Advaced meshing options

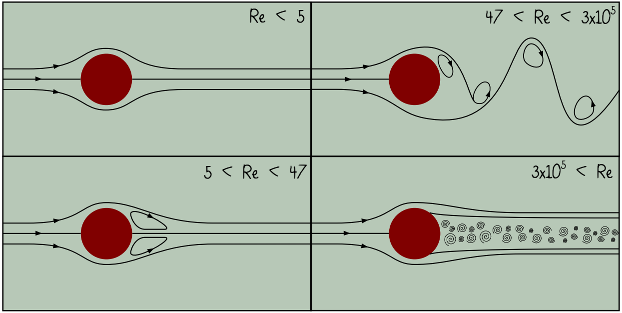
Open-Source CFD Course 2021 – Lab session 6

Miklós BALOGH
and
Josh DAVIDSON

2021

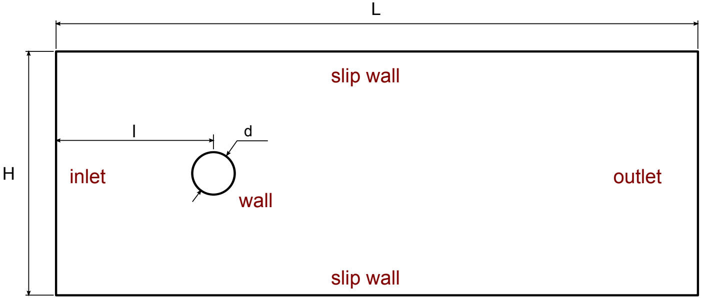
Stationary and transient flows

- Transient (unsteady) flows
  - Every real flow is transient in molecular sense
- Stationary (steady-state) flow refers to the condition
  - where the fluid properties do not change over time
  - all the time derivatives of a flow field are vanishing



Laboratory tasks I.

1. Create the geometry and mesh for the vortex street simulation with GMSH
   - Make a copy the cavity case with a new name (e.g. vonKarman)
   - Create a GMSH geo file according to the geometry:
     - $L = 25 \times d$, $H = 10 \times d$, $l = 5 \times d$, $d = 0.1m$



GMSH geo file - Geometry setup

```
// Geometry setup (parameters in meters)
cylinderD = 0.1;
cylinderX = 0.0;
cylinderY = 0.0;

channelWidth  = 0.1*cylinderD;
channelHeight = 10*cylinderD;
channelLength = 25*cylinderD;

inletX = cylinderX - 5*cylinderD;
outletX = channelLength - inletX;
```

GMSH geo file - Flow and mesh setup

```
// Flow setup (Re = Uin*D/nu -> Uin = Re*nu/D)
Re  = 400;
nu  = 0.001;
rho = 1.0;
Uin = Re*nu/cylinderD;
Printf("Uin      = %g",Uin);

// Mesh setup
desiredYplus=1.0;
desiredTplus=5*desiredYplus;
cellExpansionRatio = 1.2;
```

GMSH geo file - Calculate wall spacing

```
// Calculate wall spacing at the cylinder
Cf = 0.664/Sqrt(Re);       // Friction coeff. (laminar)
Tw = Cf*rho*(Uin^2)/2.0; // Wall shear stress
Utau = Sqrt(Tw/rho);       // Friction velocity
symin = 1.8*desiredYplus*nu/Utau; // Wall normal cell size
stmin = symin*desiredTplus/desiredYplus; // Tangential cell size

Printf("Cf      = %g",Cf);
Printf("TauWall = %g",Tw);
Printf("Utau    = %g",Utau);
Printf("symin   = %g",symin);
Printf("sxmin   = %g",stmin);
Printf("dt      = %g",symin/Uin);
```

GMSH geo file - Store short named variables


```
// Set shorter variable names
cX = cylinderX;
cY = cylinderY;
cR = cylinderD/2.0;

cW = channelWidth;
cH = channelHeight;
cL = channelLength;

iX = inletX;
oX = outletX;

q  = cellExpansionRatio;
```

GMSH geo file - Calculate cell spacing


```
// Point sizing parameters (based on geometric progression)
// an = a1*q^(n-1)
// Sn = a1*(q^n - 1)/(q - 1)
// n = log(Sn*(q - 1)/a1 + 1)/log(q)

// At the cylinder
dlcy = symin;

// At the inlet
Sn = cX - iX;
a1 = dlcy;
n = Round(Log(Sn*(q - 1.0)/a1 + 1.0)/Log(q));
an = a1*q^(n-1);
dlch = (an > 10*a1) ? 10*a1 : an; // dlch=Min(an,10*a1)
Printf("an/a1    = %g",an/a1);
```

GMSH geo file - Create cylinder (points and arcs)


```
// Create cylinder center
ipcy0 = newp;
Point(ipcy0) = {cX, cY, -cW/2.0, dlcy};

// Create cylinder points along perimeter (clockwise)
Npp = 4;
deltat = 2*Pi/Npp;

t = Pi;
For ip In {0 : Npp-1}
  ipcy[ip] = newp;
  Point(ipcy[ip]) = {cX + cR*Cos(t), cY + cR*Sin(t), -cW/2.0, dlcy};
  t -= deltat;
EndFor

// Create cylinder arcs (using start, center and end points)
For ip In {0 : Npp-2}
  ilcy[ip] = newl;
  Circle(ilcy[ip]) = {ipcy[ip],ipcy0,ipcy[ip+1]};
EndFor
ilcy[Npp-1] = newl;
Circle(ilcy[Npp-1]) = {ipcy[Npp-1],ipcy0,ipcy[0]};
```

GMSH geo file - Create channel (points and lines)

```
// Create channel (clockwise point order)
ipch[0] = newp;
Point(ipch[0]) = {iX, cY - cH/2.0, -cW/2.0, dlch};
ipch[1] = newp;
Point(ipch[1]) = {iX, cY + cH/2.0, -cW/2.0, dlch};
ipch[2] = newp;
Point(ipch[2]) = {oX, cY + cH/2.0, -cW/2.0, dlch};
ipch[3] = newp;
Point(ipch[3]) = {oX, cY - cH/2.0, -cW/2.0, dlch};

ilch[0] = newl;
Line(ilch[0]) = {ipch[0],ipch[1]};
ilch[1] = newl;
Line(ilch[1]) = {ipch[1],ipch[2]};
ilch[2] = newl;
Line(ilch[2]) = {ipch[2],ipch[3]};
ilch[3] = newl;
Line(ilch[3]) = {ipch[3],ipch[0]};
```

GMSH geo file - Create domain (line loops, plane and volume)


```
// Create line loop from cylinder arcs
illcy = newll;
Line Loop(illcy) = {ilcy[]};

// Create line loop from channel lines
illch = newll;
Line Loop(illch) = {ilch[]};

// Create surface
is = news;
Plane Surface(is) = {illcy, illch};

// Recombine triangles to quads
Recombine Surface {is};

// Extrude the surface
sVec[] = Extrude {0, 0, cW}
{
  Surface{is};
  Layers{1};
  Recombine;
};
```

GMSH geo file - Setup boundaries

```
// Set BC-s

/* sVec contains surfaces in the following order:
 [0]            - front surface (opposed to source surface (is))
 [1]            - extruded volume
 [2:Npp+1]      - surfaces (belonging to lines of cylinder)
 [Npp+2:Npp+5] - surfaces (belonging to lines of channel)
*/

For isc In {0 : Npp-1}
  sCyl[isc] = sVec[isc+2];
EndFor
Physical Volume("internal")   = {sVec[1]};
Physical Surface("backSide")  = {is};
Physical Surface("frontSide") = {sVec[0]};
Physical Surface("cylinder")  = {sCyl[]};
Physical Surface("bottom")    = {sVec[Npp+2]};
Physical Surface("outlet")    = {sVec[Npp+3]};
Physical Surface("top")       = {sVec[Npp+4]};
Physical Surface("inlet")     = {sVec[Npp+5]};
```

GMSH geo file - Mesh refinement 1.


```
// Create fields for sizing
Field[1] = Attractor;
Field[1].FacesList = {sCyl[]};

// Based on the attractor, create a threshold field
Field[2] = Threshold;
Field[2].IField = 1;
Field[2].LcMin = dlcy;
Field[2].LcMax = dlch;
Field[2].DistMin = 4*dlcy;
Field[2].DistMax = Sn + 4*dlcy;

// Refine mesh in a box 1
Field[3] = Box;
Field[3].VIn  = 3*dlcy;
Field[3].VOut = dlch;
Field[3].XMin = cX;
Field[3].XMax = oX;
Field[3].YMin = cY - 3*cR;
Field[3].YMax = cY + 3*cR;
Field[3].ZMin = -cW;
Field[3].ZMax =  cW;
```

GMSH geo file - Mesh refinement 2.


```
// Refine mesh in a box 2
Field[4] = Box;
Field[4].VIn  = 6*dlcy;
Field[4].VOut = dlch;
Field[4].XMin = cX;
Field[4].XMax = oX;
Field[4].YMin = cY - 6*cR;
Field[4].YMax = cY + 6*cR;
Field[4].ZMin = -cW;
Field[4].ZMax =  cW;

// Refine mesh in around the cylinder
Field[5] = Cylinder;
Field[5].VIn     = 6*dlcy;
Field[5].VOut    = dlch;
Field[5].Radius  = 6*cR;
Field[5].XAxis   = 0.0;
Field[5].XCenter = cX;
Field[5].YAxis   = 0.0;
Field[5].YCenter = cY;
Field[5].ZAxis   = 1.0;
Field[5].ZCenter = 0.0;
```

GMSH geo file - Mesh refinement 3.

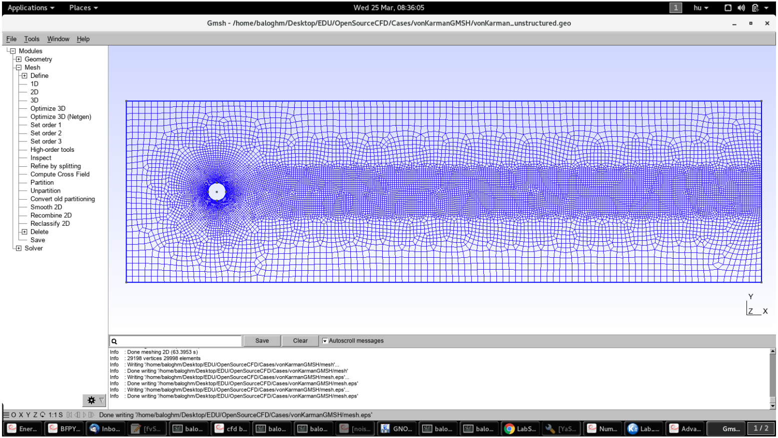

```
// Set a background field, as the min. of field 2 & 3
Field[6] = Min;
Field[6].FieldsList = {2, 3, 4, 5};
Background Field = 6;

// Set mesh algorithm
// 2D algorithm:
// 1=MeshAdapt, 2=Automatic
// 5=Delaunay, 6=Frontal
// 7=BAMG, 8=DelQuad
Mesh.Algorithm = 8;

// Set mesh smoothing steps
Mesh.Smoothing = 3;
```

Install latest version of GMSH


```
# Open a new terminal window and create Install dir
cd
mkdir Install

# Download GMSH 4.8.0 from command line
cd Install
wget http://gmsh.info/bin/Linux/gmsh-4.8.0-Linux64.tgz

# Unpack the tarball
tar -xvf gmsh-4.8.0-Linux64.tgz

# Add an alias to bashrc (in order to start gmsh 4.8.0)
cd
echo "alias gmsh480='/home/openCFD/Install/gmsh-4.8.0-Linux64/bin/gmsh'" >> .bashrc

# Reload bashrc
. .bashrc

# Start gmsh 4.8.0
gmsh480 &
```

Mesh display

Open the .geo file from gmsh



Convert to openFoam


```
# Load the openFoam environment
OF2012

# Take a copy from the original cavity case
cd $FOAM_RUN/tutorials/incompressible/icoFoam/cavity
cp -r cavity vonKarmanGMSH

# Copy geo file
cp <geofilename>.geo vonKarmanGMSH/.

# Create mesh and save to appropriate format
cd vonKarmanGMSH
gmsh480 -3 -format msh2 <geofilename>.geo -o <meshfilename>.msh

# Conversion
gmshToFoam <meshfilename>.msh

# Create changeDictionary
cp system/controlDict system/changeDictionaryDict
```

Modify the constant/polyMesh/boundary file

Edit: gedit system/changeDictionaryDict &

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  7                                     |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      changeDictionaryDict; // changed from controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

Modify the constant/polyMesh/boundary file

```
boundary
{
    "inlet"
    {
        type            patch;
        physicalType    patch;
    }
    "outlet"
    {
        type            patch;
        physicalType    patch;
    }
    "top"
    {
        type            patch;
        physicalType    patch;
    }
    "bottom"
    {
        type            patch;
        physicalType    patch;
    }
    ...
```

Modify the constant/polyMesh/boundary file

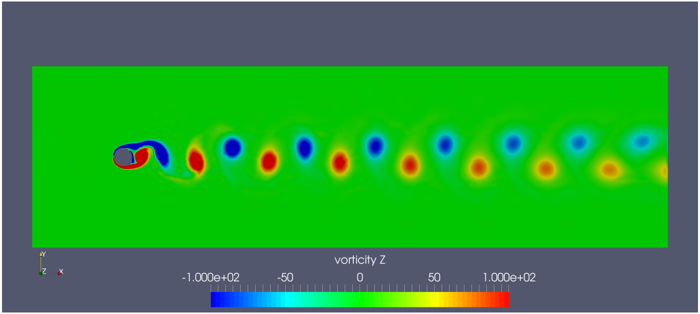

```
    ...
    "cylinder"
    {
        type            wall;
        physicalType    wall;
    }
    "backSide"
    {
        type            empty;
        physicalType    empty;
    }
    "frontSide"
    {
        type            empty;
        physicalType    empty;
    }
}
```

Modifications on boundary conditions

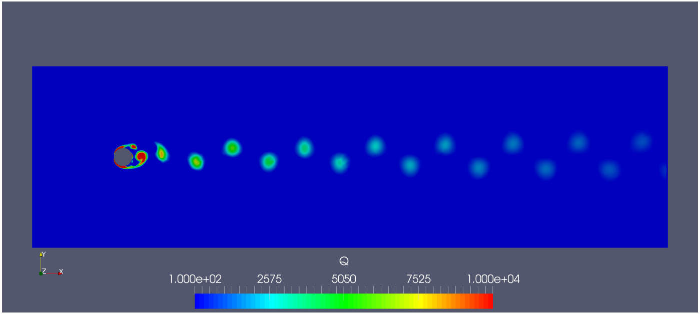
2. Modify BC-s (0/U and 0/p) according to the geometry ($U = 4ms^{-1}$)
3. Modify the viscosity (constant/transportProperties), according to $Re = 400$
4. Modify system/contolDict (according to the CFL):
   - $U = 4$ ms^{-1}, $\Delta s_{min} \approx 0.0035$ m
   - $\Delta t \approx \frac{\Delta s_{min}}{U} \approx 8$ s
   - $t_{end} >= 2$ s

Running and post-processing

2. Run the simulation!
3. Calculates vorticity and Q
   - postProcess -func vorticity
   - postProcess -func Q
4. Visualize the results (vorticity, Q)



Post-processing



Assignments

1. What is the total volume of domain in the von Karman vortex street case?
2. Where should be positioned a line source of the streamlines for the best visual experience?
3. Why could not simpleFoam results a convergent solution for this problem?
4. How could you increase the Reynolds number? List 3 possibilities!
5. Which way could you improve the mesh? List 3 of them!

Homework

1. Create a refined O-grid (12 blocks) for the von Karman vortex street case, with
   - $L = 70$, $l = 20$, $d = 1$
   - Smaller cell expansion in the wake.
   - Fine surface mesh for the cylinder
2. Compare the results to the basic case