

OPENFOAM®'A GİRİŞ ÇALIŞTAYI

Open△FOAM®

Hacer Duzman

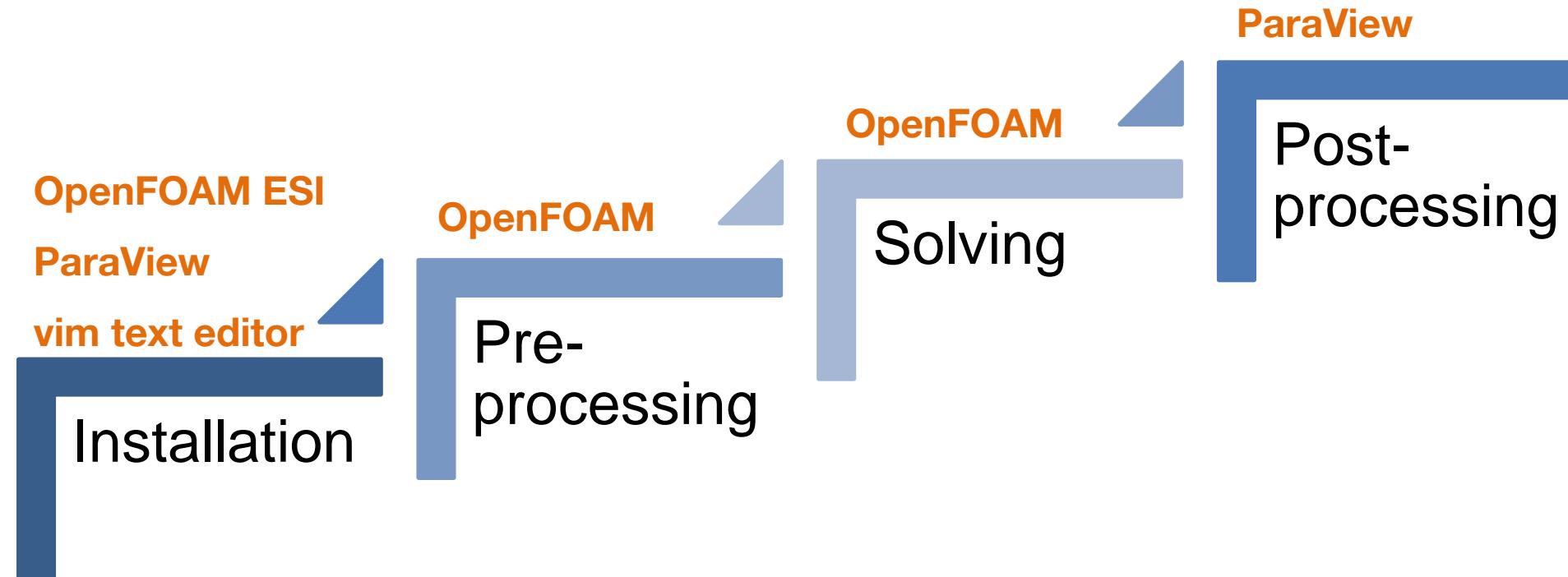
 ParaView

ITU Computational Biomechanics Research Group
<https://valve.be.itu.edu.tr/>



Workshop Supervisor: Prof. Dr. M. Serdar Çelebi

December 1, 2023



You can download 'Pre-Processing' and 'Simulations' files.

<https://github.com/mscelebi/OpenFOAMWorkshop/>

- OpenFOAM®, ParaView®, and vim installation
- Introduction to OpenFOAM®
- Examples of OpenFOAM® in various fields
- CFD simulation workflow and OpenFOAM® case directory
- Mesh generation process
- Mesh quality
- Meshes in OpenFOAM®
- blockMesh
- Examples:
 1. Cavity Example (2D - blockMesh)
 2. Cavity Grade Example (2D - blockMesh)
 3. Elbow Example (2D - blockMesh)
 4. Rectangular Prism Example (3D - blockMesh)
 5. Pipe Example (3D – blockMesh O-Grid)
- snappyHexMesh
- Examples:
 1. Tank Example (3D – snappyHexMesh - internal)
 2. Flange Example (3D – snappyHexMesh - internal)
 3. Wing Example (2D – snappyHexMesh - external)
 4. Motorbike Example (3D – snappyHexMesh - external)
 5. Coronary Artery Example (3D - .msh)
- Useful Documents

System Specifications

- Ubuntu 20.04
- OpenFOAM ESI 2206
- ParaView v5.10
- vim text editor

OpenFOAM® & vim text editor installation

Terminal:

```
$ sudo snap install curl (if not installed)
```

```
$ curl https://dl.openfoam.com/add-debian-repo.sh | sudo bash
```

```
$ wget -q -O - https://dl.openfoam.com/add-debian-repo.sh | sudo bash
```

```
$ sudo apt-get update
```

```
$ sudo apt-get install openfoam2206-default
```

- To the last line of the bashrc file:

```
alias of2206='source /usr/lib/openfoam/openfoam2206/etc/bashrc'
```

- vim text editor:

```
$ sudo apt install vim (if not installed)
```

OpenFOAM® ESI Website:

<https://develop.openfoam.com/Development/openfoam/-/wikis/precompiled/debian>

The screenshot shows a terminal window titled 'duzman@duzman:~'. The user runs several commands to set up the system:

- curl https://dl.openfoam.com/add-debian-repo.sh | sudo bash
- sudo snap install curl # version 8.1.2, or
sudo apt install curl # version 7.68.0-1ubuntu2.20
- See 'snap info curl' for additional versions.
- sudo snap install curl
- curl 8.1.2 from Wouter van Bommel (woutervb) installed
- curl https://dl.openfoam.com/add-debian-repo.sh | sudo bash
- Detected distribution code-name: focal
- Added /etc/apt/sources.list.d/openfoam.list
- Importing openfoam gpg key... done
- Added /etc/apt/trusted.gpg.d/openfoam.gpg
- Running apt-get update... done
- The repository is setup! You can now install packages.
- sudo apt-get update
- Hit:1 http://tr.archive.ubuntu.com/ubuntu focal InRelease
- Hit:2 http://tr.archive.ubuntu.com/ubuntu focal-updates InRelease
- Hit:3 http://tr.archive.ubuntu.com/ubuntu focal-backports InRelease
- Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
- Get:5 https://sourceforge.net/projects/openfoam/files/repos/deb focal InRelease [3.624 B]
- Get:6 https://sourceforge.net/projects/openfoam/files/repos/deb focal/main amd64 Packages [13,0 kB]
- Get:7 https://sourceforge.net/projects/openfoam/files/repos/deb focal/main all Packages [5.598 B]
- Fetched 22,2 kB in 9s (2.519 B/s)
- Reading package lists... Done

ParaView® installation

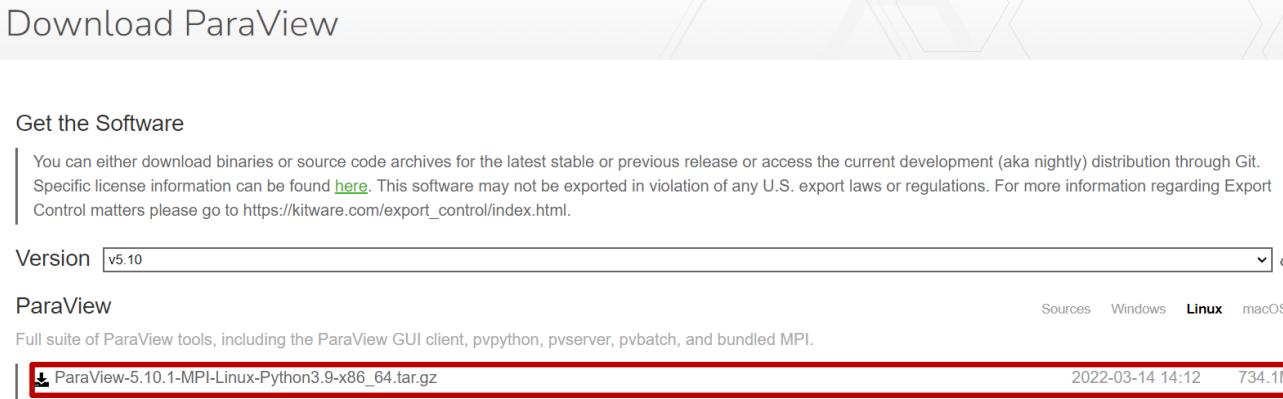
- Download ParaView-5.10.1-MPI-Linux-Python3.9-x86_64 from website
- \$ tar -xvf ParaView-5.10.1-MPI-Linux-Python3.9-x86_64.tar.gz
- You can move it to another folder.
- To the last line of the bashrc file:

```
alias paraview-5.10='~/Desktop/hacer/ParaView-5.10.1-MPI-Linux-Python3.9-x86_64/bin/paraview'
```

Download ParaView

Get the Software

You can either download binaries or source code archives for the latest stable or previous release or access the current development (aka nightly) distribution through Git. Specific license information can be found [here](#). This software may not be exported in violation of any U.S. export laws or regulations. For more information regarding Export Control matters please go to https://kitware.com/export_control/index.html.



Version v5.10

ParaView

Sources Windows Linux macOS

Full suite of ParaView tools, including the ParaView GUI client, pvython, pvserver, pbatch, and bundled MPI.

ParaView-5.10.1-MPI-Linux-Python3.9-x86_64.tar.gz 2022-03-14 14:12 734.1M

ParaView-5.10.0-MPI-Linux-Python3.9-x86_64.tar.gz 2021-12-22 20:50 734.0M

Open .bashrc Save

```
119
120 alias of2206='source /usr/lib/openfoam/openfoam2206/etc/bashrc'
121 alias paraview-5.10='~/ParaView-5.10.1-MPI-Linux-Python3.9-x86_64/bin/paraview'
122
```

sh Tab Width: 8 Ln 118, Col 1 INS

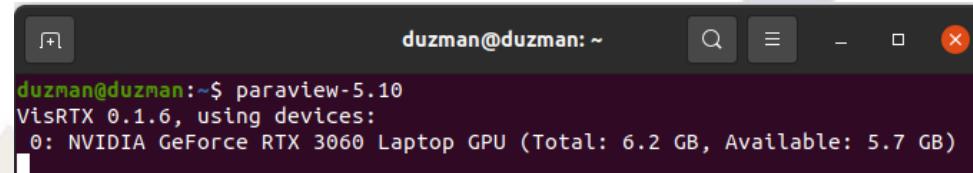
ParaView® Website:
<https://www.paraview.org/download/>
İSTANBUL TEKNİK ÜNİVERSİTESİ
Hacer Duzman

- Utilise aliases to source the OpenFOAM® and ParaView® versions
- To use OpenFOAM® version 2206, type terminal of2206



```
duzman@duzman:~$ of2206
duzman@duzman:~$ simpleFoam
/*-----*\
| ====== |
| \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox
| \ \ / O peration | Version: 2206
| \ \ / A nd | Website: www.openfoam.com
| \ \ / M anipulation |
\*-----*/
Build : _8993af73-20221106 OPENFOAM=2206 patch=221104 version=2206
Arch : "LSB;label=32;scalar=64"
Exec : simpleFoam
```

- To use ParaView® version 5.10, type terminal paraview-5.10



```
duzman@duzman:~$ paraview-5.10
VisRTX 0.1.6, using devices:
  0: NVIDIA GeForce RTX 3060 Laptop GPU (Total: 6.2 GB, Available: 5.7 GB)
```

Introduction to OpenFOAM®

- OpenFOAM = **O**pen **F**ield **O**peration and **M**anipulation
- First development at Imperial College London in 1984.
- Open-source software developed in C++ (object oriented programming).
- Finite Volume Method (FVM) based solver.
- Easily customisable.
- Various extensions such as OpenFOAM ESI, OpenFOAM Foundation and Foam-extend (WIKKI) are available.
- It can be used on parallel computers. One of the top 5 most used software on HPC (High-performance computing).
- It has a wide user base (industry, academia and research laboratories) worldwide.
- Community support:
 - - A discussion forum (www.cfd-online.com/Forums/openfoam/)
 - - A community-driven wiki (www.openfoamwiki.net)
 - - An annual Workshop (www.openfoamworkshop.org)

Examples of OpenFOAM® in various fields

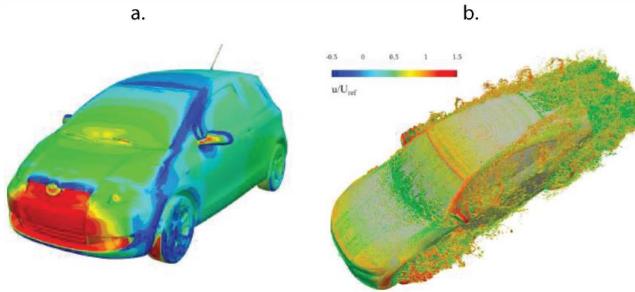


Figure 1. Automobile aerodynamics simulation, a. pressure distribution on the automobile, b. Q-criterion visualisation around the automobile (Kaya et al. 2022)

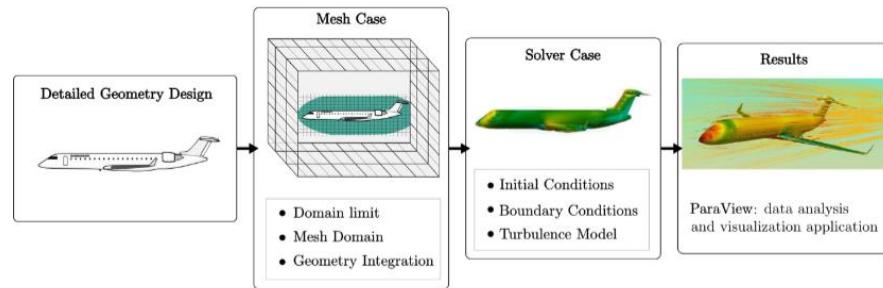


Figure 2. Bombardier Regional Jet CRJ700 (Segui et al. 2021)

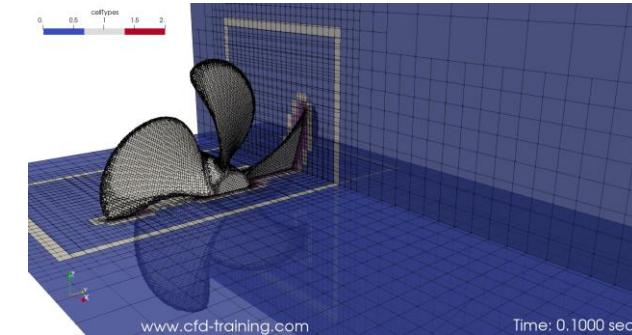


Figure 3. Open-water propeller KP 505 modeling (cfd-training.com)

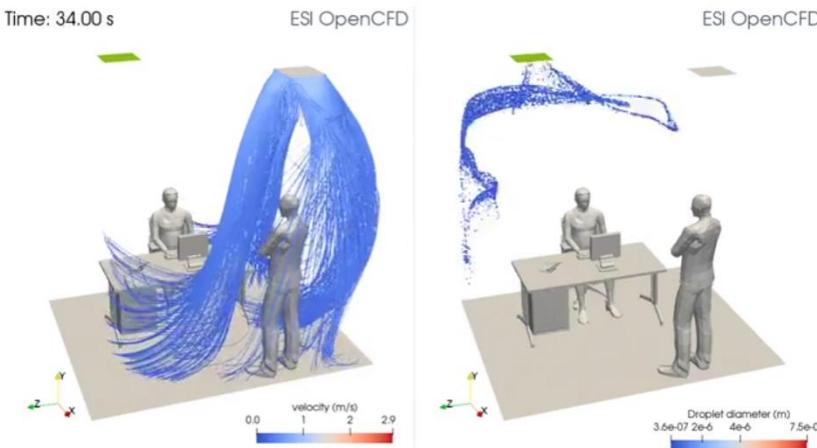


Figure 4. Ventilation in office environments (openfoam.com)

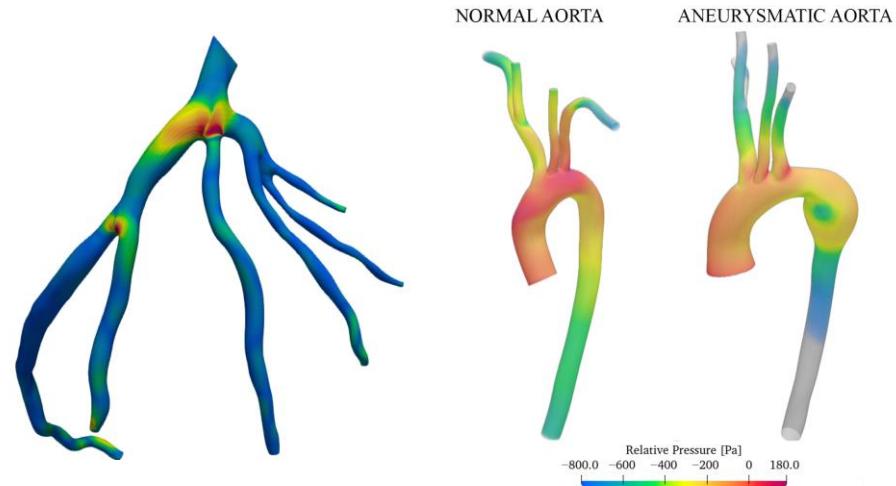


Figure 5. Left coronary artery blood flow simulation

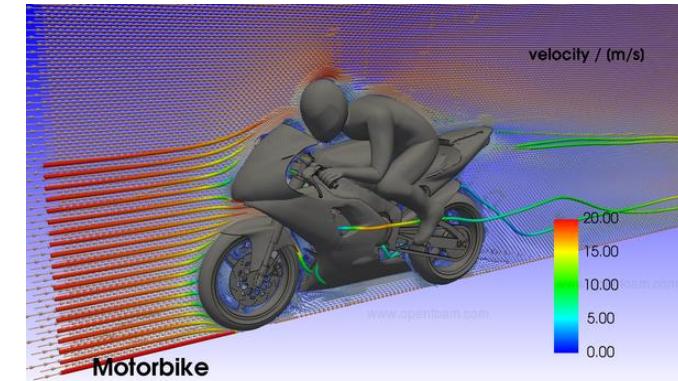


Figure 6. Aortic blood flow simulation (Duronio et al. 2023)

CFD simulation workflow and OpenFOAM® case directory

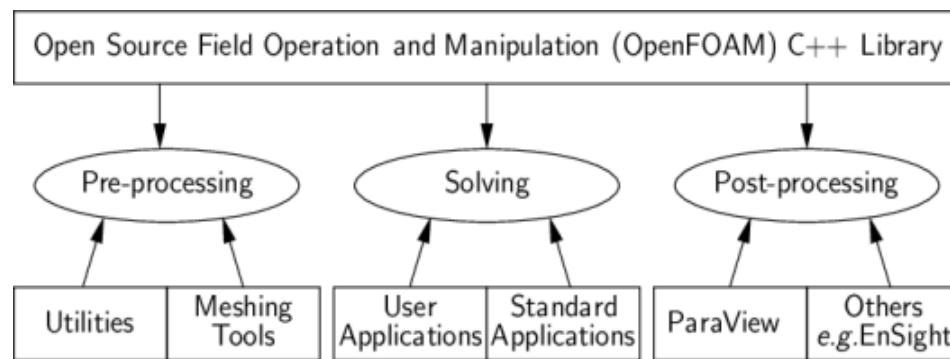
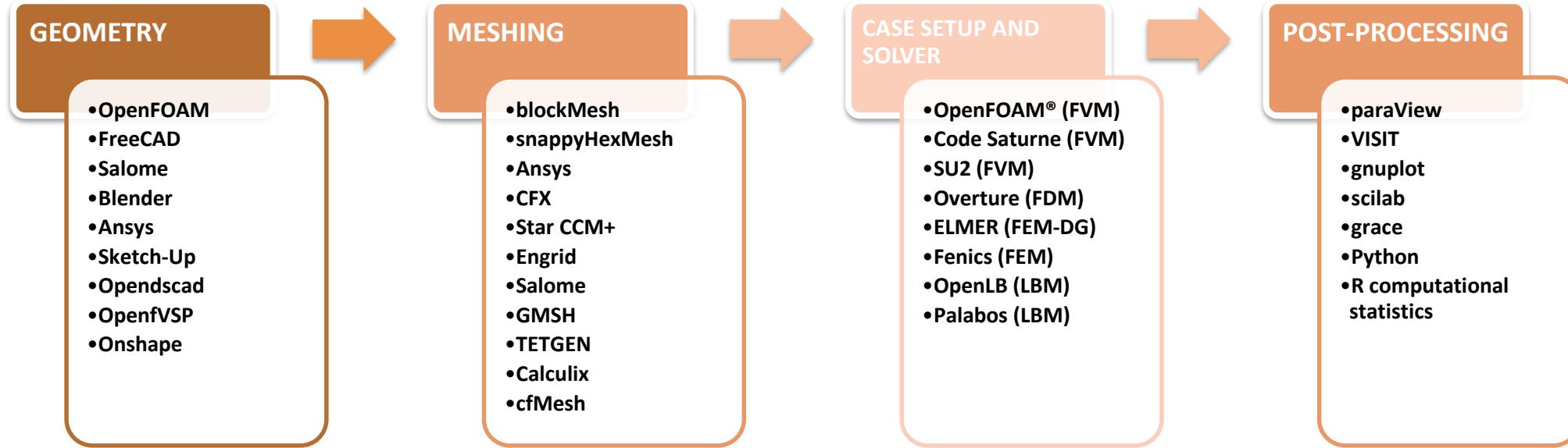


Figure 8. Overview of OpenFOAM structure. (openfoam.com)

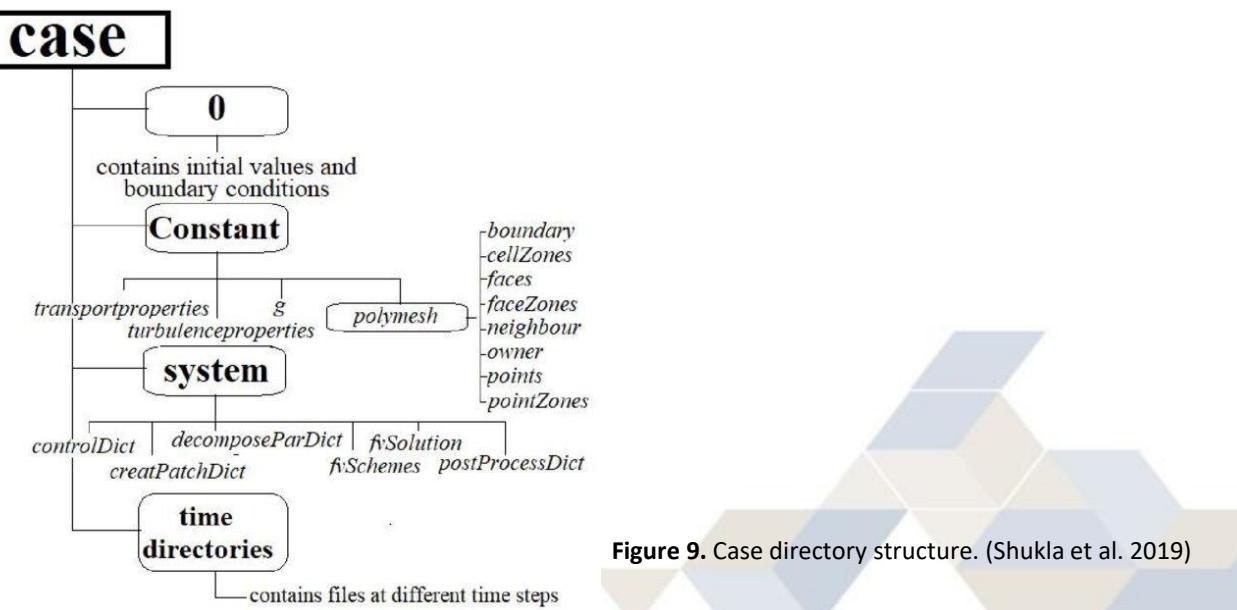


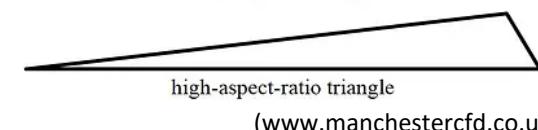
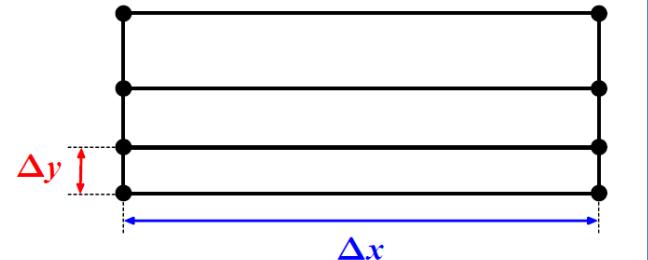
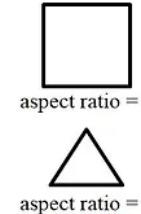
Figure 9. Case directory structure. (Shukla et al. 2019)

- **Geometry generation:** The refinement of the mesh and consequently, the quality of the solution are significantly influenced by the geometry.
- **Mesh generation:**
 - External flow: physical domain and generating a mesh around the object.
 - Internal flow: mesh the internal volume of the geometry. Inflation layers to enhance the resolution of the boundary layer.
 - Mesh quality upon completing the meshing process.
- **Definition of boundary surfaces:** We identify specific physical surfaces to apply boundary conditions. The boundary surfaces, also known as patches, are established during the meshing process. In OpenFOAM®, you can access this information in the **boundary** dictionary file situated within the **constant/polyMesh** directory. This dictionary file is automatically generated during the meshing time.

Aspect ratio

Ratio of longest edge length to shortest edge length.

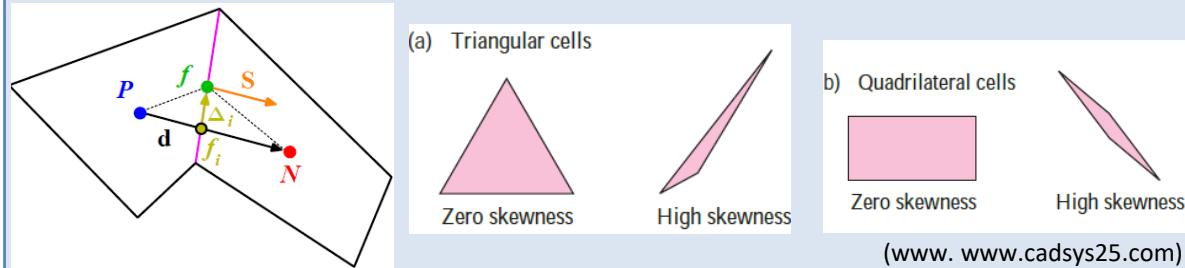
Equal to 1 (ideal)



(www.manchestercfd.co.uk/)

Skewness

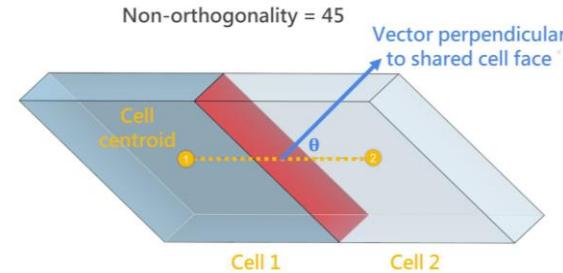
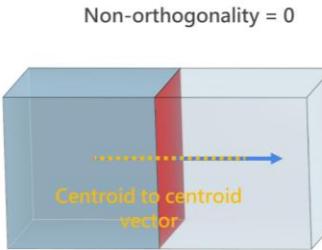
Skewness refers to the divergence of vector d , which links the two cells P and N , from the center f of the face. The deviation vector is depicted by Δ_i , with f_i denoting the point where vector d intersects the face f .



Non-orthogonality

The angle between the vector linking two neighboring cell centers and the normal of the face shared by these cells.

The range of non-orthogonality is between 0 (ideal) and 90 (worst).

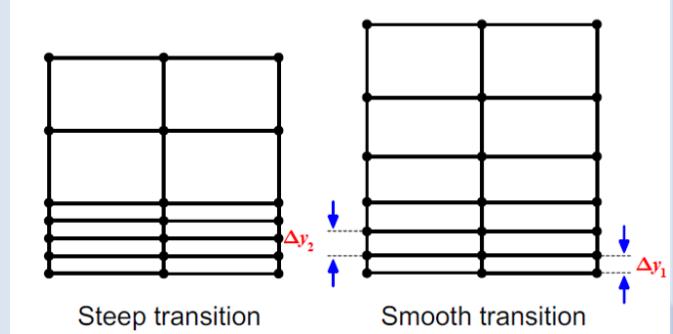


(www.simscale.com/)

Smoothness

Smoothness, alternatively referred to as expansion rate, growth factor, or uniformity, describes the change in size between adjacent cells.

$$\frac{\Delta y_2}{\Delta y_1} \leq 1.2$$



- The file primitiveMeshCheck.C, found in the directory
 - `$WM_PROJECT_DIR/src/OpenFOAM/meshes/primitiveMesh/primitiveMeshCheck/`
 - contains the quality metrics utilized in OpenFOAM®. The maximum or minimum thresholds are defined within this file.

```
// * * * * * * * * * * * * * * * * Static Data Members * * * * * * * * * * * * * * * * //  
  
Foam::scalar Foam::primitiveMesh::closedThreshold_ = 1.0e-6;  
Foam::scalar Foam::primitiveMesh::aspectThreshold_ = 1000;  
Foam::scalar Foam::primitiveMesh::nonOrthThreshold_ = 70;      // deg  
Foam::scalar Foam::primitiveMesh::skewThreshold_ = 4;  
Foam::scalar Foam::primitiveMesh::planarCosAngle_ = 1.0e-6;
```

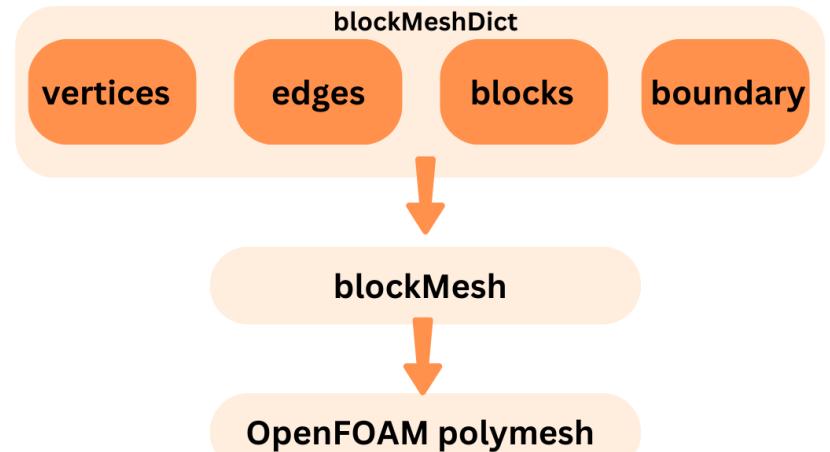
- While it's possible to proceed with mesh quality issues like high skewness, high aspect ratio, and high non-orthogonality, it's crucial to note that these issues can impact solution accuracy, potentially leading to unexpected results.
 - In some cases, they could even cause the solver to fail or produce unstable outcomes.
 - OpenFOAM® provides the utility ***checkMesh*** to verify the mesh's integrity. ***checkMesh*** examines:
 - Total count of cells for each type.
 - Validates topology (boundary condition definitions).
 - Evaluate geometry and mesh quality (cell volumes, skewness, orthogonality, aspect ratio, etc.).
 - In case of errors detected by ***checkMesh***, it provides a notification specifying the failed checks and the particular issues identified.

- OpenFOAM uses Cartesian coordinates.
- Since OpenFOAM only supports 3D geometry, geometry is always defined in 3D.
- 1D or 2D cases, it is essential to have at least one unit of volume in each axis direction.
- The following meshing applications are available for OpenFOAM®.
 - ❖ blockMesh
 - ❖ snappyHexMesh
 - ❖ foamyHexMesh
 - ❖ Third party tools and importing it. Supports many popular mesh formats. Such as CFX, Fluent, Gmsh, ideas, Netgen, plot3d, STAR-CCM+, VTK. OpenFOAM provides the following mesh conversions:
- Find source code of these applications in the directory:
 - `$ cd $FOAM_UTILITIES/mesh/`
 - Usage of mesh conservation for fluent (.msh format) and salome (.unv format):
 - `$ fluentMeshToFoam elbow.msh`
 - `$ ideasUnvToFoam elbow.unv`

MESH CONVERSIONS

ansysToFoam
cfx4ToFoam
datToFoam
fluent3DMeshToFoam
fluentMeshToFoam
foamMeshToFluent
foamToStarMesh
foamToSurface
gambitToFoam
gmshToFoam
ideasUnvToFoam
kivaToFoam
mshToFoam
netgenNeutralToFoam
plot3dToFoam
sammToFoam
star3ToFoam
star4ToFoam
tetgenToFoam
vtkUnstructuredToFoam

- **blockMesh** is a generator of a structured hexahedral mesh.
- Suitable for simple geometries that could be described by a few blocks, but difficult to apply to cases with a large number of blocks.
- The background mesh utilized in conjunction with **snappyHexMesh** comprises a singular rectangular block.
- A dictionary file called **blockMeshDict**, located in the **system** directory, is used to generate the mesh.
- Boundaries may be of different types:
 - patch (generic type)
 - wall (for solid wall condition, useful for turbulence)
 - empty (to specify that the simulation will be 2D or 1D)
 - cyclic (for cyclic simulations)
 - symmetryPlane (for symmetry plane)
 - wedge (for axi-symmetric simulations)
 - processor (for parallel computation, automatically defined during the decomposition domain process)



1- Cavity Example (2D - blockMesh)

- Copying the tutorial from the following folder to your working directory and mesh process:

OpenFOAM 2206 version: \$ of2206

Copy tutorial: \$ cp -r \$FOAM_TUTORIALS/incompressible/icoFoam/cavity/cavity .

Go to directory cavity: \$ cd cavity

Generate the grid mesh: \$ blockMesh

Check the mesh quality: \$ checkMesh

Open .foam file: \$ vim cavity.foam

View the mesh: \$ paraview-5.10 cavity.foam

You can find polymesh file in constant file.

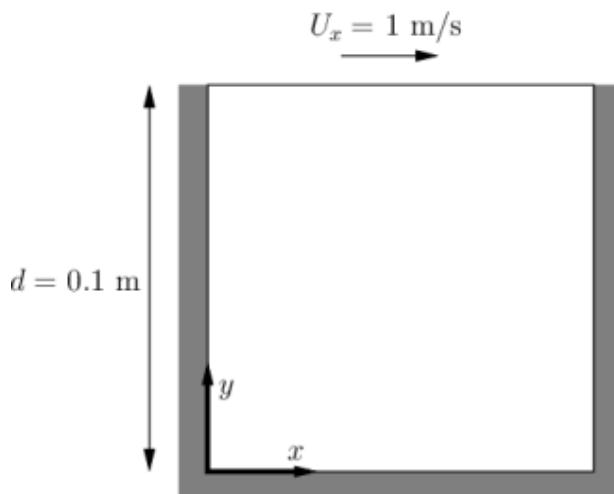


Figure 10. Geometry of the lid driven cavity.

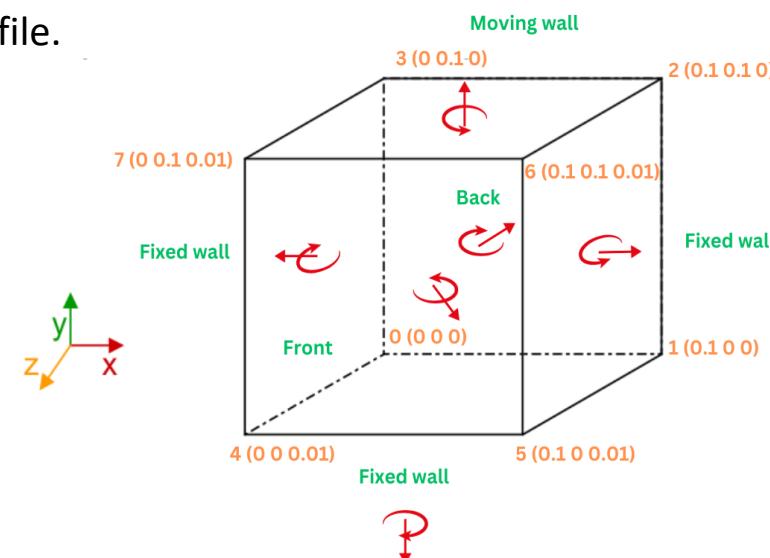


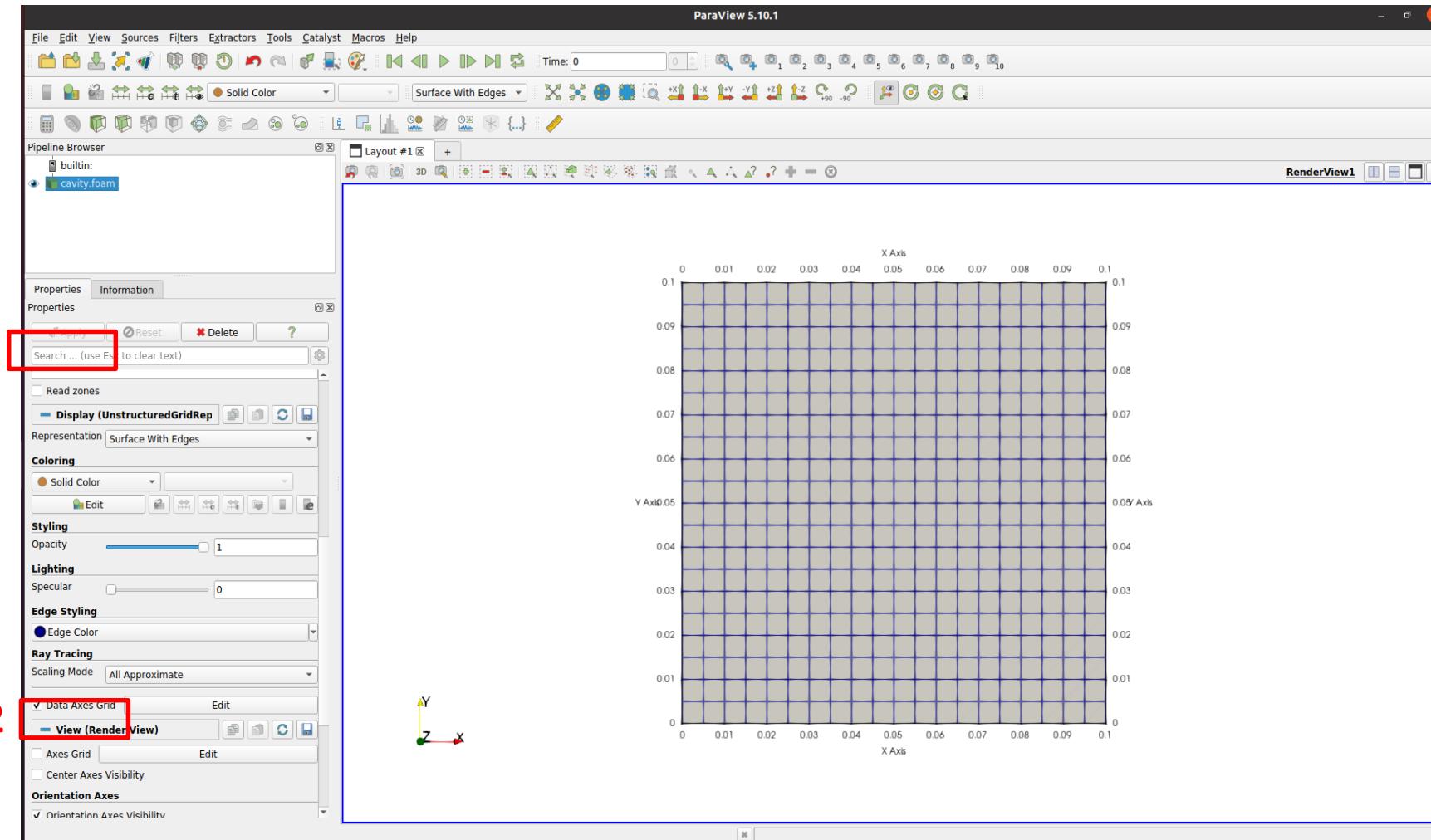
Figure 11. Block structure of the mesh for the cavity.

```
17 scale  0.1; //Scale coordinates
18
19 vertices
20 (
21   (0 0 0)  //vertex 0
22   (1 0 0)  //vertex 1
23   (1 1 0)  //vertex 2
24   (0 1 0)  //vertex 3
25   (0 0 0.1) //vertex 4
26   (1 0 0.1) //vertex 5
27   (1 1 0.1) //vertex 6
28   (0 1 0.1) //vertex 7
29 );
30
31 blocks
32 ( //Definition of the hexahedral block. Pay attention to the numbering.
33   hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1) //Uniform gradient
34 );
35   //20 cells in x-direction, 20 cells in y-direction,
36   //and only 1 cell in z-direction bcs the simulation will be 2D.
37
38 edges //This is simple geometry, edges can be empty.
39 (
40 );
41 boundary //Definition of boundary of the domain to apply the BC.
42 (
43   movingWall
44   {
45     type wall;
46     faces
47     (
48       (3 7 6 2)
49     );
50   }
51   fixedWalls
52   {
53     type wall;
54     faces
55     (
56       (0 4 7 3)
57       (2 6 5 1)
58       (1 5 4 0)
59     );
60   }
61   frontAndBack
62   {
63     type empty; // Faces normal to Oz are "empty" to specify that the simulation is 2D.
64     faces
65     (
66       (0 3 2 1)
67       (4 5 6 7)
68     );
69 }
```

blockMeshDict

1- Cavity Example (2D - blockMesh)

ParaView



2- Cavity Grade Example (2D - blockMesh)

```
$ of2206
$ cp -r $FOAM_TUTORIALS/incompressible/icoFoam/cavity/cavityGrade .
$ cd cavityGrade
$ blockMesh
$ checkMesh
$ vim cavityGrade.foam
$ paraview-5.10 cavityGrade.foam
```

The mesh needs 4 blocks as different mesh grading is needed on the left, right, top, and bottom of the domain.

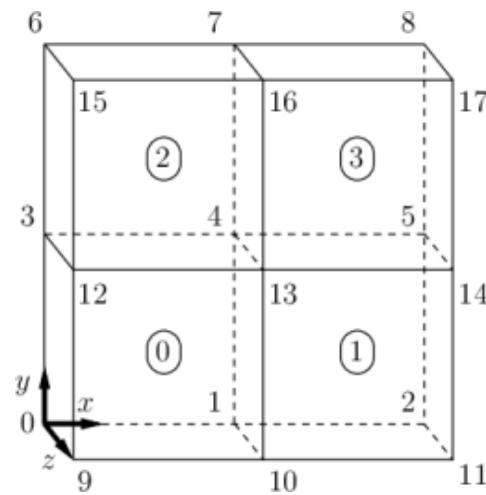


Figure 12. Block structure of the graded mesh for the cavity (block numbers encircled).

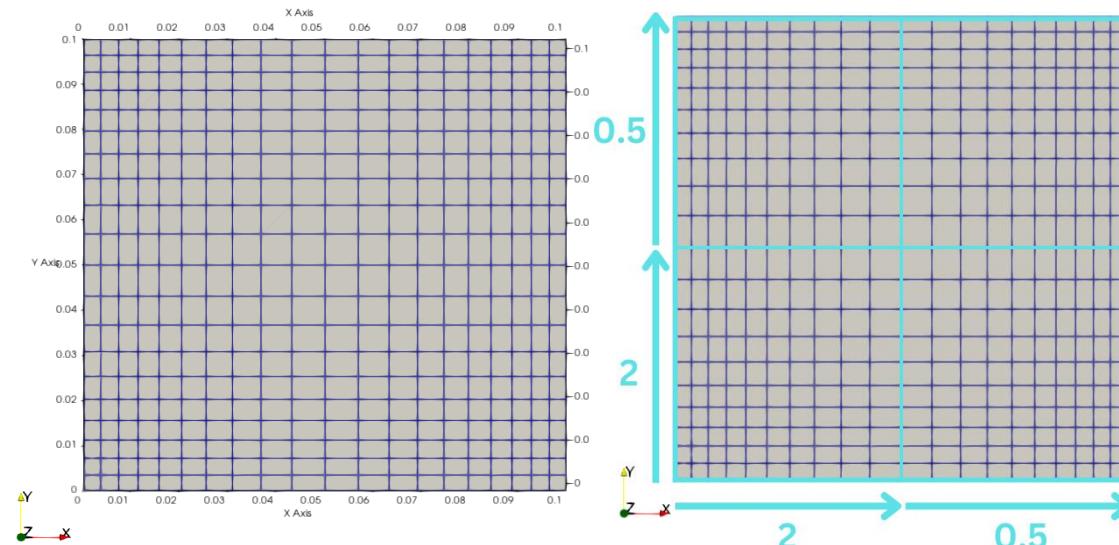


Figure 13. blockMesh

```
17 scale  0.1;
18
19 vertices
20 (
21   (0 0 0)           //vertex 0
22   (0.5 0 0)         //vertex 1
23   (1 0 0)           //vertex 2
24   (0 0.5 0)         //vertex 3
25   (0.5 0.5 0)       //vertex 4
26   (1 0.5 0)         //vertex 5
27   (0 1 0)           //vertex 6
28   (0.5 1 0)         //vertex 7
29   (1 1 0)           //vertex 8
30   (0 0 0.1)         //vertex 9
31   (0.5 0 0.1)       //vertex 10
32   (1 0 0.1)         //vertex 11
33   (0 0.5 0.1)       //vertex 12
34   (0.5 0.5 0.1)     //vertex 13
35   (1 0.5 0.1)       //vertex 14
36   (0 1 0.1)         //vertex 15
37   (0.5 1 0.1)       //vertex 16
38   (1 1 0.1)         //vertex 17
39 );
40
1 blocks
2 (
3   hex (0 1 4 3 9 10 13 12) (10 10 1) simpleGrading (2 2 1)          //Block 0
4   hex (1 2 5 4 10 11 14 13) (10 10 1) simpleGrading (0.5 2 1)        //Block 1
5   hex (3 4 7 6 12 13 16 15) (10 10 1) simpleGrading (2 0.5 1)        //Block 2
6   hex (4 5 8 7 13 14 17 16) (10 10 1) simpleGrading (0.5 0.5 1)        //Block 3
7 );
```

checkMesh

```
Checking geometry...
Overall domain bounding box (0 0 0) (0.1 0.1 0.01)
Mesh has 2 geometric (non-empty/wedge) directions (1 1 0)
Mesh has 2 solution (non-empty) directions (1 1 0)
All edges aligned with or perpendicular to non-empty directions.
Boundary openness (1.41172e-18 -1.41172e-18 1.23243e-16) OK.
Max cell openness = 1.31741e-16 OK.
Max aspect ratio = 2 OK.
Minimum face area = 1.19059e-05. Maximum face area = 6.90099e-05. Face area magnitudes OK.
Min volume = 1.19059e-07. Max volume = 4.76237e-07. Total volume = 0.0001. Cell volumes OK.
Mesh non-orthogonality Max: 0 average: 0
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 2.55087e-14 OK.
Coupled point location match (average 0) OK.

Mesh OK.

End
```

3- Elbow Example (2D - blockMesh)

Tri-mesh

```
$ of2206
$ cp -r $FOAM_TUTORIALS/incompressible/icoFoam/elbow .
$ cd elbow
$ cp $FOAM_TUTORIALS/resources/geometry/elbow.msh .
$ fluentMeshToFoam elbow.msh
$ checkMesh
$ vim elbow_tri.foam
$ paraview-5.10 elbow_tri.foam
```

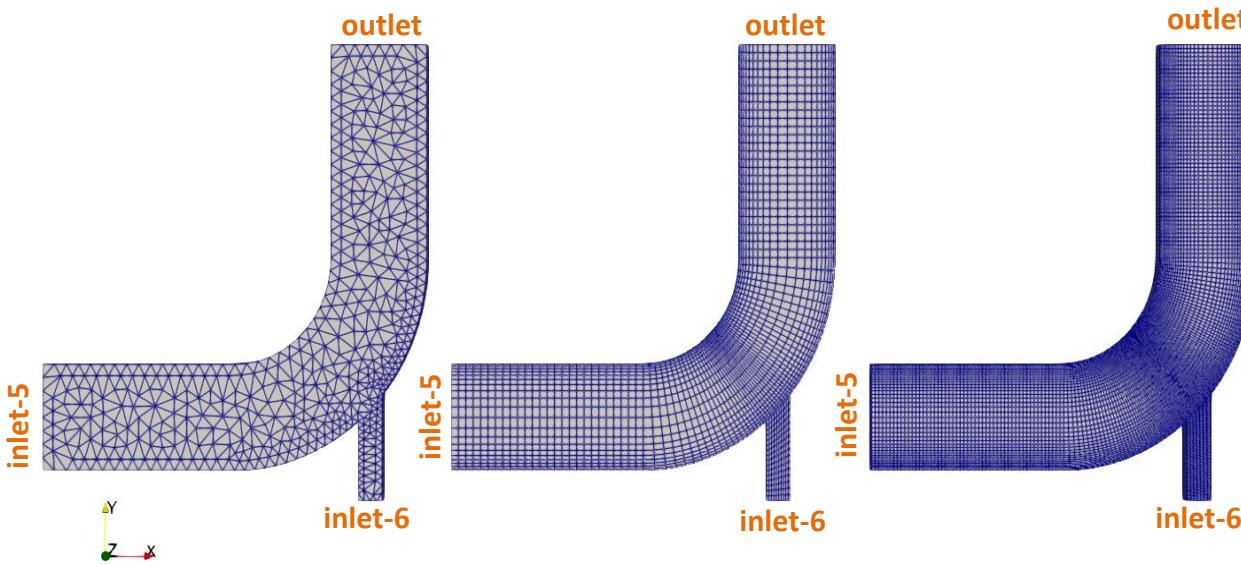


Figure 14. Tri and quad meshes of elbow.

Quad-mesh

```
$ of2206
$ fluentMeshToFoam elbow_quad.msh
$ checkMesh
$ vim elbow_quad.foam
$ paraview-5.10 elbow_quad.foam
```

Refined Quad-mesh

```
$ of2206
$ fluentMeshToFoam elbow_quad.msh
$ refineMesh -overwrite
$ checkMesh
$ vim elbow_quad_refined.foam
$ paraview-5.10 elbow_quad_refined.foam
```

4- Rectangular Prism Example (3D - blockMesh)

\$ of2206
 \$ blockMesh
 \$ checkMesh
 \$ vim rectangular.foam
 \$ paraview-5.10 rectangular.foam

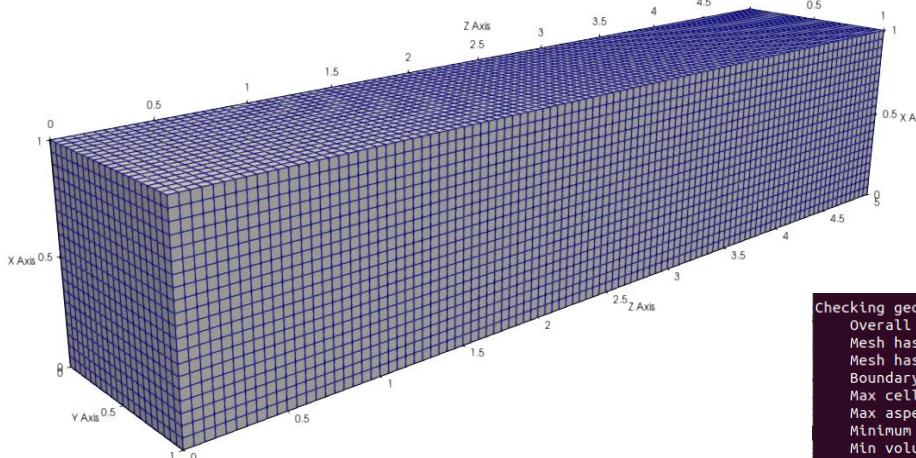


Figure 16. View of the mesh.

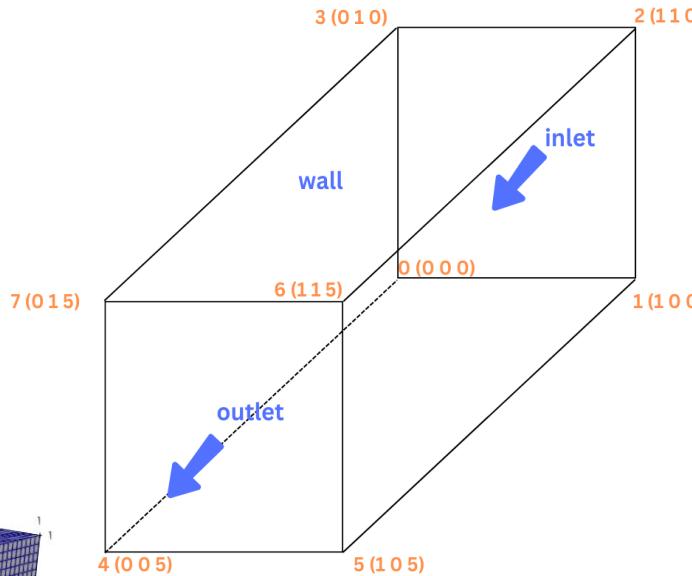


Figure 15. Block structure of rectangular prism .

checkMesh

```
Checking geometry...
Overall domain bounding box (0 0 0) (1 1 5)
Mesh has 3 geometric (non-empty/wedge) directions (1 1 1)
Mesh has 3 solution (non-empty) directions (1 1 1)
Boundary openness (7.49085e-18 -7.45143e-18 -1.63833e-16) OK.
Max cell openness = 1.73472e-16 OK.
Max aspect ratio = 1 OK.
Minimum face area = 0.0025. Maximum face area = 0.0025. Face area magnitudes OK.
Min volume = 0.000125. Max volume = 0.000125. Total volume = 5. Cell volumes OK.
Mesh non-orthogonality Max: 0 average: 0
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 1.06443e-13 OK.
Coupled point location match (average 0) OK.

Mesh OK.
End
```

blockMeshDict

```
17 scale 1; //Scale coordinates
18
19 vertices
20 (
21   (0 0 0) //vertex 0
22   (1 0 0) //vertex 1
23   (1 1 0) //vertex 2
24   (0 1 0) //vertex 3
25   (0 0 5) //vertex 4
26   (1 0 5) //vertex 5
27   (1 1 5) //vertex 6
28   (0 1 5) //vertex 7
29 );
30
31 blocks
32 ( //Definition of the hexahedral block. Pay attention to the numbering.
33   hex (0 1 2 3 4 5 6 7) (20 20 100) simpleGrading (1 1 1) //Uniform gradient
34 );
35   //20 cells in x-direction, 20 cells in y-direction, and 100 cells in z-direction.
36
37 edges //This is simple geometry, edges can be empty.
38 (
39 );
40 boundary //Definition of boundary of the domain to apply the BC.
41 (
42   inlet
43   {
44     type patch;
45     faces
46     (
47       (0 3 2 1)
48     );
49   }
50   outlet
51   {
52     type patch;
53     faces
54     (
55       (4 5 6 7)
56     );
57   }
58   wall
59   {
60     type wall;
61     faces
62     (
63       (3 7 6 2)
64       (0 4 7 3)
65       (2 6 5 1)
66       (1 5 4 0)
67     );
68   }
69 );
```

5- Pipe Example (3D – blockMesh O-Grid)

```
$ of2206  
$ blockMesh  
$ checkMesh  
$ vim pipe.foam  
$ paraview-5.10 pipe.foam
```

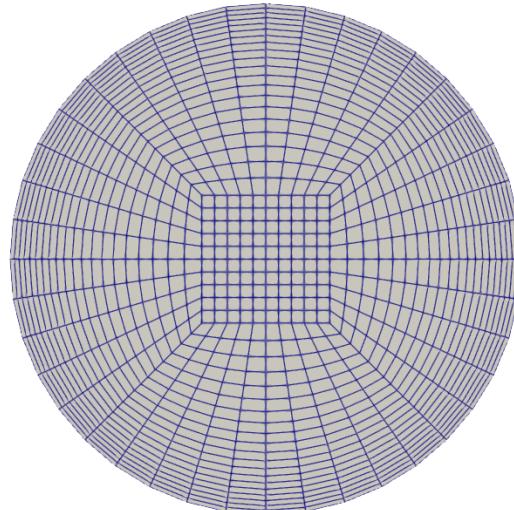


Figure 13. blockMesh with
O-Grid structure

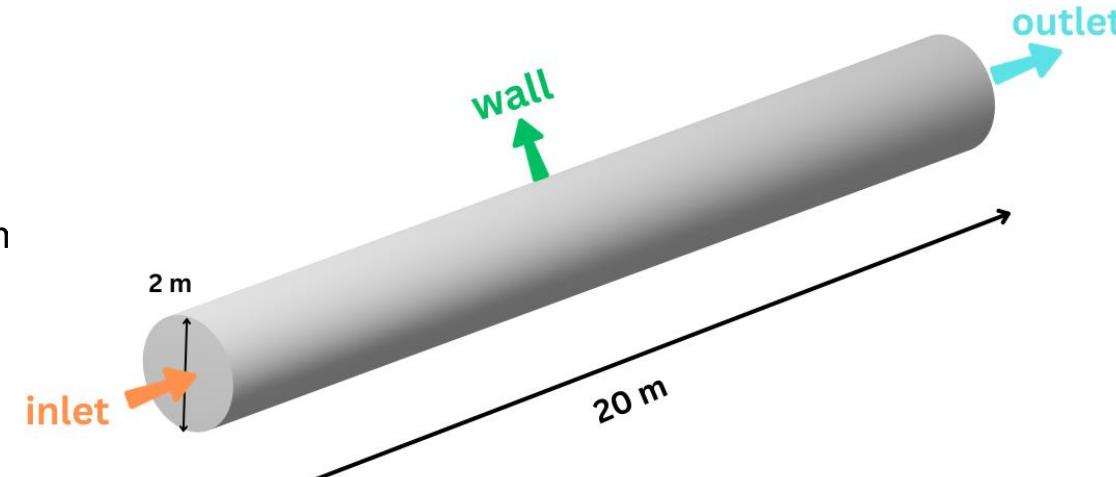


Figure 17. Pipe geometry

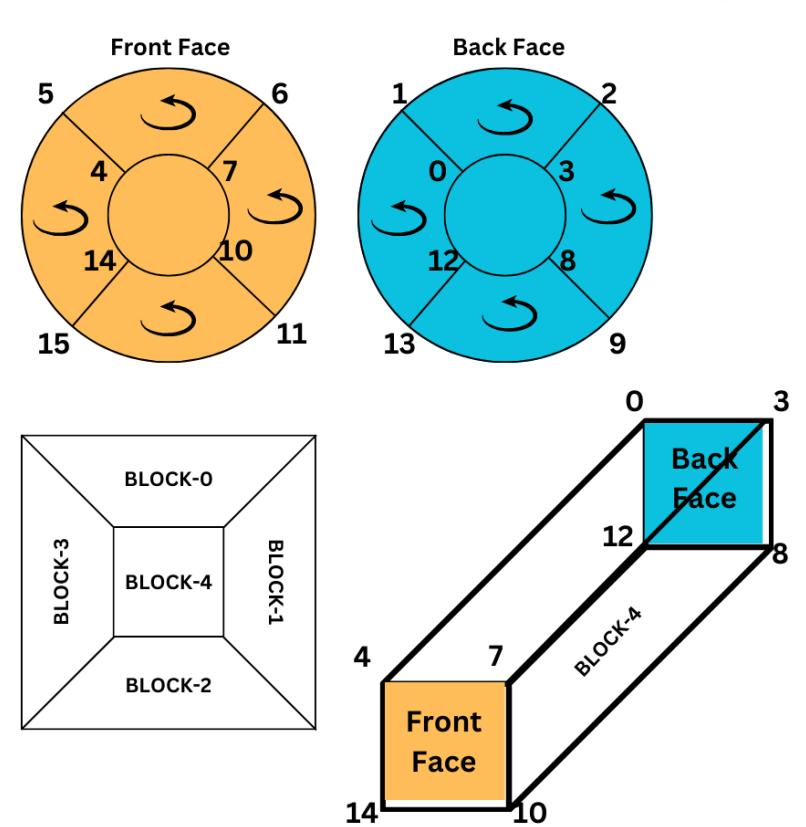
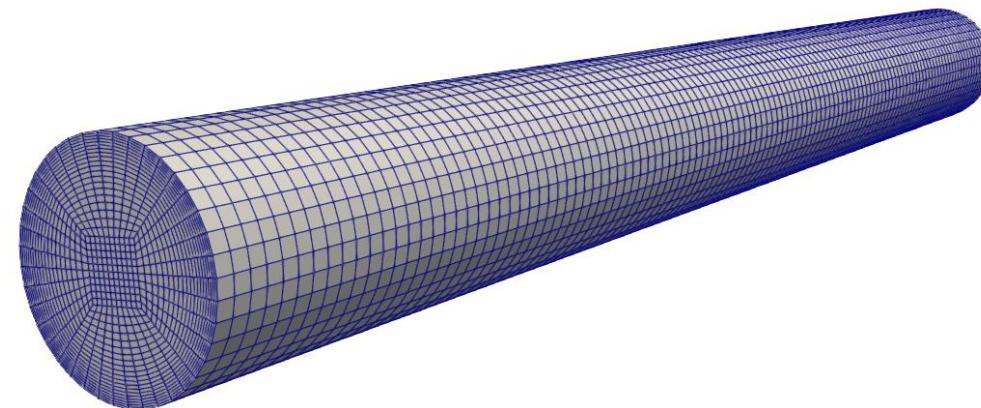


Figure 18. Block structure of pipe

- **snappyHexMesh** is an automatic split hex mesher that runs in serial or parallel.
- Reads geometry in triangulated formats, e.g. in STL, OBJ, VTK.
- **snappyHexMesh** can be used for complex geometries.
- Mesh generation relies on a dictionary file named ***snappyHexMeshDict*** in the ***system*** directory.
- The STL geometry is found within the ***constant/triSurface*** directory.
- To generate a mesh using **snappyHexMesh**, the following steps are taken:
 1. Creating a background or base mesh.
 2. Defining the geometry.
 3. Generating a **castellated** mesh or Cartesian mesh.
 4. Creating a **snapped** mesh or body-fitted mesh.
 5. Adding layers near the surfaces or **boundary layer** meshing.
 6. Checking mesh quality.

snappyHexMesh workflow

- The utility **surfaceFeatureExtract** can be utilized to extract feature edges from the STL geometry file.
- Castellation** involves removing cells located outside a predefined region set by a point.
- Snapping** reconstructs cells by adjusting their edges from within the area to align with the specified boundary.
- Layering** adds extra layers within the boundary region.

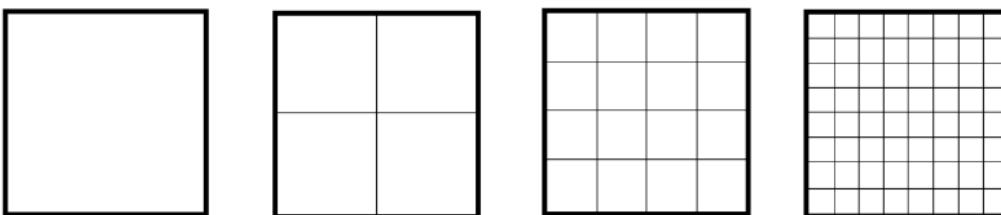


Figure 14. Refinement level 0, level 1, level 2, level 3

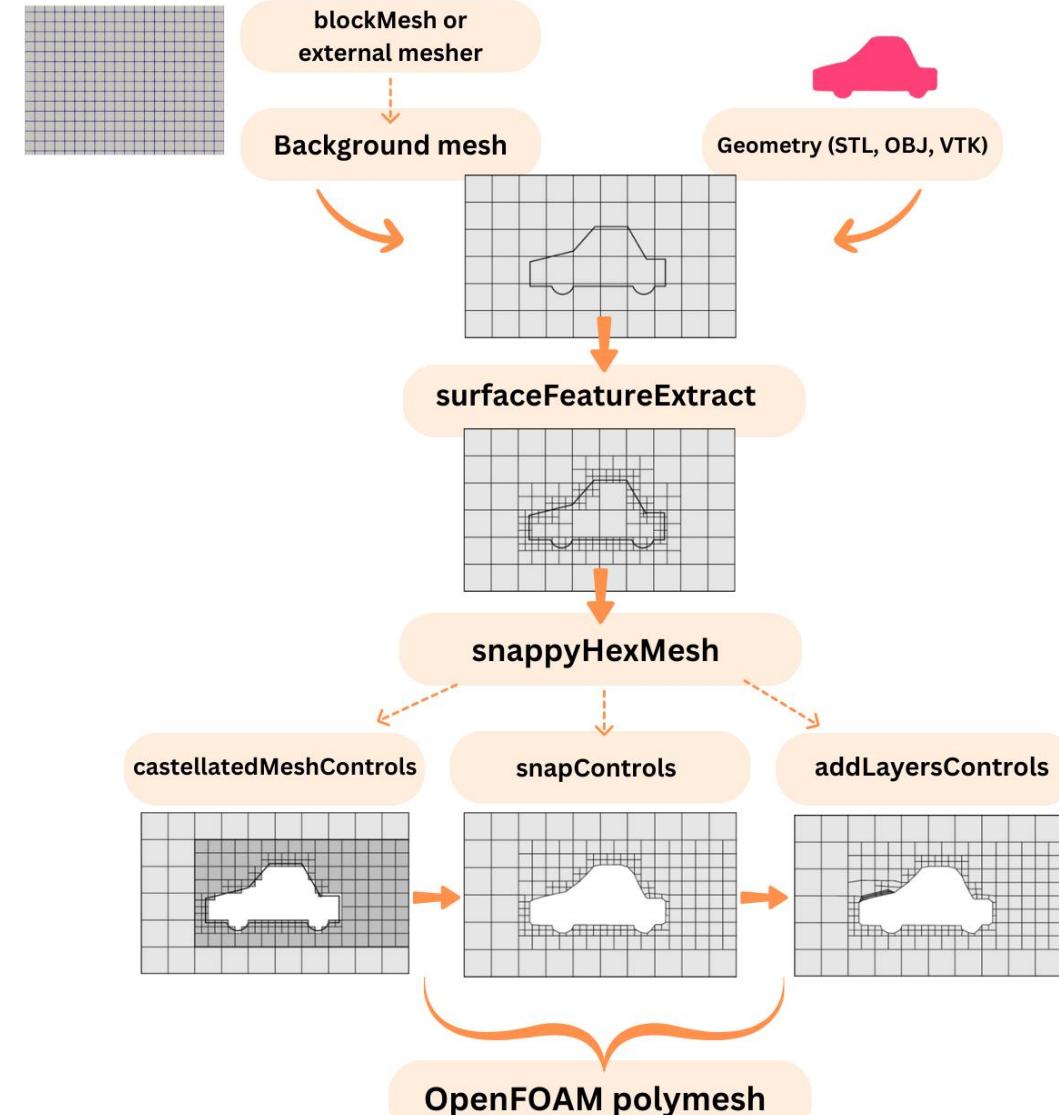


Figure 15. snappyHexMesh Workflow

6- Tank Example (3D – snappyHexMesh - internal)

```
$ of2206
$ blockMesh
$ surfaceFeatureExtract
$ snappyHexMesh
$ checkMesh
$ vim tank.foam
$ paraview-5.10 tank.foam
```

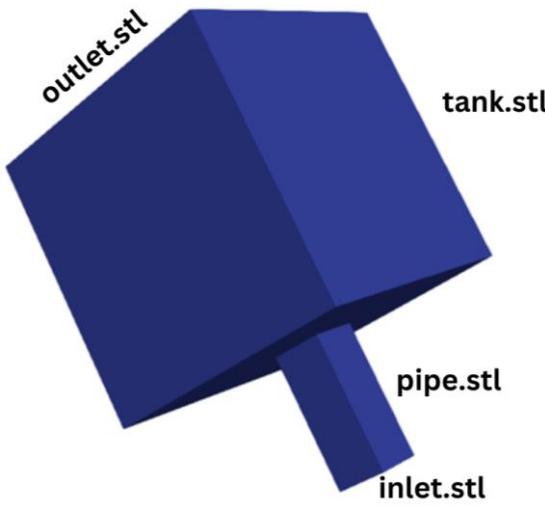


Figure 16. STL geometry

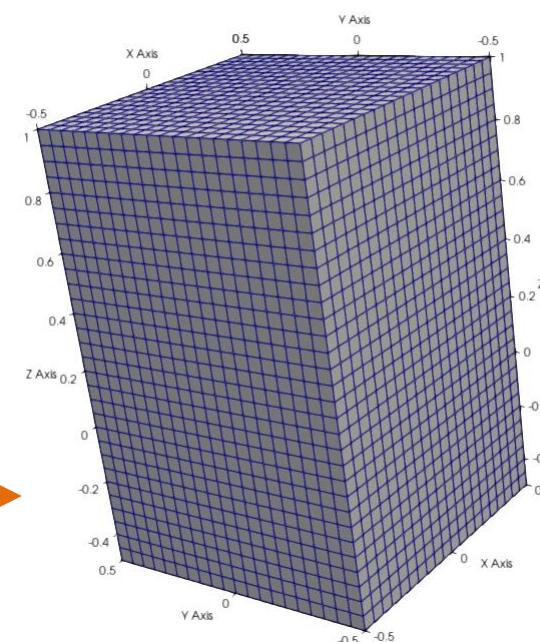


Figure 17. blockMesh

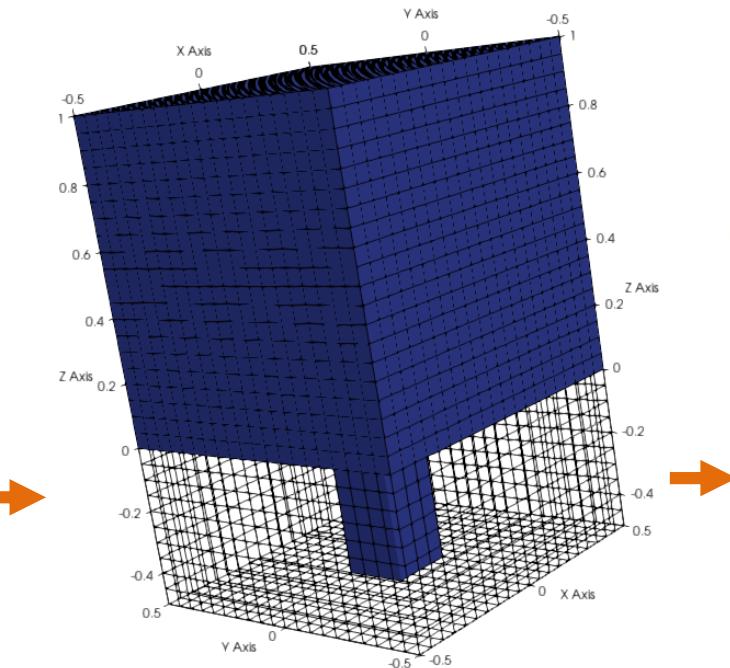


Figure 18. Showing STL and blockMesh together

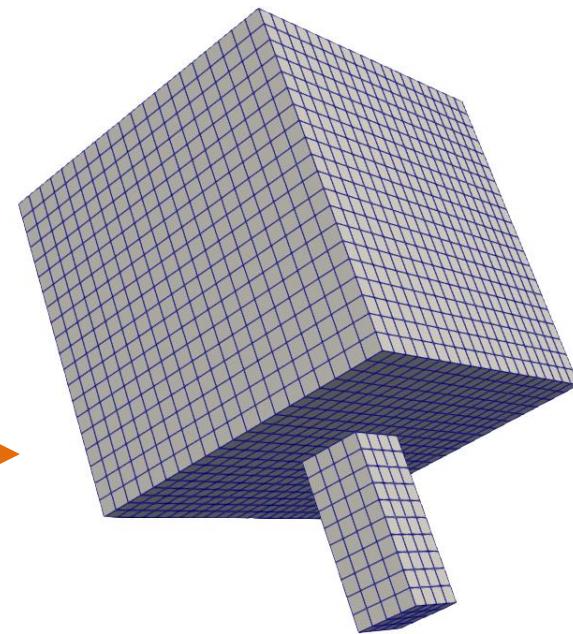


Figure 19. snappyHexMesh

7- Flange Example (3D – snappyHexMesh - internal)

```
$ of2206
$ blockMesh
$ surfaceFeatureExtract
$ snappyHexMesh
$ checkMesh
$ vim pipe.foam
$ paraview-5.10 pipe.foam
```

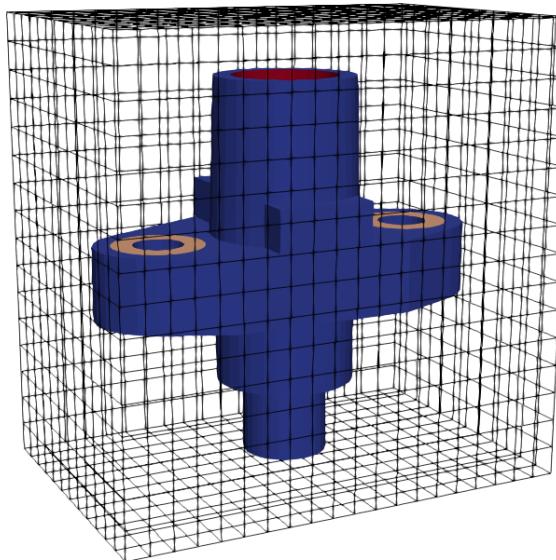


Figure 22. Showing STL and blockMesh together

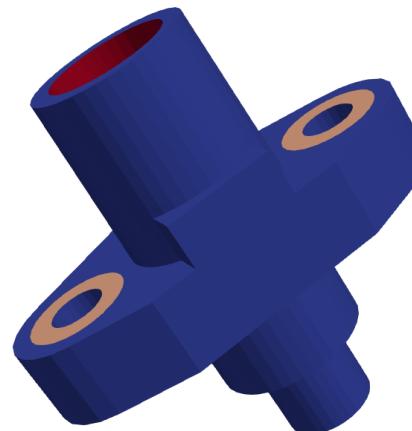


Figure 20. flange.stl

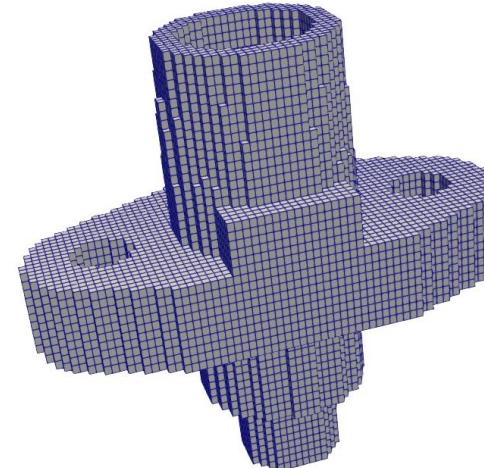


Figure 23. Castellation

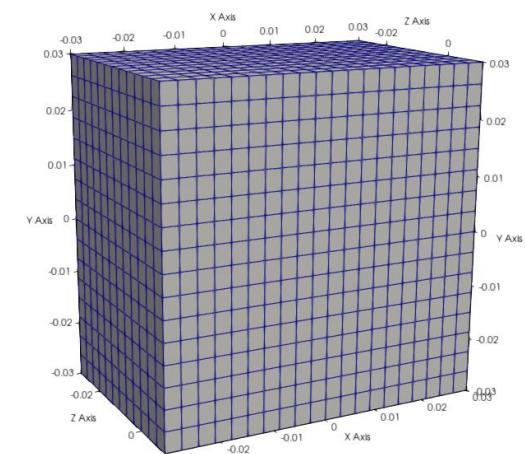


Figure 21. blockMesh

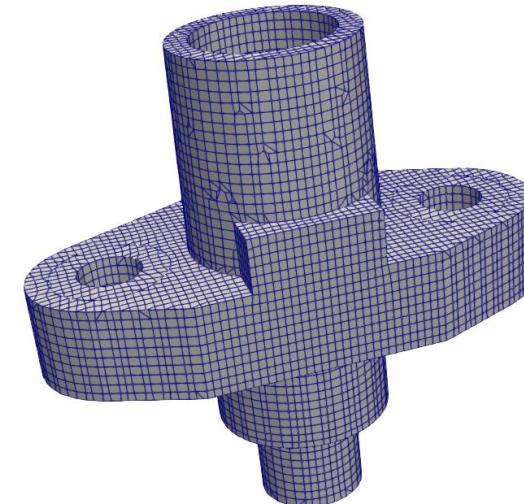


Figure 24. Snapping

8- Wing Example (2D – snappyHexMesh - external)

```
$ of2206
$ cp -r $FOAM_TUTORIALS/incompressible/pimpleFoam/RAS/wingMotion/
$ cd wingMotion/constant/triSurface
$ cp $FOAM_TUTORIALS/resources/geometry/wing_5degrees.obj.gz .
$ cd ..
$ blockMesh
$ snappyHexMesh
$ checkMesh
$ vim wing.foam
$ paraview-5.10 wing.foam
```



Figure 25. wing.stl

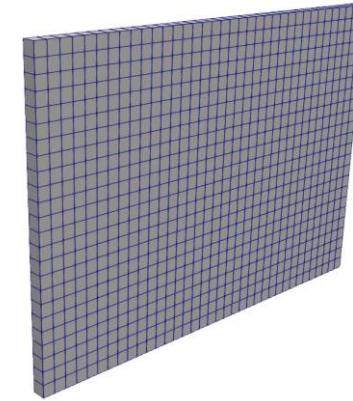


Figure 26. blockMesh

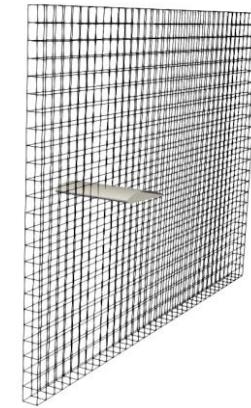


Figure 27. Showing STL and blockMesh together

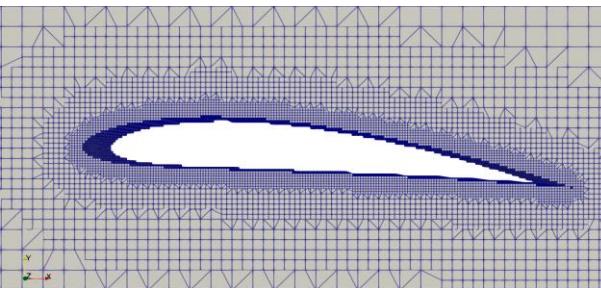


Figure 28. Castellation

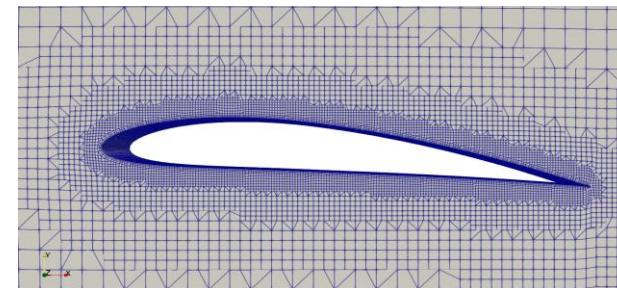


Figure 29. Snapping

Figure 30. Add layer

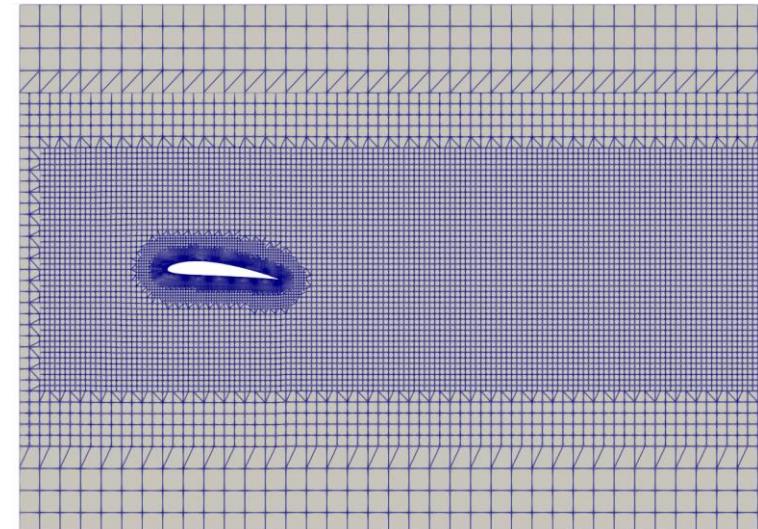


Figure 31. snappyHexMesh

9- Motorbike Example (3D – snappyHexMesh - external)

```
$ of2206
$ cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/motorbike/ .
$ cd motorbike/constant/triSurface
$ cp $FOAM_TUTORIALS/resources/geometry/motorBike.obj .
$ cd ..
$ blockMesh
$ snappyHexMesh
$ checkMesh
$ vim motorbike.foam
$ paraview-5.10 motorbike.foam
```

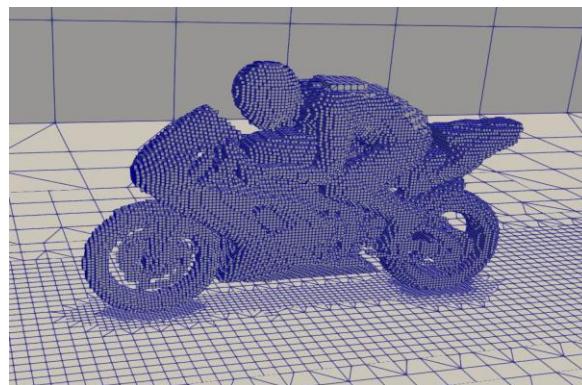


Figure 35. Castellation

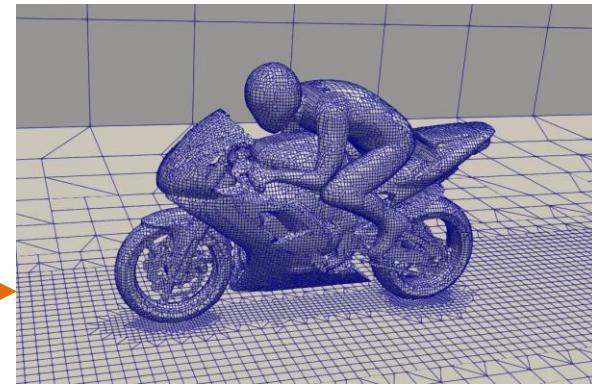


Figure 36. Snapping



Figure 32. motorBike.stl

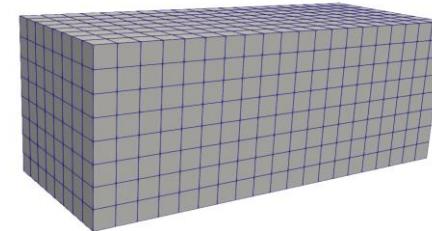


Figure 33. blockMesh

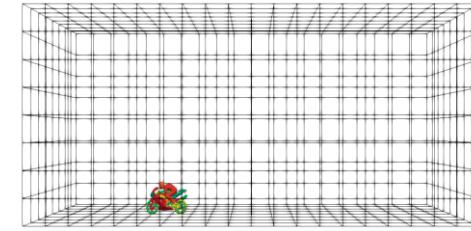


Figure 34. Showing STL
and blockMesh
together

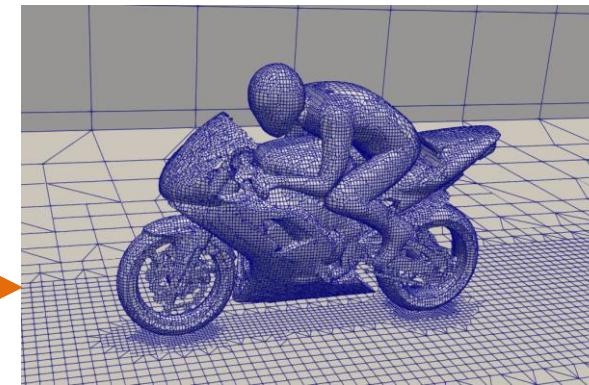


Figure 37. Add layer

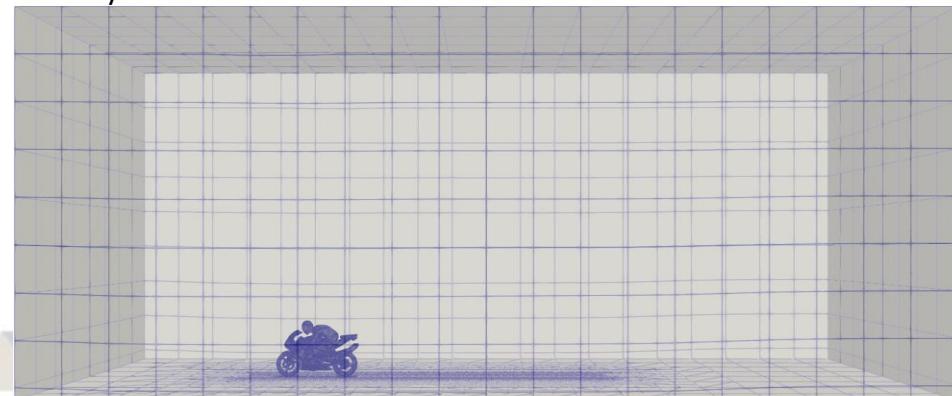


Figure 38. snappyHexMesh

10- Coronary Artery Example (3D - .msh)

```
$ of2206
$ fluentMeshToFoam lca.msh
$ checkMesh
$ vim lca.foam
$ paraview-5.10 lca.foam
```

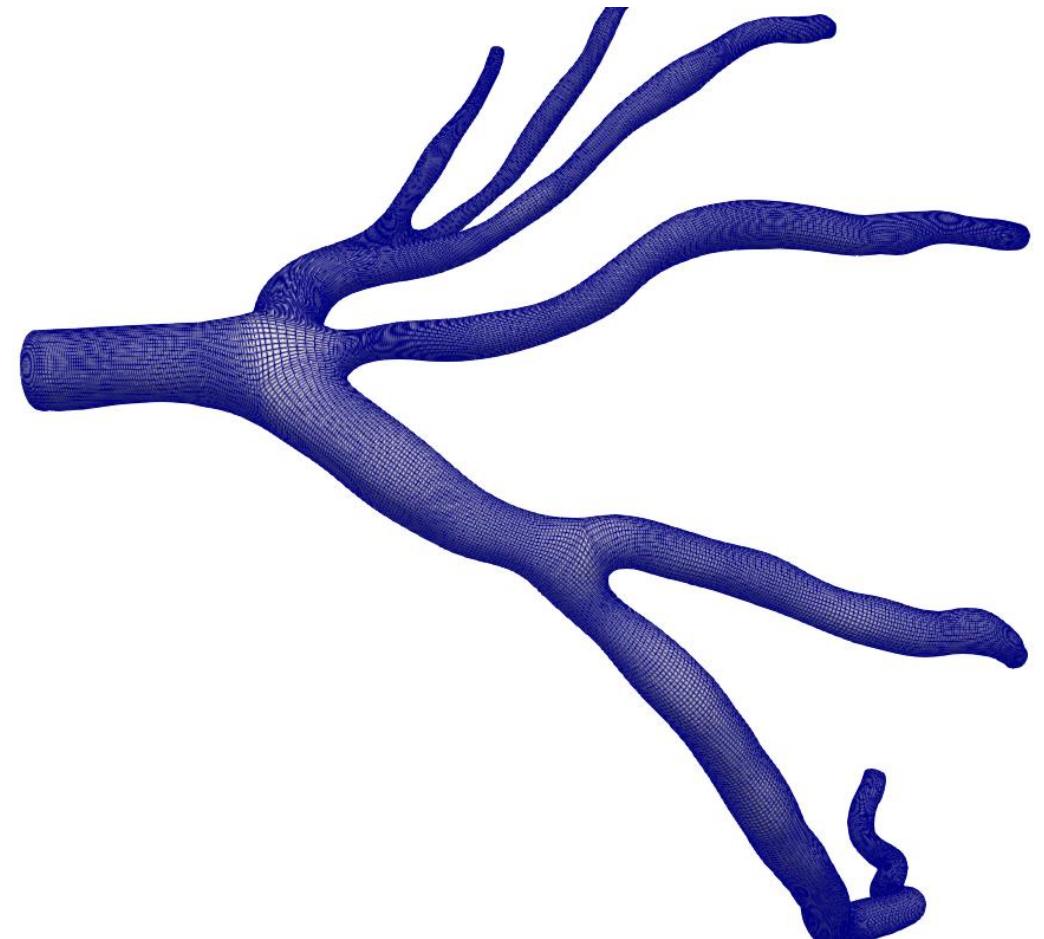
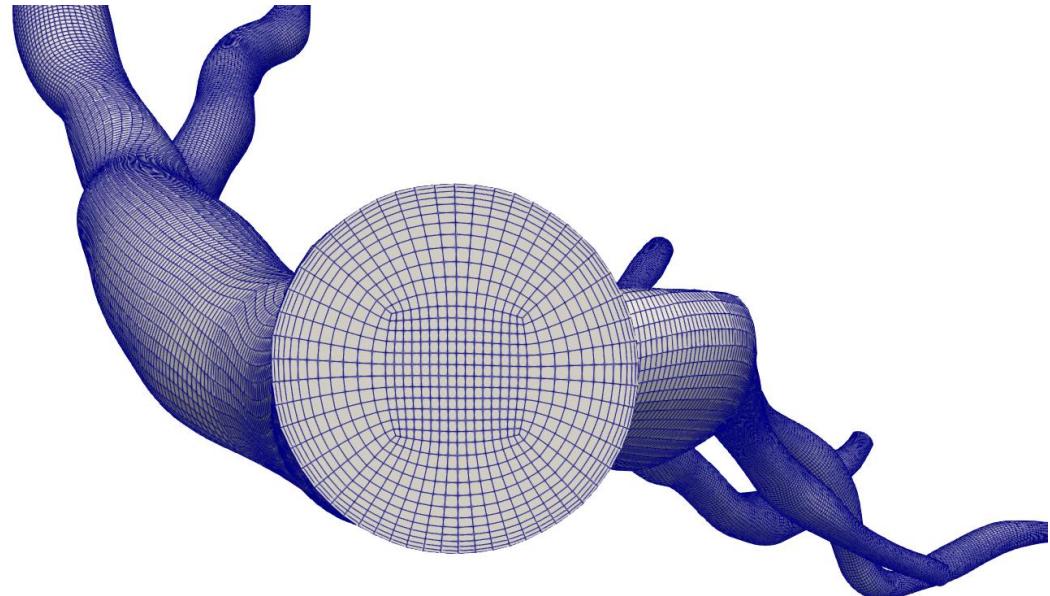


Figure 39. : Illustration of LCA mesh at inlet O-Grid structure

- OpenFOAM ESI tutorial guide: <https://www.openfoam.com/documentation/tutorial-guide>
- Engineering Fluid Dynamic - OpenFOAM Basic Training: <https://www.cfd.at/sites/default/files/tutorialsV7/OFTutorialSeries.pdf>
- Wolf Dynamic OpenFOAM Training: <http://www.wolfdynamics.com/training/introOF8/all.pdf>
- CFD Lectures with OpenFOAM by Dr. Cuneyt Sert: <https://users.metu.edu.tr/csert/foamyLectures/lecture03.html>
- Youtube channels:
 - <https://www.youtube.com/@OpenFOAMJozsefNagy/videos>
 - <https://www.youtube.com/@CFDAsmaaHadane/videos>
 - <https://www.youtube.com/@Feaforall/videos>
- Every Saturday morning Robin shares tips, tricks, thoughts & ideas to help you do the best CFD you can, with the resources you have: <https://www.cfdengine.com/>
- OpenFOAM Training by CFDSUPPORT: <https://www.cfdsupport.com/openfoam-training-by-cfdsupport.html>

- E. Kaya, E. Şimşek, and U. Şentürk, “AÇIK KAYNAK KODLU HESAPLAMALI AKIŞKANLAR DİNAMIĞİ ÇÖZÜCÜSÜ ‘OPENFOAM,’” 2022.
- M. Segui, F. R. Abel, R. Botez, and A. Ceruti, “High-fidelity aerodynamic modeling of an aircraft using OpenFoam – application on the CRJ700,” *Aeronaut. J.*, vol. 126, pp. 1–22, Oct. 2021, doi: 10.1017/aer.2021.86.
- F. Duronio and A. Di Mascio, “Blood Flow Simulation of Aneurysmatic and Sane Thoracic Aorta Using OpenFOAM CFD Software,” *Fluids*, vol. 8, no. 10, Art. no. 10, Oct. 2023, doi: 10.3390/fluids8100272.
- A. Shukla and A. Dewan, “IMPLEMENTATION OF OPENFOAM 4.1 FOR LES AND URANS APPROACHES,” 2019, pp. 67–82.

Thank you 😊

Hacer Duzman
MSc Student
Istanbul Technical University
Computational Science and Engineering
duzman21@itu.edu.tr