# Virtual Tracer Tests: Coupling CFD and CREng to Simulate WRRFs Unit Processes

## *Getting Started with OpenFOAM*

*nelson.marques@fsdynamics.pt; bruno.santos@fsdynamics.pt*

1st September 2019

Watermatex 2019
DTU
Danmarks Tekniske Universitet
LUND UNIVERSITY
IWA the international water association
1 – 4 September 2019 | Copenhagen - Denmark
WATERMATEX 2019
10th IWA Symposium on Modelling and Integrated Assessment
Photo: Jacob Schjørring and Simone Lau, Copenhagen Media Center

FS DYNAMICS optimises your technology

Watermatex 2019 the international water association

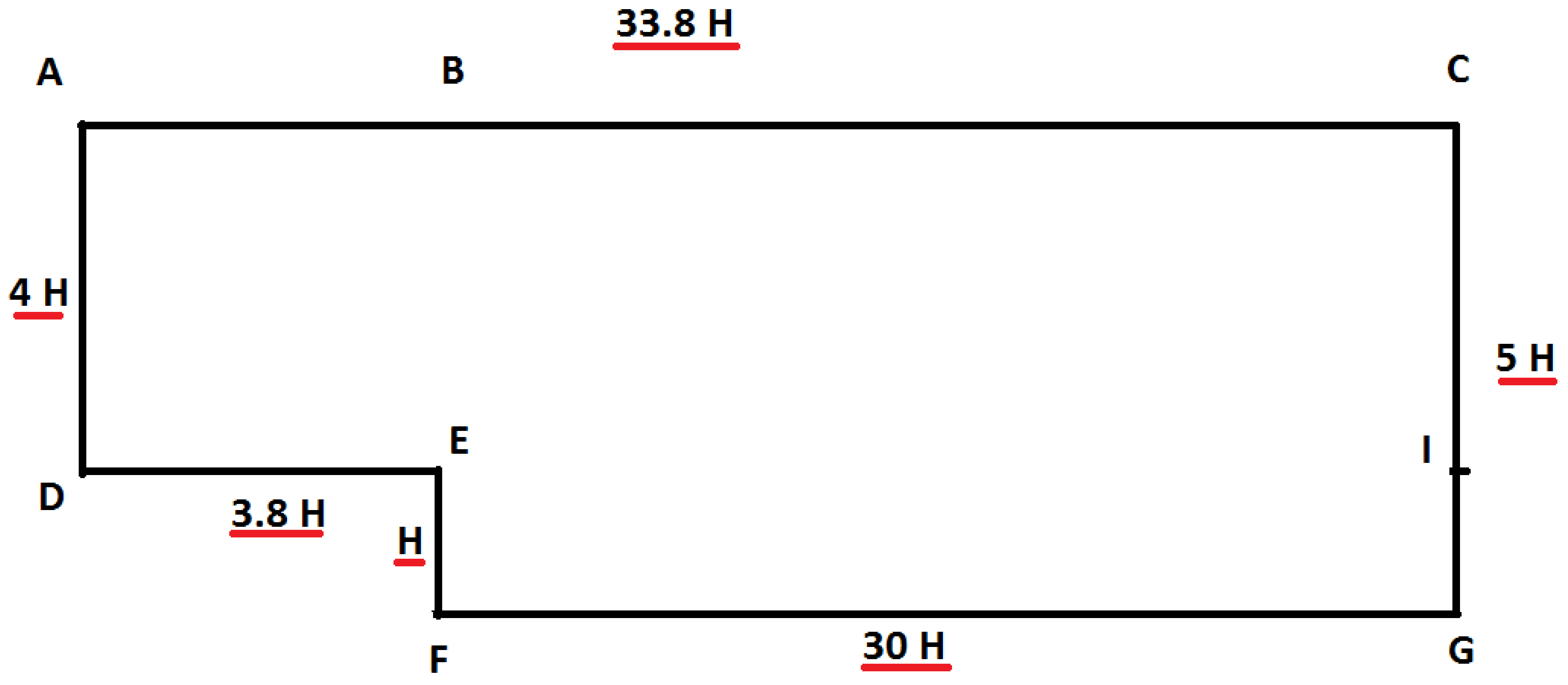
# blockMesh (1/13)

Designed for creating a mesh from various sets of blocks, this mesher is very powerful but can quickly get very complicated to use.

We will showcase its basic use with a simple backward-facing step:

- Length of initial channel section: **3.8 H**
- Height of the initial channel section: **4 H**
- Length of the final channel: **30 H**
  - Total length: **33.8 H**
- Height of the step: **H**
  - Which implies that the height of the final channel section is: **5 H**
- The characteristic height **H** to be used will be **1 meter**.

# blockMesh (2/13)

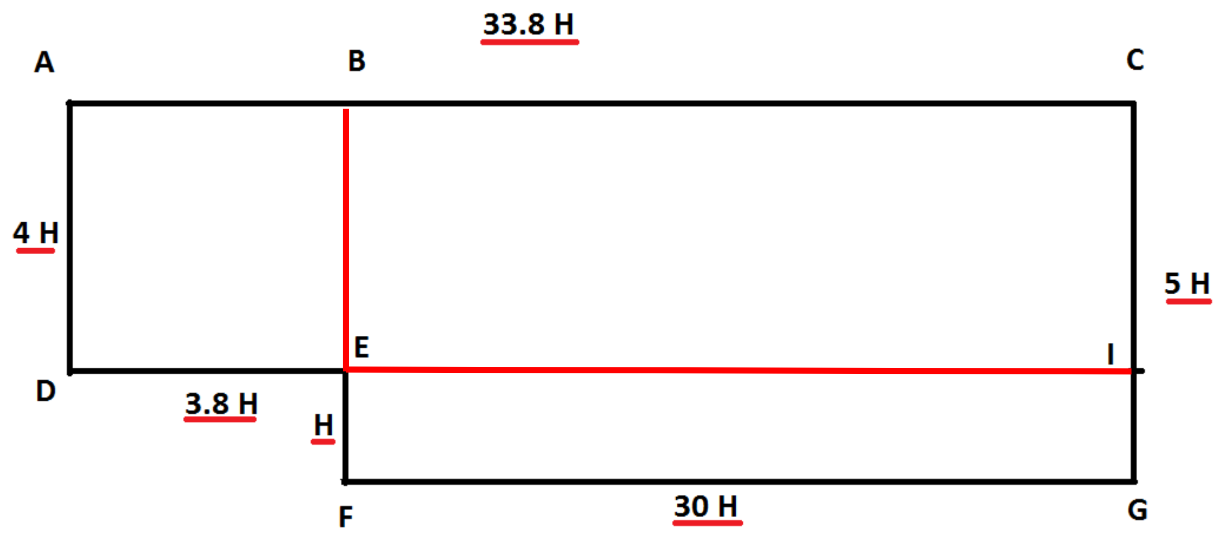


# blockMesh (3/13)

Why all those reference points?

Because we will create 3 blocks, namely:

- ADEB
- EFGI
- BEIC

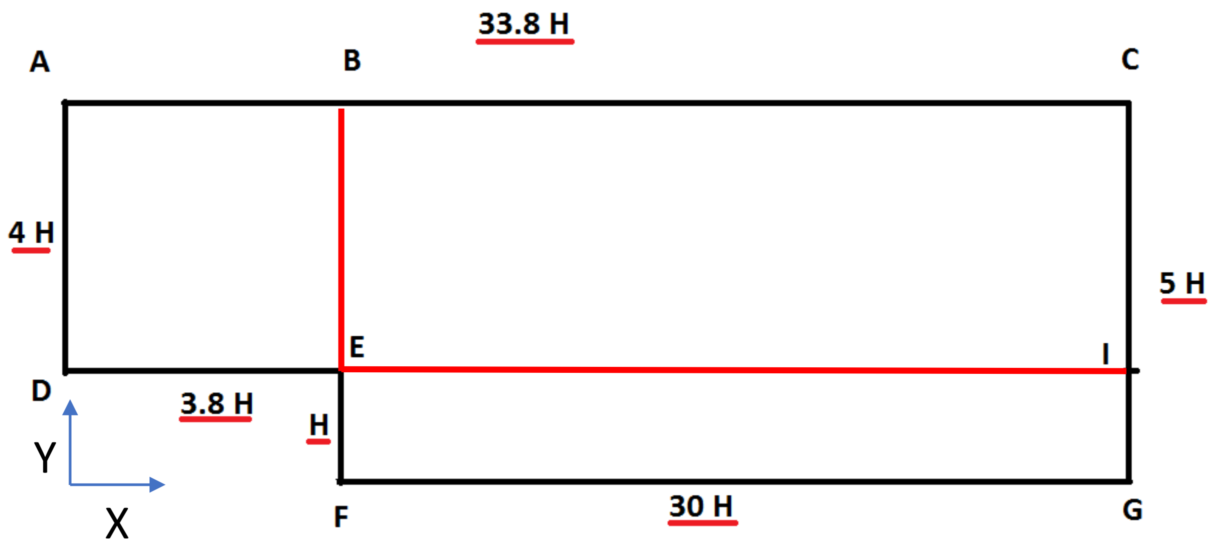


Even though the geometry is defined solely in 2D, OpenFOAM needs the 3rd Dimension just the same, which means that the list of points will be doubled, one for the front, another for the back.

# blockMesh (4/13)

To make it easier to create these blocks, we use a few strategies:

- Define the reference X positions for: AD, BEF and CIG
- Define the reference Y positions for: ABC, DEI and FG
- A list of the indexes associated to each point in the front, as well as a list of the indexes for the points in the back.

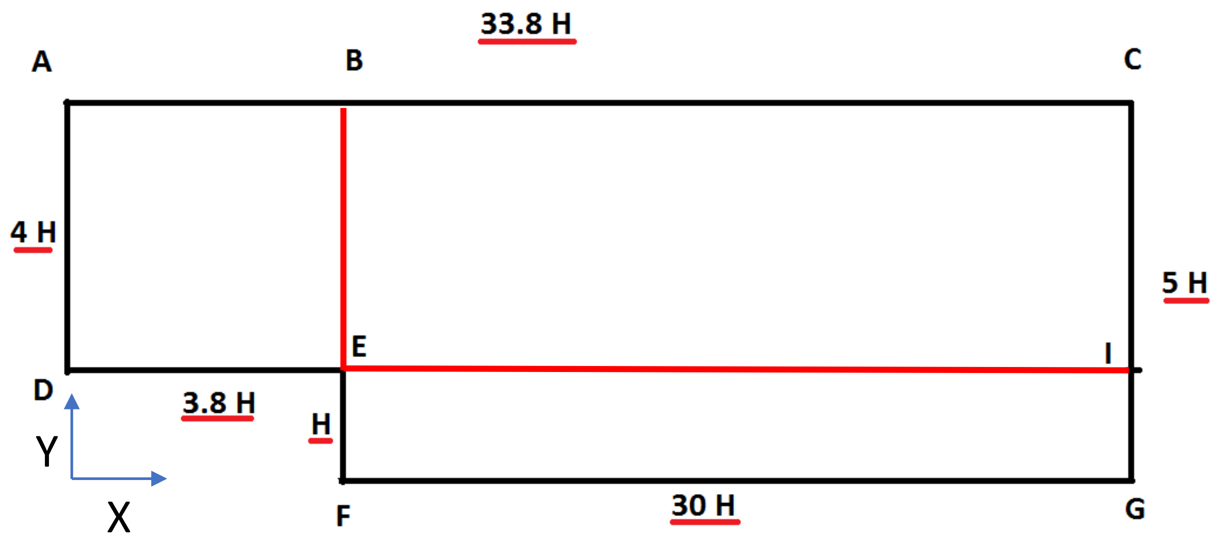


# blockMesh (5/13)

In practice, our “blockMeshDict” will look like this:

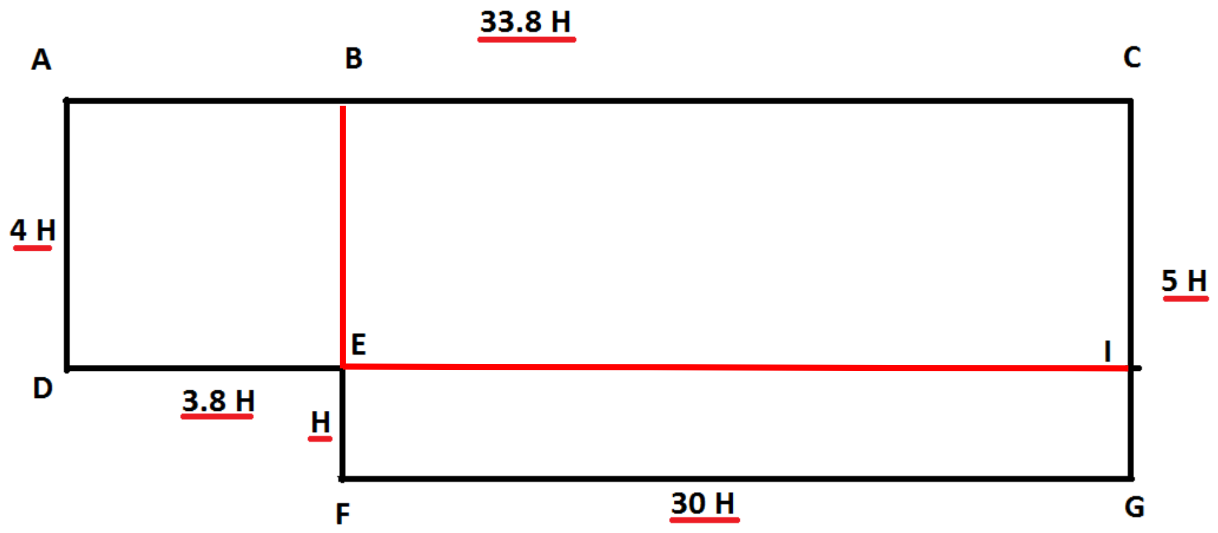
```
convertToMeters 1.0;

// positions ABCDEFGI
ADx       0.0;
BEFx      3.8;
CIGx      33.8;

ABCy    5.0;
DEIy    1.0;
FGy     0.0;

Aa  0;
Ba  1;
Ca  2;
Da  3;
Ea  4;
…
```

```
…
Fa   5;
Ga   6;
Ia   7;

Ab   8;
Bb   9;
Cb   10;
Db   11;
Eb   12;
Fb   13;
Gb   14;
Ib   15;
```



# blockMesh (6/13)

List of vertices (front & back):



```
vertices
(
    //Z=0
    ($ADx    $ABCy   0.0)   //A, 0
    ($BEFx   $ABCy   0.0)   //B, 1
    ($CIGx   $ABCy   0.0)   //C, 2
    ($ADx    $DEIy   0.0)   //D, 3
    ($BEFx   $DEIy   0.0)   //E, 4
    ($BEFx   $FGy    0.0)   //F, 5
    ($CIGx   $FGy    0.0)   //G, 6
    ($CIGx   $DEIy   0.0)   //I, 7

…
```

```
…

  //Z=0.1
     ($ADx    $ABCy   0.1)   //A, 8
     ($BEFx   $ABCy   0.1)   //B, 9
     ($CIGx   $ABCy   0.1)   //C, 10
     ($ADx    $DEIy   0.1)   //D, 11
     ($BEFx   $DEIy   0.1)   //E, 12
     ($BEFx   $FGy    0.1)   //F, 13
     ($CIGx   $FGy    0.1)   //G, 14
     ($CIGx   $DEIy   0.1)   //I, 15
);
```

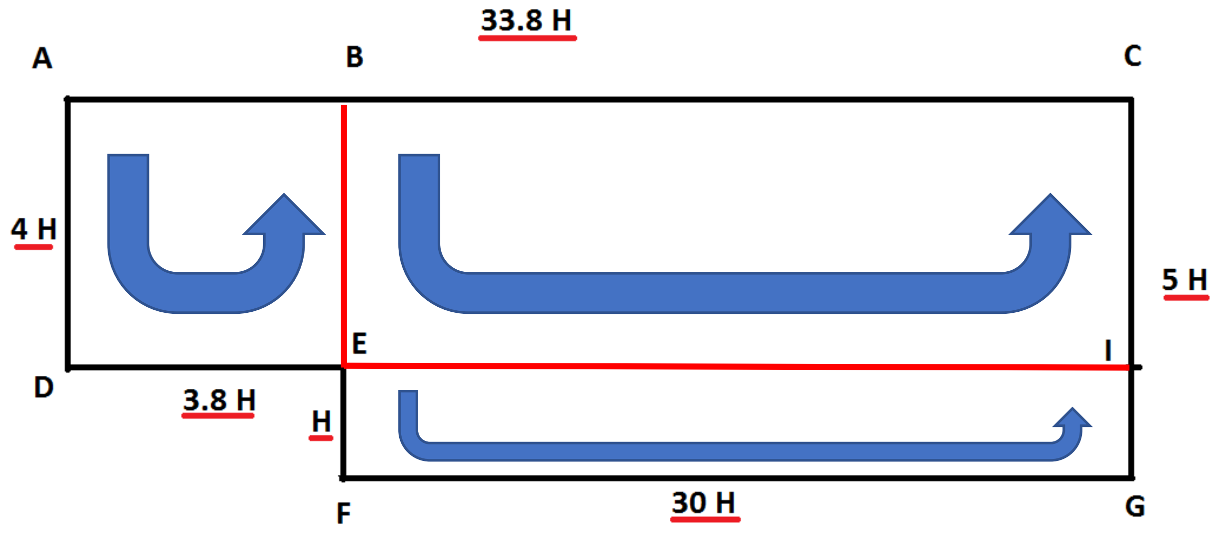
FS DYNAMICS optimises your technology

Watermatex 2019

the international water association

# blockMesh (7/13)

```
blocks
(
    //ADEB
    hex ($Aa $Da $Ea $Ba $Ab $Db $Eb $Bb) (1 1 1) simpleGrading (1 1 1)

    //EFGI
    hex ($Ea $Fa $Ga $Ia $Eb $Fb $Gb $Ib) (1 1 1) simpleGrading (1 1 1)

    //BEIC
    hex ($Ba $Ea $Ia $Ca $Bb $Eb $Ib $Cb) (1 1 1) simpleGrading (1 1 1)
);
```



# blockMesh (8/13)

Geometrical boundaries (1/2):

```
boundary
(
  inlet
  {
    type patch;
    faces
    (
      ($Aa $Da $Db $Ab)
    );
  }

  outlet
  {
    type patch;
    faces
    (
      ($Ca $Ia $Ib $Cb)
      ($Ia $Ga $Gb $Ib)
    );
  }
```

```
  upperWall
  {
    type wall;
    faces
    (
      ($Aa $Ba $Bb $Ab)
      ($Ba $Ca $Cb $Bb)
    );
  }

  lowerWall
  {
    type wall;
    faces
    (
      ($Da $Ea $Eb $Db)
      ($Ea $Fa $Fb $Eb)
      ($Fa $Ga $Gb $Fb)
    );
  }
```

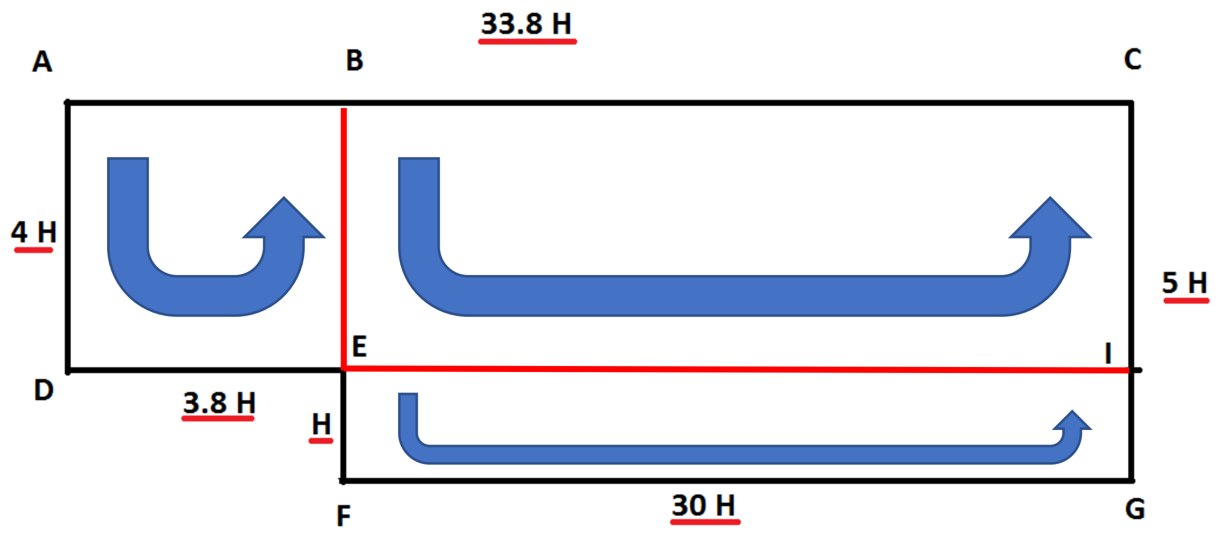
FS DYNAMICS optimises your technology

Watermatex 2019 the international water association

# blockMesh (9/13)

Geometrical boundaries (2/2):

Reminder: the vertices should be defined counter-clockwise and in the same order for the front and back.



```
    frontAndBack
    {
      type empty;
      faces
      (
        ($Aa $Da $Ea $Ba)
        ($Ba $Ea $Ia $Ca)
        ($Ea $Fa $Ga $Ia)

        ($Ab $Db $Eb $Bb)
        ($Bb $Eb $Ib $Cb)
        ($Eb $Fb $Gb $Ib)
      );
    }
);
```

# blockMesh (10/13)

Last but not least, the “edges” list and “mergePatchPairs”:

```
edges
(
);

mergePatchPairs
(
);
```

Where:

- **edges**: for providing a list of edge modifiers, e.g.:

```
arc 0 5 (0.469846 0.17101 -0.5)
```

- **mergePatchPairs**: for merging patches, e.g. if we had two geometrical boundaries that we wanted to *stitch together*.

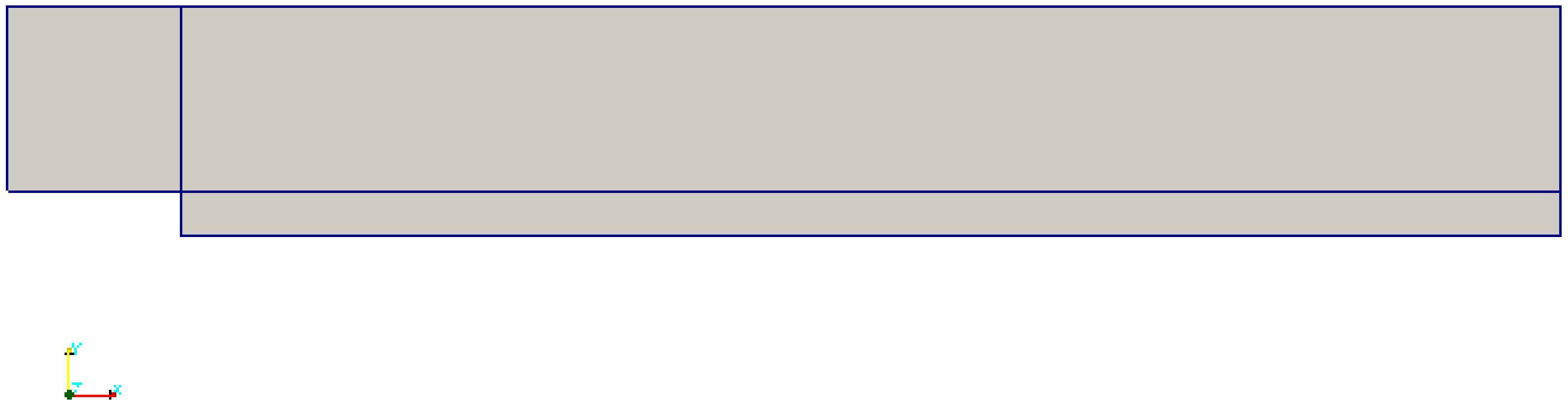
FS DYNAMICS optimises your technology

Watermatex 2019 IWA the international water association

# blockMesh (11/13)

Workflow:

1. We use a tutorial case as a basis, for example "`basic/potentialFoam/pitzDaily`".
2. Modify the file "`system/blockMeshDict`".
3. Run **blockMesh**.
4. If all goes well, run **paraFoam**.
5. What we will see in ParaView is something like this:

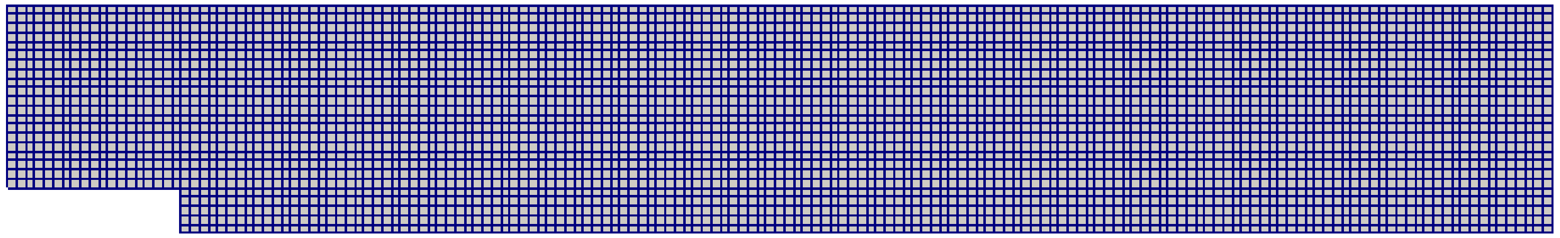


# blockMesh (12/13)

Edit the “blockMeshDict” and changing the block list to this:

```
blocks
(
  //ADEB
  hex ($Aa $Da $Ea $Ba $Ab $Db $Eb $Bb) (20 19 1) simpleGrading (1 1 1)

  //EFGI
  hex ($Ea $Fa $Ga $Ia $Eb $Fb $Gb $Ib) (5 150 1) simpleGrading (1 1 1)

  //BEIC
  hex ($Ba $Ea $Ia $Ca $Bb $Eb $Ib $Cb) (20 150 1) simpleGrading (1 1 1)
);
```

Will result in this:



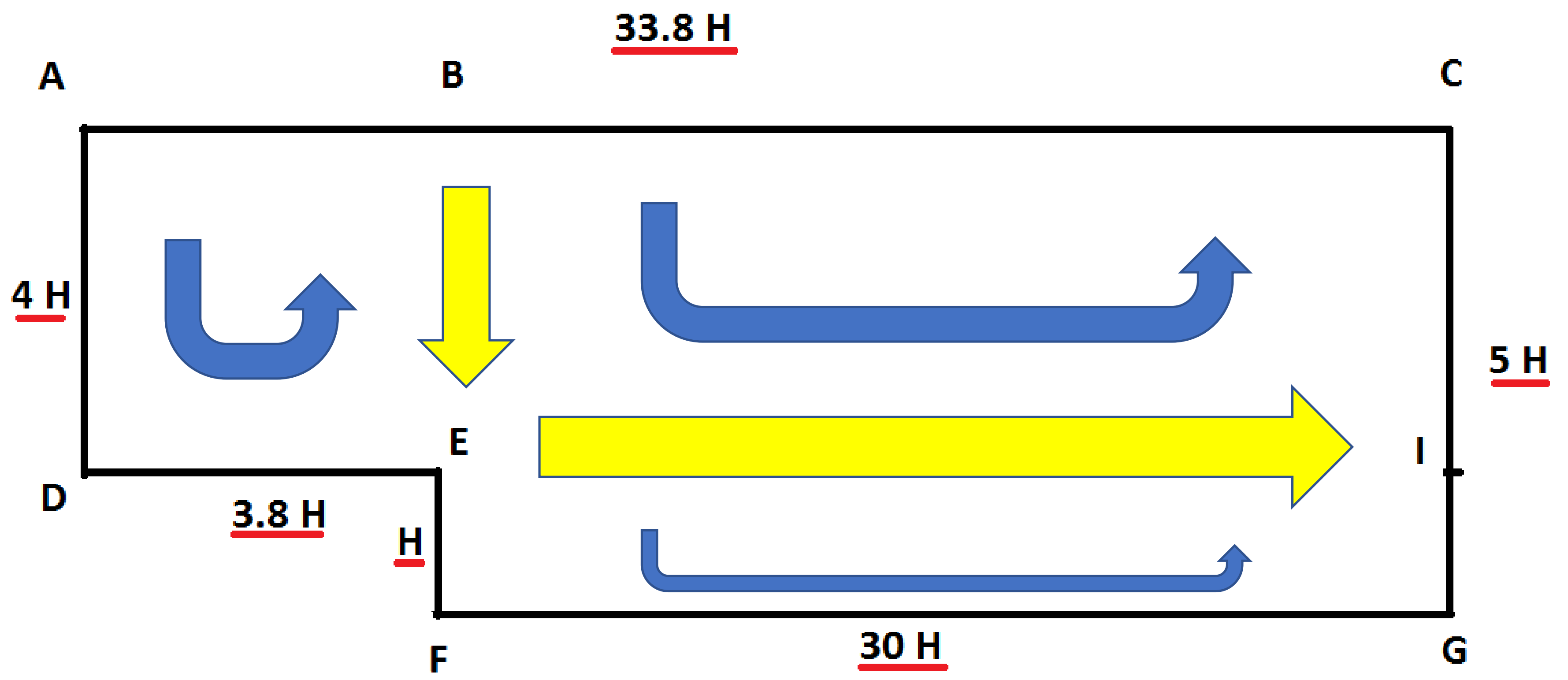
# blockMesh (13/13)

The grading over each direction depends in the order of the vertices:

```
hex ($Ba $Ea $Ia $Ca $Bb $Eb $Ib $Cb) (20 150 1)…
```

B → E

E → I


33.8 H
A
B
C
4 H
5 H
E
I
D
3.8 H
H
F
30 H
G