FOSSEE, IIT Bombay
OpenFOAM Case Study Project
June 13, 2023

Passive Propulsion in Vortex Wakes

Gade Chandramouli
Department of Mechanical Engineering, IIT Bombay

Abstract

The performance of NACA 0012 airfoil placed in vortex wakes of a cylinder is studied through simulations in OpenFOAM 7. Broadly, 2 cases have been investigated - allowing heave and rotational movement of the airfoil and rigidly fixed airfoil. Variations with angle of attack has also been studied with the rigidly fixed wing. It is shown that there is a propulsive force on the airfoil in both cases. PimpleFoam solver was used for transient and moving mesh simulations.

1 Introduction

Making energy efficient transportation is always a need. Drag reduction is one way to reduce fuel consumption in many modes of transportation. One thing that is observed is vortex formation on these bodies. Vortexes are formed on boundaries of objects due to boundary layer separation. One way is to use passive devices to avoid vortex formation and decrease the amount of drag induced by them. The second way is to extract some of the energy from these vortexes and put it back into the system. This can be done by placing additional aerodynamics structures in the wakes. This particular problem is being investigated in this study.

In this regard, Jonathan et. al. in [1] have experimentally shown that for a particular size of airfoil placed in vortex wakes can extract its energy and propel itself forward at various distances from the bluff body. Beal et. al. in [2] discusses how a dead fish is propelled forward when its flexible body oscillates with oncoming despite being well out of the suction zone. They have also performed similar experiments with NACA 0012 airfoil and shown a forward thrust along with oscillations of the airfoil. In the current work, the results of these papers is simulated and validated through simulations and a comparison between these 2 cases is being performed.

2 Problem Statement

The problem statement is to simulate flow over a cylinder with an airfoil placed behind it. Air has been chosen as the fluid to be simulated since most of cases dicussed above would involve air. The problem is being simplified to a 2-D case. In the first case no moving mesh is being used. A x/d of 5 is being used where x is distance between the cylinder and the airfoil and d is the diameter if the cylinder. Also a D/c of 1 is being used where c is length of the chord. The angle of attack has been varied from -15 to 15 degrees in steps. The coefficient of drag and lift are being calculated on the body and being compared with the experimentally obtained values by Jonathan et. al. in [1].

In the second case, the airfoil is allowed to move. Constraints were placed on the airfoil to allow to move in the heave direction and rotate in the 2D plane. So the airfoil has 2 degree of freedoms. Restraints have been put on the airfoil in both the degree of freedoms. The values of stiffness are chosen to allow for a convergent simulation. Again, coefficient of drag and lift are being calculated.

The turbulence model being used is K-ω SST. The Reynolds number is being calculated based on the chord length of the airfoil. The Reynolds number is kept at 10^5 as used in [1] which is the critical Reynolds number for NACA 0012 according to [3]. In the second case, Re of the order of $6 * 10^5$ is being used.

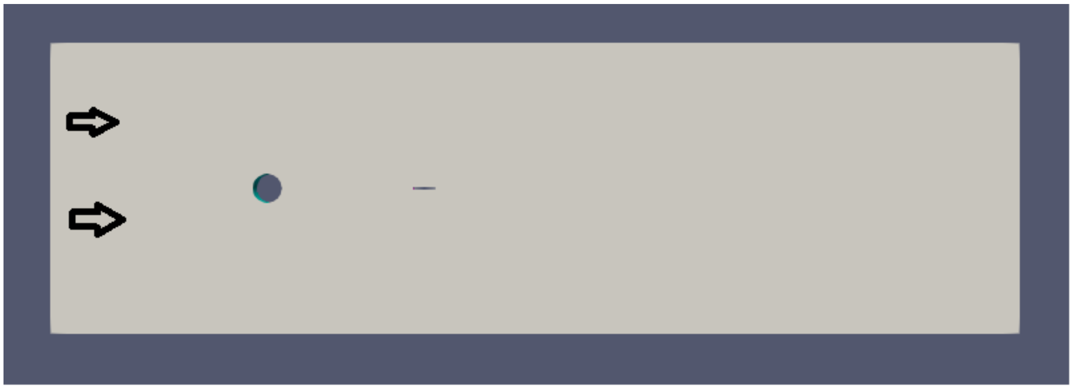


Figure 1: Domain with flow direction shown

3 Governing Equations and Models

The Navier Stokes equations for incompressible flows are be used.

$$\frac{\partial u_i}{\partial x_i} = 0$$

$$\rho\frac{\partial u_i}{\partial t} + \rho u_j\frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial t_{ji}}{\partial x_j}$$

$$where\ t_{ij} = \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$$

Typical k-ϵ model uses wall functions which might not be fully accurate for complicated flows. The k-ω model is more accurate near the wall and does not require a very fine mesh. But it is sensitive to the free stream turbulence. The k-ω SST turbulence model uses k-ω at the wall and k-ϵ in the free stream flow. The equations solved for the k-ω SST model are -

Turbulence Kinetic Energy

$$\frac{\partial k}{\partial t} + U_j\frac{\partial k}{\partial x_j} = P_k - \beta^* k\omega + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_k \nu_T)\frac{\partial k}{\partial x_j}\right]$$

Specific Dissipation Rate

$$\frac{\partial \omega}{\partial t}+U_j\frac{\partial \omega}{\partial x_j}=\alpha S^2-\beta\omega^2+\frac{\partial}{\partial x_j}\left[\left(\nu+\sigma_\omega\nu_T\right)\frac{\partial \omega}{\partial x_j}\right]+2\left(1-F_1\right)\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i}$$

Some auxillary relations -

$$\nu_T=\frac{a_1 k}{\max\left(a_1\omega, SF_2\right)}$$

$$F_2=\tanh\left[\left[\max\left(\frac{2\sqrt{k}}{\beta^*\omega y},\frac{500\nu}{y^2\omega}\right)\right]^2\right]$$

$$P_k=\min\left(\tau_{ij}\frac{\partial U_i}{\partial x_j},10\beta^* k\omega\right)$$

$$F_1=\tanh\left\{\left\{\min\left[\max\left(\frac{\sqrt{k}}{\beta^*\omega y},\frac{500\nu}{y^2\omega}\right),\frac{4\sigma_{\omega 2}k}{CD_{k\omega}y^2}\right]\right\}^4\right\}$$

$$CD_{k\omega}=\max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i},10^{-10}\right)$$

$$\phi=\phi_1 F_1+\phi_2\left(1-F_1\right)$$

$$\alpha_1=\frac{5}{9},\alpha_2=0.44$$

$$\beta_1=\frac{3}{40},\beta_2=0.0828$$

$$\beta^*=\frac{9}{100}$$

$$\sigma_{k1}=0.85,\sigma_{k2}=1$$

$$\sigma_{\omega 1}=0.5,\sigma_{\omega 2}=0.856$$

Explanation of the rigid body model

The way in which solvers with rigid body motion work is, at a given time step, first the fluid motion is solved, then the fluid solution is passed to the motion solver at the end of the time step and the position and orientation is determined from that. But this is a weak coupling between fluid and the motion. But while using pimple algorithm, the mesh is solved within the outer correctors. This allows for a stronger coupling. When the fluid solution converges, it can be implied that the motion is also converging to a certain extent since a significant change in the position or orientation of the floating body would result in a slightly changed mesh, thus requiring a different fluid solution. Every time the motion solver is called, the velocity of the rigid body v first gets updated based on the determined force f (from integrating pressure field over the contact patches) and thus the acceleration f/m of the last iteration multiplied by half the previous time step size. The position is then determined from the product of the acceleration and the new time step size.

$$v_1^{k+0.5}=v_0+\frac{\Delta t_0}{2}\left(\frac{f}{m}\right)_0^k$$

$$x_1^{k+0.5}=x_0+\Delta t_1\left(v_1^{k+0.5}\right)$$

Sometimes acceleration varies a lot in the first few iteration, which causes instability in the solution. Then acceleration is relaxed using a parameter. So an additional equation is required -

$$\left(\frac{f}{m}\right)_1^{k+1} = \left(\frac{f}{m}\right)_1^{k} + \omega_a \left[\left(\frac{\tilde{f}}{m}\right)_1^{k+1} - \left(\frac{f}{m}\right)_1^{k}\right]$$

These are representative equations. Similar equations are solved in every direction - that are 3 translational and 3 rotational equations.

4 Simulation Procedure

4.1 Geometry and Mesh

The domain is a 2D rectangular channel with a depth of 1m. Considering this a 2D simulation, the cylinder has a length of 1m and the airfoil has a span of 1 m as well covering entire depth of the channel. The distance (x) between the cylinder and the airfoil was kept at 5m. The diameter (D) and chord (c) of the cylinder and airfoil respectively is 1 m. This gives us a D/c of 1 and x/D of 5.At distance of 20 m was maintained behind the airfoil so that the outlet conditions don't affect the flow near the airfoil.

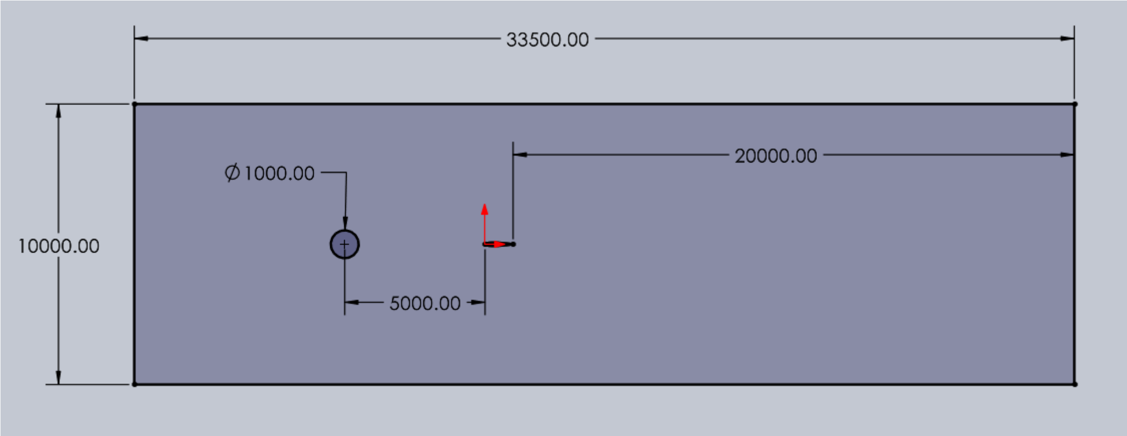


Figure 2: Geometry with relevant lengths indicated

The geometry and mesh were made using blockMeshDict in OpenFOAM. For the purpose of meshing the domain was divided into 24 blocks.

The intent while making the mesh was to make it fine near the cylinder and airfoil wall to capture the high gradients due to the no slip boundary condition. Also the mesh right after the cylinder needed to be fine since it will contain the eddies from the cylinder which have high velocity gradients again. Also, the region behind the airfoil had a refined mesh to capture the oscillatory movement of the fluid once the vortices hit the airfoil. Other considerations were to have smooth variations across the blocks and have a slightly coarser mesh for reduced simulation time and to have a $y^+ > 30$.

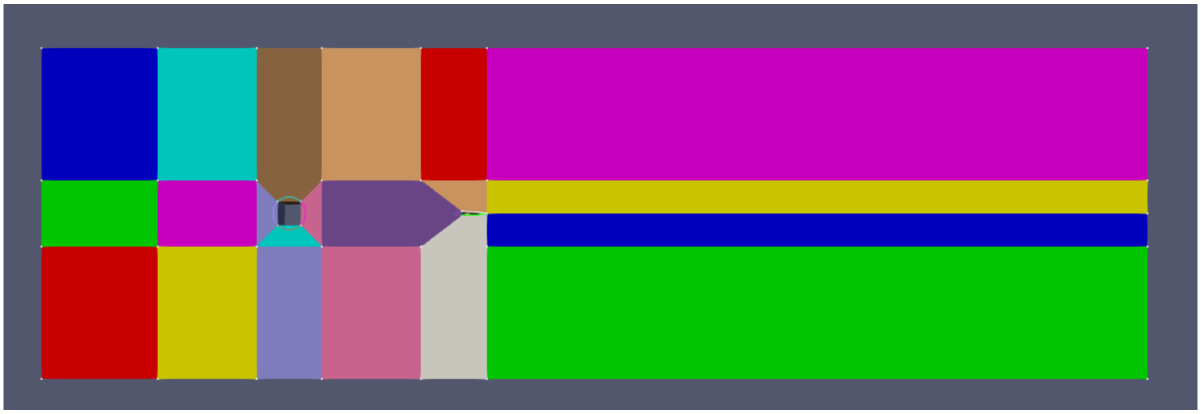


Figure 3: Blocks the domain was divided into

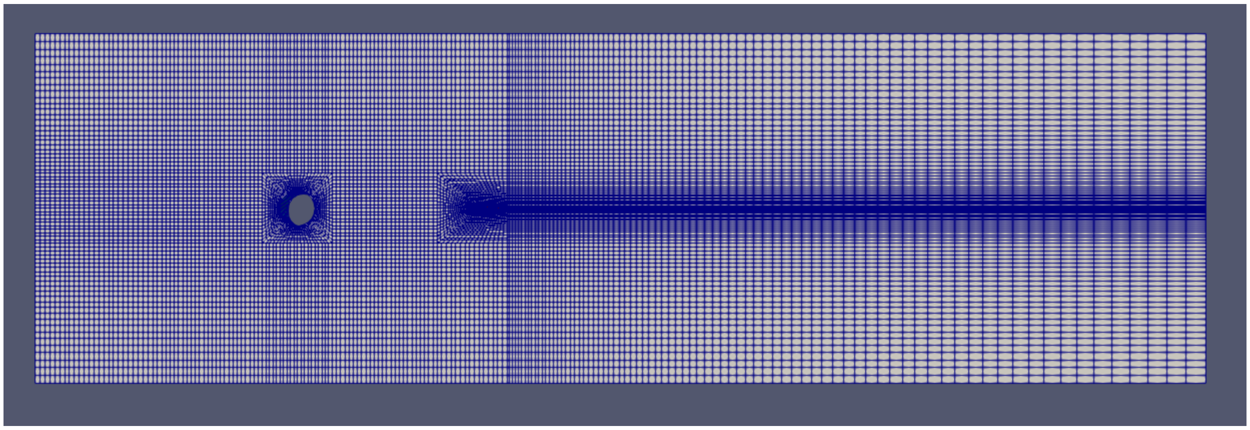


Figure 4: Full domain mesh

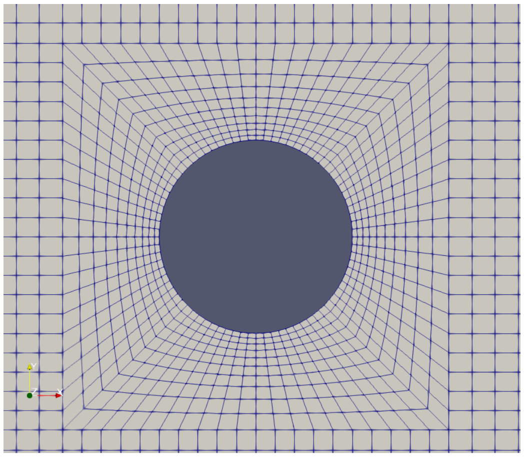


Figure 5: Mesh around the cylinder

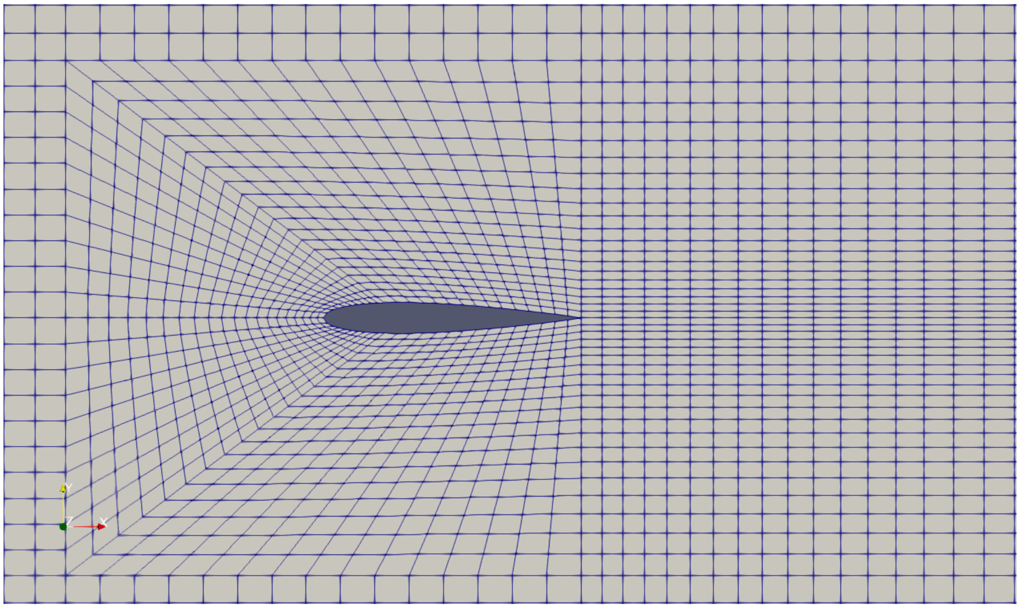


Figure 6: Mesh around the Airfoil

The blocks around the cylinder and airfoil were chosen such that the skewness of the cells around them is minimized. The mesh has been kept similar for all the angle of attacks. The mesh statistics and quality have been given below -

Nodes	32668
Elements	16000
Max aspect ratio	22.0525
Max skewness	1.20282

A total of 7 individual patches have been defined namely -

- inlet - the left side patch (refer to velocity direction in fig. 1)
- outlet - the right side patch
- frontAndBack - the front and back walls
- walls-1 - the top wall
- walls-2 - the bottom wall
- cylinder - the boundary of the cylinder
- airfoil - the boundary of the airfoil

4.2 Initial and Boundary Conditions

The initial velocity was determined from the required Reynolds number. The formula for Re being used is -

$$Re = \frac{U * c}{\nu}$$

where U is inlet velocity, c is chord of airfoil and ν is the kinematic viscosity of the fluid being used. ν of air is $1.5 * 10^{-5}\frac{m^2}{s}$. The turbulent parameters were calculated using the following formulas -

$$\begin{aligned} I &= 0.16 * Re^{\frac{-1}{8}} \\ k &= \frac{3}{2} * (U_\infty * I)^2 \\ \epsilon &= C_\mu \frac{k^{3/2}}{I} \\ \omega &= \frac{\epsilon}{C_\mu k} \\ l_m &= 0.07 * d_h \end{aligned}$$

where I is the turbulent intensity, k is the turbulent kinetic energy at inlet, ω is turbulent dissipation rate at the inlet, l_m is the mixing length, d_h is the length scale of turbulence (we need to choose the minimum length scale in all the geometries, since the diameter and chord are the same here, d_h is taken to be 1.0 m) and C_μ is a empirical constant with the value being 0.09.

For the first case the Re=10^5 was used. So the inlet velocity $U = 1.5\frac{m}{s}$ since $c = 1.0m$. The initial and boundary conditions for the first case are as follows -

Patch/Field	Velocity	Pressure
Initial	uniform 0	uniform 0
Inlet	fixed Value 1.5 $\frac{m}{s}$	zeroGradient
Outlet	zero Gradient	fixedValue 0
”Wall-1|Wall-2”	no Slip	zeroGradient
Cylinder	no Slip	zeroGradient
Airfoil	no Slip	zeroGradient
frontAndBack	empty	empty

Patch/Field	k	ω
Initial	uniform 0.004858	uniform 6.98e-2
Inlet	turbulentIntensityKineticEnergyInlet intensity = 0.038, value = internal field	turbulentMixingLengthFrequencyInlet mixing length = 0.07, value = internal field
Outlet	zeroGradient	zeroGradient
”Wall-1|Wall-2”	kqRWallFunction	omegaWallFunction
Cylinder	kqRWallFunction	omegaWallFunction
Airfoil	kqRWallFunction	omegaWallFunction
frontAndBack	empty	empty

ν_t has calculated boundary condition for inlet, outlet and wall function for the walls.

Additional Boundary Conditions for moving airfoil

For the velocity field, movingWallVelocity (0 0 0) is given for the airfoil patch. The inlet velocity has been increased to 10 m/s. Also, a new field called pointsDispalcement is needed which defines the movement of the nodes on the patches. Type Calulated is given for airfoil patch, empty for frontAndBack and fixedValue (0 0 0) for all the other patches.

4.3 Solver

PimpleFoam solver is being used since it is an incompressible, transient solver with turbulence and moving mesh capabilities. The mandatory fields it uses is kinematic pressure and velocity with additional parameters like k, ω and ν_t for k-ω SST model. It uses the PIMPLE algorithm which is a combination of PISO and SIMPLE algorithms.

The way this algorithm works is that it finds a steady state solution for each time step while under relaxation is applied. After the solution is found, solver goes to the next time-step. For this, the solver uses outer correction loops, to ensure that the equations are converged. After the given tolerance criterion is reached — within the steady-state calculation — the outer correction loop is left, and solver moves on in time. This is done until the simulation end time is reached.

To get the force coefficients (Cl, Cd and Cm), an inbuilt function is used in controlDict. The velocity reference was given to be same as the inlet velocity. The reference area given is 1.0 m^2 which is the span of the airfoil multiplied by the chord of the airfoil.

4.3.1 Fixed Wing case

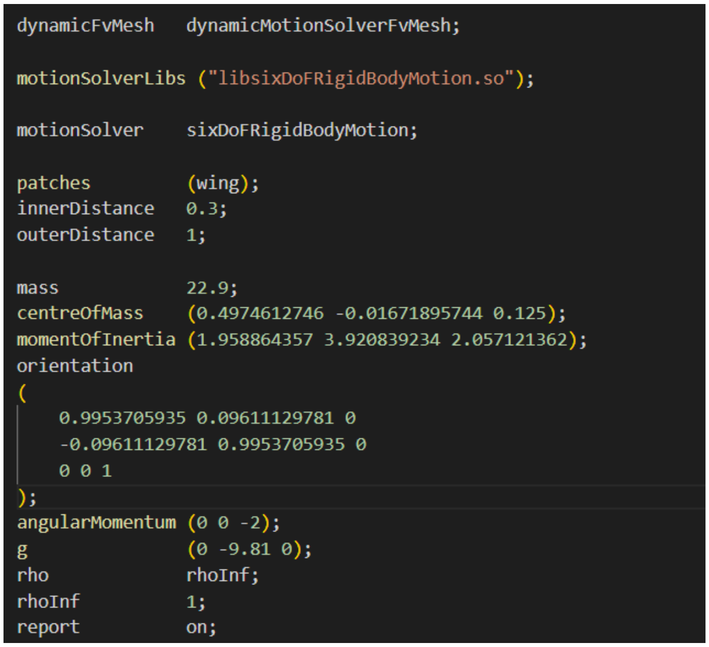
If nOuterCorrector is set to 1, the solver works in PISO mode. So generally nOuterCorrector is kept more than 50. But after applying residual control, I observed that since residual was less than the criteria specified, most of outer corrector loops were empty. So, I am using nOuterCorrector to be 2. The default residuals are being used which is 1e-5 for all parameters. A relaxation factor of 0.6 is being used for velocity equation and 0.9 is being used for all the other equations. The y^+ values are kept in the range of 30 to 300 for this case.

4.3.2 Moving Wing case

To have a reduced simulation time, a combination of simpleFoam and pimpleFoam is used for this case. First simulation is performed in simpleFoam with a fixed wing to allow the flow to develop. Then the mesh is copied and solved fields are mapped to pimpleFoam and solved further there. Here nOuterCorrector of 5 is used.

4.3.3 Dynamic Mesh Dict

The sixDOFRigidBodyMotion is the motion solver that is being used. The patch which is suppossed to be moved is mentioned. The innerDistance and outerDistance specify the regions where mesh deformation is allowed. Between the circles whose diameters are defined by these two parameters, mesh motion is permitted. The remaining parameters in the setup correspond to the properties of the rigid body.

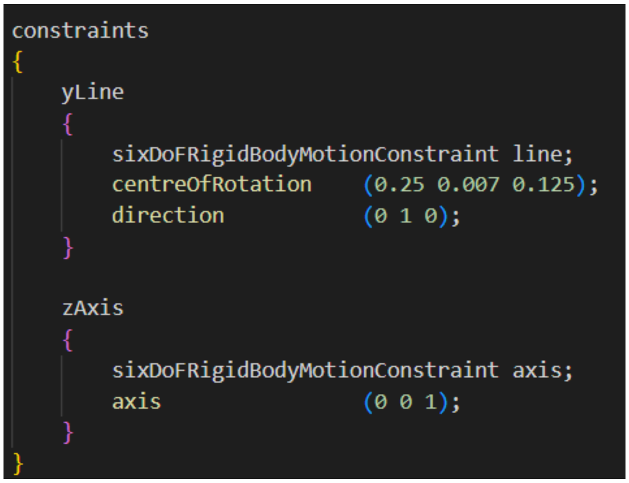


```
dynamicFvMesh   dynamicMotionSolverFvMesh;

motionSolverLibs ("libsixDoFRigidBodyMotion.so");

motionSolver    sixDoFRigidBodyMotion;

patches         (wing);
innerDistance   0.3;
outerDistance   1;

mass            22.9;
centreOfMass    (0.4974612746 -0.01671895744 0.125);
momentOfInertia (1.958864357 3.920839234 2.057121362);
orientation
(
    0.9953705935 0.09611129781 0
    -0.09611129781 0.9953705935 0
    0 0 1
);
angularMomentum (0 0 -2);
g               (0 -9.81 0);
rho             rhoInf;
rhoInf          1;
report          on;
```

Figure 7: Setup dynamic mesh Dict

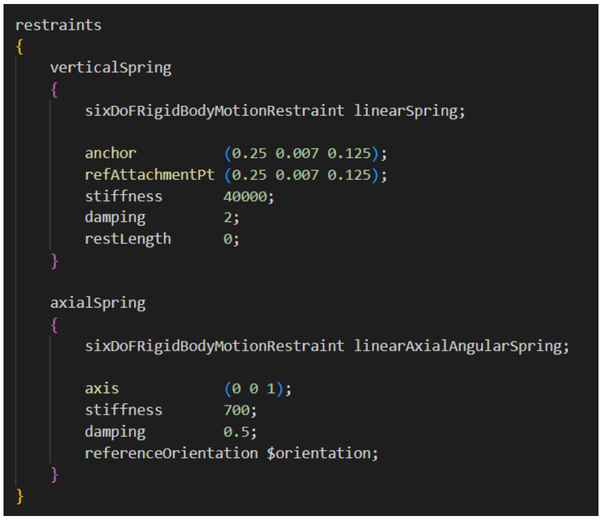
The symplectic solver is used which is 2nd-order explicit time-integrator for position and orientation prediction. Newmark or Crank-Nicolson solvers can be used which are implicit but require more time.

The wing is being constraint to move along the y axis (heave motion) and rotate about the z axis about its center of mass. It has also been restrained in these 2 directions. In the heave direction, a spring of 40000 N/m and a damping of 2 is being used. In the rotational direction, a torsional spring of 700 Nm/rad and a damping of 0.5 is used.



```
constraints
{
    yLine
    {
        sixDoFRigidBodyMotionConstraint line;
        centreOfRotation    (0.25 0.007 0.125);
        direction           (0 1 0);
    }

    zAxis
    {
        sixDoFRigidBodyMotionConstraint axis;
        axis                (0 0 1);
    }
}
```

Figure 8: Constraint



```
restraints
{
    verticalSpring
    {
        sixDoFRigidBodyMotionRestraint linearSpring;

        anchor          (0.25 0.007 0.125);
        refAttachmentPt (0.25 0.007 0.125);
        stiffness       40000;
        damping         2;
        restLength      0;
    }

    axialSpring
    {
        sixDoFRigidBodyMotionRestraint linearAxialAngularSpring;

        axis            (0 0 1);
        stiffness       700;
        damping         0.5;
        referenceOrientation $orientation;
    }
}
```

Figure 9: Restraints

5 Results and Discussions

5.1 Fixed Wing

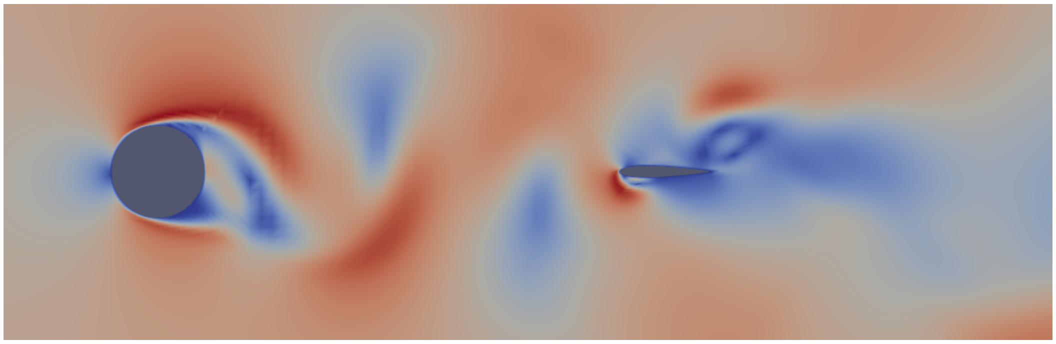


Figure 10: Velocity Field for the zero angle of attack case

The fig. 10 shows the formation of vortices from the cylinder. It is a pair of counter rotating vortices in a side-by-side configuration which generates a negative drag on the airfoil at certain angle of attacks. As we can see in fig. 11, the high pressure eddies hit the side of the airfoil pushing it forward. Although there are some situations when the airfoil has a positive drag, the average values are negative as can be seen in fig. 12. Comparison between theoretical and simulation vortex shedding frequency has been given below -

Theoretical	**deg = 0**	**deg = 5**	**deg = -5**	**deg = 15**	**deg = -15**
0.297 Hz	0.337 Hz	0.347 Hz	0.346 Hz	0.353 Hz	0.342 Hz

The theoretical frequency was found using the empirical relation -

$$\mathrm{Sr} = 0.198\left(1 - \frac{19.7}{\mathrm{Re_D}}\right)$$

where $Sr = \frac{f_s * L}{U_\infty}$, f_s being the shedding frequency, L being the length scale of eddies. The simulation vortex shedding frequency is obtained from the Fourier transform of Cl (coefficient of lift) on the cylinder varying with time.

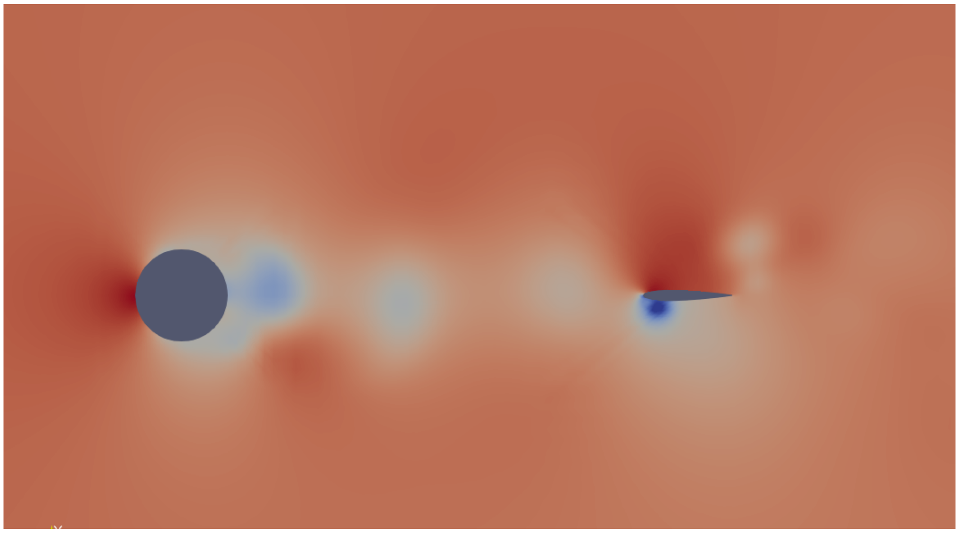


Figure 11: Pressure Field for the zero angle of attack case

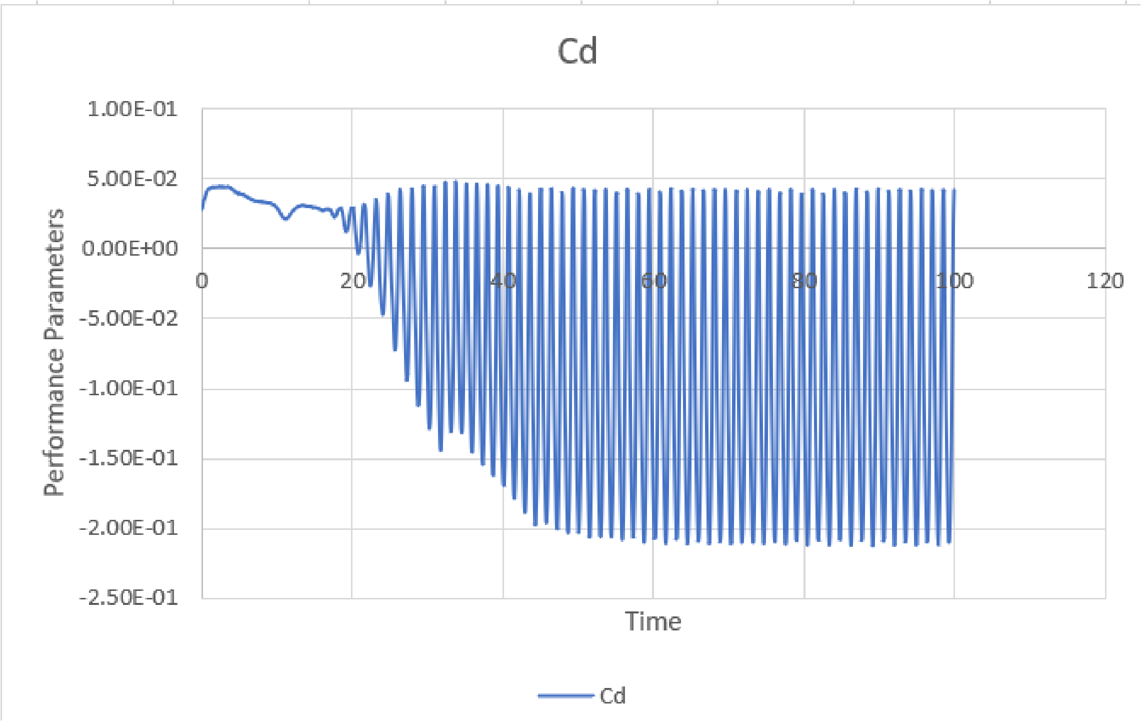


Figure 12: Variation of C_d with time for zero Angle of attack

Comparison of Force Coefficients - The averaged force-coefficients have compared here. The Experimental values were taken from [1].

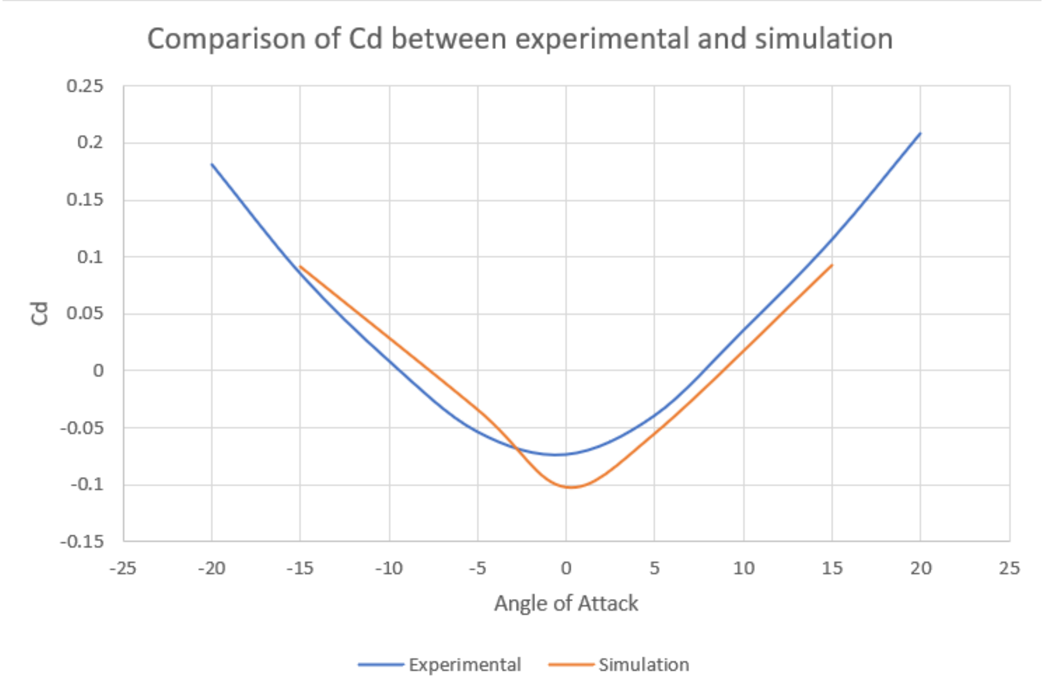


Figure 13: Comparison of C_d

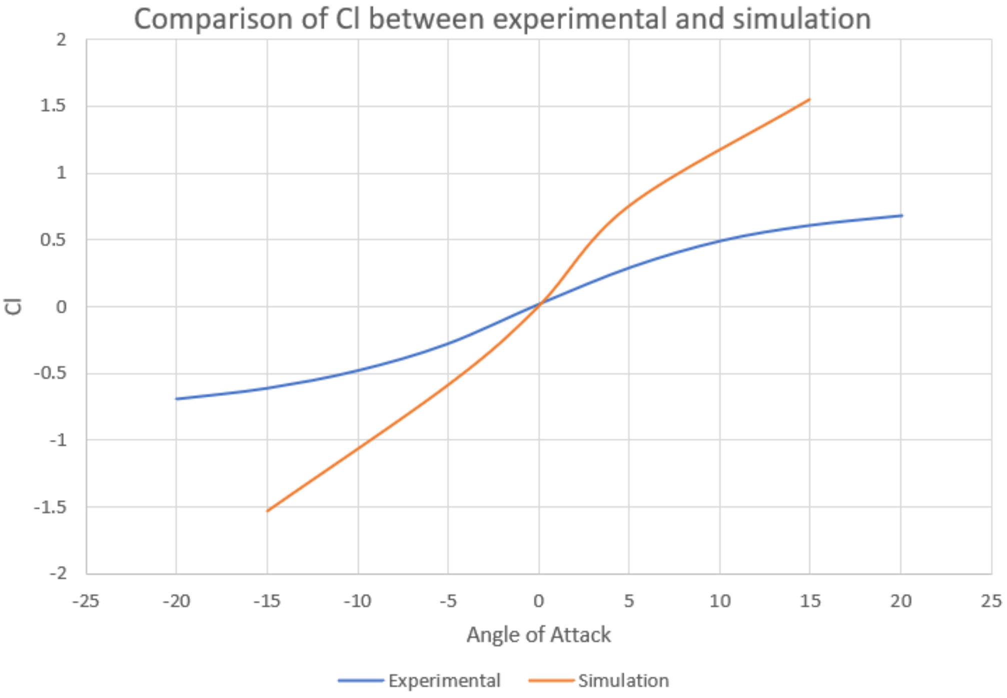


Figure 14: Comparison of C_l

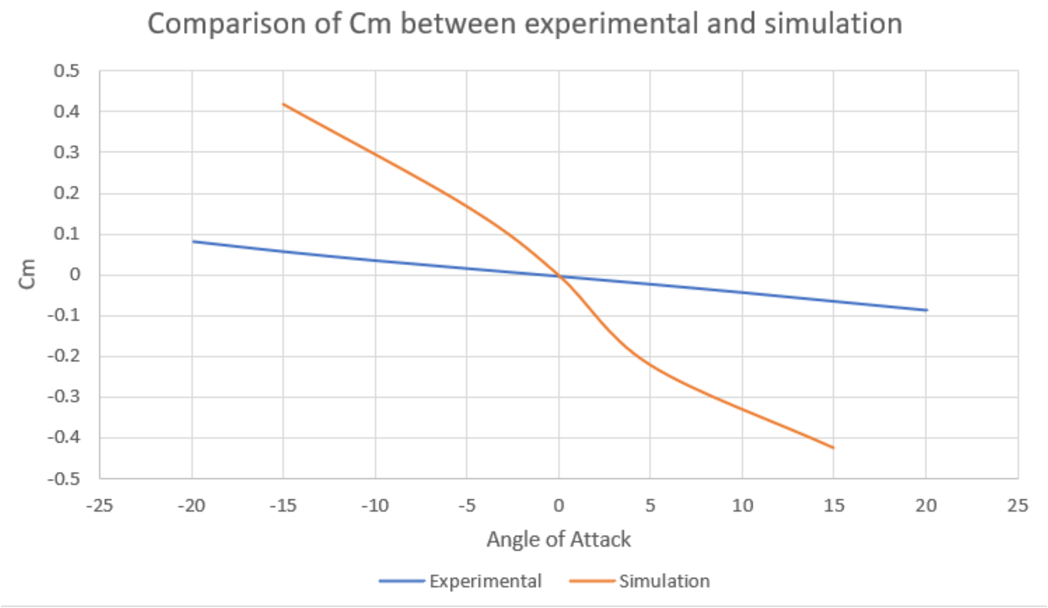


Figure 15: Comparison of C_m

The values of C_d are close to the experimental values as seen from fig. 13. But the values of C_l and C_m are over-predicted by the simulation. The errors are -

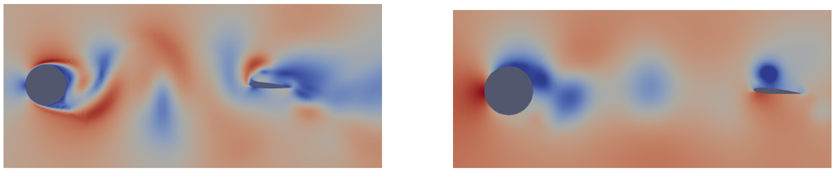
	C_d	C_l	C_m
RMS error	0.019	0.63	0.25
Percentage Error	28.3	132.2	632.1

5.2 Moving Wing

The exact replication of [2] could not be performed since it required an implementation of 2 motion solvers - sixDoFRigidBodyMotion (for the airfoil) and interpolatingSolidBody (for the oscillatory motion of the D-cylinder). This could not be implemented. The oscillatory motion of the D-cylinder was required to control both the frequency of the vortices and the phase difference between the vortices. So the results obtained in this section have not been directly compared to [2].

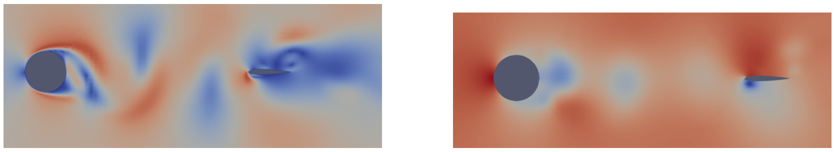
Velocity and pressure contour plots at various time instants

As we can see from the plots in figures 16-20, the wing moves according to pressure applied on it. When a vortex hits the wing from the top, a low pressure region is created and a high pressure region on the other side. Due to which there is resultant force on the wing and it tends move. The rotation is clearly visible through the plots although the heave magnitude is very low. It also seems that the wing oscillation lags the vortices which can decrease the overall thrust since it will try to move in the opposite direction of force for some amount of time.



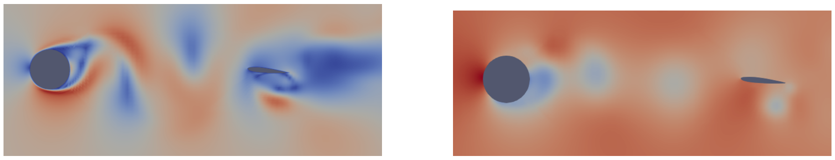
(a) Velocity (b) Pressure

Figure 16: Contours at **t=0.8s**



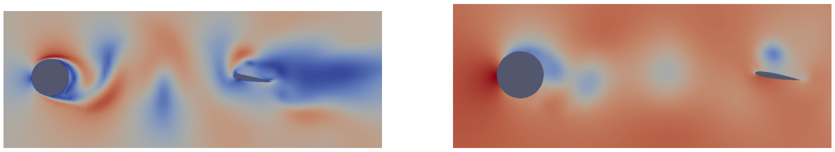
(a) Velocity (b) Pressure

Figure 17: Contours at **t=0.9s**



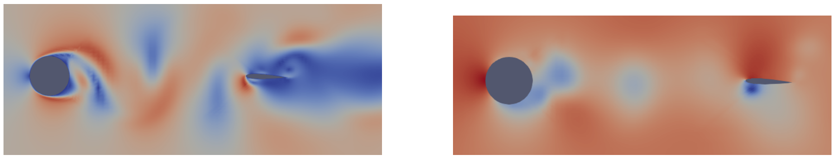
(a) Velocity (b) Pressure

Figure 18: Contours at **t=1.0s**



(a) Velocity (b) Pressure

Figure 19: Contours at **t=1.1s**



(a) Velocity (b) Pressure

Figure 20: Contours at **t=1.2s**

The variation of C_d (as seen in fig.21) is similar in positive and negative regions contrary to the fixed wing case. It is possible stiffness values of the springs are chosen in such a way that it leads to a destructive interference between the motion and oscillating force. The stiffness values of the springs can be set such that a resonance occurs. But it requires very low values of stiffness at which the solution diverges. Some possible ways could be increasing nOuterCorrectors, refining the mesh, using higher mass for the wing etc.

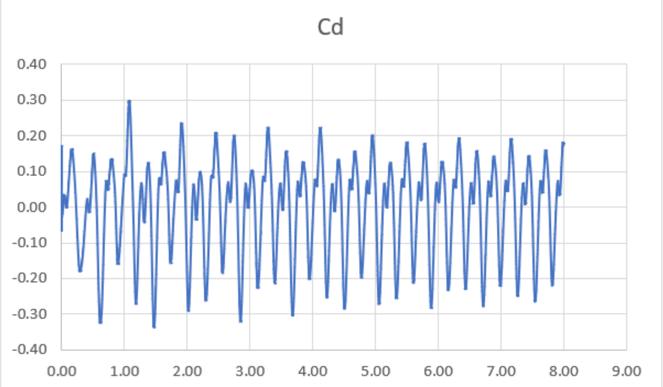


Figure 21: C_d in the case of Moving Wing

The averaged values of force coefficients -

	C_d	C_l	C_m
Moving Wing	0.023	-0.0155	0.053
Fixed Wing	0.0025	-0.094	0.006

A comparison has been shown between moving wing and fixed wing at the same velocity. A substantial amount of improvement can be done as mentioned above. A fairer comparison can be obtained between a fixed wing and when the wing is allowed to move. Also considering that a damping was provided to the springs, we can say that the airfoil is extracting energy out of the vortices and at the same propelling itself forward which agrees with [2].

As can be seen in fig. 22 and fig. 23, the wing tries to go towards a steady state, which agrees with the experimental results of [2].

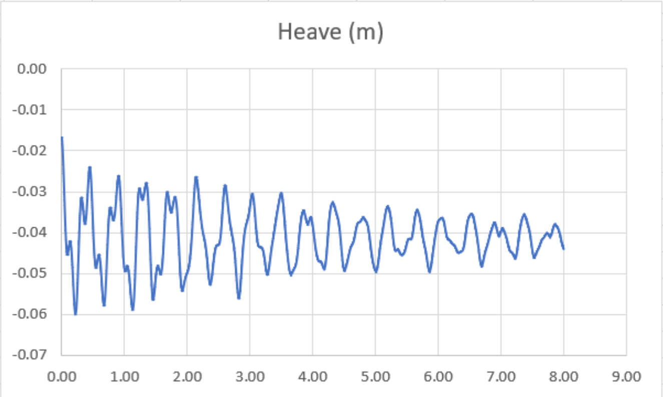


Figure 22: Heave of the Moving Wing

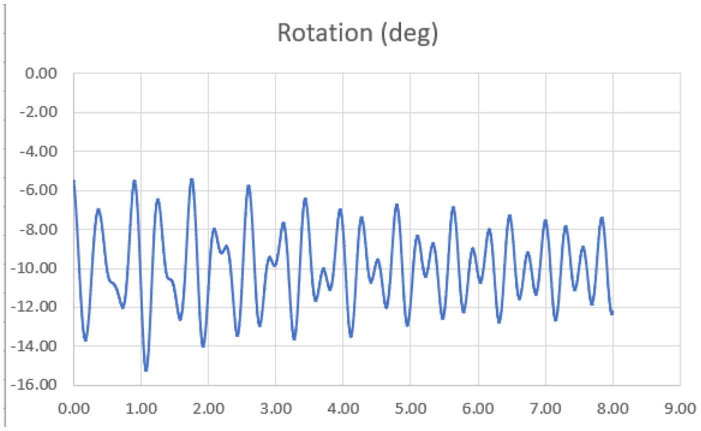


Figure 23: Rotational position of the Moving Wing

So, as an ending note we can see that a airfoil placed behind a bluff body can use the vortices in the wake of the bluff body to generate thrust and extract energy from the vortices.

References

[1] Lefebvre, Jonathan N. and Jones, Anya R., Experimental Investigation of Airfoil Performance in the Wake of a Circular Cylinder, AIAA Journal, 10.2514/1.J057468

[2] BEAL , D. & HOVER , F. & Triantafyllou, Michael & LIAO , J. & Lauder, George. (2006). Passive Propulsion in Vortex Wakes, Journal of Fluid Mechanics, 10.1017/S0022112005007925.

[3] Kianoosh Yousefi and Alireza Razeghi, Determination of the Critical Reynolds Number for Flow over Symmetric NACA Airfoils, 2018 AIAA Aerospace Sciences Meeting, 10.2514/6.2018-0818