



FOAM

Guimarães

@PT 10-11
Julho
2015



**I Encontro em Portugal de utilizadores da
tecnologia Open▽FOAM**

Mesh generation with **blockMesh**

L.L. Ferrás, C. Fernandes, J.M. Nóbrega
Institute for Polymers and Composites (i3N), University of Minho

Programa Geral

| | 10 Julho (sexta-feira) | | 11 Julho (sábado) |
|----------------|--|---|---|
| 8:45 9:00 | Sessão de Abertura | | |
| 9:00 10:00 | Curso Básico (Sessões B1 e B2) | Curso Avançado (Sessões A1 e A2) | OpenFOAM: History, current capabilities and future perspectives (Hrvoje Jasak) |
| 10:00 12:30 | B1. Introdução ao OpenFOAM 9:00-10:30 | | Apresentações de trabalhos desenvolvidos em OpenFOAM |
| 10:00 12:30 | B2. Geração de malha com o blockMesh 11:00-12:30 | | |
| 12:30 14:00 | Almoço (Restaurante Vila Flor) | | |
| 14:00 17:30 | Curso Básico (Sessões B3 e B4) | Curso Avançado (Sessões A3 e A4) | Reunião Geral do grupo português de utilizadores de OpenFOAM |

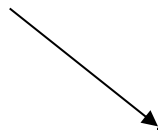
11:00-12:30

**B2 – Geração de malha com o
blockMesh**

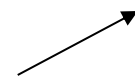
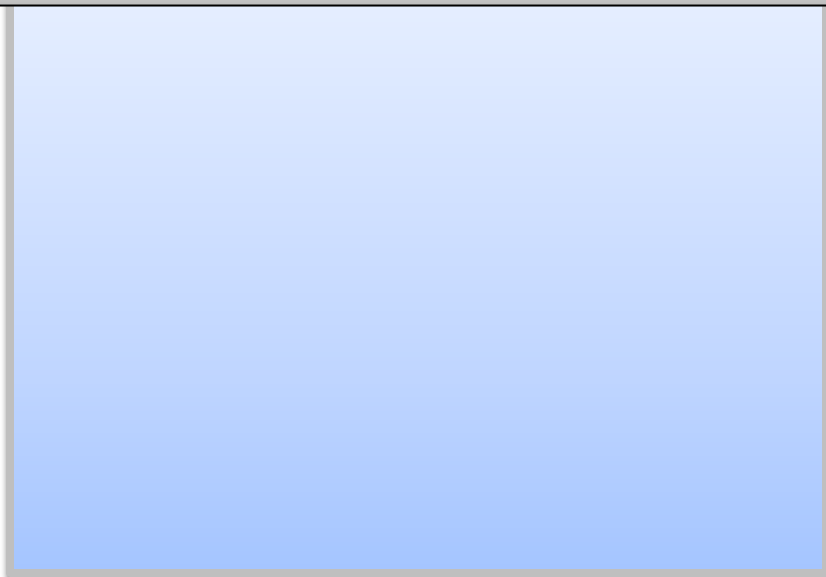
A2 – swak4Foam and pyFoam

Lid driven cavity

lid

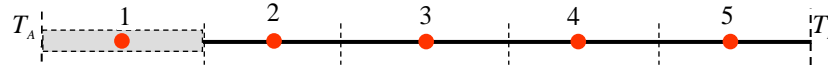


U



Container with
fluid

```
»blockMesh  
»checkMesh  
»icoFoam  
»paraFoam
```



**How to define the
mesh in
OpenFOAM?**

blockMeshDict

convertToMeters 1;

vertices

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

blocks

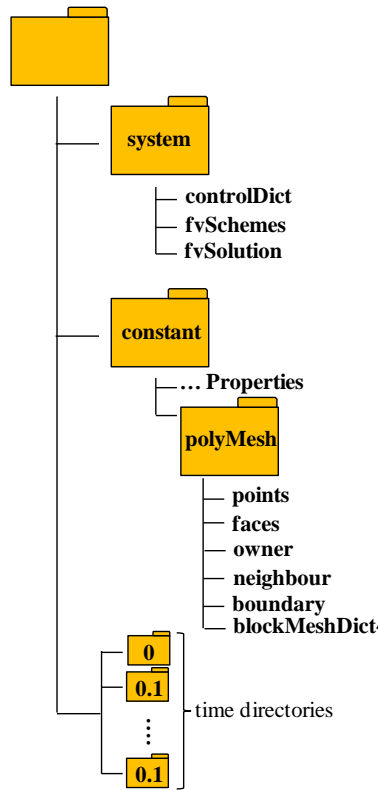
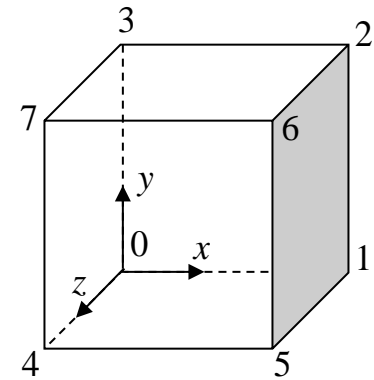
```
(hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1);
```

edges

```
( );
```

boundary

```
(  
  tempA  
  {  
    type inlet;  
    faces  
    ( (4 7 3 0) );  
  }  
  tempB  
  {  
    type outlet;  
    faces  
    ( (1 2 6 5) );  
  }  
  frontAndBack  
  {  
    type empty;  
    faces  
    ( (0 3 2 1)  
      (4 5 6 7)  
      (2 3 7 6)  
      (0 1 5 4) );  
  }  
);
```

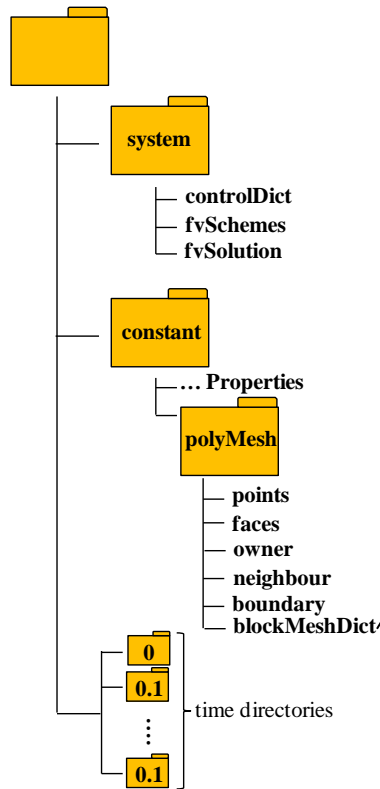
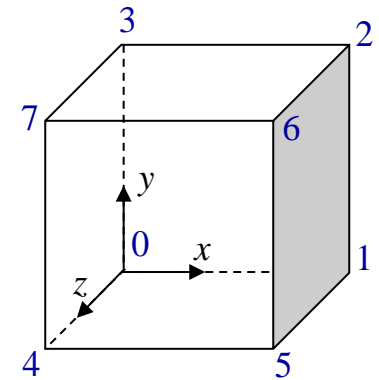


blockMeshDict

```
convertToMeters 1;
```

```
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  
);
```

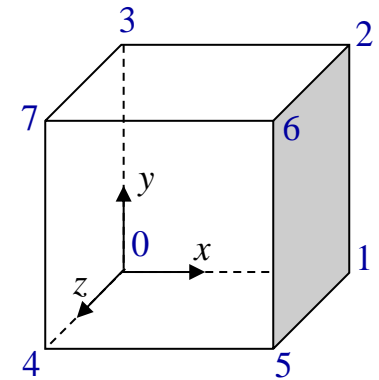


blockMeshDict

convertToMeters 1;

vertices

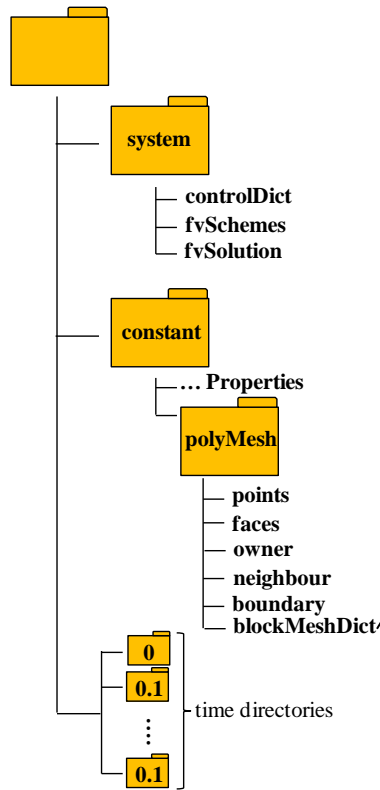
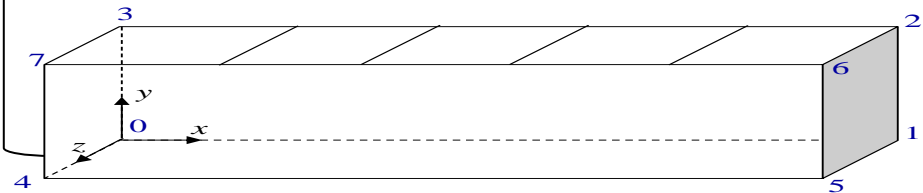
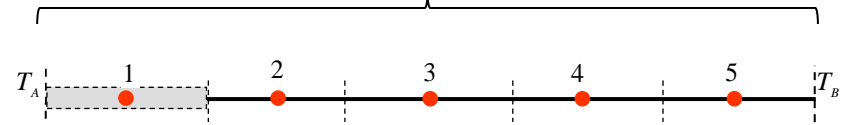
```
(
  (0 0 0) //0
  (1 0 0) //1
  (1 1 0) //2
  (0 1 0) //3
  (0 0 0.1) //4
  (1 0 0.1) //5
  (1 1 0.1) //6
  (0 1 0.1) //7
);
```

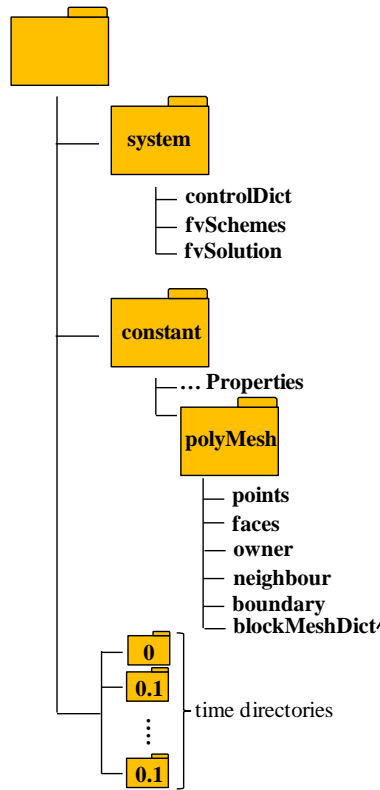


);

blocks

```
(hex (0 1 2 3 4 5 6 7) (5 1 1) simpleGrading (1 1 1));
```





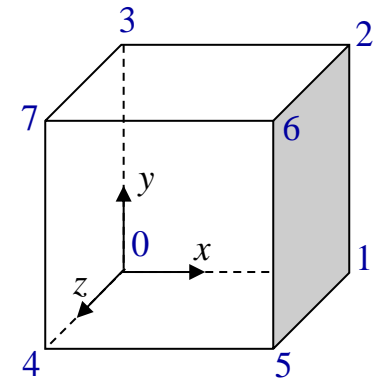
blockMeshDict

convertToMeters 1;

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

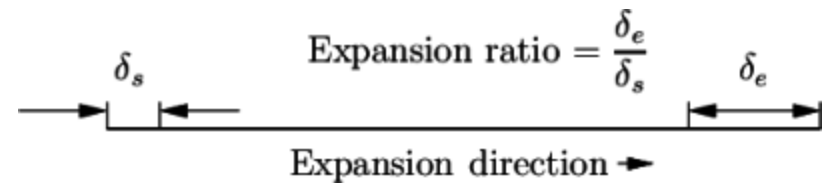
```



);

blocks

(hex (0 1 2 3 4 5 6 7) (5 1 1) simpleGrading (1 1 1));



blockMeshDict

convertToMeters 1;

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  
);
```

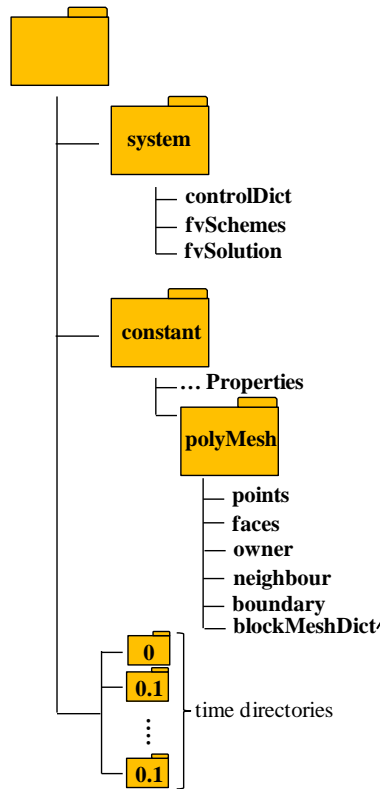
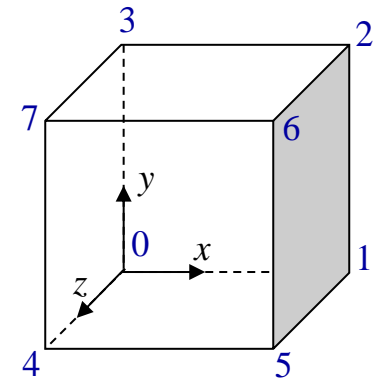
);

blocks

```
(hex (0 1 2 3 4 5 6 7) (5 1 1) simpleGrading (1 1 1));
```

edges

```
(  
  arc 4 5 (0.5 -0.1 0.1)  
  arc 0 1 (0.5 -0.1 0)  
);
```



Boundaries

Keyword selection

arc
simpleSpline
polyLine
polySpline
line

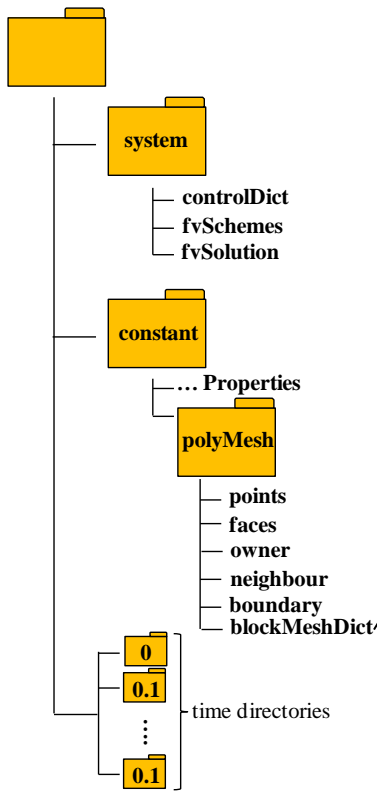
Description

Circular arc
Spline curve
Set of lines
Set of splines
Straight line

Additional entries

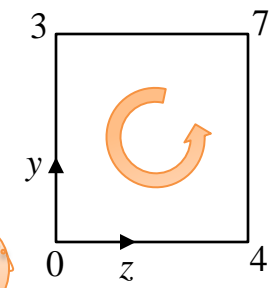
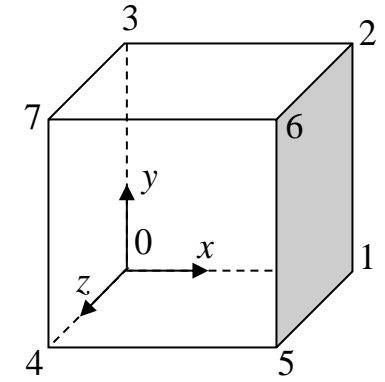
Single interpolation point
List of interpolation points
List of interpolation points
List of interpolation points
—

blockMeshDict

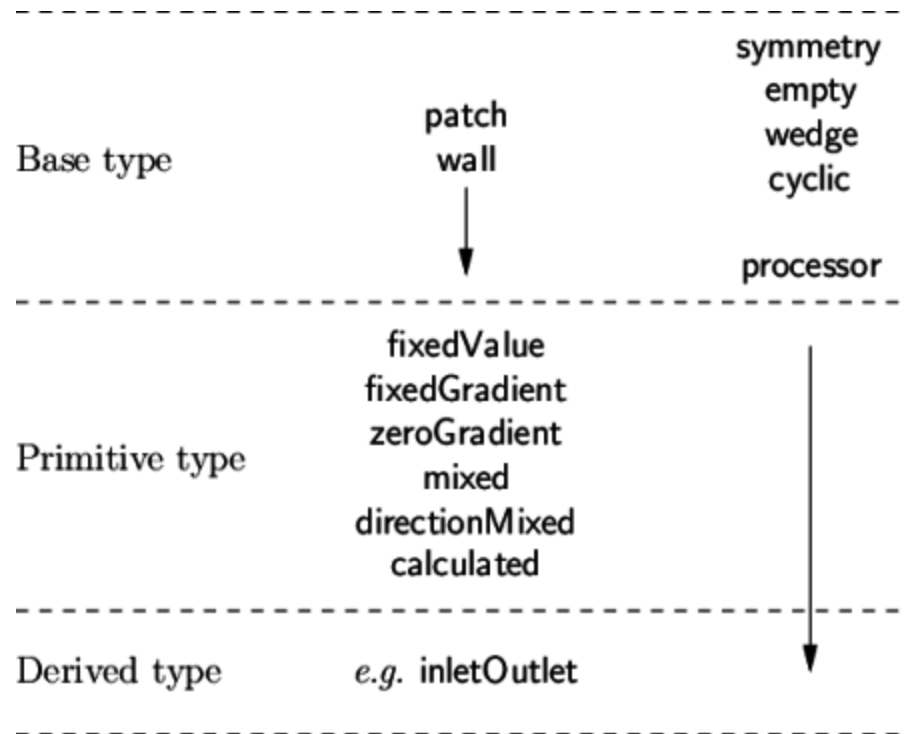


```

boundary
(
  movingWall
  {
    type wall;
    faces
    ( (3 7 6 2) );
  }
  fixedWall
  {
    type wall;
    faces
    ( (0 4 7 3)
      (2 6 5 1)
      (1 5 4 0) );
  }
  frontAndBack
  {
    type empty;
    faces
    ( (0 3 2 1)
      (4 5 6 7) );
  }
);
  
```



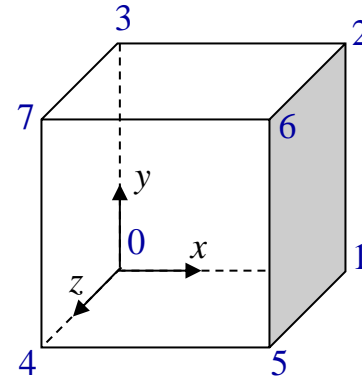
Boundaries

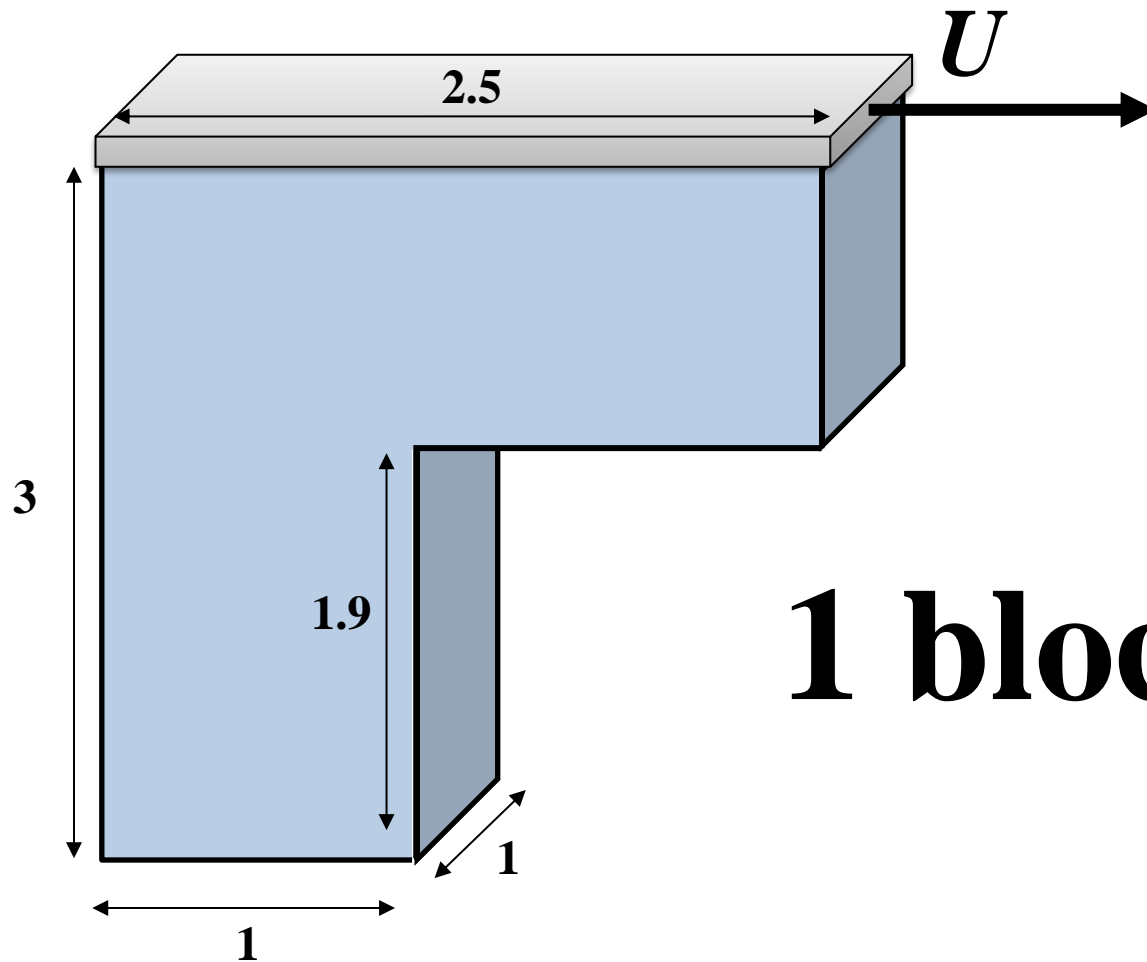


**It is time for you to
do it by yourselves!!**

Change the number of cells, simple grading and edges!

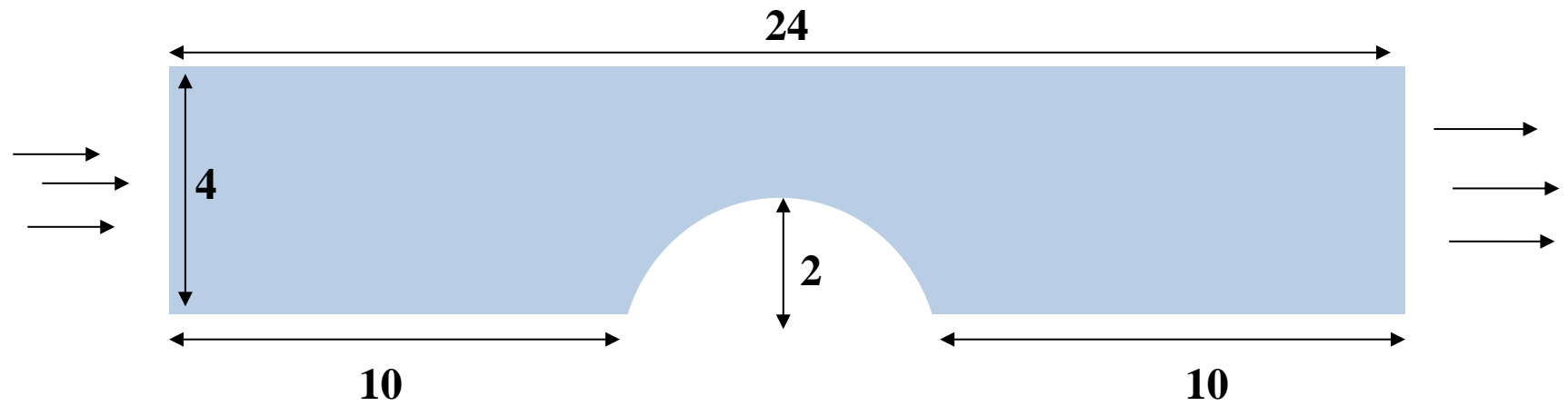
```
convertToMeters 1;  
vertices  
(  
  (0 0 0) //0  
  (1 0 0) //1  
  (1 1 0) //2  
  (0 1 0) //3  
  (0 0 0.1) //4  
  (1 0 0.1) //5  
  (1 1 0.1) //6  
  (0 1 0.1) //7  
);  
blocks  
(hex (0 1 2 3 4 5 6 7) (5 1 1) simpleGrading (1 1 1));  
  
edges  
(  
  
  arc 4 5 (0.5 -0.1 0.1)  
  arc 0 1 (0.5 -0.1 0)  
  
);
```

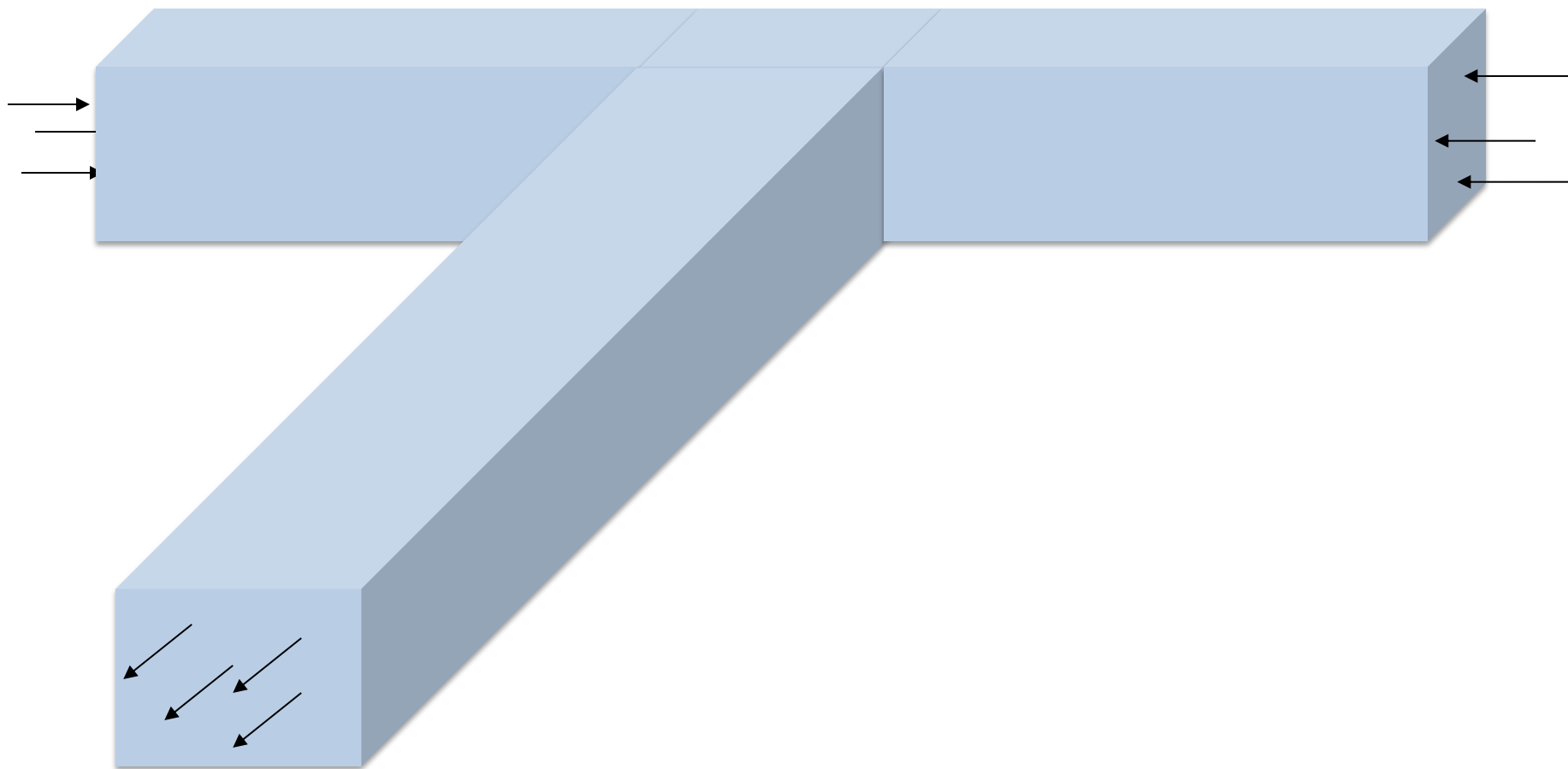




1 block?

2D





The End!