

Introduction to OpenFOAM® Computational Library and Viscoelastic Fluid Flow Simulation

P2 - Mesh Generation and Post-Processing

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

Mohammadreza Aali & J. Miguel Nórrega

mohammadreza.aali@jku.at & mnobrega@dep.uminho.pt

Outline

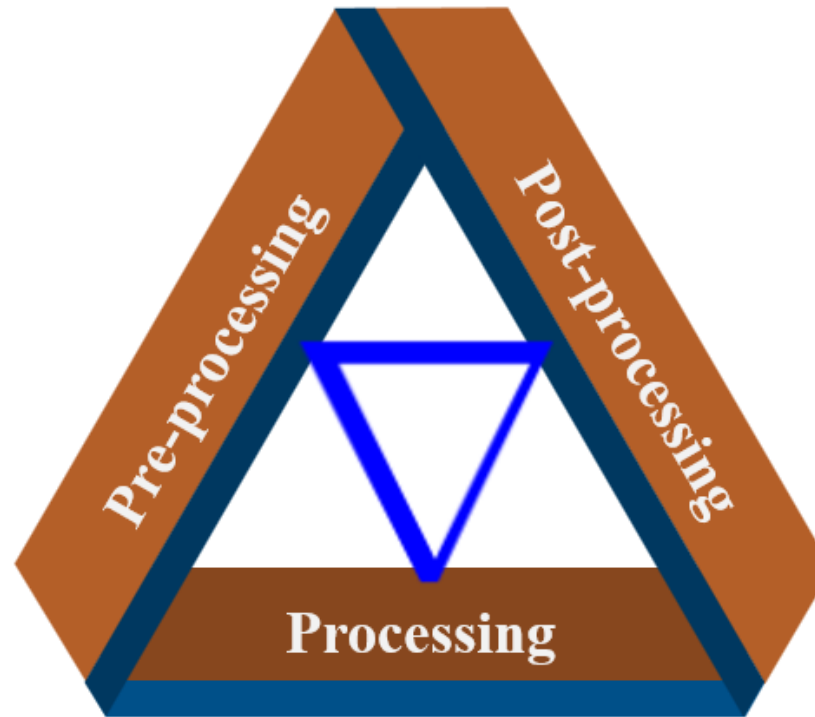
9:00 – 10:30	Introduction to OpenFOAM (P1)
10:30 – 12:00	Mesh generation and post-processing (P2)
12:00 – 13:00	Lunch break
13:00 – 14:30	Case studies: Single- and two-phase flow solvers (P3)
14:30 – 16:00	Case studies: Viscoelastic fluid flow solvers (P4)



Introduction

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

OpenFOAM®



- 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)
...
```

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

```
5468697320697320616E
206578616D706C652066
726F6D205761676E6572
2047616C7570706F206F
66206120686578616465
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)
...
```

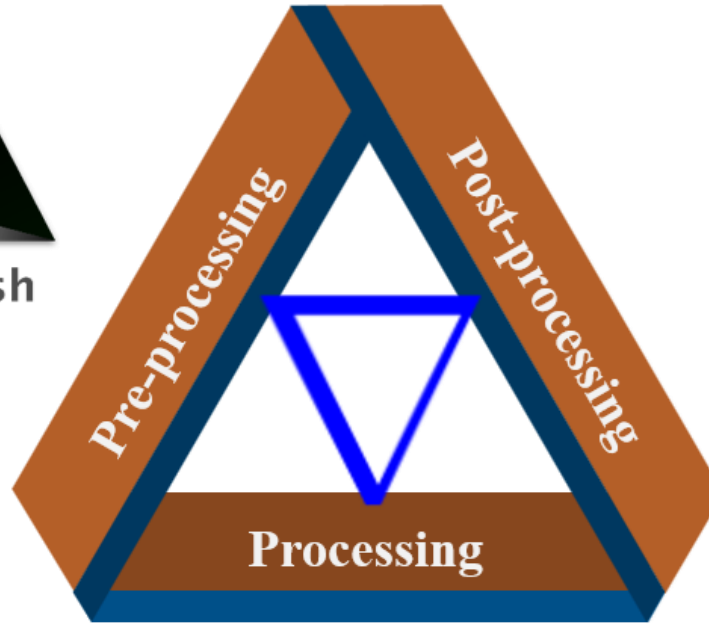


Introduction

Open▽FOAM®

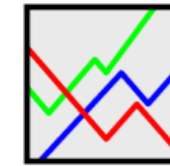


SALOME



Open▽FOAM®

Open▽FOAM®



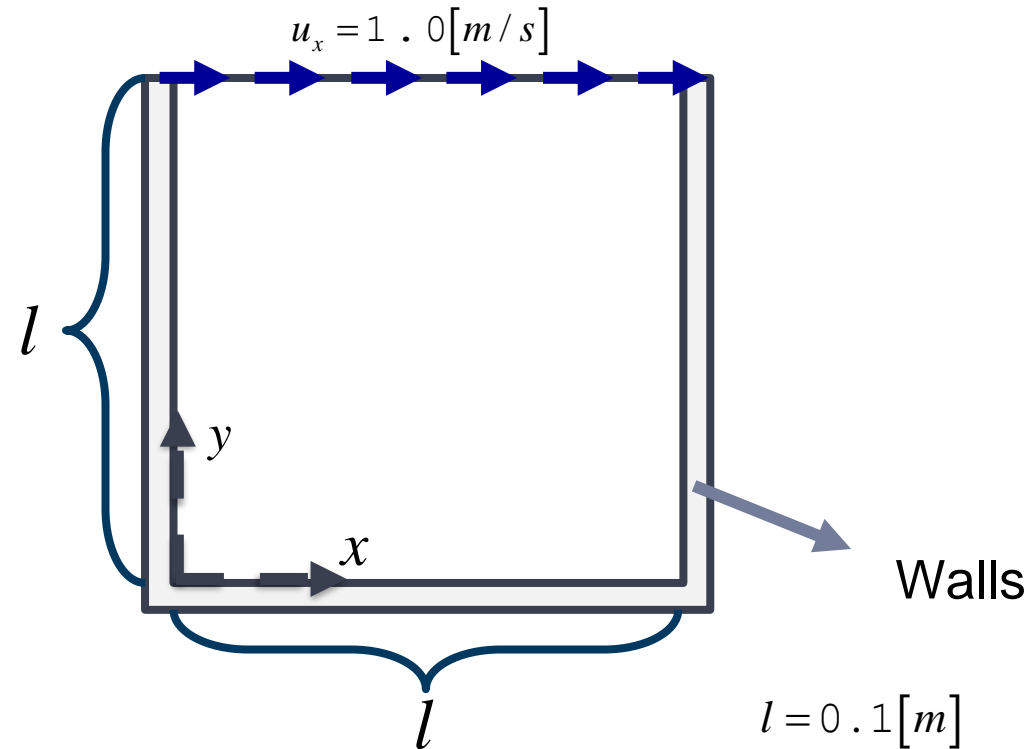
blockMesh – Case 21 (cavity)

1. Open WSL
2. `>> of2206`
3. `>> run`
4. `>> cd case21`
5. `>> code . ## Open Vscode`
6. Visualize blockMeshDict file ##in the System folder



blockMesh – Case 21 (cavity)

- Physical Problem: (Cavity tutorial)



blockMesh – Case 21 (cavity)

BlockMesh Dictionary case21/system/blockMeshDict

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

/*-----*- C++ -*-----*/
| ===== |
| \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / O p e r a t i o n | Version: v2306 |
| \ \ / A n d | Website: 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 0.1;
scale 0.1;

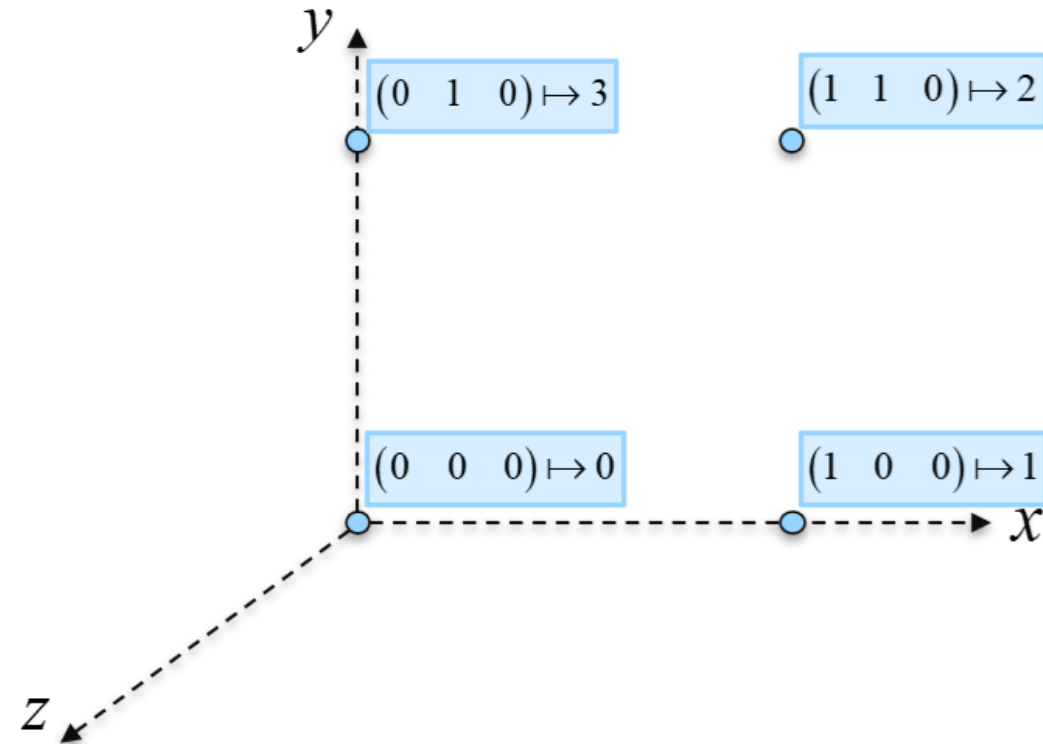
vertices
(
    (0 0 0)
    (1 0 0)
    (1 1 0)
    (0 1 0)
    (0 0 0.1)
    (1 0 0.1)
    (1 1 0.1)
    (0 1 0.1)
);
```



blockMesh – Case 21 (cavity)

- Generating Vertices

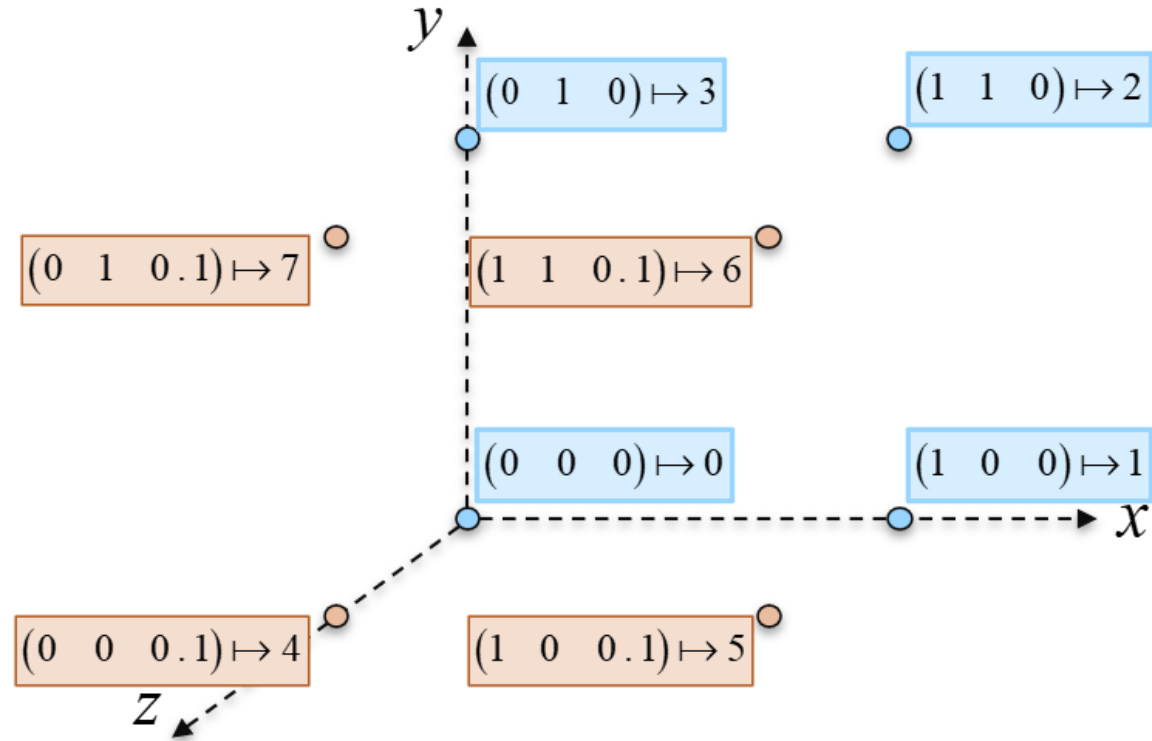
```
../case01/system/blockMeshDict
17. scale 0.1;
18.
19. vertices
20. (
21.     (0 0 0) // id: 0
22.     (1 0 0) // id: 1
23.     (1 1 0) // id: 2
24.     (0 1 0) // id: 3
25.     (0 0 0.1) // id: 4
26.     (1 0 0.1) // id: 5
27.     (1 1 0.1) // id: 6
28.     (0 1 0.1) // id: 7
29. );
```



blockMesh – Case 21 (cavity)

- Generating Vertices

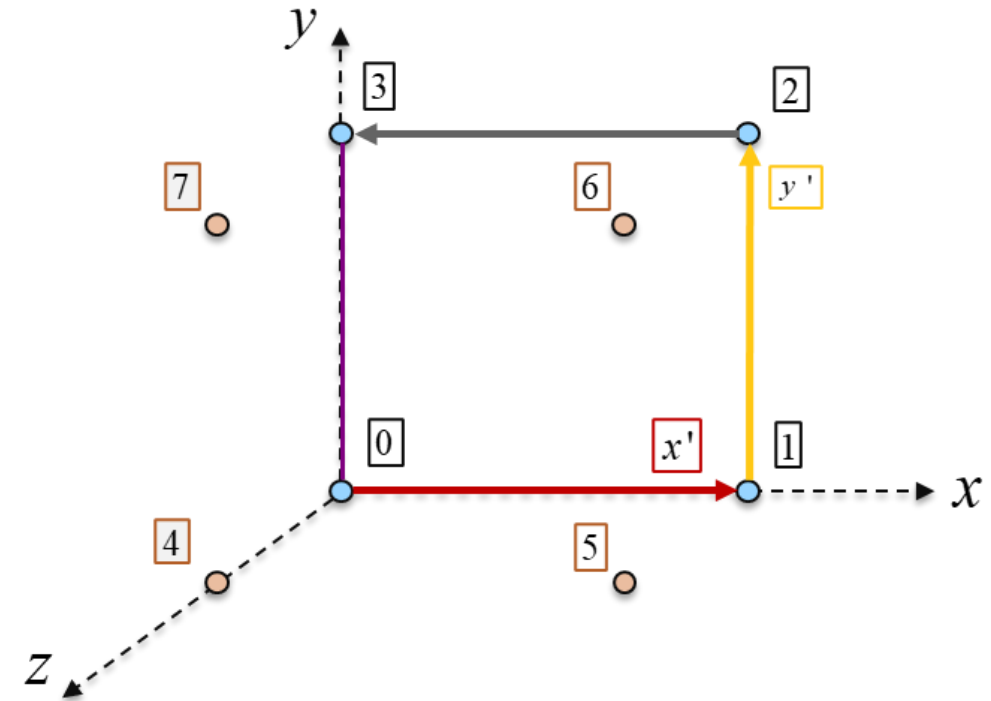
```
../case01/system/blockMeshDict
17. scale 0.1;
18.
19. vertices
20. (
21.   (0 0 0) // id: 0
22.   (1 0 0) // id: 1
23.   (1 1 0) // id: 2
24.   (0 1 0) // id: 3
25.   (0 0 0.1) // id: 4
26.   (1 0 0.1) // id: 5
27.   (1 1 0.1) // id: 6
28.   (0 1 0.1) // id: 7
29. );
```



blockMesh – Case 21 (cavity)

- Generating Vertices

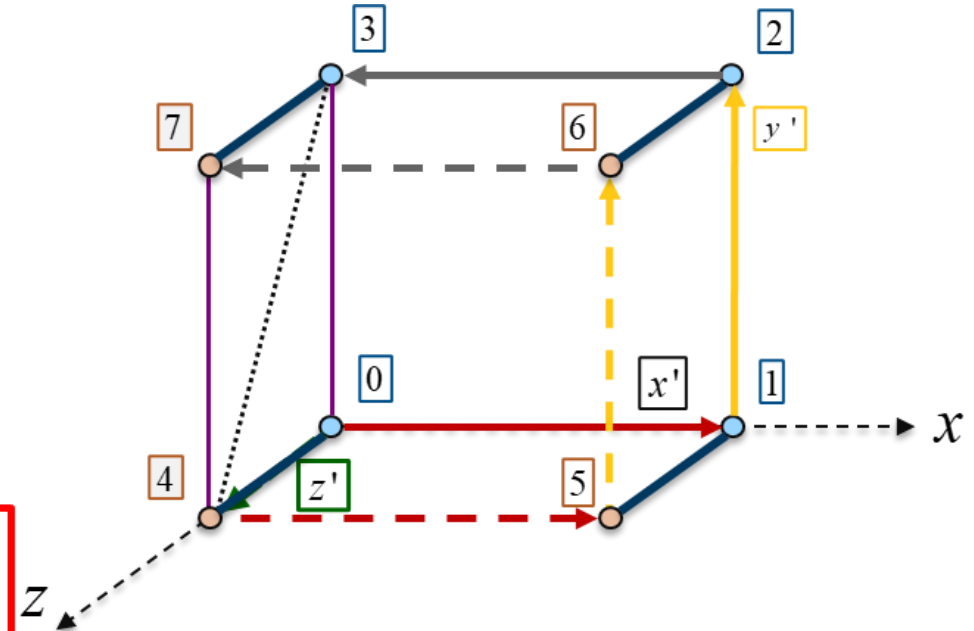
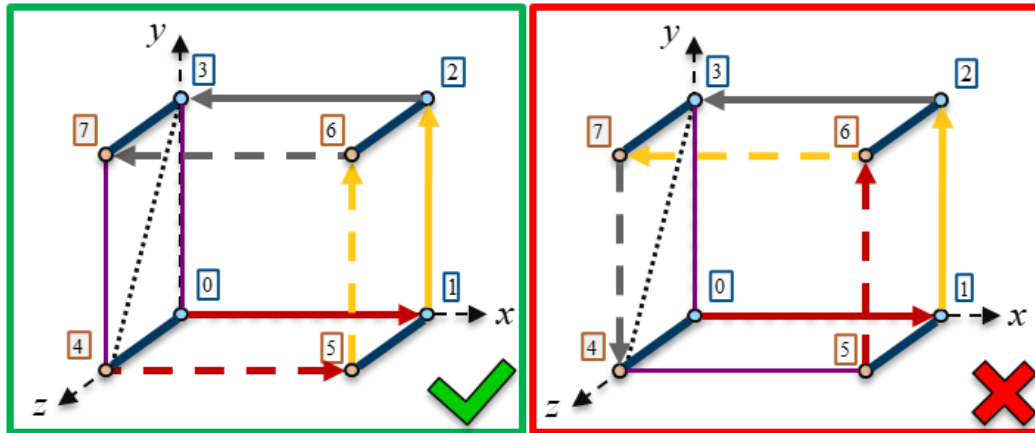
```
../case01/system/blockMeshDict
31. blocks
32. (
33.   hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
34. );
```



blockMesh – Case 21 (cavity)

- Generating Vertices

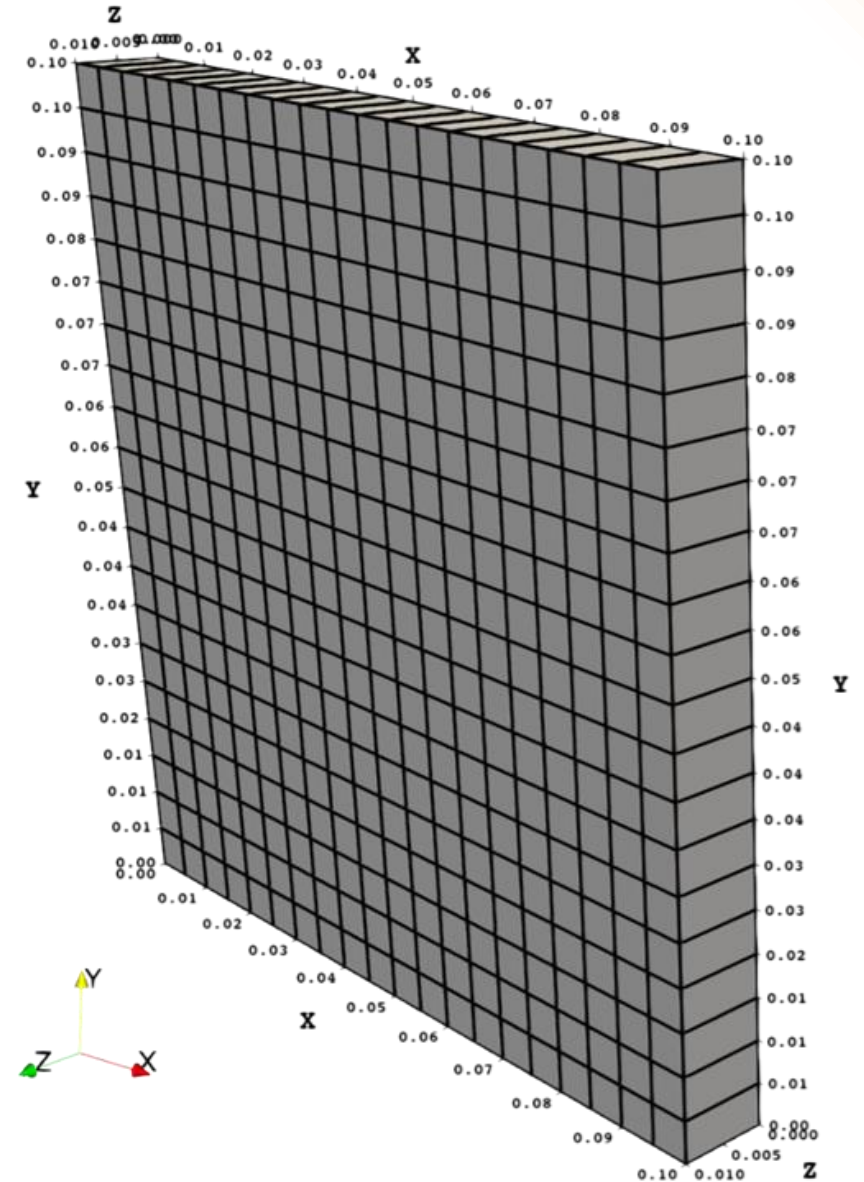
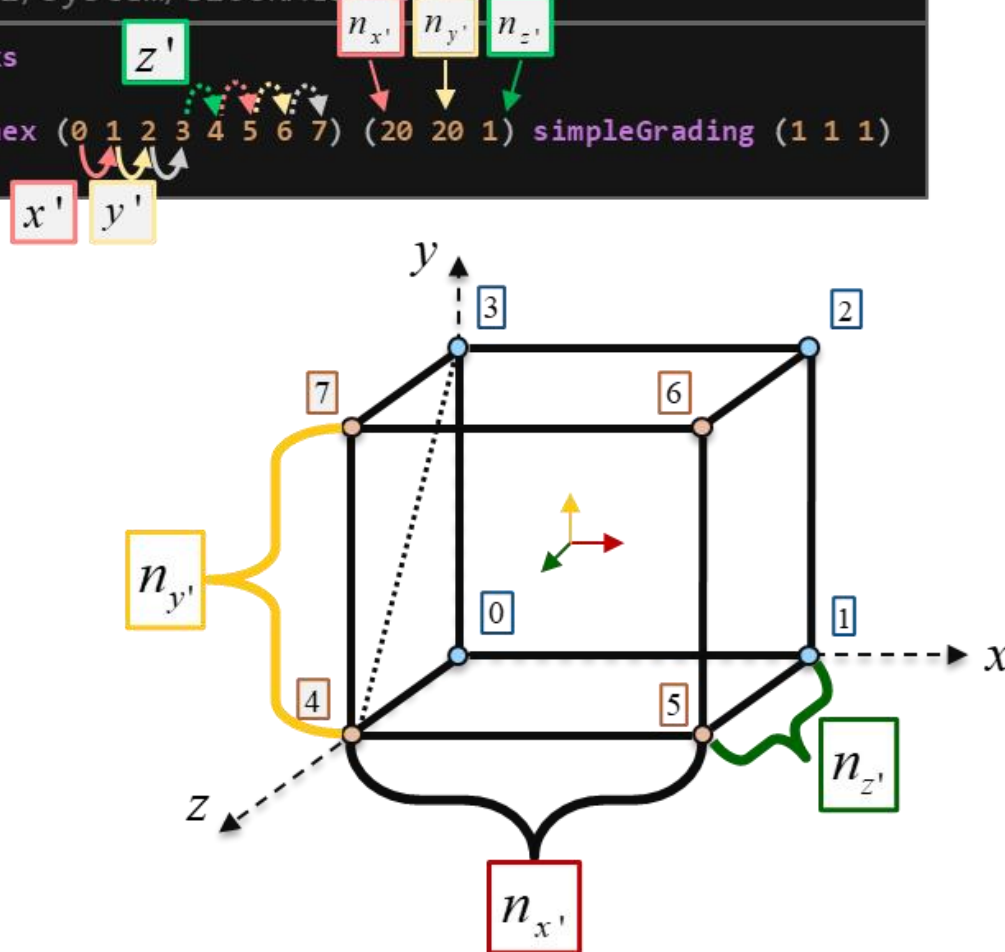
```
../case01/system/blockMeshDict  
  
31. blocks  
32. (  
33.     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)  
34. );
```



blockMesh – Case 21 (cavity)

- 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. );
```

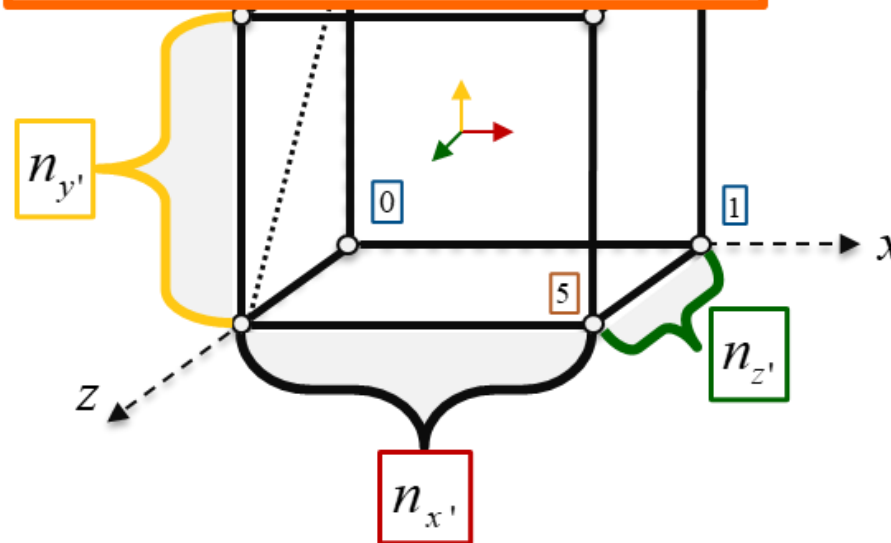


blockMesh – Case 21 (cavity)

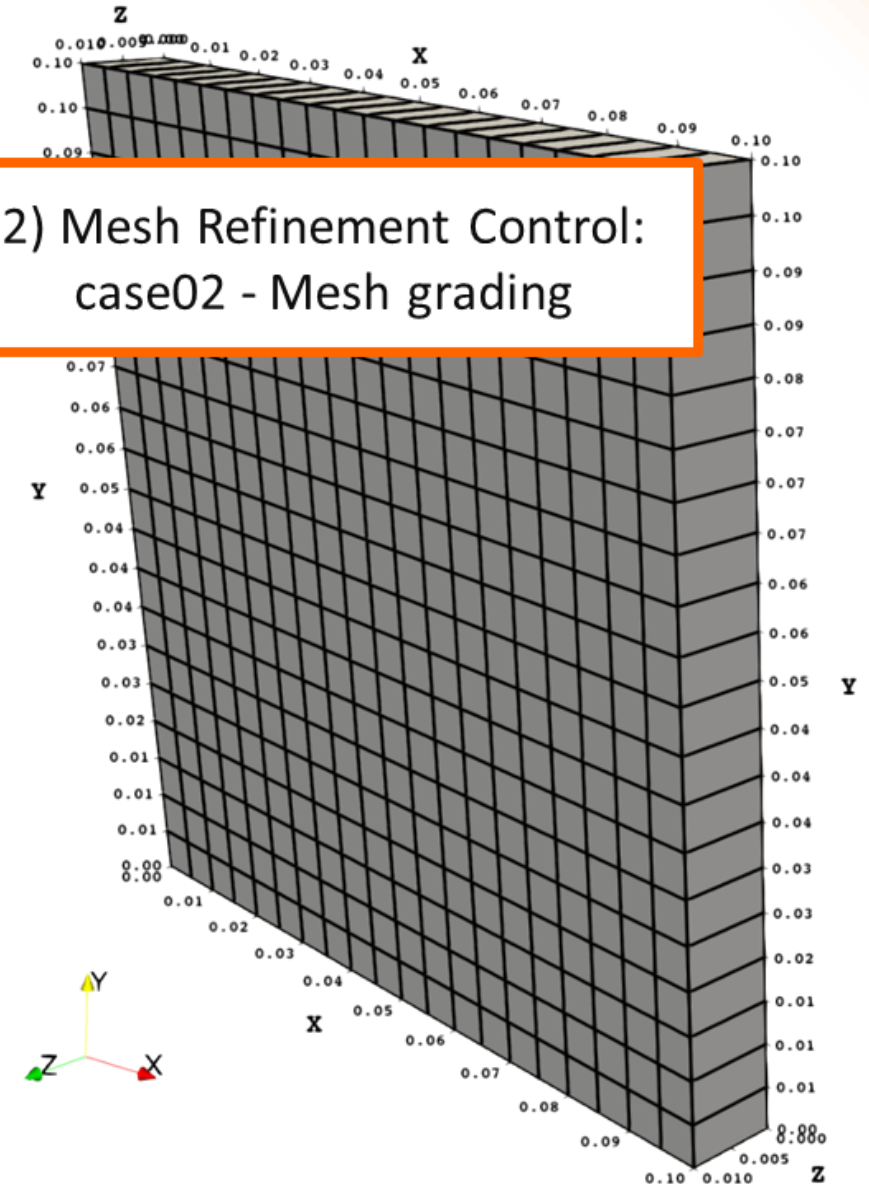
- Extra Functionalities

```
../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. );
```

4) Curved Surfaces: Designing curvilinear regions



2) Mesh Refinement Control:
case02 - Mesh grading



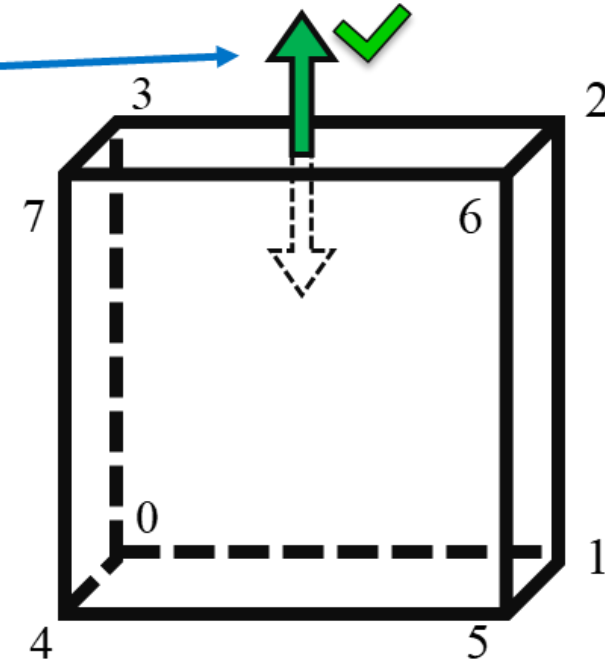
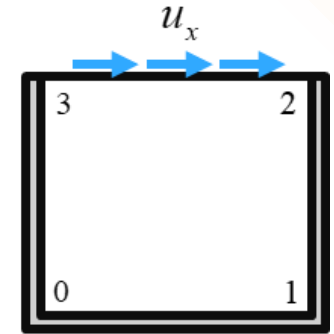
blockMesh – Case 21 (cavity)

- 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



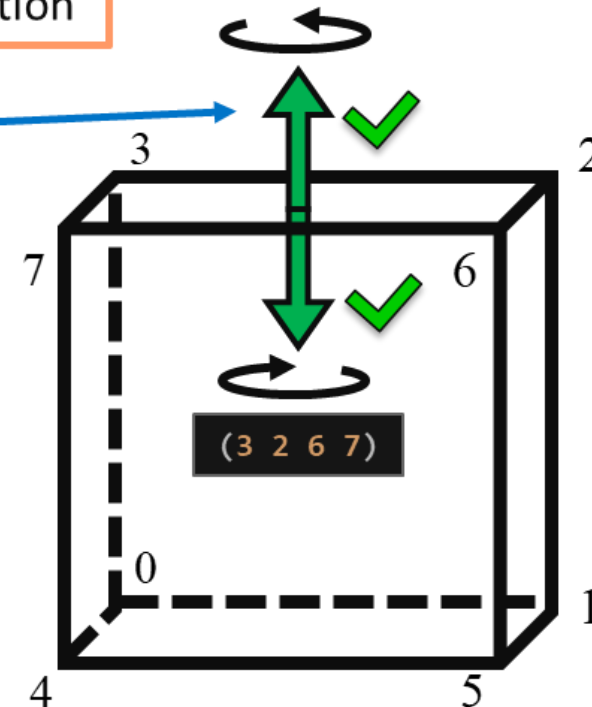
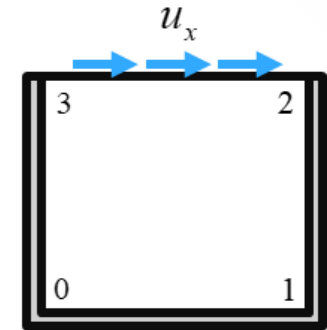
blockMesh – Case 21 (cavity)

- 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)

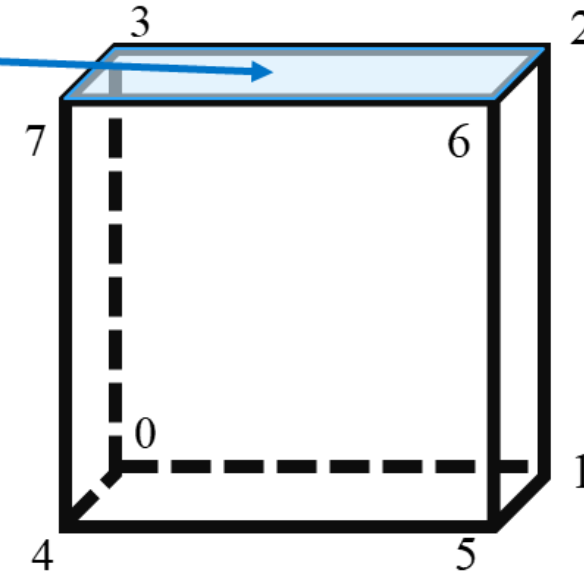
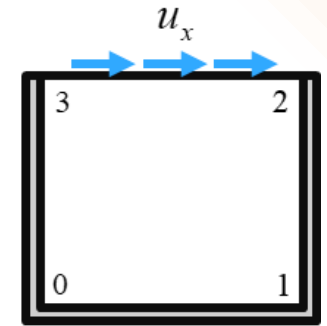
Base type boundary condition



blockMesh – Case 21 (cavity)

- Naming and Predefining Boundaries

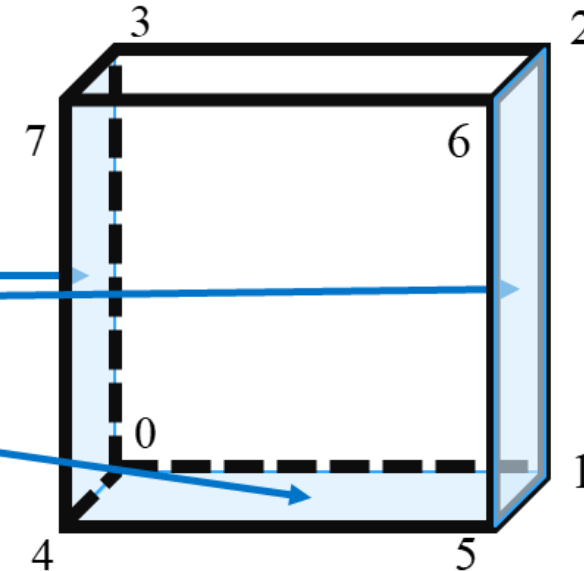
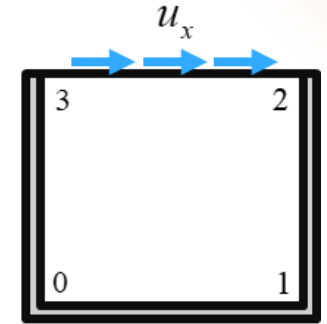
```
../case01/system/blockMeshDict
17. boundary
18. (
19.     movingWall
20.     {
21.         type wall;
22.         faces
23.         (
24.             (3 7 6 2)
25.         );
26.     }
27.     fixedWalls
28.     {
29.         type wall;
30.         faces
31.         (
32.             (0 4 7 3)
33.             (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 21 (cavity)

- Naming and Predefining Boundaries

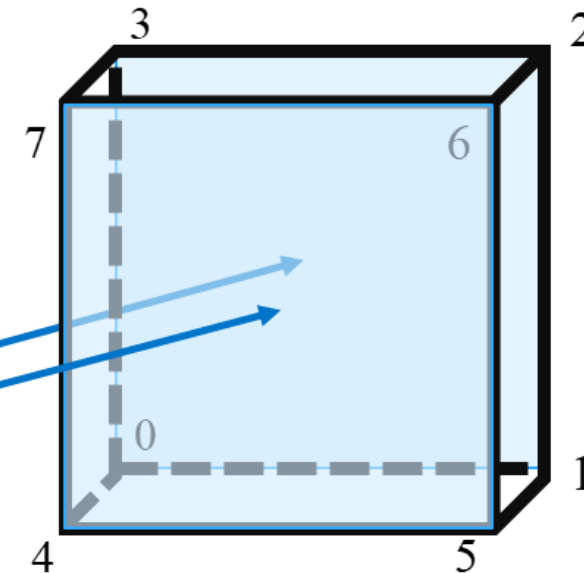
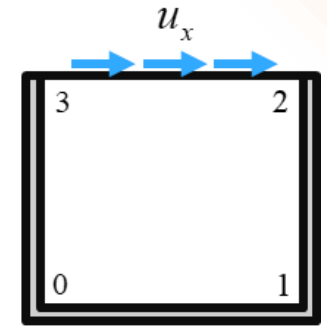
```
../case01/system/blockMeshDict
17. boundary
18. (
19.     movingWall
20.     {
21.         type wall;
22.         faces
23.         (
24.             (3 7 6 2)
25.         );
26.     }
27.     fixedWalls
28.     {
29.         type wall;
30.         faces
31.         (
32.             (0 4 7 3)
33.             (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 21 (cavity)

- 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 21 (cavity)

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 21 (icoFoam - Mesh Grading)

1. `>> run`
2. `>> cd case21/grading/`
3. Visualize blockMeshDict file in VSCode



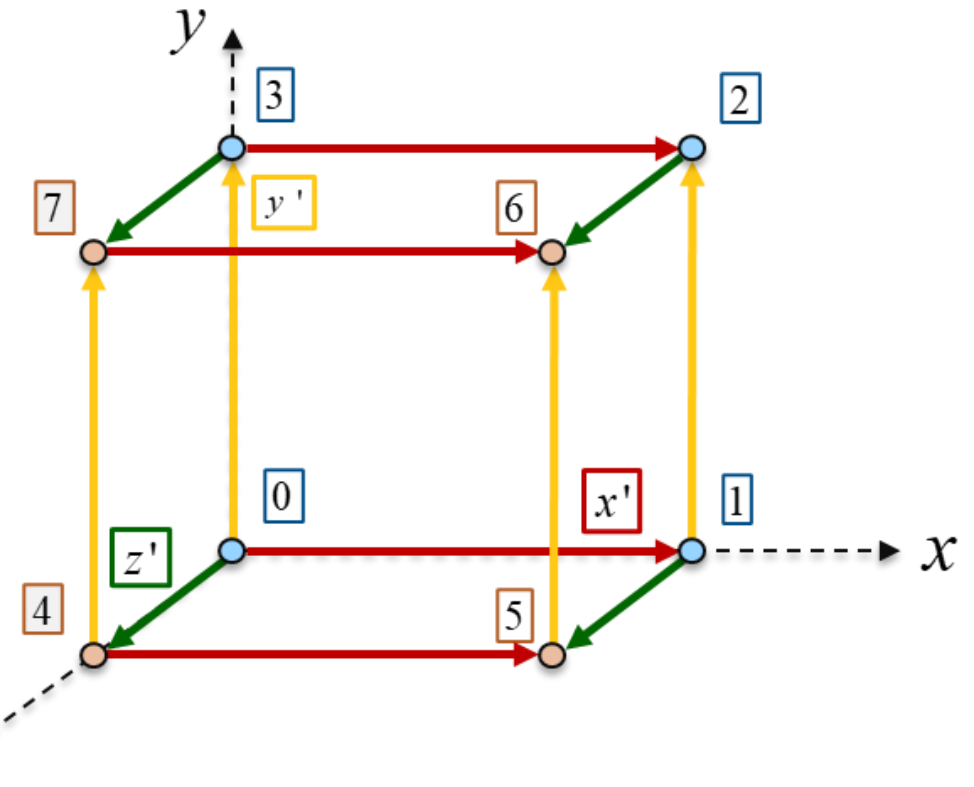
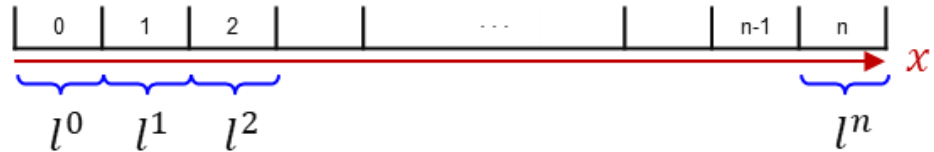
blockMesh – Case 21 (icoFoam - Mesh Grading)

- Mesh grading for stretching the mesh towards one or more Planes.
- icoFoam 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. );
```

simpleGrading = edges' cells expansion ratios

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



blockMesh – Case 21 (icoFoam - Mesh Grading)

```
../case02/system/blockMeshDict

31. blocks
32. (
33.     ...
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
```

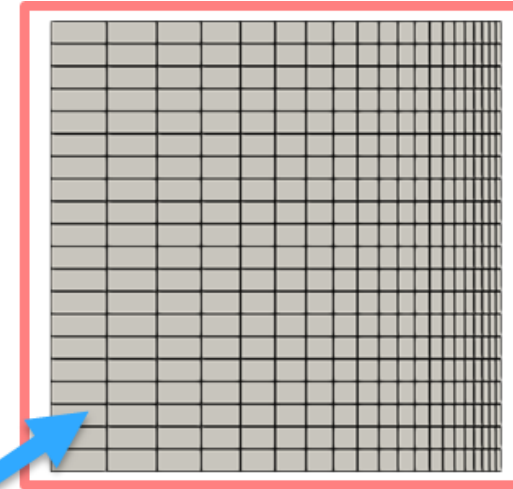


Fig. E

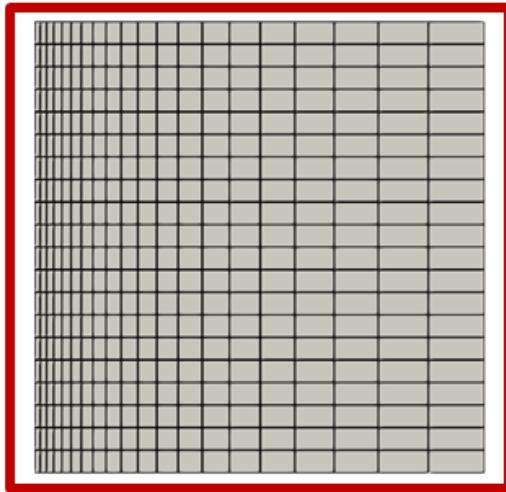


Fig. A

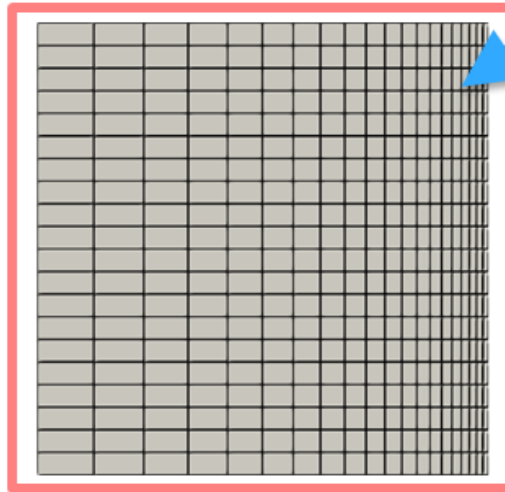


Fig. B

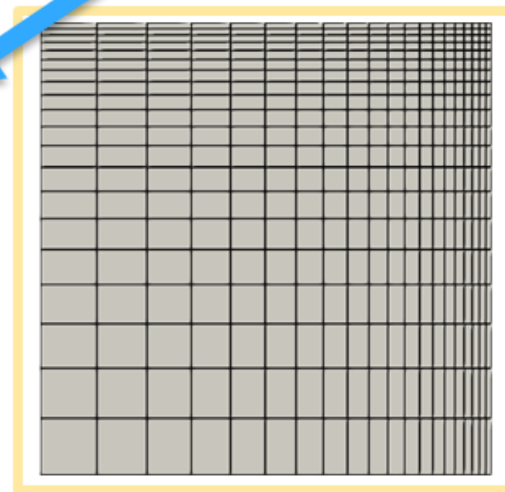


Fig. C

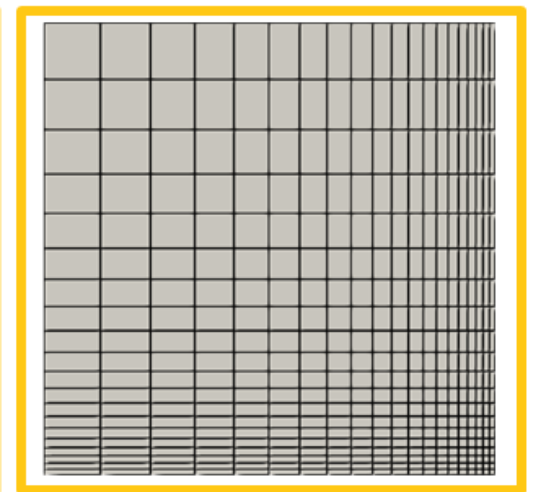


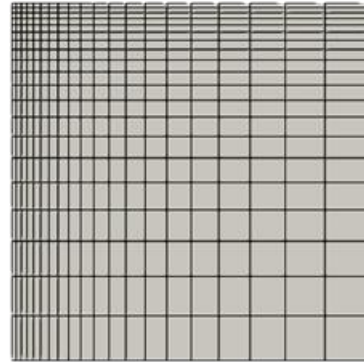
Fig. D

blockMesh – Case 21 (icoFoam - Mesh Grading)

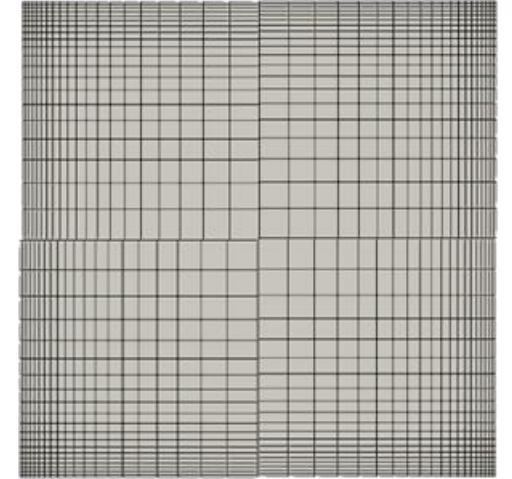
```
../case02/system/blockMeshDict
```

```
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. )
```

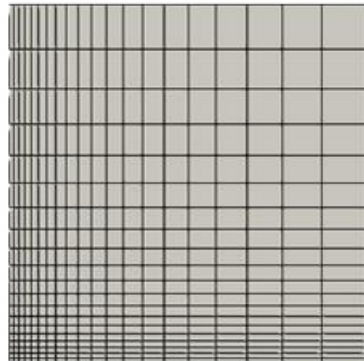
Block 0



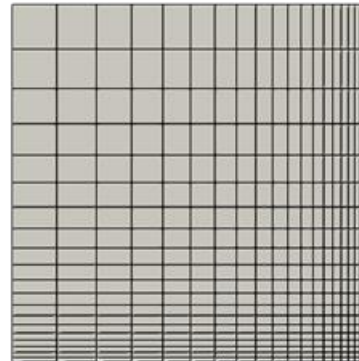
Block 1



Block 2



Block 3



blockMesh – Case 21 (cavity)

1. >> blockMesh
2. Check results in Paraview

Hands on (change and visualize)

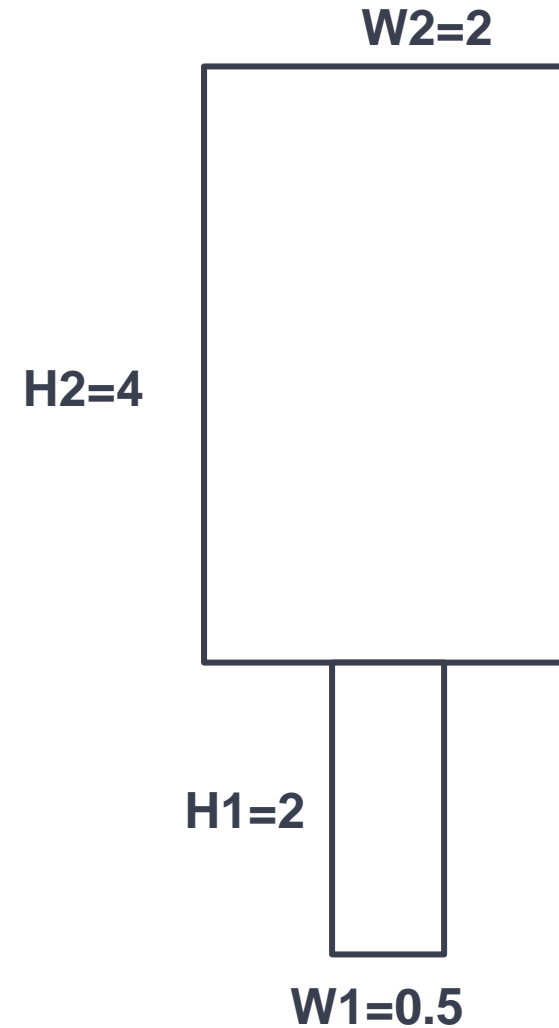
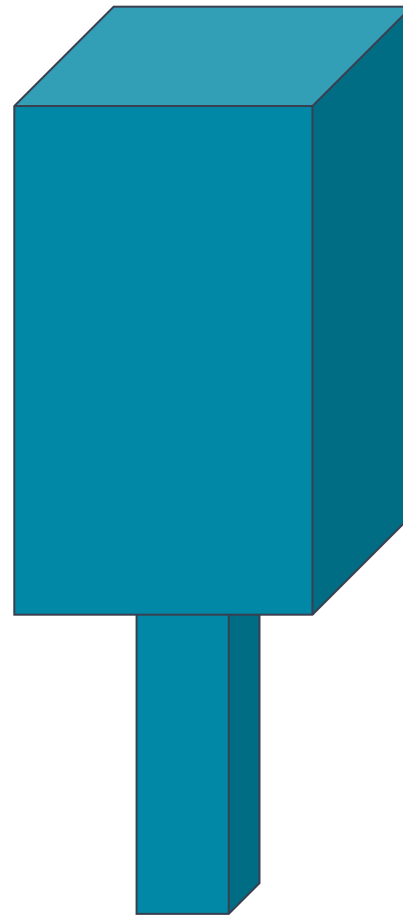
1. Test the different mesh grading configurations



blockMesh – Case 22 (General)

Geometry

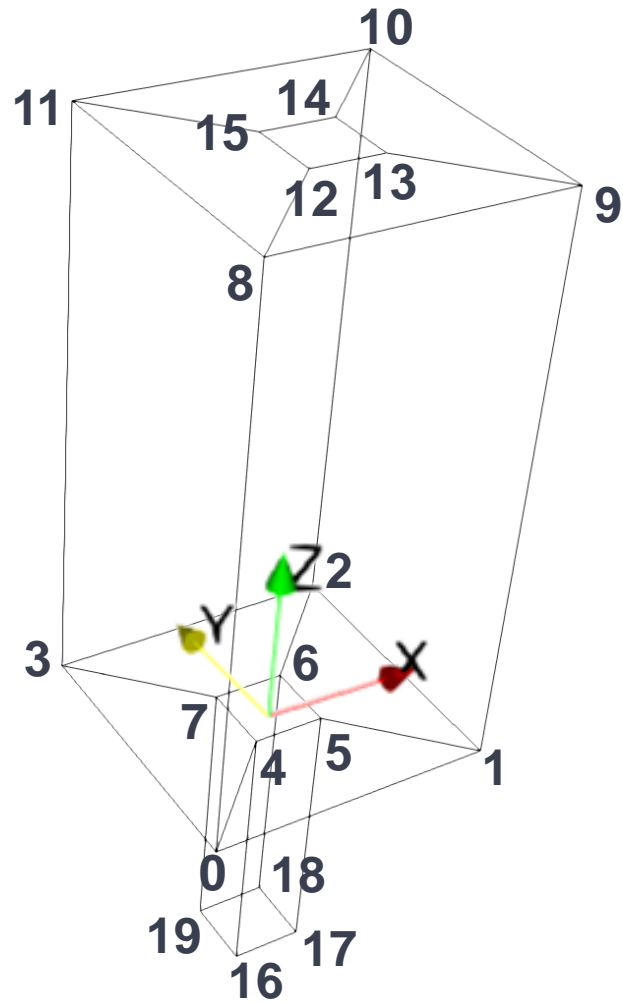
Dimensions (in m)



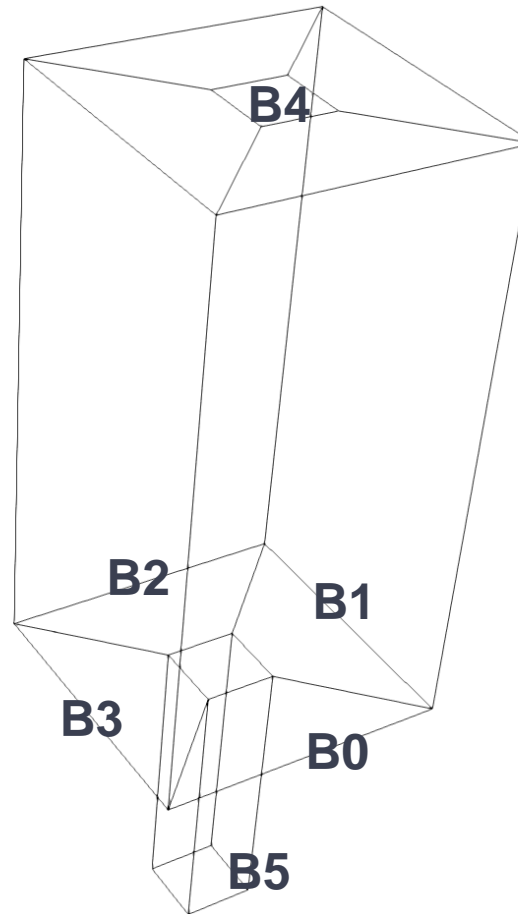
blockMesh – Case 22 (General)

Geometry

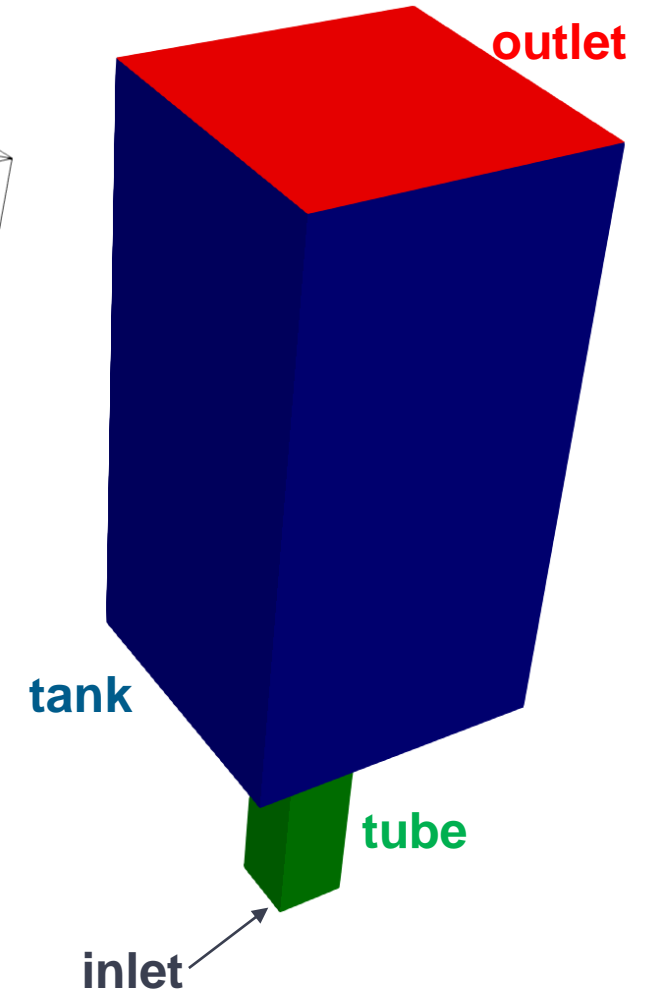
Vertices



Blocks



Patches



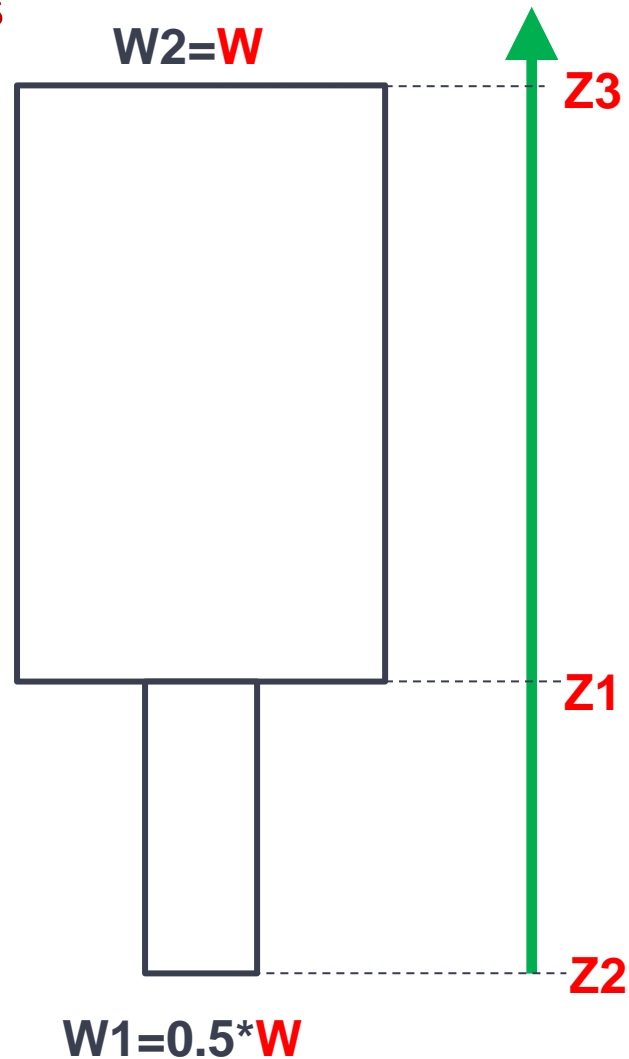
blockMesh – Case 22 (General)

1. `>> run`
2. `>> cd case22/general`
3. Visualize blockMeshDict file in VSCode
4. `>> blockMesh`
5. Visualize the Mesh in paraview

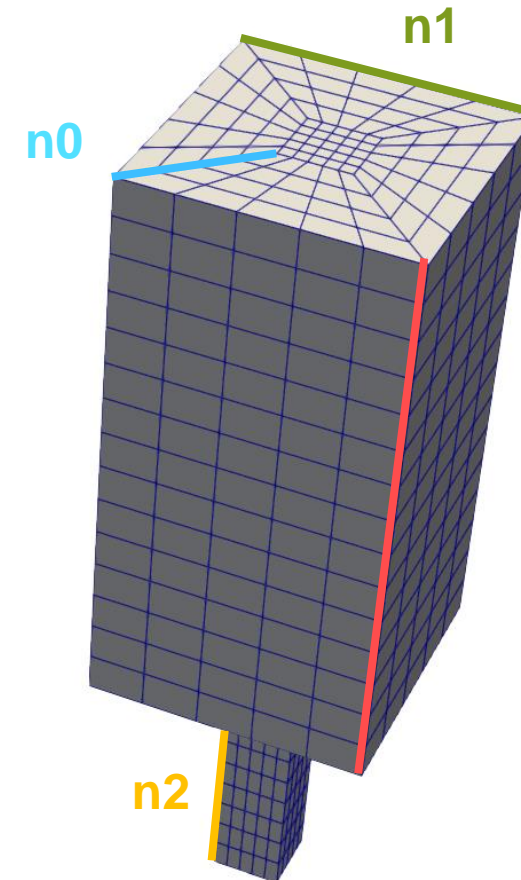


blockMesh – Case 22 (Parameterized) Geometry & Mesh

Dimensions



Number of cells



blockMesh – Case 22 (Parameterized)

1. `>> run`
2. `>> cd case22/parameterized/`
3. Visualize blockMeshDict file in VSCode



blockMesh – Case 22 (Parameterized)

- Parameterized Geometry and Mesh Generation

```
//Parameters
```

```
W 2;
```

```
Z1 0;
```

```
Z2 -2.0;
```

```
Z3 4.0;
```

```
n0 4;
```

```
n1 5;
```

```
n2 10;
```

```
n3 15;
```

```
// Calculations
```

```
X0 #calc "-0.5*$W";
```

```
Y0 #calc "1*$X0";
```

```
X1 #calc "-1*$X0";
```

```
Y1 #calc "1*$X0";
```

```
X2 #calc "-1*$X0";
```

```
Y2 #calc "-1*$X0";
```

```
X3 #calc "1*$X0";
```

```
Y3 #calc "-1*$Y0";
```

```
X4 #calc "0.25*$X0";
```

```
Y4 #calc "0.25*$X0";
```

```
X5 #calc "-0.25*$X0";
```

```
Y5 #calc "0.25*$X0";
```

```
X6 #calc "-0.25*$X0";
```

```
Y6 #calc "-0.25*$X0";
```

```
X7 #calc "0.25*$X0";
```

```
Y7 #calc "-0.25*$X0";
```



blockMesh – Case 22 (Parameterized)

- Parameterized Geometry and Mesh Generation

```
vertices
(
  (-1 -1 0) //0
  (1 -1 0) //1
  (1 1 0) //2
  (-1 1 0) //3
  (-0.25 -0.25 0) //4
  (0.25 -0.25 0) //5
  (0.25 0.25 0) //6
  (-0.25 0.25 0) //7
  (-1 -1 4) //8
  (1 -1 4) //9
  (1 1 4) //10
  (-1 1 4) //11
  (-0.25 -0.25 4) //12
  (0.25 -0.25 4) //13
  (0.25 0.25 4) //14
  (-0.25 0.25 4) //15
  (-0.25 -0.25 -2) //16
  (0.25 -0.25 -2) //17
  (0.25 0.25 -2) //18
  (-0.25 0.25 -2) //19
)
```

Normal Configuration of blockMeshDict

```
($X0 $Y0 $Z1) //0
($X1 $Y1 $Z1) //1
($X2 $Y2 $Z1) //2
($X3 $Y3 $Z1) //3
($X4 $Y4 $Z1) //4
($X5 $Y5 $Z1) //5
($X6 $Y6 $Z1) //6
($X7 $Y7 $Z1) //7

($X0 $Y0 $Z3) //8
($X1 $Y1 $Z3) //9
($X2 $Y2 $Z3) //10
($X3 $Y3 $Z3) //11
($X4 $Y4 $Z3) //12
($X5 $Y5 $Z3) //13
($X6 $Y6 $Z3) //14
($X7 $Y7 $Z3) //15

($X4 $Y4 $Z2) //16
($X5 $Y5 $Z2) //17
($X6 $Y6 $Z2) //18
($X7 $Y7 $Z2) //19
```

Parameterized Configuration of blockMeshDict



blockMesh – Case 22 (Parameterized)

- Parameterized Geometry and Mesh Generation

```
blocks
(
    hex (0 1 5 4 8 9 13 12) (5 4 15) simpleGrading (1 1 1) //B0
    hex (5 1 2 6 13 9 10 14) (4 5 15) simpleGrading (1 1 1) //B1
    hex (7 6 2 3 15 14 10 11) (5 4 15) simpleGrading (1 1 1) //B2
    hex (0 4 7 3 8 12 15 11) (4 5 15) simpleGrading (1 1 1) //B3
    hex (4 5 6 7 12 13 14 15) (5 5 15) simpleGrading (1 1 1) //B4
    hex (16 17 18 19 4 5 6 7) (5 5 10) simpleGrading (1 1 1) //B5
)
```

```
blocks
(
    hex (0 1 5 4 8 9 13 12) ($n1 $n0 $n3) simpleGrading (1 1 1) //B0
    hex (5 1 2 6 13 9 10 14) ($n0 $n1 $n3) simpleGrading (1 1 1) //B1
    hex (7 6 2 3 15 14 10 11) ($n1 $n0 $n3) simpleGrading (1 1 1) //B2
    hex (0 4 7 3 8 12 15 11) ($n0 $n1 $n3) simpleGrading (1 1 1) //B3
    hex (4 5 6 7 12 13 14 15) ($n1 $n1 $n3) simpleGrading (1 1 1) //B4
    hex (16 17 18 19 4 5 6 7) ($n1 $n1 $n2) simpleGrading (1 1 1) //B5
);
```

```
n0 4;
n1 5;
n2 10;
n3 15;
```



blockMesh – Case 22 (Parameterized)

- Parameterized Geometry and Mesh Generation

No changes for the boundaries

```
boundary
(
    inlet
    {
        type patch;
        faces
        (
            (16 17 18 19)
        );
    }
    outlet
    {
        type patch;
        faces
        (
            (8 9 13 12)
            (13 9 10 14)
            (15 14 10 11)
            (8 12 15 11)
            (12 13 14 15)
        );
    }
    tube
    {
        type wall;
        faces
        (
            (16 17 5 4)
            (17 18 6 5)
            (19 7 6 18)
            (19 16 4 7)
        );
    }
    tank
    {
        type wall;
        faces
        (
            (0 1 5 4)
            (5 1 2 6)
            (7 6 2 3)
            (0 4 7 3)
            (0 1 9 8)
            (1 2 10 9)
            (2 3 11 10)
            (3 0 8 11)
        );
    }
);
```



blockMesh – Case 22 (Parameterized)

1. >> blockMesh
2. Visualize the Mesh in paraview

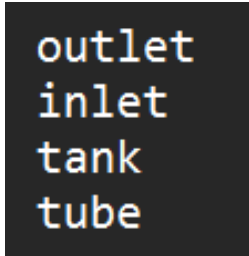
Hands on (change and visualize)

1. Change the parameters and check the result in paraview



cfMesh – Case 22 (STL based Meshing)

1. `>> run`
2. `>> cd case22/cfMesh/`
3. `>> code .`
4. `>> cd STLBuilder`
5. In VSCode check file myList contents
6. Open all the individual stl files in paraview
7. `>> ./joinSTL.x`
8. Open the total.stl file in paraview
9. `>> cp total.fms ../ ## copy total.fms to the problem root folder`
10. `>> cd .. ##return to the problem root folder`
11. In VSCode check file system/meshDict contents



outlet
inlet
tank
tube



cfMesh – Case 22 (STL based Meshing)

```
surfaceFile "total.fms";
```

→ The file created from STLs

```
maxCellSize 0.1;
```

→ Maximum Cell size

```
//boundaryCellSize 0.8;
```

```
//minCellSize 0.01;
```

```
/*localRefinement
```

```
{  
  inlet  
  {  
    cellSize 0.05;  
    //additionalRefinementLevels 1;  
    refinementThickness 0.01;  
  }  
}
```

→ Refine part of the domain near a patch

→ For refining part of the domain
inside a box

```
/*objectRefinements
```

```
{  
  Box  
  {  
    type      box;  
    cellSize  0.05;  
    centre    (0 0 -1);  
    lengthX   0.5;  
    lengthY   0.5;  
    lengthZ   2;  
  }  
}
```

```
*/
```

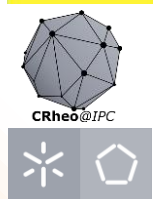


cfMesh – Case 22 (STL based Meshing)

1. >> run #Return to the case root
2. >> cartesianMesh
3. Visualize the results in Paraview

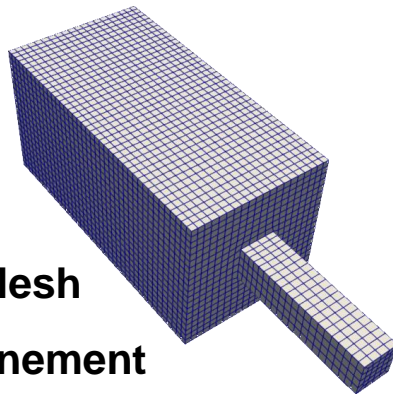
Hands on (change and visualize)

1. Uncomment the 2 mesh refinement level methods, in meshDict file, and check the result in paraview
2. Change the parameters in meshDict file and check the results in paraview

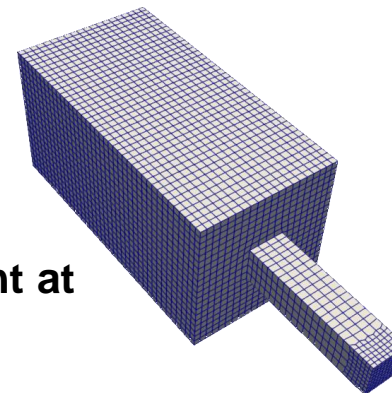


Coarse Mesh

without Refinement



**Local Refinement at
the Inlet**



Object Refinement as a box

