Studying the Viscous Flow Around a Cylinder Using OpenFoam

Marc Kornbleuth

Astronomy Department, Boston University, Boston, MA 02215

kmarc@bu.edu

Received ______________; accepted ______________

ABSTRACT

We model the incompressible, viscous flow of a fluid around a cylinder using the *OpenFoam* toolbox. We develop a mesh with varying resolutions to properly capture the flow around the cylinder. We run our simulations for varying Reynolds number (R) cases between $32 \leq R \leq 161$, and see an increase in shedding within the flow for increasing Reynolds number. We find decent qualitative agreement with observations, though quantitatively the simulated Strouhal number is not in good agreement with the expected Strouhal number. Given the conditions of our simulations, *OpenFoam* is able to accurately model the change in Strouhal number as Reynolds number changes. Additionally, *OpenFoam* is confirmed to follow the law of similarity.

1. Introduction[1]

Viscous flow around a cylinder has been an oft studied topic in the fluid community. One of the first well-known cases of studying flow around a cylinder was carried out by Hiemenz (1911), who took quantitative measurements of the pressure around a cylinder. Later, Homann (1936) used oil to visualize the fluid flow around cylinders. Tests of flow around a cylinder, such as via simulations, have piqued the imagination of many scientists and engineers, and thus have continued to this day.

An incompressible, viscous fluid is used to model the flow around a cylinder. The equations for an incompressible viscous fluid are governed by

---

[1]Introduction adopted from Secs. 5.4 and 5.5 of "The Physics of Fluids and Plasma: An Introduction for Astrophysicists" by Arnab Rai Choudhuri (1998).

$$\omega = \nabla \times \mathbf{v}, \tag{1}$$

$$\nabla \cdot \mathbf{v} = 0, \tag{2}$$

$$\frac{\partial \omega}{\partial t} = \nabla \times (\mathbf{v} \times \omega) + \nu \nabla^2 \omega, \tag{3}$$

where eq. 1 is the vorticity, eq. 2 is the condition of incompressibility, and eq. 3 is Navier-Stokes with vorticity. Here, $\mathbf{v}$ is the velocity of the fluid and ν is the kinematic viscosity of the fluid. For flow around a geometric object, with unit length L and unit fluid velocity V, we can rewrite the variables of eq. 3 in scaled units as

$$\mathbf{x} = \mathbf{x}' L, \mathbf{v} = \mathbf{v}' V, t = t' \frac{L}{V}, \omega = \omega' \frac{V}{L}. \tag{4}$$

With these scaled variables, eq. 3 can be re-written as

$$\frac{\partial \omega'}{\partial t'} = \nabla' \times (\mathbf{v}' \times \omega') + \frac{1}{R} \nabla'^2 \omega', \tag{5}$$

where $\nabla' = L\nabla$ is the gradient in space for the scaled length and R is the Reynolds number, given by

$$R = \frac{LV}{\nu}. \tag{6}$$

If the Reynolds number were the same for two cases with different flows around similarly shaped objects, then the flow patterns should be the same (assuming the boundary

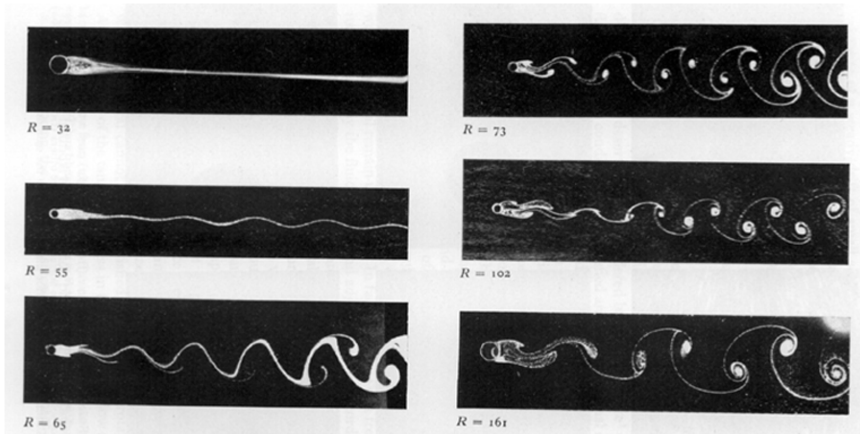

Fig. 1.—: Viscous flow around a cylinder for high Reynolds number. Originally from Homann (1936), but reproduced from Batchelor (1967).

conditions are identical). Based on this observation, Reynolds (1883) developed the law of similarity.

Stokes (1851) studied the effect of low Reynolds number. For the case that $R << 1$, eq. 4 reduces to

$$\frac{1}{R}\nabla'^2\omega' = 0. \tag{7}$$

A result of this analysis was that Stokes found the drag force on a sphere moving through a viscous fluid. The drag force for this low Reynolds number case is

$$F_D = 6\pi\mu a U, \tag{8}$$

where a is the radius of the sphere, μ is the coefficient of viscosity and U is the velocity of the sphere through the fluid. This force acts in the direction opposite to the motion of the sphere, and it is known as Stokes's Law. However, this law is not applicable in the high Reynolds number situation.

For high Reynolds number, where $R >> 1$, the drag force changes. In this case, Stokes's Law is given by

$$F_D = \frac{1}{2} C_D \pi a^2 \rho U^2, \tag{9}$$

where C_D is a dimensionless drag coefficient defined as

$$C_D = \frac{12\mu}{a\rho U} \frac{12}{R}. \tag{10}$$

Thus, in this case, the coefficient will decrease with increasing Reynolds number. Figure 1 shows the flow past a cylinder for different Reynolds numbers. This figure, originally taken from Homann (1936), but reproduced from Batchelor (1967), shows the appearance of vortices behind the cylinder with high Reynolds number. Once the Reynolds number reaches around $R = 60$, the vortices shed into a string of vortices behind the cylinder. The properties of this string was studied by von Karman (1911), and was named the Karman vortex sheet. Not shown in Fig. 1 is that for Reynolds number on the order of 10^4, a turbulent wake will develop behind the cylinder.

Another element of the viscous flow around an object, like a sphere or cylinder, is that $\mathbf{v} = 0$ at the solid boundary. In order to allow for this boundary condition, a boundary layer must exist where the velocity of the flow can quickly go to zero at the surface of the object. Within this layer, $\nabla^2 \mathbf{v}$ must be non-trivial, thus the viscosity term in the Navier-Stokes equation must be important. Regardless of the Reynolds number, viscosity plays a significant role in the layers next to a solid within a fluid. The boundary layer thickness can be approximated as

$$\delta = \sqrt{\frac{\nu L}{V}}. \tag{11}$$

From this equation, we can see that the as the fluid moves further down the length of the object, the boundary layer thickness will increase as the square root of the distance. Since viscosity is not negligible in the boundary layer, vortices in the flow can be generated downstream if none existed upstream. For large Reynolds numbers, strong velocity shears can develop, which produce turbulence. Thus, large Reynolds numbers can generate an unstable flow.

One way to model the flow around a cylinder is via computational fluid dynamics (CFD) modelers such as *OpenFoam* (Weller et al. 1998). *OpenFoam* is an open-source CFD toolbox, developed in 2004, that is able to solve complex fluid flows in systems such as through a cavity or around a cylinder. One particular solver in *OpenFoam* is called *icoFoam*. The *icoFoam* solver uses the incompressible Navier-Stokes equations and solves the equations via the Pressure Implicit with Splitting of Operator (PISO) algorithm. It is the *icoFoam* solver with the PISO algorithm which is ideal for simulating incompressible flow around a cylinder. While eqs. 1-3 are ideal for modeling the incompressible, viscous flow around cylinder, instead of eq. 3 *OpenFoam* solves the following form of the momentum equation

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v} = -\frac{1}{\rho}\nabla p + \frac{\mu}{\rho}\nabla^2\mathbf{v}. \tag{12}$$

The main difference between eq. 3 and eq. 12 is that *OpenFoam* uses the standard momentum equation instead of using the momentum equation for vorticity. Thus, *OpenFoam* does not use eq. 1 either when it solves the flow around a cylinder.

The remaining sections of this paper are as follows: in Sec. 2 we describe the mesh used in our simulation. In Sec. 3 we describe the properties of the model, including the details of the different directories and how to properly setup the physics. In Sec. 4 we present the results of our simulations, and compare these results to observations. In Sec. 5

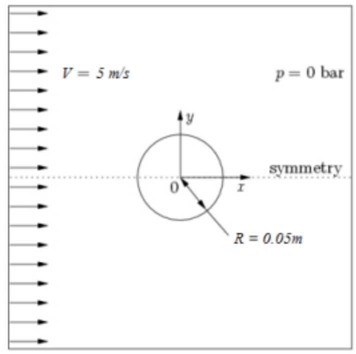

Fig. 2.—: Basic cartoon of the flow around a cylinder.

we summarize our conclusions.

2. Mesh[2]

To set up the problem of incompressible viscous flow around a cylinder, we use Fig. 2 as a reference for the basic setup. Like Fig. 2, we want a cylinder with $r = 0.05$ m, with the vertical of the cylinder in the z-direction. Additionally, we wish to have inflow moving in the +x direction from the left boundary (or inlet). Based on the physics and the scale of the problem, such as the size of the boundary layer around the cylinder, certain parts of the domain require higher resolution than others. While in theory the best way to solve the problem would be to create a mesh with equally sized grid points set to the smallest required scale, for the sake of computational simplicity and quickness we use a varied mesh. This mesh is based off of Fig. 3, where we increase our resolution around the cylinder to ensure the physics around the boundary layer is properly captured.

In *OpenFoam*, we must model the cylinder in three dimensions despite only wanting

[2]Adopted from Sec. 3.4.1 of "OpenFOAM Guide for Beginners" by Jordi Casacuberta Puig

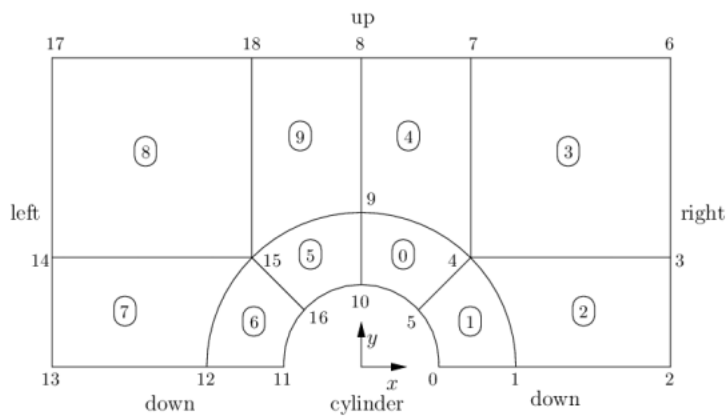

Fig. 3.—: Structure of the mesh around half of the cylinder.

a two-dimensional flow. To work around this, we set the z-direction to be one grid cell thick, so we are essentially modeling in two dimensions. We use *blockmeshDict* in */constant/polyMesh* for our mesh creation. In our mesh, we define six boundaries: (1) top, (2) bottom, (3) inlet, (4) outlet, (5) cylinder, and (6) the front and back of the domain. To develop the mesh around the cylinder, we use the *arc* instruction for a circular geometry. Additionally, we define the top and bottom boundaries via *symmetryplane* so no rigid walls exist to alter the solution. We define the cylinder via *wall*, the front and back as *empty*, and the inlet and outlet as *patch*. The *wall* instruction sets up a physical boundary, hence why it is used for the cylinder. The *empty* instruction is utilized for the front and back planes of a two-dimensional geometry, which is what we have for the front and back boundaries. Lastly, the *patch* instruction denotes a patch where there is no geometric information about the mesh, and therefore we use this for the inlet and outlet so the fluid can flow through the boundary.

The version of *blockmeshDict* used can be found at the end of this report in the appendix. Using this mesh description, we get the meshes seen in Fig. 4. On the left of Fig. 4 we have the top down view of the mesh. The fluid moves from the -x direction to the +x direction. We note the high degree of resolution around the cylinder. Additionally,

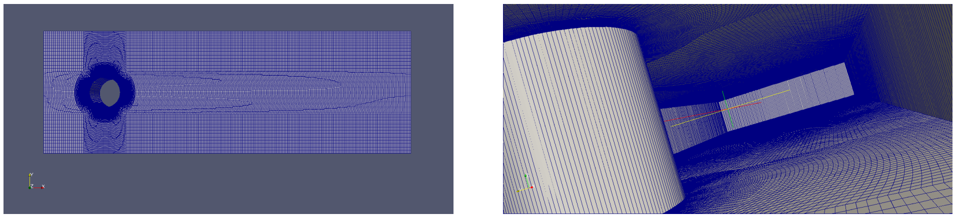

Fig. 4.—: Left: Top-down view of the mesh. Right: Inside the mesh, looking in the +x direction.

the resolution is lower along the edges of domain (except in the region near the cylinder), as compared to the mesh directly behind the cylinder. This enables better resolution of the flow as it moves around the cylinder, but it does restrict the ability to see smaller fluid elements near the edges. Regardless, as we are pursuing a simplified model that is not computationally expensive, we retain this grid setup for the simulation. Additionally, on the right of Fig. 4 we have the interior to the three-dimensional box that is one grid cell in height. This is evident by the grid on the cylinder, which is on the left in this figure. This perspective is from within the box, looking in the +x direction from the -y direction. It is possible to discern the changing grid structure from this particular figure.

3. Model[3]

The basics of the model are located in the following three directories: 1) 0, 2) *constant*, and 3) *system*. We will discuss the model files located in these directories. The exact versions of the files used can be found at the end of this document in the appendix.

[3]Adopted from Secs. 3.4.2-3.4.5 of "OpenFOAM Guide for Beginners" by Jordi Casacuberta Puig

3.1. 0 directory

In the 0 directory, there exist two files of note: p and U. These two files define the initial conditions for the fluid flow problem. In p we give the initial pressure conditions for the system. Our $internalField$ is set to be uniformly zero. Our boundaries have the conditions that the top and bottom boundaries are defined via *symmetryPlane*, the front and back is defined as an *empty* boundary, the cylinder boundary has a *zero* setting, and the inlet and outlet boundaries are defined via a $freestreamPressure$. The $freestreamPressure$ condition sets a fixed pressure value for ingoing flow, but a zero gradient at the outgoing boundary, so this is useful for external flow.

The U file gives the initial velocity conditions for the system. The $internalField$ is set to have a uniform velocity in the +x direction. The inlet and outlet boundaries also maintain a $freestream$ condition of a uniform velocity in the +x direction, with the same velocity as that in the $internalField$. The different values for the velocity are given in Table 1. We used 5 m/s for all cases except for the first and last ones. For the first one, $OpenFoam$ was unable to handle a high viscosity in the simulation, so we used a low viscosity and changed the fluid velocity. In the second case of $R = 161$ we wished to test the law of similarity so we used a different velocity and viscosity from case six. The top and bottom bottom boundaries are given the *symmtetryPlane* condition. The front and back boundaries are given the *empty* condition as per usual. Lastly, the cylinder is given a $fixedValue$ condition of have zero velocity at the cylinder boundary, hence creating a boundary layer.

$Case$	R	U $[km/s]$	ν $[m^2/s]$
1	32	0.82	0.002564
2	55	5.00	0.009091
3	65	5.00	0.007692
4	73	5.00	0.006849
5	102	5.00	0.004902
6	161	5.00	0.003106
7	161	4.13	0.002564

Table 1:: Reynolds number cases run in *OpenFoam* for flow around a cylinder with $r = 0.05$ m.

3.2. *constant* directory

The *constant* directory contains the files which define the kinematic viscosity and the Reynolds-averaged Simulation (RAS) model. The first flag in the RAS file, called $RASProperties$, is the RAS Model (denoted by $RASModel$), which calls the RAS turbulence model. In this case we are using laminar flow, so we set that flag to $laminar$. Additionally, for the $turbulence$ and $printCoeffs$ flags, we set both to off. In the viscosity file, called $transportProperties$, we set the value of nu (aka the kinematic viscosity) to our desired value. Table 1 shows the different values of ν we used for each case in order to vary the Reynolds number.

3.3. *system* directory

The *system* directory contains the files which tell the code how to run. We find the *controlDict* file which tells the code how long to run, with what time step to run, and how to write the data. We also find $fvSchemes$, which tells the code the schemes to use, and $fvSolutions$, which tells the code what solvers to use.

The important aspects of *controlDict* are as follows. The solver being used for flow around a cylinder is *icoFoam*. We set the start time to be at 0, since this is where our initial conditions are stored. For the purposes of this project, we use an end time of 2.60 for all of our cases, with a *deltaT* of 0.00001 and a *writeInterval* of 1000.

For $fvSchemes$, it is mostly the same as for the cavity simulation, but instead under *laplacianSchemes* we have the $default$ set to *none*, and we also have $laplacian(nu, U)$ as *Gausslinearorthogonal*, and likewise for $laplacian((1|A(U)), p)$. For $fvSolutions$, there are two main differences from the cavity simulation. First, the U solver is $PBiCG$, with a preconditioner of $DILU$, a *tolerance* of $1e-05$ and a *relTol* of 0. Additionally, under $PISO$, we have *nNonOrthogonalCorrectors* set as 3 instead of 0 (as it is in the cavity case), because this helps in getting more physically accurate results.

4. Results

The results of our *OpenFoam* simulations are shown in Fig. 5. Figure 5 shows a comparison between the observed flow around a cylinder originally done by Homann (1936), but then reproduced in Batchelor (1967), and the simulated flow from *OpenFoam*. For these comparisons, we display cases one through six, and decent agreement is observed. For the $R = 32$ case, there is a lack of shedding as the fluid propagates past the cylinder. This simulated result is in good agreement with the observed flow. For the $R = 55$ case, the flow pattern is similar between the simulated and observed cases, though we begin to see evidence of shedding in the simulated case, which is not necessarily observed in the experimental case. As we get to $R = 65$, we see the flow steepening relative to the earlier

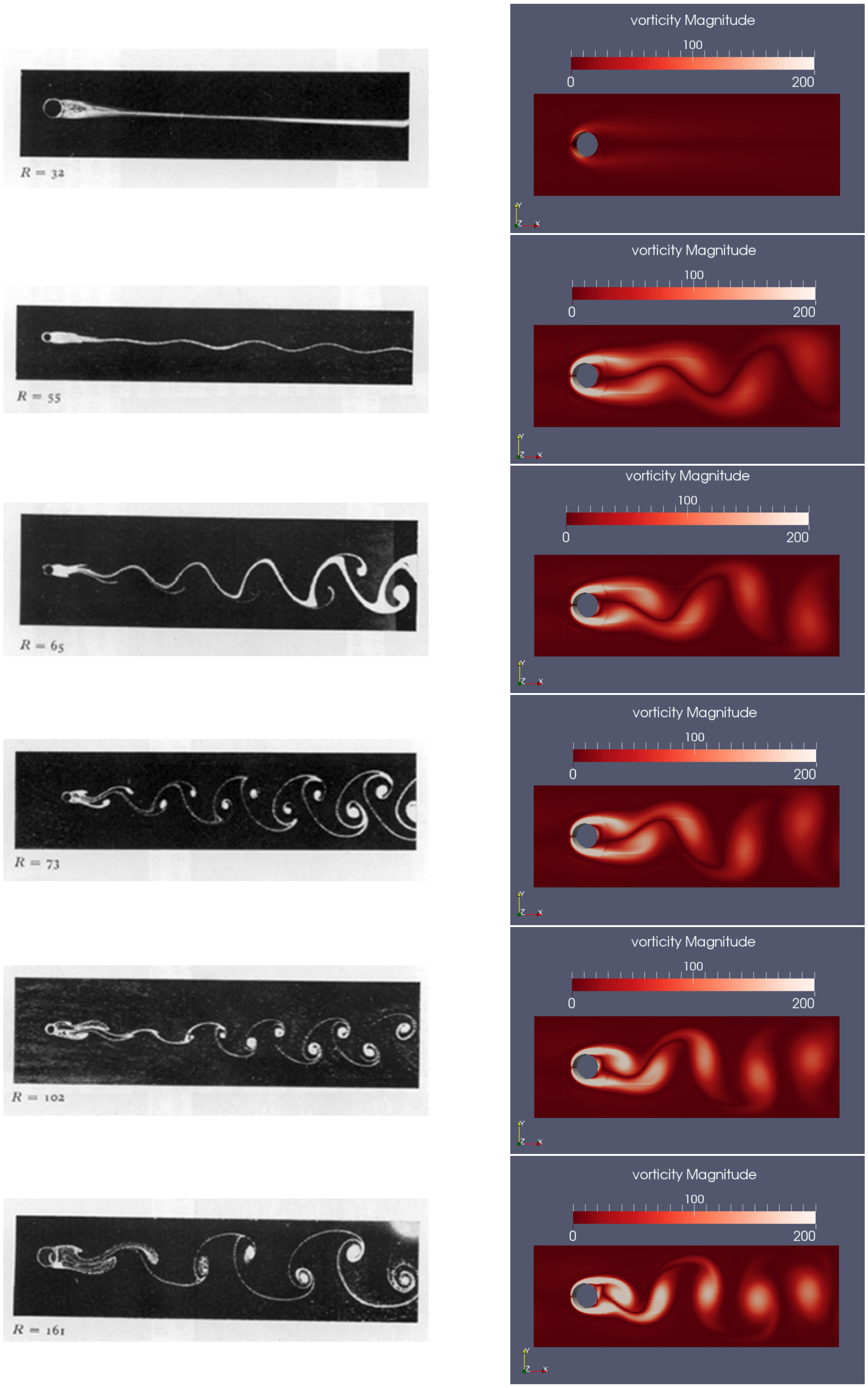

Fig. 5.—: Left: Results from Fig.1. Right: Results from OpenFoam for cases 1-6.

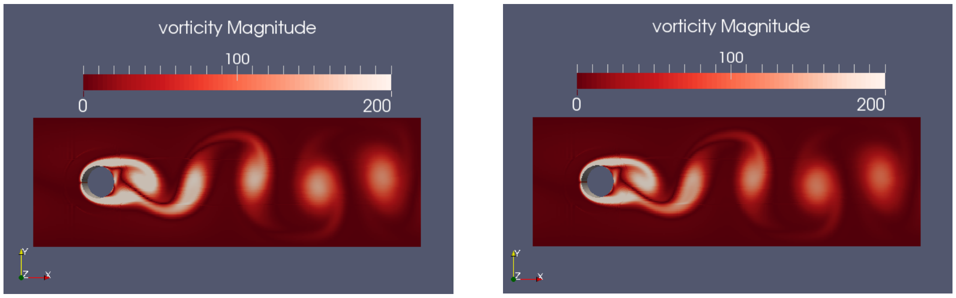

Fig. 6.—: Comparison of OpenFoam simulations which both have $R = 161$, but with different velocities and viscosities. Left: Case 6. Right: Case 7.

cases, and more shedding. As the observed case at $R = 65$ displays shedding, we have better agreement here than for $R = 55$, but the shedding is still much more significant than should be observed. In the $R = 73$ through $R = 161$ cases, we see the flow steepening and the shedding becoming more significant as Reynolds number increases, which agrees with observations.

We also wanted to test the law of similarity. For this, two flows with the same Reynolds number should look the same even if their velocities and kinematic viscosities are different. Here we used cases 6 and 7, and the results can be seen in Fig. 6. As may be noted, even with different flow parameters the solutions look exactly the same because they have the same Reynolds numbers. Thus, OpenFoam is able to confirm the law of similarity.

While Fig. 5 shows a good qualitative comparison between the observed and simulated results, a quantitative approach is necessary to properly evaluate the effectiveness of $OpenFoam$. One way to quantitatively describe the flow around a cylinder is via the Strouhal number. The Strouhal number is used to describe the shedding frequency of a fluid and is defined as

$$S = \frac{fL}{U}, \tag{13}$$

R	f_{emp} $[Hz]$	S_{emp}	f_{OF} $[Hz]$	S_{OF}
55	6.440	0.1288	19.61	0.3922
65	6.995	0.1399	21.31	0.4262
73	7.36	0.1472	22.09	0.4418
102	8.295	0.1659	22.52	0.4504
161	9.340	0.1868	24.26	0.4852

Table 2:: Frequencies and Strouhal numbers for OpenFoam and calculated from an empirical formula from Fey et al. (1998)

where f is the shedding frequency. Fey et al. (1998) found an empirical relationship between the Strouhal number and the Reynolds number given by

$$S(R) = 0.2684 - \frac{1.0356}{\sqrt{R}}, \tag{14}$$

which can be used to calculate the Strouhal number with only the Reynolds number. By using this empirical relationship to find what the Strouhal number should be for a given case, we can compare the calculated Strouhal number to our simulated Strouhal number to test the accuracy of the simulations. Since we know the diameter of the cylinder and the speed of the fluid for each case (see Table 1), we only need to find the shedding frequency to calculate the Strouhal number. We extract vorticity data from a single point located behind the cylinder in the Karman vortex sheet over the entire time domain. We then approximate the range over which the the fluid has reached a steady state, and we execute a fast-fourier-transform via *numpy* in Python to calculate the shedding frequency over the period. We use this frequency to calculate the Strouhal number. Likewise, while we can calculate the expected Strouhal number from the Reynolds number, we can also use this Strouhal number to calculate the expected frequency. The results are shown in Table 2. We

R_i	R_{i+1}	$(S_{i+1}/S_i)_{OF}$	$(S_{i+1}/S_i)_{emp}$	$\frac{(S_{i+1}/S_i)_{OF}}{(S_{i+1}/S_i)_{emp}}$
55	65	1.087	1.087	1.000
65	73	1.037	1.052	0.9857
73	102	1.019	1.127	0.9042
102	161	1.077	1.126	0.9565

Table 3:: Comparison of Strouhal numbers from *OpenFoam* and from the empirical formula from Fey et al. (1998)

did not find the Strouhal number for the case of $R = 32$ because of the lack of shedding.

At first look, the results in Table 2 are not promising. Our simulated frequencies and Strouhal numbers are larger than the expected values by approximately a factor of three. Due to these over-estimations, we also investigated how the Strouhal number changed with Reynolds number. Since the results of Fig. 5 indicate there is an expected amount of change in the shedding as Reynolds number changes, we can investigate if our shedding is changing properly even if the values are off. To do this, we take a ratio of the Strouhal number for a given case with the case directly below it. So, for instance, we find the ratio of $R = 65$ to $R = 55$ to test the change in Strouhal number as R changes. We do this for both the simulated and empirical Strouhal numbers, and compare them by taking a ratio of the resultant ratios. The closer this ratio is to one, the more accurate the simulation in terms of how the shedding changes as Reynolds number changes. The results are shown in Table 3, and are very promising. The worst case is the change between $R = 73$ to $R = 102$, but the three remaining cases are within 95% of the expected Strouhal ratio from the empirical relationship. At the change from $R = 55$ to $R = 66$ we even find the ratios are exactly the same, which is extremely promising. From this, we can see that while OpenFoam is not accurately predicting the Strouhal number, it is accurately modeling the change in the

Strouhal number.

5. Summary

After simulating an incompressible, viscous flow around a cylinder using the *icoFoam* solver in *OpenFoam*, we compared our results to observations. Regarding a qualitative comparison, our results showed decent agreement with experimental results from Homann (1936), which were reproduced by Batchelor (1967). We also produced a qualitative comparison of simulations done for the same Reynolds number, but with different fluid parameters, and found good agreement. This showed *OpenFoam* obeyed the law of similarity.

In order to further our analysis, we found the Strouhal number to better understand the shedding frequency in our model. We compared the Strouhal numbers from our simulations to Strouhal numbers calculated from an empirical relationship in Fey et al. (1998) which shows a relationship between the Strouhal and Reynolds numbers. We used this empirical relationship to find an expected value of the Strouhal number and did not find good agreement between our simulations and the expected results. To further our analysis, we explored the change in the Strouhal number as the Reynolds number changes. For this, we found very good agreement. We thus conclude that, using the conditions in this simulation, we are not able to accurately model the shedding in an incompressible, viscous flow around a cylinder, but we are able to accurately model how the shedding should change as Reynolds number changes.

We acknowledge that the computational work reported in this paper was performed on the Shared Computing Cluster which is administered by Boston University's Research Computing Services. We would also like to acknowledge the assistance of Prof. Sheryl

Grace, Connor Robinson, and Luisa Capannolo in providing assistance when it was needed. Additionally, we would like to acknowledge my penchant for using the royal “we”, even when I am the lone author; some habits are hard to kick.

REFERENCES

Batchelor, G. K., 1967, "An Introduction to Fluid Dynamics", Cambridge University Press

Choudhuri, A.R., 1998, "The Physics of Fluids and Plasmas: An Introduction for Astrophysicists", Cambridge University Press

Fey, U., Konig, M., Eckelmann, H., "A New Strouhal-Reynolds Number Relationship for the Circular Cylinder in the Range 47¡Re¡2e5", 1998, Physics of Fluids, 10, 1547

Hiimenz, K., 1911, "Die Grenschicht an Einem den Gleichformigen Flussigkeitsstrom Eingetauchten Geraden Kreiszlinder", Dinglers Polytech. J., 326

Homann, F., 1936, Forsch. Ing.-Wes., 7, 1

Puig, J.C., "OpenFOAM Guide for Beginners", the-foam-house5.webnode.es

Weller, H.G., Tabor, G., Jasak, H., & Fureby, C., 1998, "A Tensorial Approach to Computational Continuum Mechanics Using Object-Oriented Techniques", Comput. Phys., 12, 620

This manuscript was prepared with the AAS LaTeX macros v5.2.

A. blockMeshDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1606+                                |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
     version     2.0;
     format      ascii;
     class       dictionary;
     object      blockMeshDict;
 }
 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

convertToMeters 0.1;

vertices
(
        (0.5 0 -0.5)
        (1 0 -0.5)
        (10 0 -0.5)
        (10 0.707107 -0.5)
        (0.707107 0.707107 -0.5)
        (0.353553 0.353553 -0.5)
        (10 2 -0.5)
        (0.707107 2 -0.5)
        (0 2 -0.5)
        (0 1 -0.5)
        (0 0.5 -0.5)

        (-0.5 0 -0.5)
        (-1 0 -0.5)
        (-2 0 -0.5)
        (-2 0.707107 -0.5)
        (-0.707107 0.707107 -0.5)
        (-0.353553 0.353553 -0.5)
        (-2 2 -0.5)
        (-0.707107 2 -0.5)

        (0.5 0 0.5)
        (1 0 0.5)
        (10 0 0.5)
        (10 0.707107 0.5)
        (0.707107 0.707107 0.5)
        (0.353553 0.353553 0.5)
        (10 2 0.5)
        (0.707107 2 0.5)
        (0 2 0.5)
        (0 1 0.5)
        (0 0.5 0.5)

        (-0.5 0 0.5)
        (-1 0 0.5)
        (-2 0 0.5)
        (-2 0.707107 0.5)
        (-0.707107 0.707107 0.5)
        (-0.353553 0.353553 0.5)
        (-2 2 0.5)
        (-0.707107 2 0.5)
```

```
        (10 -0.707107 -0.5)
        (0.707107 -0.707107 -0.5)
        (0.353553 -0.353553 -0.5)
        (10 -2 -0.5)
        (0.707107 -2 -0.5)
        (0 -2 -0.5)
        (0 -1 -0.5)
        (0 -0.5 -0.5)

        (-2 -0.707107 -0.5)
        (-0.707107 -0.707107 -0.5)
        (-0.353553 -0.353553 -0.5)
        (-2 -2 -0.5)
        (-0.707107 -2 -0.5)

        (10 -0.707107 0.5)
        (0.707107 -0.707107 0.5)
        (0.353553 -0.353553 0.5)
        (10 -2 0.5)
        (0.707107 -2 0.5)
        (0 -2 0.5)
        (0 -1 0.5)
        (0 -0.5 0.5)

        (-2 -0.707107 0.5)
        (-0.707107 -0.707107 0.5)
        (-0.353553 -0.353553 0.5)
        (-2 -2 0.5)
        (-0.707107 -2 0.5)
);

blocks
(
        hex (5 4 9 10 24 23 28 29) (80 20 1) simpleGrading (10 1 1)
        hex (0 1 4 5 19 20 23 24) (80 20 1) simpleGrading (10 1 1)
        hex (1 2 3 4 20 21 22 23) (200 20 1) simpleGrading (1 1 1)
        hex (4 3 6 7 23 22 25 26) (200 40 1) simpleGrading (1 1 1)
        hex (9 4 7 8 28 23 26 27) (20 40 1) simpleGrading (1 1 1)
        hex (16 10 9 15 35 29 28 34) (20 80 1) simpleGrading (1 10 1)
        hex (11 16 15 12 30 35 34 31) (20 80 1) simpleGrading (1 10 1)
        hex (12 15 14 13 31 34 33 32) (20 20 1) simpleGrading (1 1 1)
        hex (15 18 17 14 34 37 36 33) (40 20 1) simpleGrading (1 1 1)
        hex (9 8 18 15 28 27 37 34) (40 20 1) simpleGrading (1 1 1)


        hex (40 45 44 39 53 58 57 52) (20 80 1) simpleGrading (1 10 1)
        hex (0 40 39 1 19 53 52 20) (20 80 1) simpleGrading (1 10 1)
        hex (1 39 38 2 20 52 51 21) (20 200 1) simpleGrading (1 1 1)
        hex (39 42 41 38 52 55 54 51) (40 200 1) simpleGrading (1 1 1)
        hex (44 43 42 39 57 56 55 52) (40 20 1) simpleGrading (1 1 1)
        hex (48 47 44 45 61 60 57 58) (80 20 1) simpleGrading (10 1 1)
        hex (11 12 47 48 30 31 60 61) (80 20 1) simpleGrading (10 1 1)
        hex (12 13 46 47 31 32 59 60) (20 20 1) simpleGrading (1 1 1)
        hex (47 46 49 50 60 59 62 63) (20 40 1) simpleGrading (1 1 1)
        hex (44 47 50 43 57 60 63 56) (20 40 1) simpleGrading (1 1 1)
);

edges
(
        arc 0 5 (0.469846 0.17101 -0.5)
        arc 5 10 (0.17101 0.469846 -0.5)
```

```
        arc 1 4 (0.939693 0.34202 -0.5)
        arc 4 9 (0.34202 0.939693 -0.5)
        arc 19 24 (0.469846 0.17101 0.5)
        arc 24 29 (0.17101 0.469846 0.5)
        arc 20 23 (0.939693 0.34202 0.5)
        arc 23 28 (0.34202 0.939693 0.5)
        arc 11 16 (-0.469846 0.17101 -0.5)
        arc 16 10 (-0.17101 0.469846 -0.5)
        arc 12 15 (-0.939693 0.34202 -0.5)
        arc 15 9 (-0.34202 0.939693 -0.5)
        arc 30 35 (-0.469846 0.17101 0.5)
        arc 35 29 (-0.17101 0.469846 0.5)
        arc 31 34 (-0.939693 0.34202 0.5)
        arc 34 28 (-0.34202 0.939693 0.5)


        arc 0 40 (0.469846 -0.17101 -0.5)
        arc 40 45 (0.17101 -0.469846 -0.5)
        arc 1 39 (0.939693 -0.34202 -0.5)
        arc 39 44 (0.34202 -0.939693 -0.5)
        arc 19 53 (0.469846 -0.17101 0.5)
        arc 53 58 (0.17101 -0.469846 0.5)
        arc 20 52 (0.939693 -0.34202 0.5)
        arc 52 57 (0.34202 -0.939693 0.5)
        arc 11 48 (-0.469846 -0.17101 -0.5)
        arc 48 45 (-0.17101 -0.469846 -0.5)
        arc 12 47 (-0.939693 -0.34202 -0.5)
        arc 47 44 (-0.34202 -0.939693 -0.5)
        arc 30 61 (-0.469846 -0.17101 0.5)
        arc 61 58 (-0.17101 -0.469846 0.5)
        arc 31 60 (-0.939693 -0.34202 0.5)
        arc 60 57 (-0.34202 -0.939693 0.5)
);

boundary
(

        top
        {
                type symmetryPlane;
                faces
                (
                        (7 8 27 26)
                        (6 7 26 25)
                        (8 18 37 27)
                        (18 17 36 37)
                );
        }

        bottom
        {
                type symmetryPlane;
                faces
                (
                        (49 50 63 62)
                        (50 43 56 63)
                        (43 42 55 56)
                        (42 41 54 55)
                );
        }

        inlet
```

```
        {
                type patch;
                faces
                (
                        (14 13 32 33)
                        (17 14 33 36)

                        (46 13 32 59)
                        (46 49 62 59)
                );
        }

        outlet
        {
                type patch;
                faces
                (
                        (2 3 22 21)
                        (3 6 25 22)

                        (38 51 21 2)
                        (41 54 51 38)
                );
        }

        cylinder
        {
                type wall;
                faces
                (
                        (10 5 24 29)
                        (5 0 19 24)
                        (16 10 29 35)
                        (11 16 35 30)

                        (48 11 30 61)
                        (45 48 61 58)
                        (40 45 58 53)
                        (0 40 53 19)
                );
        }

        frontAndBack
        {
                type empty;
                faces
                (
                        (5 10 9 4)
                        (24 23 28 29)
                        (0 5 4 1)
                        (19 20 23 24)
                        (1 4 3 2)
                        (20 21 22 23)
                        (4 7 6 3)
                        (23 22 25 26)
                        (4 9 8 7)
                        (28 23 26 27)
                        (16 15 9 10)
                        (35 29 28 34)
                        (12 15 16 11)
                        (31 30 35 34)
                        (13 14 15 12)
```

```
                        (32 31 34 33)
                        (14 17 18 15)
                        (33 34 37 36)
                        (15 18 8 9)
                        (34 28 27 37)

                        (45 40 39 44)
                        (58 57 52 53)
                        (40 0 1 39)
                        (53 52 20 19)
                        (39 1 2 38)
                        (52 51 21 20)
                        (39 38 41 42)
                        (52 55 54 51)
                        (44 39 42 43)
                        (57 56 55 52)
                        (47 48 45 44)
                        (60 57 58 61)
                        (12 11 48 47)
                        (31 60 61 30)
                        (13 12 47 46)
                        (32 59 60 31)
                        (49 46 47 50)
                        (62 63 60 59)
                        (50 47 44 43)
                        (63 56 57 60)
                );
        }
);

mergePatchPairs
(
);

  // ************************************************************************* //
```

B. U

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (5 0 0);

boundaryField
{
    top
    {
        type                    symmetryPlane;
    }

    bottom
    {
        type                    symmetryPlane;
    }

    inlet
    {
        type                    freestream;
        freestreamValue         uniform (5 0 0);
    }

    outlet
    {
        type                    freestream;
        freestreamValue         uniform (5 0 0);
    }

    cylinder
    {
      type                      fixedValue;
        value                   uniform (0 0 0);
    }

    frontAndBack
    {
        type            empty;
    }
}

// ************************************************************************* //
```

C. p

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    top
    {
        type            symmetryPlane;
    }

    bottom
    {
        type            symmetryPlane;
    }

    inlet
    {
        type            freestreamPressure;
    }

    outlet
    {
        type            freestreamPressure;
    }

    frontAndBack
    {
        type            empty;
    }

    cylinder
    {
        type            zeroGradient;
    }
}

// ************************************************************************* //
```

D. transportProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      transportProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

transportModel  Newtonian;

nu              nu [0 2 -1 0 0 0 0] 0.003106;

// ************************************************************************* //
```

E. RASProperties

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      RASProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

RASModel        laminar;

turbulence      off;

printCoeffs     off;

// ************************************************************************* //
```

F. controlDict

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application         icoFoam;

startFrom           startTime;

startTime           0;

stopAt              endTime;

endTime             1.75;

deltaT              0.00001;

writeControl        timeStep;

writeInterval       1000;

purgeWrite          0;

writeFormat         ascii;

writePrecision      6;

writeCompression    off;

timeFormat          general;

timePrecision       6;

runTimeModifiable   true;

// ************************************************************************* //
```

G. fvSchemes

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         Euler;
}

gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
}

divSchemes
{
    default         none;
    div(phi,U)      Gauss linear;
}

laplacianSchemes
{
    default         none;
    laplacian(nu,U) Gauss linear orthogonal;
    laplacian((1|A(U)),p) Gauss linear orthogonal;
}

interpolationSchemes
{
    default             linear;
    interpolate(HbyA)   linear;
}

snGradSchemes
{
    default         orthogonal;
}

fluxRequired
{
    default         no;
    p               ;
}


// ************************************************************************* //
```

H. fvSolution

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    p
    {
        solver                  PCG;
        preconditioner          DIC;
        tolerance               1e-06;
        relTol                  0;
    }

    U
    {
        solver                  PBiCG;
        preconditioner          DILU;
        tolerance               1e-05;
        relTol                  0;
    }
}

PISO
{
    nCorrectors     2;
    nNonOrthogonalCorrectors 3;
    pRefCell        0;
    pRefValue       0;
}


// ************************************************************************* //
```