



# My First Tutorial in OpenFOAM

Federico Piscaglia \*

Dept. of Aerospace Science and Technology, Politecnico di Milano, via La Masa 34, I-20156 Milano (ITALY)

**Abstract.** The present is a short lab handout with the main commands and steps used to set up and run the OpenFOAM ‘cavity’ tutorial.

## 1 Learning outcome

The software used is the open-source CFD software OpenFOAM®-7 by OpenCFD®. In this section you will learn how to:

- Generate a mesh using blockMesh and check the result
- Run a simple solver (icoFoam) saving the output for later reference
- Post-process velocity fields with ParaView
- Change setup and run the case

## 2 Baseline case

### 2.1 Setup

1. Open a terminal and enter the ‘run’ folder:

```
user@host:~$ run
```

2. Copy the entire tutorials folder into your run

```
user@host:run$ cp -r $FOAM_TUTORIALS .
```

3. enter the ‘cavity’ folder:

```
user@host:run$ cd tutorials/incompressible/icoFoam/cavity/cavity
```

4. Generate the mesh (and write screen output on file):

```
user@host:cavity$ blockMesh > log.blockMesh
```

5. Check the mesh quality

```
user@host:cavity$ checkMesh > log.checkMesh
```

6. See output of ‘checkMesh’ (‘q’ to return to shell):

```
user@host:cavity$ less log.checkMesh
```

---

\*Tel. (+39) 02 2399 8620, E-mail: [federico.piscaglia@polimi.it](mailto:federico.piscaglia@polimi.it)

## 2.2 Run the solver

1. Compute solution

```
user@host:cavity$ icoFoam > log.icoFoam
```

## 2.3 Postprocessing

1. Open ParaView:

- if you have compiled ParaView: `user@host:cavity$ paraFoam`
- if you have not: `user@host:cavity$ paraFoam -builtin`
- if you wish to use precompiled ParaView for Windows:
  - i. create an empty file: `paraFoam -builtin -touch`
  - ii. open ParaView for Win
  - iii. go to the tutorials folder and select 'cavity/cavity.foam'

2. Generate a vector glyph:

- i select main source, either 'cavity.foam' or 'cavity.OpenFOAM'
- ii Filters → Glyph
- iii In the 'vectors' drop-down list, select  $U$  (cell-based)
- iv In the 'scale mode' list, select 'vector'
- v Click 'Apply'
- vi Select wireframe representation

3. Generate streamlines:

- i select main source, either 'cavity.foam' or 'cavity.OpenFOAM'
- ii Filters → Cell centers
- iii Filters → Mask Points
- iv Choose 50 points; activate the 'advanced options' (gear icon) and select 'random sampling'
- v Filters → Alphabetical → Stream tracer with custom source
- vi Select as 'input' the main source and as 'seed source' 'MaskPoints1'
- vii Click 'Apply'

4. Extract data along a line

- i Select main source
- ii Filters → Plot Over Line
- iii Align the line along the desired direction (e.g. Y-axis), by manually dragging points, enter endpoints coordinates or using predefined directions
- iv Click on 'Apply'
- v You can display/undisplay fields using checkboxes

## 3 Hands-on

It is a good practice to keep a copy of the original case when changing the settings:

1. Go up one level: `user@host:cavity$ cd ../`
2. Copy the folder: `user@host:cavity$ cp -r cavity cavityFiner`
3. Enter the new case folder: `user@host:cavity$ cd cavityFiner`
4. Clear old results: `user@host:cavityFiner$ rm -r 0.?`

### 3.1 Refine the mesh

1. Open 'system/blockMeshDict' with a text editor
2. Increase the number of cells along X- and Y- direction (e.g. 20  $\Rightarrow$  100)
3. Save and close
4. Re-generate the mesh: `user@host:cavityFiner$ blockMesh`
5. Check the new mesh: `user@host:cavityFiner$ checkMesh > log.checkMesh`
6. Run the solver: `user@host:cavityFiner$ icoFoam > log.icoFoam`

The solver has crashed. Why?

**Hint:** read the log of icoFoam and check the maximum Courant number. It must remain below 1. Co is defined as:

$$Co = \frac{U \Delta t}{\Delta x}$$

### 3.2 Make the case 3D

**Assignment** Modify the case to make it 3D and solve it.

**Hint**

- Modify the 'blockMeshDict' by putting more than 1 cells in Z- direction
- Set 'frontAndBack' patches to 'wall' instead of 'empty' in blockMeshDict, '0/p' and '0/U' files.