



Basic Meshing

A Comprehensive Introduction Guide



University of Minho
School of Engineering



INSTITUTE FOR
POLYMERS AND COMPOSITES



Computational
Rheology
@IPC

Wagner de Campos Galuppo



wagnergaluppo@gmail.com

PhD Researcher and Scholar @ IPC-UMinho



<http://www.ipc.uminho.pt/>



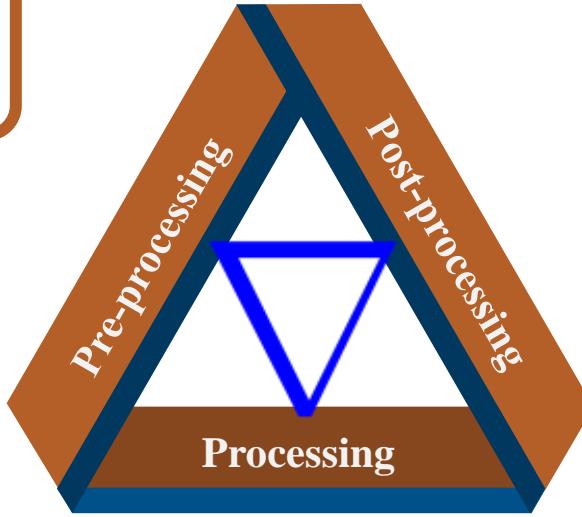
<https://crheo.org/>

Disclaimer

This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trademarks.

OpenFOAM®

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



```
(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)
...
```

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

```
5468697320697320616E
206578616D0706C652066
726F6D205761676E6572
2047616C7570706F206F
66206120686578616465
63696D616C20636F6465
20666F722B0696C6C7573
74726174696F6E207075
72626F736573
...

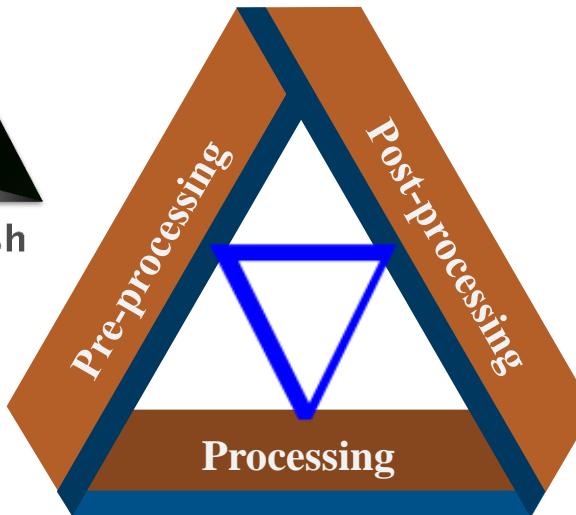
```

- 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

```
(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)
...

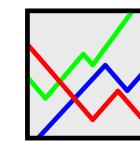
```

Open ∇ FOAM®



Open ∇ FOAM®

/// **ParaView**



Summary

1

blockMesh utility

case01 - icoFoam Tutorial Review

2

Mesh Refinement Control

case02 - Mesh grading

3

Multiblock Mesh

case03 - Blocks connections



CRheo@IPC



4

Curved Surfaces

case04 - Designing curvilinear regions

5

Case Study Mesh

case05 - Interconnected Course Mesh

6

Mesh Metrics

Quality control parameters

1

blockMesh utility

case01 - icoFoam Tutorial Review



CRheo@IPC



```
of2306 # Your alias set to load your OpenFOAM bin files..
```

<Desktop> Should be your user Windows desktop path location

```
mkdir -p <Desktop>/foamIberia # Check your Extracted files in here
```

```
cd <Desktop>/FOAM@Iberia-Courses/Beginner/C2/tutorials
```

```
cp -r $FOAM_TUTORIALS/incompressible/icoFoam/cavity/cavity ./
```

```
mv cavity case01
```

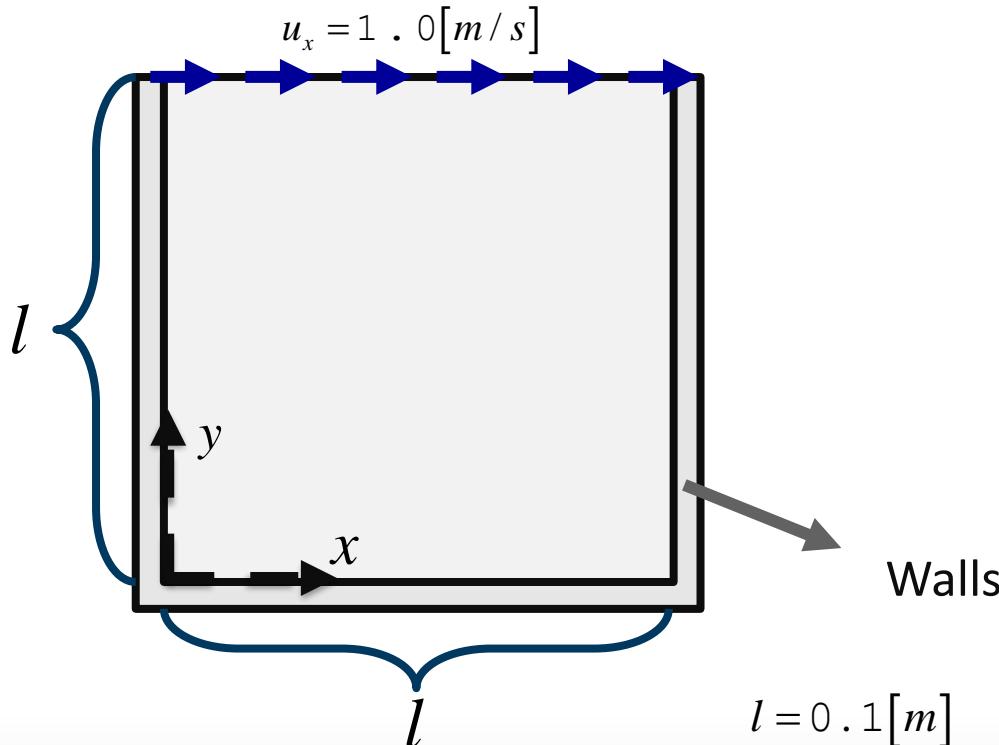
```
cd case01
```

```
tree -L 3
```

```
$ tree -L 3
.
├── 0
└── p
    └── u
        ├── constant
        │   └── transportProperties
        ├── system
        ├── blockMeshDict
        ├── controlDict
        ├── decomposeParDict
        ├── fvSchemes
        ├── fvSolution
        └── PDRblockMeshDict

3 directories, 9 files
```

- Physical Problem: (Cavity tutorial)



blockMesh utility

case01 - icoFoam Tutorial Review

code <Desktop>/FOAM@Iberia-Courses/Beginner/C2/tutorials/case01/system/blockMeshDict

```
..../basicMeshing/tutorials/case01/system/blockMeshDict
1.  /*----- C++ -----*/
2.  / ====== F ield      / OpenFOAM: The Open Source CFD Toolbox
3.  / \ \ / O peration   / Version: v2306
4.  / \ \ / A nd         / Website: www.openfoam.com
5.  / \ \ \ M anipulation /
6.  \*-----*/
7. 
8.  FoamFile
9.  {
10.    version    2.0;
11.    format      ascii;
12.    class       dictionary;
13.    object     blockMeshDict;
14. }
15. // * * * * *
16. 
17. // convertToMeters 0.1;
18. scale 0.1;
19. 
20. vertices
21. (
22.   (0 0 0)
23.   (1 0 0)
24.   (1 1 0)
25.   (0 1 0)
26.   (0 0 0.1)
27.   (1 0 0.1)
28.   (1 1 0.1)
29.   (0 1 0.1)
30. );
```

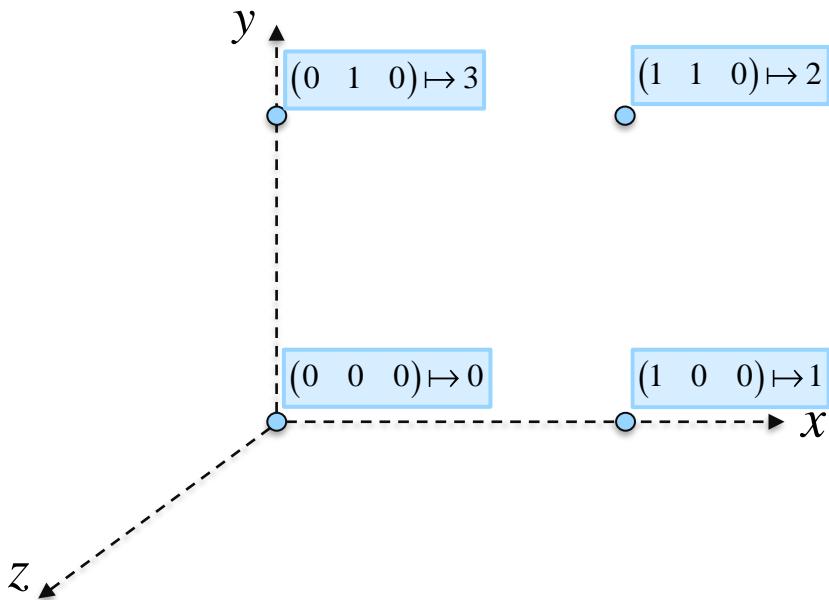


CRheo@IPC



- 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. );
```

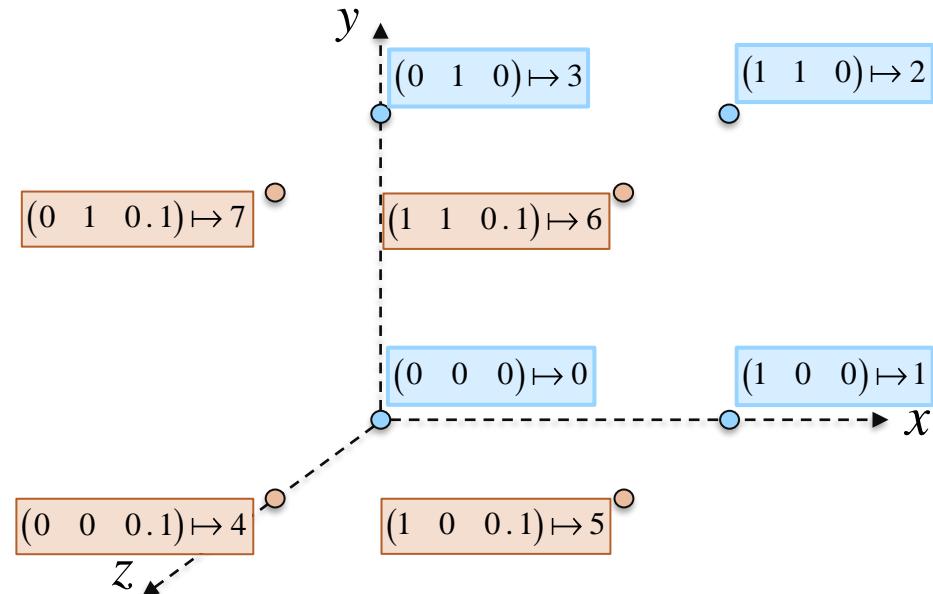


CRheo@IPC



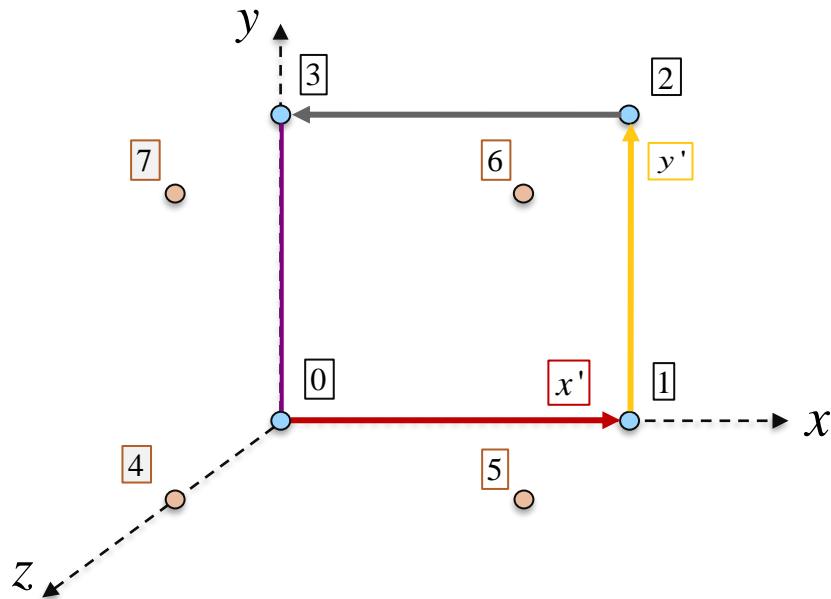
- Generating Vertices

```
./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. };
```



- Connecting 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. );
```



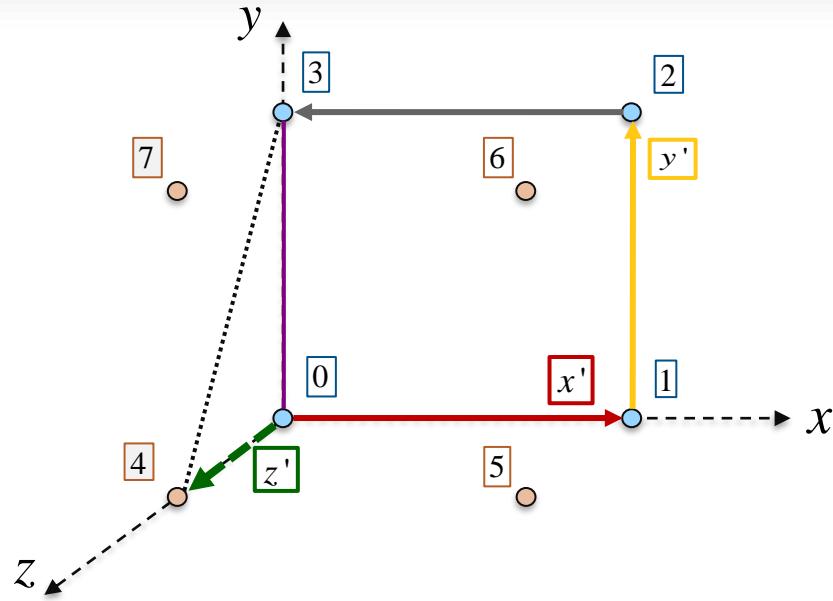
CRheo@IPC



- Connecting 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. );
```

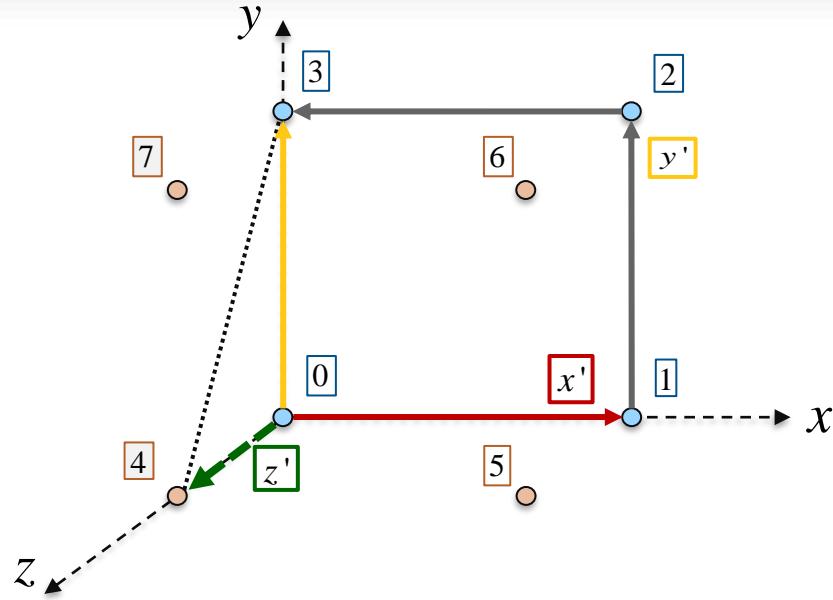


CRheo@IPC



- Connecting 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. );
```



CRheo@IPC

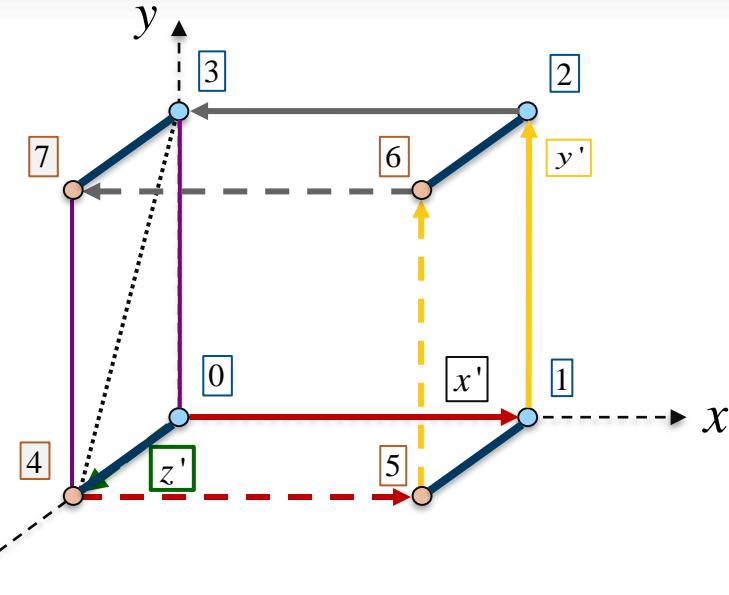
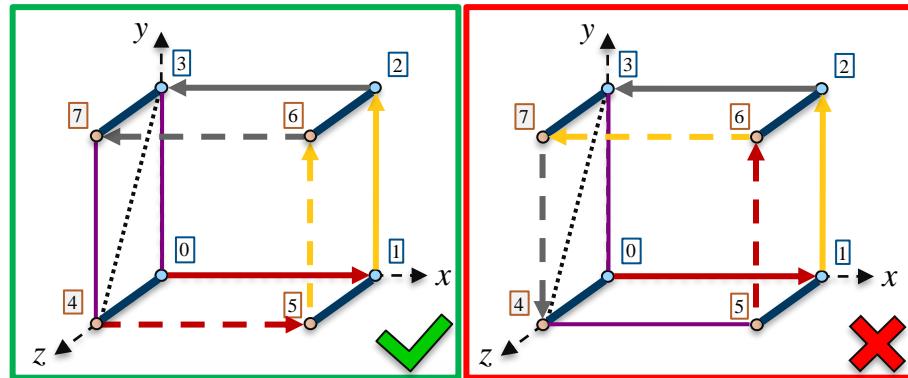


blockMesh utility

case01 - icoFoam Tutorial Review

- Connecting 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. );
```

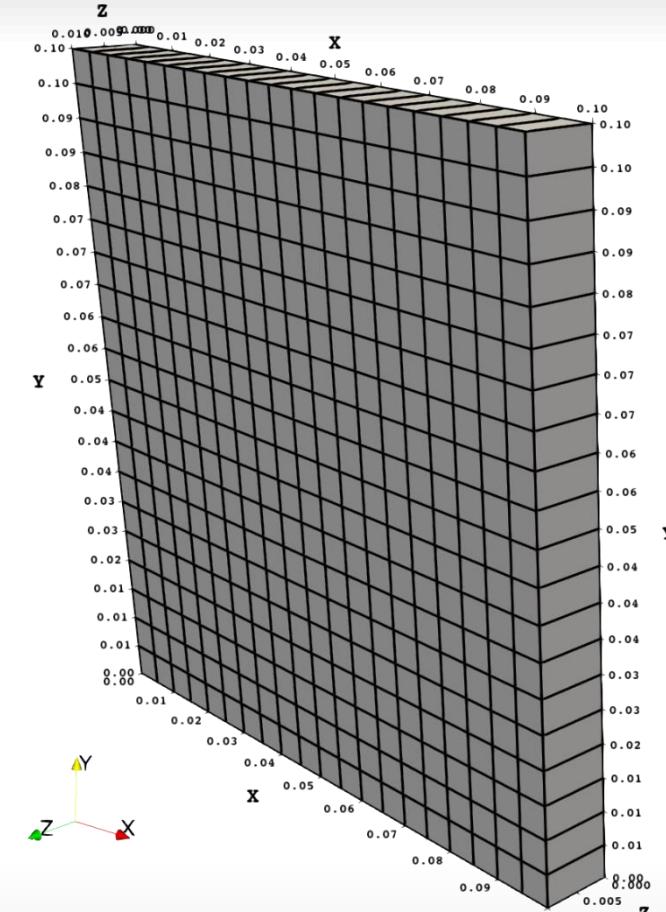
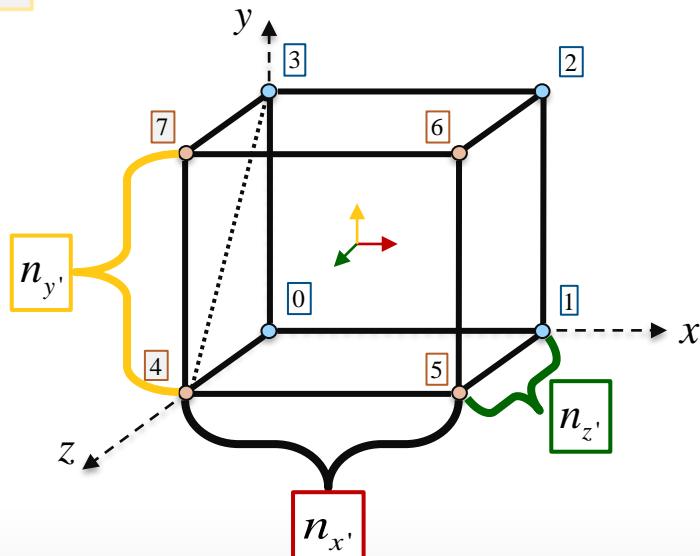


blockMesh utility

case01 - icoFoam Tutorial Review

- 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. );  
      x' y'  
      z'
```



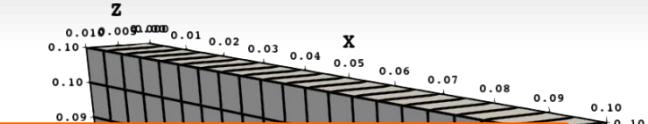
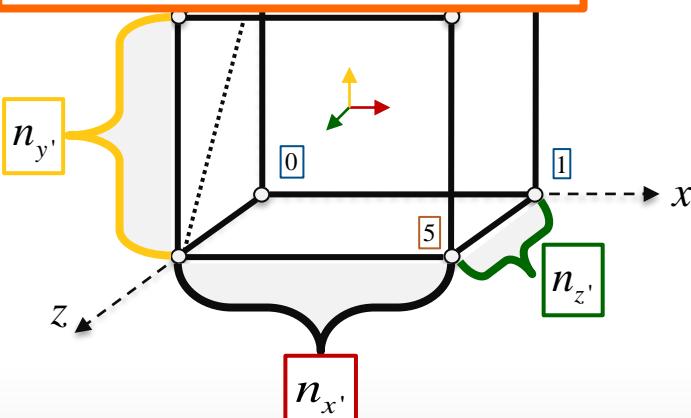
blockMesh utility

case01 - icoFoam Tutorial Review

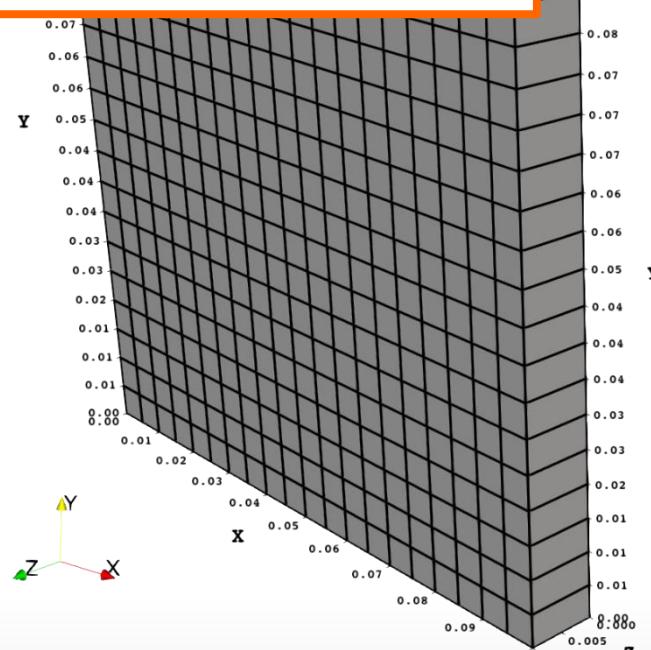
- Extra functionalities

```
..//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 utility

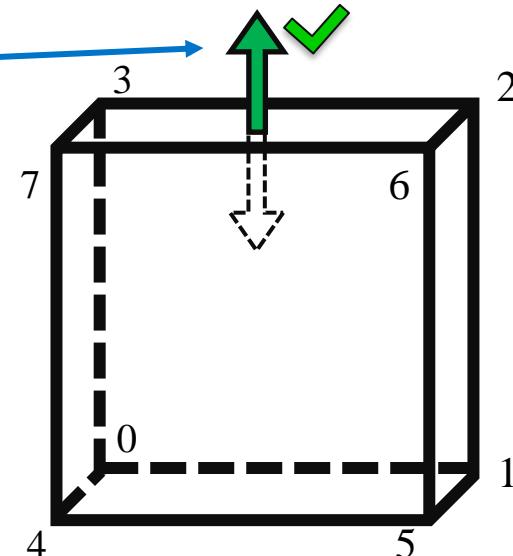
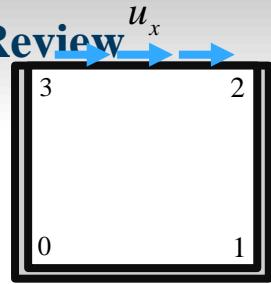
case01 - icoFoam Tutorial Review

- 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 utility

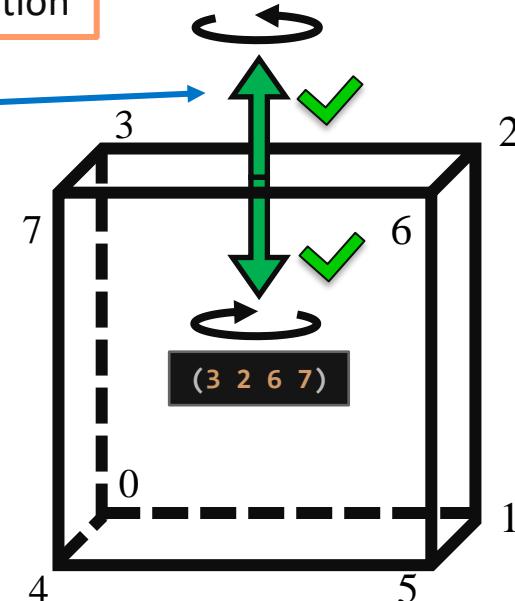
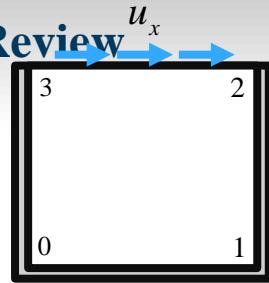
case01 - icoFoam Tutorial Review

- 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 utility

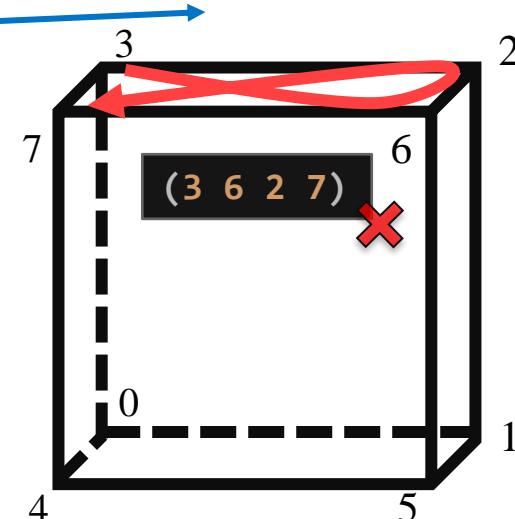
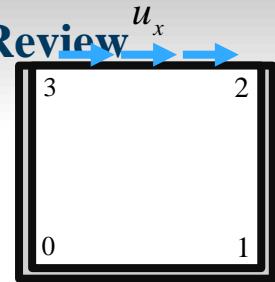
case01 - icoFoam Tutorial Review

- 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

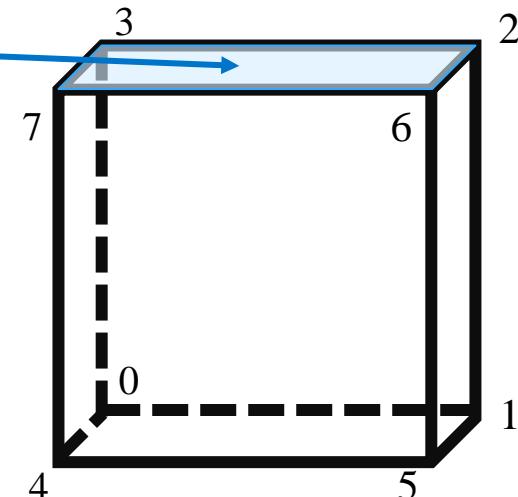
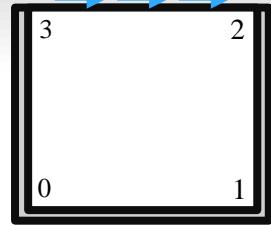


blockMesh utility

- 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. };
```

case01 - icoFoam Tutorial Review u_x

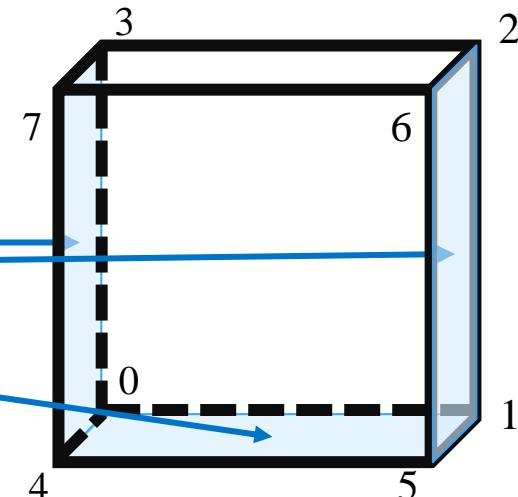


blockMesh utility

case01 - icoFoam Tutorial Review

- 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. };
```

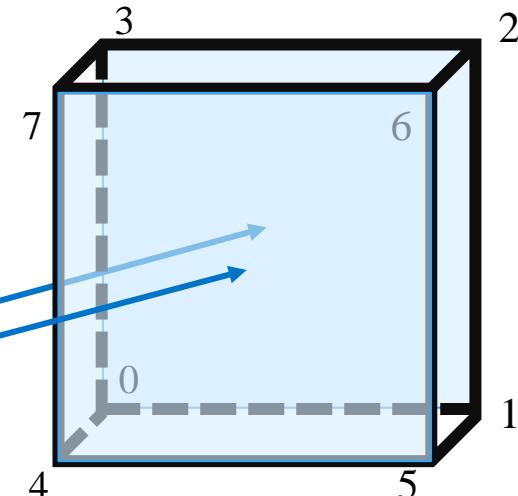
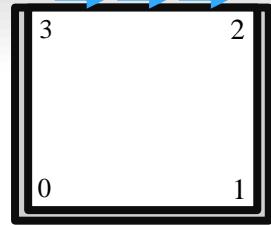


blockMesh utility

- 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. }
```

case01 - icoFoam Tutorial Review



2

Mesh Refinement Control

case02 - Mesh grading

```
cd <Desktop>/FOAM@Iberia-Courses/Beginner/C2/tutorials  
cp -r ./caseTutorials/case02 ./  
cd case02  
code .
```



CRheo@IPC



Mesh Refinement Control

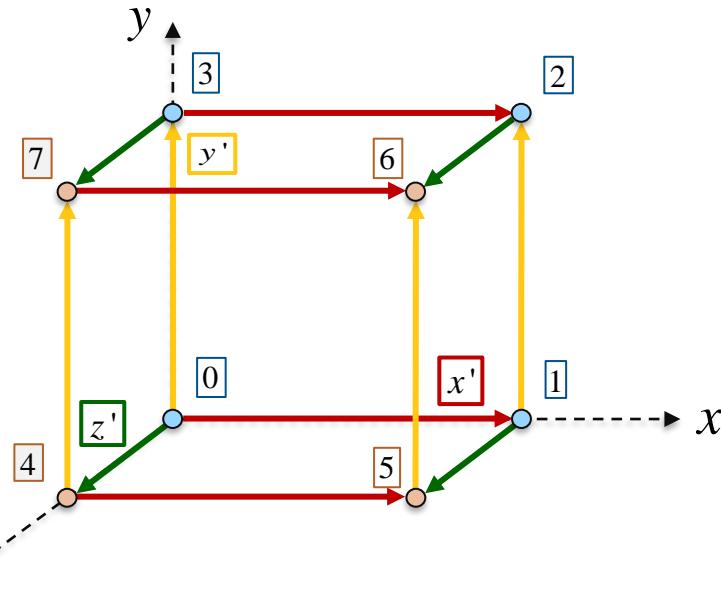
case02 - Mesh grading

- 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

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



Mesh Refinement Control

case02 - Mesh grading

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

```
31. blocks
32. (
33.
34.     ...
35.     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 1.0 1.0 1.0 ) // original
36.     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (10.0 1.0 1.0 ) // Fig. A
37.     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 0.1 1.0 1.0 ) // Fig. B
38.     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 0.1 0.1 1.0 ) // Fig. C
39.     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 0.1 10.0 1.0 ) // Fig. D
40.
41.     ...
42.
43. // Compare the setup of the following block with Fig. A
44. blocks
45. (
46.     hex (1 2 3 0 5 6 7 4) (20 20 1) simpleGrading ( 1.0 10.0 1.0 ) // Fig. E
47.
```

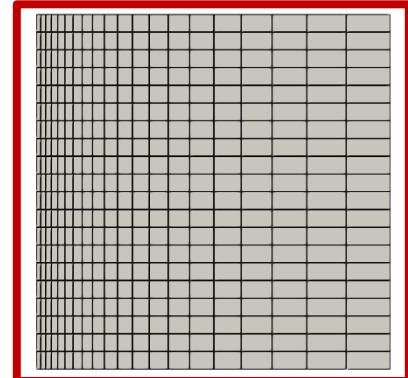


Fig.A

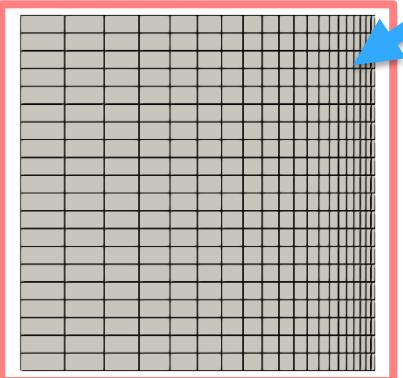


Fig.B

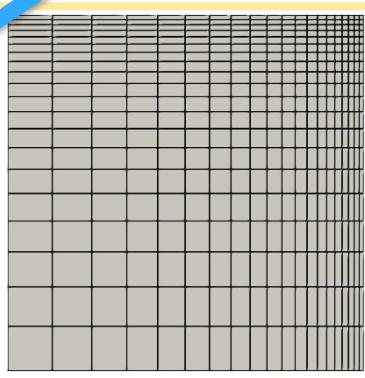


Fig.C

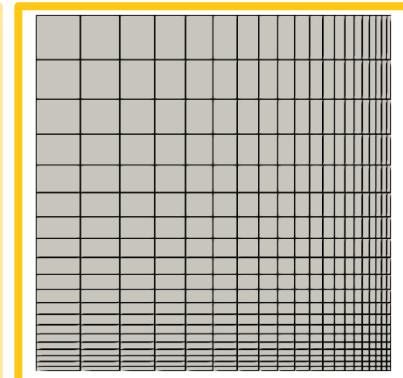


Fig.D

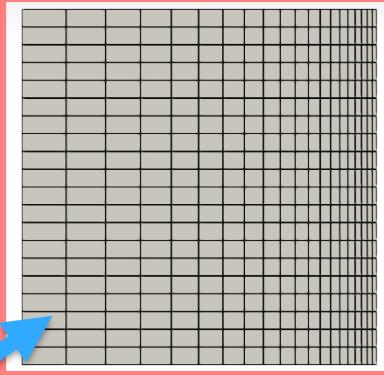


Fig.E



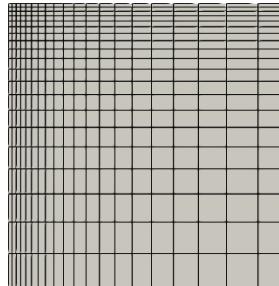
Mesh Refinement Control

case02 - Mesh grading

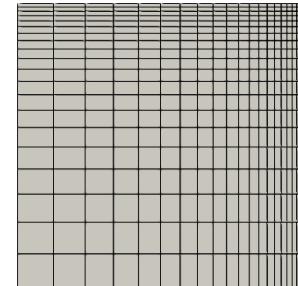
```
./case02/system/blockMeshDict
```

```
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.
```

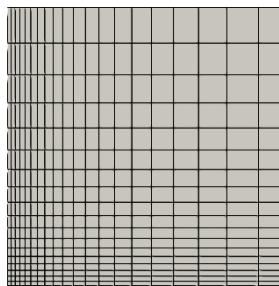
Block 0



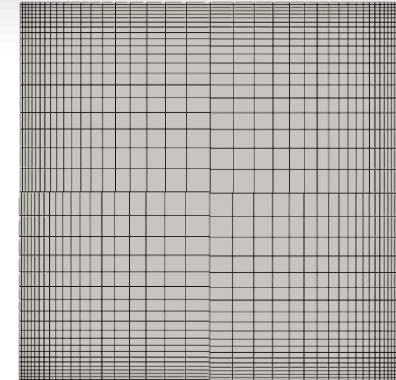
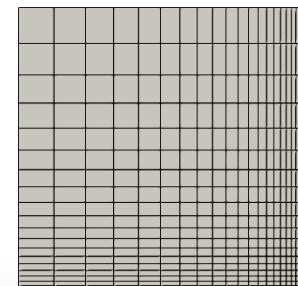
Block 1



Block 2



Block 3

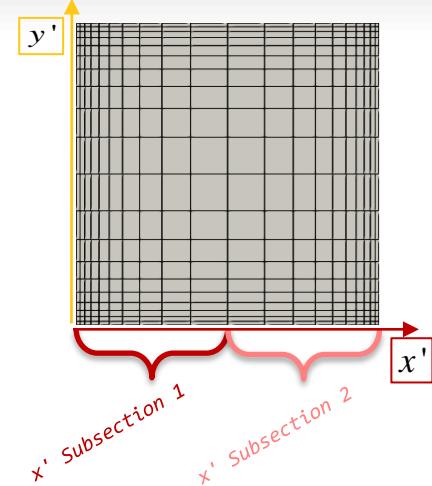


Mesh Refinement Control

case02 - Mesh grading

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

```
31. blocks
32. (
33.     hex (0 1 2 3 4 5 6 7) (20 20 1)
34.     simpleGrading
35.     (
36.         // x' direction
37.         (
38.             // x' Subsection 1:
39.             // Going from 0% to 50 % of the block's direction total Length
40.             // Using 50 % of the cell of this direction with a factor of 10.0
41.             (0.5 0.5 10.0) //Division 1
42.             // x' Subsection 2:
43.             // Going from 50% to 100 % of the block's direction total Length
44.             // Using the remaining 50 % of the cell of this direction with a factor of 0.10
45.             (0.5 0.5 0.10) //Division 2
46.         )
47.         // y' direction
48.         (
49.             // y' Subsection 1:
50.             // Going from 0% to 50 % of the block's direction total Length
51.             // Using 50 % of the cell of this direction with a factor of 10.0
52.             (0.5 0.5 10.0) //Division 1
53.             // y' Subsection 2:
54.             // Going from 50% to 100 % of the block's direction total Length
55.             // Using the remaining 50 % of the cell of this direction with a factor of 0.10
56.             (0.5 0.5 0.10) //Division 2
57.         )
58.         // z' direction
59.         1
60.     ) // Block 0
```



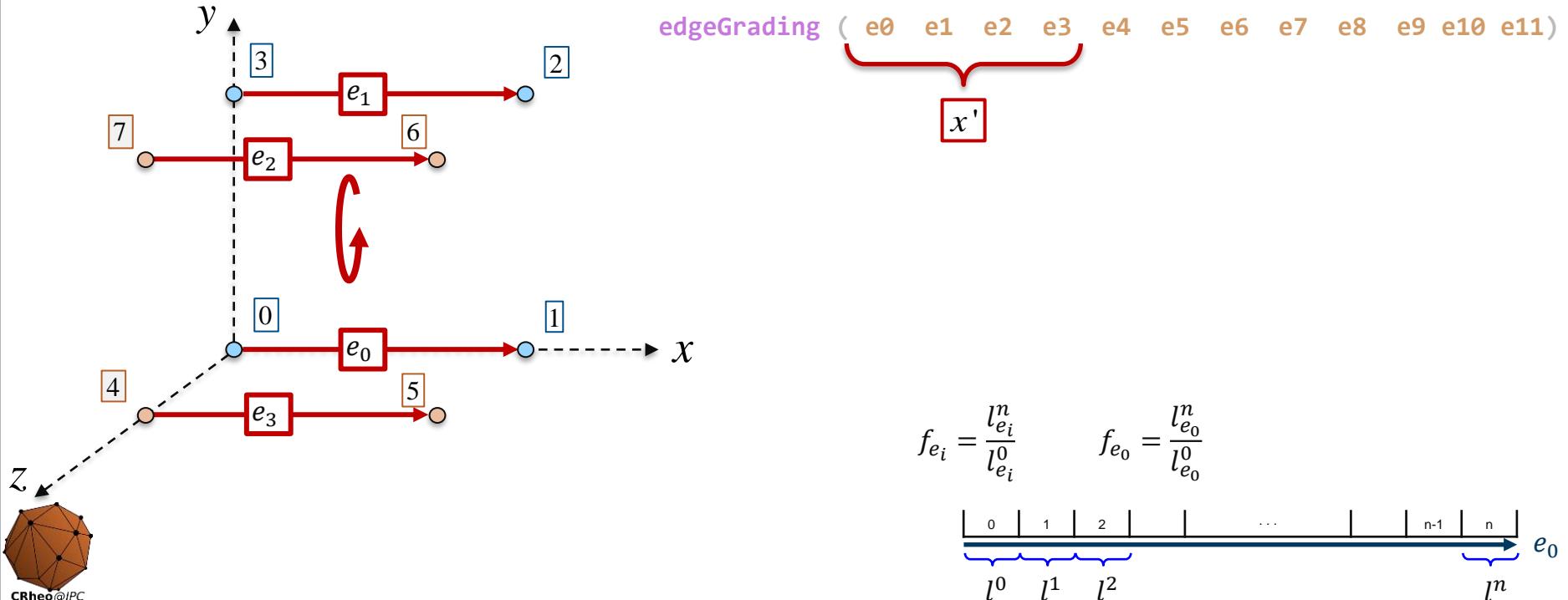
You can use more subsection segments with different expansion ratios



Mesh Refinement Control

case02 - Mesh grading

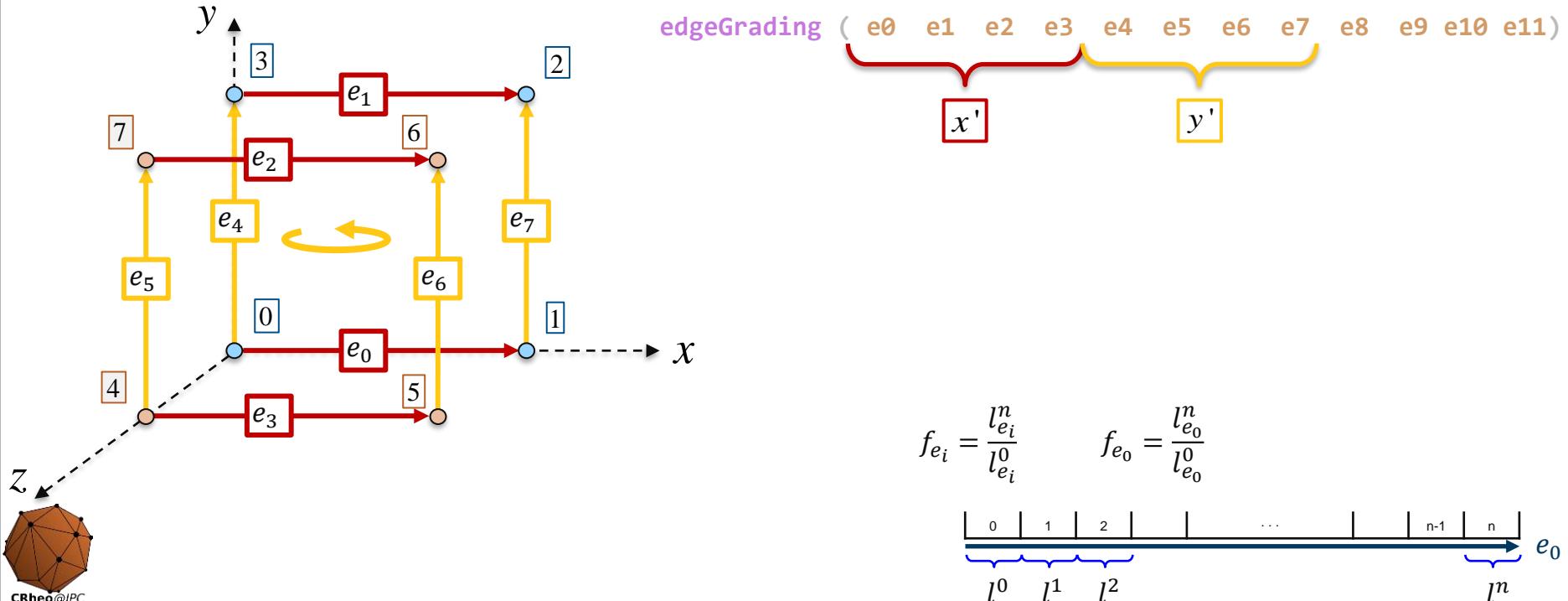
- Mesh grading for stretching the mesh towards one or more **Vertex**.



Mesh Refinement Control

case02 - Mesh grading

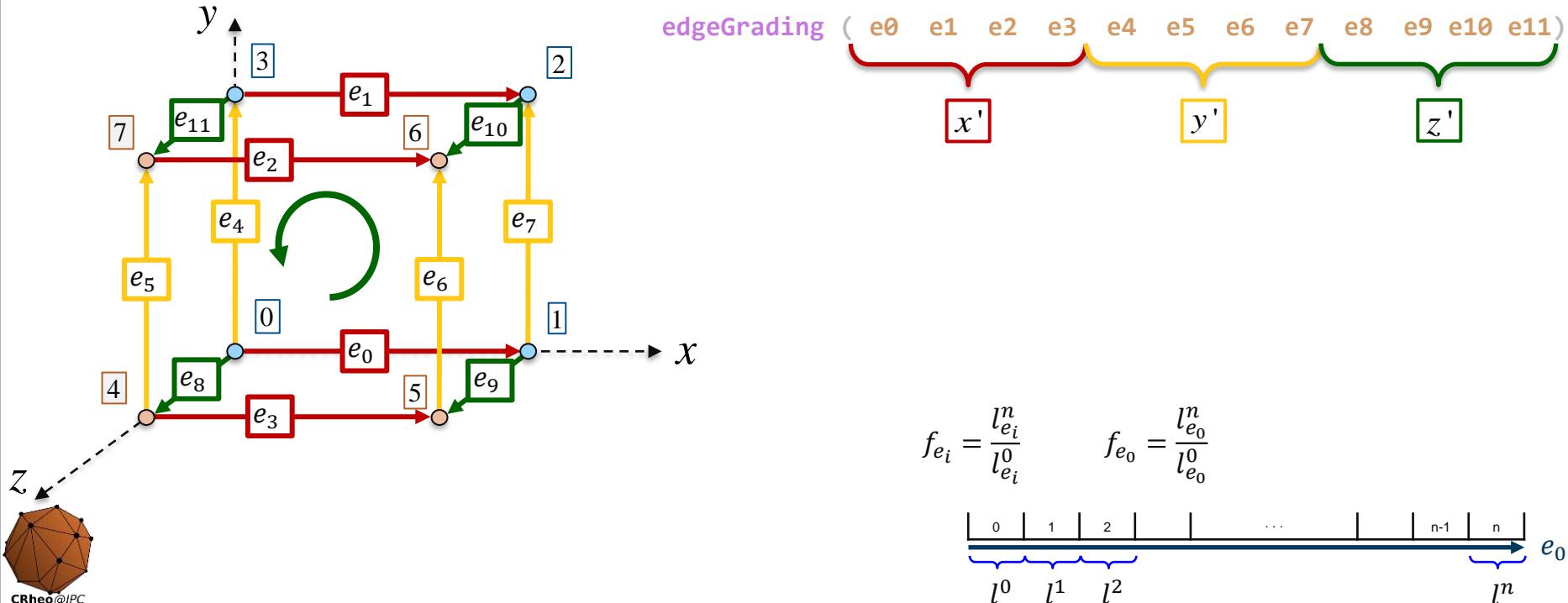
- Mesh grading for stretching the mesh towards one or more **Vertices**.



Mesh Refinement Control

case02 - Mesh grading

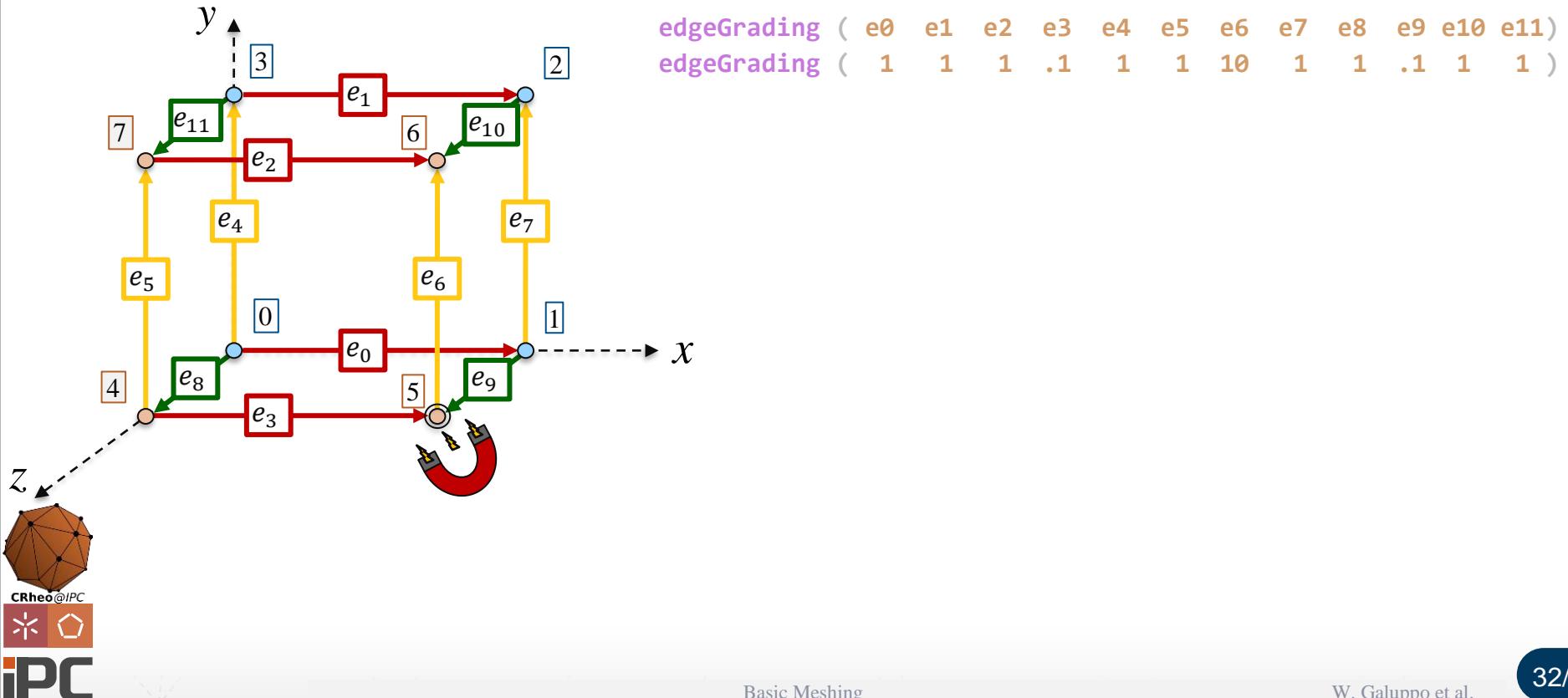
- Mesh grading for stretching the mesh towards one or more Vertices.



Mesh Refinement Control

case02 - Mesh grading

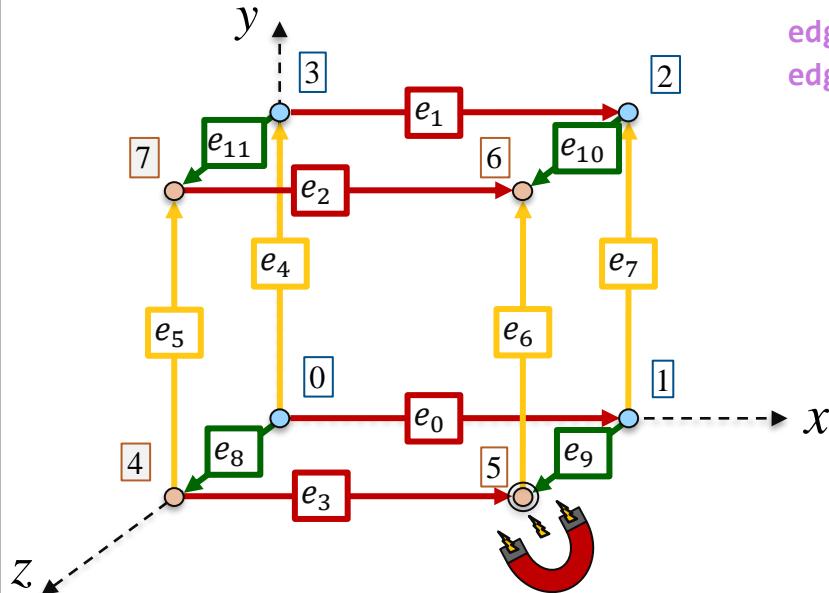
- Mesh grading for stretching the mesh towards one or more **Vertices**.



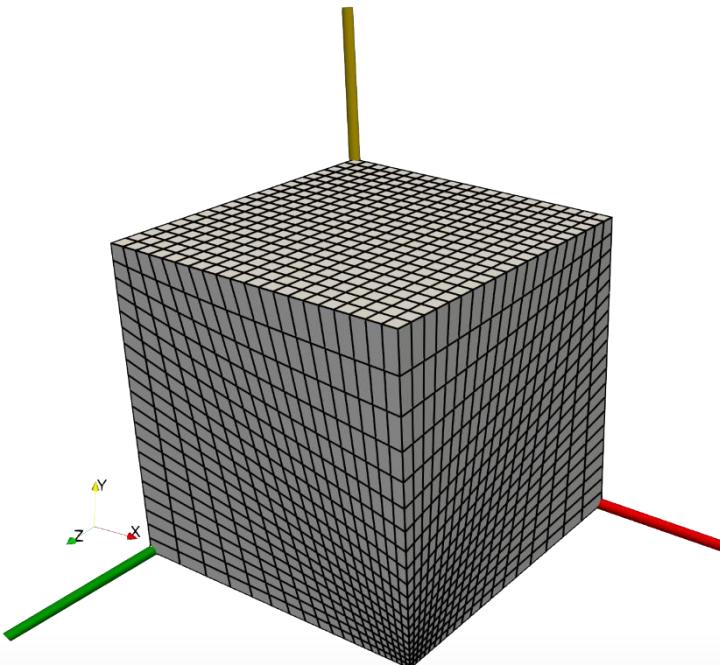
Mesh Refinement Control

case02 - Mesh grading

- Mesh grading for stretching the mesh towards one or more **Vertices**.



```
edgeGrading ( e0  e1  e2  e3  e4  e5  e6  e7  e8  e9  e10  e11 )
edgeGrading ( 1   1   1   .1  1   1   10  1   1   .1  1   1 )
```



3

Multiblock Mesh

case03 - Blocks connections

```
cd <Desktop>/FOAM@Iberia-Courses/Beginner/C2/tutorials  
cp -r ./caseTutorials/case03 ./  
cd case03  
code .
```



CRheo@IPC



- Physical Problem



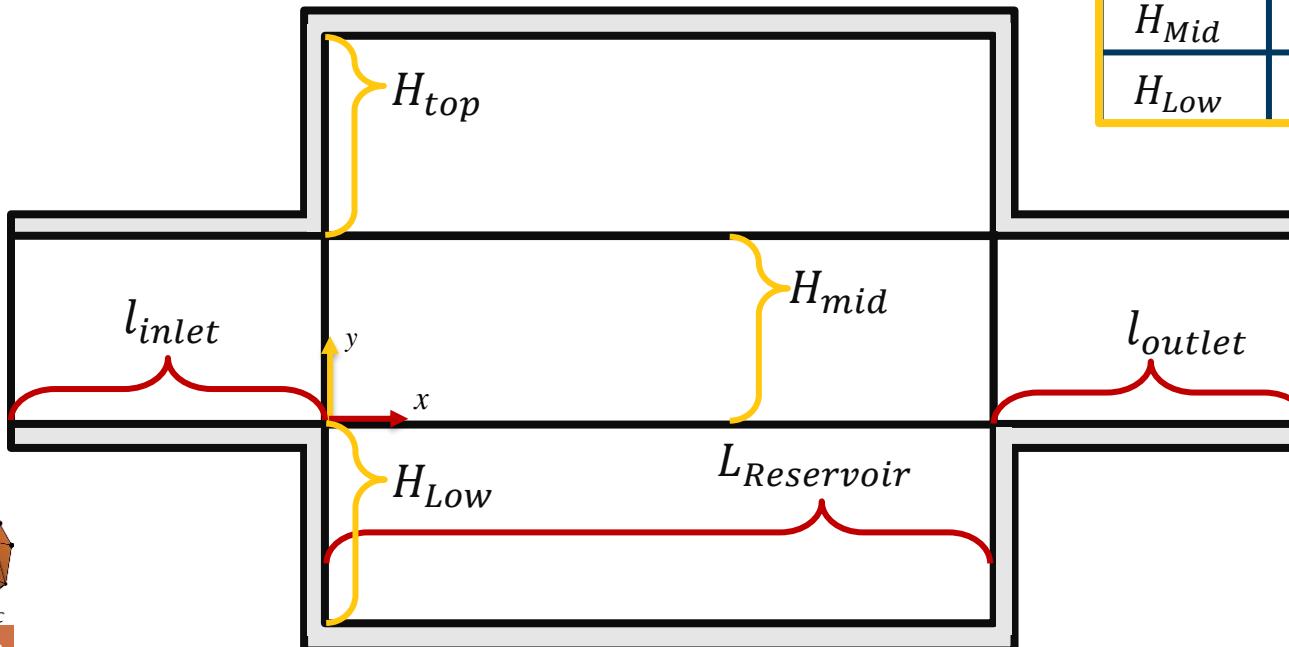
CRheo@IPC



Multiblock Mesh

case03 - Blocks connections

- Physical Problem Dimensions



	H (m)		L (m)
H_{Top}	0.75	L_{in}	1.00
H_{Mid}	0.25	L_{Reserv}	4.50
H_{Low}	0.50	L_{out}	1.00

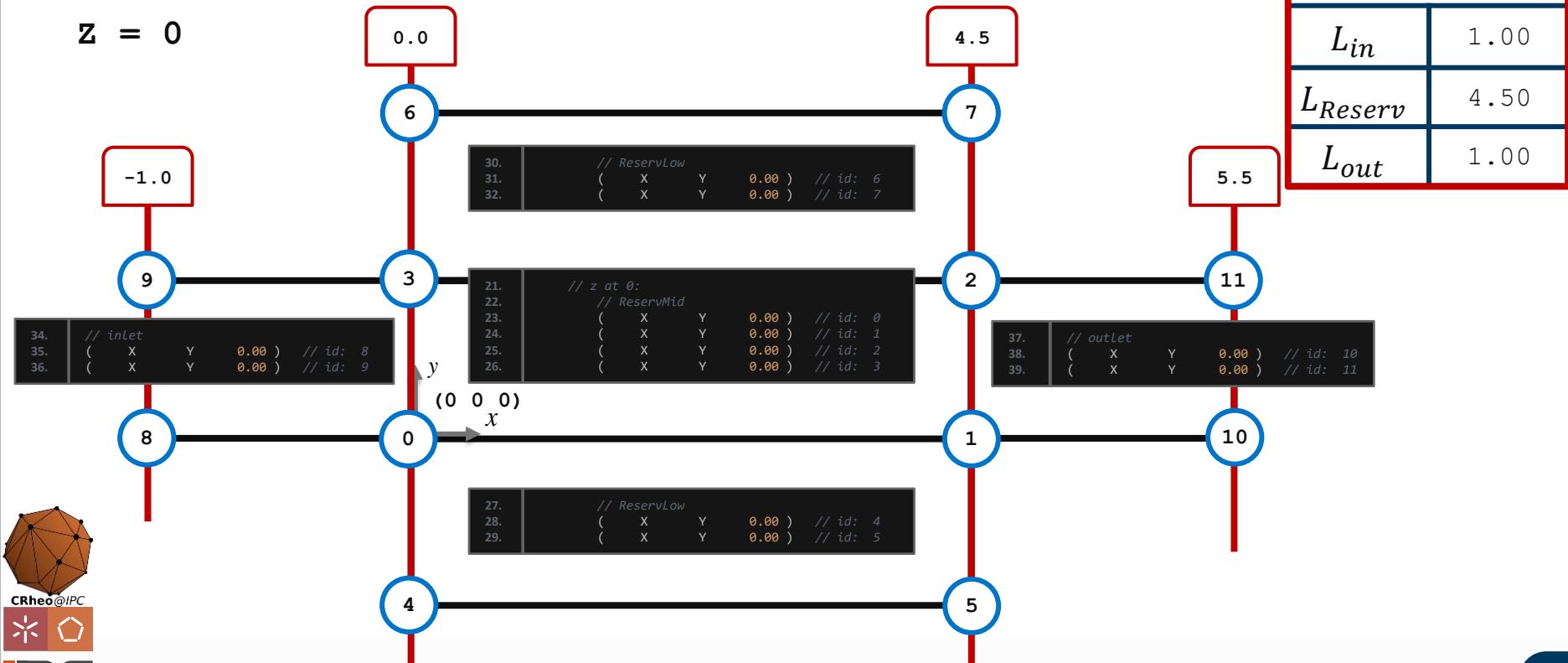
$$\Delta z = 0.5$$

Multiblock Mesh

case03 - Blocks connections

- Geometry “X” Coordinates

$z = 0$

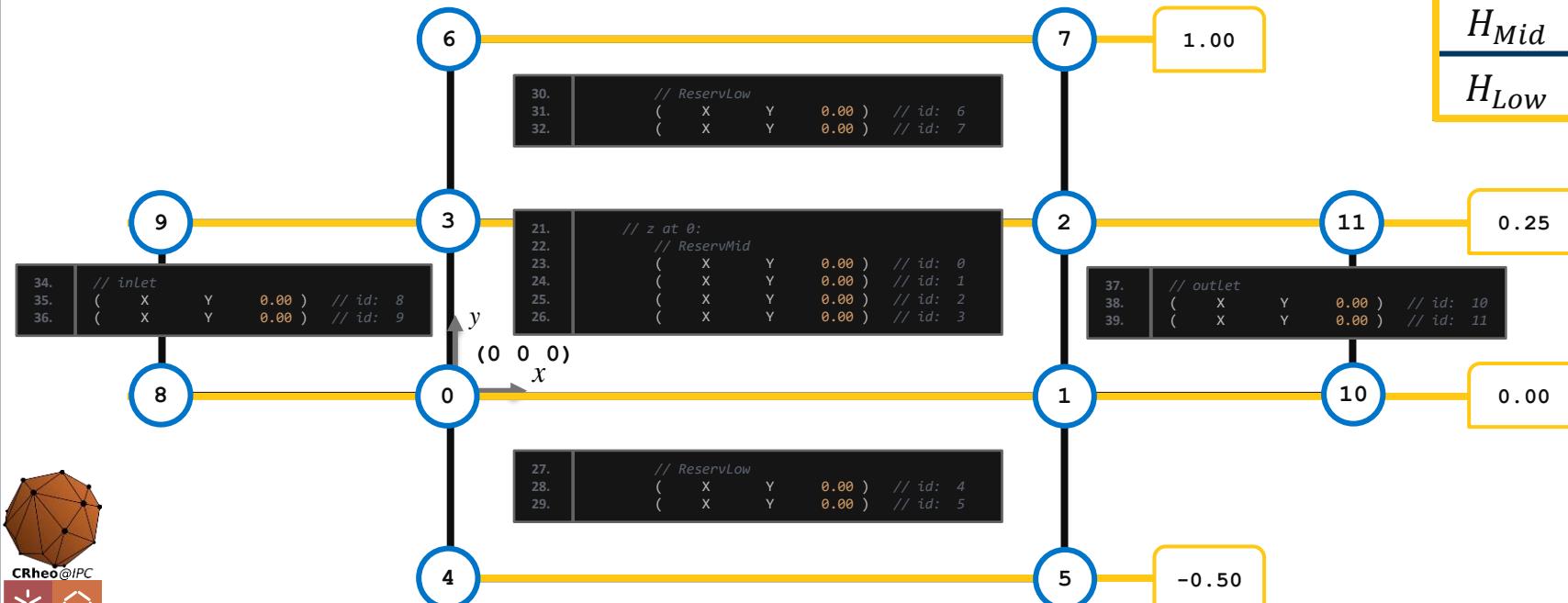


Multiblock Mesh

case03 - Blocks connections

- Geometry "Y" Coordinates

$z = 0$

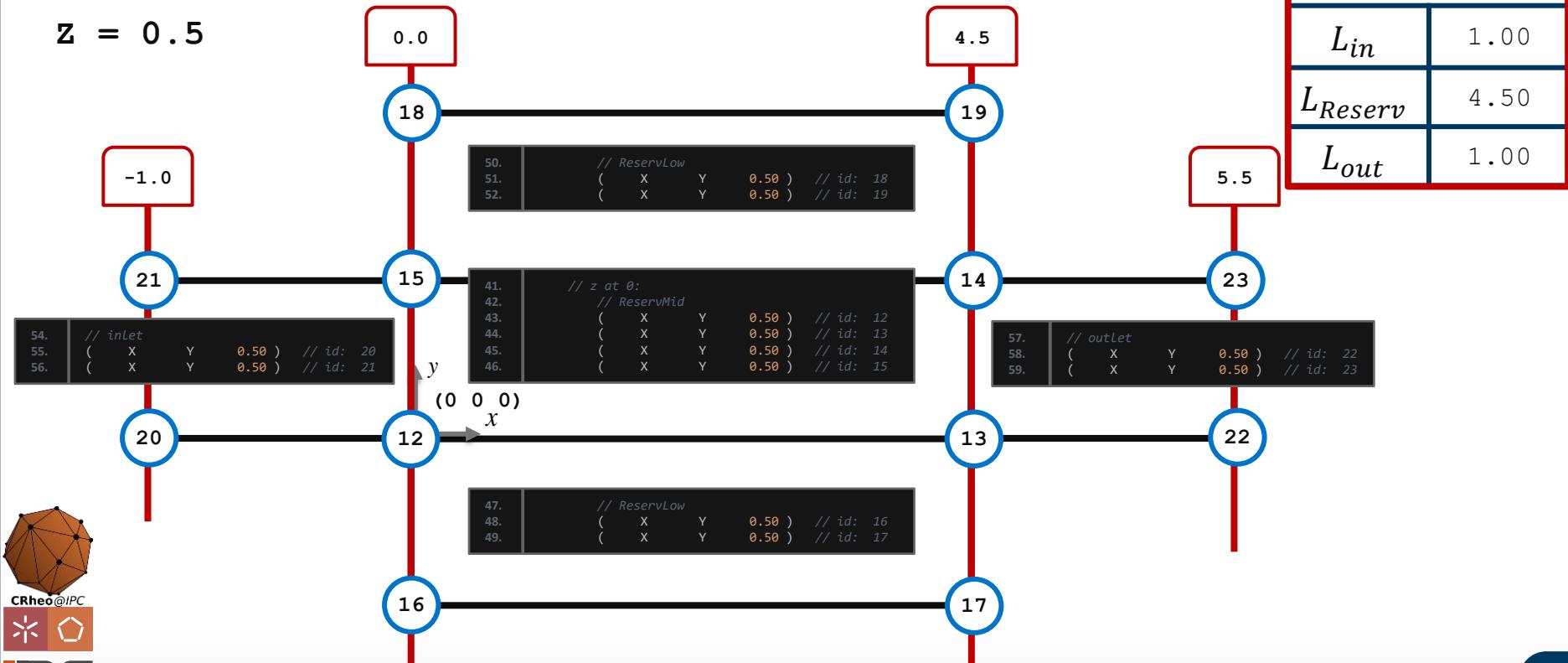


Multiblock Mesh

case03 - Blocks connections

- Geometry “X” Coordinates

Z = 0.5

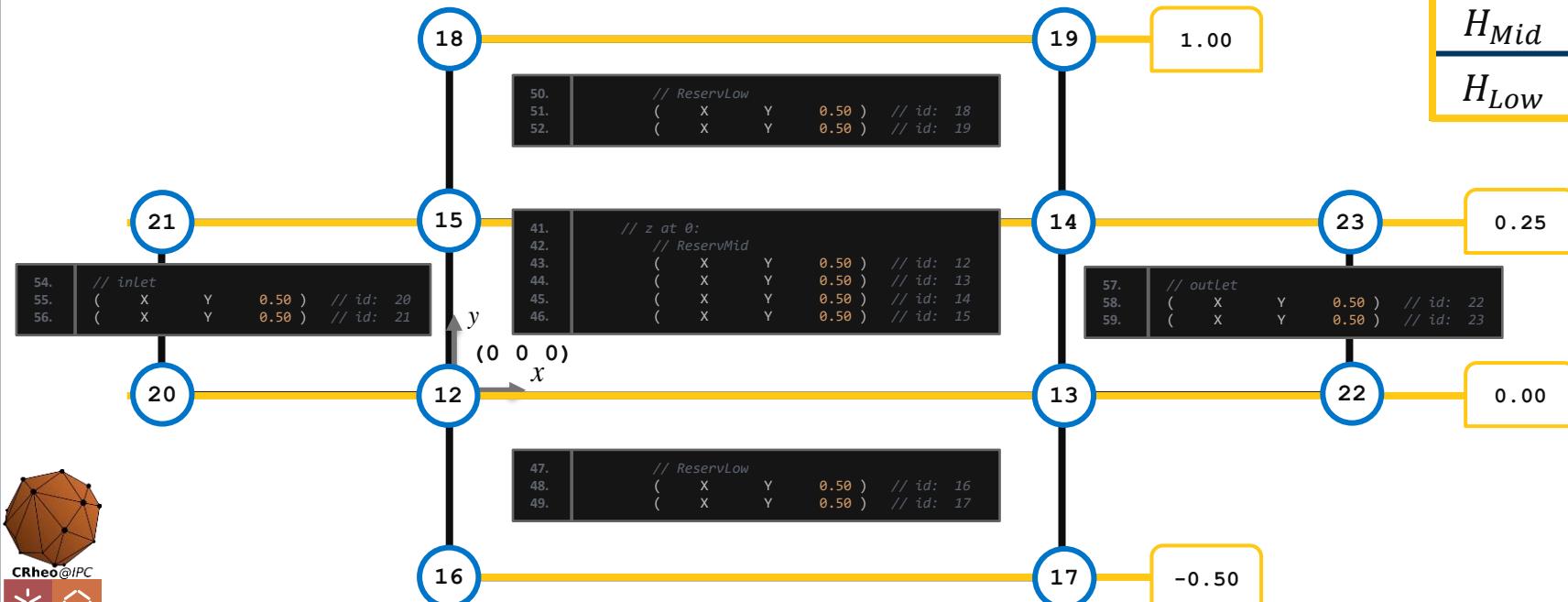


Multiblock Mesh

case03 - Blocks connections

- Geometry "Y" Coordinates

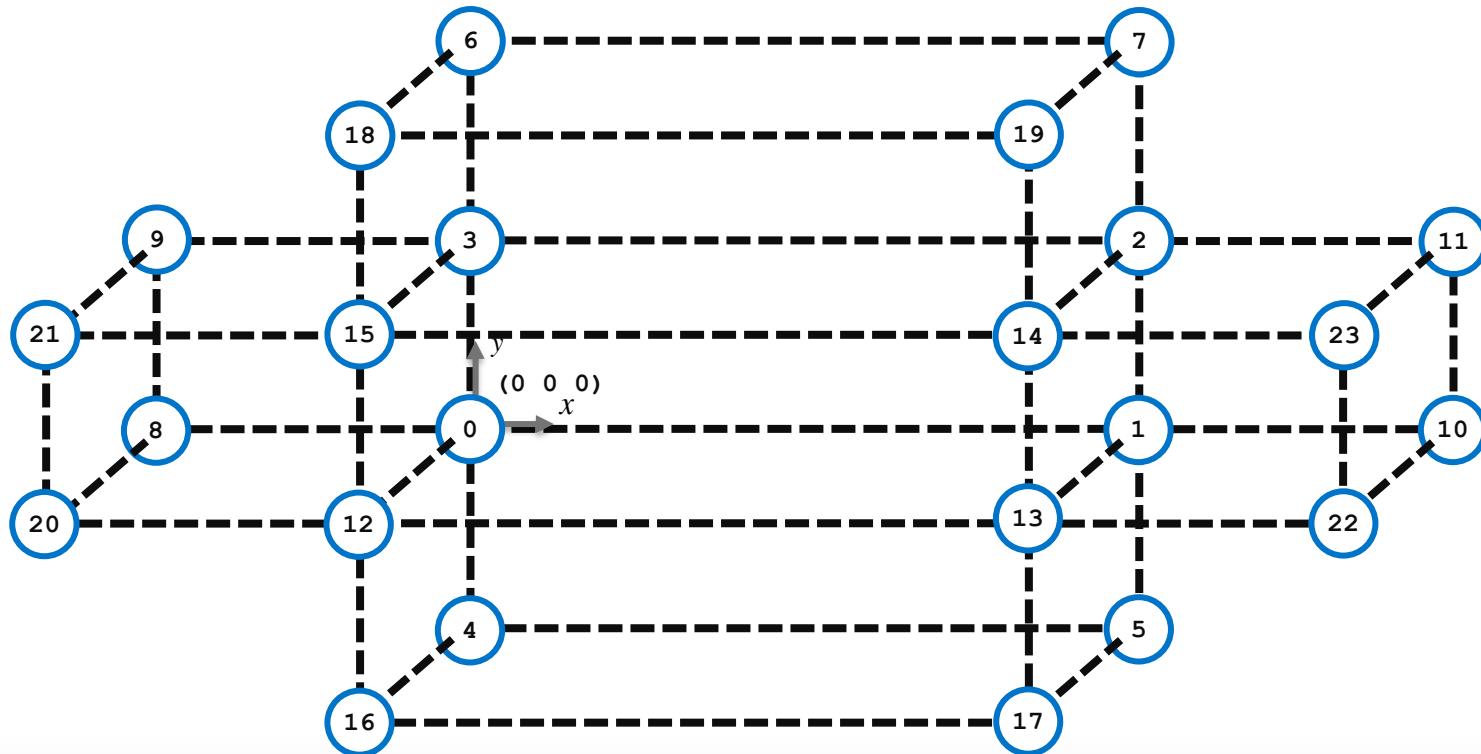
$z = 0.5$



Multiblock Mesh

case03 - Blocks connections

- Blocks Creation



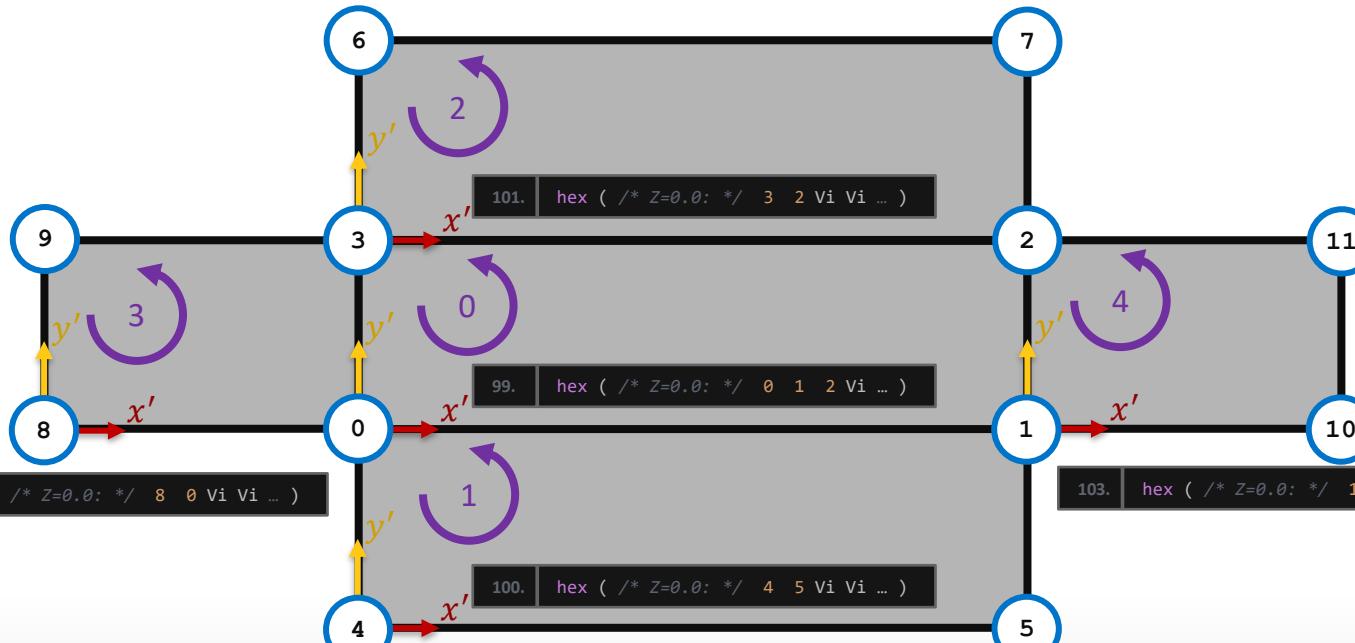
Multiblock Mesh

case03 - Blocks connections

- Blocks Creation

Z = 0.0

```
96. blocks
97.   ( /* hex (           Vertices indices to create a block           ) ( slicing the block ) simpleGrading ( expansion Ratio factor )
98.     // Block Down
99.     hex ( /* Z=0.0: */  0  1  2 Vi /* Z=0.5: */ Vi Vi Vi Vi ) ( 1  1  1 ) simpleGrading ( 1 1 1 ) // block id: 0 - ReservMid
```

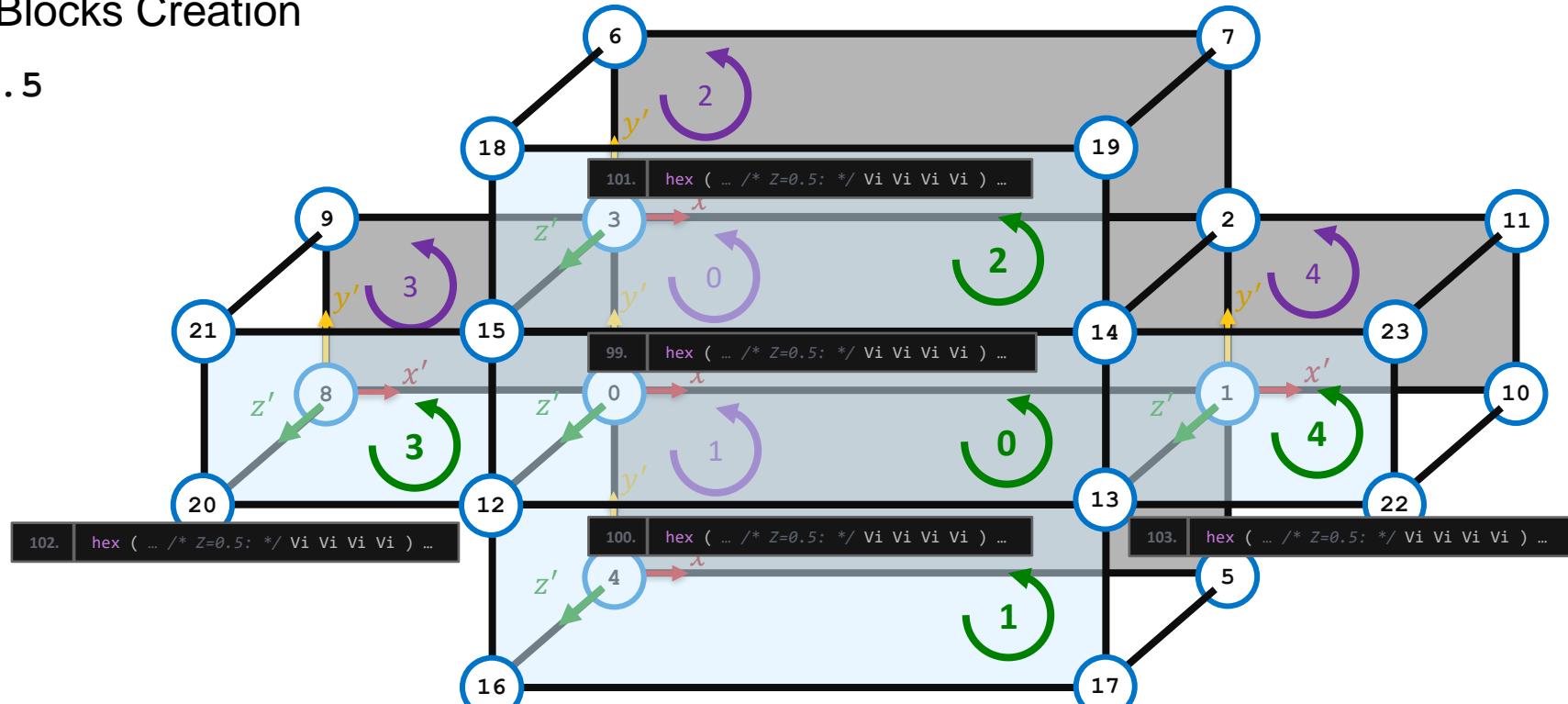


Multiblock Mesh

case03 - Blocks connections

- Blocks Creation

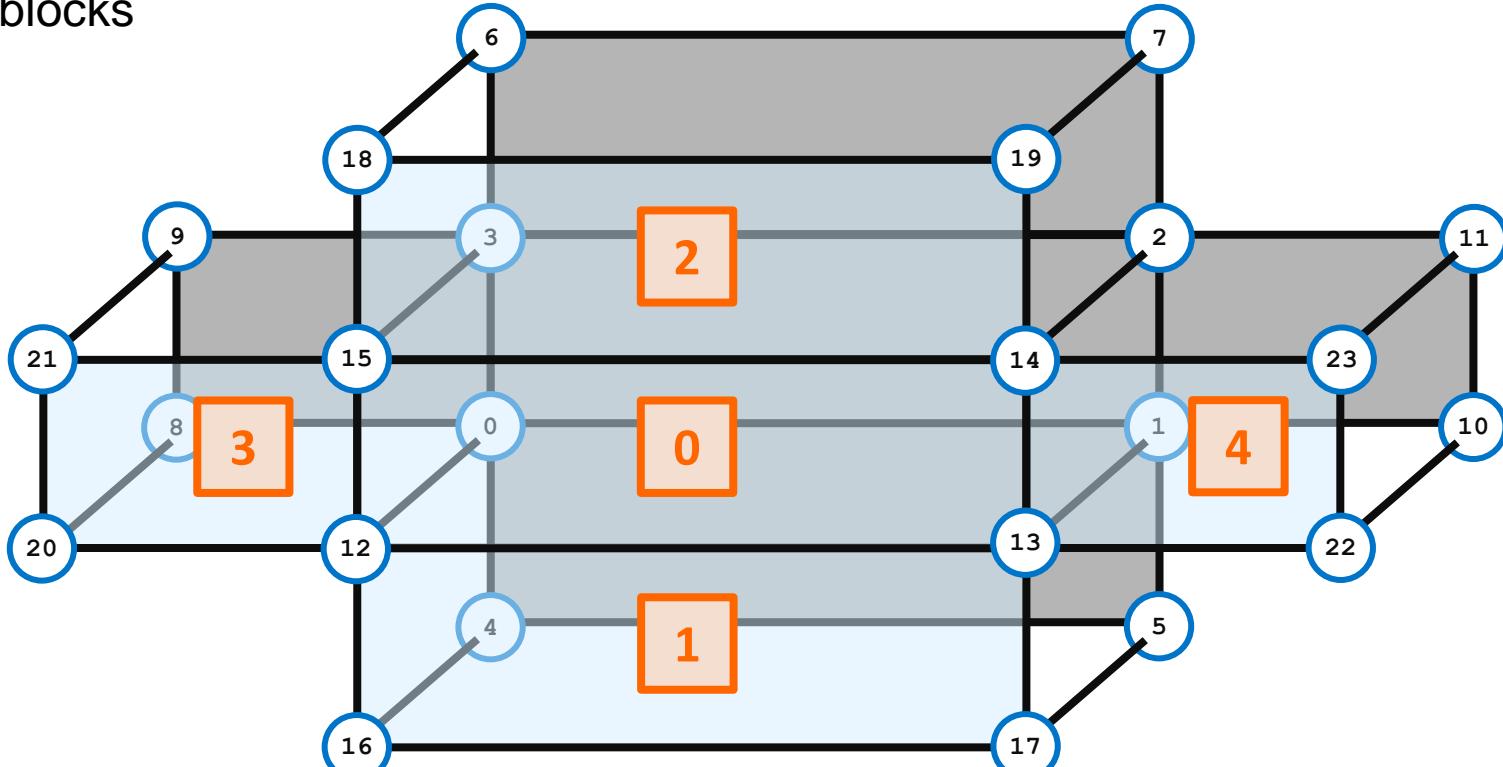
$z = 0.5$



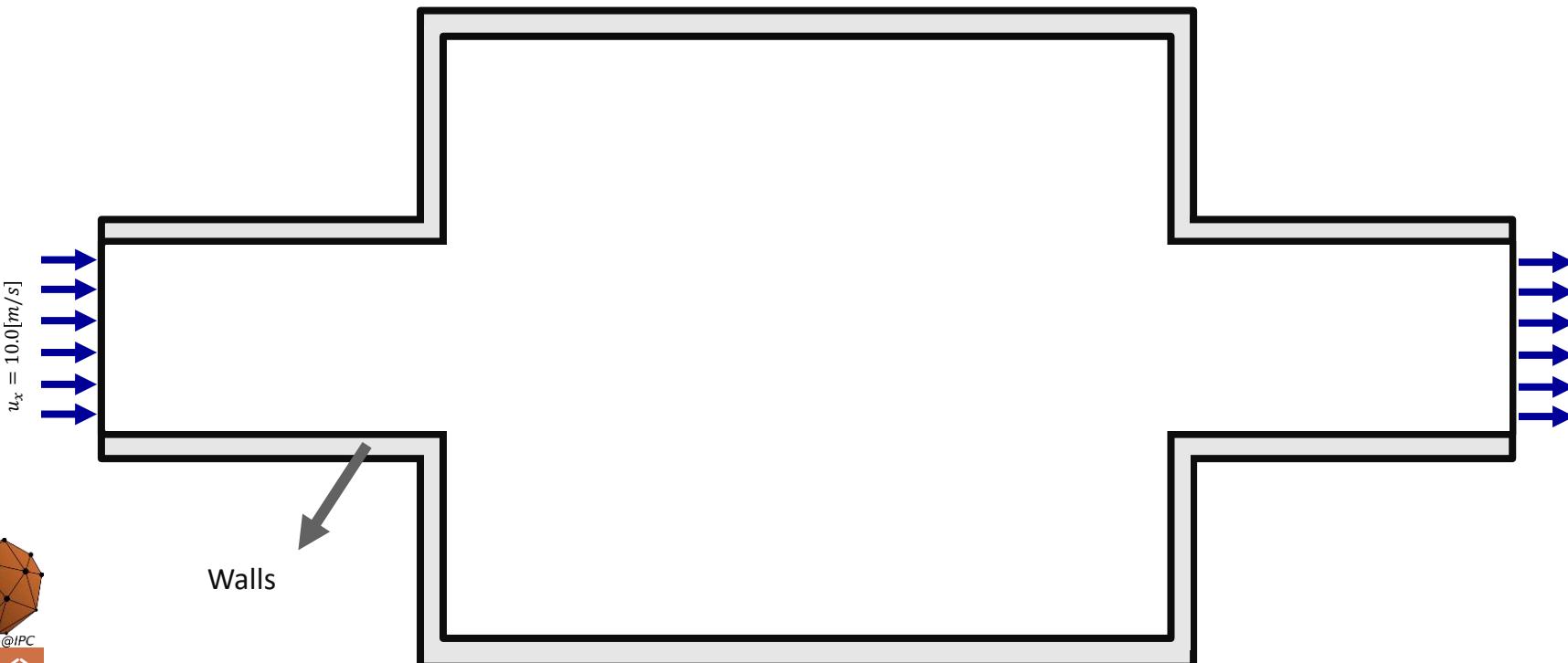
Multiblock Mesh

case03 - Blocks connections

- Created blocks



- Physical Problem

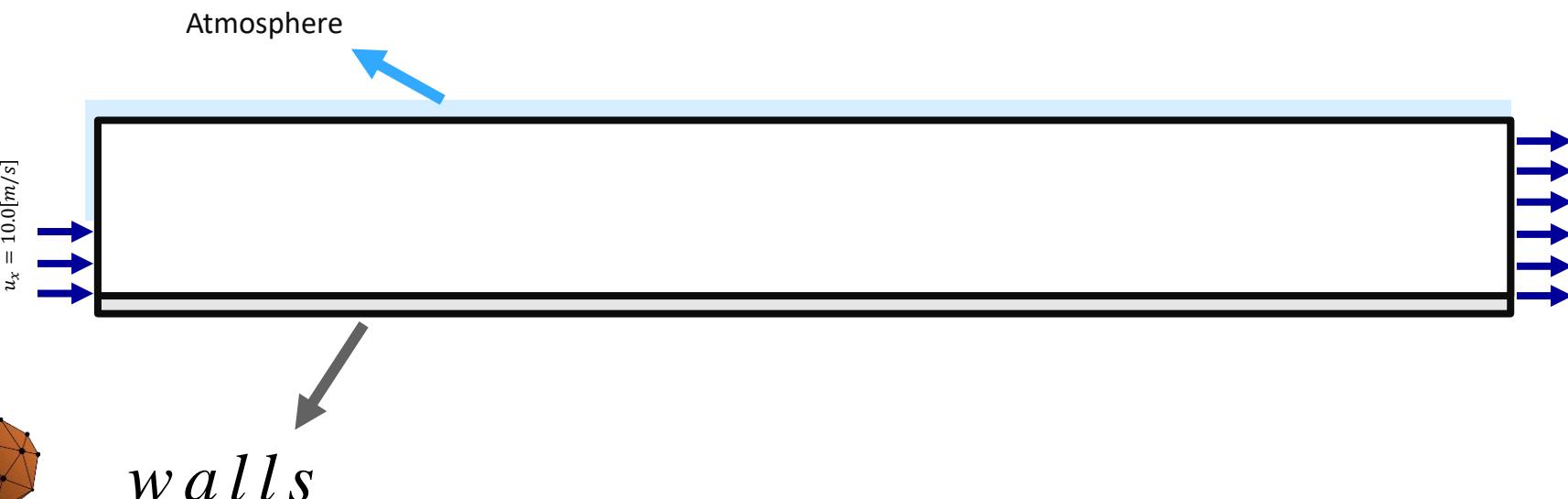


CRheo@IPC



iPC

- Physical Problem



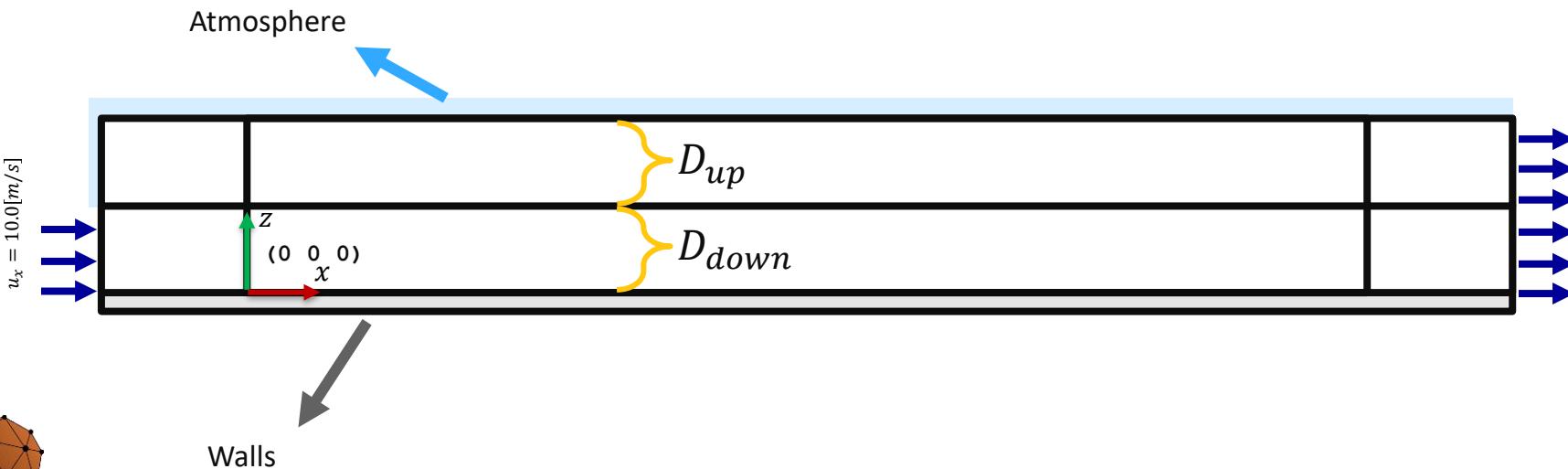
CRheo@IPC



iPC

- Physical Problem

Depths



CRheo@IPC

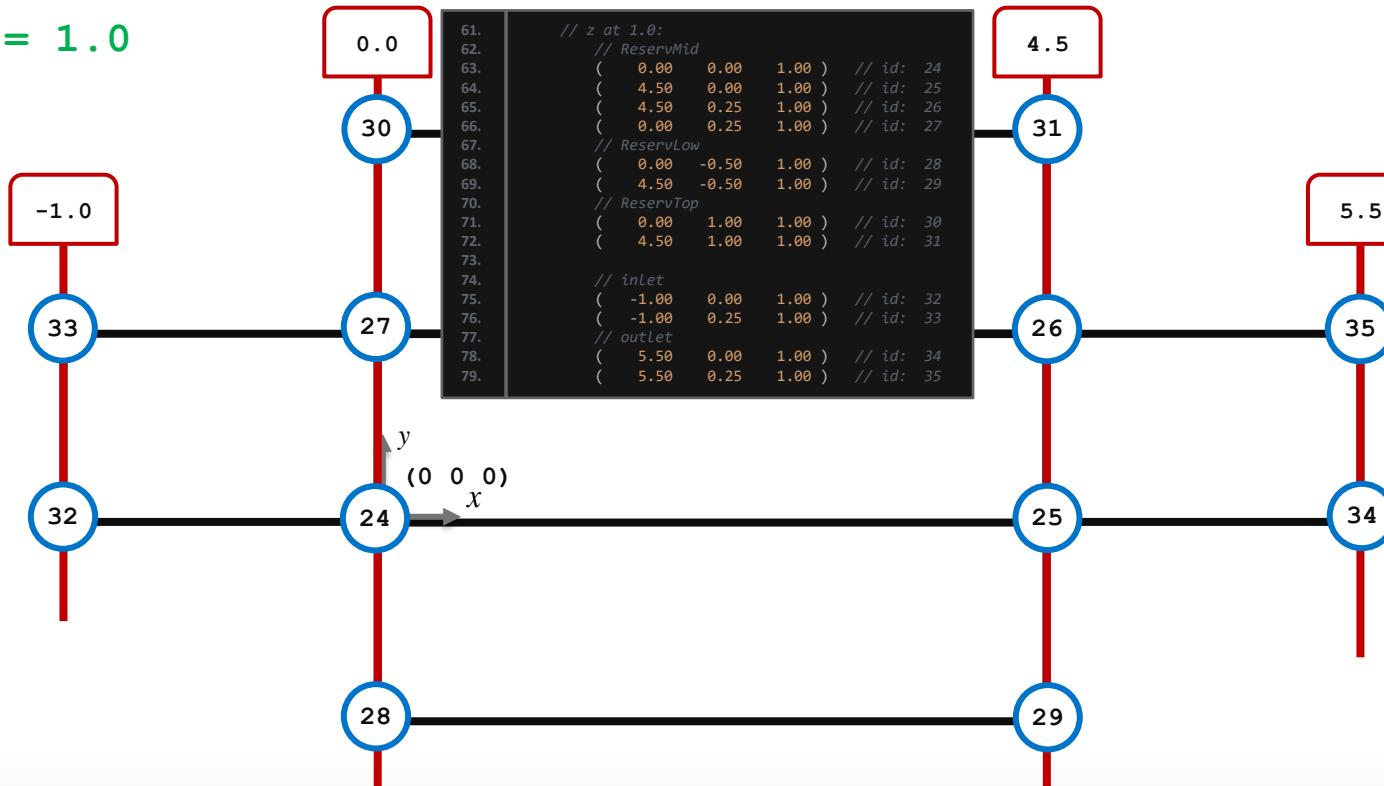


Multiblock Mesh

case03 - Blocks connections

- Geometry “X” Coordinates

$z = 1.0$



L (m)	
L_{in}	1.00
L_{Reserv}	4.50
L_{out}	1.00

Multiblock Mesh

case03 - Blocks connections

- Geometry "Y" Coordinates

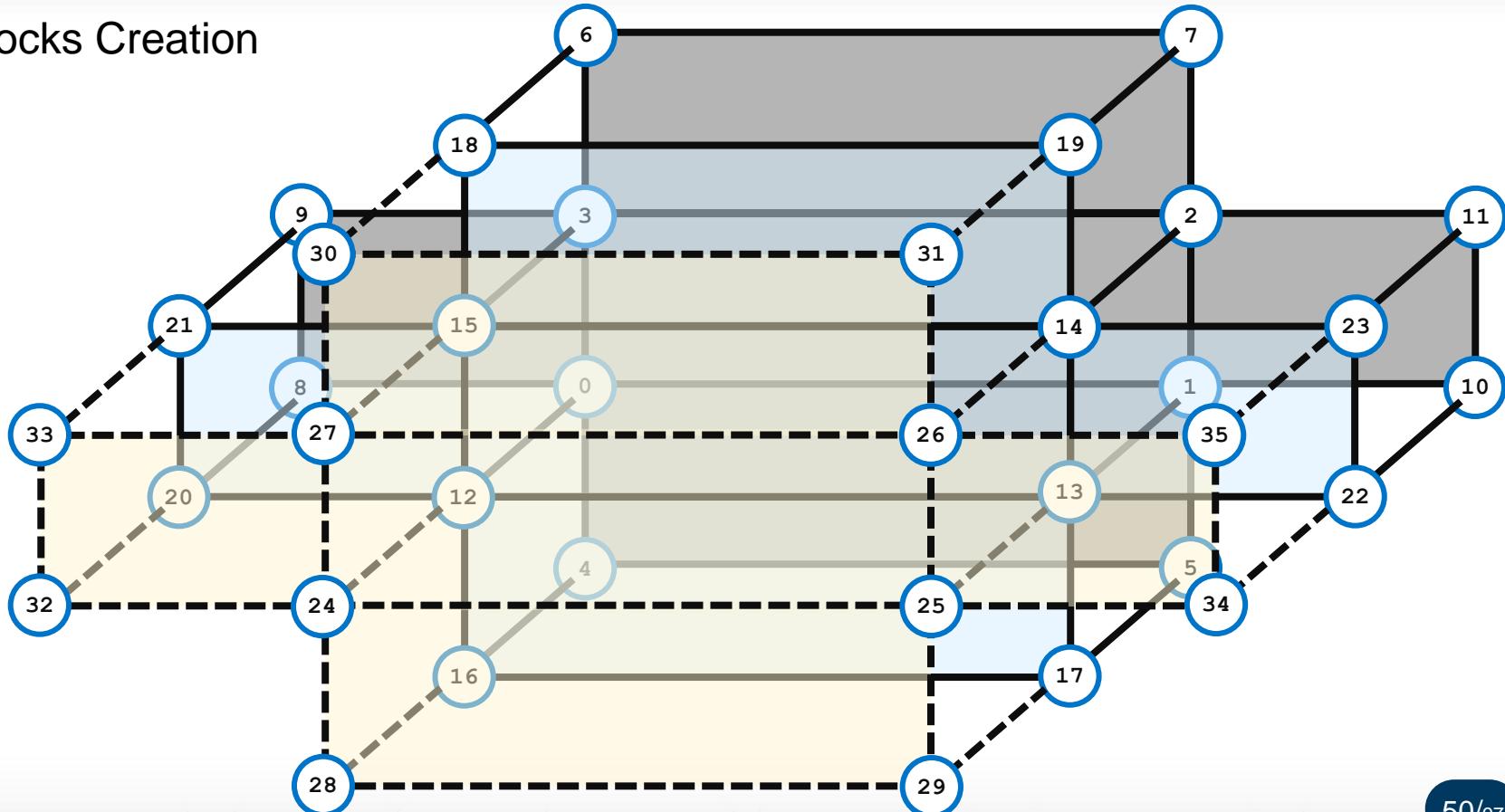
$z = 1.0$



Multiblock Mesh

case03 - Blocks connections

- Blocks Creation

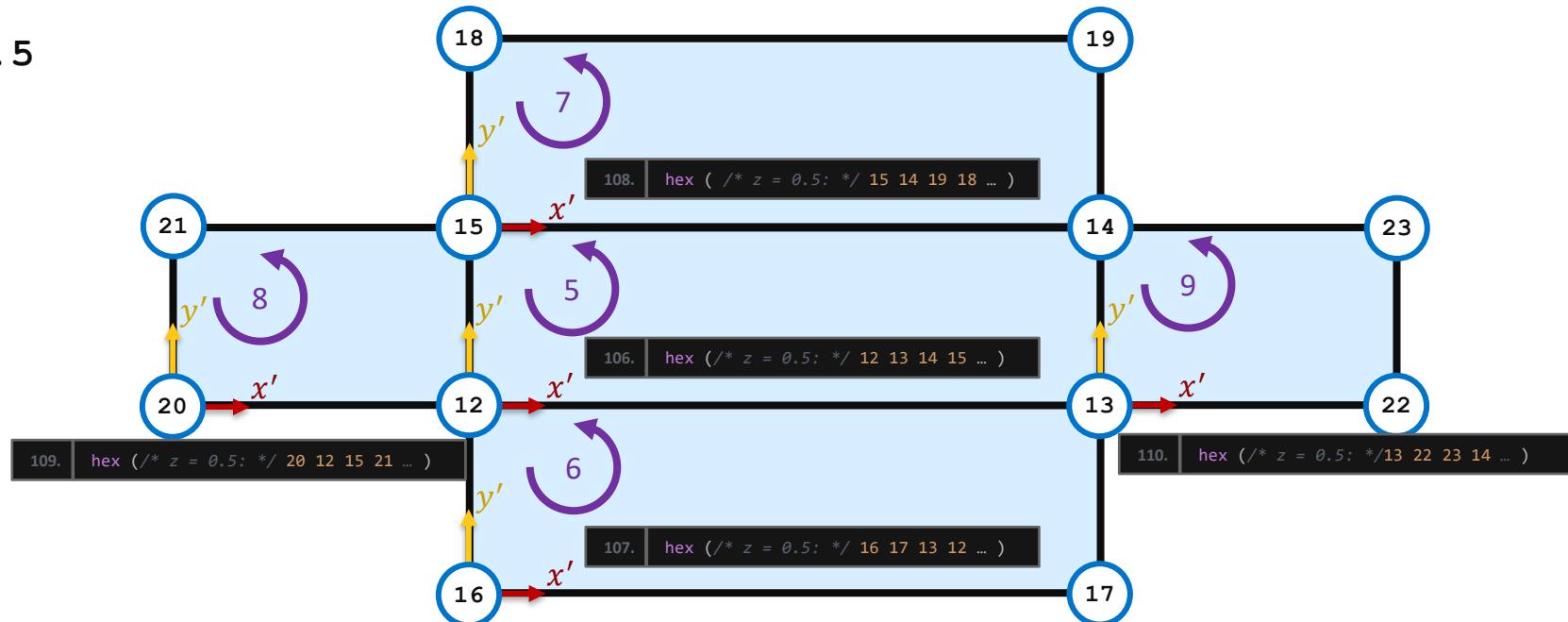


Multiblock Mesh

case03 - Blocks connections

- **Blocks Creation**

Z = 0.5



CBHeo@IPC



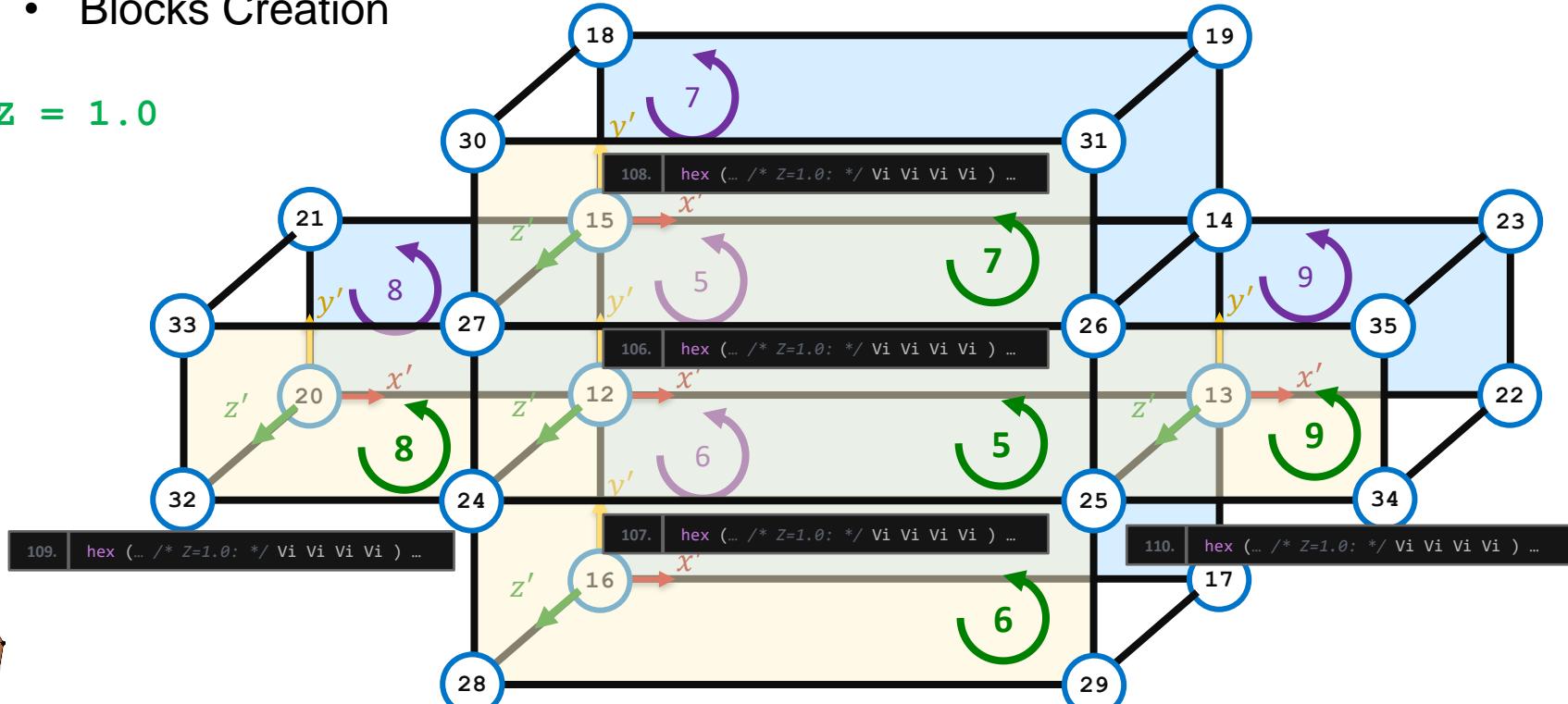
iPC

Multiblock Mesh

case03 - Blocks connections

- Blocks Creation

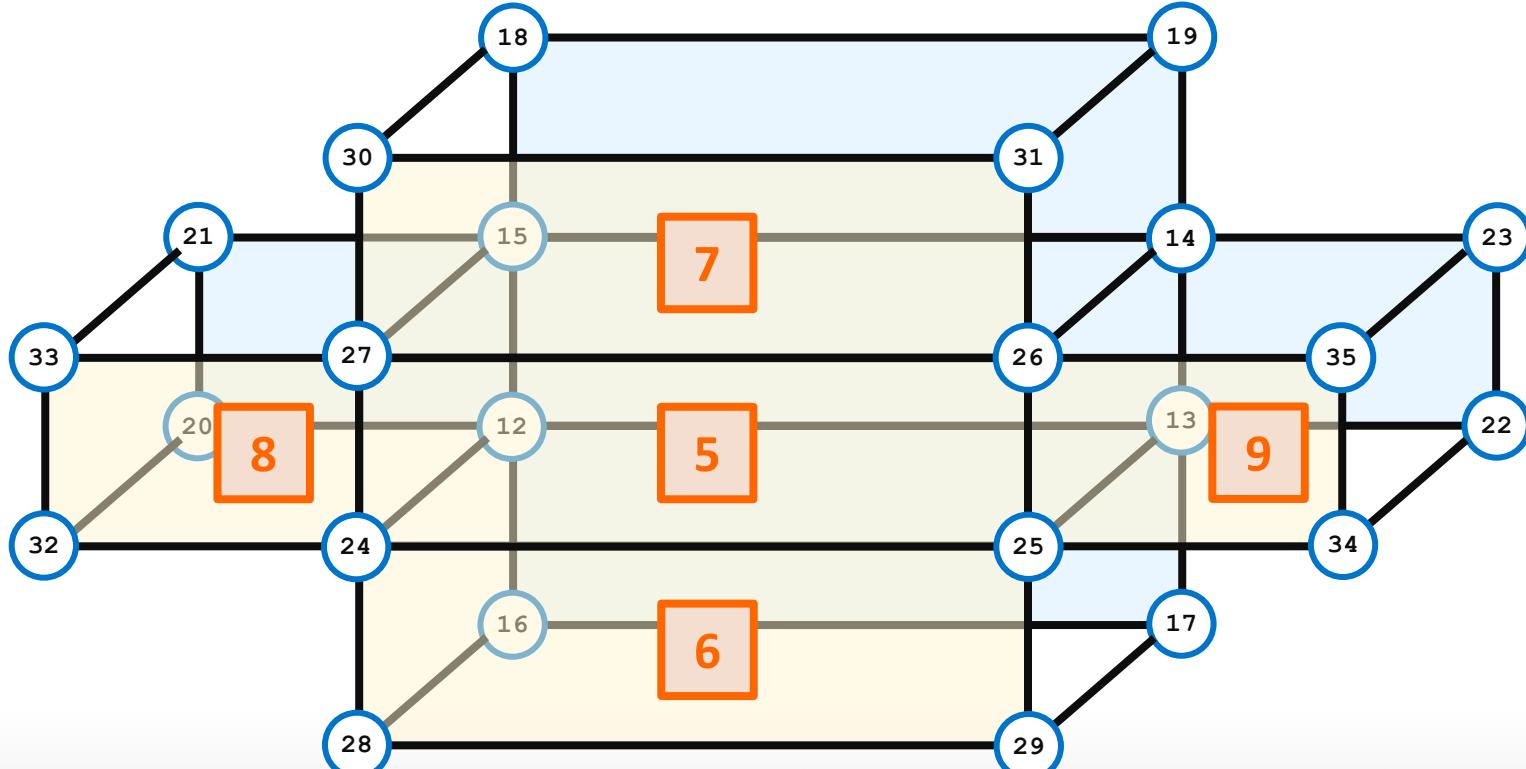
$z = 1.0$



Multiblock Mesh

case03 - Blocks connections

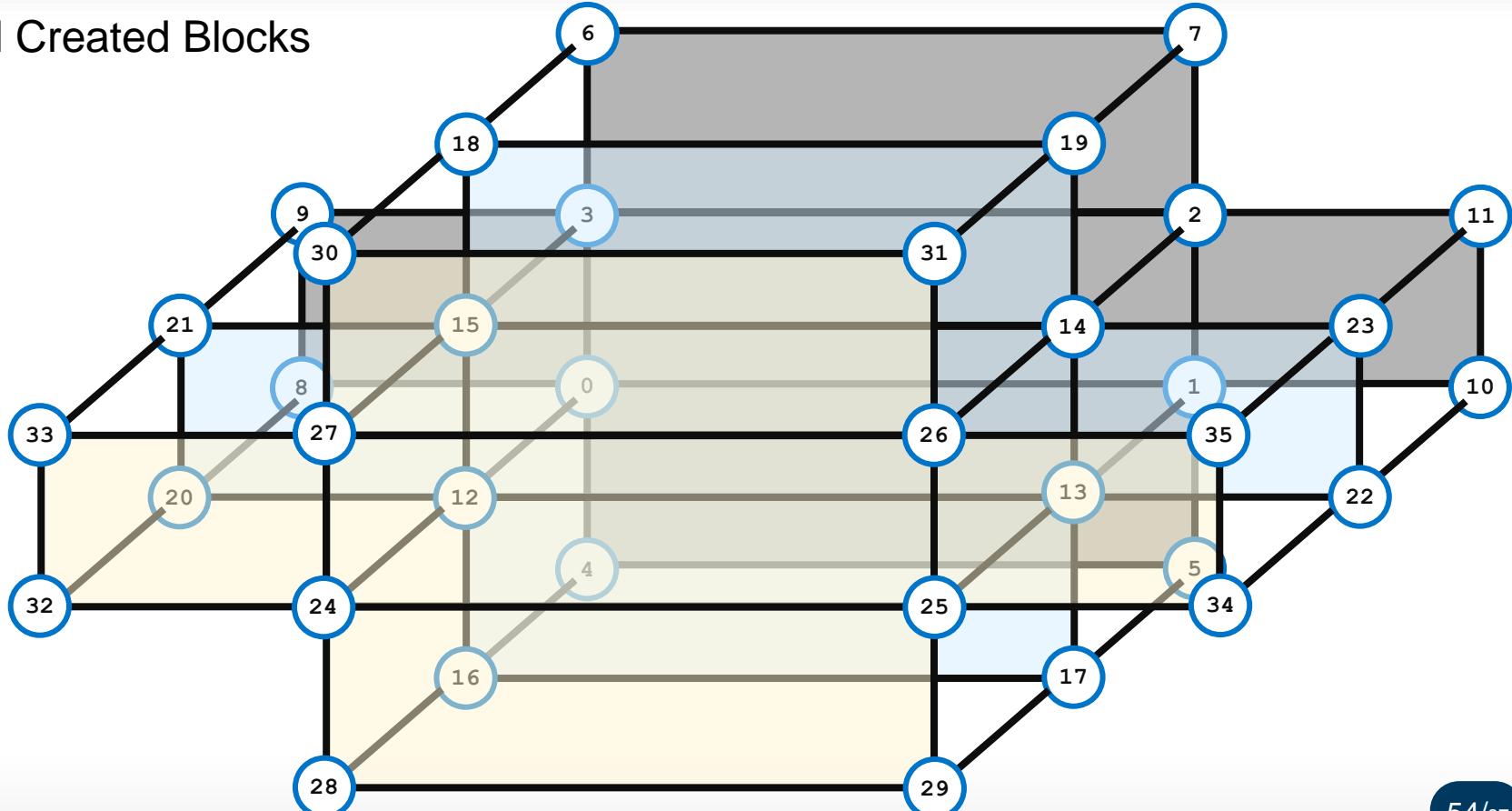
- Created Blocks



Multiblock Mesh

case03 - Blocks connections

- All Created Blocks



CRheo@IPC

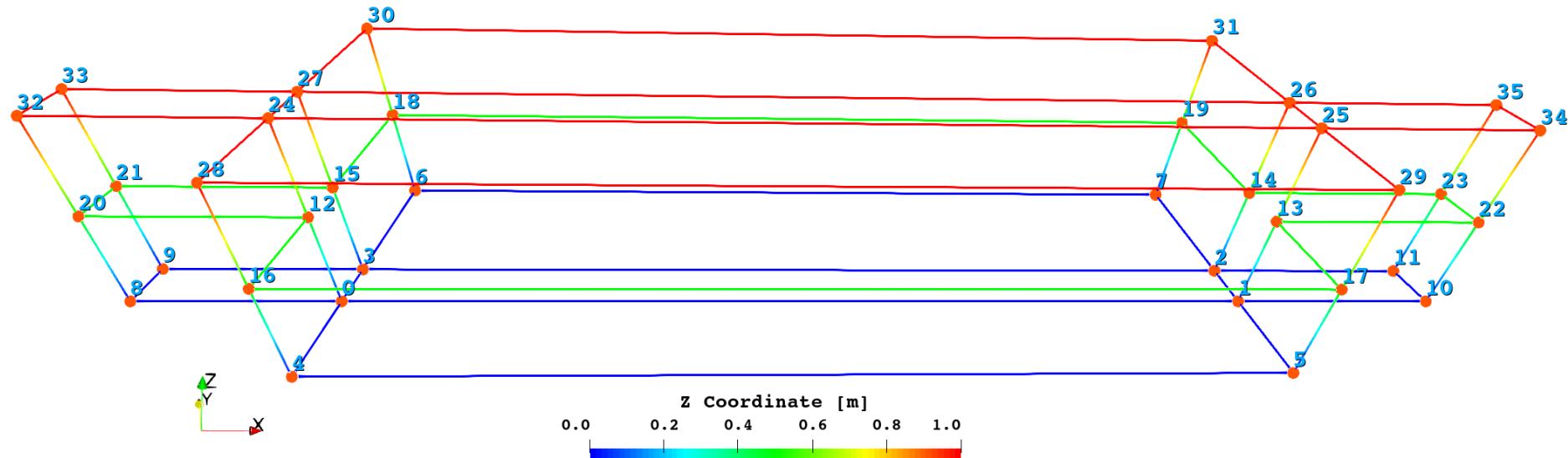


iPC

Multiblock Mesh

case03 - Blocks connections

- All Created Blocks



Script in Bash

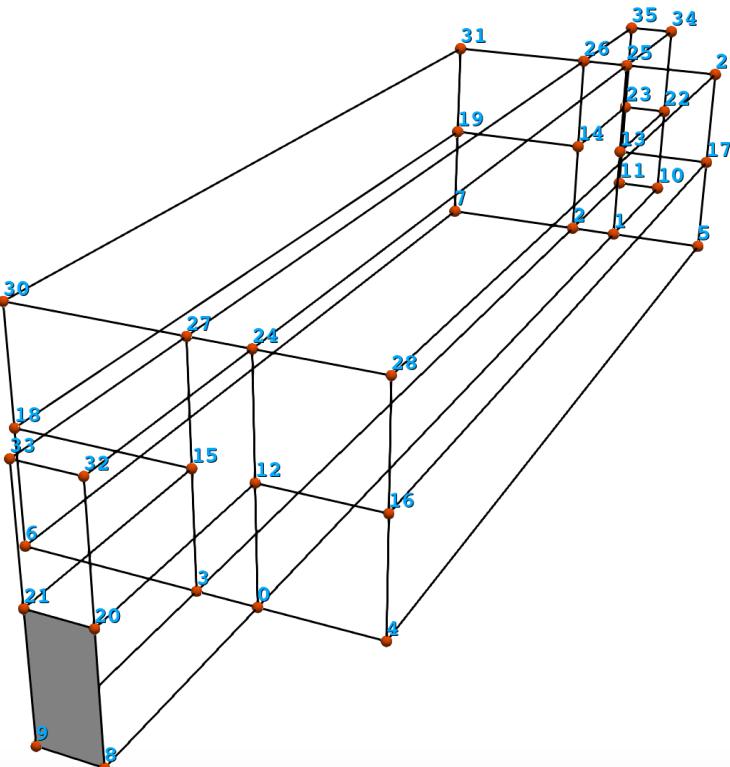


Multiblock Mesh

case03 - Blocks connections

- Boundaries - inlet

```
.../case03/system/blockMeshDict
118. boundary
119. (
120.     inlet
121.     {
122.         type patch;
123.         faces
124.         (
125.             ( 9 /* */ )
126.         );
127.     }
}
```

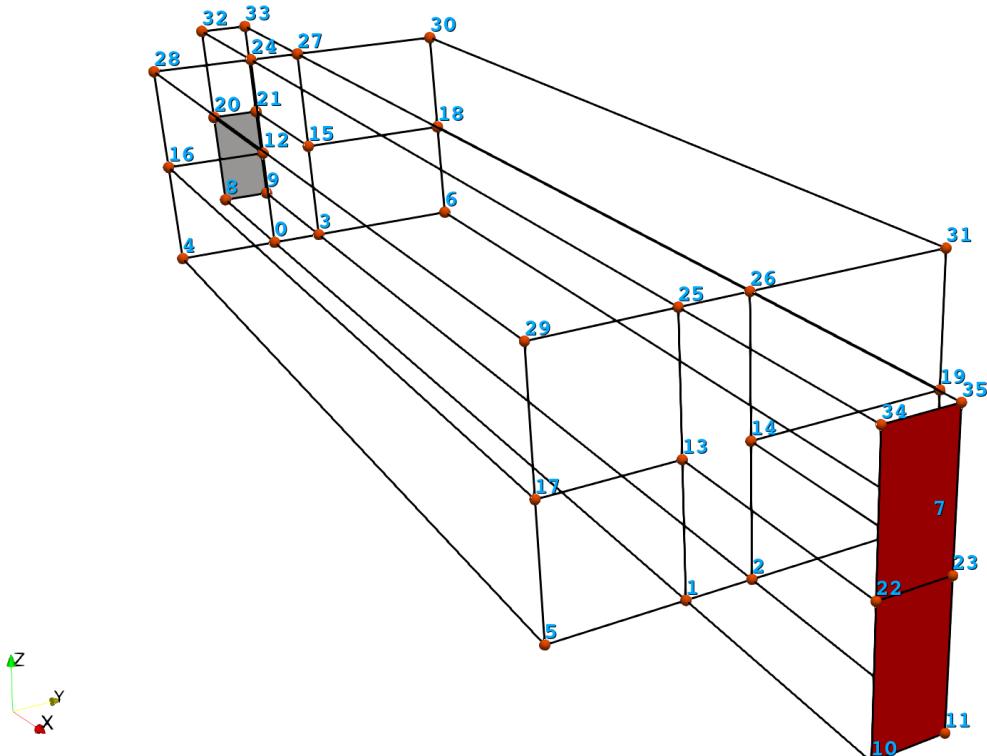


Multiblock Mesh

case03 - Blocks connections

- Boundaries - outlet

```
.../case03/system/blockMeshDict  
128.    ... outlet  
129.    {  
130.        type patch;  
131.        faces  
132.        (  
133.            // Z Down  
134.            (10 /* */)  
135.            // Z Up  
136.            (22 /* */)  
137.        );  
138.    }  
...
```



Multiblock Mesh

case03 - Blocks connections

- Boundaries - atmosphere

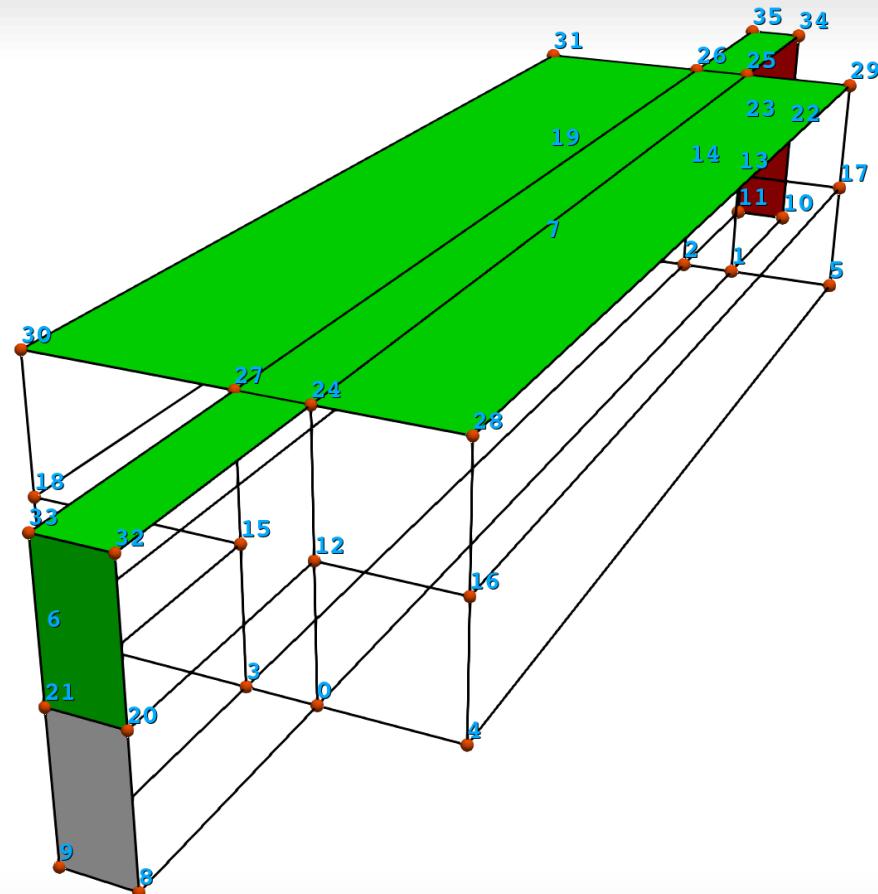
```
.../case03/system/blockMeshDict  
139. ... atmosphere  
140. {  
141.     type patch;  
142.     faces  
143.     (  
144.         // Z up near inlet  
145.         (21 /* */)  
146.  
147.  
148.         // Front  
149.         // Reservoir Mid  
150.         (24 /* */)  
151.         // Reservoir Low  
152.         (28 /* */)  
153.         // Reservoir Top  
154.         (27 /* */)  
155.         // Inflow channel  
156.         (32 /* */)  
157.         // Outflow Channel  
158.         (25 /* */)  
159.     );  
...}
```



CRheo@IPC



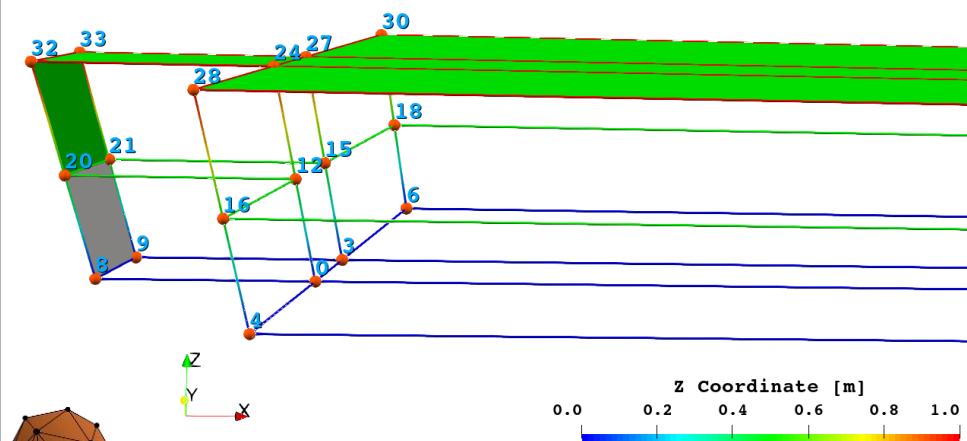
iPC



Multiblock Mesh

case03 - Blocks connections

- Boundaries - atmosphere

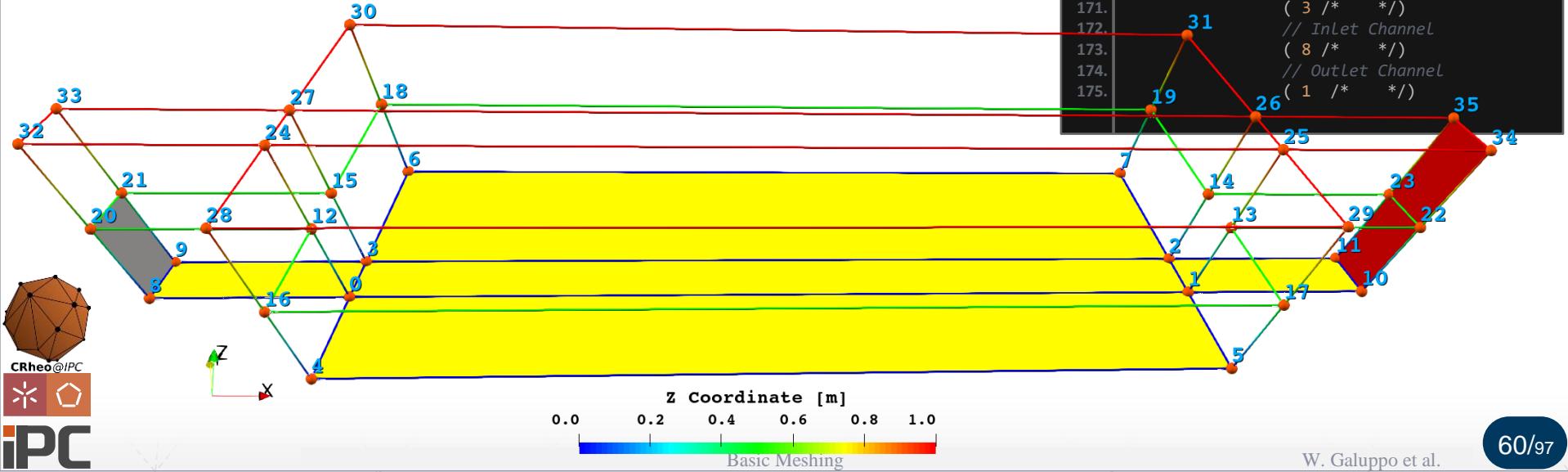


```
.../case03/system/blockMeshDict
139. ...
140. atmosphere
141. {
142. type patch;
143. faces
144. (
145. // Z up near inlet
146. (21 /* */)
147. // Front
148. // Reservoir Mid
149. (24 /* */)
150. // Reservoir Low
151. (28 /* */)
152. // Reservoir Top
153. (27 /* */)
154. // Inflow channel
155. (32 /* */)
156. // Outflow Channel
157. (25 /* */)
158. );
159. ...
...
```

Multiblock Mesh

case03 - Blocks connections

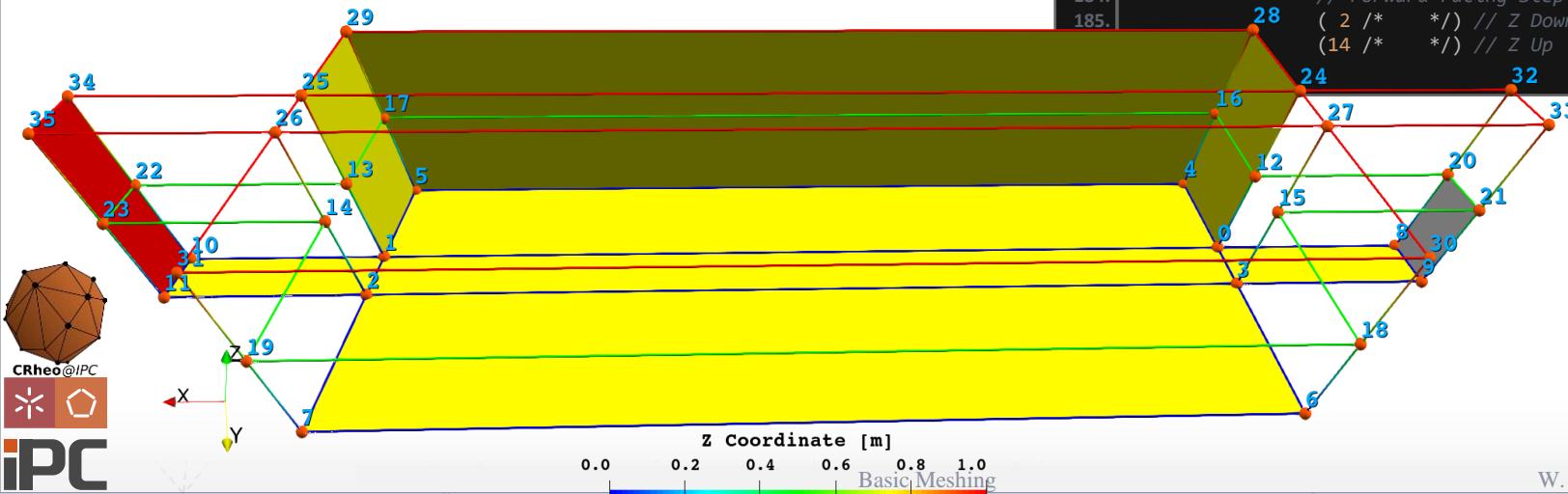
- Boundaries - walls



Multiblock Mesh

case03 - Blocks connections

- Boundaries - walls



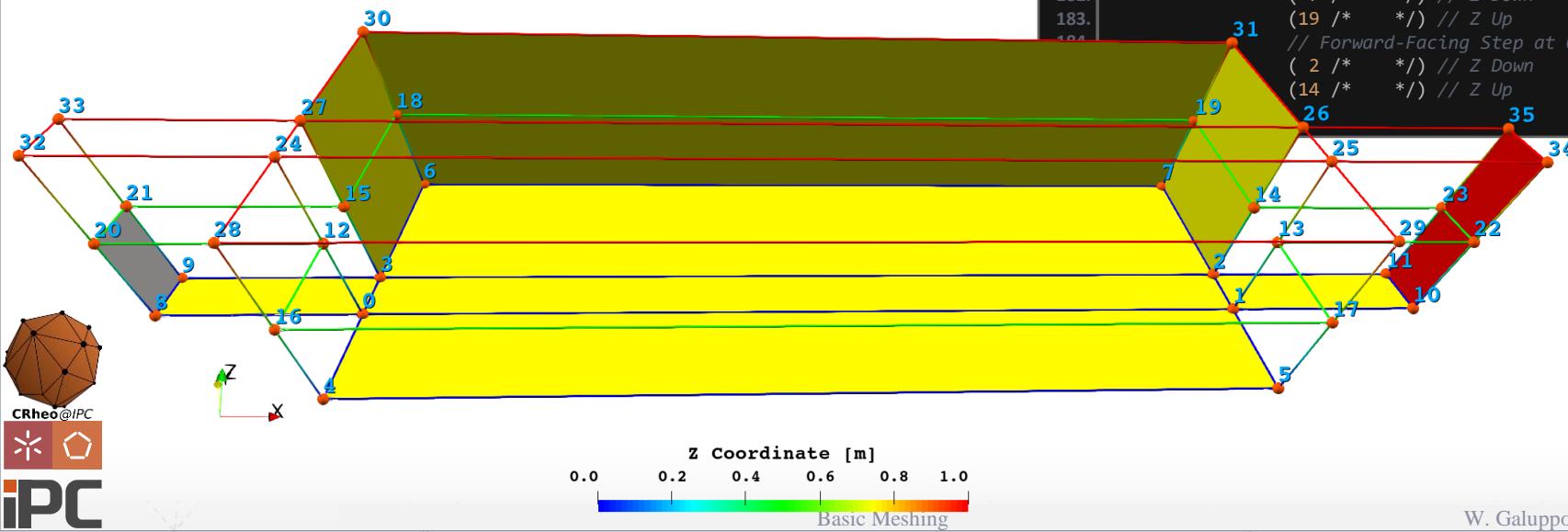
```
..//case03/system/blockMeshDict
```

```
160.     walls
161.     {
162.         type wall;
163.         faces
164.         (
165.             ...
166.             // Region: Reservoir Top
167.             // Backward-Facing Step at Inflow Channel
168.             ( 6 /*      */ ) // Z Down
169.             (18 /*      */ ) // Z Up
170.             // Parallel Facing Reservoir Low Wall Face
171.             ( 7 /*      */ ) // Z Down
172.             (19 /*      */ ) // Z Up
173.             // Forward-Facing Step at Outflow Channel
174.             ( 2 /*      */ ) // Z Down
175.             (14 /*      */ ) // Z Up
```

Multiblock Mesh

case03 - Blocks connections

- Boundaries - walls



```
./case03/system/blockMeshDict
```

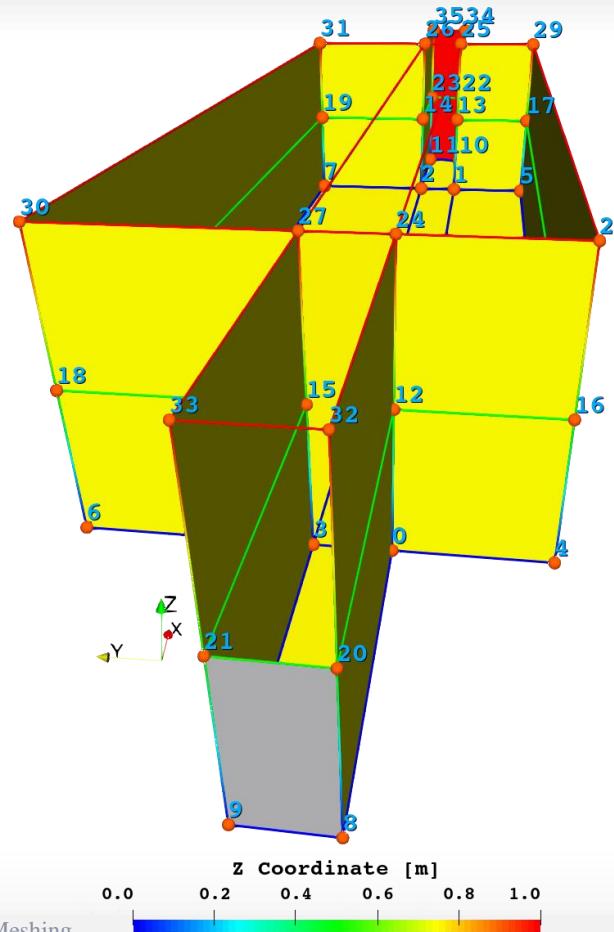
Multiblock Mesh

case03 - Blocks connections

- Boundaries - walls

```
.../case03/system/blockMeshDict
```

```
160.  
161.  
162.  
163.  
164.  
  
199.  
200.  
201.  
202.  
203.  
204.  
205.  
206.  
  
    ...  
    walls  
    {  
        type wall;  
        faces  
        (  
            ...  
            // Inflow channel walls  
            // Z Down  
            ( 8 0 12 20)  
            ( 3 9 21 15)  
            // Z Up  
            (20 12 24 32)  
            (15 21 33 27)  
            ...  
        );  
    }  
}
```



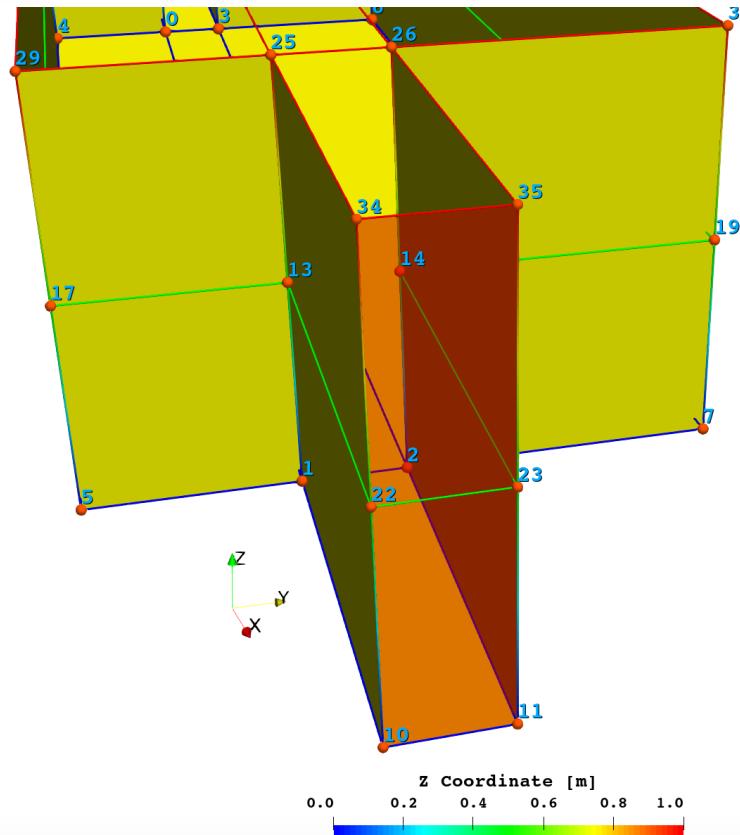
Multiblock Mesh

case03 - Blocks connections

- Boundaries - walls

```
.../case03/system/blockMeshDict
```

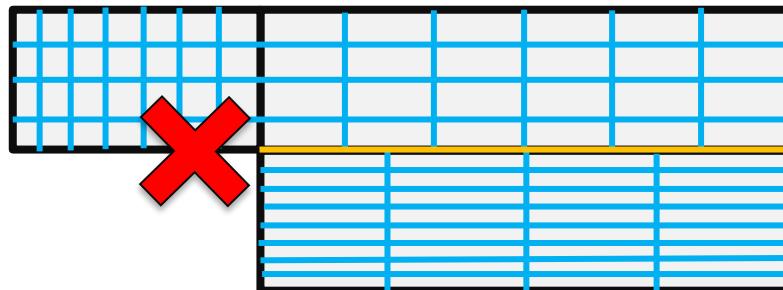
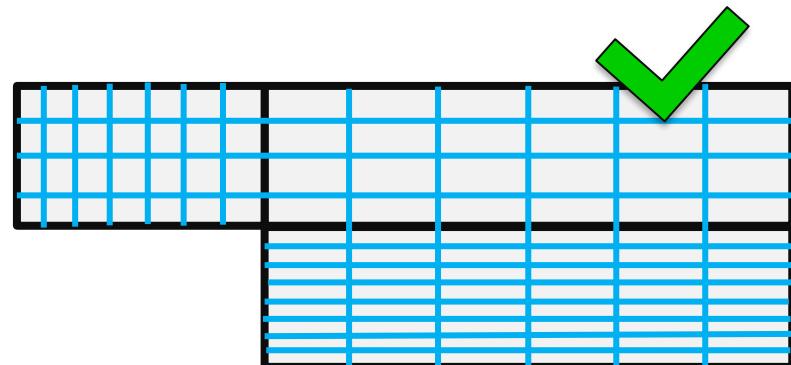
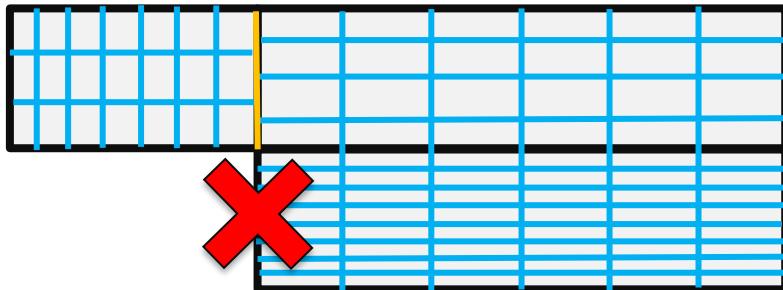
```
160.  
161.  
162.  
163.  
164.  
165.  
166.  
167.  
168.  
169.  
170.  
171.  
172.  
173.  
174.  
175.  
176.  
177.  
178.  
179.  
180.  
181.  
182.  
183.  
184.  
185.  
186.  
187.  
188.  
189.  
190.  
191.  
192.  
193.  
194.  
195.  
196.  
197.  
198.  
199.  
200.  
201.  
202.  
203.  
204.  
205.  
206.  
207.  
208.  
209.  
210.  
211.  
212.  
213.  
214.  
215.  
216.
```



Multiblock Mesh

case03 - Blocks connections

- Slicing and Connecting blocks



- Slicing and Connecting blocks + Setup using Macros in Expressions syntax

```

96. blocks
97. ( // hex (      Vertices indices to create a block      ) ( slicing the block      simpleGrading ( expansion Ratio factor )
98. // Block Down
99.     hex ( /* z = 0.0: */ 0 1 2 3 /* z = 0.5: */ 12 13 14 15 ) ( 1 1 1 1 ) simpleGrading ( 1 1 1 ) // block id: 0 - ReservMid
100.    hex ( /* z = 0.0: */ 4 5 1 0 /* z = 0.5: */ 16 17 13 12 ) ( 1 1 1 1 ) simpleGrading ( 1 1 1 ) // block id: 1 - ReservLow
101.    hex ( /* z = 0.0: */ 3 2 7 6 /* z = 0.5: */ 15 14 19 18 ) ( 1 1 1 1 ) simpleGrading ( 1 1 1 ) // block id: 2 - ReservTop
102.    hex ( /* z = 0.0: */ 8 0 3 9 /* z = 0.5: */ 20 12 15 21 ) ( 1 1 1 1 ) simpleGrading ( 1 1 1 ) // block id: 3 - inflow channel
103.    hex ( /* z = 0.0: */ 1 10 11 2 /* z = 0.5: */ 13 22 23 14 ) ( 1 1 1 1 ) simpleGrading ( 1 1 1 ) // block id: 4 - outflow channel
104.

```

```

83. // Mesh Refinements User Input Controls
84.
85. // X - Direction slicings
86. NXI 25; // Number of cell in the x direction of the inflow Channel
87. NXR 125; // Number of cell in the x direction of the Reservoir Section
88. NXO 25; // Number of cell in the x direction of the outflow Channel
89. // Y - Direction slicings
90. NYL 20; // Number of cell in the y direction of the Lower Reservoir
91. NYM 9; // Number of cell in the y direction of the Middle Reservoir
92. NYT 30; // Number of cell in the y direction of the Top Reservoir
93. // Z - Direction slicings
94. Nz 10; // Z direction Channel depth
95.

```

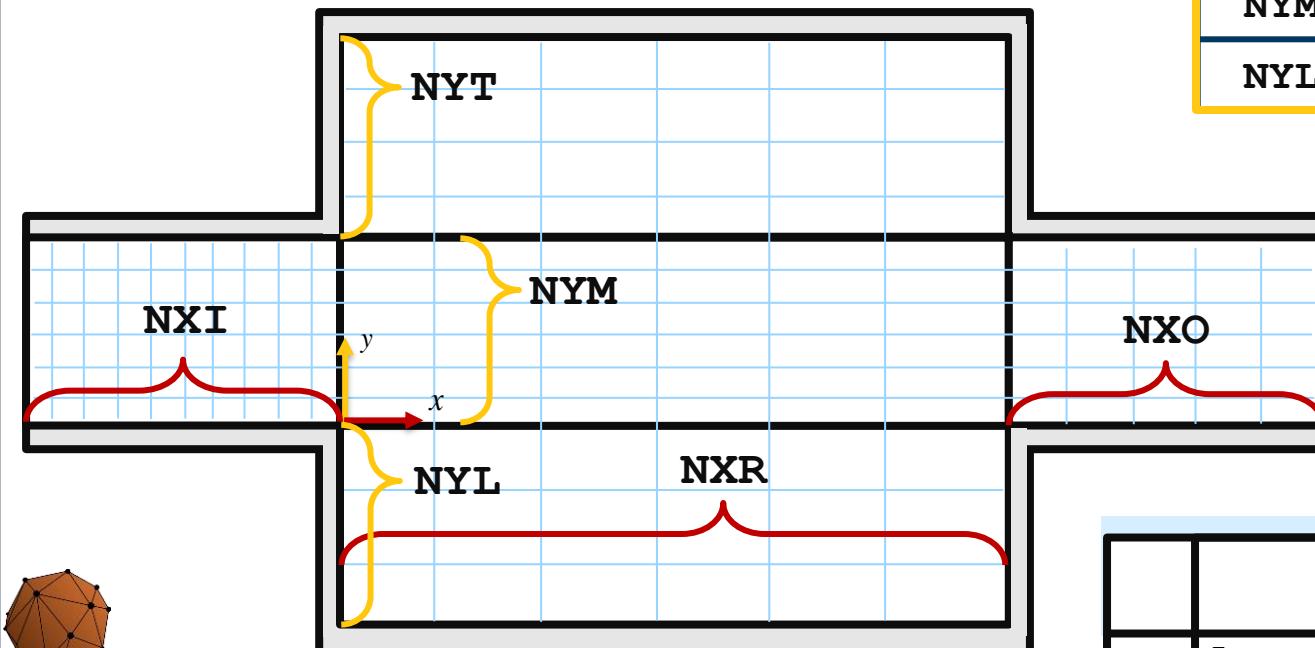


\$NXI

Multiblock Mesh

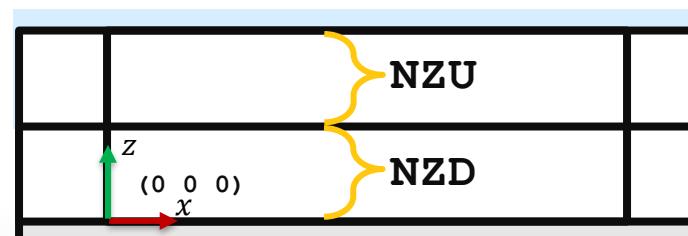
case03 - Blocks connections

- Slicing and Connecting blocks



NYT	30	NXI	25
NYM	9	NXR	125
NYL	20	NXO	25

$NZU=NZD=NZ$ 10



CRheo@IPC



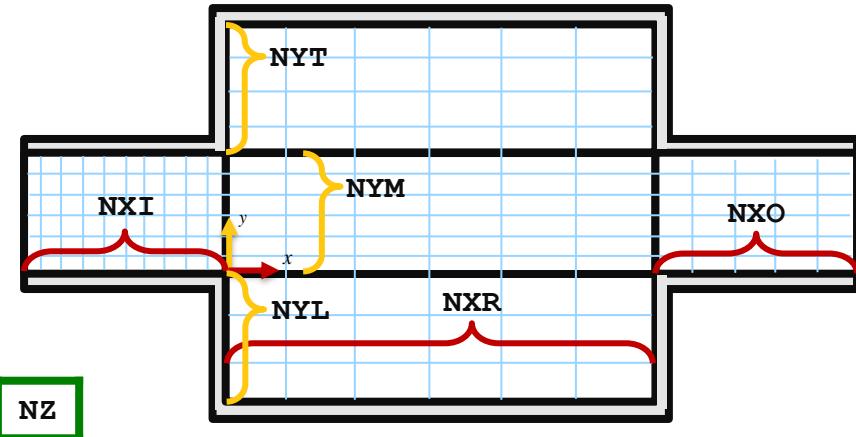
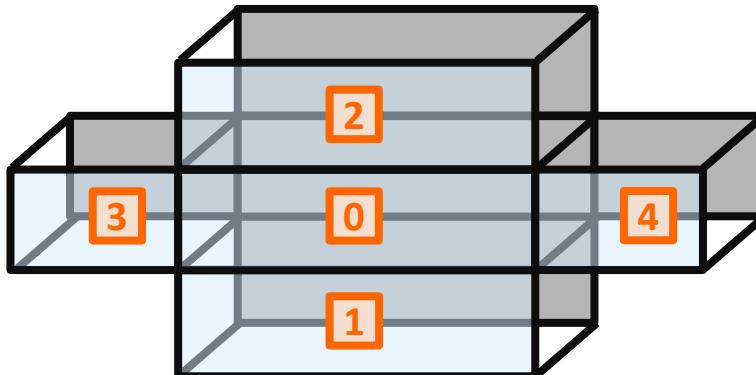
IPC

Multiblock Mesh

case03 - Blocks connections

- Slicing and Connecting blocks

```
96. blocks
97. ( // hex (      Vertices indices to create a block      ) ( slicing the block ) simpleGrading ( expansion Ratio factor )
98. // Block Down
99.     hex ( /* z = 0.0: */ 0 1 2 3 /* z = 0.5: */ 12 13 14 15 ) (
100.        simpleGrading ( 1 1 1 ) // block id: 0 - ReservMid
101.        hex ( /* z = 0.0: */ 4 5 1 0 /* z = 0.5: */ 16 17 13 12 ) (
102.            simpleGrading ( 1 1 1 ) // block id: 1 - ReservLow
103.            hex ( /* z = 0.0: */ 3 2 7 6 /* z = 0.5: */ 15 14 19 18 ) (
104.                simpleGrading ( 1 1 1 ) // block id: 2 - ReservTop
105.                hex ( /* z = 0.0: */ 8 0 3 9 /* z = 0.5: */ 20 12 15 21 ) (
106.                    simpleGrading ( 1 1 1 ) // block id: 3 - inflow channel
107.                    hex ( /* z = 0.0: */ 1 10 11 2 /* z = 0.5: */ 13 22 23 14 ) (
108.                        simpleGrading ( 1 1 1 ) // block id: 4 - outflow channel
109. 
```

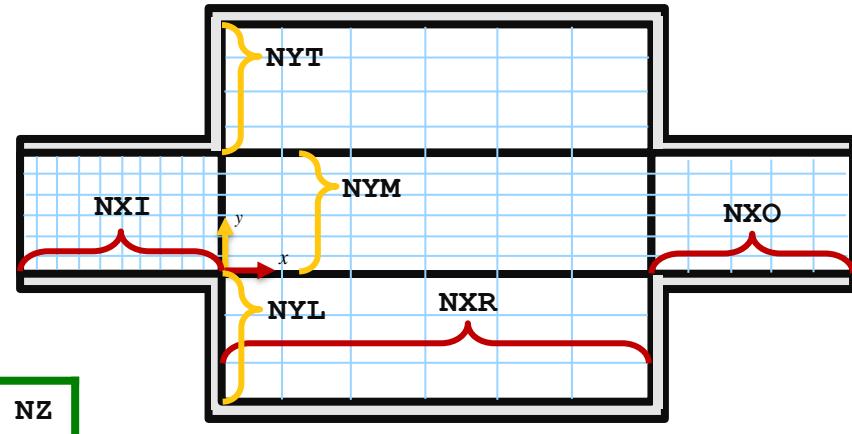
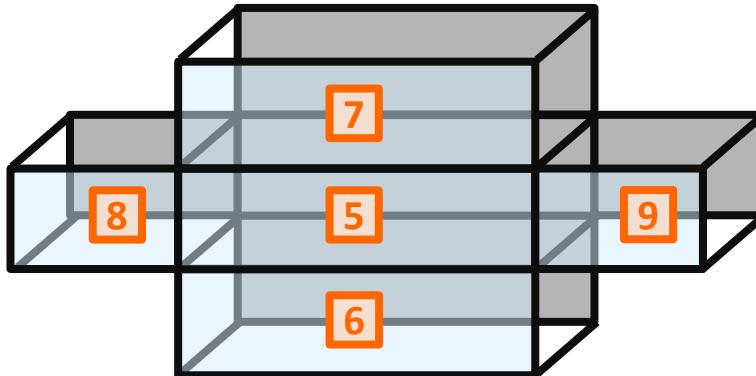


Multiblock Mesh

case03 - Blocks connections

- Slicing and Connecting blocks

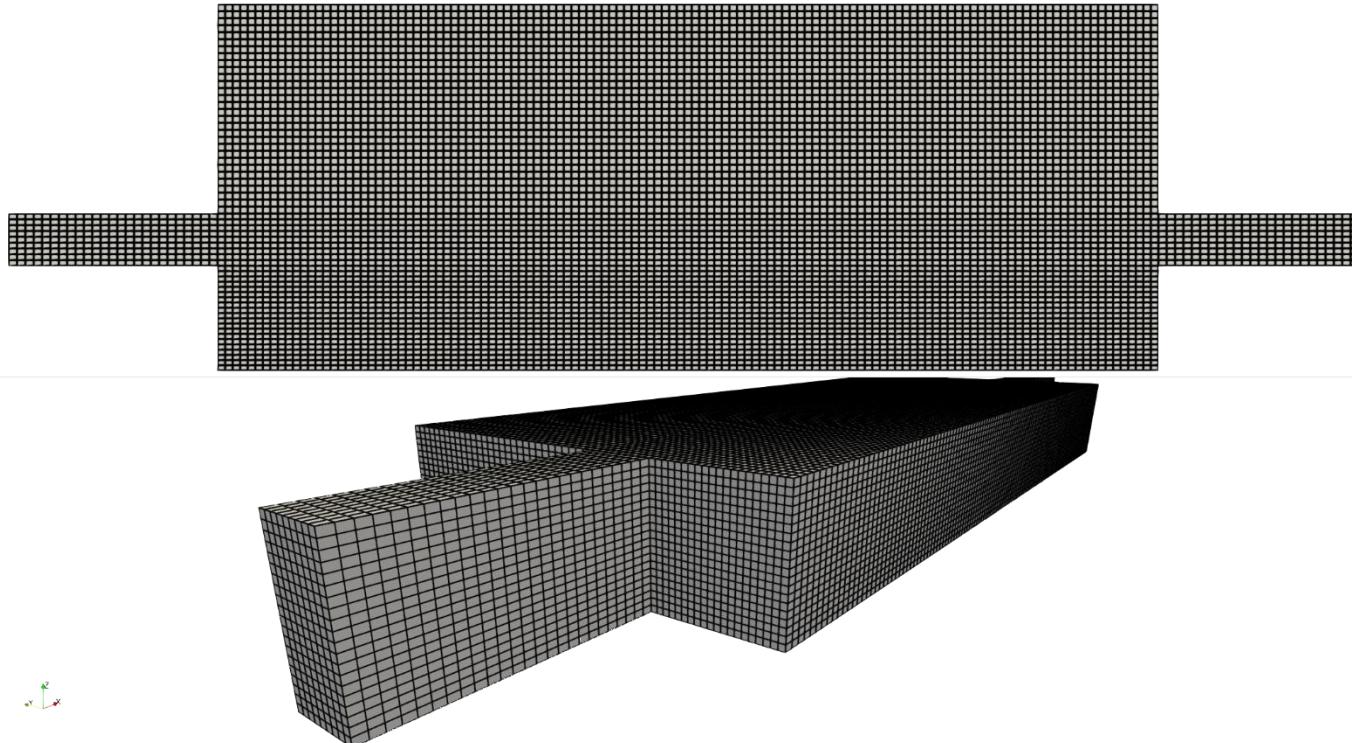
```
104.  
105. ...  
106. // Block Up  
107. hex ( /* z = 0.5: */ 12 13 14 15 /* z = 1.0: */ 24 25 26 27 ) ( ) simpleGrading ( 1 1 1 ) // block id: 5 - ReservMid  
108. hex ( /* z = 0.5: */ 16 17 13 12 /* z = 1.0: */ 28 29 25 24 ) ( ) simpleGrading ( 1 1 1 ) // block id: 6 - ReservLow  
109. hex ( /* z = 0.5: */ 15 14 19 18 /* z = 1.0: */ 27 26 31 30 ) ( ) simpleGrading ( 1 1 1 ) // block id: 7 - ReservTop  
110. hex ( /* z = 0.5: */ 20 12 15 21 /* z = 1.0: */ 32 24 27 33 ) ( ) simpleGrading ( 1 1 1 ) // block id: 8 - inflow channel  
111. hex ( /* z = 0.5: */ 13 22 23 14 /* z = 1.0: */ 25 34 35 26 ) ( ) simpleGrading ( 1 1 1 ) // block id: 9 - outflow channel  
111. );
```



Multiblock Mesh

case03 - Blocks connections

- Created Mesh



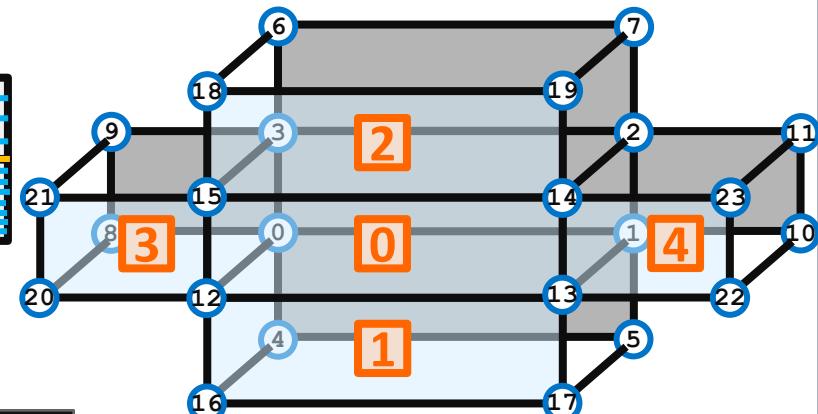
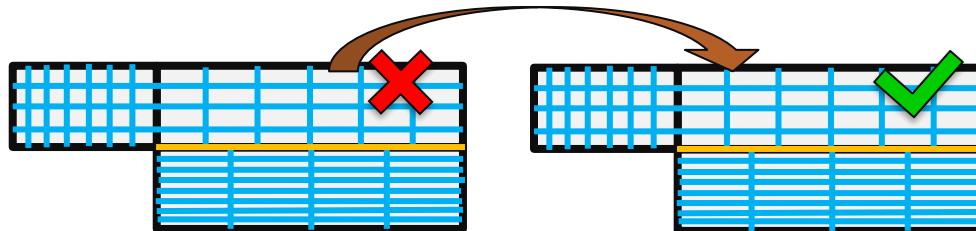
CRheo@IPC



Multiblock Mesh

case03 - Blocks connections

- If required, there is a way:



weldedCase03

```
boundary
(
    ...
    surfaceToWeldBlock0
    {
        type patch;
        faces
        (
            ( 0  1 13 12)
        );
    }
    ...
);
```

```
boundary
(
    ...
    surfaceToWeldBlock1
    {
        type patch;
        faces
        (
            (24 25 27 26)
        );
    }
);
```

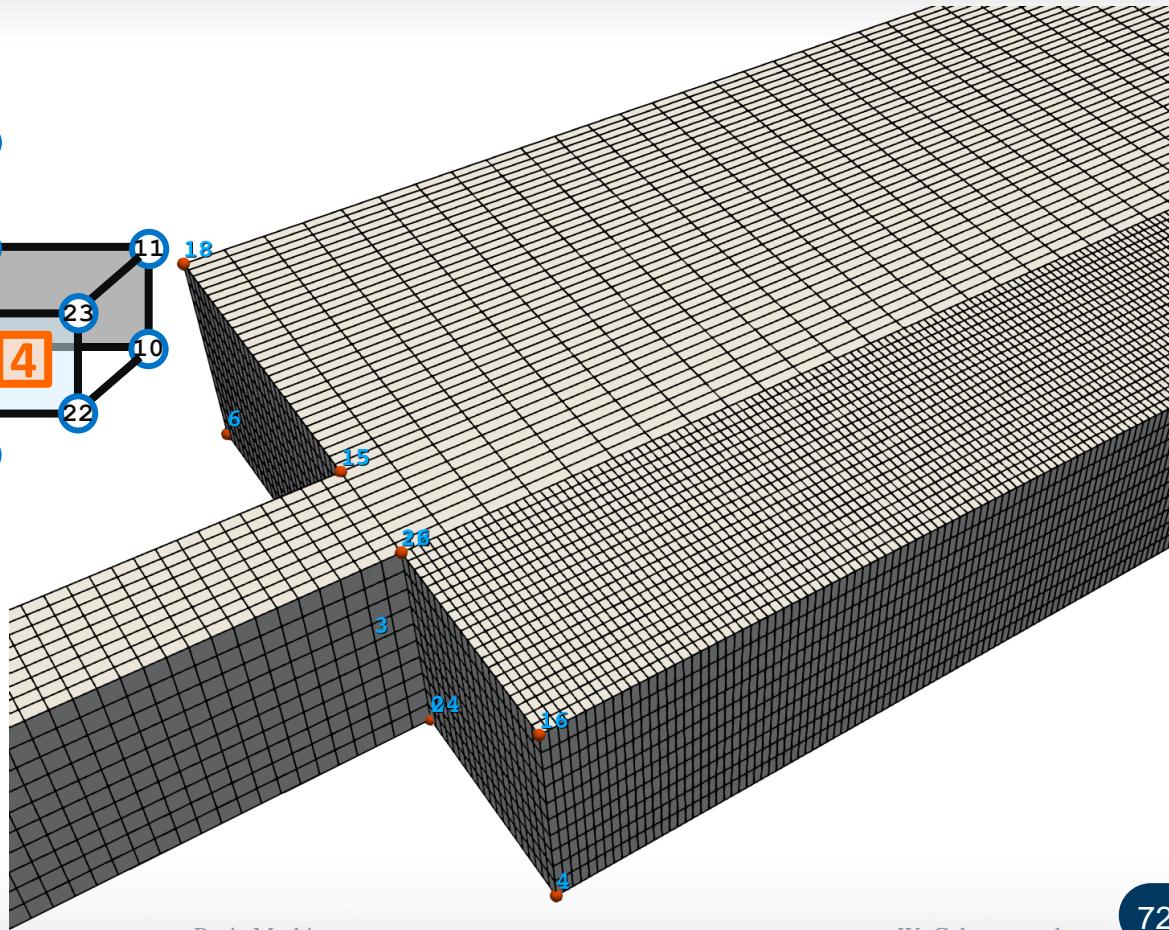
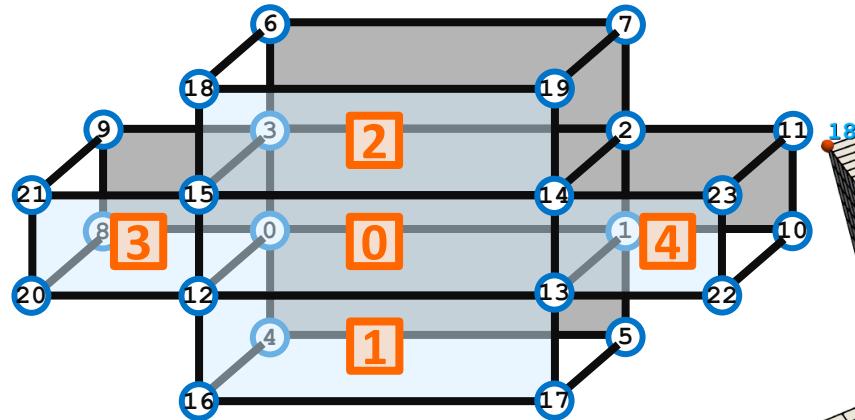
```
mergePatchPairs
(
    (surfaceToWeldBlock0 surfaceToWeldBlock1)
);
```



Multiblock Mesh

case03 - Blocks connections

- Created Mesh



4

Curved Surfaces

Designing curvilinear regions

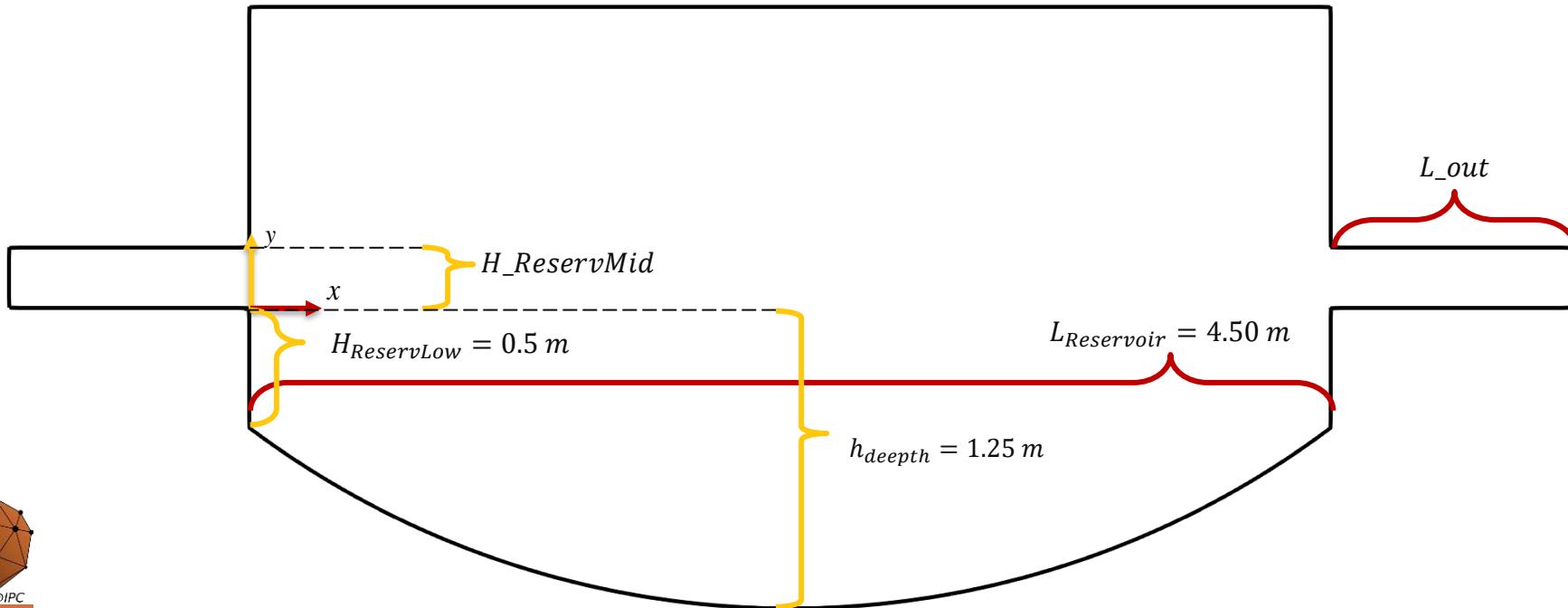
```
cd <Desktop>/FOAM@Iberia-Courses/Beginner/C2/tutorials  
cp -r ./caseTutorials/case04 ./  
cd case04  
code .
```



CRheo@IPC



- The previous geometry can sometimes be as the following shape



```
.../case04/system/blockMeshDict
```

```
113. // Curved Surface for block at ReservLow
114.
115.     // Method 01:
116.         // Arcs Mid Points
117.         pXVA    2.25;
118.         pYVA   -1.25;
119.
120.     edges
121.     (
122.         arc 4 5 ( $pXVA $pYVA 0.0)
123.         arc 16 17 ( $pXVA $pYVA 0.5)
124.         arc 28 29 ( $pXVA $pYVA 1.0)
125.     );
126.
127.     // Method 02:
128.         // Arcs Circle Center Points
129.         radius 3.75;
130.         cXV1 #eval "0.5*4.50";
131.         cYV1 #eval "$radius - 1.25";
132.         // //Degrees to Radians
133.         // AlphaStart #calc "degToRad($AlphaStartDeg)";
134.         // AlphaEnd   #calc "degToRad($AlphaEndDeg)";
135.
136.     edges
137.     (
138.         arc 4 5 origin ( $cXV1 $cYV1 0.0)
139.         arc 16 17 origin ( $cXV1 $cYV1 0.5)
140.         arc 28 29 origin ( $cXV1 $cYV1 1.0)
141.     );
```



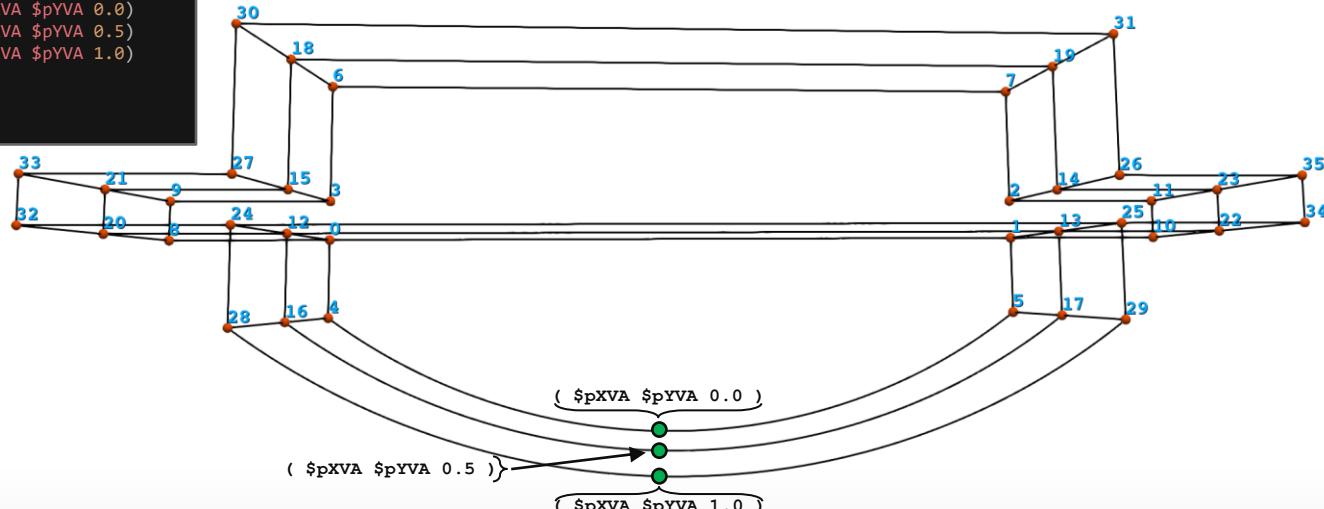
CRheo@IPC



- Method 1:

```
113. // Curved Surface for block at ReservLow
114.
115. // Method 01:
116.     // Arcs Mid Points
117.     pXVA    2.25;
118.     pYVA   -1.25;
119.
120. edges
121.
122.     (
123.         arc 4 5 ( $pXVA $pYVA 0.0)
124.         arc 16 17 ( $pXVA $pYVA 0.5)
125.         arc 28 29 ( $pXVA $pYVA 1.0)
126.
...

```



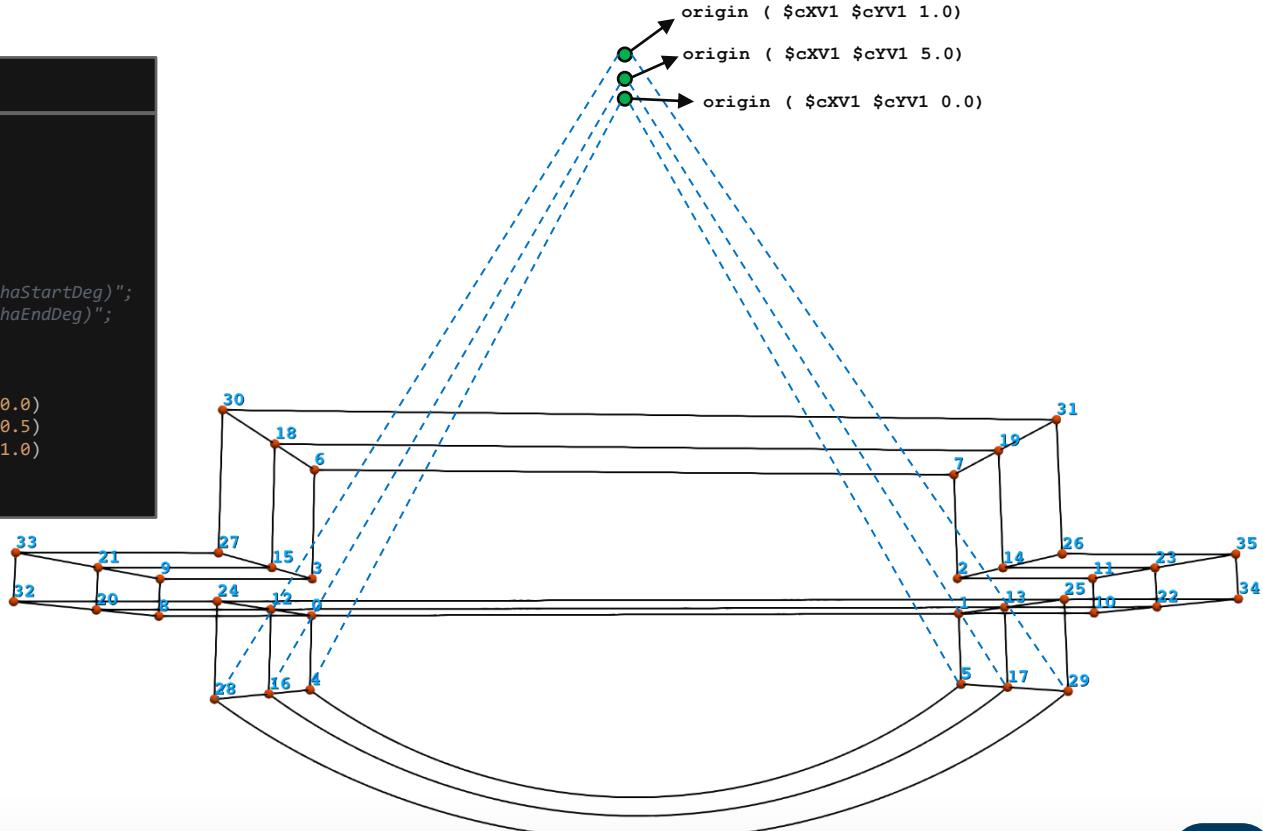
Curved Surfaces

Designing curvilinear regions

- Method 2:

```
.../case04/system/blockMeshDict
```

```
127. // Method 02:  
128. // Arcs Circle Center Points  
129. radius 3.75;  
130. cXV1 "#eval \"0.5*4.50\";  
131. cYV1 "#eval \"$radius - 1.25\";  
132. // //Degrees to Radians  
133. // AlphaStart "#calc "degToRad($AlphaStartDeg)";  
134. // AlphaEnd "#calc "degToRad($AlphaEndDeg)";  
135.  
136.  
137. edges  
(  
    arc 4 5 origin ( $cXV1 $cYV1 0.0)  
    arc 16 17 origin ( $cXV1 $cYV1 0.5)  
    arc 28 29 origin ( $cXV1 $cYV1 1.0)  
);  
141.  
142.
```



CRheo@IPC



- Mesh metrics

```
checkMesh || checkMesh -writeAllFields
```

```

23. Create time
24.
25. Create mesh for time = 0
26.
27. Check mesh...
28.
29. Time = 0
30.
31. Mesh stats
32.   points:      169260
33.   faces:       482005
34.   internal faces: 456995
35.   cells:        156500
36.   faces per cell: 6
37.   boundary patches: 4
38.   point zones:    0
39.   face zones:     0
40.   cell zones:     0
41.
42. Overall number of cells of each type:
43.   hexahedra:    156500
44.   prisms:        0
45.   wedges:        0
46.   pyramids:      0
47.   tet wedges:    0
48.   tetrahedra:    0
49.   polyhedra:     0
50.
51. Checking topology...
52.   Boundary definition OK.
53.   Cell to face addressing OK.
54.   Point usage OK.
55.   Upper triangular ordering OK.
56.   Face vertices OK.
57.   Number of regions: 1 (OK).
58.

59. Checking patch topology for multiply connected surfaces...
60.   Patch      Faces   Points   Surface topology
61.   inlet       90      110      ok (non-closed singly connected)
62.   outlet      180      210      ok (non-closed singly connected)
63.   atmosphere  7915     8160     ok (non-closed singly connected)
64.   walls       16825    17100    ok (non-closed singly connected)
65.   "*"         25010    25012    ok (closed singly connected)
66.
67.
68. Checking faceZone topology for multiply connected surfaces...
69.   No faceZones found.
70.
71. Checking basic cellZone addressing...
72.   No cellZones found.
73.
74. Checking basic pointZone addressing...
75.   No pointZones found.
76.
77. Checking geometry...
78.   Overall domain bounding box (-1 -1.24995 0) (5.5 1 1)
79.   Mesh has 3 geometric (non-empty/wedge) directions (1 1 1)
80.   Mesh has 3 solution (non-empty) directions (1 1 1)
81.   Boundary openness (-4.86392e-17 -2.39761e-15 -4.55032e-14) OK.
82.   Max cell openness = 2.13586e-16 OK.
83.   Max aspect ratio = 3.00418 OK.
84.   Minimum face area = 0.000798996. Maximum face area = 0.00312488. Face area magnitudes OK.
85.   Min volume = 3.99498e-05. Max volume = 0.000120447. Total volume = 9.54907. Cell volumes OK.
86.   Mesh non-orthogonality Max: 34.9365 average: 4.77889
87.   Non-orthogonality check OK.
88.   Face pyramids OK.
89.   Max skewness = 0.903677 OK.
90.   Coupled point location match (average 0) OK.
91.
92. Mesh OK.
93.
94. End

```



CRheo@IPC



Curved Surfaces

Designing curvilinear regions

- Mesh metrics

```
checkMesh -writeAllFields
```

```
94. Writing fields with mesh quality parameters
95.     Writing non-orthogonality (angle) to nonOrthoAngle
96.     Writing face interpolation weights (0..0.5) to faceWeight
97.     Writing face skewness to skewness
98.     Writing cell determinant to cellDeterminant
99.     Writing aspect ratio to aspectRatio
100.    Writing approximate aspect ratio to cellAspectRatio
101.    Writing cell shape (hex, tet etc.) to cellShapes
102.    Writing cell volume to cellVolume
103.    Writing cell volume ratio to cellVolumeRatio
104.    Writing minTetVolume to minTetVolume
105.    Writing minPyrVolume to minPyrVolume
106.    Writing cell region to cellRegion
107.    Writing cell zoning to cellZone
108.    Writing face zoning to faceZone
109.
110.
111. End
```

```
$ tree -L 3 ./case04
```

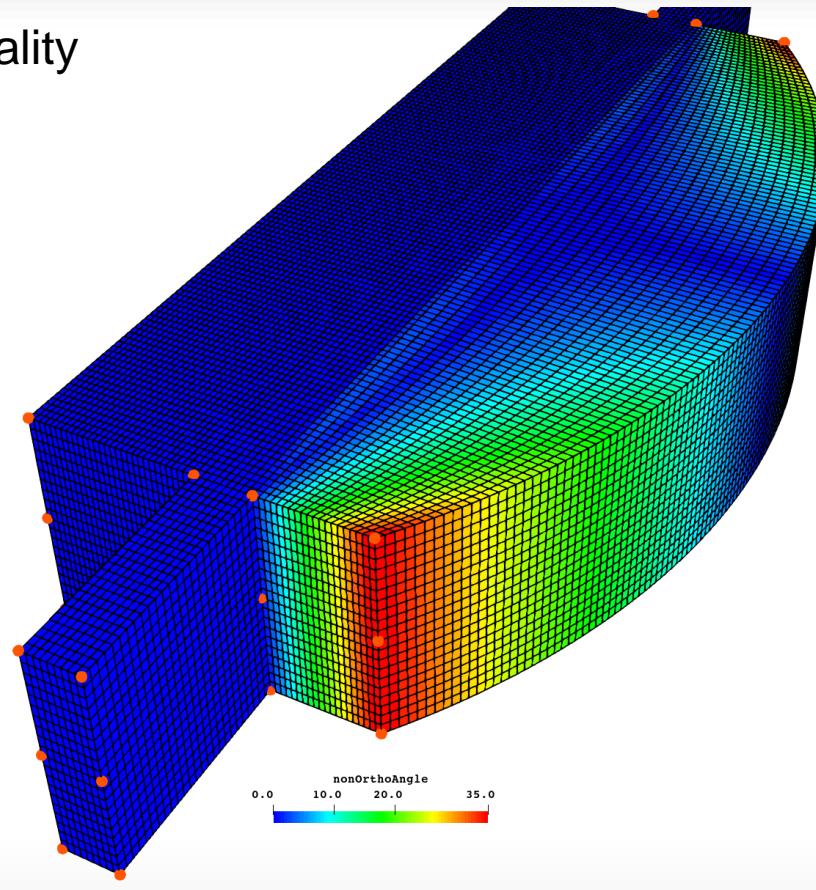
```
.
├── 0
│   ├── aspectRatio
│   ├── cellAspectRatio
│   ├── cellDeterminant
│   ├── cellRegion
│   ├── cellShapes
│   ├── cellVolume
│   ├── cellVolumeRatio
│   ├── cellZone
│   ├── faceWeight
│   ├── faceZone
│   ├── minPyrVolume
│   ├── minTetVolume
│   └── nonOrthoAngle
        └── skewness
└── constant
    ├── polyMesh
    └── transportProperties
...
```



CRheo@IPC



- Mesh metrics – Non-Orthogonality



Basic Meshing



CRheo@IPC



5

Case Study Mesh

case05 - Interconnected Course Mesh

```
cd <Desktop>/FOAM@Iberia-Courses/Beginner/C2/tutorials  
cp -r ./caseTutorials/case05 ./  
cd case05  
code .
```



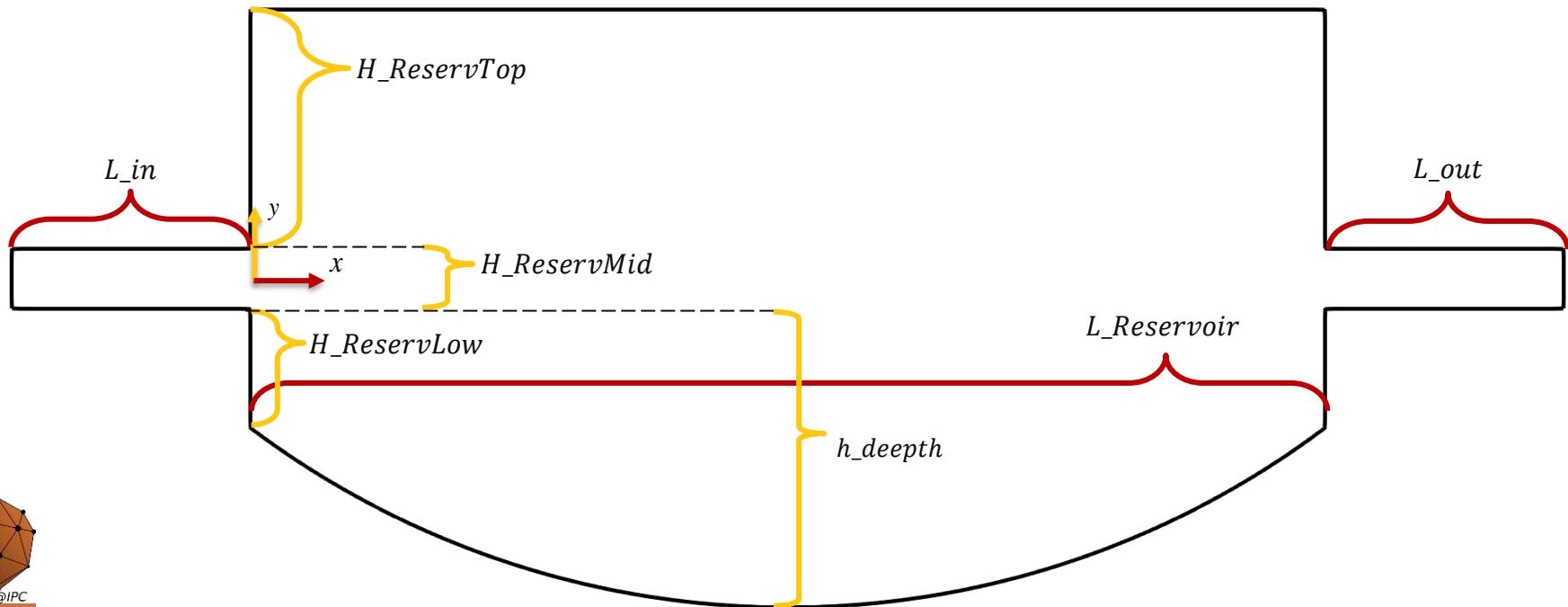
CRheo@IPC



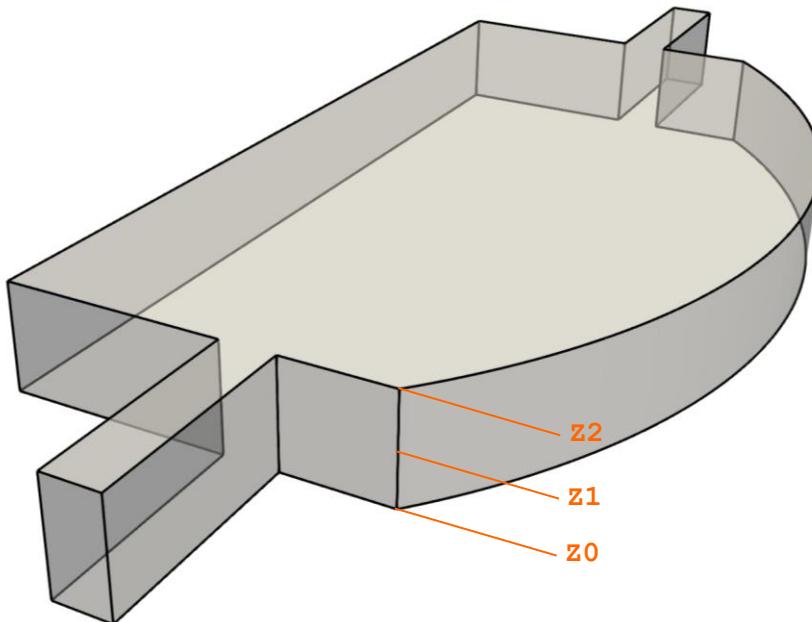
Case Study Mesh

case05 - Interconnected Course Mesh

- The parametrized geometry



- The parametrized geometry



Case Study Mesh

case05 - Interconnected Course Mesh

./case05/system/blockMeshDict

```

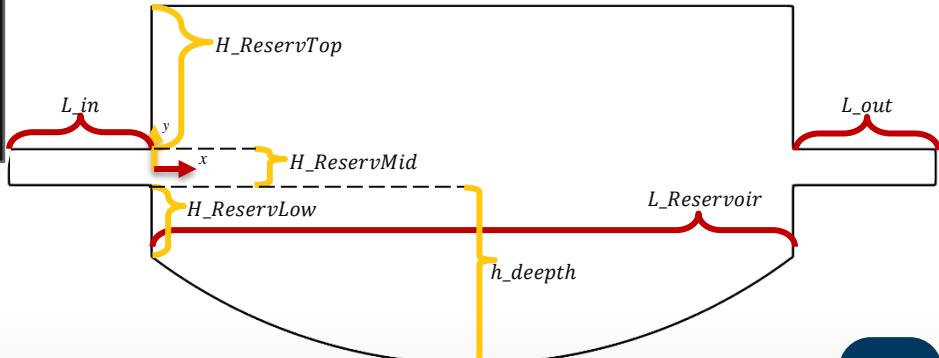
17. scale 1; // convertToMeters 1;
18.
19. // The mesh parametrization is before the "blocks" section:
20. // Geometry Parametrization
21.
22. /*-----
23. /*User input Section -----*/
24.
25. // Lengths:
26. L_in      1.00;
27. L_out     1.00;
28. L_Reservoir 4.50;
29.
30. // Heights:
31. H_ReservTop 1.00;
32. H_ReservMid 0.25;
33. H_ReservLow 0.50;
34.
35. h_depth    1.25;
36.
37. // Depths:
38. z0 0.00; // floor plane
39. z1 0.25; // Middle plane
40. z2 0.50; // Upper plane
41.
42.
43. /*-----
44. /*- END UI SECTION -----*/

```

```

45. /*
46. /*Mathematical Operations-----*/
47.
48. // Global Coordinate Positions Calculations
49.
50. // YMid #calc "0.5*$H_ReservMid";
51. YMid   #eval "0.5*$H_ReservMid";
52. YLowNeg #eval "(-1*$H_ReservLow)+(-$YMid)";
53. YTopPos #eval "($H_ReservTop)+($YMid)";
54.
55. X_InChn #eval "-1*L_in";
56. X_Reserv #eval "$L_Reservoir";
57. X_OutChn #eval "$X_Reserv + $L_out";
58.
59. /*-----*/
60. /*- END Mth.O. SECTION -----*/
61.

```



Case Study Mesh

case05 - Interconnected Course Mesh

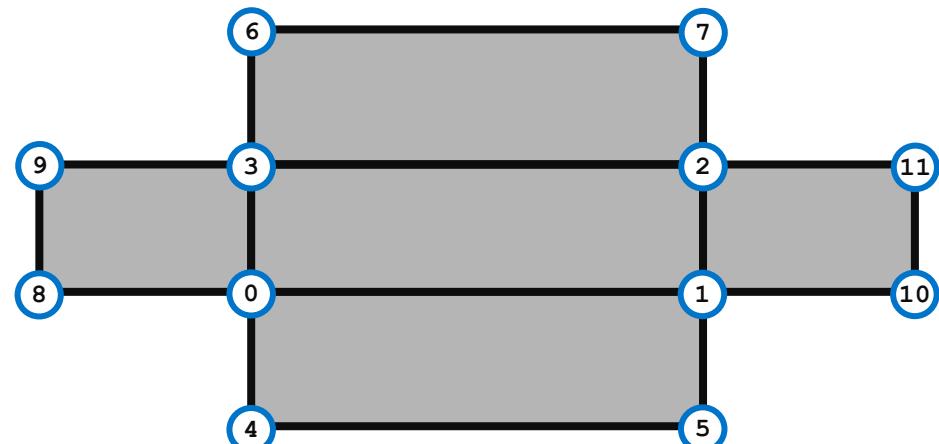
```
..../case05/system/blockMeshDict
```

```
64. vertices
65. (
66.     // z at z0:
67.
68.     // Reservoir Mid    0.0
69.     (    0.0      -$YMid    $z0 ) // id:  0
70.     ( $X_Reserv   -$YMid    $z0 ) // id:  1
71.     ( $X_Reserv   $YMid    $z0 ) // id:  2
72.     (    0.0      $YMid    $z0 ) // id:  3
73.
74.     // Reservoir Low
75.     (    0.0      $YLowNeg   $z0 ) // id:  4
76.     ( $X_Reserv   $YLowNeg   $z0 ) // id:  5
77.
78.     // Reservoir Top
79.     ( $X_Reserv   $YTopPos   $z0 ) // id:  6
80.     (    0.0      $YTopPos   $z0 ) // id:  7
81.
82.     // inlet
83.     ( $X_InChn   -$YMid    $z0 ) // id:  8
84.     ( $X_InChn   $YMid    $z0 ) // id:  9
85.
86.     // outlet
87.     ( $X_OutChn   -$YMid    $z0 ) // id: 10
88.     ( $X_OutChn   $YMid    $z0 ) // id: 11
89.
90.     // z at z1:
91.
92.         // Reservoir Mid
93.         (    0.0      -$YMid    $z1 ) // id: 12
...

```

```
51. ...
52.     YMid      #eval "0.5*$H_ReservMid";
53.     YLowNeg   #eval "(-1*$H_ReservLow)+(-$YMid)";
54.     YTopPos   #eval "($H_ReservTop)+($YMid)";
55. ...
56.     X_InChn   #eval "-1*$L_in";
57.     X_Reserv   #eval "$L_Reservoir";
58.     X_OutChn   #eval "$X_Reserv + $L_out";
...

```



CRheo@IPC



Case Study Mesh

case05 - Interconnected Course Mesh

./case05/system/blockMeshDict

```
...
// Mesh refinement parametrization controls

/*-----*/
/*User input Section -----*/
/*-----*/

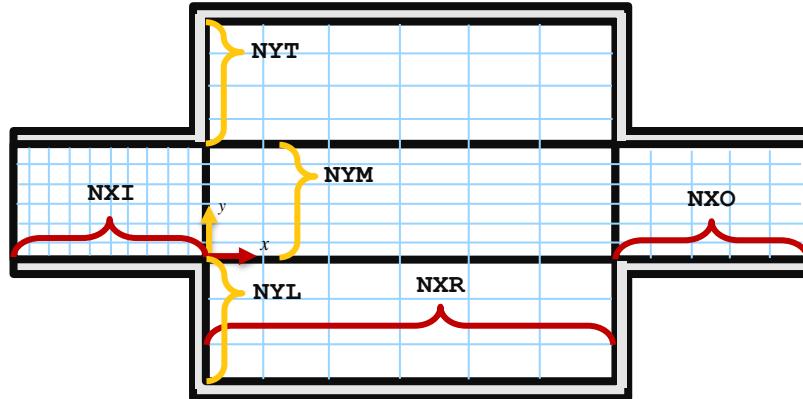
NXI 25; // Number of cell in the x direction of the inlet Channel
NXR 125; // Number of cell in the x direction of the Reservoir Section
NXO 25; // Number of cell in the x direction of the outlet Channel

NYL 20; // Number of cell in the y direction of the Lower Reservoir
NYM 9; // Number of cell in the y direction of the Middle Reservoir
NYT 30; // Number of cell in the y direction of the Top Reservoir

Nz 10; // Z direction Channel depth

/*-----*/
/*- END UI SECTION -----*/
/*-----*/

blocks
(
    // Reservoir Mid
    hex ( 0 1 2 3 12 13 14 15 ) ($NXR $NYM $Nz) simpleGrading (1 1 1) // block id: 0
    hex ( 12 13 14 15 24 25 26 27 ) ($NXR $NYM $Nz) simpleGrading (1 1 1) // block id: 1
    // Reservoir Low
    hex ( 4 5 1 0 16 17 13 12 ) ($NXR $NYL $Nz) simpleGrading (1 1 1) // block id: 2
    hex ( 16 17 13 12 28 29 25 24 ) ($NXR $NYL $Nz) simpleGrading (1 1 1) // block id: 3
    // Reservoir Top
    hex ( 3 2 6 7 15 14 18 19 ) ($NXR $NYT $Nz) simpleGrading (1 1 1) // block id: 4
    ...
)
```

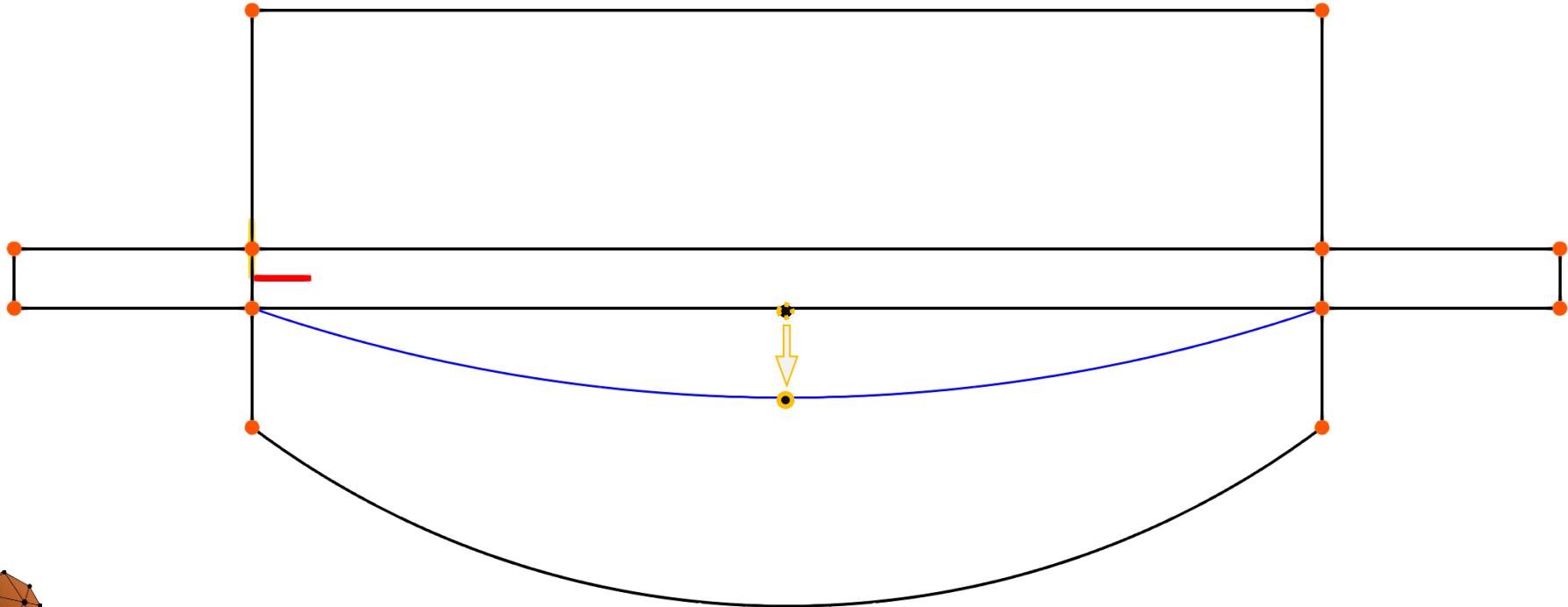


Cell Zone Generation

```
hex ( 3 2 6 7 15 14 18 19 ) cellZone01 ($NXR $NYT $Nz) ...
```

Case Study Mesh

case05 - Interconnected Course Mesh

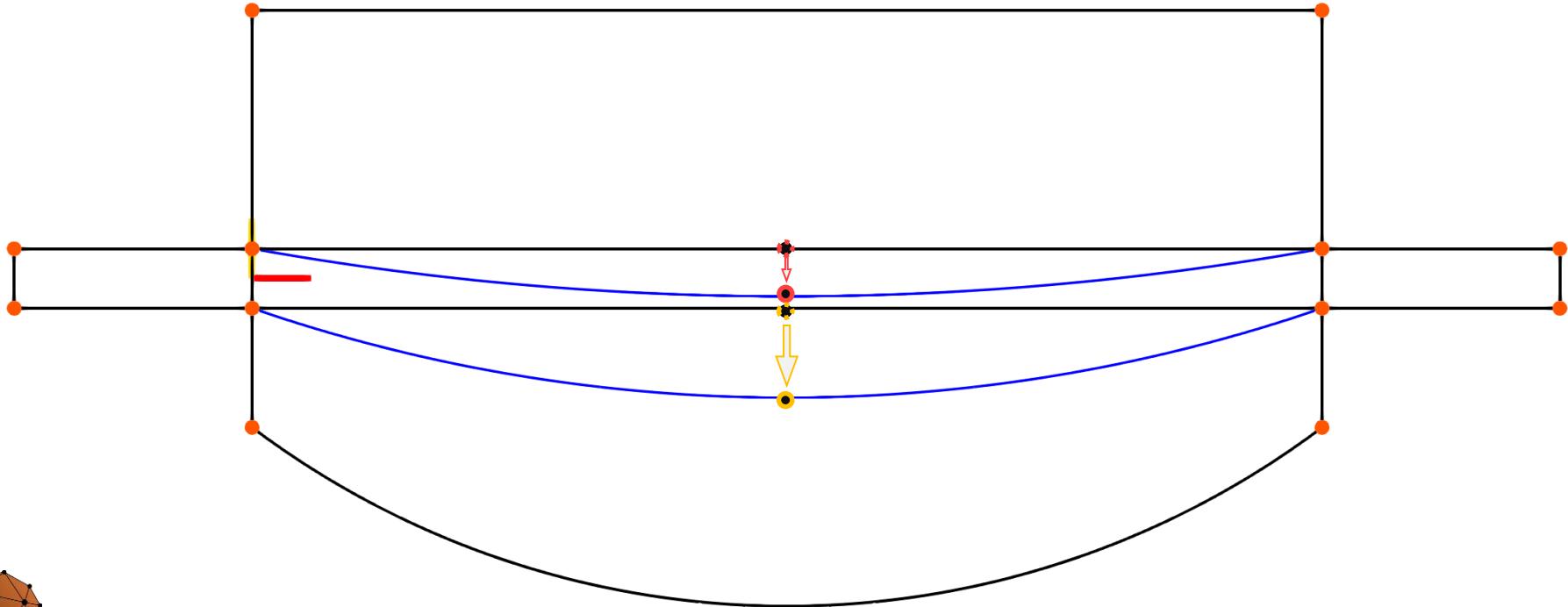


CRheo@IPC



Case Study Mesh

case05 - Interconnected Course Mesh



CRheo@IPC



Case Study Mesh

case05 - Interconnected Course Mesh

./case05/system/blockMeshDict

```

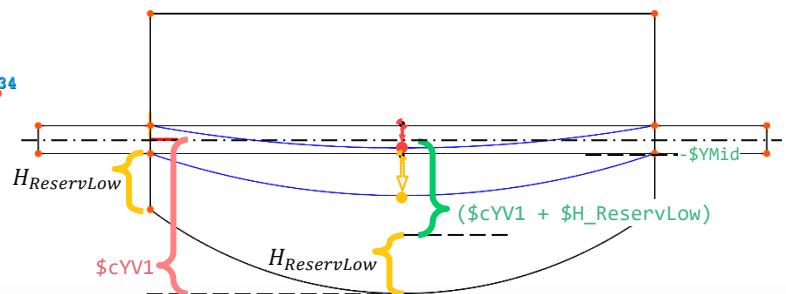
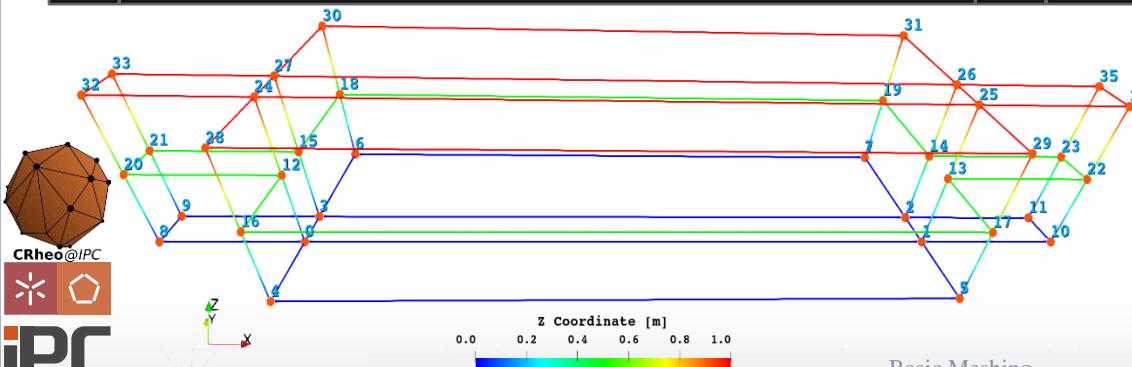
178. //Arc Parametrization:
179.
180. /*-----*/
181. /*Mathematical Operations-----*/
182.
183. // Curved Surface for block atReservoir Low
184.
185.     cXV1 #eval "$L_Reservoir*0.5";
186.     cYV1 #eval "(-1)*($YMid+$h_depth)";
187.
188. // Curved Surface for block atReservoir Mid
189. // Common surface between Low and Mid blocks
190.     cXV2 #eval "$cXV1";
191.     cYV2 #eval "((cYV1 + $H_ReservLow) + (-$YMid ))*1 / 2";
192.
193. // Common surface between Mid and Top blocks
194.     cXV3 #eval "$cXV1";
195.     cYV3 #eval "((cYV2 + $H_ReservMid) + ( $YMid ))*3 / 5";
196.
197. /*-----*/
198. /*- END Mth.O. SECTION -----*/

```

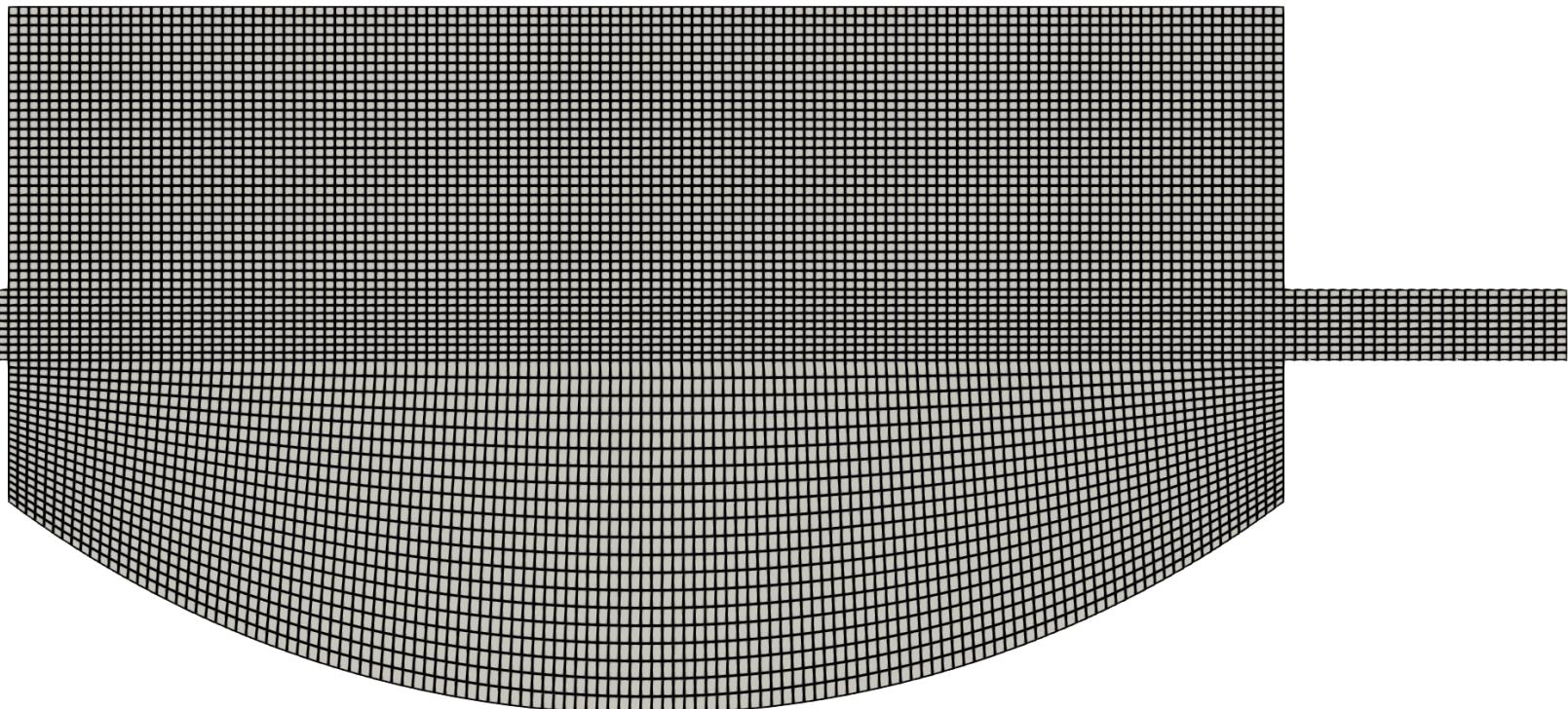
```

200. ...
201. edges
202. (
203.     // Reservoir Low
204.     arc 4 5 ($cXV1 $cYV1 $z0)
205.     arc 16 17 ($cXV1 $cYV1 $z1)
206.     arc 28 29 ($cXV1 $cYV1 $z2)
207.     // Reservoir Mid
208.     // Low/Mid blocks face
209.     arc 0 1 ($cXV2 $cYV2 $z0)
210.     arc 12 13 ($cXV2 $cYV2 $z1)
211.     arc 24 25 ($cXV2 $cYV2 $z2)
212.     // Top/Mid blocks face
213.     arc 3 2 ($cXV3 $cYV3 $z0)
214.     arc 15 14 ($cXV3 $cYV3 $z1)
215.     arc 27 26 ($cXV3 $cYV3 $z2)
216.
217. );
218. ...

```



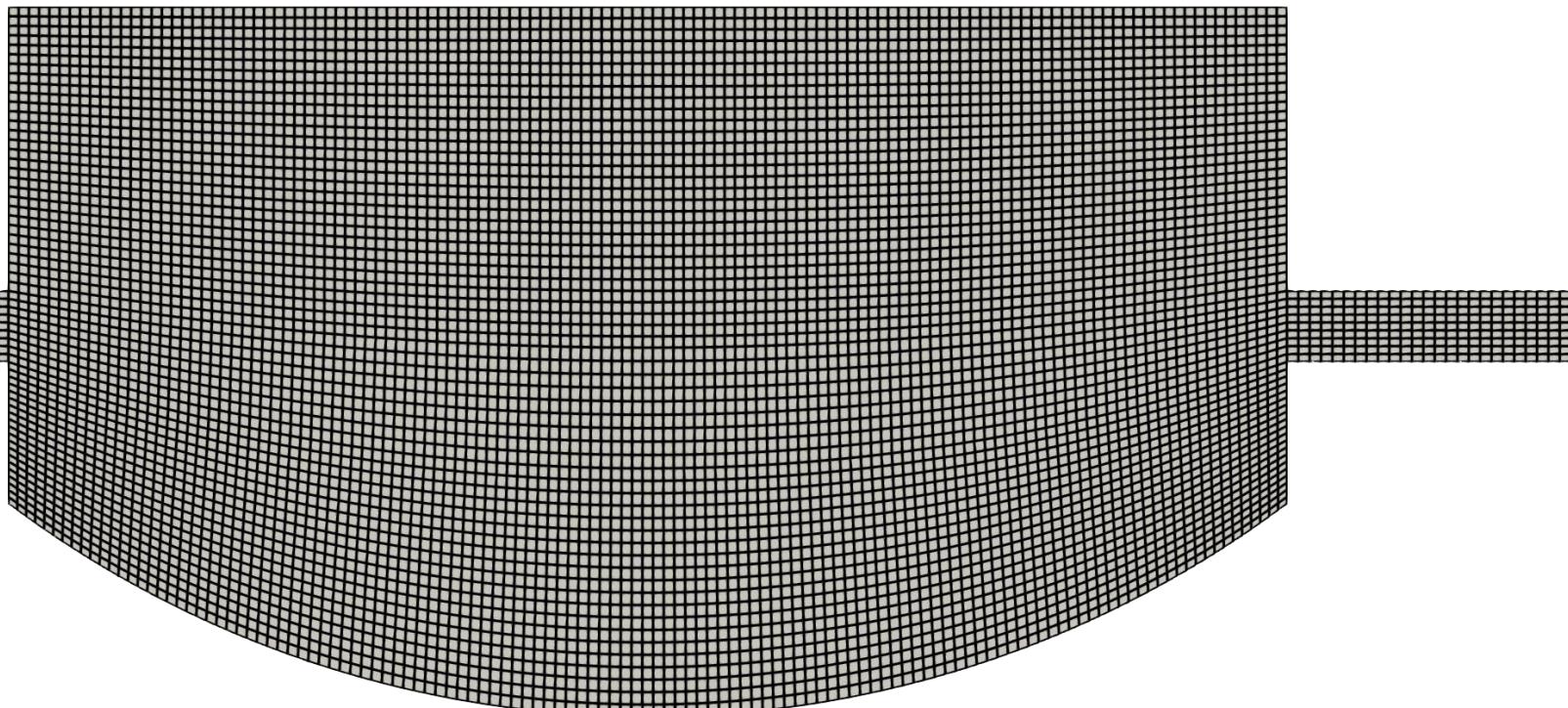
Before



CRheo@IPC



After

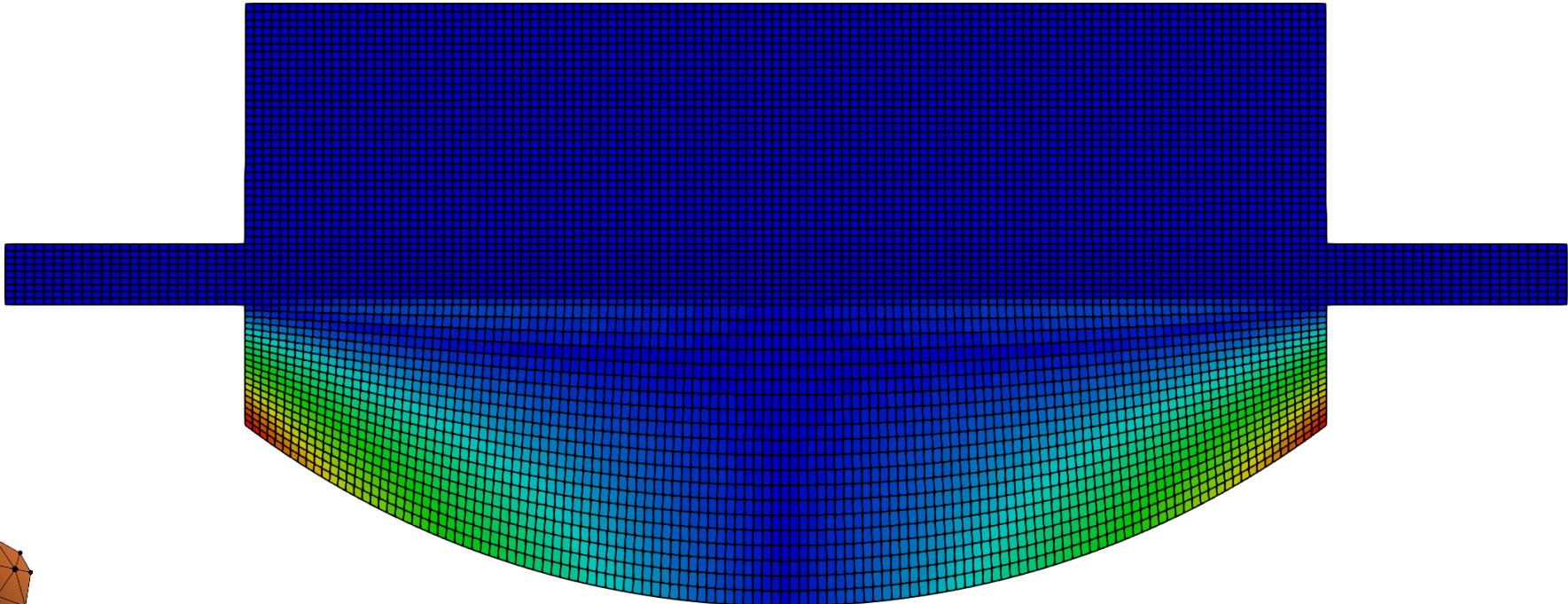
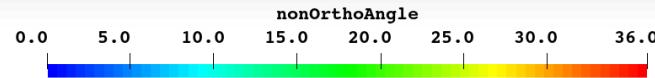


CRheo@IPC



Mesh Metrics

Quality control parameters

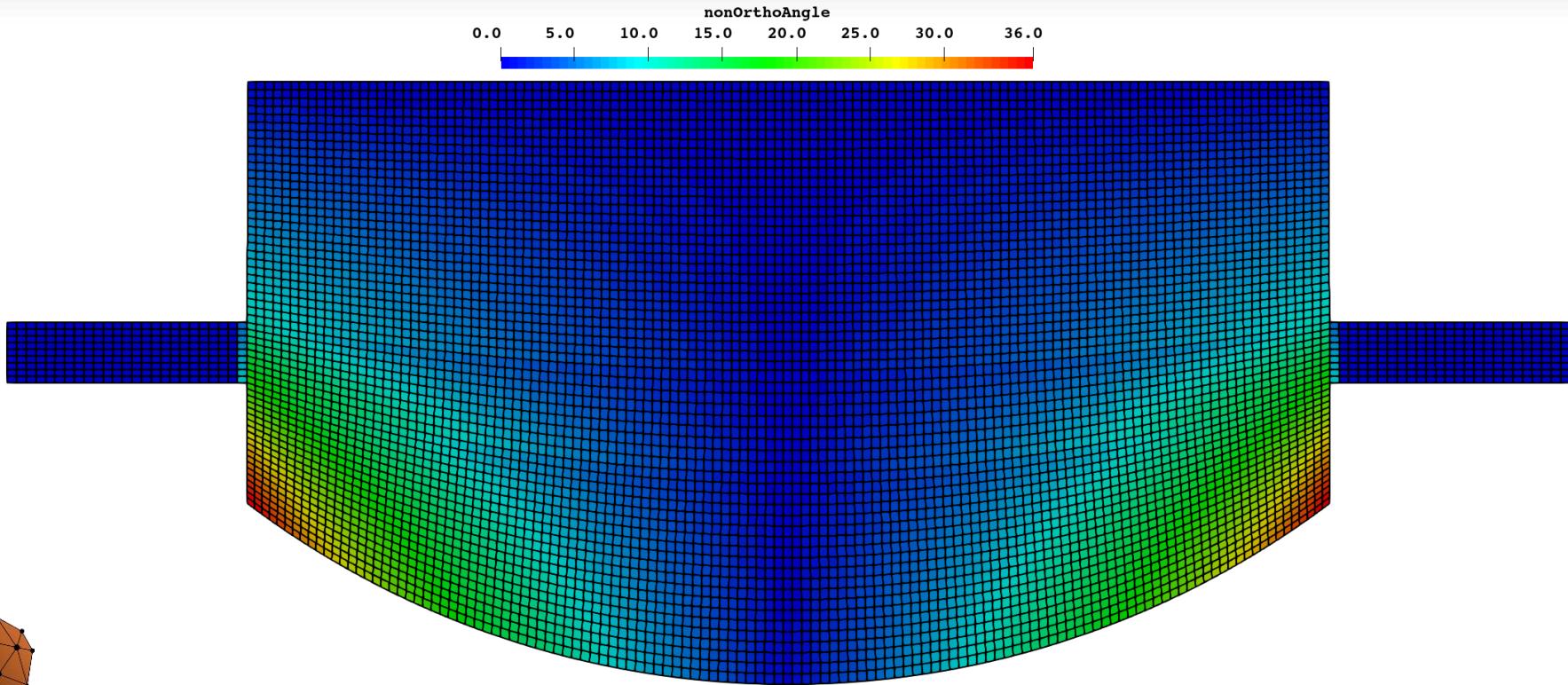


CRheo@IPC



Mesh Metrics

Quality control parameters

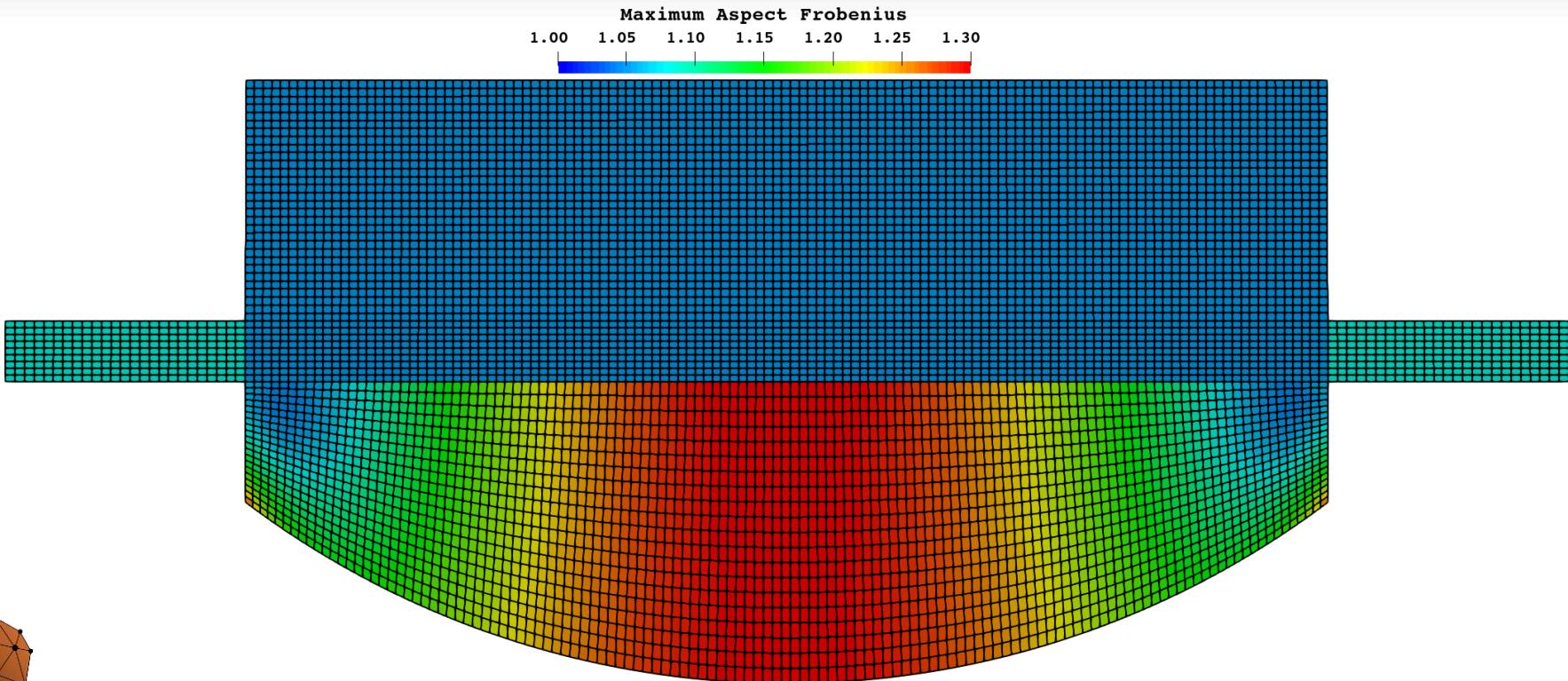


CRheo@IPC



Case Study Mesh

Quality control parameters

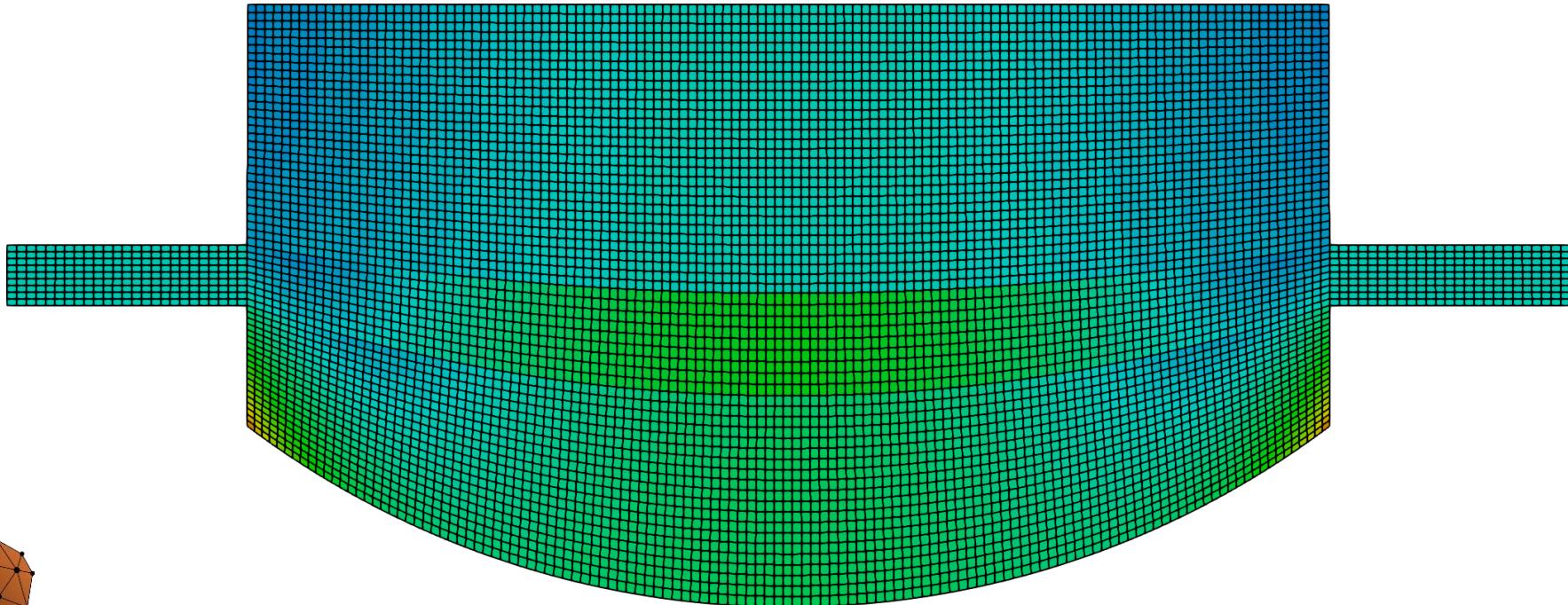
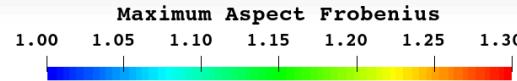


CRheo@IPC



Mesh Metrics

Quality control parameters



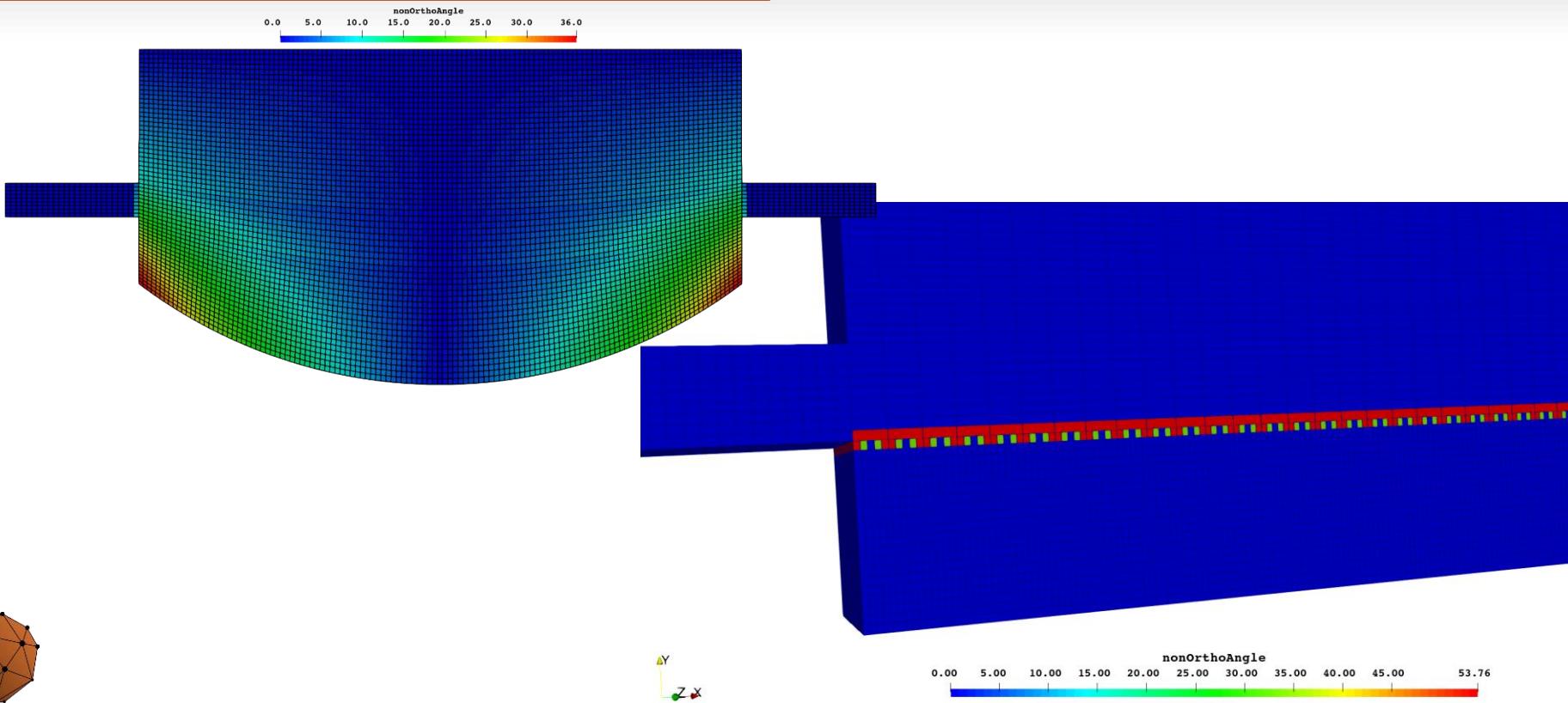
CRheo@IPC



iPC

Mesh Metrics

Quality control parameters



CRheo@IPC



GPT

In meshes for CFD and its metrics, what are relevant parameters to be considered? Non-orthogonality etc

When working with meshes for Computational Fluid Dynamics (CFD) simulations, there are several relevant parameters and metrics to consider to ensure the accuracy and efficiency of your simulations. These parameters and metrics are crucial for assessing the quality of the mesh and its impact on the simulation results. Some of the key considerations include:

- Grid Resolution:** Grid resolution or mesh density is the spacing between grid points or elements. Finer grids capture more detail in the flow field but require more computational resources.
- Mesh Size:** The overall size of the mesh domain should be appropriate to capture the entire flow field and any boundary layers or regions of interest.
- Orthogonality:** Grid orthogonality is a measure of the angle between grid lines or faces. A more orthogonal grid is preferable because it reduces numerical errors and simplifies the solution process. High non-orthogonality can lead to interpolation errors.
- Skewness:** Skewness measures the distortion of grid elements (e.g., triangles, quadrilaterals, or hexahedra). Highly skewed elements can introduce errors and may lead to convergence issues.
- Aspect Ratio:** The aspect ratio is the ratio of the length to the width of grid elements. Elements with high aspect ratios can introduce numerical errors, especially in the presence of strong gradients.
- Quality Metrics:** Various mesh quality metrics, such as the aspect ratio, skewness, and non-orthogonality, can be quantified using specific mathematical criteria. Tools like the Jacobian determinant can help assess mesh quality.
- Y+ Value:** The $y+$ value is used to assess the adequacy of the mesh near solid surfaces in capturing the boundary layer. It ensures that the mesh resolves the viscous sublayer and maintains accuracy in modelling wall turbulence.

- Boundary Layer Resolution:** Properly resolving the boundary layer is crucial, especially near solid walls. This includes ensuring that the first few grid points are inside the viscous sublayer and the turbulent boundary layer is adequately resolved.
- Grid Transition Zones:** Smooth transitions between different grid densities (e.g., near-wall refinements or far-field) are important to minimize numerical errors.
- Grid Independence Study:** Conducting a grid independence study involves running the simulation with different mesh resolutions to determine if the results converge to a consistent solution. This helps identify the optimal grid resolution.
- Conformity:** Mesh conformity ensures that neighbouring grid elements fit together without gaps or overlaps, promoting smooth data transfer and reducing interpolation errors.
- Mesh Element Types:** The choice of mesh element types (e.g., triangles, quadrilaterals, tetrahedra, hexahedra) affects accuracy and simulation efficiency. Different types may be more suitable for specific geometries or flow conditions.
- Mesh Generation Method:** The method used to generate the mesh, such as structured (block-structured or multi-block) or unstructured, can impact the ease of grid generation and quality.
- Grid Metrics Tools:** Software tools like Pointwise, ICEM CFD, or OpenFOAM's SnappyHexMesh provide metrics and visualization to evaluate mesh quality.
- Adaptive Mesh Refinement (AMR):** For complex problems, adaptive mesh refinement can be used to dynamically refine the mesh in regions of interest, improving solution accuracy and reducing computational cost.

Balancing these parameters and metrics is essential for achieving accurate and efficient CFD simulations. The specific requirements will depend on the problem, geometry, and flow conditions, so careful consideration and validation are necessary in each case.



Thank you!

This offering is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trademarks.