Fluid-structure interaction of a rigid cylinder and flexible splitter plate

1 Introduction

Fluid-structure interaction (FSI) is the mutual interaction of a fluid and a solid. Pressure and viscous forces from a moving fluid, exerted on a structure, excite the structure causing it to deform in space, which in turn alters the flow field. This coupled interaction between the fluid and solid poses a challenging problem, both numerically and computationally.

FSI problems occur abundantly in both nature and industry. In nature examples include, the flow of blood through arteries, the pumping of blood in the heart, the flapping of bird wings, and the propulsion of fish through water. In industry examples include, the generation of energy through wind and tidal turbines and in more adverse circumstances the famed collapse of the Tacoma bridge.

A range of software tools are developed to solve FSI problems, and a typical test case to validate the software, is the analysis of a fluid over a rigid circular cylinder with an elastic flag. This is the test case that is considered in this workshop. The computational configuration is presented in Figure 1

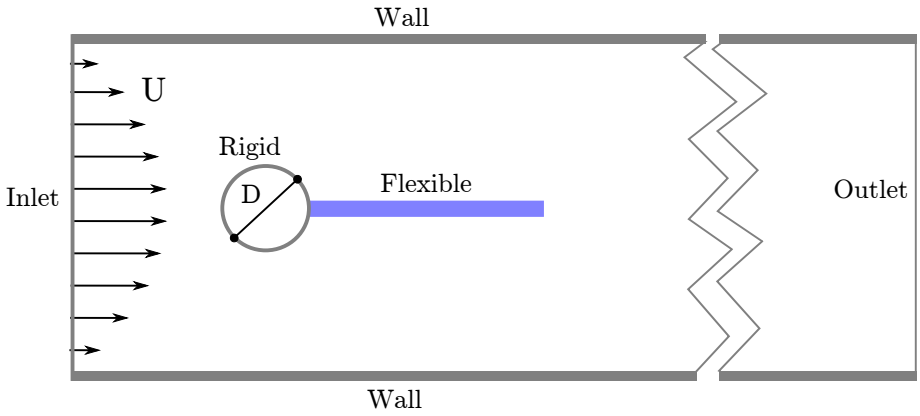


Figure 1: Computational domain.

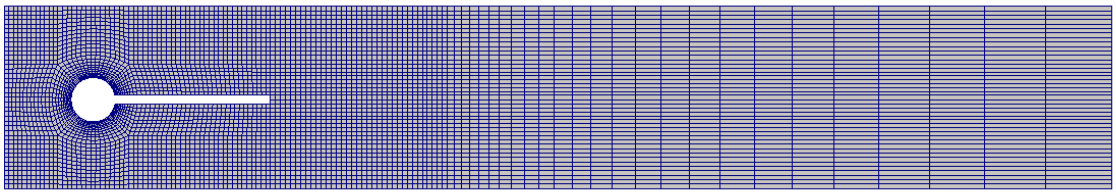
The flow over the cylinder results in a oscillatory force felt by the flexible structure, coloured in blue within the Figure 1 causing it to vibrate periodically. This is a laminar case with a Reynolds number of 200 based on the cylinder diameter (D). The walls and rigid cylinder have non-slip velocity and zero-gradient pressure boundary conditions (BCs) applied to them. The elastic flag also has a zero-gradient

pressure condition but the velocity is set to the velocity imparted on it by the structural solution. There is a fixed velocity at the inlet and a fixed pressure at the outlet. The material properties of the problem are presented in Table 1.

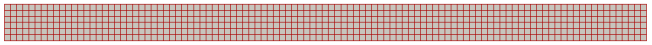
Table 1: Material properties for the structure and fluid

Property	**Symbol**	**Value**
Fluid density	ρ_f	1000
Kinematic viscosity	ν_f	0.001
Mean inlet velocity	$\overline{U}$	2
Solid density	ρ_s	10000
Poissons ratio	ν_s	0.4
Young modulus	E	1.4×10^6
Mass ratio	$M = \frac{\rho_s}{\rho_f}$	1
Reynolds number	$Re = \frac{\bar{u}D}{\nu_f}$	100
Cauchy number	$Ae = \frac{E}{\rho_f \bar{u}^2}$	1.4×10^3

The meshes that will be generated within this tutorial use OpenFOAM's blockMesh application, that builds meshes based on an input file, blockMeshDict. The fluid domain is made up of 5336 hexahedral cells, whilst the solid domain is made from 630 hexahedral elements. This is a two-dimensional problem however OpenFOAM is limited to three-dimensional elements and so the domain is only one element thick throughout. A front on view of these meshes is shown in Figure 2.



(a) Example fluid domain, containing 5336 cells.



(b) Example solid domain, containing 630 elements.

Figure 2: Example computational meshes.

2 Objectives

The key objectives of this workshop are:

1. To gain more experience using open-source software tools for simulation.
2. To practice using OpenFOAM (OpenFOAM-Extend) and ParaFEM to simulate a multiphysics problem.
3. To practice post processing results using python scripts.

3 Investigation

After setting up the test case and running the simulation we are going to undertake some qualitative and quantitative post-processing. The qualitative post-processing will be analysing and generating snapshots of the velocity and pressure contours. The quantitative post-processing will involve plotting the force experienced by the cylinder and plate and the displacement of the tip of the plate throughout the simulation.

4 Case setup

4.1 Prepare the directory

To begin with we will prepare a directory in which to run our case and store our results. We are going to use the Linux operating system so the workshop will introduce a number of commands that help to create and organise files. Firstly we must open a terminal by pressing Ctrl+Alt+t or alternatively right clicking on the desktop and selecting the open terminal option. The following command can subsequently be entered to create a directory:

```
mkdir -p ${HOME}/foam/${USER}-4.0/fsi_workshop
```

mkdir is a linux command to create a directory, where $USER and $HOME are environment variable that are replaced by your user name and home directory. To enter into the newly created directory enter the following command:

```
cd ${HOME}/foam/${USER}-4.0/fsi_workshop
```

This take us into the directory. There are currently no files in this directory, so we need to download the files required for this lab. The files already exist on the system, they need to be copied from their current location to the fsi_workshop directory.

```
cp -r ${HOME}/OpenFPCI/run/HronTurek .
```

This has copied the hronTurek directory to the current location. To view this directory use the ls command in the terminal. The ls command lists all the files and directories in the current directory. To navigate around these files the cd command can be used.

```
cd HronTurek          # Enters the HronTurek directory
cd ..                 # Moves back a directory
```

Within the HronTurek directory two key directories exist, fluid and solid. These directories contain the mesh information and solution methods used to solve the fluid problem, the solid problem and the coupling between the two. The case is ready to run, however the following two section will highlight how to set-up the initial conditions (ICs), the boundary conditions (BCs), the material properties, and the solution methods for the solid and fluid problem.

4.2 Setting up the solid domain

The governing equations for the solid domain are solved using a package called ParaFEM, however the coupling of the two is such that the problem is built using the OpenFOAM interface.

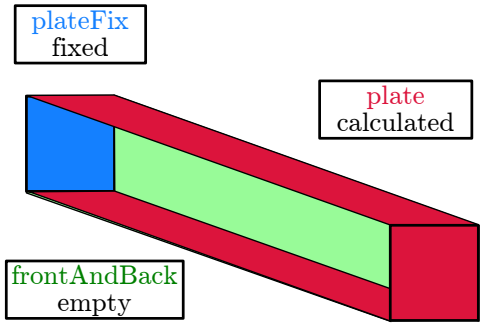
Starting with the solid domain, we can enter into the directory and view the files that are present. There are three directories, 0, constant and system. The 0 directory contains the ICs and BCs for the solid domain, constant contains information about the material properties, and the system directory contains information about solution methods.

Firstly we are going to take a quick look at the ICs and BCs in the 0 directory. It contains four files D, pointD, pointU, and pointA. These files show the cells average displacement, the displacement, velocity, and acceleration at the nodes respectively. We are going to browse the pointD file. Figure 3 shows the nodal displacement boundary conditions for the plate.

It can be seen that the backend of the plate is fixed and so the nodal displacements are set to (0 0 0). The empty condition implies there are no deformations in the z-direction and the calculated conditions tells us it is calculated during the simulation. These have been set up correctly for this case, so can be left as they are.

```
15  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
16
17  dimensions      [0 1 0 0 0 0 0];
18
19  internalField   uniform (0 0 0);
20
21  boundaryField
22  {
23      plate
24      {
25          type            calculated;
26          value           uniform (0 0 0);
27      }
28      plateFix
29      {
30          type            fixedValue;
31          value           uniform (0 0 0);
32      }
33      frontAndBackPlanes
34      {
35          type            empty;
36      }
37  }
38
39
40  // ************************************************************************* //
```

(a) Snapshot of file containing the ICs for the nodal displacements of the patches



(b) Patches and their BCs. Blue is the plateFix, red is the plate and green is the front and back patches. The front patch has been left unfilled to aid in the visualisation.

Figure 3: The displacement boundary conditions for the solid domain

However, the case is currently set-up for a different problem, so we need to change some of the material properties of the solid. In order to set the material properties, the file of interest is in the constant directory. You can use a text editor of your choice (gedit,vim,emacs) to view this file.

```
gedit constant/rheologyProperties
```

It requires the user to set the density, Young's modulus and Poisson's ratio of the problem. They are currently set for a different problem. Can you set these according to the material properties shown in Table 1?

Finally, we want to change some of the solution parameters that can be found and set in the solidProperties file found in the constant directory. A number of values need to be changed, the Newmark parameters need to be changed in accordance with Figure 4.

```
26
27      gravity (0 0 0);        Gravity is set to zero
28
29      beta 0.4;               Parameters for the Newmark method
30      delta 0.6;
31
32      PCGTolerance    1e-6;   Solution parameters for the linear and
33      PCGLimit        500;    non-linear algebraic solvers for the solid domain
34      NRTolerance     1e-5;
35
```

Figure 4: Snapshot of file containing the solution properties for the solid domain.

The Newmark time stepping scheme is used with parameters $\delta = 0.4$ and $\beta = 0.6$ for added numerical diffusivity. Geometric non-linearities are included within this problem and so they are solved

using a classic Newton-Raphson scheme with a tolerance set at $1e-5$. The linear system of equations is subsequently solved using the preconditioned conjugate gradient (PCG) method with a tolerance of $1e-6$ and with a maximum number of iterations of 500.

This completes the setup of the solid domain. To check you have completed everything for this section a quick check-list has been provided:

1. Change the material properties in **constant/rheologyProperties**
2. Change the Newmark parameters in **constant/solidProperties**

4.3 Setting up the fluid domain

The fluid domain is set-up in a similar manner to the solid domain. Navigate into the fluid directory using the cd command. The ICs for the fluid are however different than the solid. Within the 0 directory there are files that define the pressure (p), the velocity (U) and the motion smoother of the mesh (pointMotionU). The boundary conditions applied are typical of general CFD simulations, however one BC of interest is the inlet velocity. Figure 5 provides a summary of the patch names in the simulation.

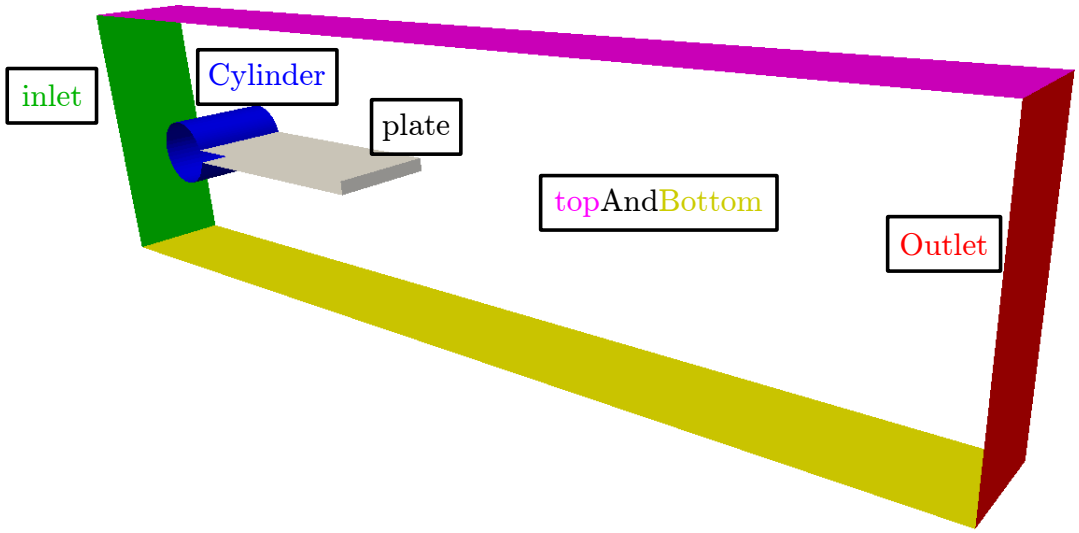


Figure 5: Summary of the fluid domain.

We need to set the inlet velocity boundary condition in the **0/U** file as follows. The velocity needs a transition period over 2 seconds, this means the velocity is slowly ramped over the first 2 seconds.

```
    inlet
    {
        type            transitionalParabolicVelocity;
        maxValue         1.5;
        n               ( 1 0 0 );
        y               ( 0 1 0 );
        transitionPeriod  2;
        boundBoxMin     ( 0 0 -0.025334 );
        boundBoxMax     ( 0 0.41 0.025334 );
        value           uniform ( 0 0 0 );
    }
```

Figure 6: Snapshot of file containing the velocity ICs for the fluid domain.

Secondly we need to set the maximum value. In this case the max value needs to be set to 1.5m/s. These values have been highlighted in Figure 6.

Secondly the fluid properties are set within the constant directory through two files, transportProperties and fluidProperties. They describe the fluids flow properties and the high level properties of the fluid solution. Snapshots of these files are provided in Figure 7

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

nu              nu [0 2 -1 0 0 0 0] 1e-3;

rho             rho [1 -3 0 0 0 0 0] 1000;

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

(a) transportProperties

```

fluidSolver icoFluid;

icoFluidCoeffs
{
    nCorrectors 2;
    nNonOrthogonalCorrectors 1;
    nOuterCorrectors 1;
}
```

(b) fluidProperties

Figure 7: Snapshots of files from the **fluid/constant** directory used to set the fluid parameters.

We want to set the density of the fluid to 1000 and the viscosity to 0.001, as specified in Table 1. Edit the **constant/transportProperties** file. Finally we want to edit the fluid solution properties. We want the case to run quickly, so we can reduce the number of nCorrectors to 2. This can be changed in **constant/fluidProperties**.

The last thing we want to do is check the time step of the case is set correctly. This is done by opening the **system/controlDict** file, and checking that deltaT is set to 2e-2. Whilst in this file it is useful to note that at the bottom there is a set of functions. These are libraries that help us output results, that we will see later.

Please use the following check-list to ensure you have completed all the step for the fluid setup:

1. Set the inlet BC in **0/U**
2. Change the transport properties in **constant/transportProperties**
3. Reduce the number of nCorrectors in **constant/fluidProperties**

4. Check the time step size in **system/controlDict**

4.4 Setting up the FSI coupling

This is the final check that needs to be completed before the case can be run. In order to solve the FSI problem using the OpenFOAM and ParaFEM we need to set the coupling properties. This is done in the fsiProperties file found in the **fluid/constant/fsiProperties** directory. This file is as follows:

```
24  relaxationFactor  0.3;
25
26  interfaceDeformationLimit 0;
27
28  outerCorrTolerance  1e-5;
29  nOuterCorr          8;
30
31  interpolatorUpdateFrequency 0;
32
33  //couplingScheme Aitken;
34  couplingScheme IQN-ILS;
35
36  couplingReuse 0;
37
38  rbfInterpolation  no;
39
40  coupled no;
41  predictor yes;
```

Relaxation of the displacements

Parameters for the strong coupling scheme

Selection of a dynamic relaxation scheme

Set the coupling and if to use the predictor
Note: The coupling is switched to yes by a functionObject

Figure 8: Snapshot of file containing the coupling information.

Again the case is setup for a different simulation the values need to be altered those shown in Figure 8. The test case uses both relaxation and a strong coupling with multiple iterations of the fluid and solid solutions each time step. The parameters for this strong coupling scheme are set with a maximum of 8 iterations (nOuterCorr) of each solver per time step and a tolerance of 1e-5 (outer CorrTolerance), i.e. if the difference between the solution of the previous iteration and the current iteration are smaller than this tolerance the simulation is allowed to continue to the subsequent time step. The relaxation factor is set at 0.3 in the first iteration of a time step and then dynamically using an iterface-quasi-newton (IQN) method, selected within the fsiProperties file.

4.5 Running the case

The setup has been complete and we are ready to run the case.

To run the case a run script has been provided. This run script contains a list of commands that can each be used separately. However the script can be run using the following command.

```
./Allrun
```

This script builds the meshes for the fluid and solid and links the cases with a series of soft links. It then runs the executable **fsiFoam**. This executable uses a strong coupling between the fluid and solid. The case takes approximately 10 minutes to run to completion. This script stores the output of each command in a log file. The executable is stored in **log.fsiFoam**. During the simulation the end of the log file can be viewed using the tail command:

```
tail -f log.fsiFoam
```

This command shows the last few lines of the file, constantly updating in real time. The simulation can take up to 10 minutes to run to completion however we can begin to plot results before the simulation has completed.

4.6 Post processing

4.7 Quantitative analysis

During the simulation two files are output, that can be found in the directory **history/0**. These files are labelled **force.dat** and **point_356.dat**. The **force.dat** file contains the integration of force over the cylinder and plate at each time step and the **point_356.dat** contains the displacement of a node at the tip of the beam.

Plotting this data can be done using python. Two scripts are provided, **plotDisplacement.py** and **plotForces.py** that read in the data of the simulation and output pdf images of the displacements and forces in the x-direction and y-direction. These can be called using the following commands

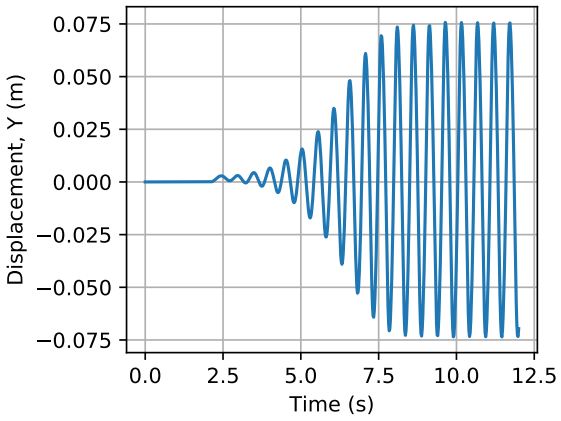
```
python plotDisplacements.py
python plotForces.py
```

From this directory view the pdf files that are created, for example, to view **dispY.pdf**, use the command:

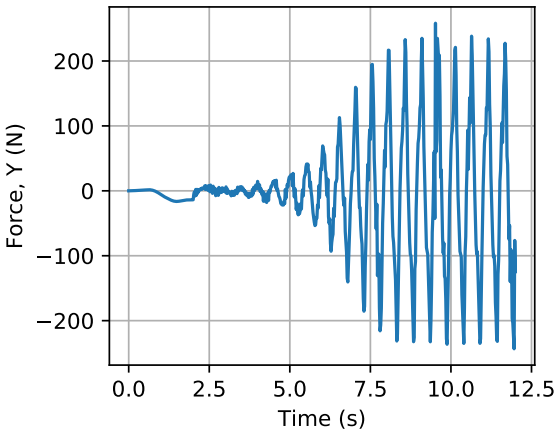
```
gnome-open dispY.pdf
```

Figure 9 shows the plots of the displacement in the Y direction and the sum of the forces over the plate and cylinder in the y direction.

A few lines have been commented out within the python scripts, use a few minutes to alter some of the parameters within the plotting scripts and uncomment these lines and see how they change the plots.



(a) Displacement in the Y direction of the tip of the plate



(b) Force in the Y direction over the cylinder and plate

Figure 9: Quantitative analysis of the results output from the simulation.

4.8 Qualitative analysis

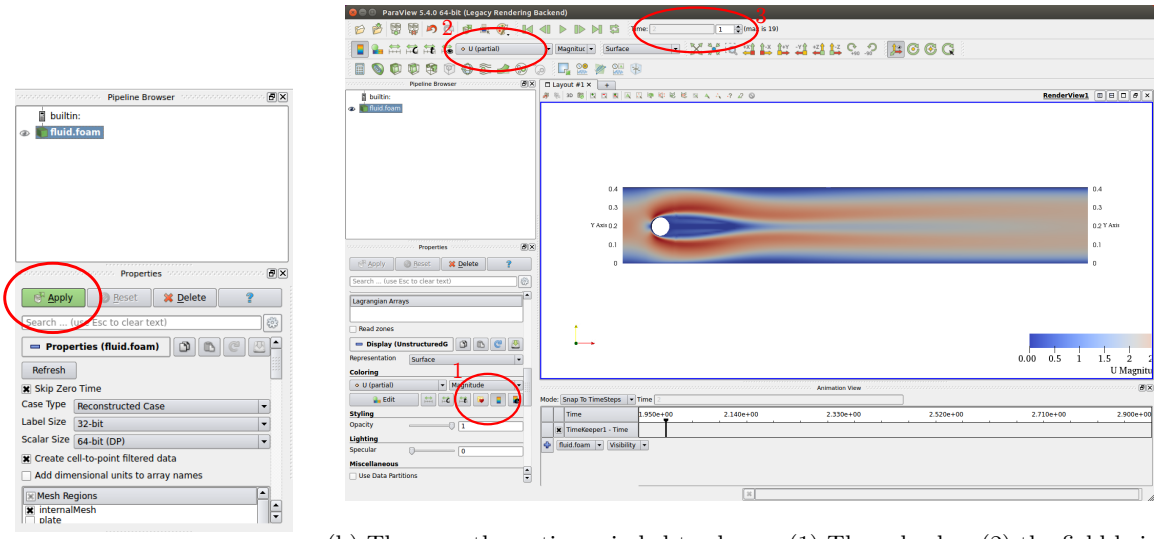
We can begin by considering the velocity and pressure field at the point at which the structure is allowed to vibrate, t=2.0s. To plot the velocity and pressure fields, open Paraview using the following command:

```
paraFoam -nativeReader
```

Once Paraview has opened, you will be asked to select apply in the pipeline browser on the left-hand side of the page. Then use options 1,2 and 3 shown in Figure 10 to change the color bar, the field being displayed in the screen and the time that is being shown.

If the above is done the images of pressure and velocity can be output at different time frames. In doing so we can see how the deforming structure effects the velocity and pressure fields. If we want to visualise vorticity we can use the OpenFOAM command **vorticity**. Close Paraview type vorticity into the terminal and reopen Paraview. Figure 11 shows plots of velocity, pressure and vorticity in the fluid domain at time=11s

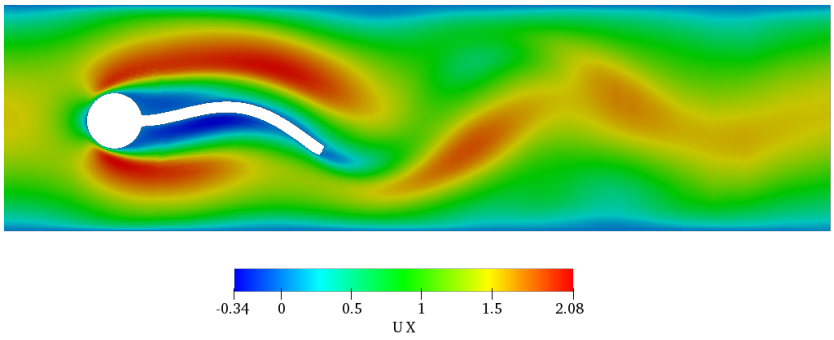
Spend some time experimenting with the parameters in Paraview. Hint to output a video click on file in the top left and select the outputanimation command.



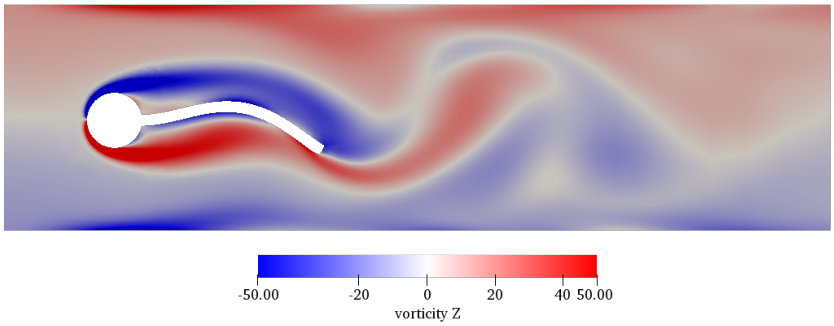
(a) Firstly hit the apply button

(b) Then use the options circled to change (1) The color bar (2) the field being displayed and (3) the time that is being displayed

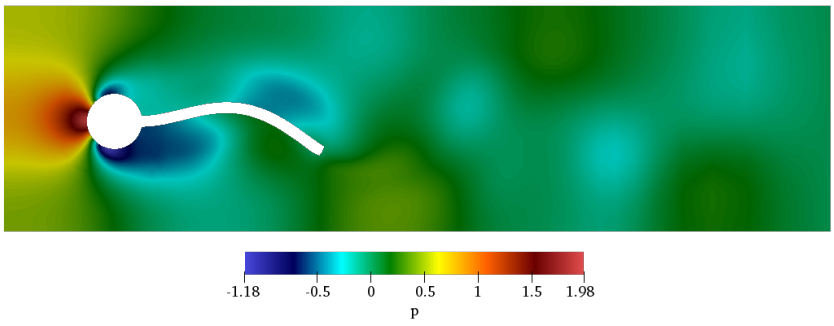
Figure 10: Viewing the velocity contours using Paraview.



(a) Velocity



(b) Vorticity



(c) Pressure

Figure 11: Viewing the velocity, vorticity and pressure contours using Paraview.