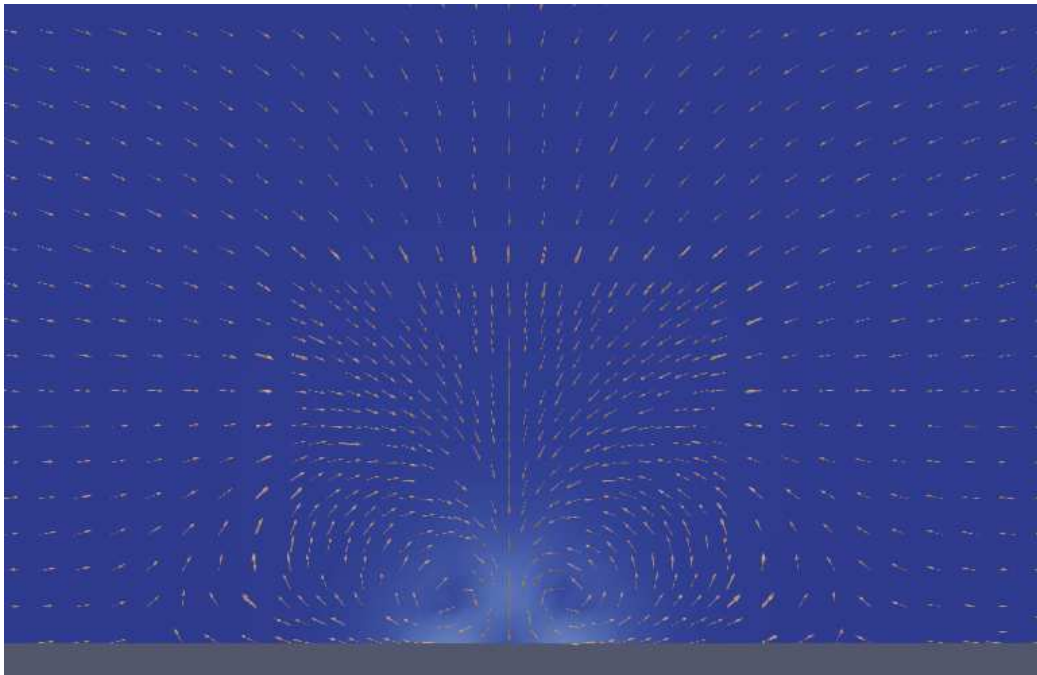




**Northumbria
University**
NEWCASTLE

KB6003 Vehicle Aerodynamics

Tutorial 4



Ahmed Body Post Processing

Date: 6th November 2018

Contents

TABLE OF FIGURES	2
TABLE OF TABLES	2
1 INTRODUCTION	3
2 FLOW FIELD DATA	3
2.1 CONTOURS	3
2.2 VECTOR PLOTS	6
2.3 EXPORTING FLOW FIELD DATA.....	8
3 FORCE AND COEFFICIENT CALCULATION.....	8
3.1 FORCE AND FORCE COEFFICIENTS SETUP	8
3.2 RUN-TIME FORCES AND COEFFICIENTS CALCULATIONS.....	10
3.3 FORCES USING PARAVIEW	10
APPENDIX A: FORCE AND COEFFICIENT CALCULATION (RUNTIME)	13
3.4 FORCE AND FORCE COEFFICIENTS SETUP	13

Table of Figures

Figure 2.1: Extract block.....	4
Figure 2.2: Insert a contour plot.	5
Figure 2.3: Contour plot with U representation.....	5
Figure 2.4: Create 'Surface Vectors'.	6
Figure 2.5: Initial 'Glyph' plot.	7
Figure 2.6: Finished 'Glyph' plot.	7
Figure 2.7: Exporting data to CSV.....	8
Figure 3.1: Generate surface normals.....	11
Figure 3.2: Add 'Calculation' filter with $p \cdot \text{Normals}_X \cdot 1.225$ formula.	12
Figure 3.3: Drag force calculation using ParaView.....	12

Table of Tables

Table 2.1: Parameters for 'Glyphs'	6
--	---

1 Introduction

This tutorial would look at post processing further with flow field visualisation, exporting flow field data, processing of lift, drag coefficients, calculating forces.

This tutorial will assume the completion of previous tutorials, therefore explanation of common steps with figures (such as buttons) have been omitted.

Due to the limited time available during tutorials, a finer case with a solution has been prepared. This case has a limited number of time steps saved and can be downloaded from Github under Tutorial4.

<https://github.com/NU-Aero-Lab/OpenFOAM-Cases/tree/master/KB6003>

Navigate to the unzipped case folder ('ahmed4'), open a terminal inside it and continue below.

2 Flow Field Data

In this section, different representations of flow field sections (clips) would be looked at with methods of exporting data to files.

2.1 Contours

This section will look at applying the 'Contour' filter and representing data on 'Clips'. To proceed;

- 1) Open ParaView by;

```
paraFoam -builtin
```

- 2) Make sure the 'internalMesh', 'ahmed1', 'ahmed2' and 'Reconstructed Case' is selected from the 'Properties' tab and select 'Apply'.
- 3) Skip to the last time step.
- 4) Select 'Extract Block' from the 'Filter>Alphabetical' menu, select 'ahmed1' and 'ahmed2' in 'Block Indices' and click 'Apply' (figure 2.1).

Tip: This is an alternative to loading another instance of the case (another '<case>.foam') which reduces memory usage.

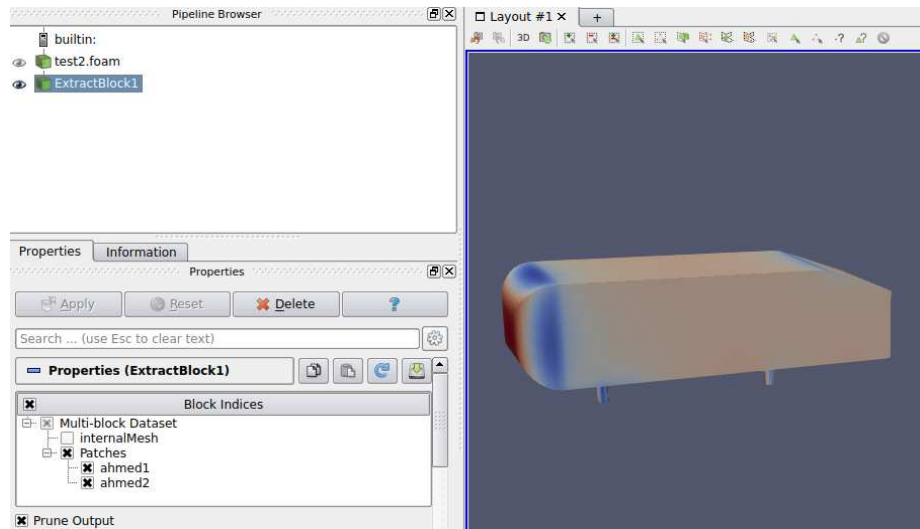


Figure 2.1: Extract block.

- 5) Select the "<case>.foam" in the pipeline browser and make a 'Clip' from the 'Common' toolbar (similar to mesh visualisation in tutorial 2 and general visualisation in tutorial 3).
 - 6) Select 'x' as the normal-to-axis and drag the plane to an area of interest (such as the wake at $x=0.7$).
- Tip:** Once fixed, you can hide the plane by unticking 'Show Plane'.
- 7) Go to 'Filters>Alphabetical>Countour' filter with default values and click apply (figure 2.2). This should output a countour plot similar to figure 2.2 below.

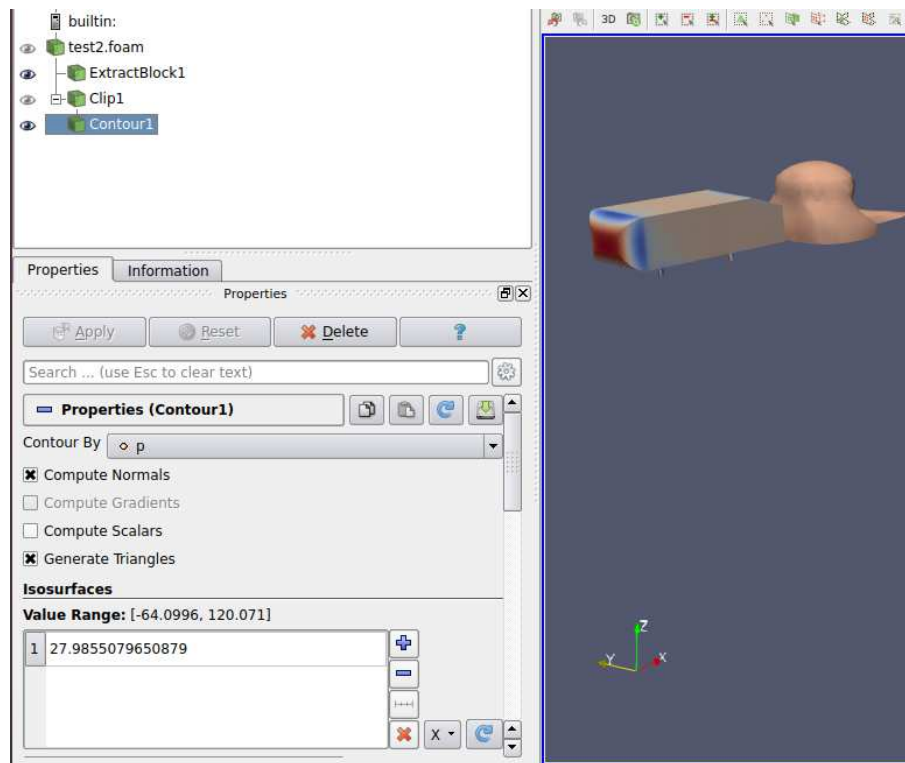


Figure 2.2: Insert a contour plot.

- 8) Change the 'Coloring' to U and adjust the range if needed.

This creates an isosurface of pressure magnitude of the selected range and shows the velocity in it as shown in figure 2.3 below.

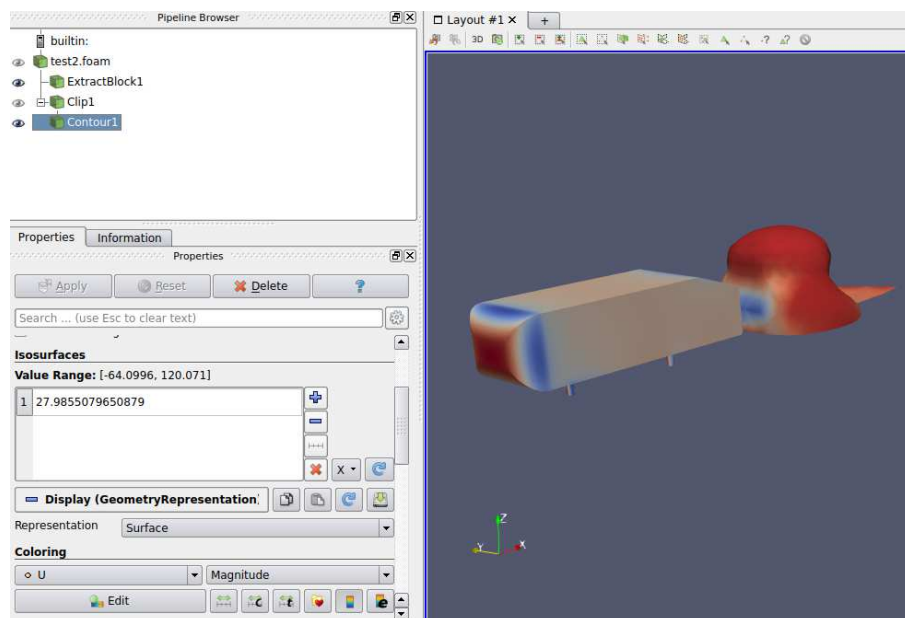


Figure 2.3: Contour plot with U representation.

Try this at different planes, areas and other variables. For vectors such as U a scalar needs to be calculated. This can be done using the 'Calculator' filter (section 3 for example use of 'Calculator' filter.)

Tip: These representations can be exported by preferably by selecting 'Save Screenshot' from 'File' menu or grabbing a screenshot.

2.2 Vector Plots

These can be created using the 'Glyphs' filter which reads from the selected 'Vectors' field. There are multiple 'Glyph Types' and for this tutorial, only 'Arrow' type would be looked at with scaling proportional to vector magnitude. This provides a clear vector plot.

- 1) Follow the steps 1-5 from section 2.1 BUT create a 'Slice' with $x=1.9$.
- 2) Click on the slice, add 'Surface Vectors Filter' from 'Filters>Alphabetical' menu, select 'U' from the 'Select Input' dropdown and click 'Apply' (figure 2.4).

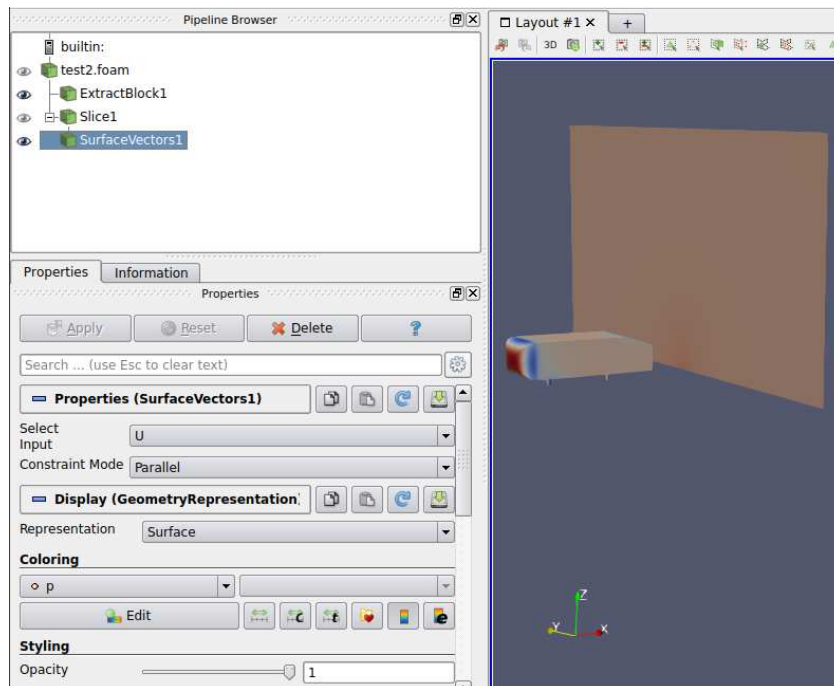


Figure 2.4: Create 'Surface Vectors'.


- 3) Select the 'SurfaceVecots1' filter and click on 'Glyphs' () from the 'Common' toolbar.
- 4) Select the parameters in table 2.1 below for the 'Glyphs'.

Table 2.1: Parameters for 'Glyphs'

Parameter	Value
Glyph Type	Arrow
Scalar	None
Vectors	U
Scale Mode	Off
Scale Factor	0.05 (adjust if needed for smaller arrow size)
Maximum Number of Points	5000 (adjust as required)
Colouring	Solid Color (change if required)

- 5) Click 'Apply'. The plot should look similar to figure 2.5 below.

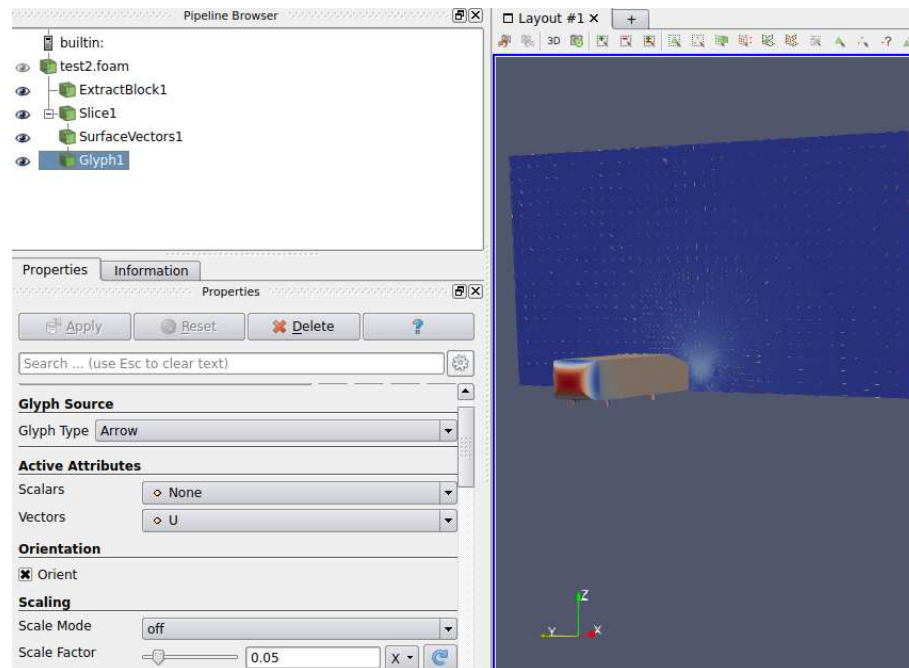


Figure 2.5: Initial 'Glyph' plot.

- 6) The 'ExtractBlock1' created for reference purposes can be hidden and the view can be adjusted to obtain a plot similar to figure 2.6.

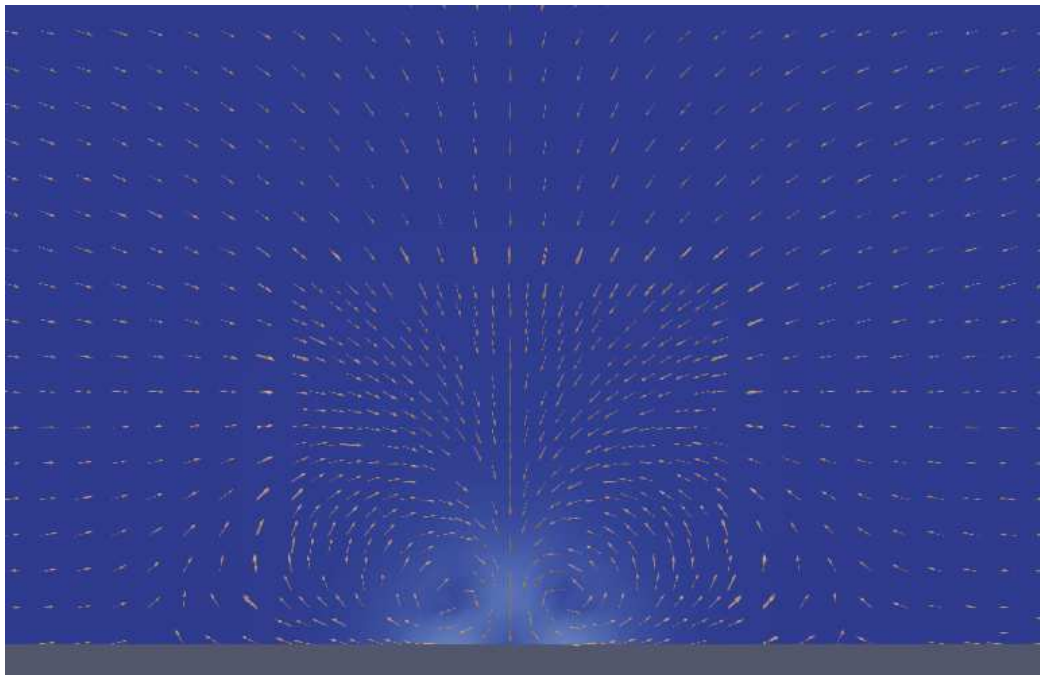


Figure 2.6: Finished 'Glyph' plot.

2.3 Exporting Flow Field Data

The data in above clippings can be exported to different formats such as CSV, for further analysis and/or plotting. This can be done by;

- 1) Selecting the desired 'Clip' from the 'Pipeline' browser.
- 2) Click 'File>Save Data' and chose a format of your liking (preferably CSV).
- 3) Click 'Save' and select the desired 'Precision and 'Notation' then click 'OK'.

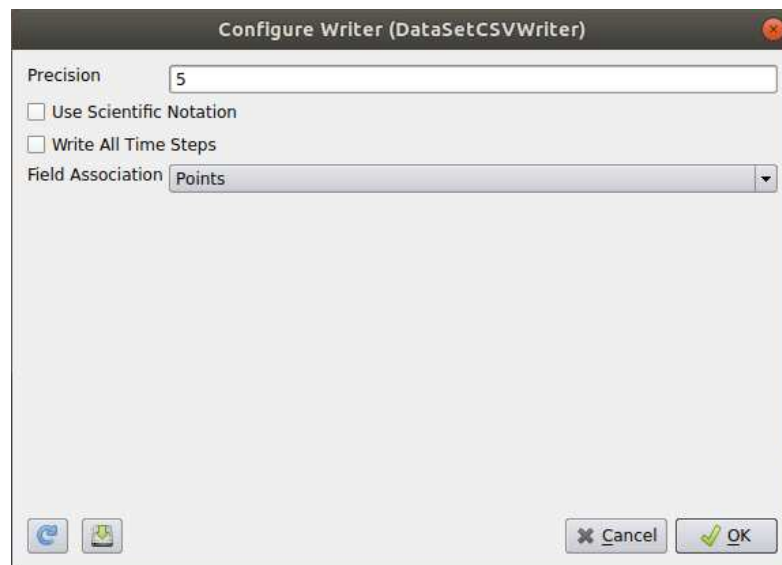


Figure 2.7: Exporting data to CSV.

- 4) Alternatively, a high quality graphical image can be saved by selecting 'File>Export Scene' as well.

3 Force and Coefficient Calculation

During tutorial 3, force and force coefficient processing was omitted to simplify the case set-up. This section will look at how to add force and force coefficient calculations to the simulation. These would then be extracted.

3.1 Force and Force Coefficients Setup

In order to calculate these, functions are placed in the 'systems' directory and then referred to from the 'functions{}' section in 'controlDict'.

- 1) Place the following code under the 'functions{}' section.

```
#include forceCoeffsIncompressible;  
#include forcesIncompressible;
```

- 2) These refer to the 2 dictionaries with the same name. These include the parameters for calculations.

forcesCoeffsIncompressible:


```

type          forceCoeffs;

libs          ("libforces.so");


writeControl   timeStep;
timeInterval   1;


log           yes;


patches        ("ahmed.*"); // patches to include
rho           rhoInf;       // Indicates incompressible
rhoInf        1.225;        // Redundant for incompressible
liftDir        (0 0 1); // Lift direction
dragDir        (1 0 0); // Drag direction
CofR          (0 0 0); // Axle midpoint on ground
pitchAxis      (0 1 0);
magUInf        30;
lRef           0.288; // Wheelbase length
Aref           0.115; // Estimated frontal reference area

```

Note: The coefficients can be calculated per patch ('ahmed1') or together using a wildcard (such as 'ahmed.*' to include 'ahmed1', 'ahmed2', 'ahmed3', etc.).

forcesIncompressible:

```

type forces;
functionObjectLibs ("libforces.so");

enabled        true;
log           yes;

writeControl   timeStep;
writeInterval   1;

patches        ("ahmed.*"); // patches to include

```

```
pName      p;  
Uname      U;  
  
rho        rhoInf;  
rhoInf      1.225;  
  
CofR       (0 0 0); // Centre of rotation
```

- 3) Run the simulation.
- 4) When the simulation has finished, enter the following command to generate the coefficients and forces.

```
simpleFoam -postProcess -func forceCoeffsIncompressible  
simpleFoam -postProcess -func forcesIncompressible
```

This will generate a 'postProcessing' folder which includes a 'forceCoeffs.dat' and 'forces.dat' file inside its subdirectory(s).

3.2 Run-time Forces and Coefficients Calculations

Alternative to section 3.1, the same forces and coefficients can be calculated during run-time. Run time processing enables users to calculate values such as force coefficients, forces, yPlus, etc. during the simulation. This method will be used as opposed to post process extraction. It allows the user to monitor the solution and modify the simulation if needed before completion (useful in long resource intensive simulations). This method can be seen in Appendix A.

3.3 Forces Using ParaView

Forces can also be calculated using ParaView by calculating the pressure normal to surfaces and integrating them. In order to proceed;

- 1) Open ParaView by;

```
paraFoam -builtin
```

- 2) Make sure the 'internalMesh', 'ahmed1', 'ahmed2' and 'Reconstructed Case' is selected from the 'Properties' tab and select 'Apply'.
- 3) Select 'Extract Block' from the 'Filter>Alphabetical' menu, select 'ahmed1' and 'ahmed2' in 'Block Indices' and click 'Apply' (figure 2.1).

Tip: This is an alternative to loading another instance of the case (another '<case>.foam') which reduces memory usage.

- 4) Select 'Extract Surfaces' from the 'Filter>Alphabetical' menu and click 'Apply' to extract surfaces.

- 5) Now select 'Generate Surface Normals' from the 'Filter>Alphabetical' menu.
- 6) Set the 'Feature Angle' to 15 and select 'Compute Cell Normals' and click 'Apply' (figure 3.3).

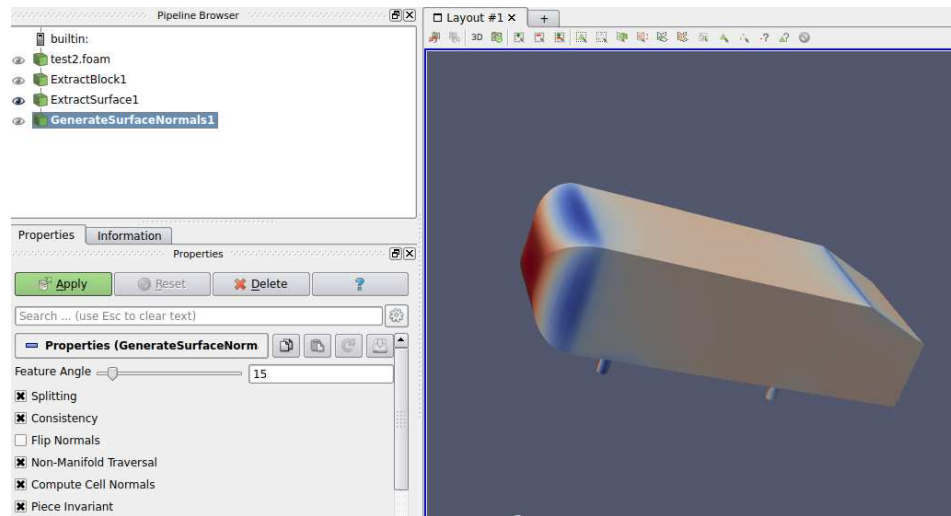


Figure 3.1: Generate surface normals.

This should generate a new variable called 'Normals'.

- 7) Click on the 'Calculate' (🧮) filter from the 'Commons' toolbar.
- 8) Select 'Attribute Mode' as 'Cell Data' from the dropdown.
- 9) Enter the following formula: $p * \text{Normals}_X * 1.225$

Note: 1.225 is the 'rho' (density).

This would output the pressure normals on the surfaces in x positive direction.

- 10) Name the result as 'pX'.

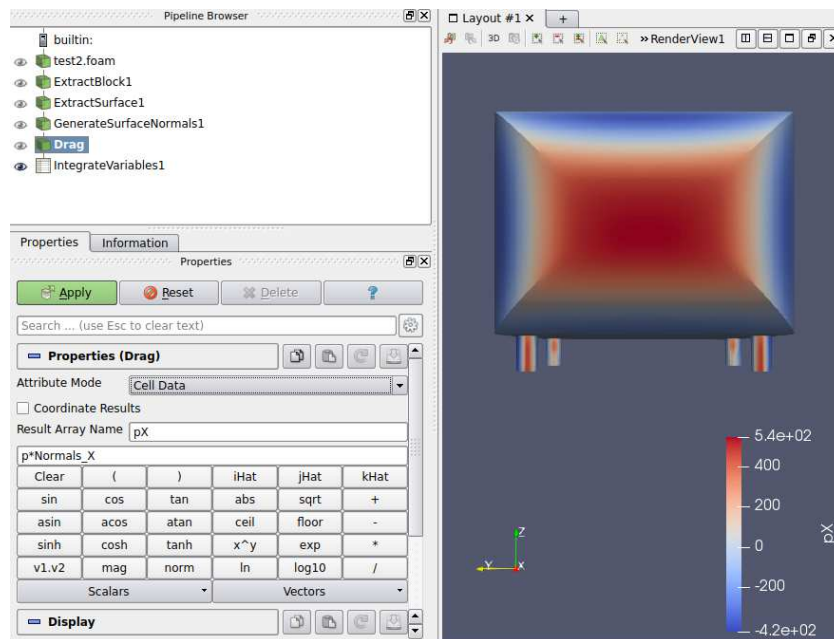


Figure 3.2: Add 'Calculation' filter with $p * \text{Normals}_X * 1.225$ formula.

The values now need to be intergrated to obtain the forces.

- 11) To intergrate, select 'Intergrate' from the 'Filter>Alphabetical' menu.

A new pane would appear as shown in figure 3.5 below. When the 'Cell Data' are selected from the drop down, the force value can be seen under 'pX'.

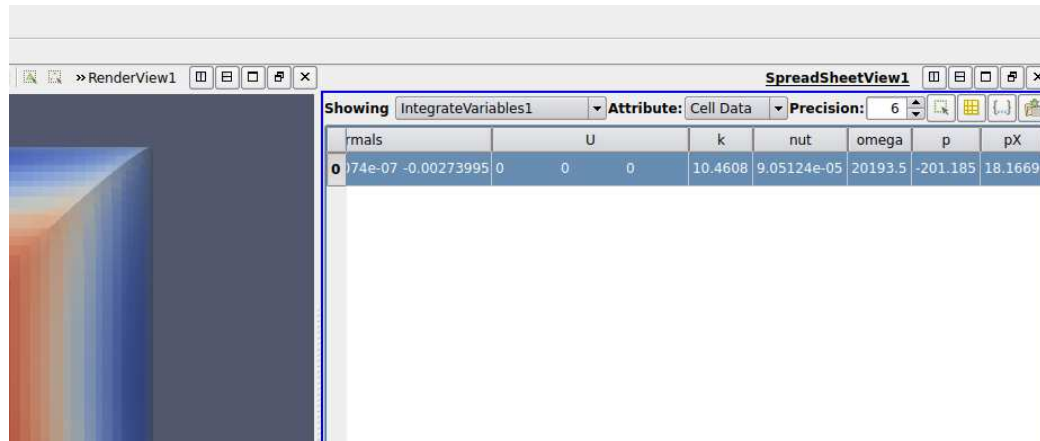


Figure 3.3: Drag force calculation using ParaView.

Note: The values may differ form the figure.

Similar to above, down-force can be calculated by replacing 'Normals_X' in step 9 by 'Normals_Z'.

Appendix A: Force and Coefficient Calculation (Runtime)

3.4 Force and Force Coefficients Setup

In order to calculate these, functions are placed in the 'functions{' section in 'controlDict'.

- 1) Place the following code under the 'functions{' section.

For force coefficients:

```
forcesCoe_object
{
    type                forceCoeffs;

    libs                ("libforces.so");

    writeControl        timeStep;
    timeInterval        1;

    patches ("ahmed.*"); // Patch name

    enabled             true;
    log                 yes;

    magUInf             33; // Free-stream velocity
    lRef                0.288; // Reference length (wheelbase)
    Aref                0.115; // Reference frontal area

    rhoInf              1.225;
    rho                 rhoInf;

    liftDir              (0 0 1); // Lift direction
    dragDir              (1 0 0); // Drag direction

    CofR                (0 0 0); // Centre of rotation
    pitchAxis            (0 1 0);

}
```

Note: The coefficients can be calculated per patch ('ahmed1') or together using a wildcard (such as 'ahmed.*' to include 'ahmed1', 'ahmed2', 'ahmed3', etc.).

For forces:

```
forces_object
{
    type forces;
    functionObjectLibs ("libforces.so");

    enabled             true;
    log                 yes;
```

```

writeControl    timeStep;
writeInterval    1;

patches    ("ahmed.*");

pName        p;
Uname        U;

rho        rhoInf;
rhoInf        1.225;

CofR        (0 0 0);

}

```

- 2) Run the simulation. This will generate a 'postProcessing' folder which includes a 'forceCoeffs.dat' and 'forces.dat' file inside its subdirectory(s).
- 3) These can be plotted in a similar manner to residuals in tutorial 3 by using the following script below;

For force coefficients:

```

set term x11 persist

set multiplot layout 1,2

# Plot Cd
set size 1, 0.5

set xlabel 'time'
set ylabel 'cd'
plot [10:][]
'postProcessing/forcesCoe_object/0/forceCoeffs.dat' using
1:3 with line notitle

#Plot Cl
set size 1, 0.5
set origin 0, 0.5

set xlabel 'time'
set ylabel 'cl'
plot [10:][]
'postProcessing/forcesCoe_object/0/forceCoeffs.dat' using
1:4 with line notitle

unset multiplot

pause 2
reread

```

For forces:

```
set title "Forces"
set xlabel 'Iteration'
set ylabel 'Force'
plot '<sed -e "s/[(),]//g"
postProcessing/forces_object/0/forces.dat' using 1:2 with
lines title 'fx',\
'<sed -e "s/[(),]//g"
postProcessing/forces_object/0/forces.dat' using 1:3 with
lines title 'fy',\
postProcessing/forces_object/0/forces.dat' using 1:3 with
lines title 'fz',\

pause 2
reread
```

Note: The 'sed -e "s/[(),]//g" forces/0/forces.dat' command is used to replace round brackets by nothing. This formats the file to a readable format by 'gnuplot'.

The forces are saved in vector format with (x y z) in the forces.dat file with which the type of force can be identified (such as 'x' positive direction corresponds to drag).