Advaced meshing options

Open-Source CFD Course 2021 – Lab session 5

Miklós BALOGH
and
Josh DAVIDSON

2021

GMSH tutorial

- Create unstructured hexahedral, tetrahedral and polyhedral meshes with Gmsh.
  - Write input files (.geo) for these purposes
  - Open these files with Gmsh to see how they looks like
  - Create scripts for the different tasks
  - Execute the simulations and compare the results with paraFoam

Create the points

Create the Gmsh script

```
mkdir gmshMeshes
cd gmshMeshes
gedit cavity.geo &
```

Set everything parametrically

```
// Input
intervalNumber = 40; // Number of intervals per side
layerNumber = 1;     // Only one cell layer, since it is 2D
squareSide = 0.1;    // Size in meters
meshThickness = squareSide / intervalNumber;
gridSize = squareSide / intervalNumber;
```

Create corner points

```
// Create points
Point(1) = {0,          0,          0, gridSize};
Point(2) = {squareSide, 0,          0, gridSize};
Point(3) = {squareSide, squareSide, 0, gridSize};
Point(4) = {0,          squareSide, 0, gridSize};
```

Create lines and a surface

Create lines between points

```
// Create lines
Line(1) = {1, 2};        // bottom line
Line(2) = {2, 3};        // right line
Line(3) = {3, 4};        // top line
Line(4) = {4, 1};        // left line
```

Create a surface from the lines listed in a loop

```
// Create surface (first a line loop for that)
Line Loop(5) = {1, 2, 3, 4}; // Counter-clockwise
Plane Surface(6) = {5};      // Surface from the loop
```

Create volume via extrusion

Create a volume via extruding the previously created surface

```
// Create volume (store entity ID-s in surfaceVector[])
surfaceVector[] = Extrude {0, 0, meshThickness}
{
  Surface{6};
  Layers{layerNumber};
  Recombine;
};
/* surfaceVector contains entities in the following order:
[0]  - back surface (opposed to source surface)
[1] - extruded volume
[2] - bottom surface (belonging to 1st line in "Line Loop (5)")
[3] - right surface (belonging to 2nd line in "Line Loop (5)")
[4] - top surface (belonging to 3rd line in "Line Loop (5)")
[5] - left surface (belonging to 4th line in "Line Loop (5)") */
```

Create BC-s for the case and define mesh

Define patch names for boundary conditions

```
Physical Surface("frontAndBack") = {surfaceVector[0],6};
Physical Volume("internal") = surfaceVector[1];
Physical Surface("fixedWalls") = {
  surfaceVector[2],surfaceVector[3],surfaceVector[5]
};
Physical Surface("movingWall") = surfaceVector[4];
```

The following should be included only for the hexahedral mesh

```
// This will recombine the surface mesh
// from triangles to quadrilaterals
Recombine Surface{6};
```

Generate the mesh and convert it to openFoam

In command line, do the following

```
# Create the Gmsh mesh file via command line
gmsh -3 -format msh2 cavity.geo -o cavity.msh

# Load the openFoam environment
OF2012

# Take a copy from the original cavity case
cd $FOAM_RUN/tutorials/incompressible/icoFoam/cavity/
cp -r cavity cavityGMSH # or cavityGMSHt, cavityGMSHq, cavityGMSHp

# Convert Gmsh mesh to OpenFOAM
cd cavityGMSH
cp <pathtomesh>/cavity.msh
gmshToFoam cavity.msh
```

For the boundary file correction

```
# Find a prototype for changeDictionaryDict
grep -r "changeDictionary" $FOAM_RUN/tutorials
cp <one_from_the_results> system/changeDictionaryDict
```

Modify the mesh boundaries via changeDictionary

Modify system/changeDictionaryDict

```
boundary
{
    "fixedWalls"
    {
        type            wall;
        physicalType    wall;
    }

    "movingWall"
    {
        type            wall;
        physicalType    wall;
    }

    "frontAndBack"
    {
        type            empty;
        physicalType    empty;
    }
}
```

Topology change for polyhedral mesh

Find, copy and modify a system/topoSetDict file to this content

```
actions
(
    {
        name    c0;
        type    cellSet;
        action  clear;
    }
    {
        name    c0;
        type    cellSet;
        action  invert;
    }
    {
        name    c0;
        type    cellSet;
        action  subtract;
        source  boxToCell;
        box     (0 0 0.00125) (1 1 0.0025);
    }
);
```

Write a script for meshing

Write the following script named as e.g. "Meshing"

```
#!/bin/bash

# Source tutorial clean functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

# Create mesh with GMSH 4.3.0
gmsh -3 -format msh2 cavity.geo -o cavity.msh > log.GMSH

# Convert the GMSH mesh to OpenFOAM mesh
runApplication gmshToFoam cavity.msh

# Change boundary types
runApplication changeDictionary

# Some extra part should be included here for polyhedral meshes
```

Write a script for polyhedral meshing

Include this part to “Meshing” script for polyhedral mesh

```
# Backup 0 dir (polyDulaMesh overwrite it)
cp -r 0 0.orig

# Create poly-dual of the tetra mesh
runApplication polyDualMesh 75 -overwrite

# Delete the unnecessary 2nd layer
runApplication topoSet
runApplication subsetMesh c0 -overwrite -patch side2

# Fix the mesh
runApplication combinePatchFaces 180 -overwrite

# Restore 0 dir
rm -r 0
cp -r 0.orig 0
```

Write a script to cleanup and execute simulation

Write the following script called “Allrun”

```
#!/bin/bash

# Source tutorial clean functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

# Running the simulation
runApplication `getApplication`
```

Write the following script called “Allclean”

```
#!/bin/bash

# Source tutorial clean functions
. $WM_PROJECT_DIR/bin/tools/CleanFunctions

cleanCase
```

Execute the simulations using the scripts

Execute the scripts from command line

```
# Make them executable
chmod +x Meshing Allrun Allclean

# Load the OpenFOAM environment
OF2012

# Run them one-by-one
./Allclean
./Meshing
./Allrun
```