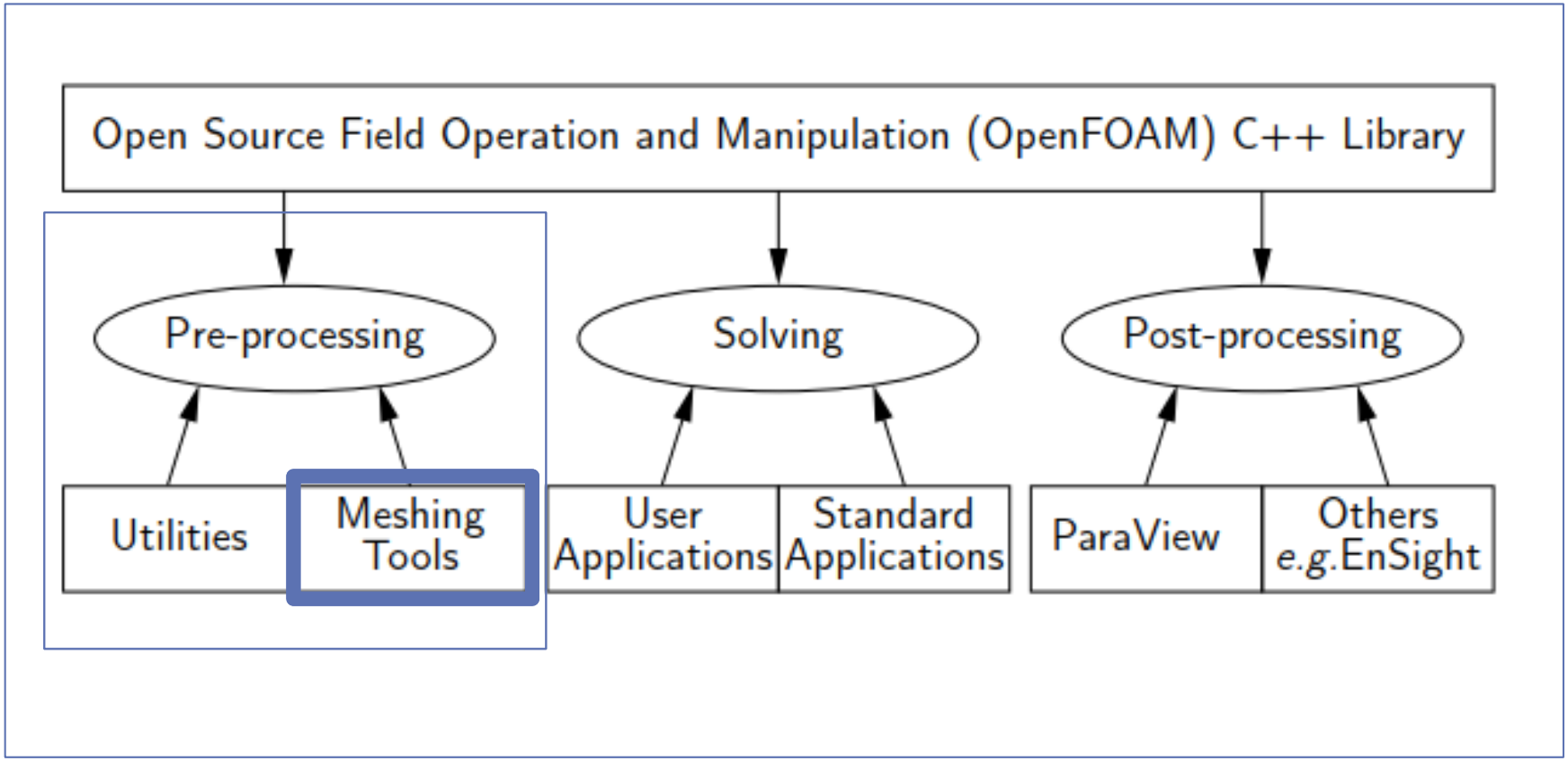
Basic Courses (BI)

Basic Meshing

Wagner de Campos Galuppo

Basic meshing with blockMeshDict



Summary

- Basic meshing: icoFoam tutorial review
- Controlling Meshing refinement: Mesh grading
- Multiblock Mesh: Connecting 3 Blocks
- Curved surfaces: Creating a circular surface
- 3D Geometry: Curved tube mesh

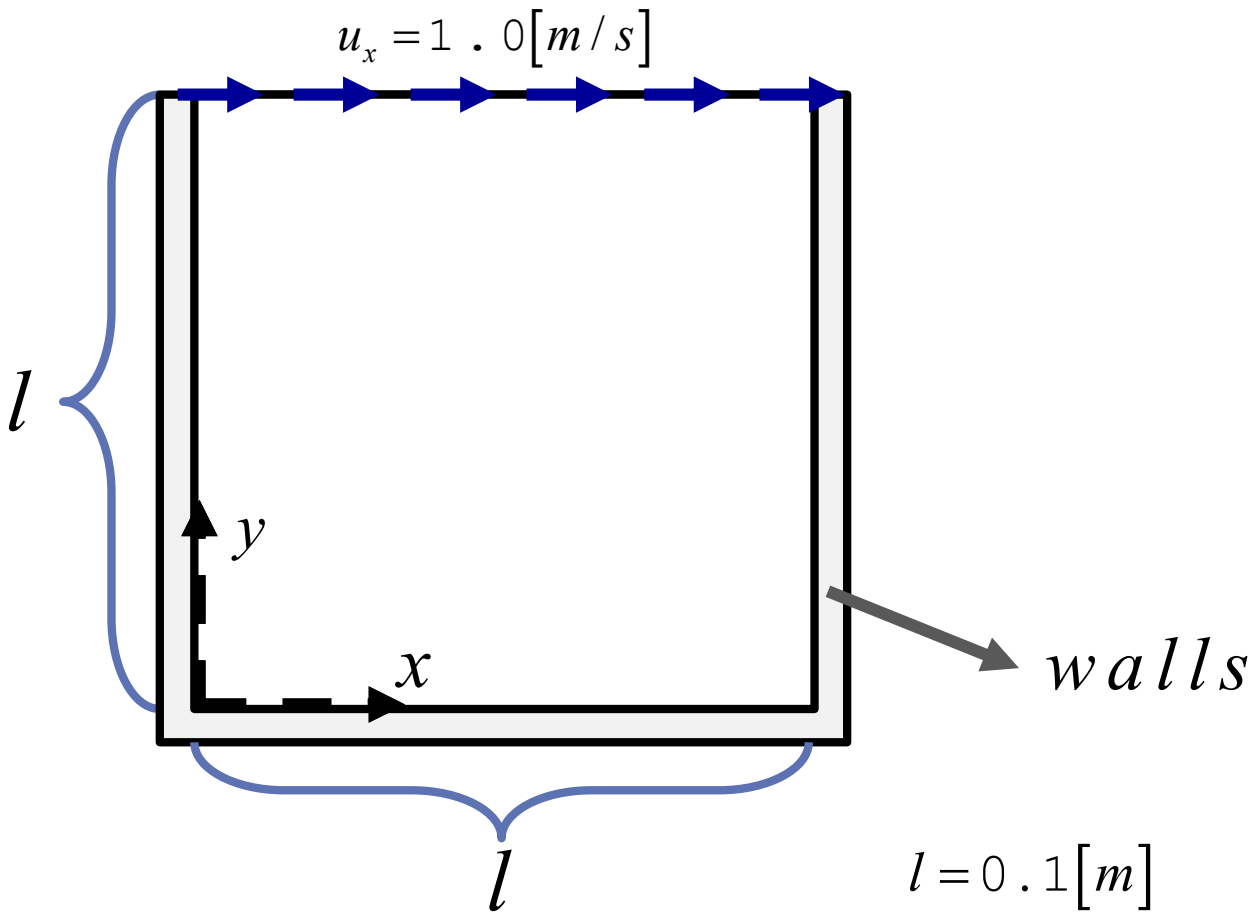
Basic Meshing: case01

- $ mkdir -p $FOAM_RUN/basicMeshing
- $ cd $FOAM_RUN/basicMeshing
- $ cp –r $FOAM_TUTORIALS/incompressible/icoFoam/cavity .
- $ mv –r cavity case01
- $ cd case01
- $ ls

```
wgaluppo@wgaluppo:~$ OpenFOAMv5
wgaluppo@wgaluppo:~$ cd $FOAM_RUN/basicMeshing
wgaluppo@wgaluppo:~/OpenFOAM/wgaluppo-5.0/run/basicMeshing$ tree -L 3
.
└── case01
    ├── 0
    │   ├── p
    │   └── U
    ├── constant
    │   └── transportProperties
    └── system
        ├── blockMeshDict
        ├── controlDict
        ├── fvSchemes
        └── fvSolution

4 directories, 7 files
wgaluppo@wgaluppo:~/OpenFOAM/wgaluppo-5.0/run/basicMeshing$
```

Basic Meshing: case01

- Problem: (Cavity tutorial)



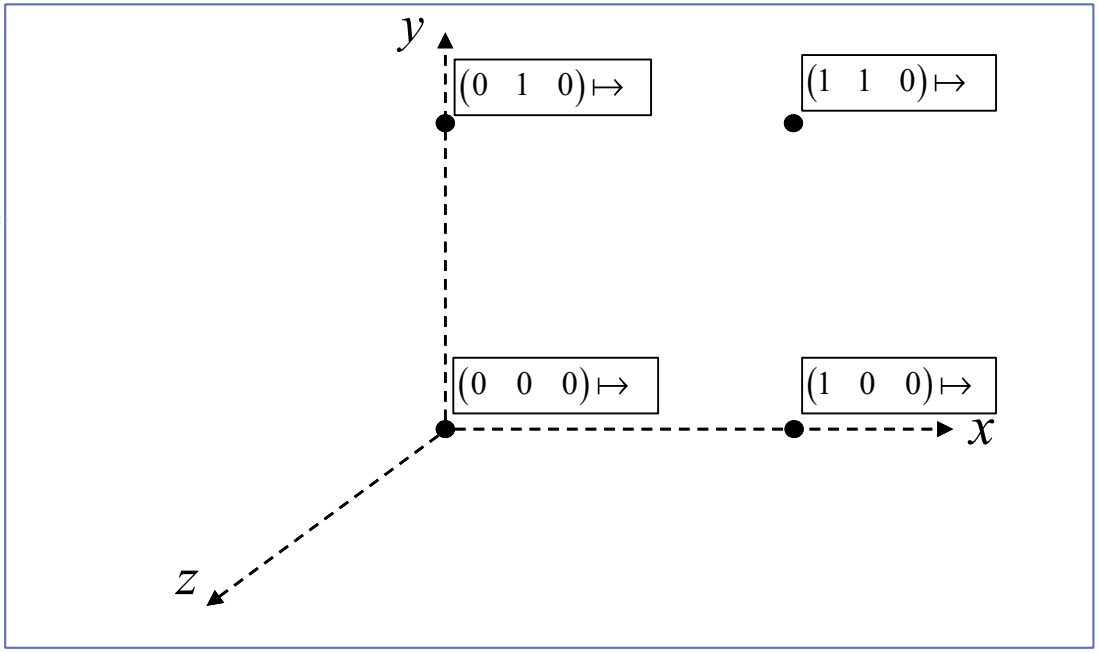
Basic Meshing: case01

- $ gedit system/blockMesh

File Edit View Search Tools Documents Help

Open

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * *

convertToMeters 0.1; //Scalling factor

vertices
(
    (0 0 0)     //0
    (1 0 0)     //1
    (1 1 0)     //2
    (0 1 0)     //3
    (0 0 0.1)   //4
    (1 0 0.1)   //5
    (1 1 0.1)   //6
    (0 1 0.1)   //7
);
```



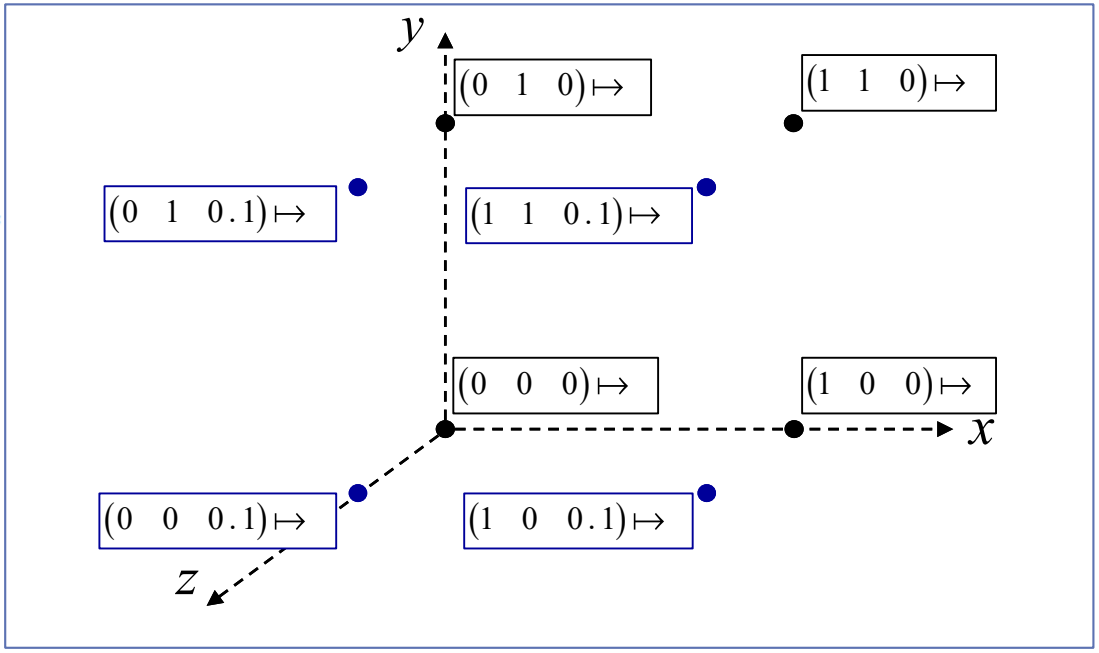
Basic Meshing: case01

- $ gedit system/blockMesh

File Edit View Search Tools Documents Help

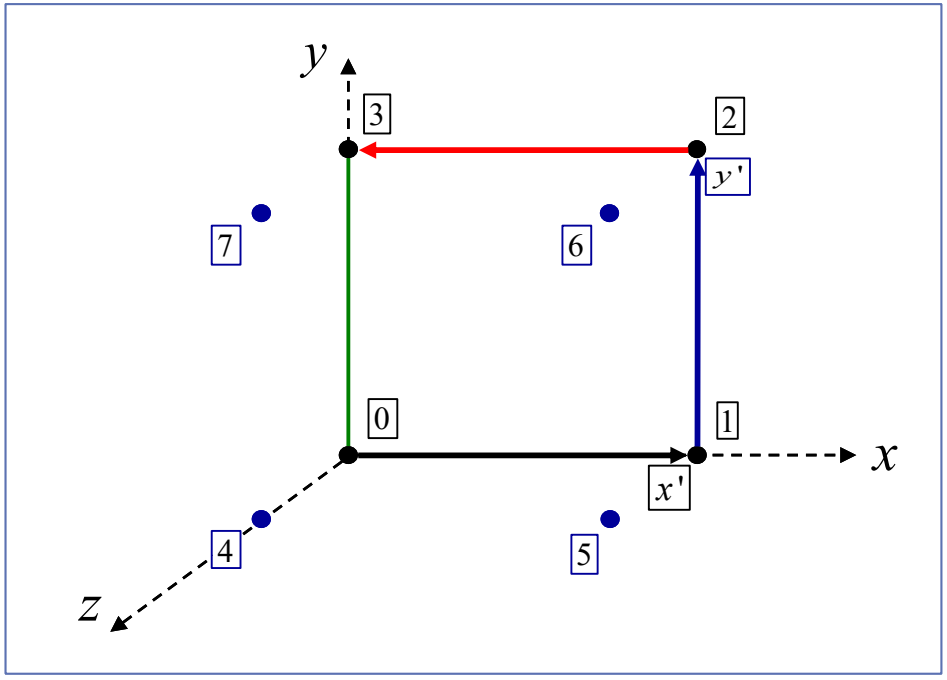
Open

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  5                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * *

convertToMeters 0.1; //Scalling factor

vertices
(
    (0 0 0)     //0
    (1 0 0)     //1
    (1 1 0)     //2
    (0 1 0)     //3
    (0 0 0.1)   //4
    (1 0 0.1)   //5
    (1 1 0.1)   //6
    (0 1 0.1)   //7
);
```



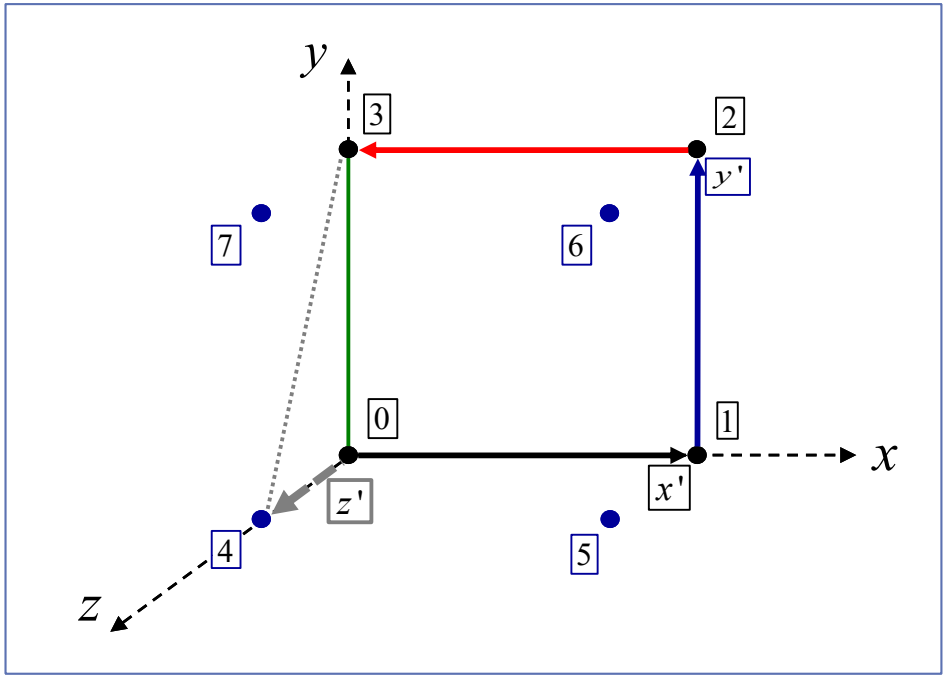
Basic Meshing: case01


```
30
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 );
39
```



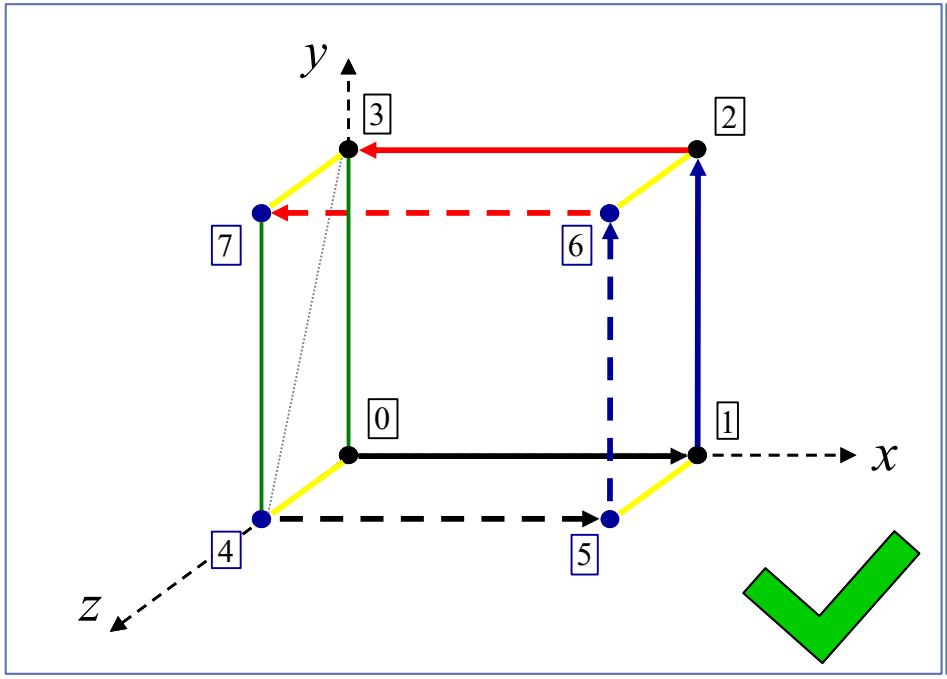
Basic Meshing: case01

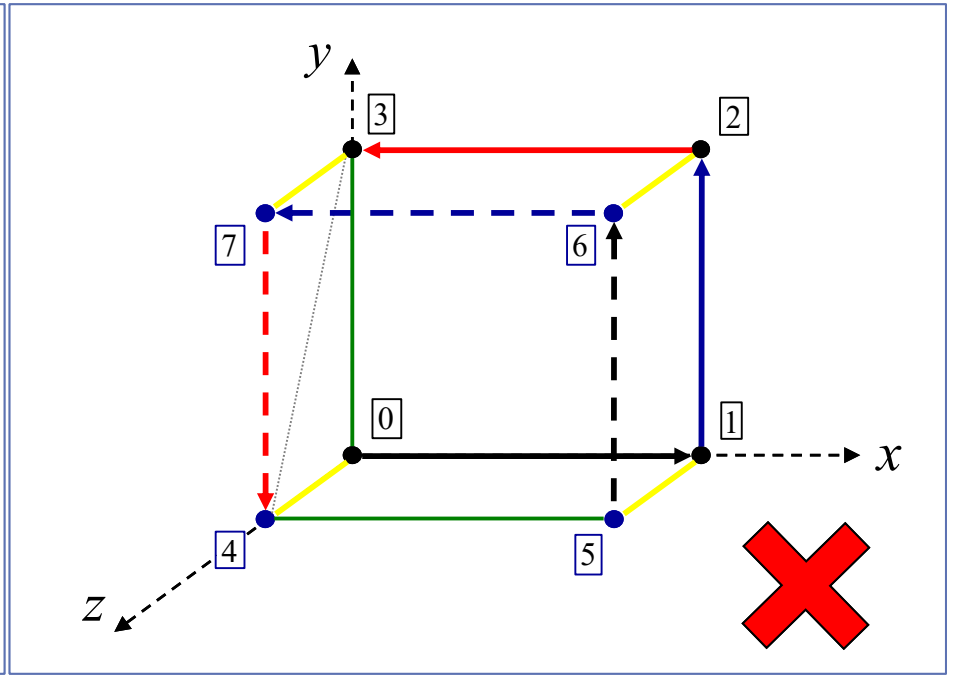

```
30
31 blocks
32 (
33     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
34 );
35
36 edges
37 (
38 );
39
```



Basic Meshing: case01

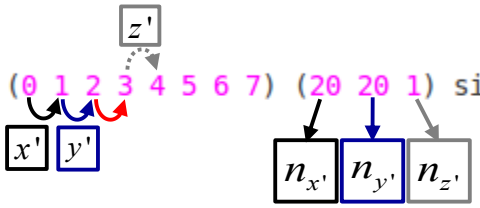
```
30
31 blocks
32 (
33     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
34 );
35
36 edges
37 (
38 );
39
```

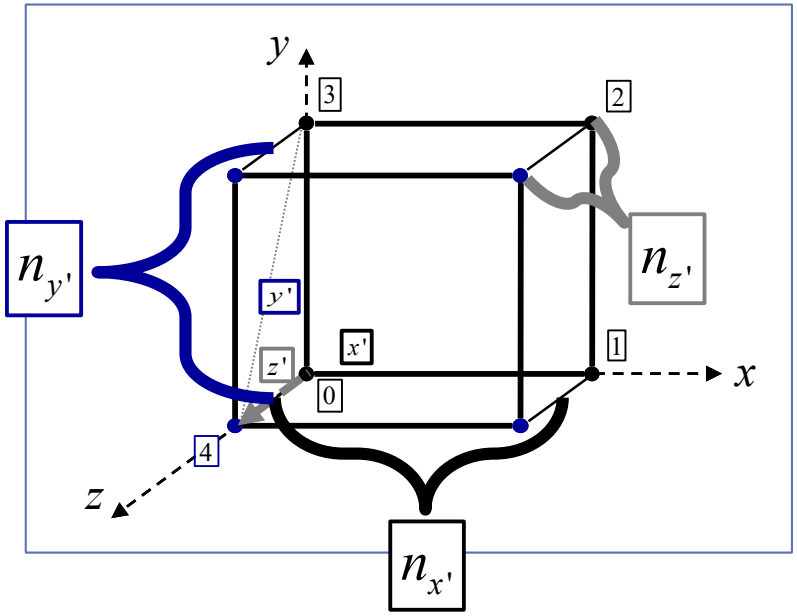


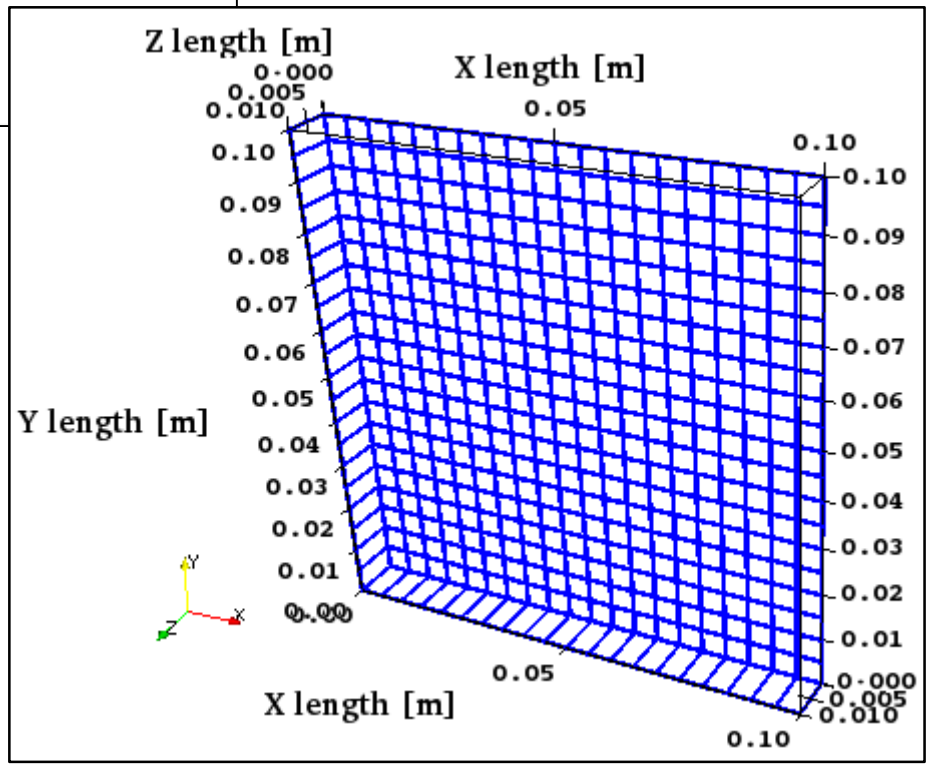


Basic Meshing: case01

```
30
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 );
39
```





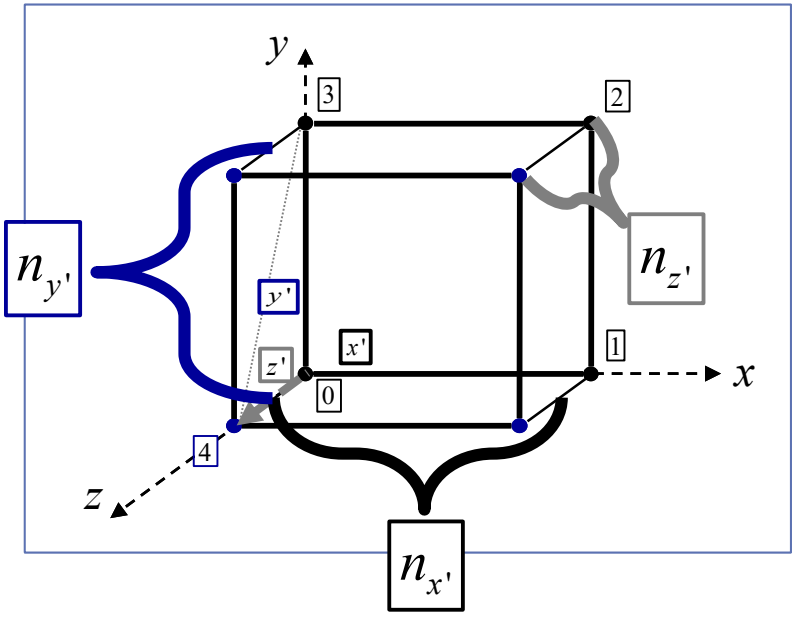


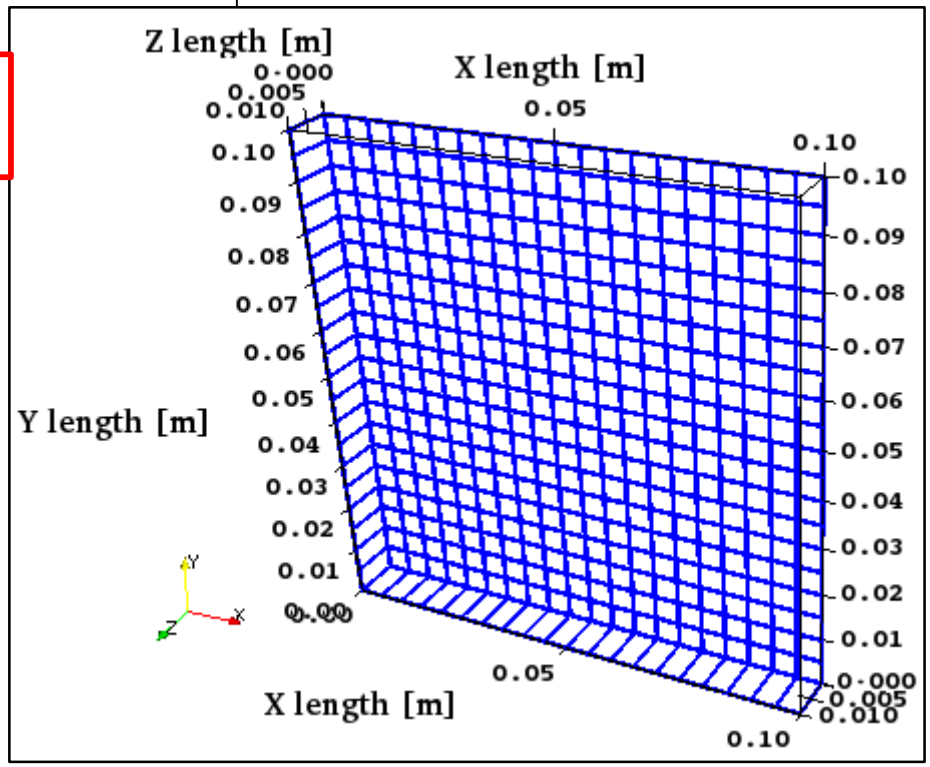
Basic Meshing: case01

```
30
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 );
39
```

Case02 – Controlling Meshing refinement: Mesh grading

Case04 – Curved surfaces: Creating a circular surface



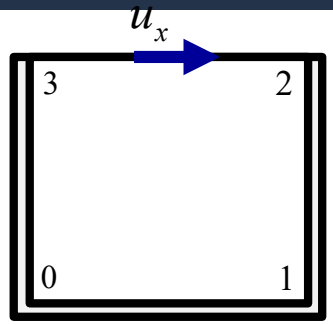


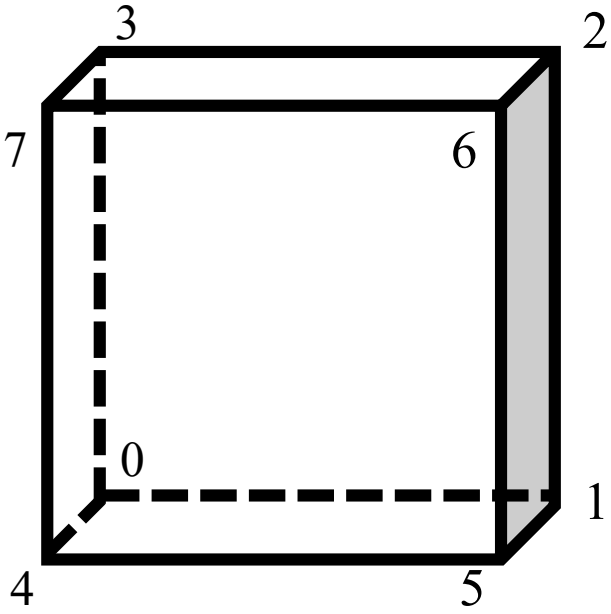
Basic Meshing: case01

```
40 boundary
41 (
42     movingWall
43     {
44         type wall;
45         faces
46         (
47             (3 7 6 2)
48         );
49     }
50     fixedWalls
51     {
52         type wall;
53         faces
54         (
55             (0 4 7 3)
56             (2 6 5 1)
57             (1 5 4 0)
58         );
59     }
60     frontAndBack
61     {
62         type empty;
63         faces
64         (
65             (0 3 2 1)
66             (4 5 6 7)
67         );
68     }
69 );
```

User input for naming the following face(s)

Base type boundary condition



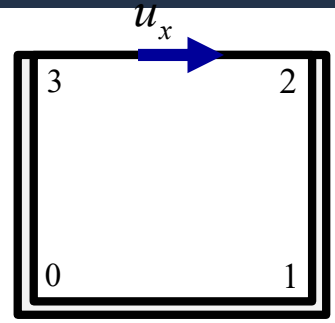


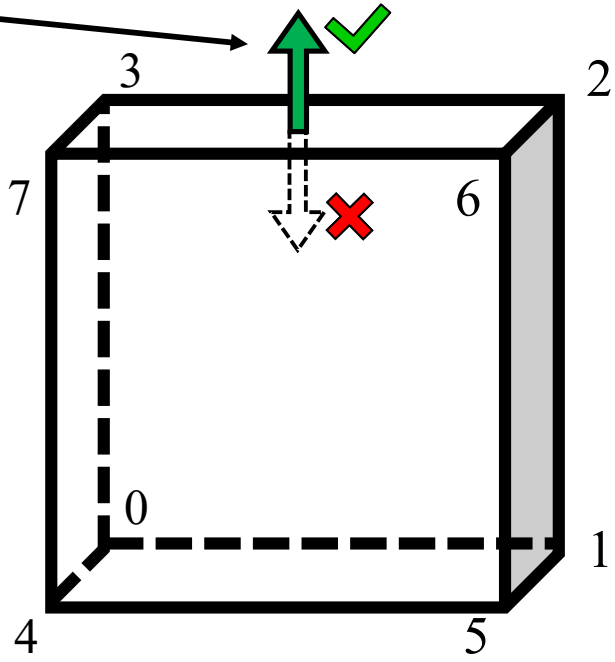
Basic Meshing: case01

```
40 boundary
41 (
42     movingWall
43     {
44         type wall;
45         faces
46         (
47             (3 7 6 2)
48         );
49     }
50     fixedWalls
51     {
52         type wall;
53         faces
54         (
55             (0 4 7 3)
56             (2 6 5 1)
57             (1 5 4 0)
58         );
59     }
60     frontAndBack
61     {
62         type empty;
63         faces
64         (
65             (0 3 2 1)
66             (4 5 6 7)
67         );
68     }
69 );
```

User input for naming the following face(s)

Base type boundary condition



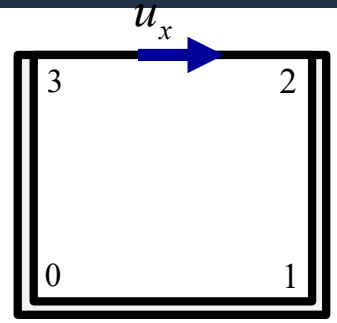


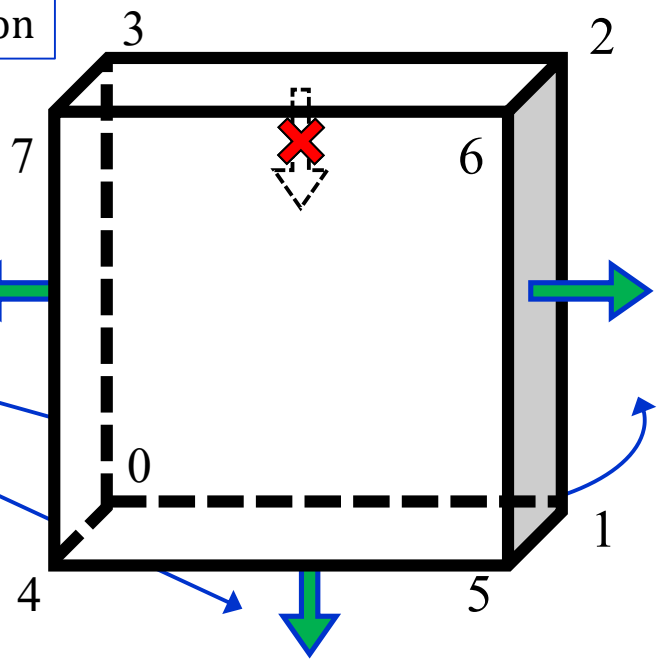
Basic Meshing: case01

```
40 boundary
41 (
42     movingWall
43     {
44         type wall;
45         faces
46         (
47             (3 7 6 2)
48         );
49     }
50     fixedWalls
51     {
52         type wall;
53         faces
54         (
55             (0 4 7 3)
56             (2 6 5 1)
57             (1 5 4 0)
58         );
59     }
60     frontAndBack
61     {
62         type empty;
63         faces
64         (
65             (0 3 2 1)
66             (4 5 6 7)
67         );
68     }
69 );
```

User input for naming the following face(s)

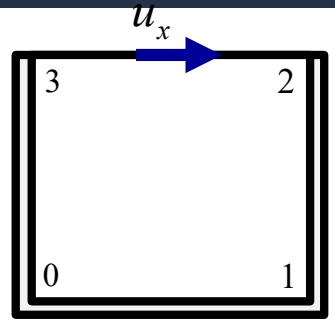
Base type boundary condition

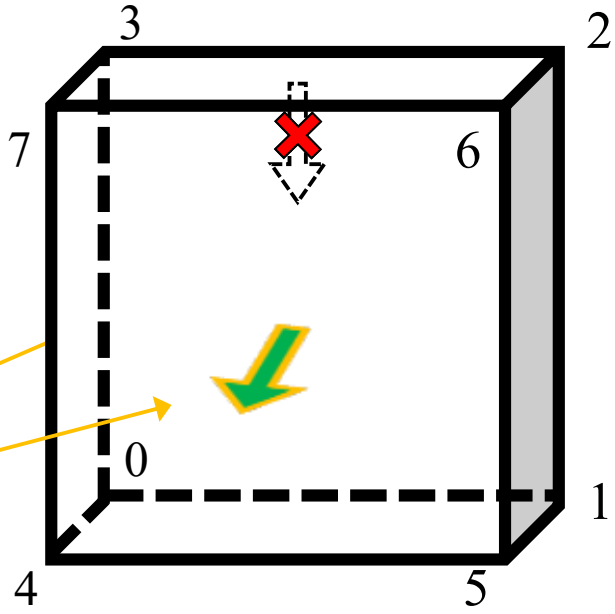




Basic Meshing: case01

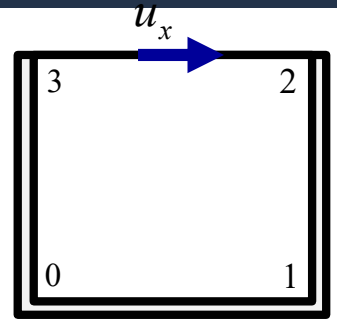
```
40 boundary
41 (
42     movingWall
43     {
44         type wall;
45         faces
46         (
47             (3 7 6 2)
48         );
49     }
50     fixedWalls
51     {
52         type wall;
53         faces
54         (
55             (0 4 7 3)
56             (2 6 5 1)
57             (1 5 4 0)
58         );
59     }
60     frontAndBack
61     {
62         type empty;
63         faces
64         (
65             (0 3 2 1)
66             (4 5 6 7)
67         );
68     }
69 );
```

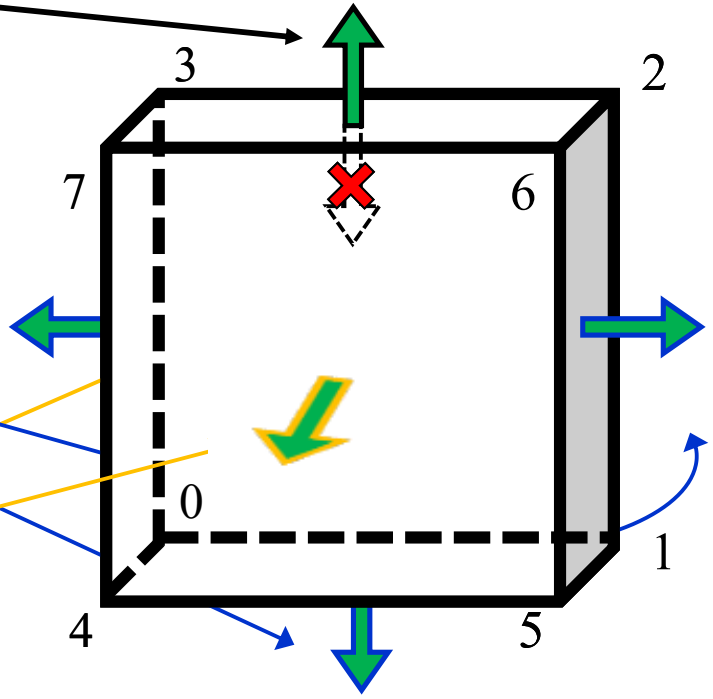




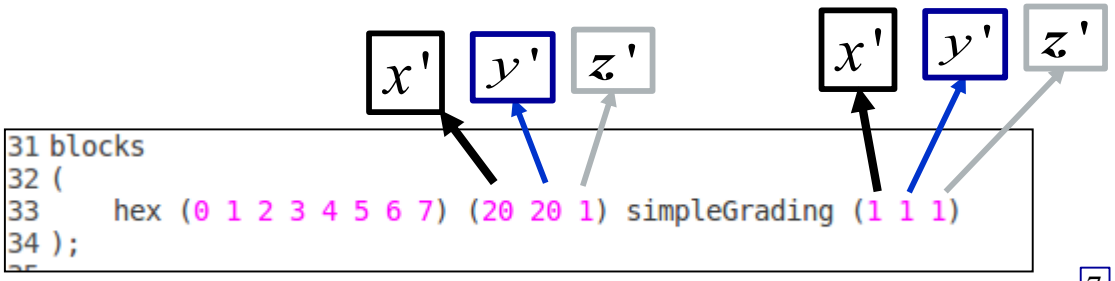
Basic Meshing: case01

```
40 boundary
41 (
42     movingWall
43     {
44         type wall;
45         faces
46         (
47             (3 7 6 2)
48         );
49     }
50     fixedWalls
51     {
52         type wall;
53         faces
54         (
55             (0 4 7 3)
56             (2 6 5 1)
57             (1 5 4 0)
58         );
59     }
60     frontAndBack
61     {
62         type empty;
63         faces
64         (
65             (0 3 2 1)
66             (4 5 6 7)
67         );
68     }
69 );
```

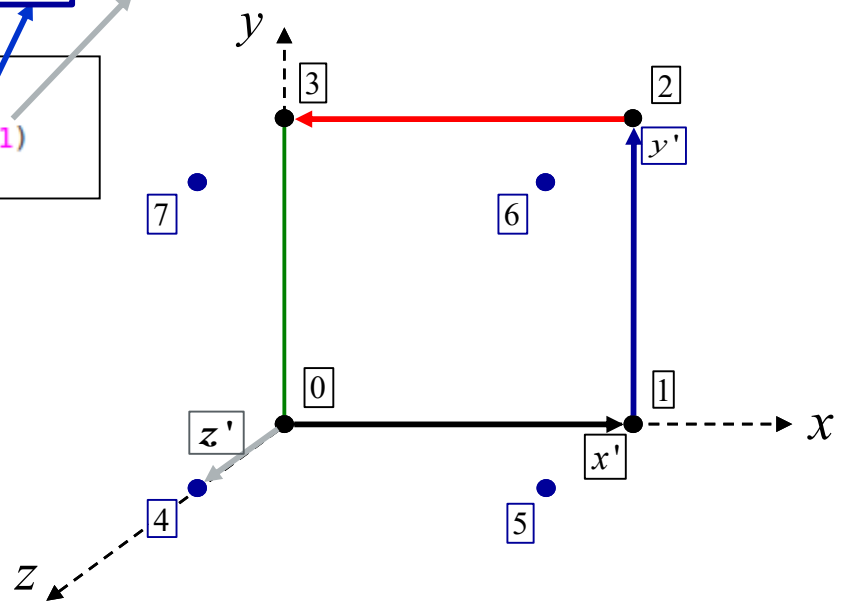




- Mesh grading for stretching the mesh towards one Plane
- icoFoam Cavity case

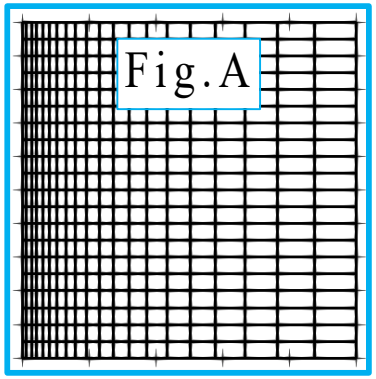


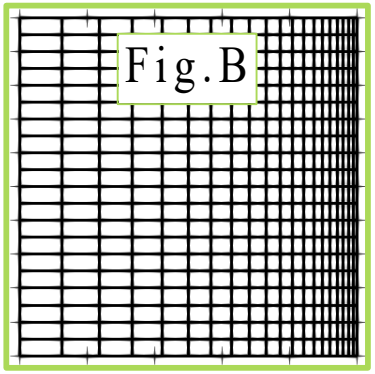
`simpleGrading` = cell expansion ratios

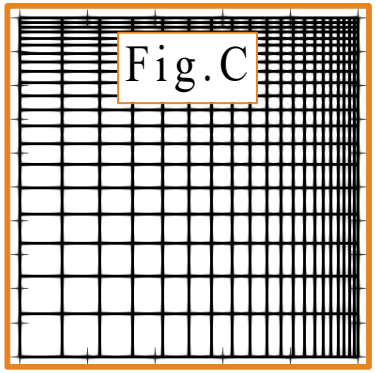


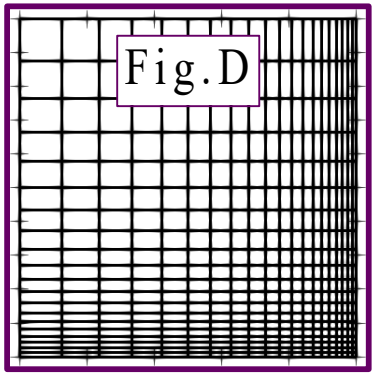
Controlling Meshing refinement: Case02

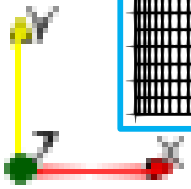
```
31 blocks
32 (
33   // In order to check out how it works just uncomment a line, run blockMesh
34   //and check the result in paraview.
35   // simpleGrading ( X1l X2l X3l ) - in the local block directions
36   // It means that the last length in the Xil direction for the corresponding
37   //block divided by the first one will be equal to the user input value in XiL
38
39     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 1.0  1.0  1.0 )// .org
40     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (10.0  1.0  1.0 )// Fig. A
41     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 0.1  1.0  1.0 )// Fig. B
42     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 0.1  0.1  1.0 )// Fig. C
43     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading ( 0.1 10.0  1.0 )// Fig. D
44
45   // Compare the setup of the following block with Fig. A
46
47     hex (1 2 3 0 5 6 7 4) (20 20 1) simpleGrading ( 1.0  10.0  1.0 )// Fig. E
```









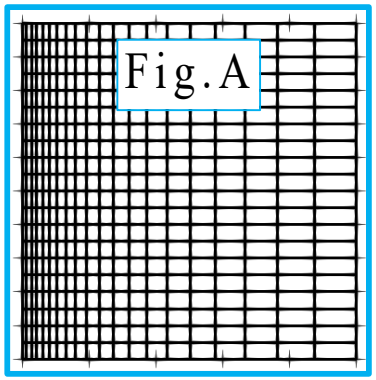


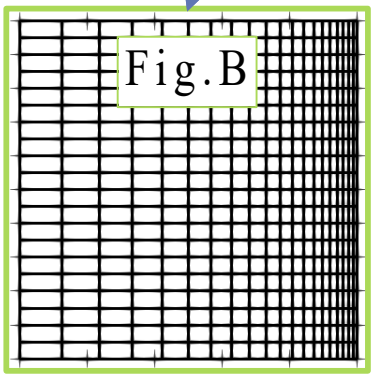
Controlling Meshing refinement: Case02

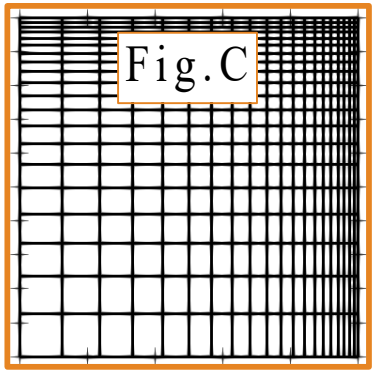
```
31 blocks
32 (
33   // In order to check o                    comment a line, run blockMesh
34   //and check the result
35   // simpleGrading ( X11                   al block directions
36   // It means that the l                   irection for the corresponding
37   //block divided by the                   l to the user input value in XiL
38
39     hex (0 1 2 3 4 5 6 7                   ng ( 1.0  1.0  1.0 )// .org
40     hex (0 1 2 3 4 5 6 7                   ng ( 10.0 1.0  1.0 )// Fig. A
41     hex (0 1 2 3 4 5 6 7                   ng ( 0.1  1.0  1.0 )// Fig. B
42     hex (0 1 2 3 4 5 6 7                   ng ( 0.1  0.1  1.0 )// Fig. C
43     hex (0 1 2 3 4 5 6 7                   ng ( 0.1 10.0  1.0 )// Fig. D
44
45   // Compare the setup of the following block with Fig. A
46
47     hex (1 2 3 0 5 6 7 4) (20 20 1) simpleGrading ( 1.0  10.0  1.0 )// Fig. E
```

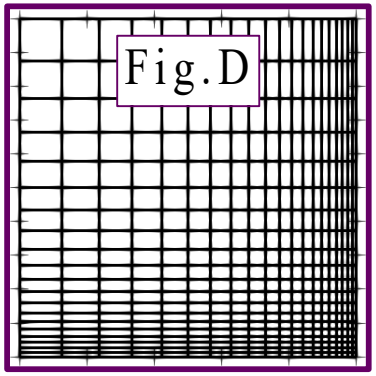
Fig.E

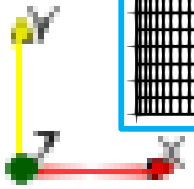
x' y' z'









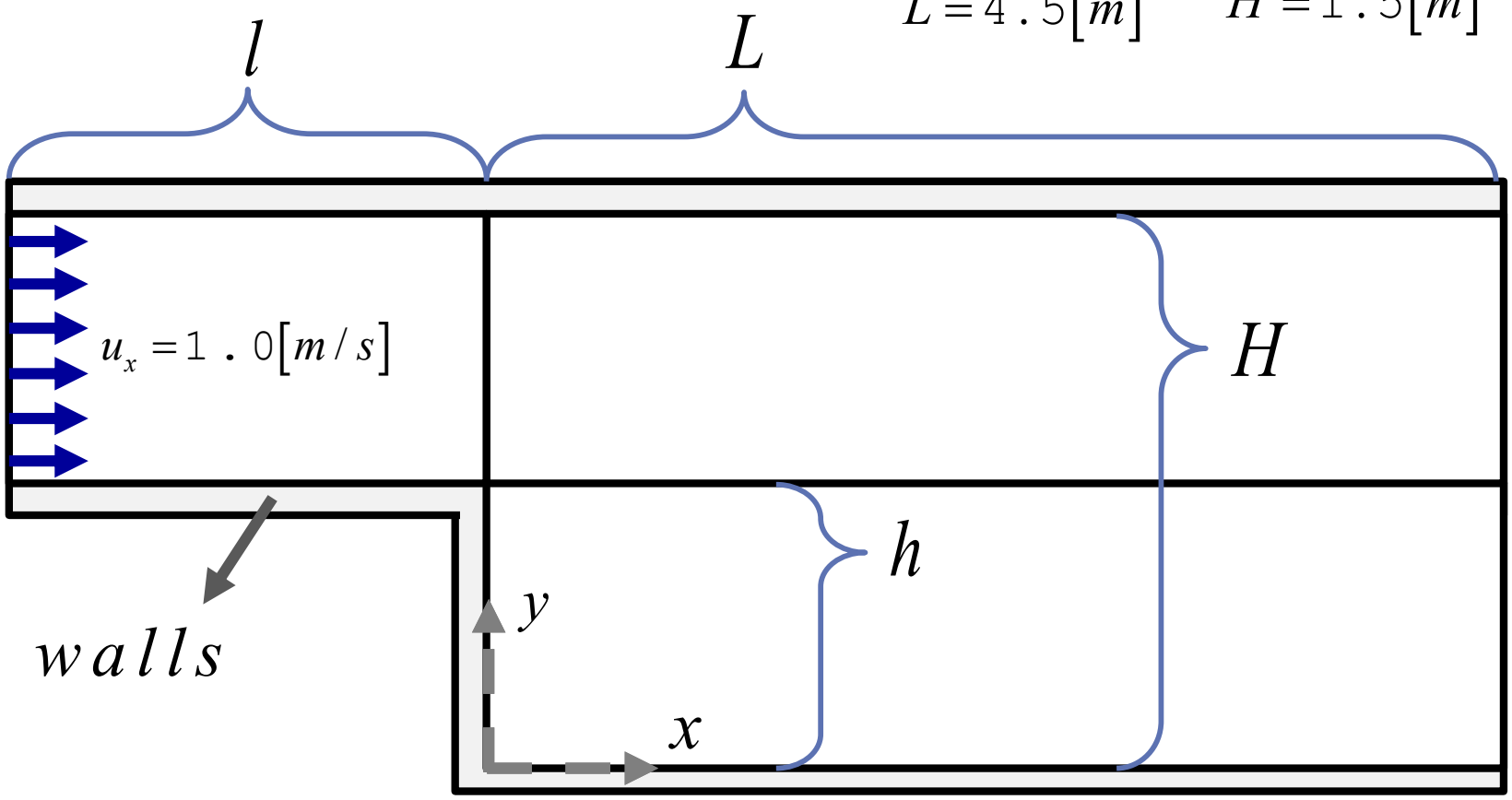


Multiblock Mesh: Case03

- Connecting 3 blocks

$l = 2.5[m]$ $h = 0.5[m]$

$L = 4.5[m]$ $H = 1.5[m]$



Multiblock Mesh: Case03

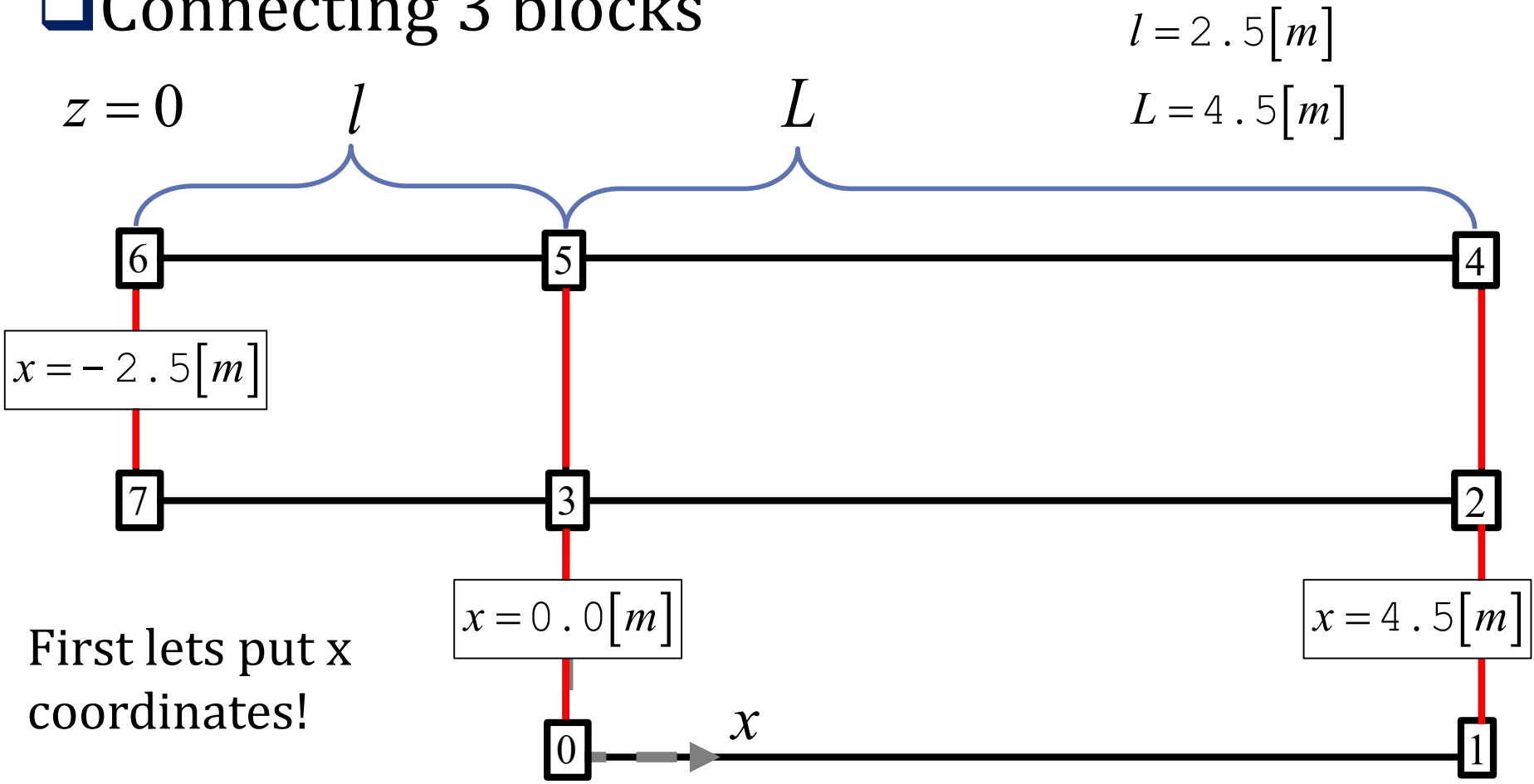
- Connecting 3 blocks
- This tutorial blockMesh file

```
17 convertToMeters 1.0;
18
19 vertices
20 (
21     //Plane z=0
22     (  x    y    0.0 )      //0
23     (  x    y    0.0 )      //1
24     (  x    y    0.0 )      //2
25     (  x    y    0.0 )      //3
26     (  x    y    0.0 )      //4
27     (  x    y    0.0 )      //5
28     (  x    y    0.0 )      //6
29     (  x    y    0.0 )      //7
30
31     //Plane z=0.5
32     (  x    y    0.5 )      //8
33     (  x    y    0.5 )      //9
34     (  x    y    0.5 )      //10
35     (  x    y    0.5 )      //11
36     (  x    y    0.5 )      //12
37     (  x    y    0.5 )      //13
38     (  x    y    0.5 )      //14
39     (  x    y    0.5 )      //15
```

We are going to replace x and y to its corresponding coodinates!

Multiblock Mesh: Case03

- Connecting 3 blocks



First lets put x coordinates!

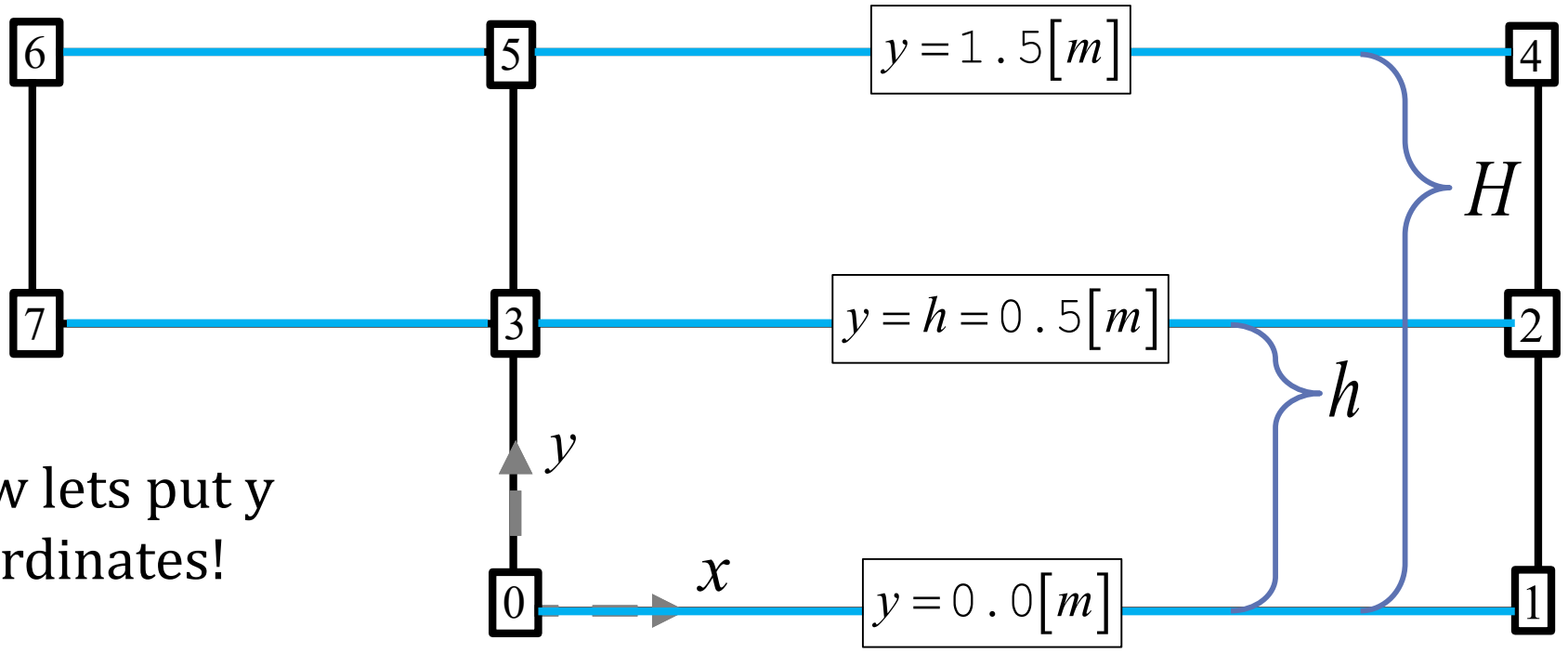
Multiblock Mesh: Case03

- Connecting 3 blocks

$h = 0.5[m]$

$H = 1.5[m]$

$z = 0$



Now lets put y coordinates!

Multiblock Mesh: Case03

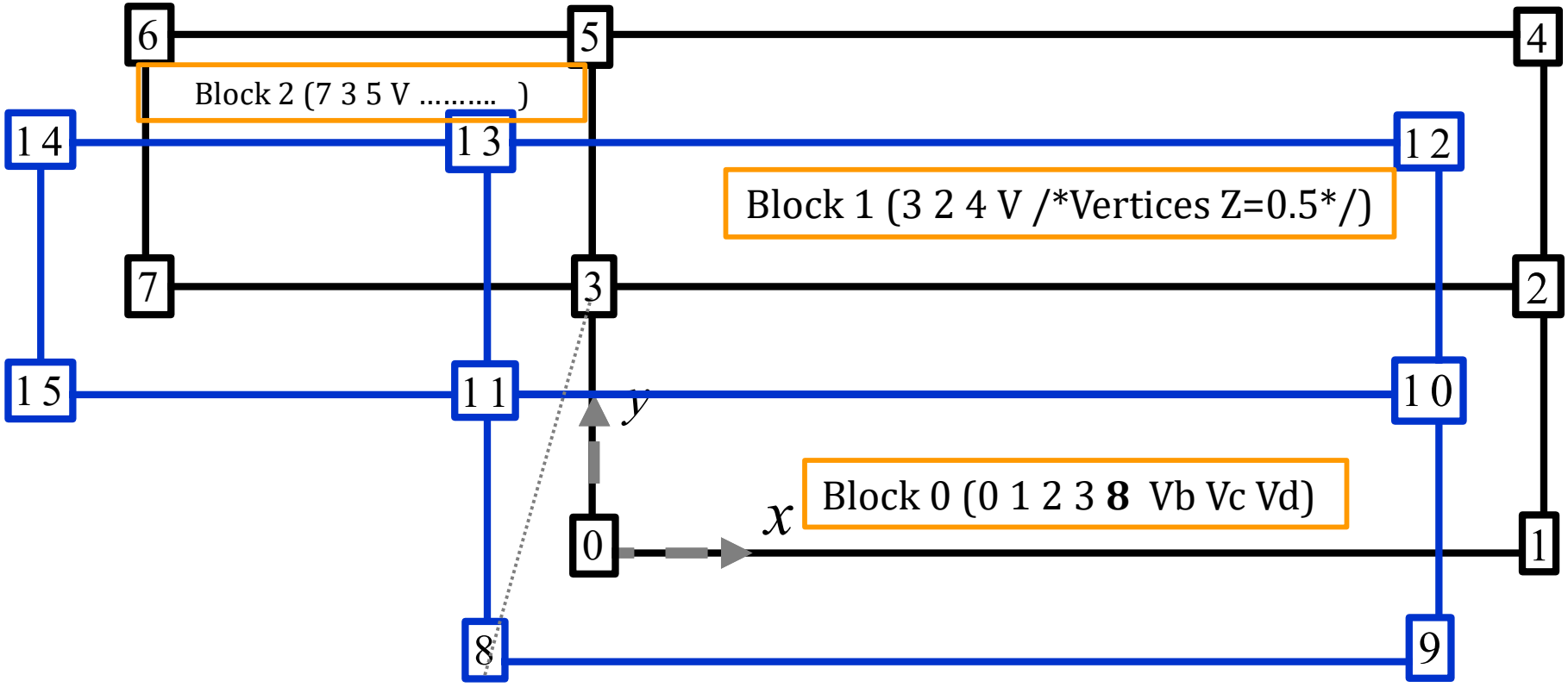
- Now lets connect vertices to create blocks.

```
41
42 blocks
43 (
44     hex (0 1 2 3 /*Vertices Z=0.5*/ ) (NXD NYS 1) simpleGrading (1 1 1) //Block 0
45     hex (3 2 4 V /*Vertices Z=0.5*/ ) (NXD NYI 1) simpleGrading (1 1 1) //Block 1
46     hex (7 3 5 V /*Vertices Z=0.5*/ ) (NXU NYI 1) simpleGrading (1 1 1) //Block 2
47 );
48
49 edges
50 (
51 );
```

Multiblock Mesh: Case03

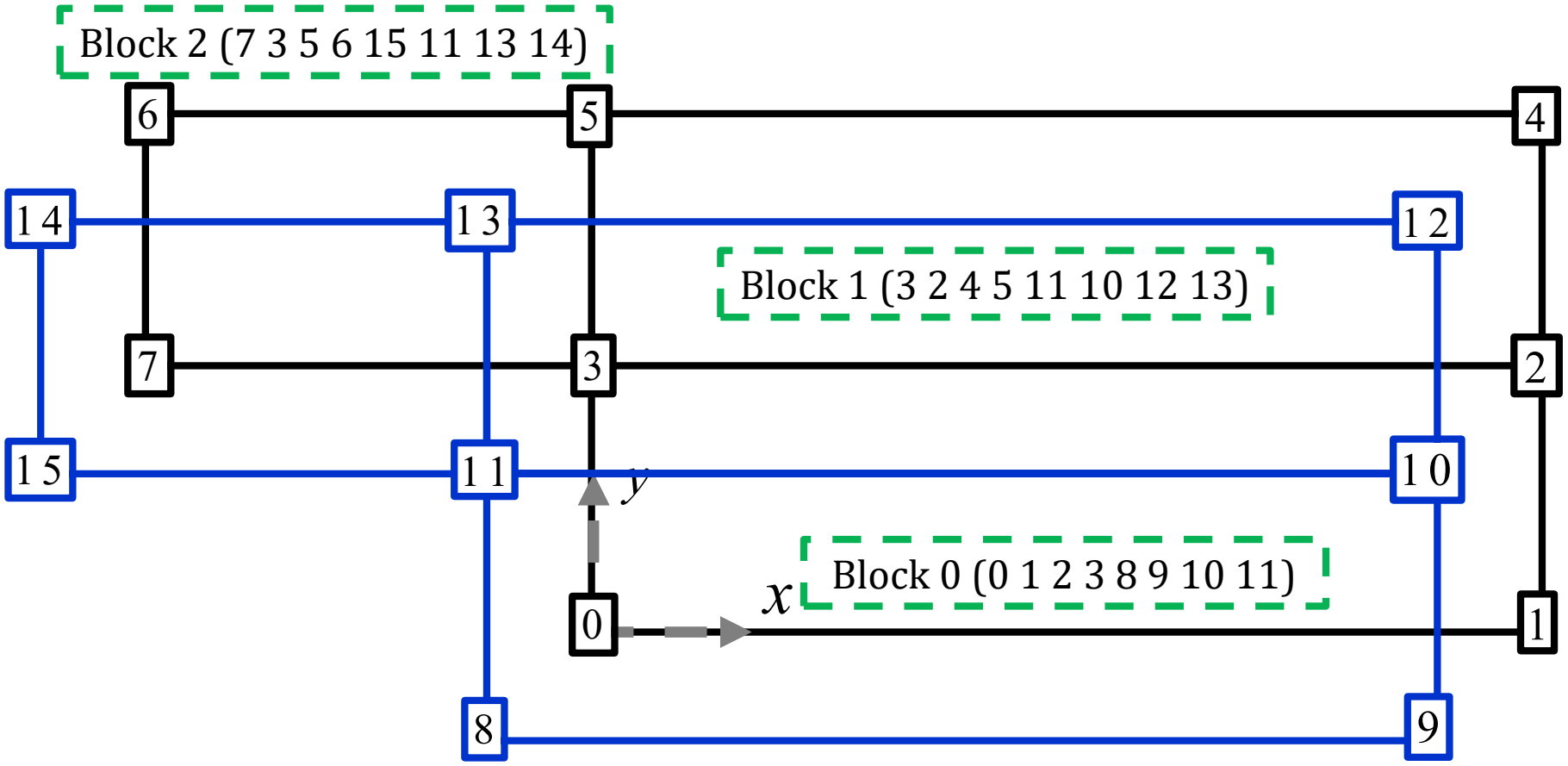
Creating 3 blocks

```
hex (0 1 2 3 /*Vertices Z=0.5*/ ) (NXD NYS 1) simpleGrading (1 1 1) //Block 0
hex (3 2 4 V /*Vertices Z=0.5*/ ) (NXD NYI 1) simpleGrading (1 1 1) //Block 1
hex (7 3 5 V /*Vertices Z=0.5*/ ) (NXU NYI 1) simpleGrading (1 1 1) //Block 2
```



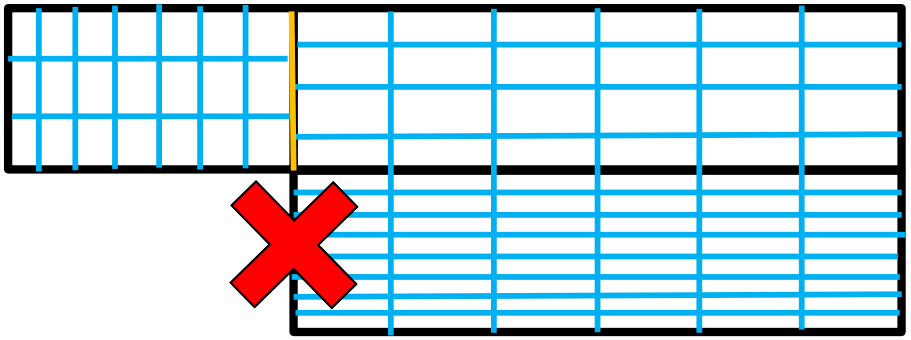
Multiblock Mesh: Case03

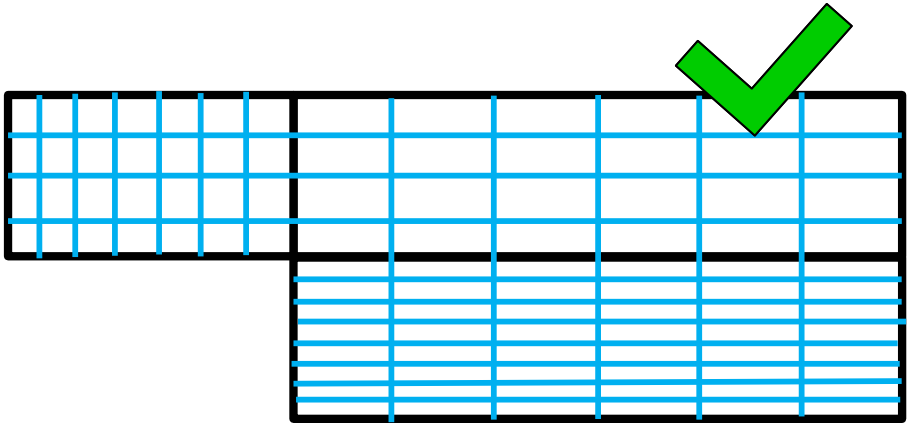
- Creating 3 blocks

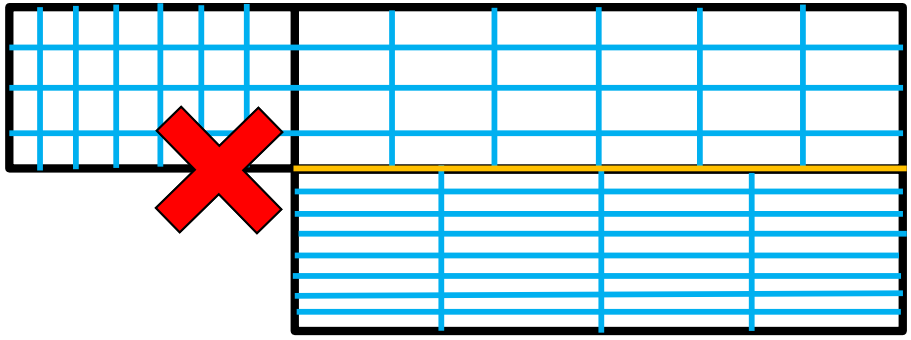


Multiblock Mesh: Case03

- Connecting 3 blocks



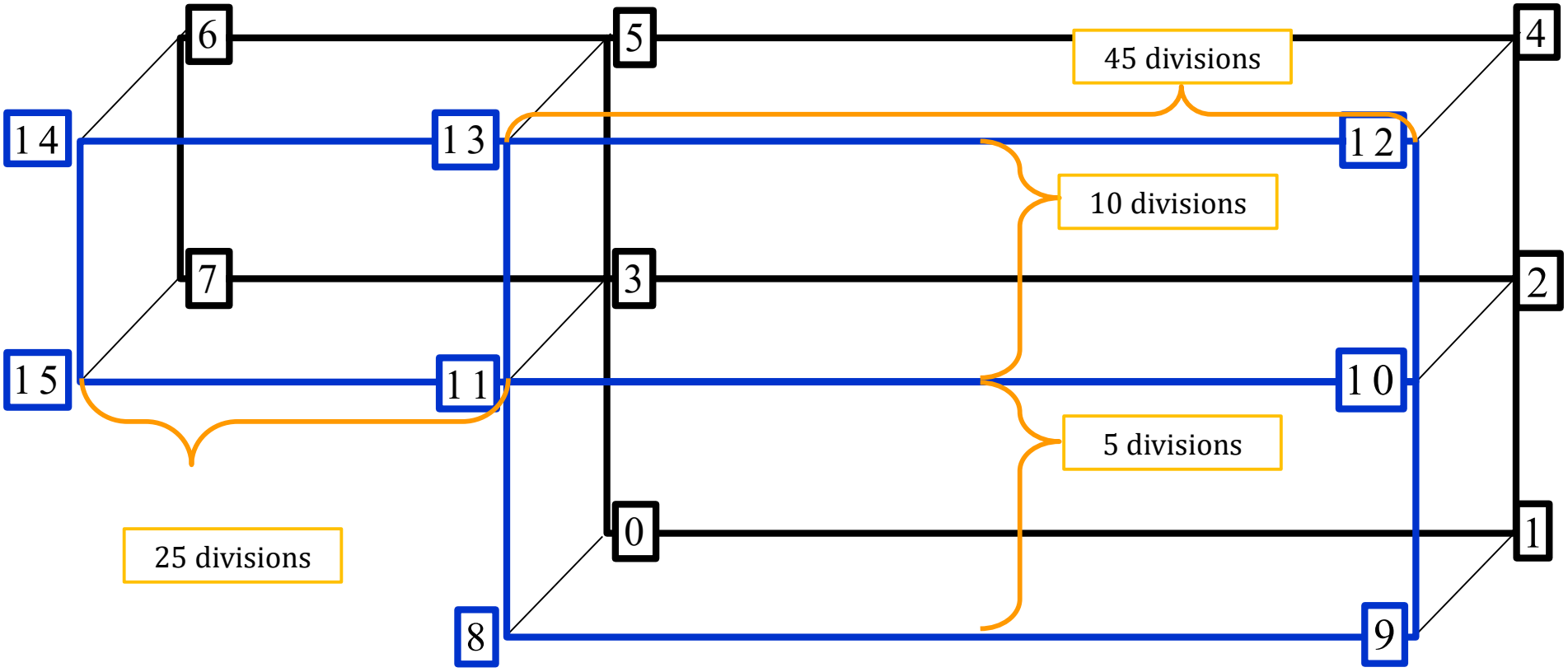




Multiblock Mesh: Case03

Connecting 3 blocks

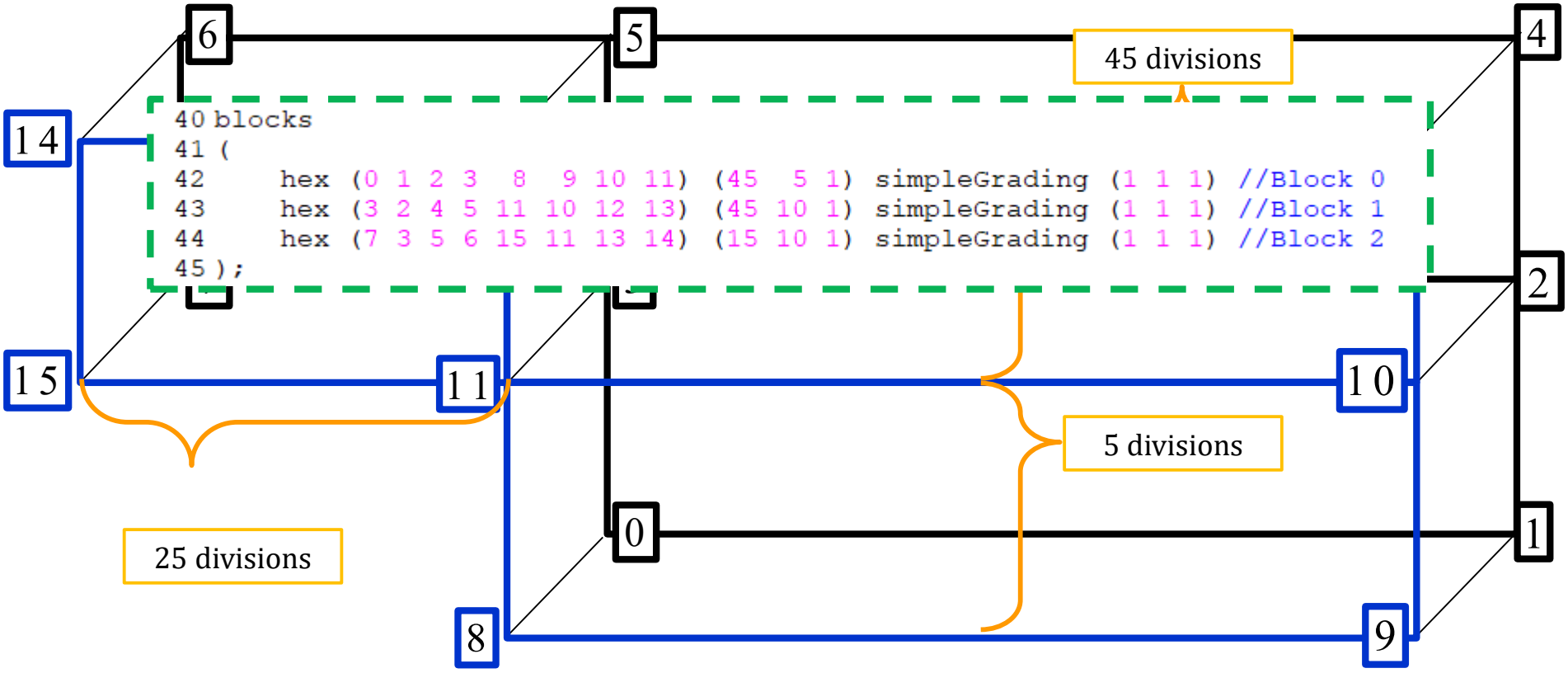
```
(NXD NYS 1) simpleGrading (1 1 1) //Block 0
(NXD NYI 1) simpleGrading (1 1 1) //Block 1
(NXU NYI 1) simpleGrading (1 1 1) //Block 2
```



Multiblock Mesh: Case03

Connecting 3 blocks

```
(NXD NYS 1) simpleGrading (1 1 1) //Block 0
(NXD NYI 1) simpleGrading (1 1 1) //Block 1
(NXU NYI 1) simpleGrading (1 1 1) //Block 2
```



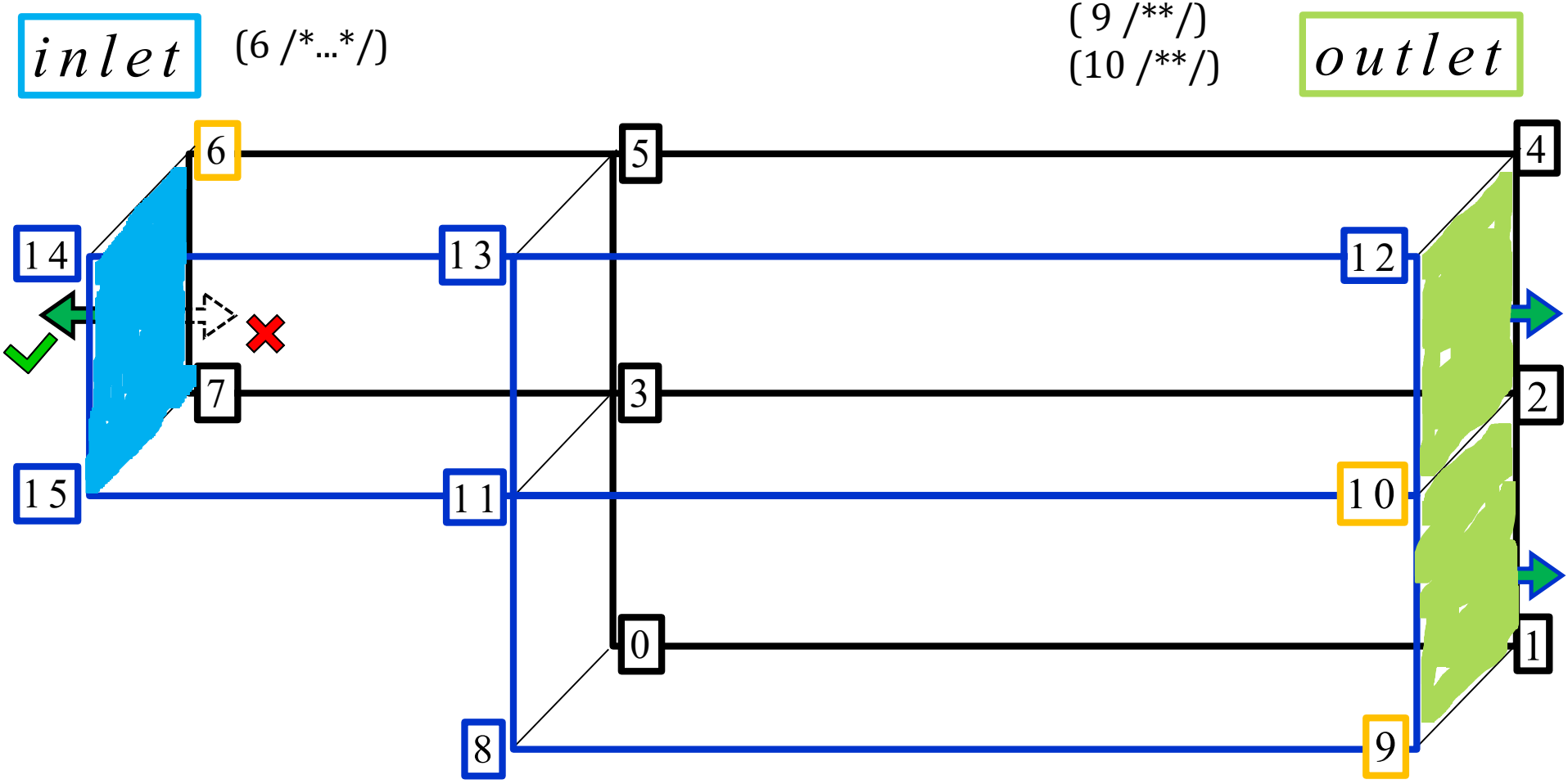
Multiblock Mesh: Case03

- Now let's give names and its base boundary conditions to these planes.

```
53 boundary
54 (
55     inlet
56     {
57         type patch;
58         faces
59         (
60             (6 /*...*/)
61         );
62     }
63     outlet
64     {
65         type patch;
66         faces
67         (
68             ( 9 /**/)
69             (10 /**/)
70         );
71     }
72     walls
73     {
74         type wall;
75         faces
76         (
77             //Bottom
78             ( 7  /**/)
79             ( 3  /**/)
80             ( 0  /**/)
81
82             //Top
83             ( 4  /**/)
84             ( 5  /**/)
85         );
86     }
87     frontAndBack
88     {
89         type empty;
90         faces
91         (
92             //front
93             ( 8  /**/)
94             (11  /**/)
95             (15  /**/)
96
97             //Back
98             (0 /**/)
99             (3 /**/)
100            (7 /**/)
101        );
102    }
103 );
104
```

Multiblock Mesh: Case03

- Connecting 3 blocks



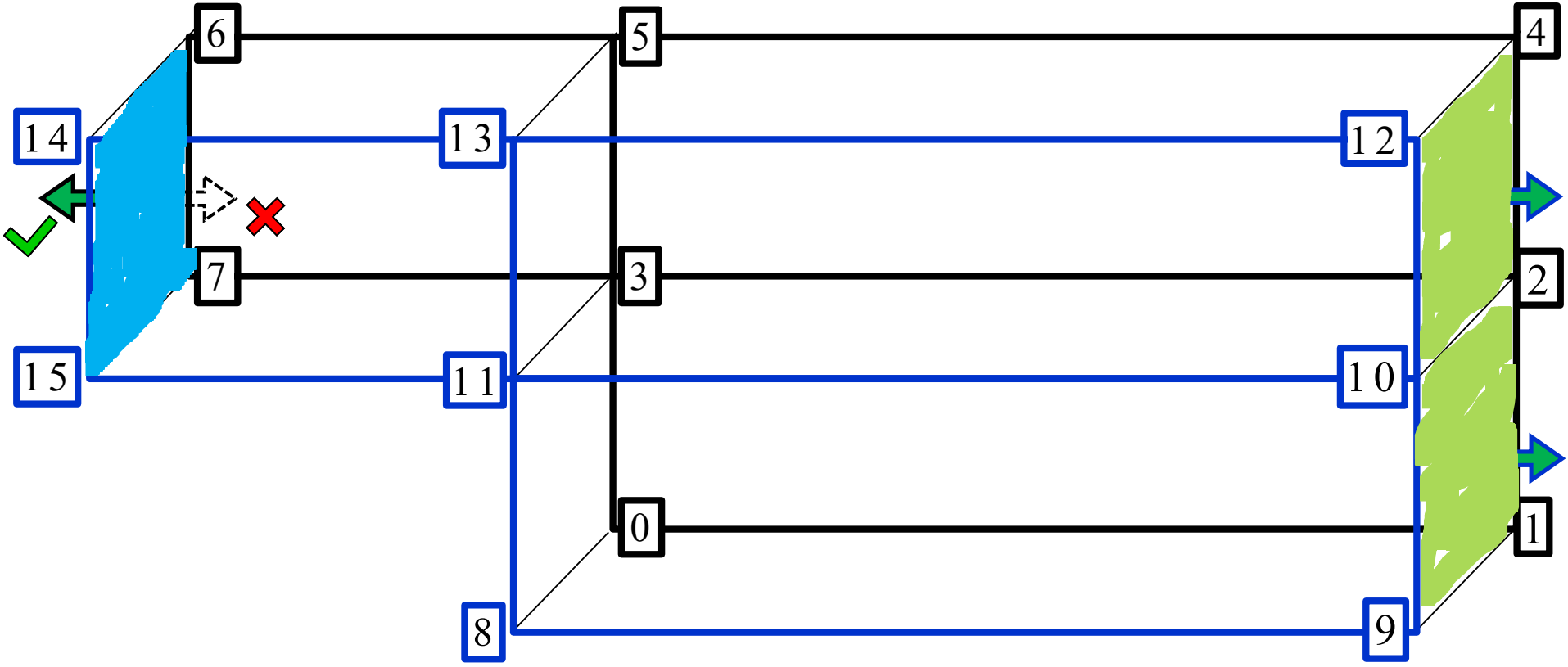
Multiblock Mesh: Case03

- Connecting 3 blocks

(9 1 2 10)
(10 2 4 12)

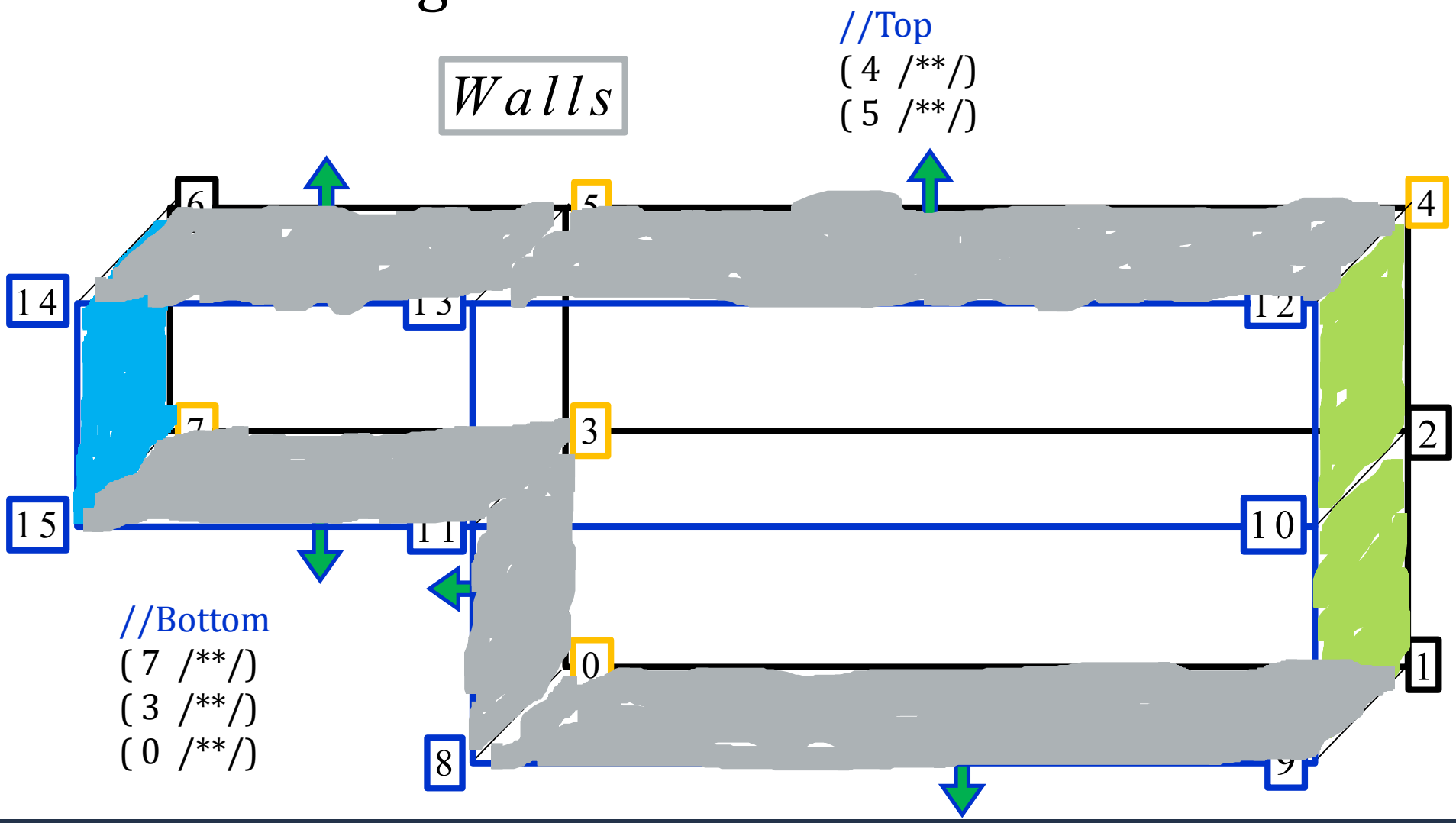
inlet (6 7 15 14)

outlet



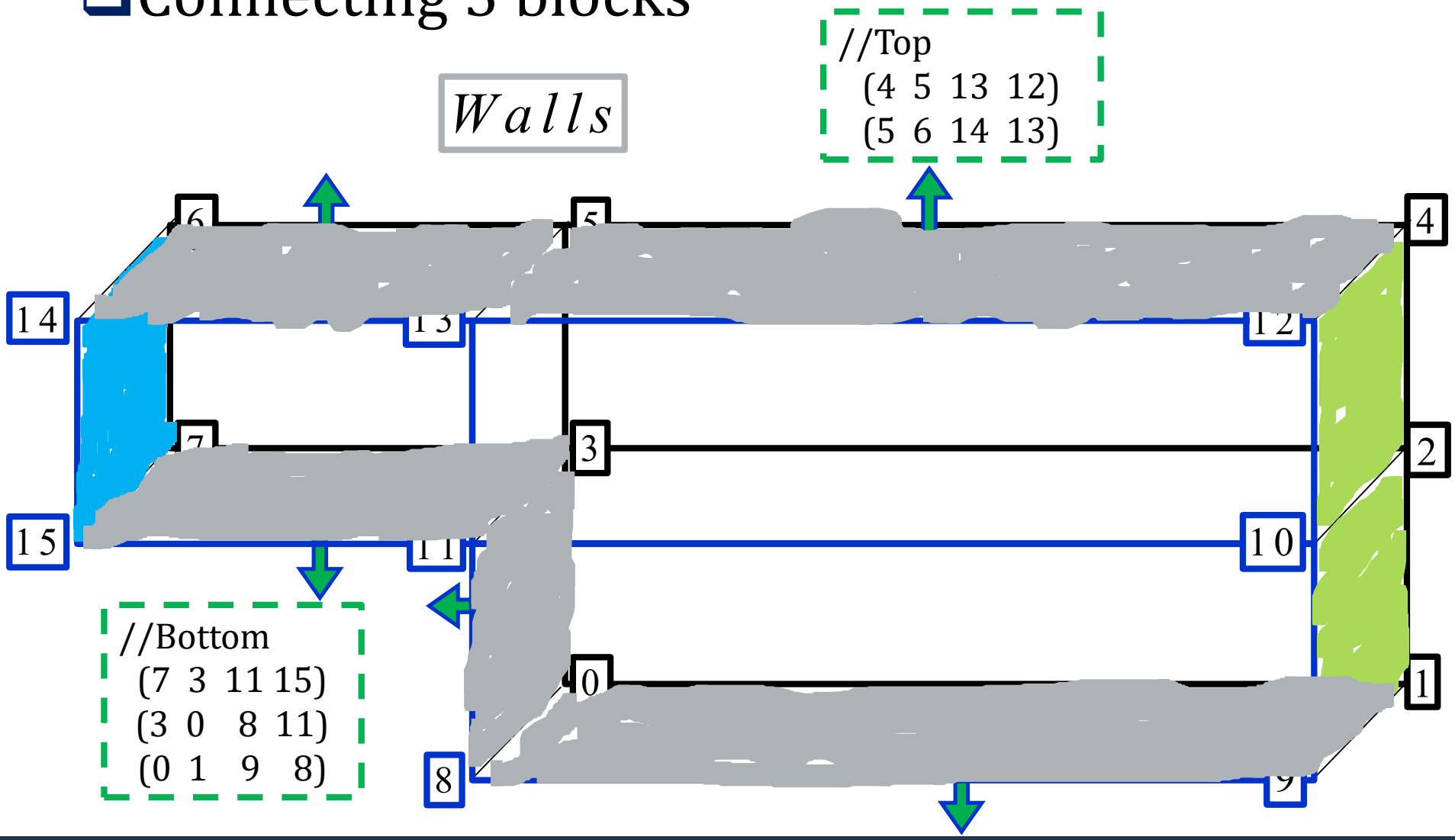
Multiblock Mesh: Case03

- Connecting 3 blocks



Multiblock Mesh: Case03

- Connecting 3 blocks

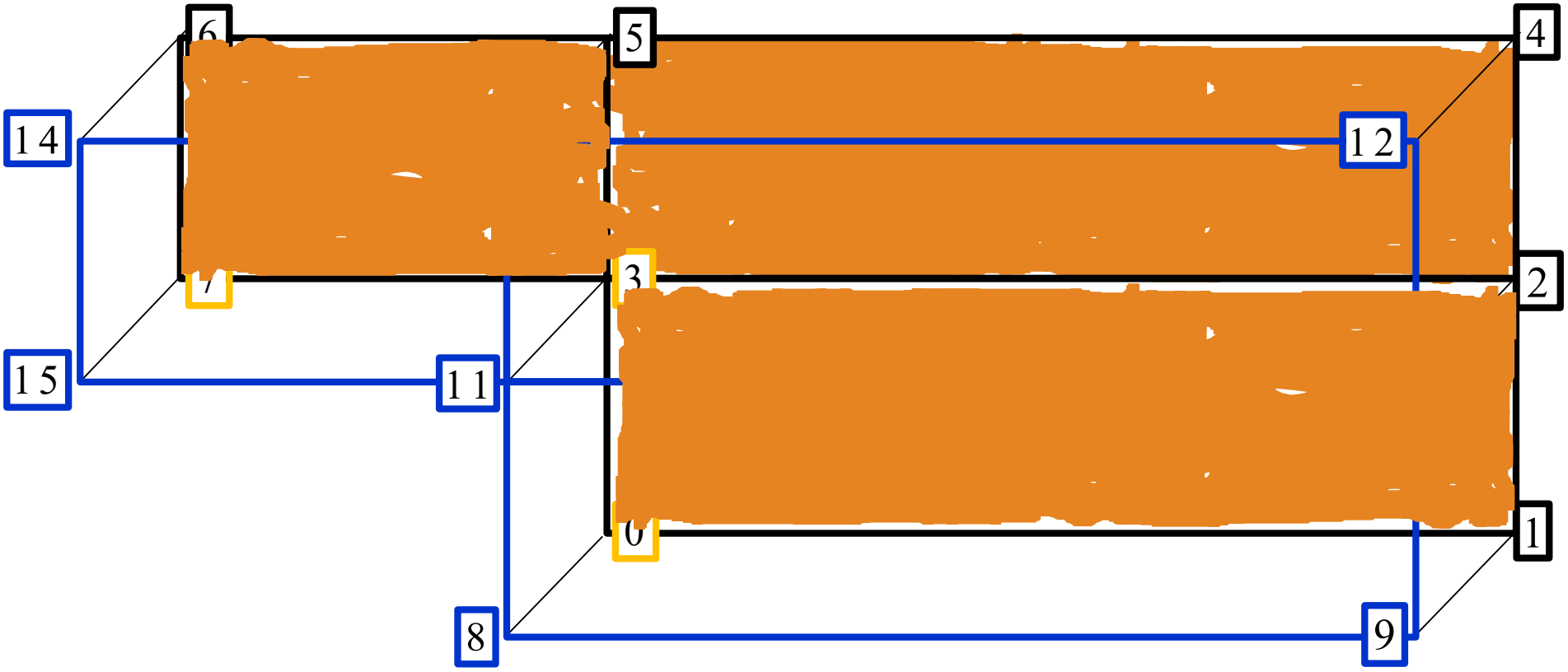


Multiblock Mesh: Case03

- Connecting 3 blocks

frontAndBack

```
//Back
(0 /**/)
(3 /**/)
(7 /**/)
```

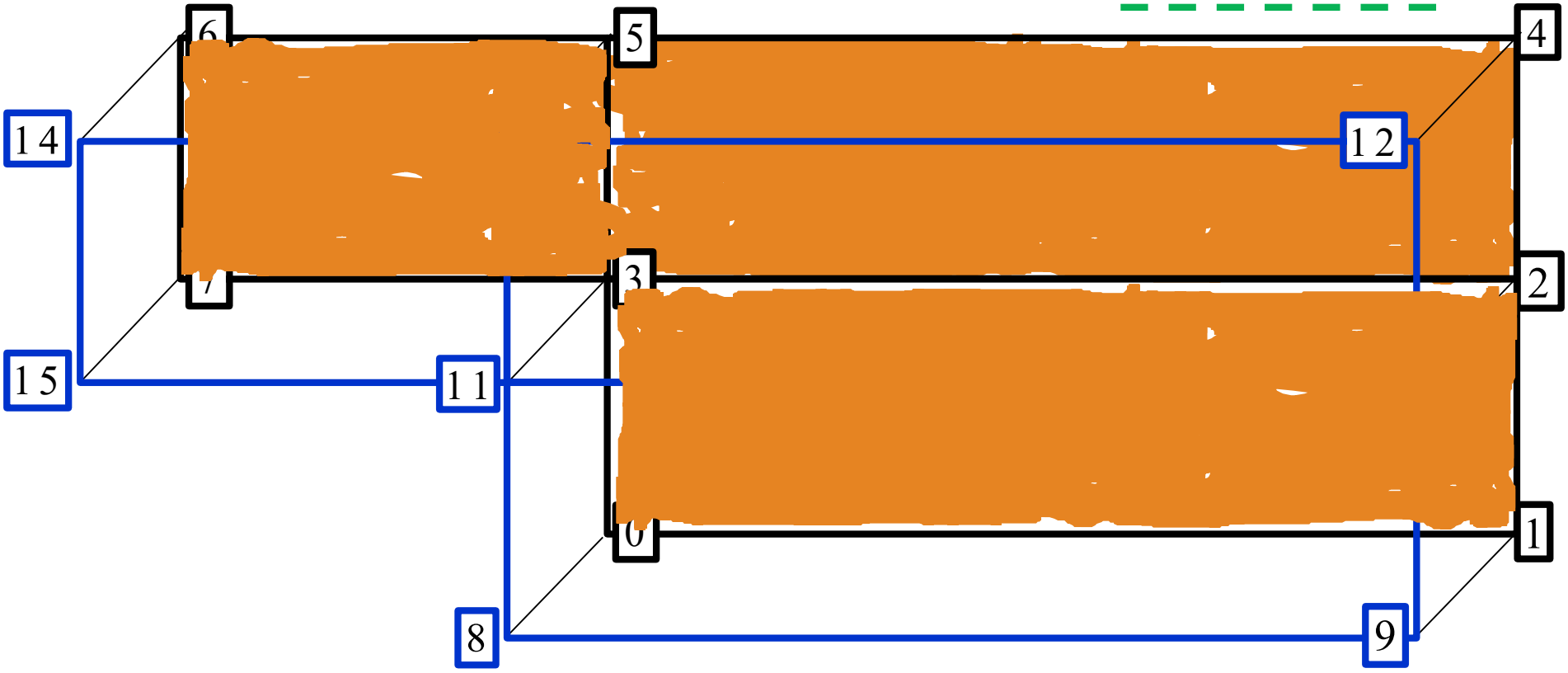


Multiblock Mesh: Case03

- Connecting 3 blocks

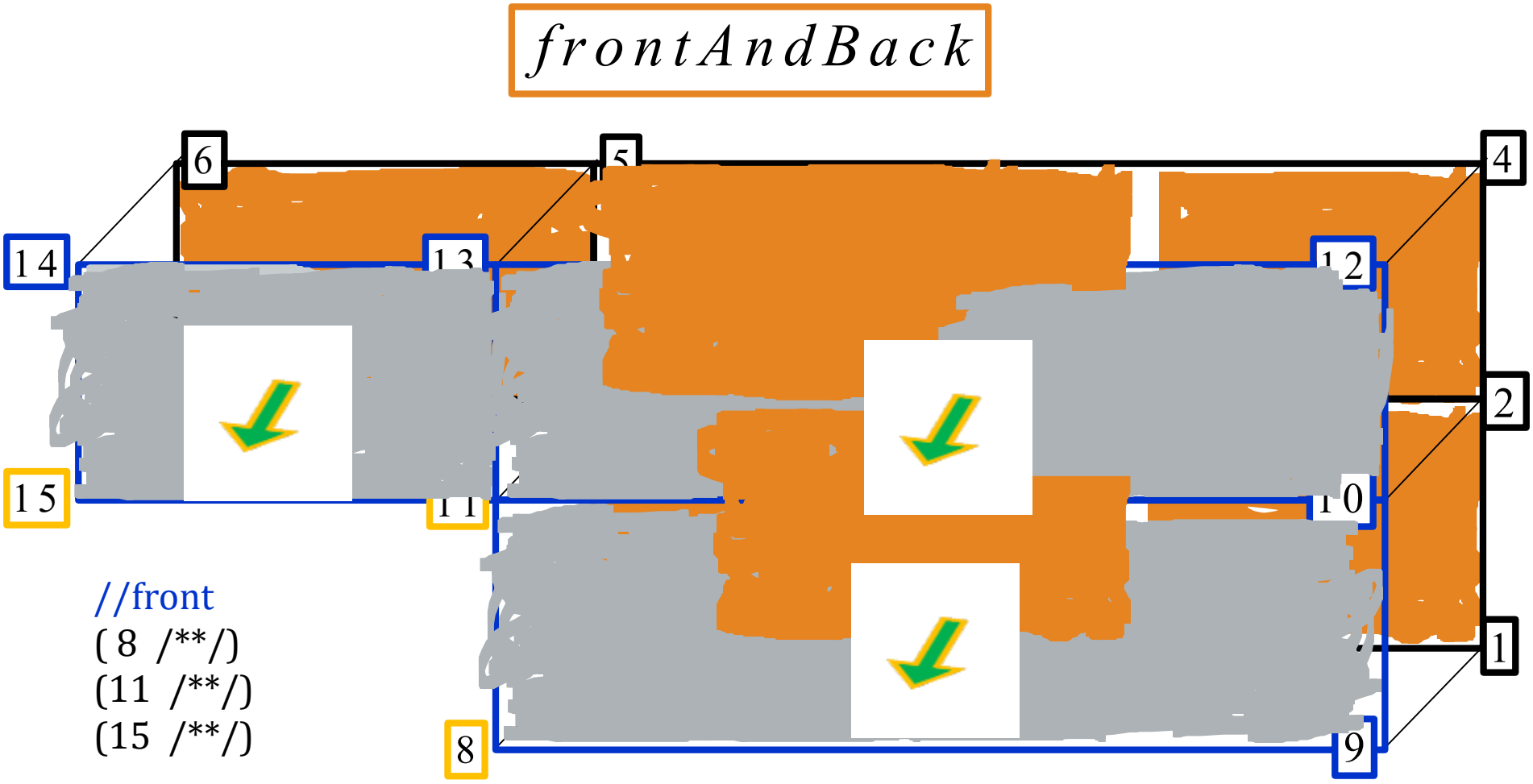
frontAndBack

```
//Back
    (0 3 2 1)
    (3 5 4 2)
    (7 6 5 3)
```



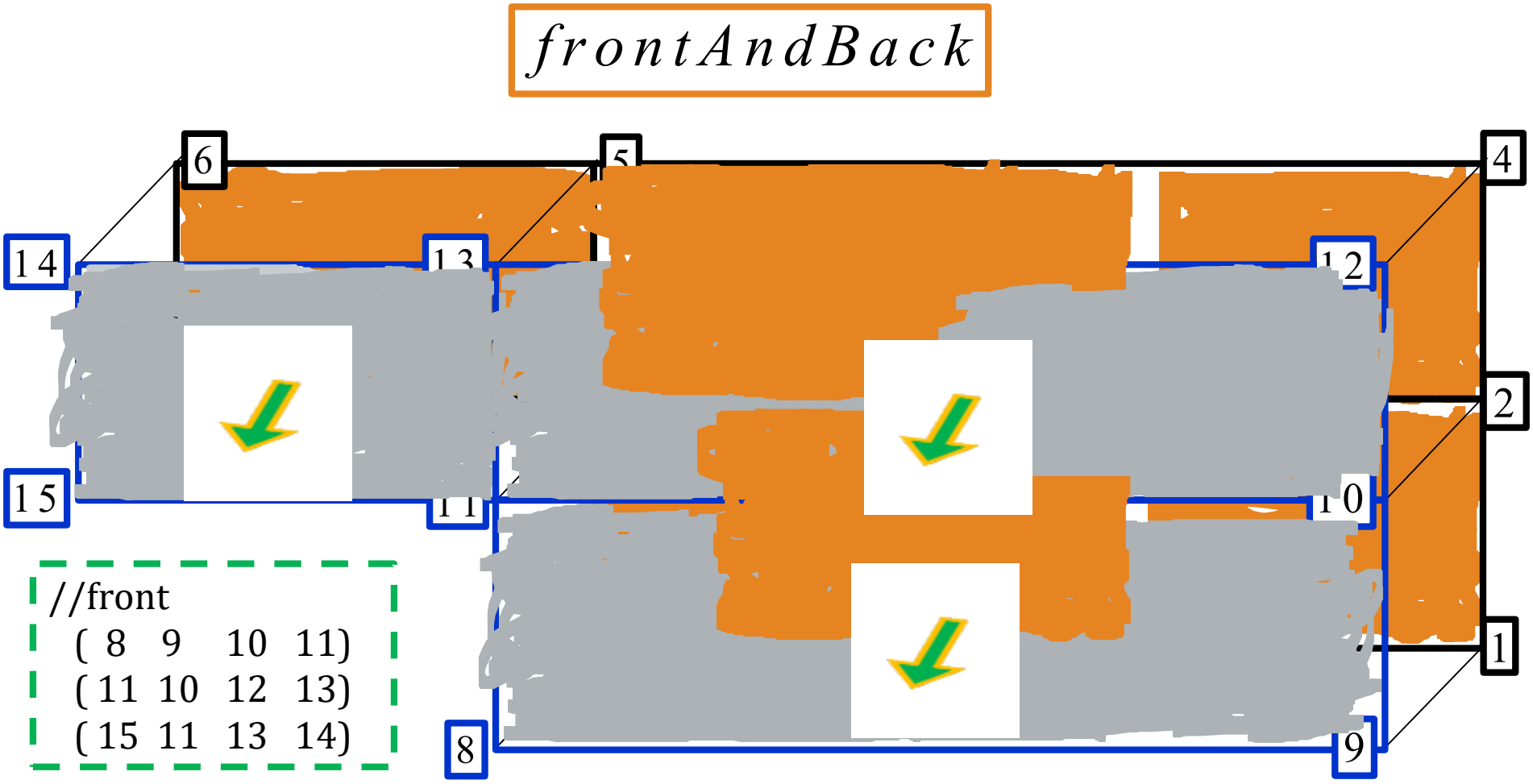
Multiblock Mesh: Case03

- Connecting 3 blocks



Multiblock Mesh: Case03

- Connecting 3 blocks



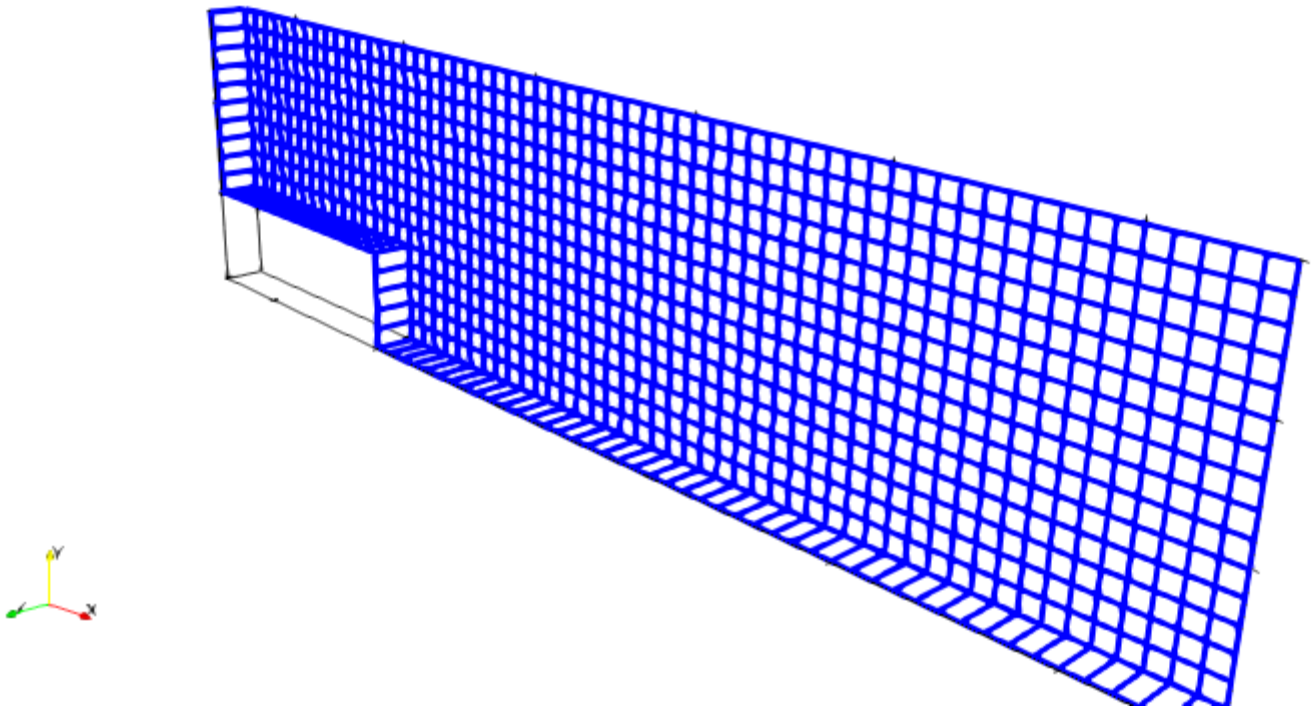
Multiblock Mesh: Case03

❑ boundary

```
51 boundary
52 (
53     inlet
54     {
55         type patch;
56         faces
57         (
58             (6 7 15 14)
59         );
60     }
61     outlet
62     {
63         type patch;
64         faces
65         (
66             ( 9 1 2 10)
67             (10 2 4 12)
68         );
69     }
70     walls
71     {
72         type wall;
73         faces
74         (
75             //Bottom
76             ( 7  3 11 15)
77             ( 3  0  8 11)
78             ( 0  1  9  8)
79
80             //Top
81             ( 4  5 13 12)
82             ( 5  6 14 13)
83         );
84     }
85     frontAndBack
86     {
87         type empty;
88         faces
89         (
90             //Back
91             (0 3 2 1)
92             (3 5 4 2)
93             (7 6 5 3)
94
95             //front
96             ( 8  9 10 11)
97             (11 10 12 13)
98             (15 11 13 14)
99         );
100    }
101 );
```

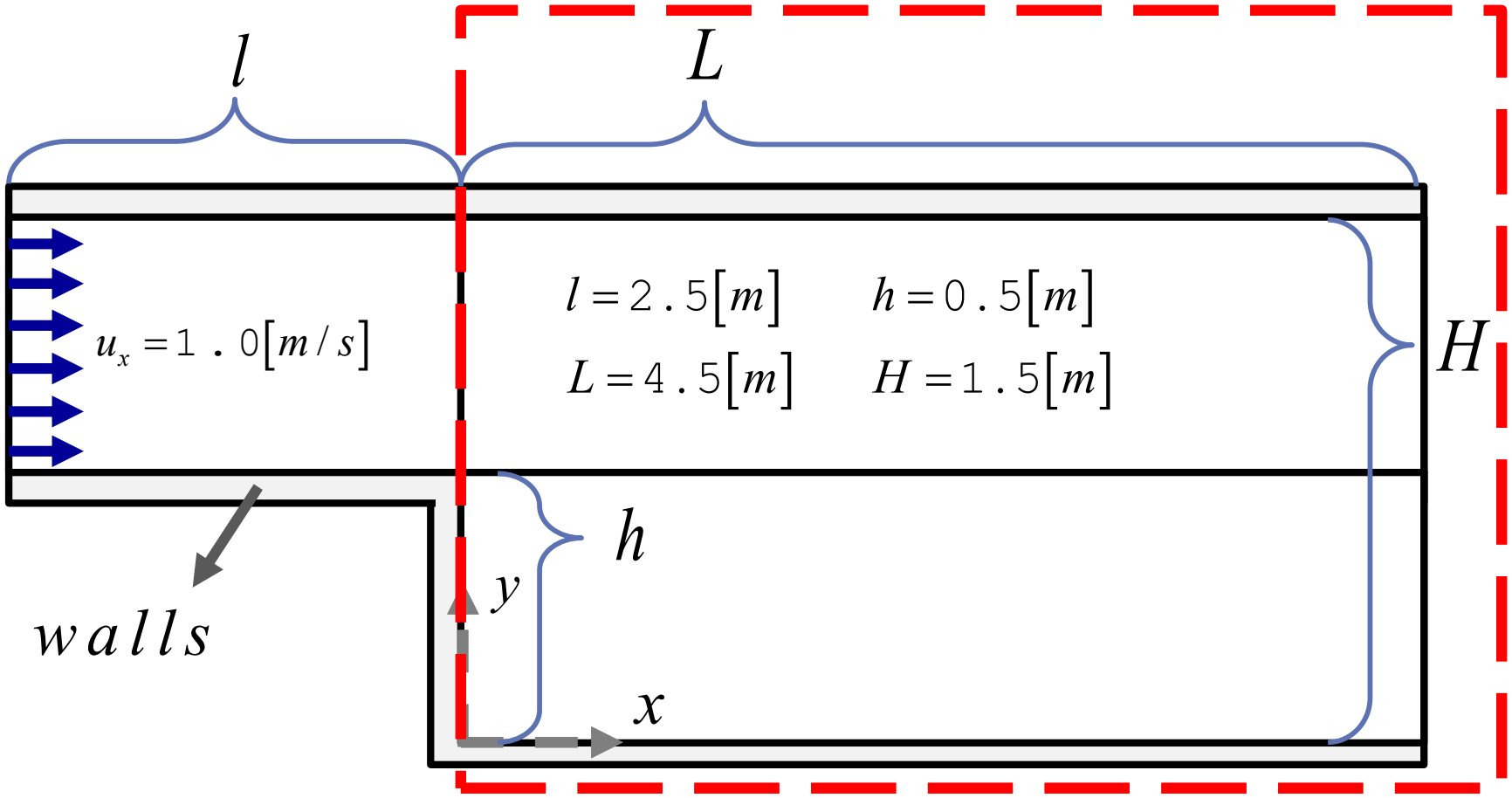
Multiblock Mesh: Case03

- blockMesh result



Curved surfaces: Case04

Geometry parametrization



Curved surfaces: Case04

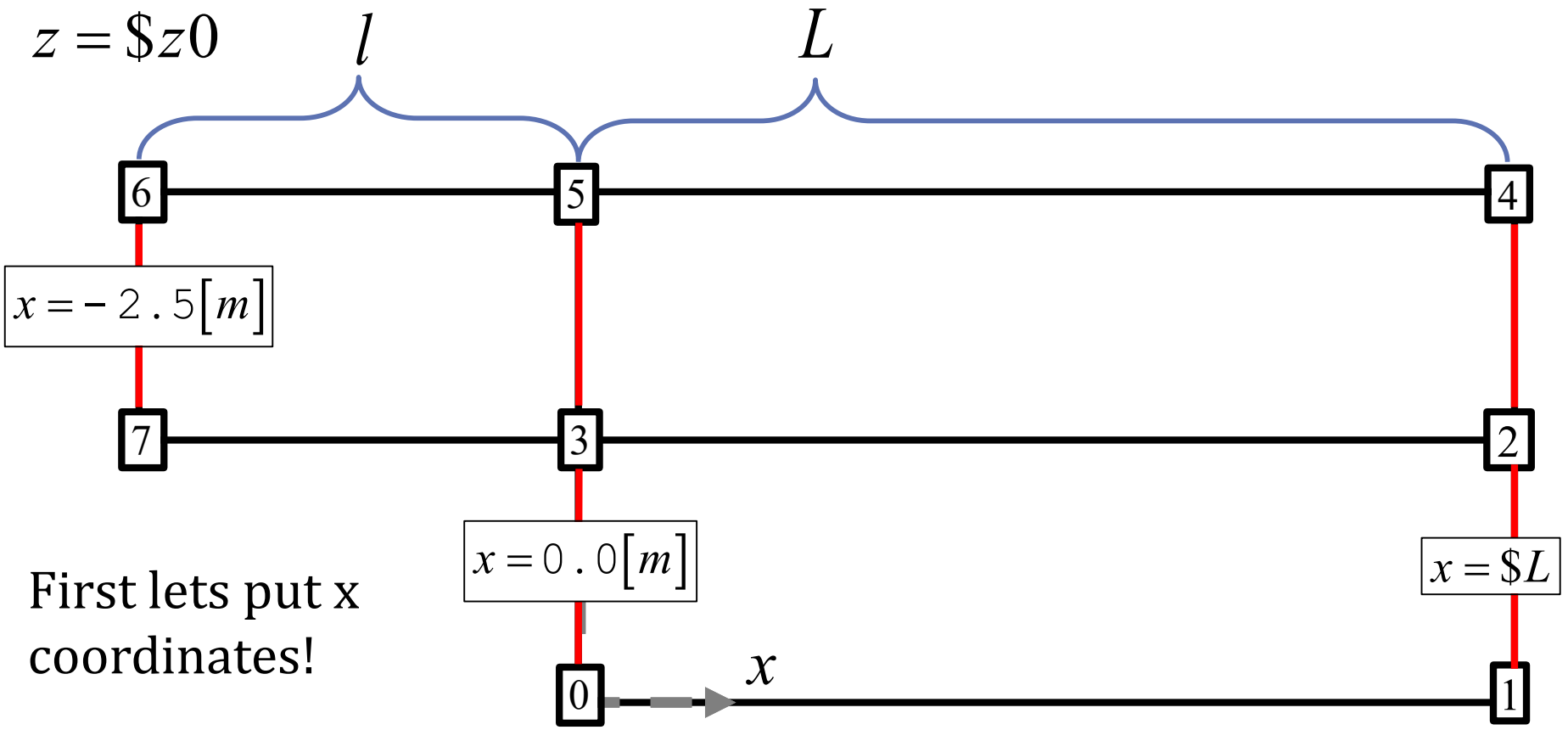
blockMesh

```
17 convertToMeters 1.0;
18
19 // Geometry Parametrization
20     //Lenghts
21     L    4.5;
22     //Heigths
23     h    0.5;
24     H    1.5;
25     //depth
26     z0 -0.1; // back plane
27     z1  0.1; // front plane
28
```

```
29 vertices
30 (
31     ( 0.0   0.0   $z0  )    //0
32     (  X    0.0   $z0  )    //1
33     (  X     Y    $z0  )    //2
34     ( 0.0    Y    $z0  )    //3
35     (  X     Y    $z0  )    //4
36     ( 0.0    Y    $z0  )    //5
37     (-2.5    Y    $z0  )    //6
38     (-2.5    Y    $z0  )    //7
39
40     ( 0.0   0.0   $z1  )    //8
41     (  X    0.0   $z1  )    //9
42     (  X     Y    $z1  )    //10
43     ( 0.0    Y    $z1  )    //11
44     (  X     Y    $z1  )    //12
45     ( 0.0    Y    $z1  )    //13
46     (-2.5    Y    $z1  )    //15
47     (-2.5    Y    $z1  )    //16
48 );
```

Curved surfaces: Case04

- Connecting 3 blocks

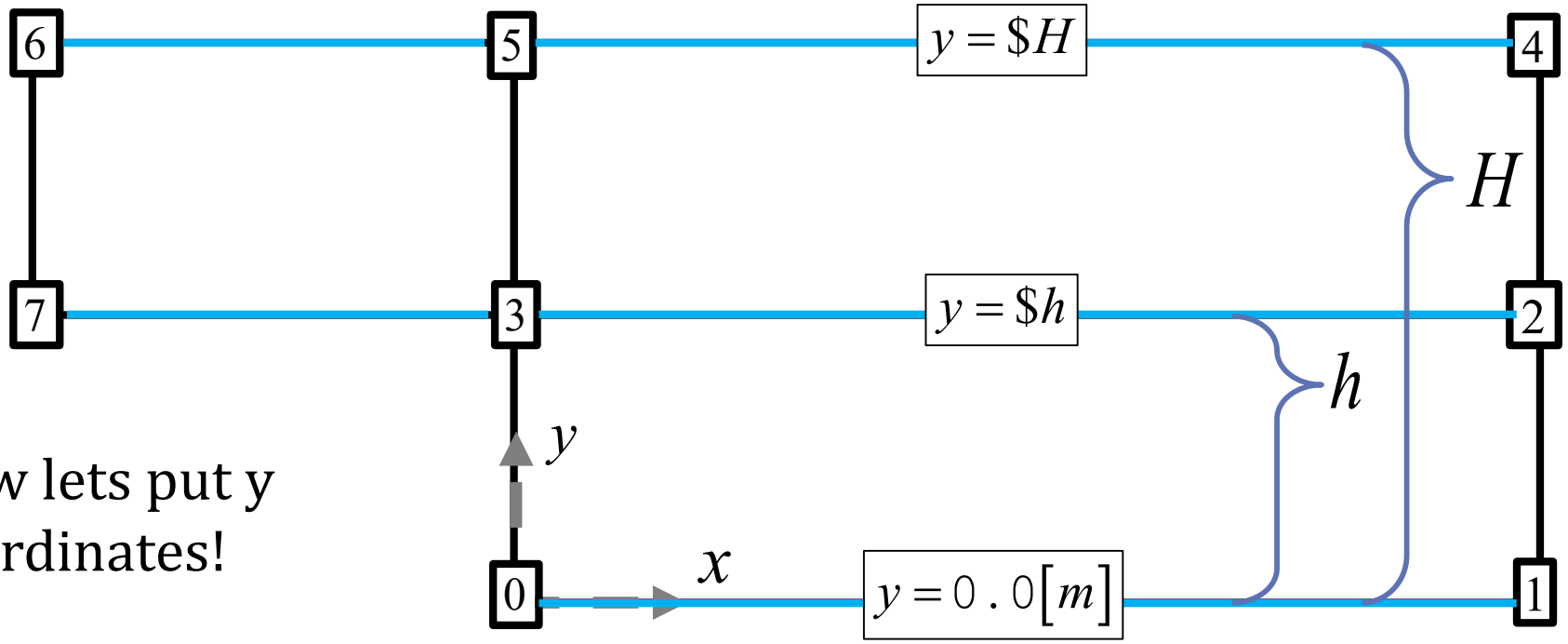


First lets put x coordinates!

Curved surfaces: Case04

- Connecting 3 blocks

$z = \$z0$



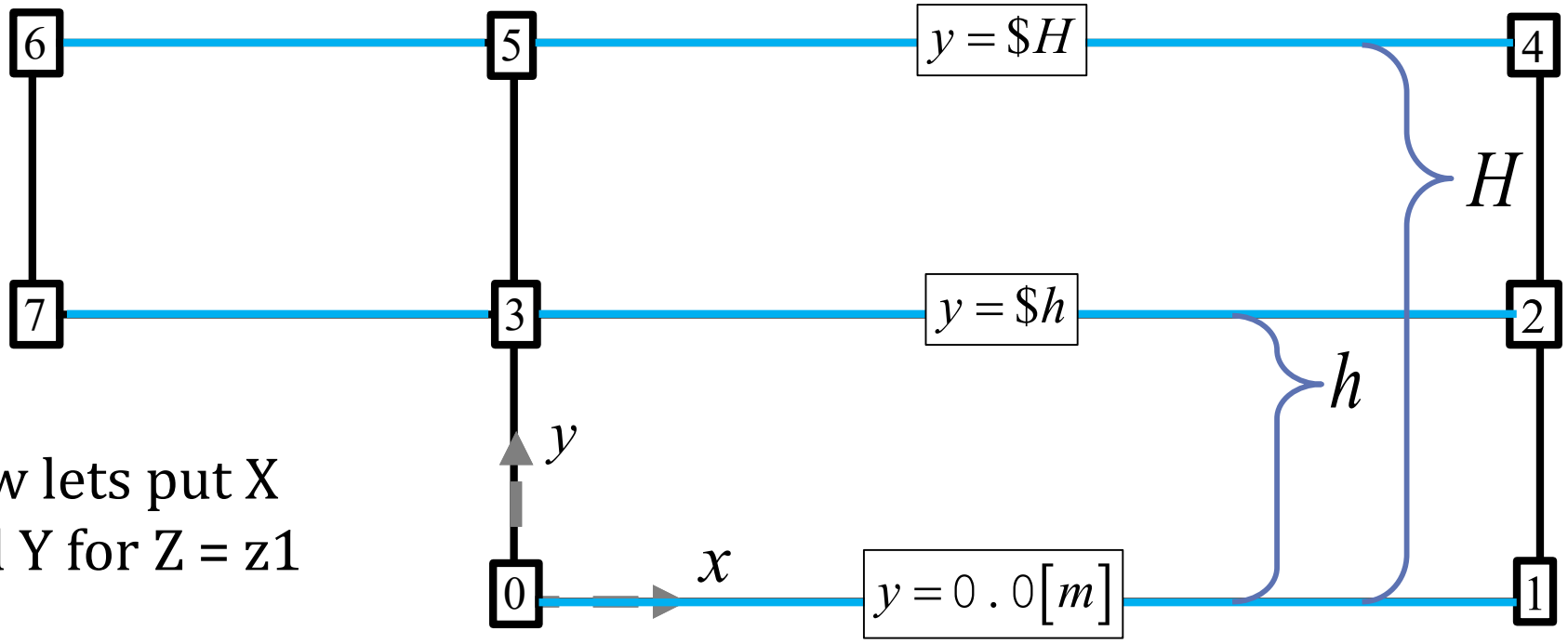
Now lets put y coordinates!

Curved surfaces: Case04

Connecting 3 blocks

$z = \$z0$

$z = \$z1$

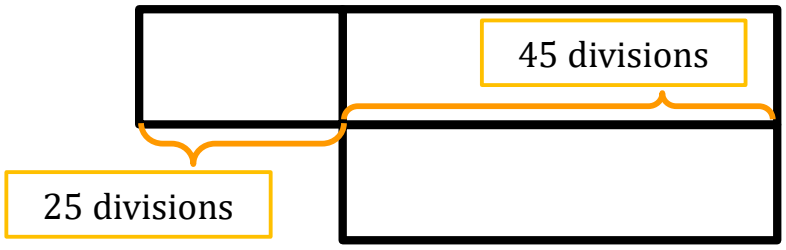


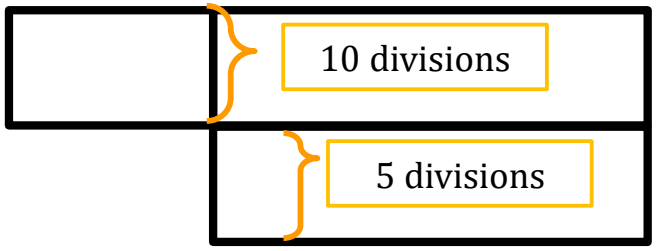
Now lets put X and Y for Z = z1

Curved surfaces: Case04

Geometry parametrization

```
50 blocks
51 (
52 // Mesh refinement controls
53
54     NX 45; // X direction Downstream Step
55     NY  5; // Y direction Downstream Step Step height
56     Nx 25; // x direction Upstream Step
57     Ny 10; // y direction Upstream Step inlet channel height
58     Nz  1; // Z direction Channel depth
59
60     hex (0 1 2 3  8  9 10 11) (      ) simpleGrading (1 1 1)
61     hex (3 2 4 5 11 10 12 13) (      ) simpleGrading (1 1 1)
62     hex (7 3 5 6 15 11 13 14) (      ) simpleGrading (1 1 1)
63 );
```

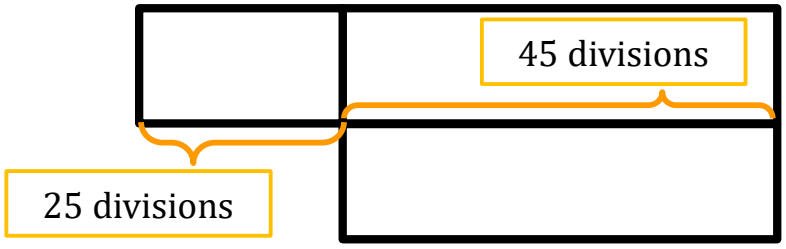


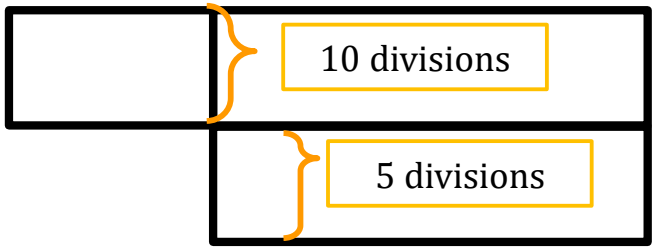


Curved surfaces: Case04

Geometry parametrization

```
50 blocks
51 (
52 // Mesh refinement controls
53
54     NX 45; // X direction Downstream Step
55     NY  5; // Y direction Downstream Step Step height
56     Nx 25; // x direction Upstream Step
57     Ny 10; // y direction Upstream Step inlet channel height
58     Nz  1; // Z direction Channel depth
59
60    hex (0 1 2 3  8  9 10 11) ($NX $NY $Nz) simpleGrading (1 1 1)
61    hex (3 2 4 5 11 10 12 13) ($NX $Ny $Nz) simpleGrading (1 1 1)
62    hex (7 3 5 6 15 11 13 14) ($Nx $Ny $Nz) simpleGrading (1 1 1)
63 );
```





Curved surfaces: Case04

- blockMesh :

```
50 blocks
51 (
52 // Mesh refinement controls
53
54     NX 45; // X direction Downstream Step
55     NY  5; // Y direction Downstream Step Step height
56     Nx 25; // x direction Upstream Step
57     Ny 10; // y direction Upstream Step inlet channel height
58     Nz  1; // Z direction Channel depth
59
60     hex (0 1 2 3  8  9 10 11) (      ) simpleGrading (1 1 1)
61     hex (3 2 4 5 11 10 12 13) (      ) simpleGrading (1 1 1)
62     hex (7 3 5 6 15 11 13 14) (      ) simpleGrading (1 1 1)
63 );
```

Curved surfaces: Case04

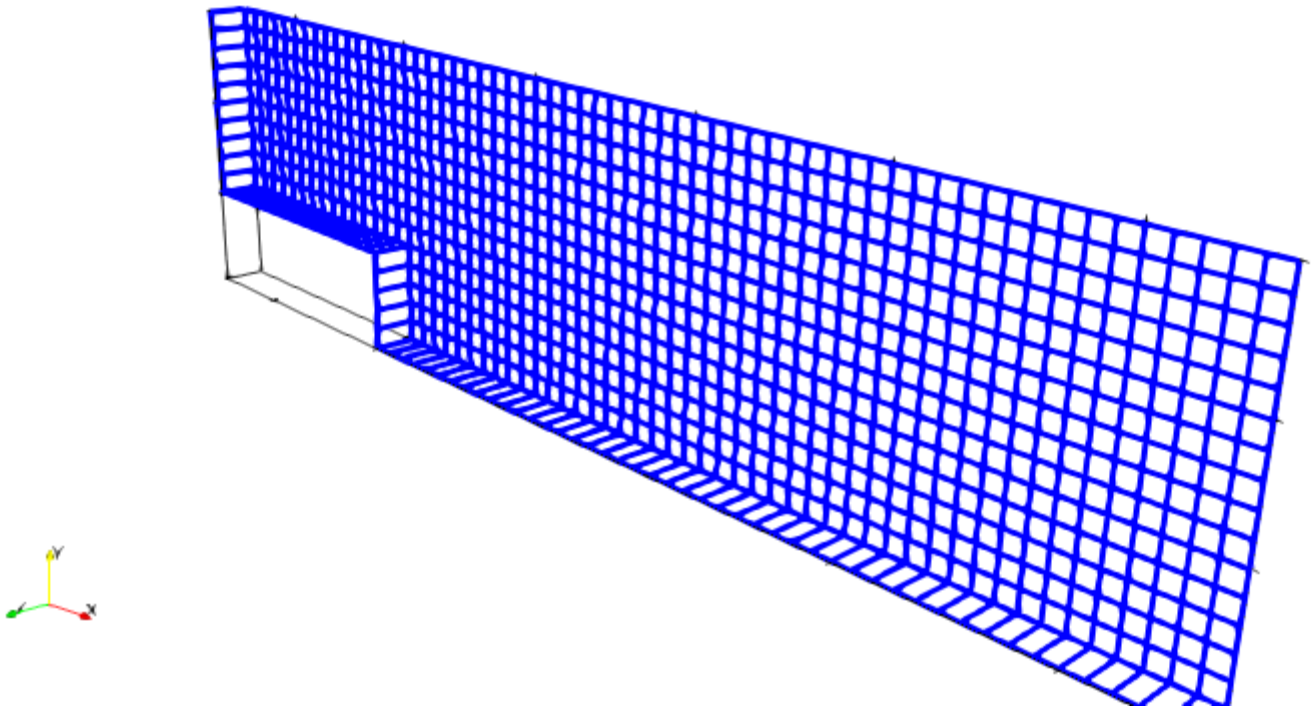
- blockMesh : ERROR Why?

```
50 blocks
51 (
52 // Mesh refinement controls
53
54     NX 45; // X direction Downstream Step
55     NY  5; // Y direction Downstream Step Step height
56     Nx 25; // x direction Upstream Step
57     Ny 10; // y direction Upstream Step inlet channel height
58     Nz  1; // Z direction Channel depth
59
60     hex (0 1 2 3  8  9 10 11) (      ) simpleGrading (1 1 1)
61     hex (3 2 4 5 11 10 12 13) (      ) simpleGrading (1 1 1)
62     hex (7 3 5 6 15 11 13 14) (      ) simpleGrading (1 1 1)
63 );
```

Ctrl + X and Ctrl + V

Curved surfaces: Case04

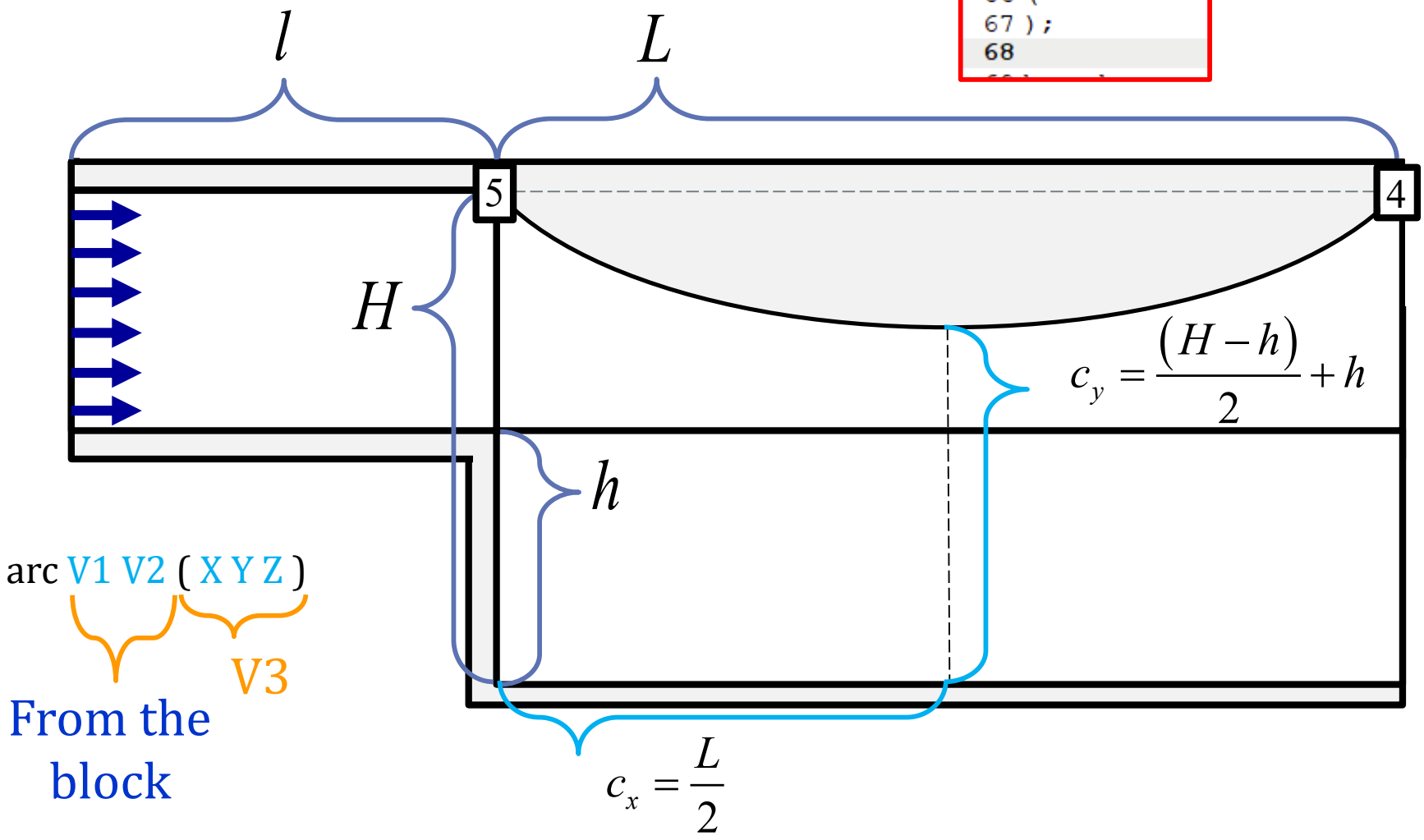
- run blockMesh



- Working smarter is better than harder, no?!

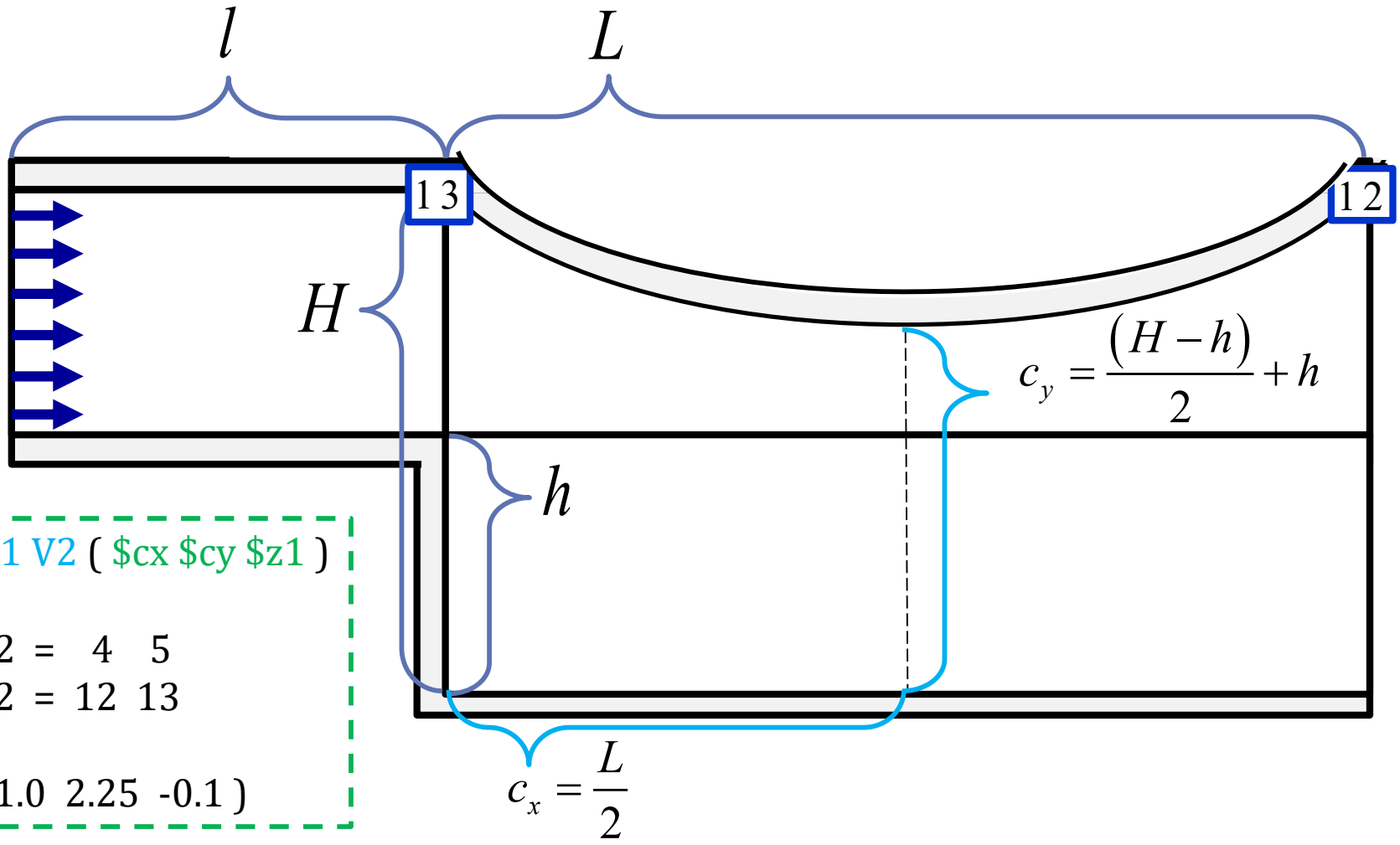
Creating a curved surface

```
64
65 edges
66 (
67 );
68
```



Curved surfaces: Case04

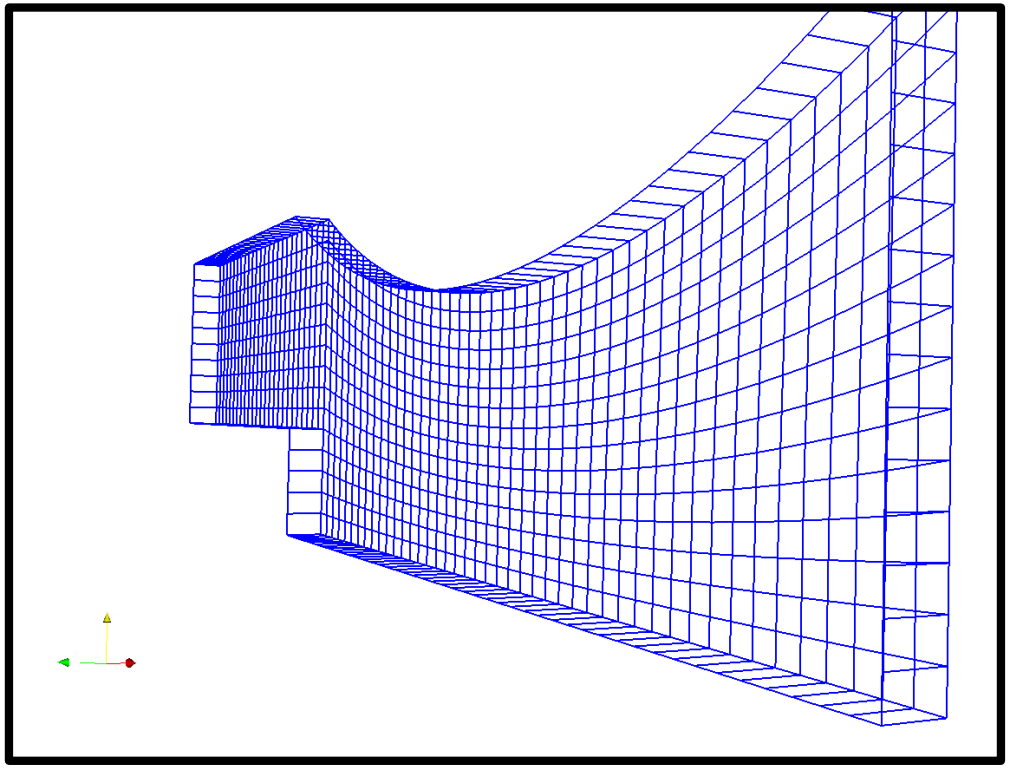
Creating a curved surface




```
arc V1 V2 ( $cx $cy $z1 )

V1 V2 =   4   5
V1 V2 =  12  13

 or ( 1.0  2.25  -0.1 )
```

Curved surfaces: Case04

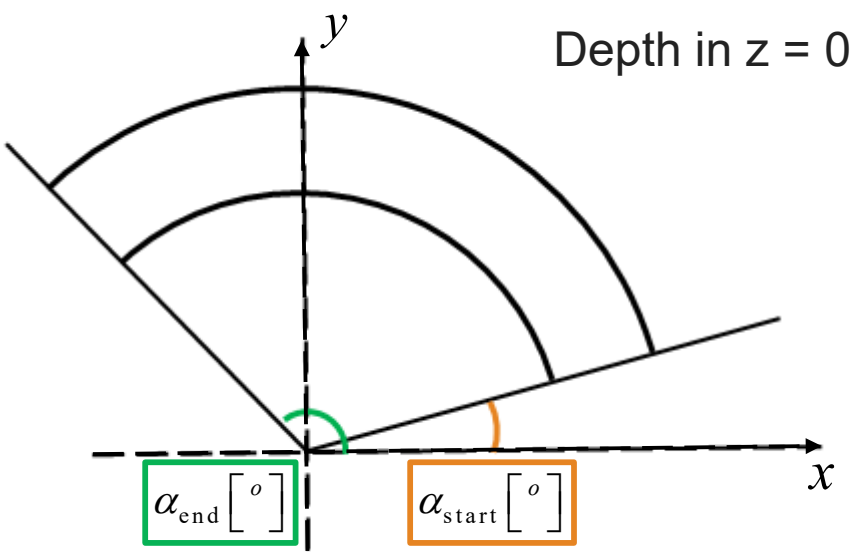
- blockMesh



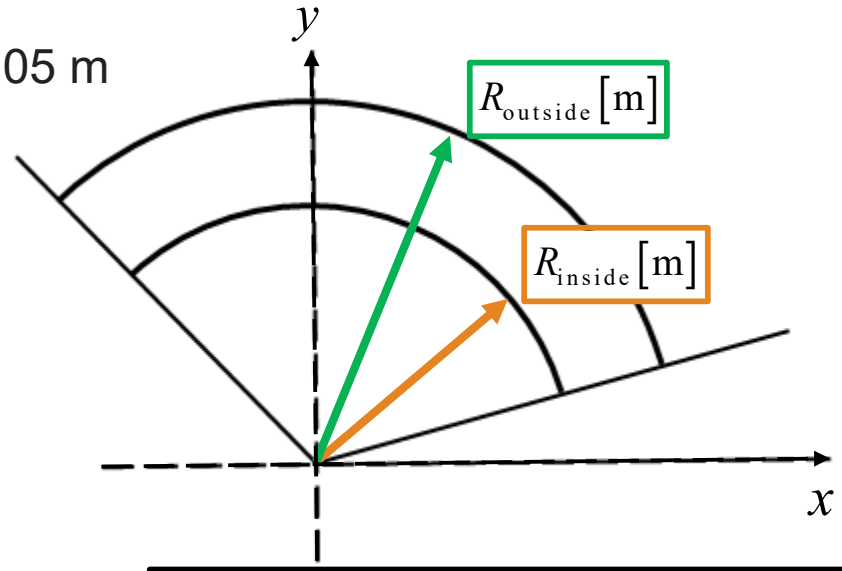
- Working smarter is better than harder, no?!

3D Geometry: Case05

Curved Tube



Depth in z = 0.05 m

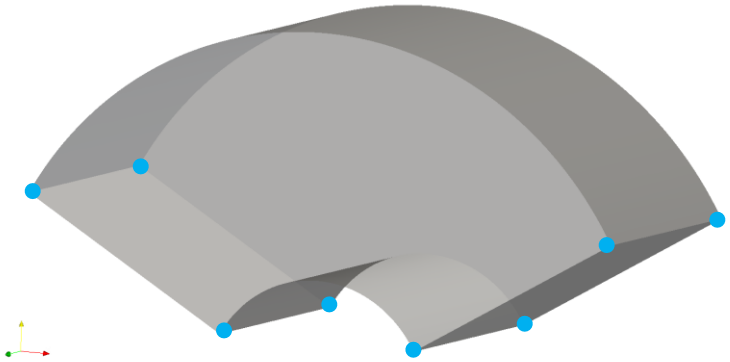


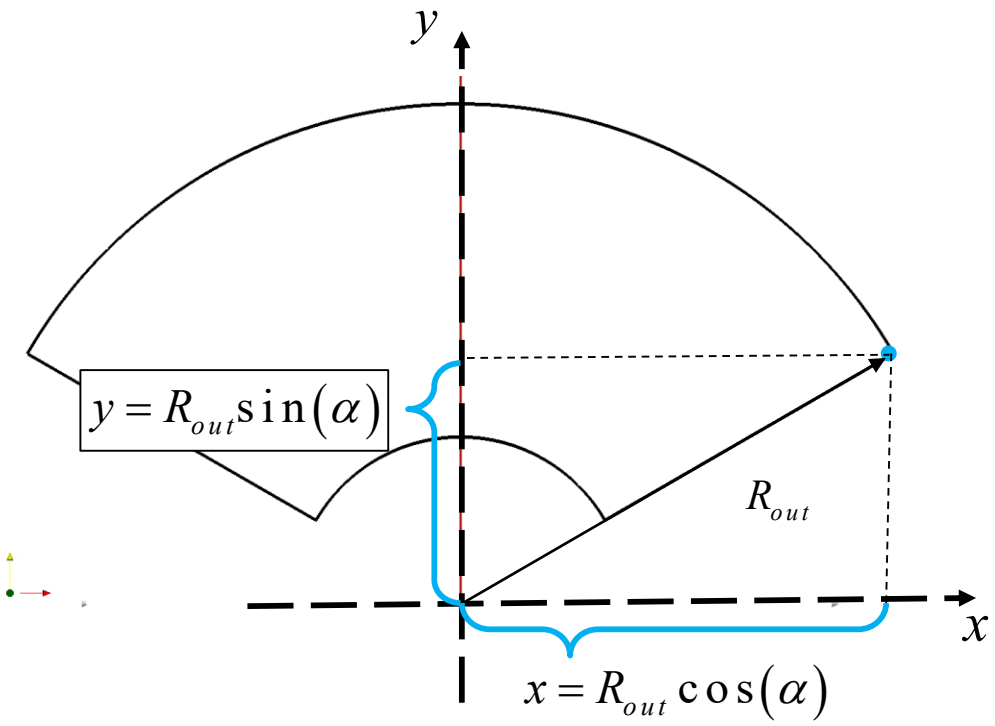

```
//Angles
    // Angle of Inlet plane (Where the Circle
    AlphaStartDeg     030.000000;
    // Angle of Inlet plane (Where the Circle
    AlphaEndDeg       150.000000;

//depth for the tube
    z0 -10.0; // back plane
    z1  10.0; // front plane
```

```
17 // Geometry Parametrization
18
19   /*-----------------------------
20   /*User input Section ----------
21
22     //Radius
23         //Outside Circle radius
24         Rout     30.0;
25         //Inside Circle radius
26         Rin      10.0;
```

3D Geometry: Case05

Curved Tube



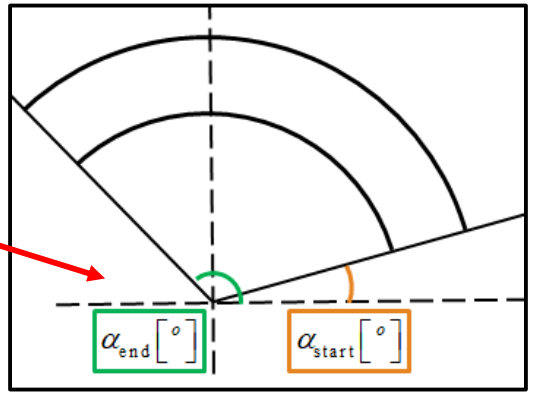


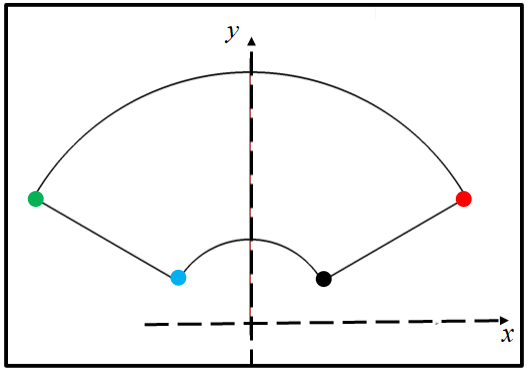
So for combinations of "R_i" and "α_j" we will have pairs of "x" and "y"

❑ Curved Tube

So for combinations of "R_i" and "α_j" we will have pairs of "x" and "y"

```
42  /*---------------------------------------------------------------
43  /*Mathematical Operations-----------------------------------------
44
45  //Convesions to get each corresponding rectangular coordinates
46    //Degrees to Radians
47        AlphaStart  #calc "degToRad($AlphaStartDeg)";
48        AlphaEnd    #calc "degToRad($AlphaEndDeg)";
49    //Starting rectangular coodinates
50      ● XRinAlphaS  #calc "$Rin*cos($AlphaStart)";
51      ● YRinAlphaS  #calc "$Rin*sin($AlphaStart)";
52
53      ● XRoutAlphaS #calc "$Rout*cos($AlphaStart)";
54      ● YRoutAlphaS #calc "$Rout*sin($AlphaStart)";
55
56    //Ending rectangular coordinates
57      ● XRinAlphaE  #calc "$Rin*cos($AlphaEnd)";
58      ● YRinAlphaE  #calc "$Rin*sin($AlphaEnd)";
59
60      ● XRoutAlphaE #calc "$Rout*cos($AlphaEnd)";
61      ● YRoutAlphaE #calc "$Rout*sin($AlphaEnd)";
62
63  /*---------------------------------------------------------------
64  /*- END Mth.O. SECTION --------------------------------------------
65
66 //BlockMesh file standard input:
```



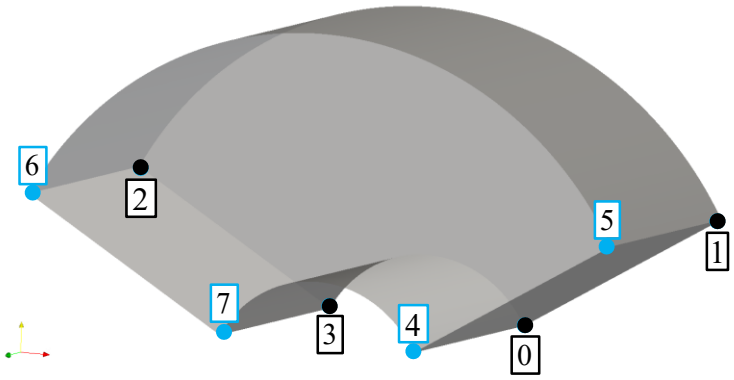


3D Geometry: Case05

❑ Curved Tube

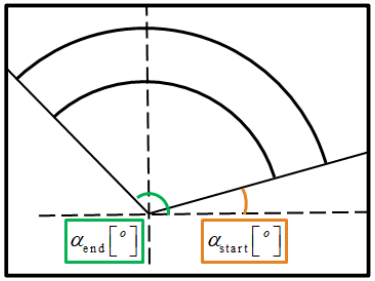
So for combinations of "R_i" and "α_j" we will have pairs of "x" and "y"

```
66 //BlockMesh file standard input:
67
68 convertToMeters 0.01;
69
70 vertices
71 (
72     //First Plane
73     ( $XRinAlphaS   $YRinAlphaS  $z0 )   //0
74     ( $XRoutAlphaS  $YRoutAlphaS $z0 )   //1
75     ( $XRoutAlphaE  $YRoutAlphaE $z0 )   //2
76     ( $XRinAlphaE   $YRinAlphaE  $z0 )   //3
77     //First Plane
78     ( $XRinAlphaS   $YRinAlphaS  $z1 )   //4
79     ( $XRoutAlphaS  $YRoutAlphaS $z1 )   //5
80     ( $XRoutAlphaE  $YRoutAlphaE $z1 )   //6
81     ( $XRinAlphaE   $YRinAlphaE  $z1 )   //7
82
83 );
```



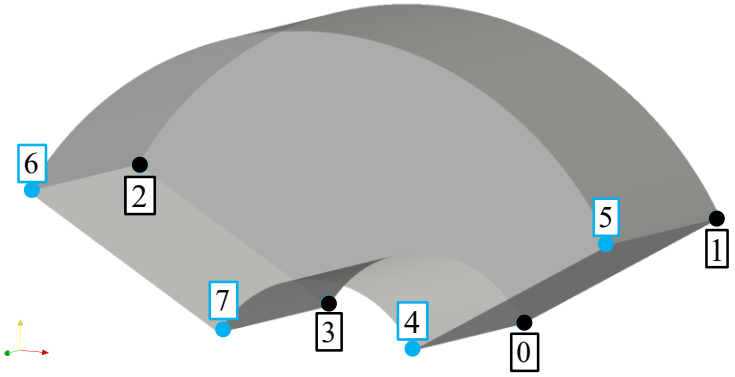

```
//Starting rectangular coodinates
    XRinAlphaS   #calc "$Rin*cos($AlphaStart)";
    YRinAlphaS   #calc "$Rin*sin($AlphaStart)";

    XRoutAlphaS  #calc "$Rout*cos($AlphaStart)";
    YRoutAlphaS  #calc "$Rout*sin($AlphaStart)";
```

```
//Ending rectangular coordinates
    XRinAlphaE   #calc "$Rin*cos($AlphaEnd)";
    YRinAlphaE   #calc "$Rin*sin($AlphaEnd)";

    XRoutAlphaE  #calc "$Rout*cos($AlphaEnd)";
    YRoutAlphaE  #calc "$Rout*sin($AlphaEnd)";
```



3D Geometry: Case05

Curved Tube

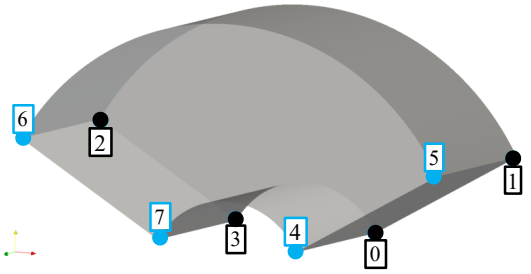


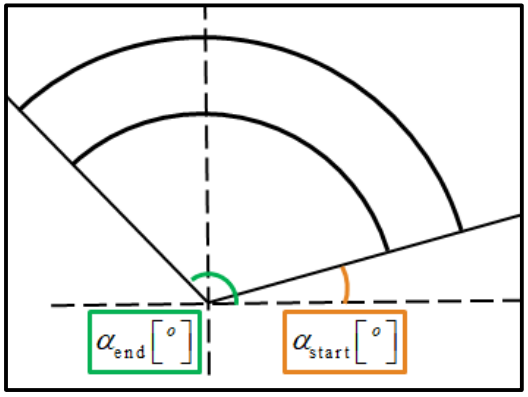

```
// U.I. Mesh refinement controls

    NR      15;  // Number of cells through the radial direction
    NAlpha  150; // Number of cell in the Alpha angle direction
    Nz      15;  // Z direction Channel depth

blocks
(
    hex (0 1 2 3 4 5 6 7) ($NR $NAlpha $Nz) simpleGrading (1 1 1)
);
```

3D Geometry: Case05

Curved Tube

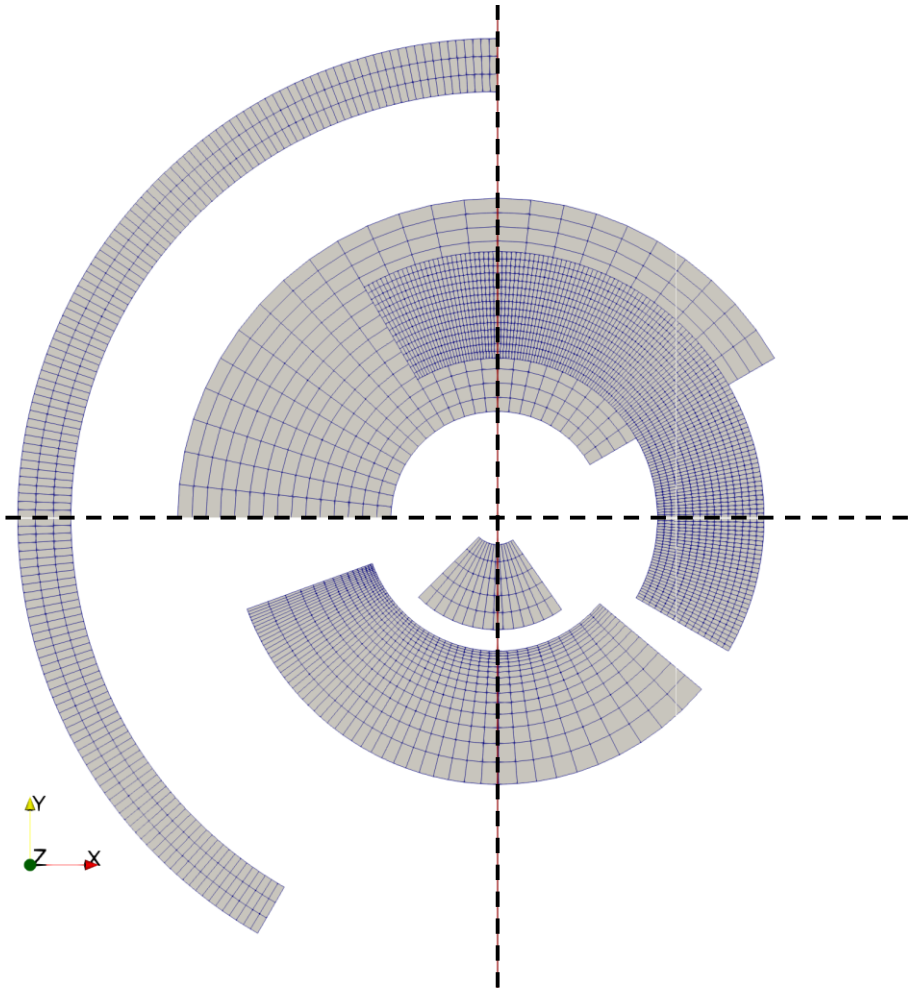
```
// Arc Parametrization

    //Intermediate alpha angle:
    AlphaAVGDeg #calc   "($AlphaStartDeg+$AlphaEndDeg)*0.5";
    AlphaAVG    #calc   "degToRad($AlphaAVGDeg)";

    //Convesions to get arcs corresponding rectangular coordinates
    XArcOut     #calc   "$Rout*cos($AlphaAVG)";
    YArcOut     #calc   "$Rout*sin($AlphaAVG)";

    XArcIn      #calc   "$Rin*cos($AlphaAVG)";
    YArcIn      #calc   "$Rin*sin($AlphaAVG)";

edges
(
    arc  1  2  ( $XArcOut   $YArcOut    $z0 ) //Arc outside
    arc  5  6  ( $XArcOut   $YArcOut    $z1 )

    arc  0  3  ( $XArcIn    $YArcIn     $z0 ) //Arc midline
    arc  4  7  ( $XArcIn    $YArcIn     $z1 )

);

boundary
```

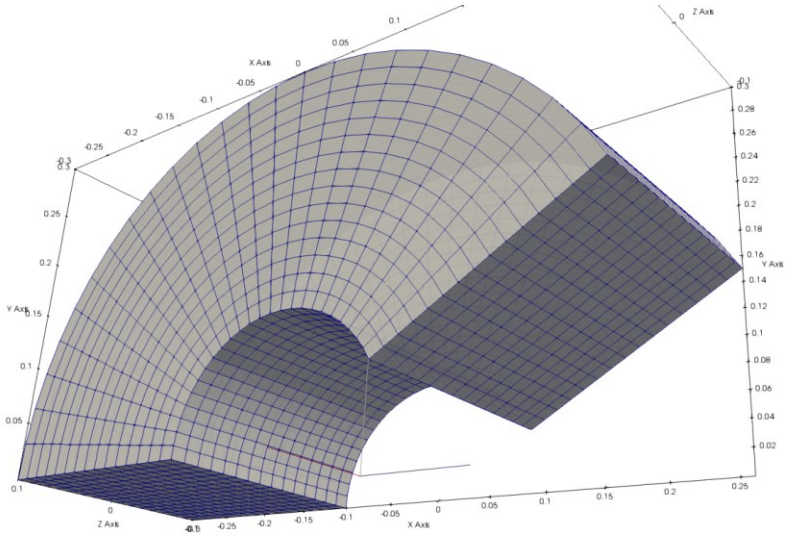




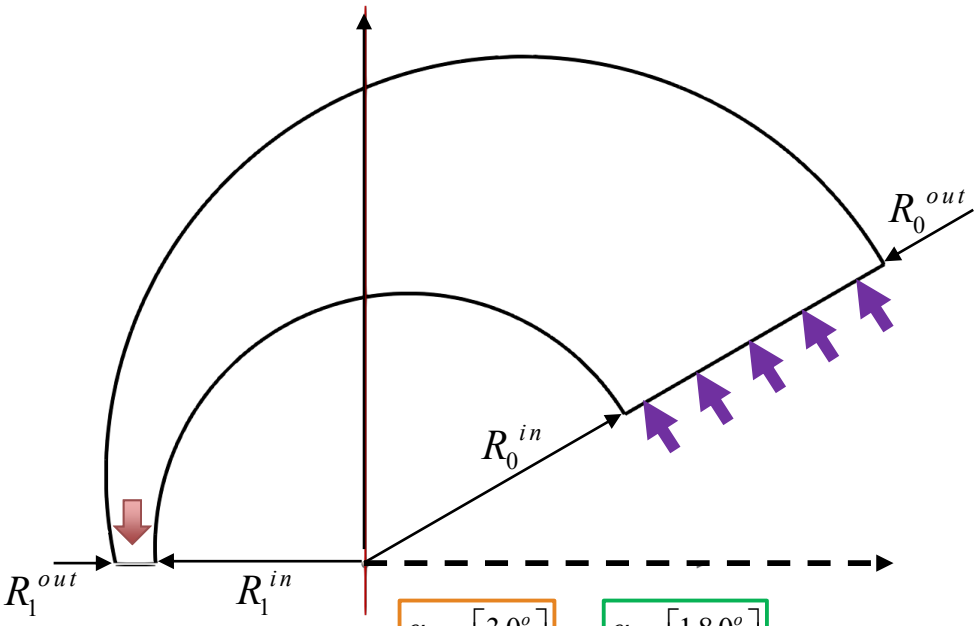
3D Geometry: Case05

Curved Tube





Curved Nozzle



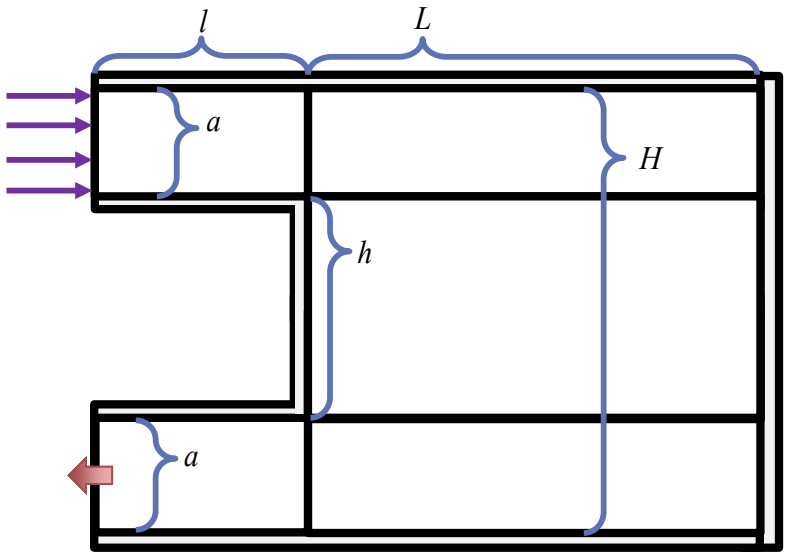
$R_0^{in} = 30.0[\text{cm}]$

$R_0^{out} = 15.0[\text{cm}]$

$R_1^{in} = 12.5[\text{cm}]$

$R_1^{out} = 10.0[\text{cm}]$

Squared elbow turn



$l = 5.5[m]$

$L = 1.5[m]$

$a = 0.15[m]$

$h = 0.25[m]$

$H = 2.5[m]$

inlets

outlets

Other faces are walls

Thank you for your attention!

Questions / Suggestions

Contact: wagnergaluppo@gmail.com