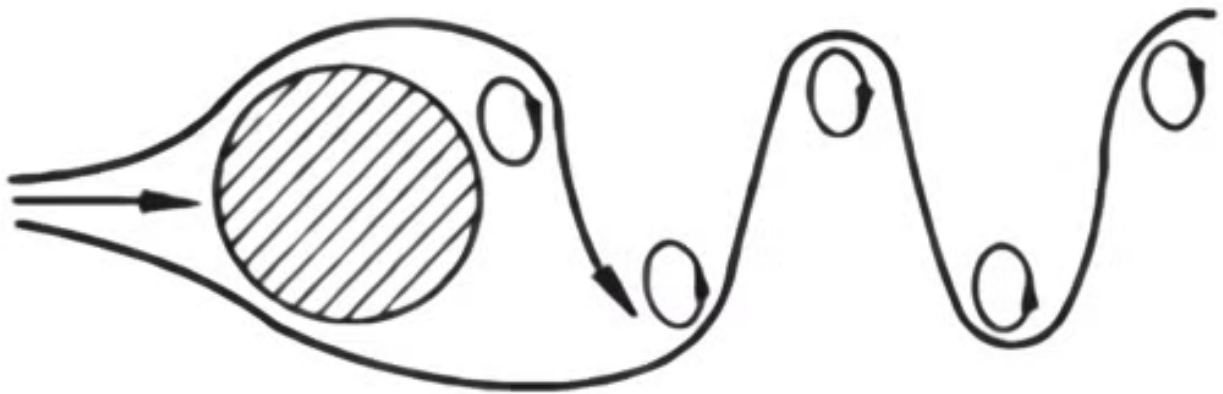

MM2041: Transport Phenomena in Materials

Assignment

VORTEX SHEDDING

Explanation & OpenFoam Simulation



KRISHNARJUN J

MM23B048

Dept. Metallurgical and Materials
Engg.

Indian Institute of Technology Madras

Table of Contents

Contents

1	Introduction	3
2	Objective of the simulation	4
3	Simulation Details and Setup in OpenFoam	4
4	Code Structure	5
4.1	0:	5
4.1.1	0/U:	5
4.1.2	0/p:	6
4.2	constant:	7
4.2.1	constant/transportProperties:	7
4.2.2	constant/turbulenceProperties:	8
4.2.3	constant/Polymesh:	8
4.3	system:	9
4.3.1	system/blockMeshDict:	9
4.3.2	system/controlDict:	9
4.3.3	system/fvSchemes:	10
4.3.4	system/fvSolution:	11
5	Results - Flow visualization	12
5.1	Velocity Magnitude Contours	12
5.2	Pressure Magnitude Contours	12
6	Effect of Reynolds Number on Vortex Shedding	13
7	Conclusion	15
8	References	15

1 Introduction

Vortex shedding is a fluid dynamic effect that takes place when a fluid like air or water moves against an object with a wide, flat front instead of a streamlined front. At these velocities, the flow becomes unstable and breaks away from both sides, of the body. This creates a periodic pattern of swirling vortices in the object's wake or back-side. This other pattern is called Kármán vortex street.

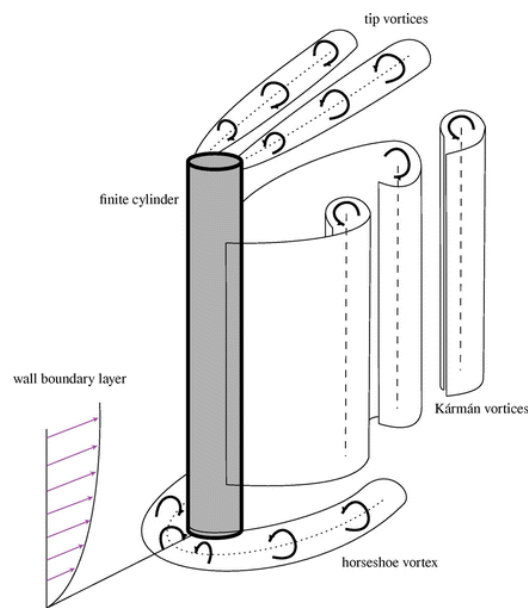


Figure 1: Vortex Shedding around a cylinder

When vortices break away, they form alternating low-pressure areas at the back of the object. This produces a pressure gradient, that can result in periodic lateral movement of the object. If the object is poorly mounted and the vortex shedding frequency is equal to the object's resonance frequency, it can result in intense vibrations.

Some actual effects resulting from vortex shedding include oscillations in tall buildings like chimneys and car antennas. To prevent this, spiral wings are fitted at the top.



Figure 2: Chimney with spiral wings at top

2 Objective of the simulation

To simulate laminar vortex shedding over a 2D cylinder in OpenFOAM, we simulate a stream flow over the cylinder. This flow creates the periodic vortices behind the cylinder, creating a pattern called the von Kármán Vortex Street. We also visualized velocity contours (U) and pressure contours (p) to comprehend the flow pattern around the cylinder. The simulation was also conducted over different Reynolds numbers (Re) to examine the influence of inertial and viscous forces on vortex shedding.

3 Simulation Details and Setup in OpenFoam

- Geometry - 2D flow past a cylinder placed in a channel. Made with three blocks - inlet, outlet and obstacle (which is the cylinder in this case). With the help of available OpenFoam tutorials.
- Mesh generation - Generated using blockMesh
- Physical Parameters:
 - Cylinder diameter = 0.05m
 - Fluid used is water
 - viscosity = $1e-6$
 - $U = 0.05$ m/s
 - Reynolds number = $U.L/(\text{viscosity}) = 2500$
- Solver used - simpleFoam
- Time Step - deltaT 1;
- Duration - endTime 10000;
- Turbulence Model - simulationType laminar;

4 Code Structure

This simulation is configured in a typical OpenFOAM case directory with the typical 0/, constant/, and system/ directories. Subsequently, these files also include the initial condition of the simulation, physical model, and other numerical information.

```
krish@Krishnarjun:~/MM2041_Vortex_Shedding/1$ tree
.
├── 0
│   ├── U
│   └── p
├── constant
│   ├── polyMesh
│   │   ├── boundary
│   │   ├── faces.gz
│   │   ├── neighbour.gz
│   │   ├── owner.gz
│   │   └── points.gz
│   ├── transportProperties
│   └── turbulenceProperties
└── system
    ├── blockMeshDict
    ├── controlDict
    ├── fvSchemes
    └── fvSolution

4 directories, 13 files
```

Now lets look into each file one by one.

4.1 0:

Stores the starting conditions for **velocity** (U), **pressure** (p), and other fields at simulation initiation (time = 0).

```
krish@Krishnarjun:~/MM2041_Vortex_Shedding/1/0$ tree
.
├── U
└── p

0 directories, 2 files
```

4.1.1 0/U:

This file's job is to set the initial and boundary conditions for velocity field (U), a vector field (volVectorField) in units of m/s.

```

// ****
dimensions      [0 1 -1 0 0 0]; // [Mass
internalField   uniform (0 0 0);
boundaryField
{
    inlet
    {
        type      fixedValue;
        value      uniform (0.054 0 0);
    }
    outlet
    {
        type      zeroGradient;
    }
    wall
    {
        type      fixedValue;
        value      uniform (0 0 0);
    }
    obstacle
    {
        type      fixedValue;
        value      uniform (0 0 0);
    }
    frontAndBack
    {
        type      empty;
    }
}
// ****

```

- **internalField:** Specifies the initial velocity within the domain as zero.
- **inlet:** Describes a steady velocity of (0.05 0 0) m/s (x-axis direction).
- **outlet:** Uses zeroGradient, so that flow can exit naturally.
- **wall and obstacle:** Both are defined as no-slip condition (velocity = 0).
- **frontAndBack:** Identified as empty for a 2D simulation.

4.1.2 0/p:

This file's job is to set the initial and boundary conditions for pressure (p), a scalar field (volScalarField) with units of [Pa].

```

// ****
dimensions      [0 2 -2 0 0 0];
internalField   uniform 0;
boundaryField
{
    inlet
    {
        type      zeroGradient;
    }
    outlet
    {
        type      fixedValue;
        value      uniform 0;
    }
    wall
    {
        type      zeroGradient;
    }
    obstacle
    {
        type      zeroGradient;
    }
    frontAndBack
    {
        type      empty;
    }
}
// ****

```

- **internalField:** Specifies pressure as constant zero across the domain.
- **inlet:** zeroGradient allows natural pressure development as flow enters
- **outlet:** fixedValue specifies pressure as 0 Pa, to serve as a reference.
- **wall and obstacle:** zeroGradient indicates no change in pressure normal to surfaces. Known to be empty for a 2D simulation.
- **frontAndBack:** Recognized as vacant for a 2D simulation.

4.2 constant:

Stores fluid properties, mesh definition, and turbulence models. The most important files are **transportProperties** (fluid viscosity), **turbulenceProperties** (turbulence model), and **polyMesh** (mesh definition).

```
krish@Krishnarjun:~/MM2041_Vortex_Shedding/1/constant$ tree
.
├── polyMesh
│   ├── boundary
│   ├── faces.gz
│   ├── neighbour.gz
│   ├── owner.gz
│   └── points.gz
├── transportProperties
└── turbulenceProperties
1 directory, 7 files
```

4.2.1 constant/transportProperties:

This file specifies the fluid's kinematic viscosity and transport model.

```
// * * * * *
transportModel  Newtonian;

nu              nu [0 2 -1 0 0 0 0] 1.5e-5;

// * * * * *
```

- **transportModel Newtonian;** States that the fluid obeys Newton's law of viscosity (i.e., linear dependence between shear stress and velocity gradient).
- **nu [0 2 -1 0 0 0 0] 1.5e-5;** Defines kinematic viscosity nu as a property of air at room temperature.

Used to calculate the Reynolds number and specify the flow regime.

4.2.2 **constant/turbulenceProperties:**

This file specifies the turbulence model used in the simulation.

```
// * * * * *
simulationType laminar;

// * * * * *
```

- **simulationType laminar;** Specifies that the flow is laminar, i.e., there is no turbulence model. The flow is smooth, and no turbulent fluctuation is present.

4.2.3 **constant/Polymesh:**

The polyMesh directory contains the critical files defining the computational mesh of the simulation. These files describe how the domain is divided into cells.

```
krish@Krishnarjun:~/MM2041_Vortex_Shedding/1/constant/polyMesh$ tree
.
├── boundary
├── faces.gz
├── neighbour.gz
├── owner.gz
└── points.gz

0 directories, 5 files
```

- **boundary:** Defines the boundaries of the simulation domain (e.g., inlet, outlet, walls).
- **faces.gz, neighbour.gz, owner.gz, points.gz:** Save the mesh data, for example, connectivity (faces and neighbors), ownership of cells, and mesh points coordinates.
- These files ensure that the solver knows how to deal with the mesh during simulation and are crucial for precise data exchange between cells.
- The **.gz mesh files** (such as points.gz, faces.gz, owner.gz, and neighbour.gz) were produced by the textbfblockMesh utility, which creates a structured mesh from the geometry and settings in the **blockMeshDict** file.

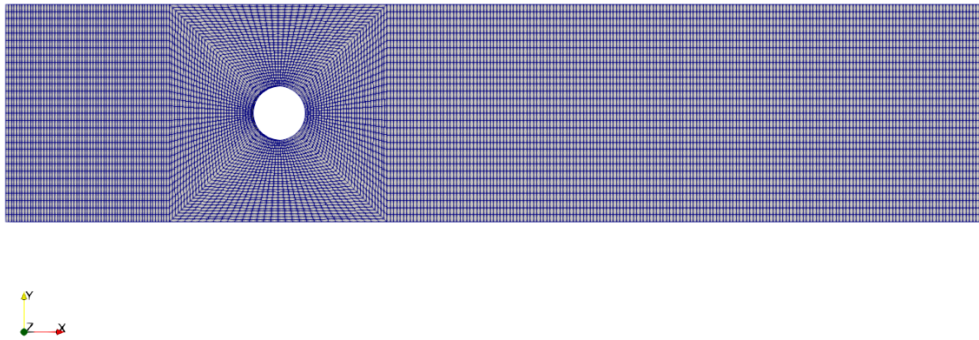
4.3 system:

Contains simulation running control files, e.g., controlDict (simulation time parameters), fvSchemes (discretization schemes), and fvSolution (solver parameters).

```
krish@Krishnarjun:~/MM2041_Vortex_Shedding/1/system$ tree
.
├── blockMeshDict
├── controlDict
├── fvSchemes
└── fvSolution
0 directories, 4 files
```

4.3.1 system/blockMeshDict:

The mesh used to simulate the vortex shedding was created using OpenFOAM's blockMesh utility. The mesh was defined in the blockMeshDict file.



After defining the mesh, the **blockMesh** command was executed to generate the mesh files, which were saved in the **constant/polyMesh** folder in compressed.gz format (e.g., points.gz, faces.gz). These are the mesh structure for the simulation.

4.3.2 system/controlDict:

controlDict is employed in order to establish control parameters of the simulation by using the simpleFoam solver.

Major options are:

- **Application:** The used solver is simpleFoam, which is steady-state incompressible flow simulation specific.
- **Start Time:** The starting time for the simulation is **0**.
- **End Time:** The ending time for the simulation is **10000**.

- **Time Step:** The simulation increments by **1 time step** at each increment.
- **Output Settings:**
 - Results are stored every **100 time steps** (writeInterval).
 - Output is saved in ASCII format with six decimal places precision (writePrecision 6).
 - Data is saved in compressed format (writeCompression compressed).

Modifiable Runtime: The runTimeModifiable is yes, enabling changes during the simulation run.

```
// *****
application    simpleFoam;
startFrom      startTime;
startTime      0;
stopAt         endTime;
endTime        10000;
deltaT         1;
writeControl    timeStep;
writeInterval   100;
purgeWrite     0;
writeFormat     ascii;
writePrecision  6;
writeCompression compressed;
timeFormat      general;
timePrecision   6;
runTimeModifiable yes;
// *****
```

4.3.3 system/fvSchemes:

The fvSchemes file determines the numerical discretization schemes employed for the different terms within the simulation.

- **Time Derivative Schemes (ddtSchemes):**
 - The steadyState scheme is employed, i.e., no time dependence and steady flow.
- **Gradient Schemes (gradSchemes):**
 - The Gauss linear scheme is employed, giving a plain first-order gradient calculation.
- **Divergence Schemes (divSchemes):**

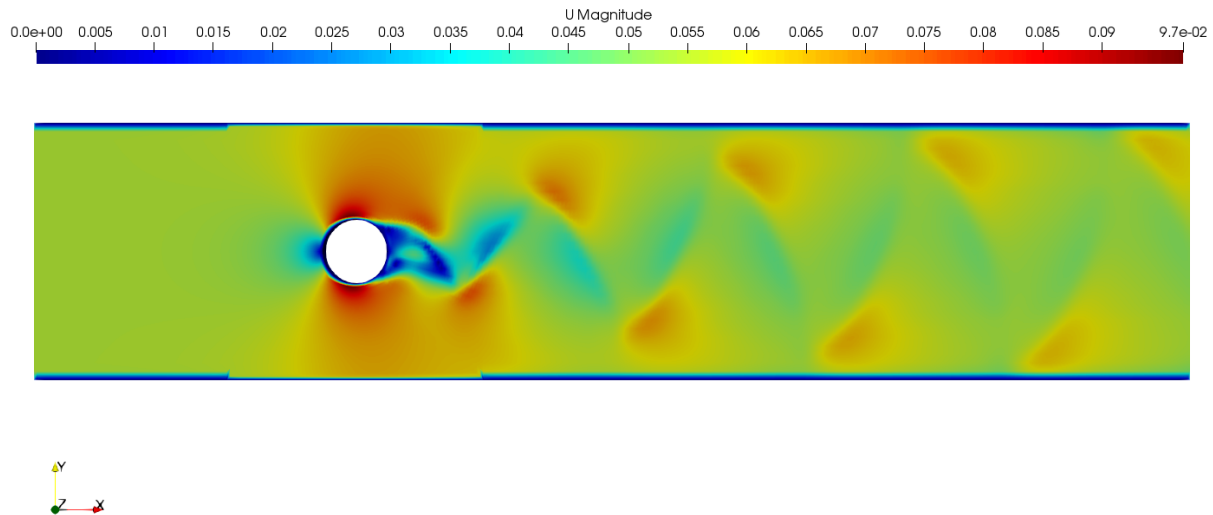
- For $\text{div}(\phi, U)$, the bounded Gauss limitedLinearV 1 scheme is employed for stable velocity calculations.
- For $\text{div}(\phi, k)$, $\text{div}(\phi, \epsilon)$, and alike terms, the bounded Gauss limitedLinear scheme is employed for turbulence and viscosity terms.
- **Laplacian Schemes (laplacianSchemes):**
 - Gauss linear corrected scheme is used for more precise Laplacian computations.
- **Interpolation Schemes (interpolationSchemes):**
 - A linear scheme is used for basic interpolation between grid cells.
- **Surface Normal Gradient Schemes (snGradSchemes):**
 - A corrected scheme is applied for improved normal gradient calculations.

4.3.4 **system/fvSolution:**

The fvSolution file sets solver options and algorithm controls. It uses the **GAMG solver** with Gauss-Seidel smoothing for the pressure, velocity, and turbulence variables. Pressure-velocity coupling is controlled by the **SIMPLEC algorithm** with 1 non-orthogonal correction. Relaxation factors are applied to stabilize convergence.

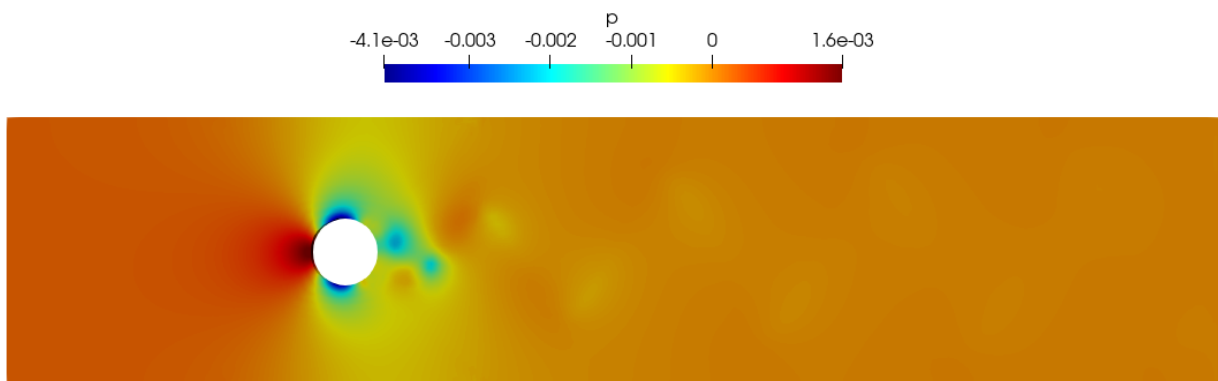
5 Results - Flow visualization

5.1 Velocity Magnitude Contours



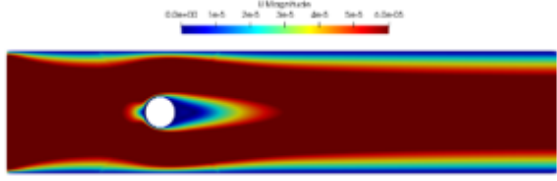
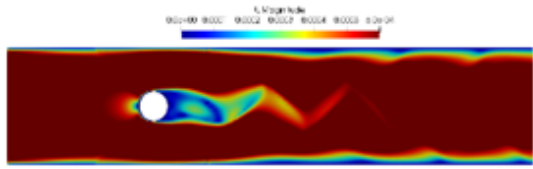
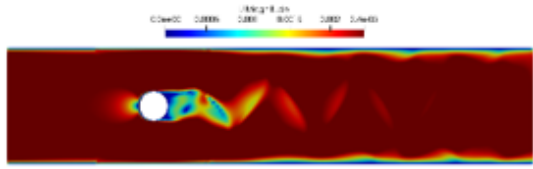
- The velocity field around the cylinder was visualized through color contours of the magnitude of velocity (UMagnitude).
- The flow is detached from the cylinder surface, creating periodic vortices in the downstream direction — a typical Kármán vortex street.
- High-velocity areas are observed close to the cylinder sides as a result of acceleration of the flow, whereas low-velocity areas are seen in the wake.

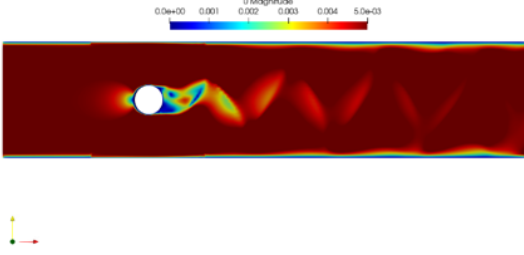
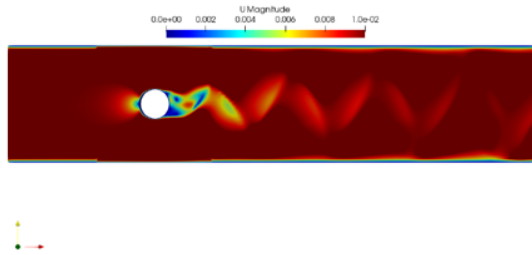
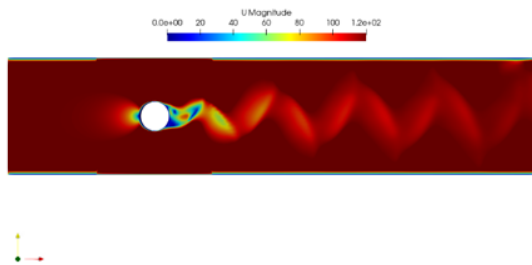
5.2 Pressure Magnitude Contours



- Color contours were used to visualize the pressure field about the cylinder.
- Low-pressure can easily be observed behind the cylinder from flow separation.
- The stagnation point in the front leads to the development of the high-pressure zone as flow directly hits the cylinder there.
- These differences in pressure give rise to the drag force of the cylinder.

6 Effect of Reynolds Number on Vortex Shedding

	$Re < 5$	Regime of unseparated flow
	$5 < Re < 40$	A fixed pair of vortices in the wake
	$90 < Re < 150$	Two regimes in which vortex street is laminar periodicity governed in high Re Range by vortex shedding

	$150 < Re < 300$	Transition Range to turbulence in vortex
	$300 < Re < 3 \times 10^5$	Vortex Street is fully turbulent
	$3 \times 10^5 < Re < 3.5 \times 10^6$	Laminar boundary layer has undergone turbulent transition. The wake is narrower and disorganized.

The vortex shedding behavior behind the cylinder is highly sensitive to the Reynolds number (Re). In order to examine this, the simulations were done at various Reynolds numbers, and the resulting flow patterns, the vortex structures, and the shedding frequencies were measured.

This simulation is run for various ranges of Reynolds number by changing the U value in $0/U$ file, which is in the x - direction. Then the solver can be used and observed using paraFoam.

7 Conclusion

In this assignment, I was able to simulate laminar vortex shedding behind a two-dimensional cylinder with OpenFOAM. The formation of the von Kármán Vortex Street was clearly observed, as well as the periodic alternation of vortices in the wake region. By changing the Reynolds number, we saw how the flow behavior varies, such as the onset, intensity, and frequency of vortex shedding. Contour plots of velocity (U) and pressure (p) gave additional insight into the flow behavior around the cylinder. In general, the simulation results agreed well with the anticipated physical behavior of vortex shedding, showing the ability of OpenFOAM to simulate complex flow even using relatively straightforward discretization schemes.

8 References

- Cover Page image - Computational Fluid Dynamics and Turbulence Modeling Laboratory - University of Waterloo link
- Figure 1 - Research Gate link
- Figure 2 - Physics Stack Exchange link
- B. Mutlu Sumer, Jørgen Fredsøe. Hydrodynamics Around Cylindrical Structures. World Scientific.
- Frank M. White. Viscous Fluid Flow. McGraw-Hill, 4th Edition, 2016
- Mesh generation - OpenFoam Official Tutorials link
- OpenFoam documentation link
- Variation of reynolds number link
- AI was only used for consulting help with setup simulation, \LaTeX documentation, and report generation and no direct use.