CFD using OpenFOAM
Lecture 8: Solution to Complex Geometry Problems in OpenFOAM

Instructor : Sumant R Morab (Ph.D Research Scholar)
Co-ordinator : Prof. Janani S Murallidharan
Indian Institute of Technology, Bombay

Outline

Recap of Lecture 7

Complex Geometry Flows

Flow over circular cylinder

Course Summary

- Mass Conservation (Continuity) : 1 Equation
- Momentum Conservation : U, V, and W

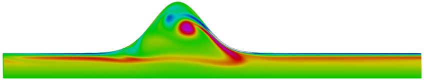
- Mass Conservation (Continuity) : 1 Equation
- Momentum Conservation : U, V, and W
- Finite Volume Discretisation :
- Challenges Faced (Decoupling, Non-linearity) & Solution Methodology

Recap: Flow Governing Laws, Discretisation & Solution Method

- Mass Conservation (Continuity) : 1 Equation
- Momentum Conservation : U, V, and W
- Finite Volume Discretisation :
- Challenges Faced (Decoupling, Non-linearity) & Solution Methodology
- Flow Behavior : Steady & Transient, Laminar & Turbulent
- Solvers : icoFoam, simpleFoam, pisoFoam etc.

- In real world scenarios, fluid flow occurs in complex geometries i.e, one which cannot be defined by cuboidal geometry.

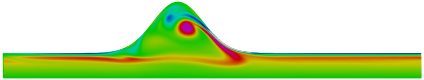
Blood Flow in Aneurysm



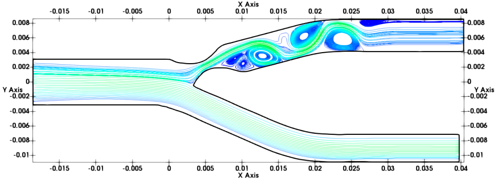
Flow in Complex Geometry

- In real world scenarios, fluid flow occurs in complex geometries i.e, one which cannot be defined by cuboidal geometry.

Blood Flow in Aneurysm



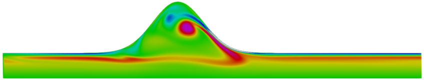
Blood Flow in Carotid Bifurcation



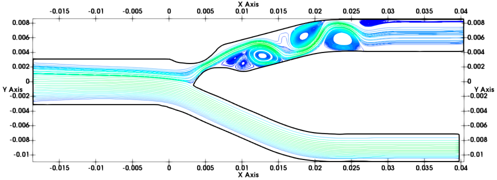
Flow in Complex Geometry

- In real world scenarios, fluid flow occurs in complex geometries i.e, one which cannot be defined by cuboidal geometry.

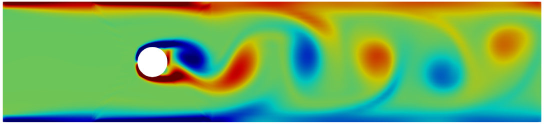
Blood Flow in Aneurysm



Blood Flow in Carotid Bifurcation



Vortex shedding behind circular cylinder



1. Geometry and Mesh generation. 2 options
   1.1 Creation of geometry using vertices, multiple-blocks and edge features in OpenFOAM and mesh using ‘Grading’ feature.

1. Geometry and Mesh generation. 2 options
   1.1 Creation of geometry using vertices, multiple-blocks and edge features in OpenFOAM and mesh using 'Grading' feature.
   1.2 Create geometry and mesh in external software and import it.

Steps for Solving Complex Geometry Problem

1. Geometry and Mesh generation. 2 options
   1.1 Creation of geometry using vertices, multiple-blocks and edge features in OpenFOAM and mesh using 'Grading' feature.
   1.2 Create geometry and mesh in external software and import it.
2. Implementation of boundary condition in '0/U' and '0/p' files.

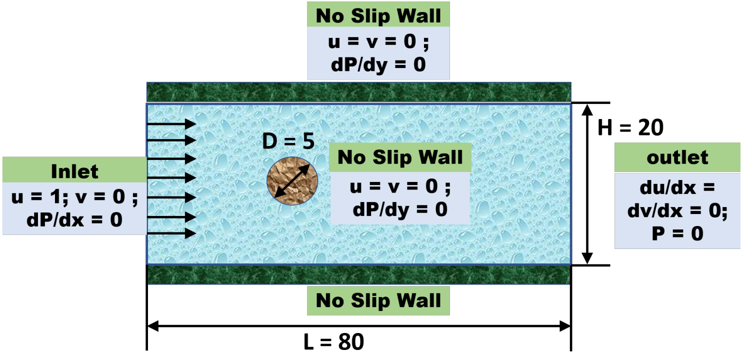
Steps for Solving Complex Geometry Problem

1. Geometry and Mesh generation. 2 options
   1.1 Creation of geometry using vertices, multiple-blocks and edge features in OpenFOAM and mesh using ‘Grading’ feature.
   1.2 Create geometry and mesh in external software and import it.
2. Implementation of boundary condition in ‘0/U’ and ‘0/p’ files.
3. Setting up flow parameters in ‘constant/transportProperties’ file.

1. Geometry and Mesh generation. 2 options
   1.1 Creation of geometry using vertices, multiple-blocks and edge features in OpenFOAM and mesh using 'Grading' feature.
   1.2 Create geometry and mesh in external software and import it.
2. Implementation of boundary condition in '0/U' and '0/p' files.
3. Setting up flow parameters in 'constant/transportProperties' file.
4. Setting up start and end time along with post-processing functions in 'system/controlDict' file.

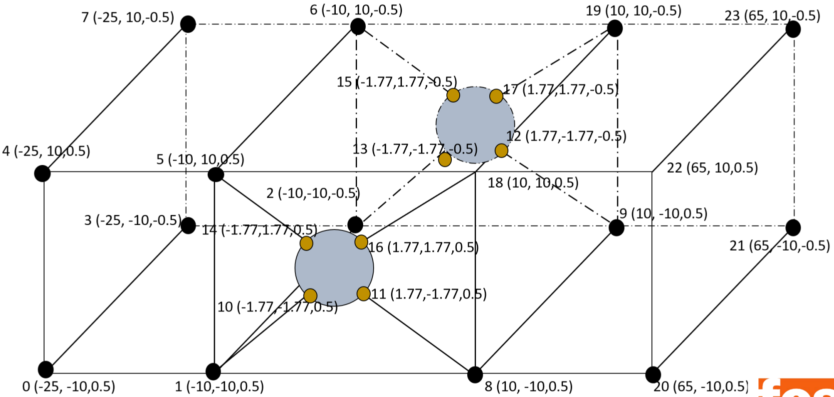
Illustrative Example : Flow over cylinder

- Consider flow over circular cylinder confined inside a channel as shown :



- A Cartesian grid cannot be employed using in-built discretisation scheme.

- The geometry is divided into 6 different blocks as shown (along with vertex coordinates)[2]:



- arcs and blocks are defined as below: (system/blockMeshDict)

```
edges
(
    arc 10 11 (   0 -2.5  0.5)
    arc 12 13 (   0 -2.5 -0.5)
    arc 14 10 (-2.5    0  0.5)
    arc 15 13 (-2.5    0 -0.5)
    arc 14 16 (   0  2.5  0.5)
    arc 15 17 (   0  2.5 -0.5)
    arc 16 11 ( 2.5    0  0.5)
    arc 17 12 ( 2.5    0 -0.5)
);
```

- arcs and blocks are defined as below: (system/blockMeshDict)

```
edges
(
    arc 10 11 (   0 -2.5  0.5)
    arc 12 13 (   0 -2.5 -0.5)
    arc 14 10 (-2.5    0  0.5)
    arc 15 13 (-2.5    0 -0.5)
    arc 14 16 (   0  2.5  0.5)
    arc 15 17 (   0  2.5 -0.5)
    arc 16 11 ( 2.5    0  0.5)
    arc 17 12 ( 2.5    0 -0.5)
);
```

```
blocks
(
    // before cylinder
    hex ( 0  1  2  3  4  5  6  7) ( 60 1 30) simpleGrading (1 1 1)

    // cylinder
    hex ( 1  8  9  2 10 11 12 13) (30 1 30) simpleGrading (1 1 1) // bottom
    hex ( 1 10 13  2  5 14 15  6) (30 1 30) simpleGrading (1 1 1) // left
    hex (14 16 17 15  5 18 19  6) (30 1 30) simpleGrading (1 1 1) // top
    hex (11  8  9 12 16 18 19 17) (30 1 30) simpleGrading (1 1 1) // right

    // after cylinder
    hex ( 8 20 21  9 18 22 23 19) (180 1 30) simpleGrading (1 1 1)
);
```

Go to ‘0/U’ and ‘0/P’

```
boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform (1.0 0 0);
    }
    outlet
    {
        type        zeroGradient;
    }
    wall
    {
        type        fixedValue;
        value       uniform (0 0 0);
    }
    obstacle
    {
        type        fixedValue;
        value       uniform (0 0 0);
    }
    frontAndBack
    {
        type        empty;
    }
}
```

Go to '0/U' and '0/P'

```
boundaryField
{
    inlet
    {
        type            fixedValue;
        value           uniform (1.0 0 0);
    }
    outlet
    {
        type        zeroGradient;
    }
    wall
    {
        type        fixedValue;
        value       uniform (0 0 0);
    }
    obstacle
    {
        type        fixedValue;
        value       uniform (0 0 0);
    }
    frontAndBack
    {
        type        empty;
    }
}
```

```
boundaryField
{
    inlet
    {
        type            zeroGradient;
    }
    outlet
    {
        type            fixedValue;
        value           uniform 0;
    }
    wall
    {
        type            zeroGradient;
    }
    obstacle
    {
        type            zeroGradient;
    }
    frontAndBack
    {
        type            empty;
    }
}
```

- The main objective is to calculate lift and drag co-efficients on cylinder.
- This is achieved by adding forceCeffs monitor in ‘system/controlDict’ file.

- The main objective is to calculate lift and drag co-efficients on cylinder.
- This is achieved by adding forceCeffs monitor in ‘system/controlDict’ file.

Setting-up post-processing parameters

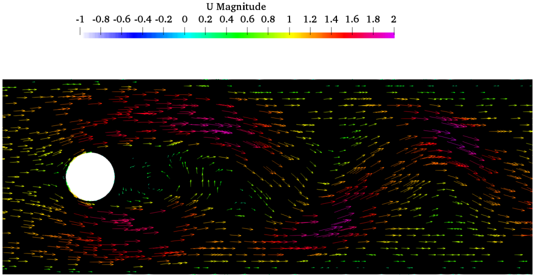
- The main objective is to calculate lift and drag co-efficients on cylinder.
- This is achieved by adding forceCeffs monitor in 'system/controlDict' file.

```
functions
{
    forceCoeffs
    {
    type          forceCoeffs;
    libs ( "libforces.so" );
    writeControl timeStep;
    writeInterval 1;

    patches       ( "obstacle" );
    pName         p;
    UName         U;
    rho           rhoInf;   // Indicates incompressible
    log           true;
    rhoInf        1;        // Redundant for incompressible
    liftDir       (0 1 0);  // Lift Direction
    dragDir       (1 0 0);  // Drag Direction
    CofR          (0.0 0.0 0.0);  // Axle midpoint on ground
    pitchAxis     (0 0 1);
    magUInf       1.0;  // Velocity
    lRef          5.0;     // Wheelbase length
    Aref          5.0;    // Cross section Area
    }
}
```

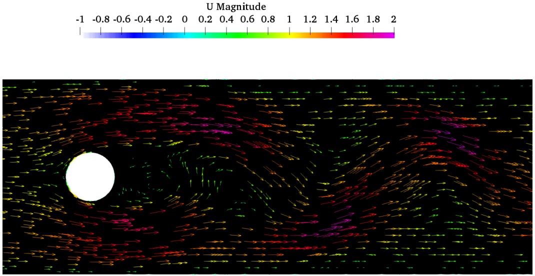
- After completing the set-up, run 'blockMesh' and 'icoFoam' to obtain the solution.

velocity vectors (2D glyphs in paraFoam)

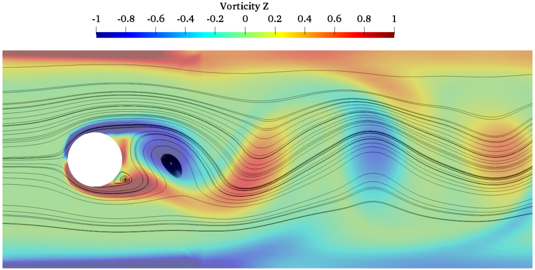


- After completing the set-up, run ‘blockMesh’ and ‘icoFoam’ to obtain the solution.

velocity vectors (2D glyphs in paraFoam)

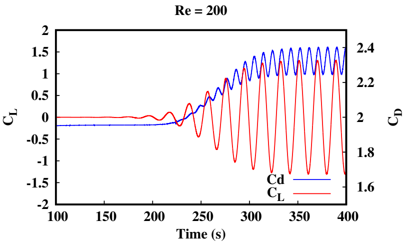


Streamlines (stream-tracer in paraFoam)



Variation of lift and drag

- The variation of lift and drag co-efficients on cylinder is as follows:



- Observations
  1. Periodic steady state is obtained after t = 240 units.
  2. Mean lift is zero while mean drag is around 2.

Summary of Lecture

1. Need for complex geometry implementation in OpenFOAM

Summary of Lecture

1. Need for complex geometry implementation in OpenFOAM
2. Steps involved for pre-processing and post-processing.

Summary of Lecture

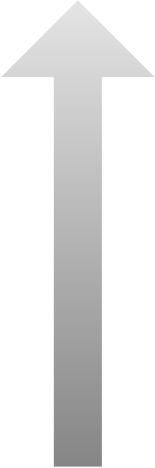
1. Need for complex geometry implementation in OpenFOAM
2. Steps involved for pre-processing and post-processing.
3. Test Case : Flow over circular cylinder

1. Need for complex geometry implementation in OpenFOAM
2. Steps involved for pre-processing and post-processing.
3. Test Case : Flow over circular cylinder
4. post-processing through controlDict and Visualization of results

1. Need for complex geometry implementation in OpenFOAM
2. Steps involved for pre-processing and post-processing.
3. Test Case : Flow over circular cylinder
4. post-processing through controlDict and Visualization of results

This completes the short course on CFD using OpenFOAM. For more complex cases and tutorials, one may refer `https://cfd.fossee.in/home` .

Summary of Course



Introduction to CFD & OpenFOAM as CFD tool : What is CFD → Why to study → importance of OpenFOAM

Governing Laws in CFD and Discretisation

Introduction to CFD & OpenFOAM as CFD tool : What is CFD $\rightarrow$ Why to study $\rightarrow$ importance of OpenFOAM

Summary of Course

OpenFOAM solution Methodology → various solvers and their capability

Governing Laws in CFD and Discretisation

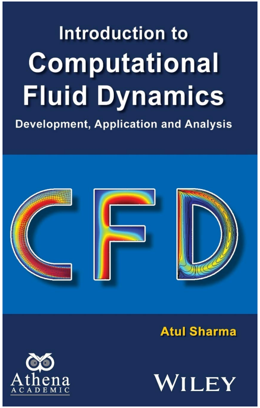
Introduction to CFD & OpenFOAM as CFD tool : What is CFD → Why to study → importance of OpenFOAM

Summary of Course

- Illustrative examples for simple & complex geometry
- OpenFOAM solution Methodology → various solvers and their capability
- Governing Laws in CFD and Discretisation
- Introduction to CFD & OpenFOAM as CFD tool : What is CFD → Why to study → importance of OpenFOAM

References

1. Sharma, A. (2016). Introduction to computational fluid dynamics: development, application and analysis. John Wiley & Sons.
2. https://github.com/AsmaaHADANE/Youtube-Tutorials/blob/master/flow_around_cylinder.zip
3. https://www.openfoam.com/



Thank you for listening!

Sumant R Morab