UNIVERSITY OF Southampton

Meshing with OpenFOAM

Uttam Cadambi Padmanaban[1]

[1]Department of Engineering
Aerodynamics and Flight Mechanics Group
The University of Southampton

October 16, 2025

Contents

UNIVERSITY OF Southampton

A good mesh *may* give you the right result, but a bad mesh will *definitely* give you the wrong result.

[1] blockMesh

UNIVERSITY OF Southampton

- The most basic meshing tool.
- Think of it as building the mesh with LEGOs.
- Available in OpenFOAM - no need for separate installation.
- Can pretty much handle all basic shapes, both 2D and 3D.
- **Cannot take STL files as input - complex geometries cannot be meshed this way.**
- Example usage - flat plate boundary layers, flow past cylinders, cubes, duct flows, channel flows, pipe flows.

[1] blockMesh

UNIVERSITY OF Southampton

- `blockMeshDict` in `system` dictionary is used to define the mesh.
- Treats domain/sub-domain as **closed** hexahedral blocks.
- The mesh is defined using
  - vertices.
  - blocks.
  - boundary.
- Ordering of vertices and blocks should be done carefully.
- I will demonstrate meshing of the following:
  - Flat plate.
  - Cylinder.

[1] blockMesh

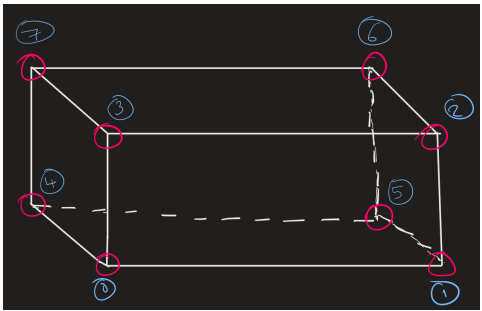

Figure: Vertices numbering

- The ordering of vertices should be done as shown in the figure for a single block (this can be for a flat plate).
- Once the vertices are ordered, the blocks are created.
- The faces are then added as **patches** and the type of patch (wall, generic patch, symmetry, cyclic) are included in this file.

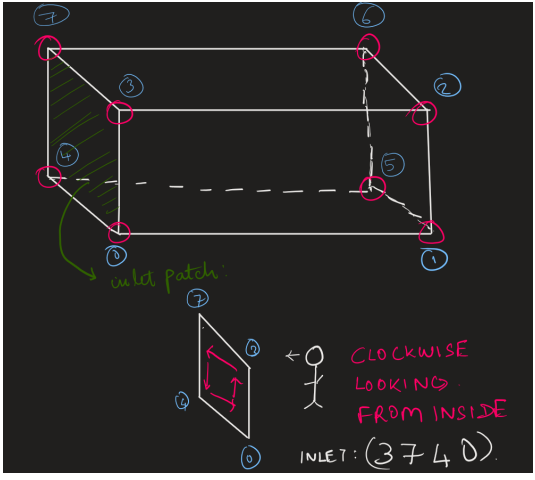

Figure: Faces numbering

- Edge sizing can be provided for the block as a whole.
- Uni-directional and bi-directional sizing possible. /item This can be done using expansion ratio.

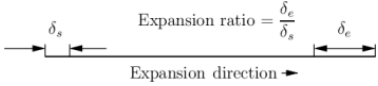

Figure: Edge expansion

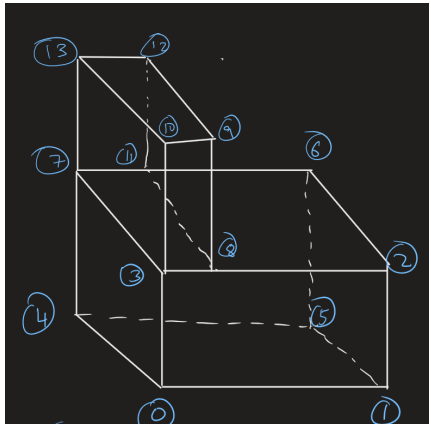

Figure: Vertices numbering for a multi-block

Let us look at some examples. For more details, visit: `https://bit.ly/3KTisui`

[2] cfMesh

UNIVERSITY OF Southampton

- If `blockMesh` is like building with LEGOs, `cfMesh` is like a smart 3D scanner and filler.
- Provide a 3D model of your object (typically in the form of an STL file) and it automatically fills in the spaces with cells.
- It is very smart - fills up the space so that the regions near the object have smaller cells compared to the regions away (unless explicitly asked by the user to do otherwise).
- It is wicked fast and has very low memory consumption.
- I will show ones example:
  - An airfoil placed close to the ground.

Requirements for constructing a **2D mesh** using cfMesh

- STL file of the object to be meshed (keep in mind, if this is a simple geometry, you can create it using primitive shapes in cfMesh). However, I will be showing everything using STL files.
- A bounding box.
- That is it.

I will demonstrate this purely using the method that I follow. **Please bear in mind: there may be simpler ways of doing this. You are more than welcome to look it up. Feel free to use this as the starting point.**

For 2D meshes, you need a bounding box **without the front and back or lateral faces**. You should also ensure that the depth of the box coincides with the depth of the object. The best way to do this is to first use `blockMesh` to create a single

hexahedral block and then extract the inlet, outlet, top and bottom faces as STL files.

OR

If you are well-versed with CAD - just create an STL file with these surfaces (it is essentially a frame).

For 2D meshes, you need a bounding box **without the front and back or lateral faces**. You should also ensure that the depth of the box coincides with the depth of the object. The best way to do this is to first use `blockMesh` to create a single

hexahedral block and then extract the inlet, outlet, top and bottom faces as STL files.

OR

If you are well-versed with CAD - just create an STL file with these surfaces (it is essentially a frame).

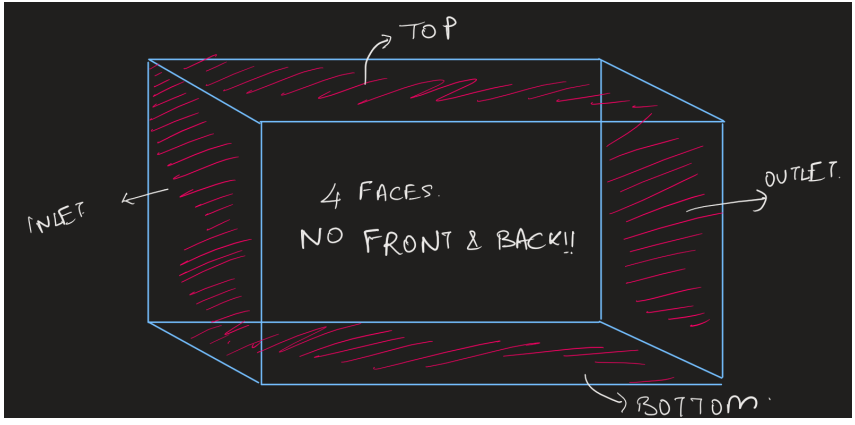

Figure: The bounding box for cfMesh

[2] cfMesh

UNIVERSITY OF Southampton

Creation of the bounding box:

- Use `blockMesh` (why, of course!).
- Make sure to set the out-of-plane length (or depth or lateral thickness) to match that of your object STL (in this case, it is an airfoil and it is 0.1 units).
- Extract the STL from a patch using `surfaceMeshTriangulate`. Here is the command `surfaceMeshTriangulate -patches '(inlet outlet top bottom)' bb.stl`
- Needless to say, the command in the previous point uses the patches that YOU used in YOUR `blockMeshDict` file. So inlet outlet top and bottom could be four patch names that are entirely different for you. Addtionally, the file name "bb.stl" could also be different.
- Now you have a file called bb.stl that has the four faces. Assuming you have an STL file of an airfoil called "airfoil_0deg.stl", the next step is to merge the two.
- Simply copy the contents of the STL file "airfoil_0deg.stl" and append it to the top of the "bb.stl" file. Rename this to "airfoil.stl".
- Visualize in paraview to check if everything looks fine.

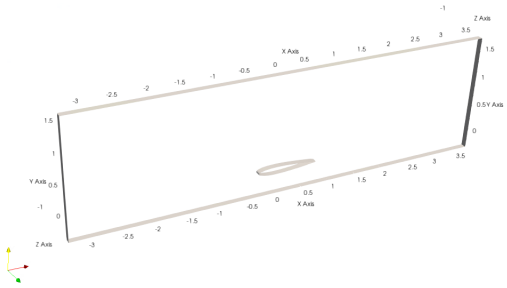

Figure: Completed STL file with domain and airfoil

Convert to fms using the command: surfaceFeatureEdges -angle <resolution_angle> <*.stl> <*.fms>

- We use the `meshDict` file to change the settings of the mesh.
- There are three components to it:
  - Local refinement.
  - Object refinement.
  - Boundary layers.
- Once the settings are ready, you simply run the command `cartesian2DMesh` and your domain is meshed.

Let us look at some examples.

[3] Importing meshes from ANSYS

- It is possible to import meshes from other platforms.
- I will be showing how to do this from ANSYS Fluent.
- Once you are done meshing in ANSYS, you need to export mesh as `*.msh` format. **Make sure that you switch the format from binary to ASCII**.
- You can use the command `fluentMeshToFoam` from the command line to convert your ANSYS fluent mesh which is in `*.msh` format to OpenFOAM format.
- Additionally you have to also run `autoPatch` and `createPatch` commands (more on that in the following section).

[3] Importing meshes from ANSYS

UNIVERSITY OF Southampton

Let us look at some examples.

[4] Mesh renumbering, creating patches, editing the boundary file

UNIVERSITY OF Southampton

- The renumberMesh utility in OpenFOAM is used to optimize the numbering of mesh cells, reducing bandwidth and improving computational efficiency. Below is a guide on how to use it effectively.
- When you use a feature such as `fluentMeshToFoam`, you need to rename the patches and also their types.
- This is best done using `createPatch`.
- Bear in mind - the boundary file in `polyMesh` directory can be changed.
- Let us look at these in detail.

[5] Mesh quality

UNIVERSITY OF Southampton

- As always, the mesh quality is very important.
- OpenFOAM uses the `checkMesh` utility that can detect the quality of the mesh.
- In addition to that, it can also detect the mismatch of cyclic faces and the extent of the mismatch.
- If the mesh fails on certain metrics, a cell set is created immediately.
- It is possible to then visualize this cell set using the `foamToVTK` utility.
- The syntax is: `foamToVTK -cellSet <cellSetName>` where `cellSetName` can be obtained from `constant/polyMesh/sets`.
- I will demonstrate this in OpenFOAM.