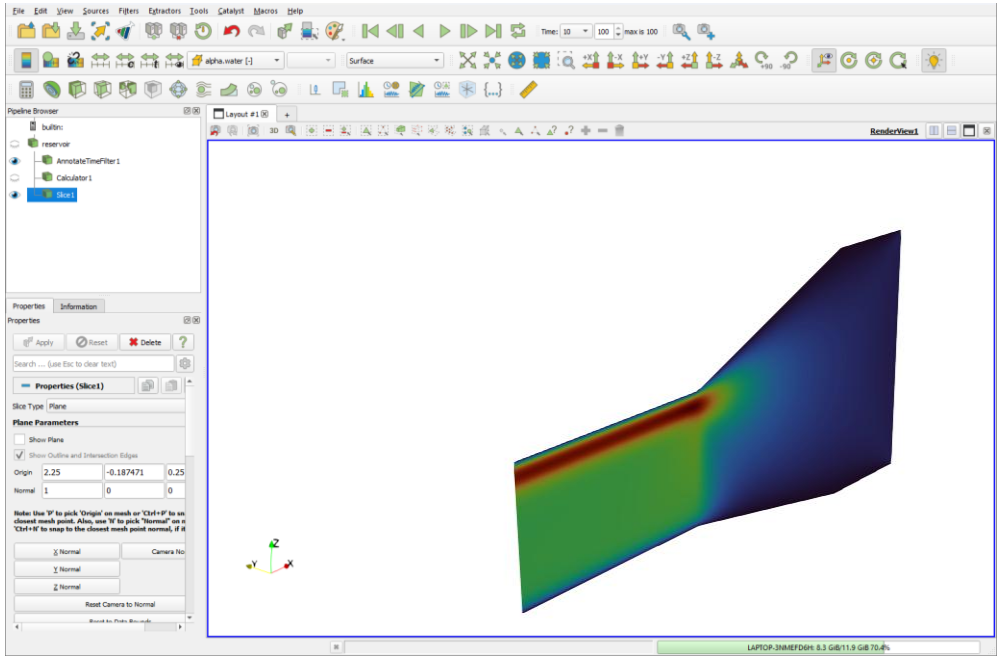


Post-processing – Data Visualizing

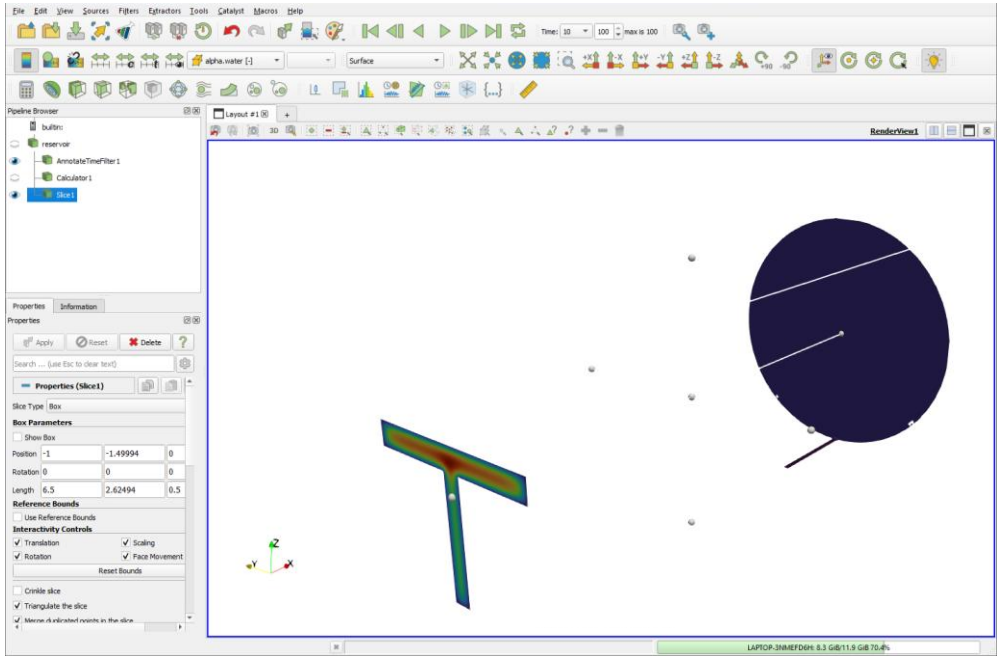
1. Displaying scalar and vector fields
2. Adjusting colors, data range and labels
3. Applying Common Filters
4. Select cells and show values

Slice

Slice - Plane



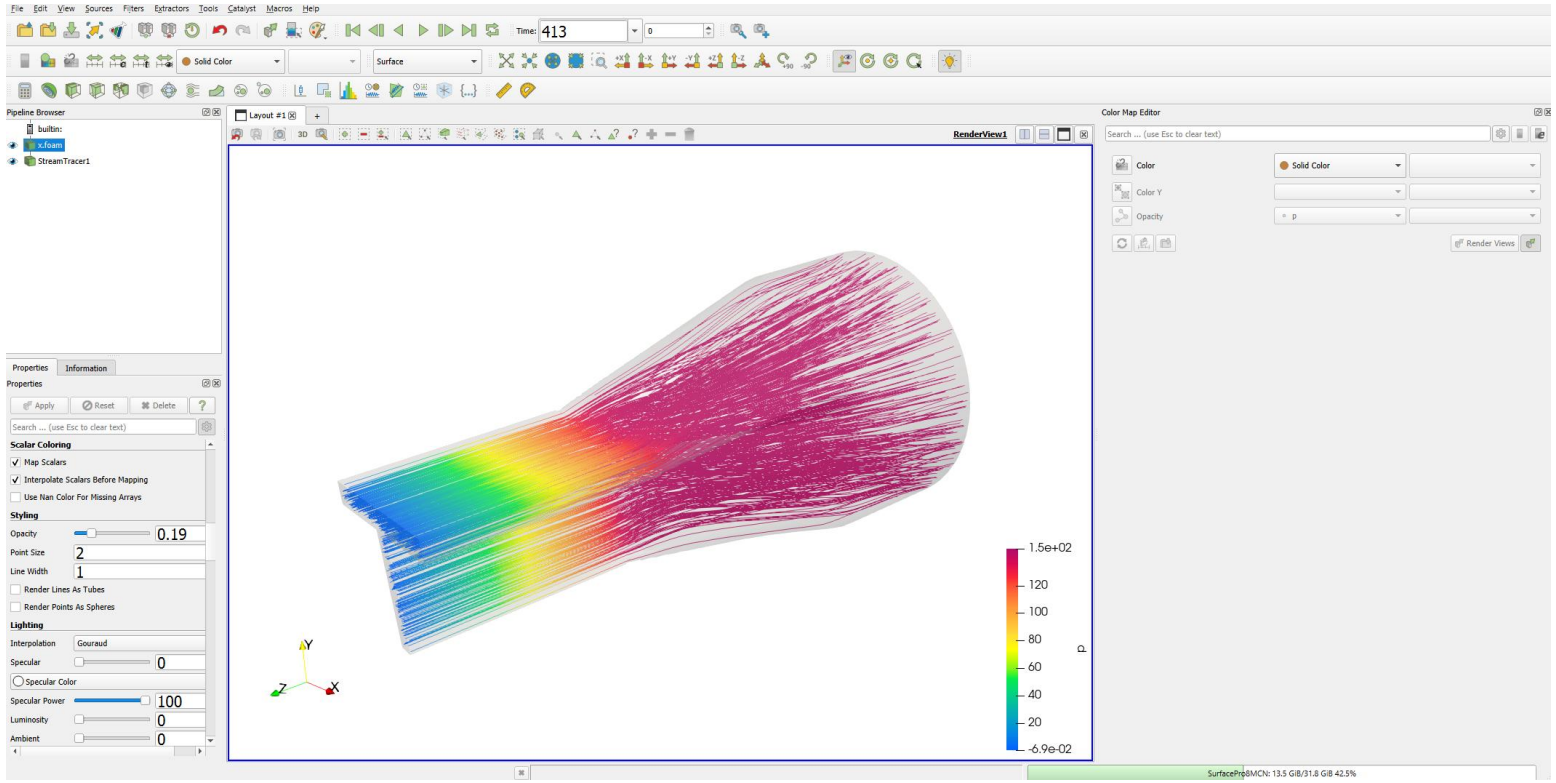
Slice - Box



Post-processing – Data Visualizing

1. Displaying scalar and vector fields
2. Adjusting colors, data range and labels
3. Applying Common Filters
4. Select cells and show values

Stream Tracer

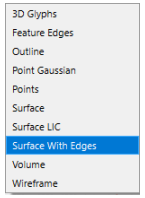


Post-processing – Data Visualizing

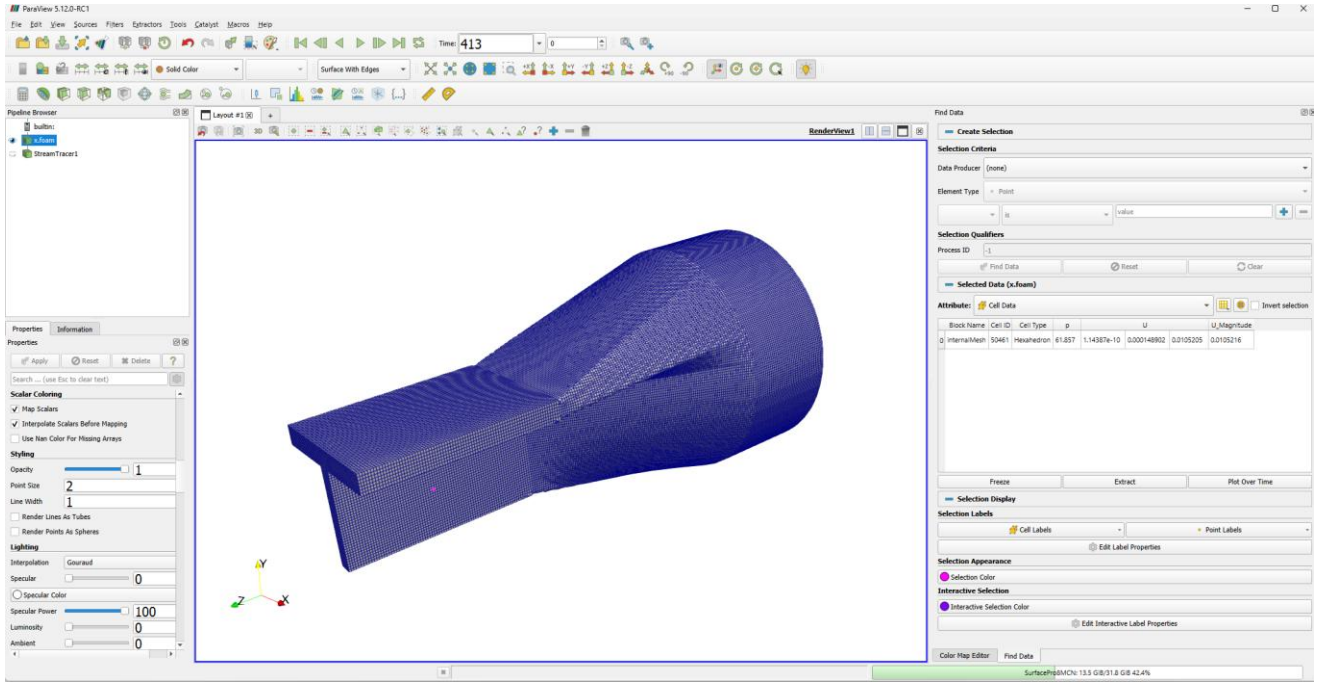
1. Displaying scalar and vector fields
2. Adjusting colors, data range and labels
3. Applying Common Filters
4. Select cells and show values

Select Cells and Show Values

a. Show the mesh

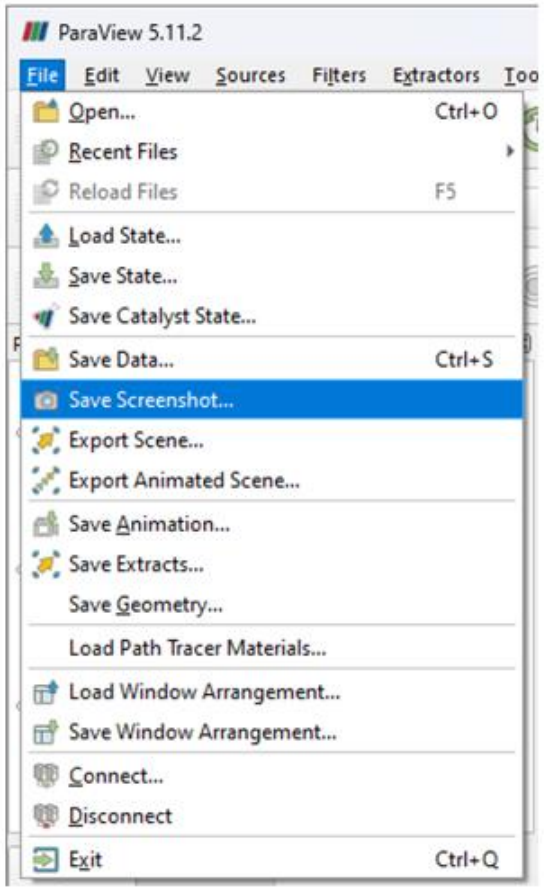


b. Press “s”
c. Select the desired cell(s)
d. Press “v”



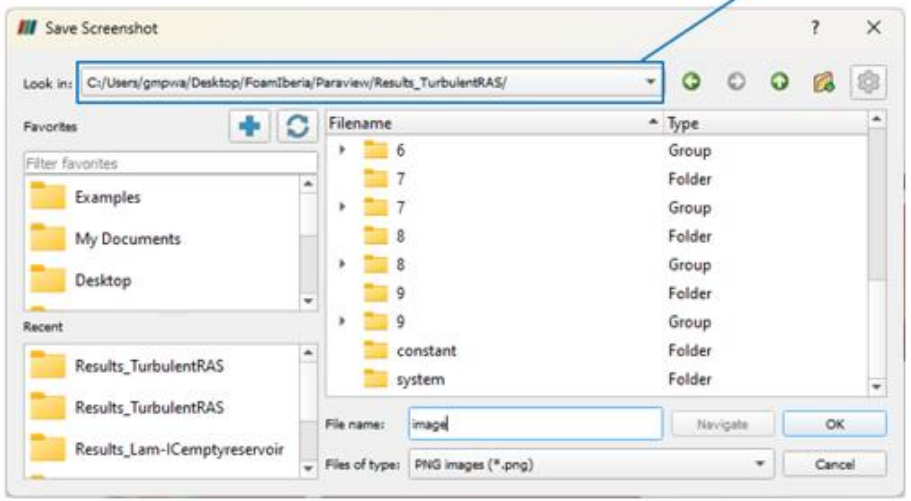
Post-processing – Saving Progress

(1)



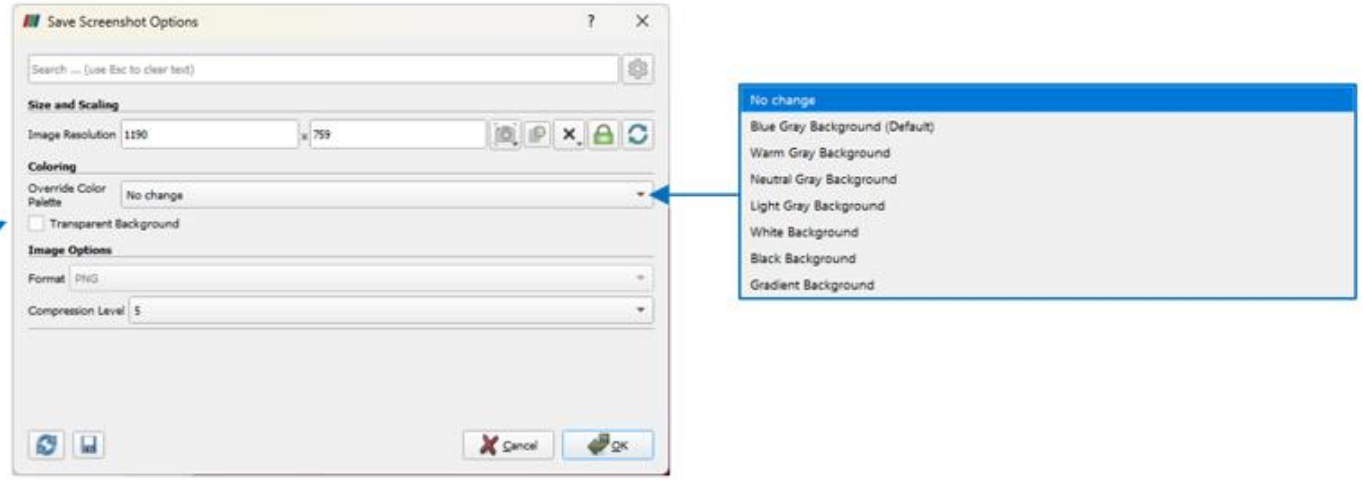
(2)

Saving location



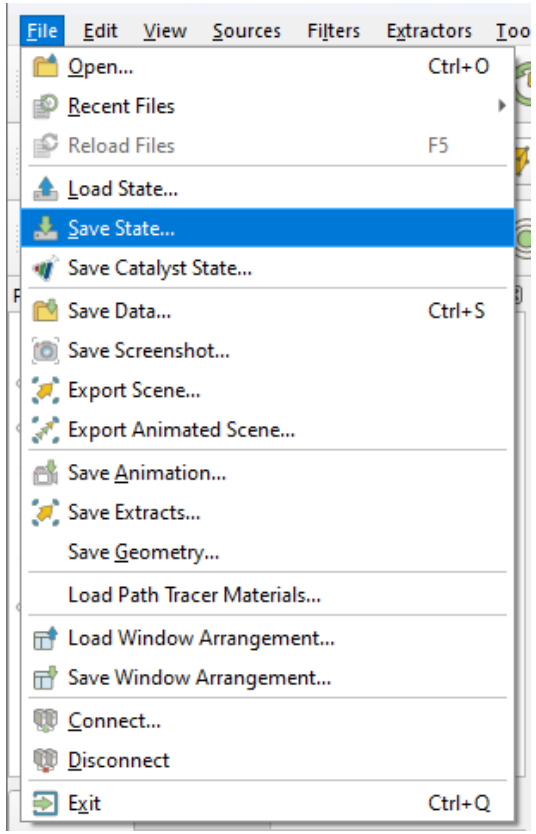
(3)

background

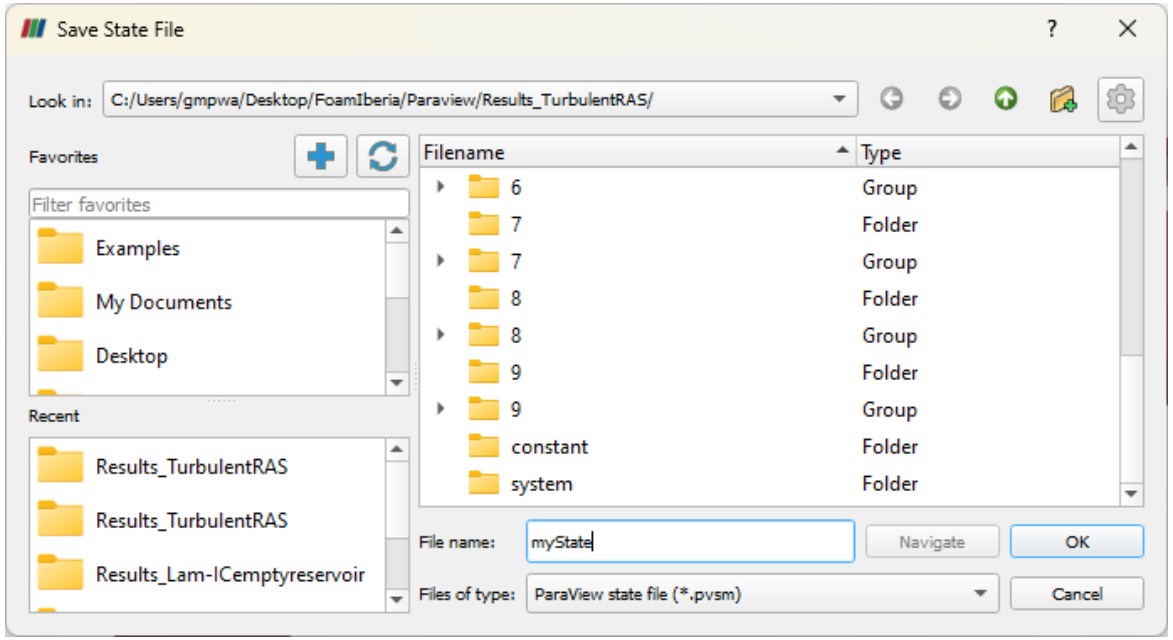


Post-processing – Saving Progress

(1)



(2)



Saving a state allow you to reproduce your visualization settings and view without applying post-processing steps again.

It will save all your progress and it will work as a macro: once you load it, you will open your Paraview visualization the same way you saved it.