

*Student: Giannoni Alberto*

*Professor: Ing. Jan Pralits, Advanced Fluid Dynamics Course*

*Co-Professor: Ing. Matteo Colli*

## AERODYNAMIC OPTIMIZATION OF REAR AND FRONT FLAPS ON A CAR

### ABSTRACT

*In this work we would like to investigate performances of four flaps installed on a real-size car through the OpenSource CFD Software, Openfoam. The purpose is to evaluate  $C_L$  and  $C_D$  in order to reach the best compromise.*

### 1 - INTRODUCTION

In the automotive sport world, speed has been increased a lot in the last decades but at the same time cars become lighter. To ensure sufficient grip at high speed, especially during cornering, aerodynamic plays an important role because it allows to have a downforce just when it's needed avoiding the addition of permanent weight. The downforce is a vertical force generated by an air flow that runs over an airfoil with a negative incidence. During last years engineers developed many solutions to create downforce but the most effective seemed to be flaps or spoilers. As usually these solutions initially were introduced in Formula 1 cars and then started to be available for sportcar on sale. There are two different types of flaps: fixed and moving. A fixed spoiler is cheaper and easy to produce but the drawback is that it only has one angle of incidence. This is bad because even if a flap increases downforce very much, it introduces a resistance in the airflow generating more drag. This is true also at low speed when a flap would not be necessary. That's why a moving flap is a smarter solution because the control unit of the car, reading some fundamental parameters like velocity, decides which is the best angle to guarantee downforce with the acceptable increase of drag. This system could be used not only for rear flaps but also for front ones giving to the car more stability.



Figura 1-2 - Pagani Huayra

## 2 - GEOMETRY

Before starting to talk about the CFD part, a geometry is required. It is inspired by a real car (shown in the above figure) with two little flaps in the front and two in the back. The size of the car is approximately the same but we decide to remove the wheels and any other detail that should not influence so much aerodynamic coefficient in order to keep the geometry as simple as possible because, as we will see in a while, it is already an heavy simulation from a computational point of view. The analysis has been done with two different incidence angles in the front and three in the back which lead to six different combination and so six different geometries and six CFD simulation. Geometries were created in PTC Creo Parametric 2.0 (ProEngineer) with this setup information:

Length:	4.81 m
Height:	1.16 m
Width:	2.03 m

The angles we have chosen are  $40^{\circ}$ - $50^{\circ}$  degrees for the front flaps and  $15^{\circ}$ - $30^{\circ}$ - $40^{\circ}$  for the back.

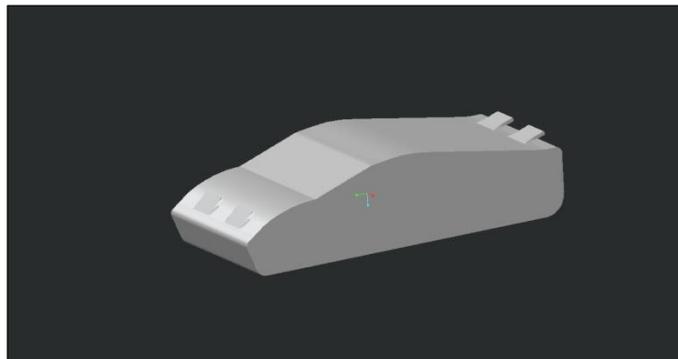


Figura 3 - Model of the car in Creo

After that every single geometry was passed in Salome in which we could orientate the axis in the proper way and define the patch of the car and finally save the output file as an .stl file which is the format that Openfoam reads as an input geometry file.

## 3 - CFD SIMULATION

### 3.1 Preliminary consideration

OpenFaom is an opensource code that deals with Navier Stokes equation: it solves momentum equation giving values for velocity and pressure both in laminar or turbulent regime. There are different turbulence model available and they are essential to solve the toughest problems of fluid dynamics. We chose to use the simplest one that is to say RANS model in which the attention is focused on the mean flow and the effect of turbulence on mean flow properties. To solve the extra terms that appear in the Reynolds averaged Navier-Stokes equations, we used the  $k-\omega$  SST model which has the advantages of  $k-\omega$  in the boundary layer and switches to  $k-\epsilon$  model in the freestream. A spatial grid of the control volume is required and refining it lead to a more accurate solution.



Other fundamental parameters are:

---

$V = 27.78 \text{ [m/s]} (100\text{km/h})$		$Re \approx 7450000$
$L \approx 4 \text{ [m]}$		
$\nu = 1.5 \times 10^{-5} \text{ [m}^2\text{/s]}$		

---

This is a very high Reynolds number to deal with and certainly we're in a turbulent regime. To make a truthful simulation we couldn't take a lower velocity,  $\nu$  is the kinematic viscosity of air and  $L$  is more or less the length of the car as vehicle aerodynamic suggest as a characteristic value.

### 3.3 Mesh

Before solving the equation we need a grid of points of discretization. For complex geometries, the mesh generation utility *snappyHexMesh* can be used. The *snappyHexMesh* utility generates 3D meshes containing hexahedra and split-hexahedra from a triangulated surface geometry in Stereolithography (STL) format. The mesh is generated from a dictionary file named *snappyHexMeshDict* located in the system directory and a triangulated surface geometry file located in the directory `constant/triSurface`. First of all we had to define a background mesh, so we use the standard tool of OpenFoam *blockMesh*. This control volume is very big indeed it's 4 cars long, 2 cars wide and 2 cars high because we want the wake to be completely developed. Then we performed cells splitting using sHM: as we can see in the image below the mesh was refined focusing on some particular parts such as the ground, the car and the wake zone. It's extremely important to have the best refinement as possible in these parts because some fundamental values of the post-processing (like *Yplus*, as we will see) reach correct value only with a properly solution refinement. For the wake zone a refinement box was added and this includes the car as well.

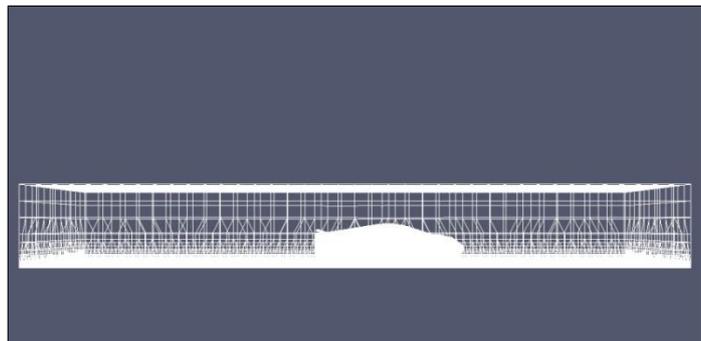


Figura 7 - The blockMesh volume



Figura 8 - A slice of the final mesh

The mesh is the result of tens of trials following the principle of trial and error and took the majority of the time for the project. Many times the mesh created didn't lead to a convergence of the solution and needed to be change considering around 30 minutes only to run the mesh and 3 hours to run the solver and see if the mesh was right. To reach satisfactory results we inserted in *snappyHexMesh Dict* some layers around the car and the ground, following their geometry, and this helped very much in modelling the critical zone between the ground and the underneath part of the car.

After the execution of *snappyHexMesh* 3.9 million of cells were obtained. Before running the solver we did an additional refinement of the cells near the ground and the car using the command *refineWallLayer 0.5*: that is an automatic division in two cells of the first cell next to a patch and permits an even better resolution in those zones.

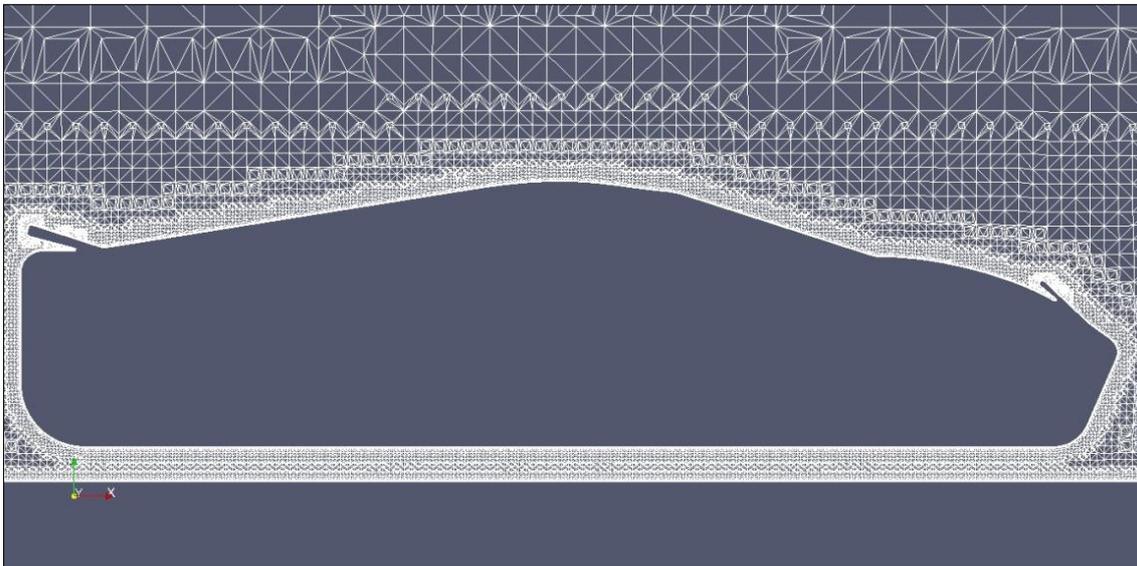


Figura 9 - A zoom of the mesh around the car

### 3.4 Solver

As already mentioned we have a RANS problem with a  $k-\omega$  SST model. SimpleFoam was the solver chosen for the calculation: it solves incompressible, steady state turbulent flow, that is to say it just aims to reach a converged solution in a set number of iterations; a steady state solver is fine for finding out Drag and Lift.

### 3.5 $C_L$ and $C_D$

In order to compute the lift and drag coefficients, we can use *forceCoeffs* functionObject. The functionObjects are general libraries that can be attached run-time to any solver, without having to re-compile the solver. We need to add the below part to the *system/controlDict* and modify the values relevant to our case. In this function, the *liftDir* and *dragDir* are defined and the *magUInf* denotes the magnitude of the freestream velocity.

```

forceCoeffs1
{
    type            forceCoeffs;

    functionObjectLibs ( "libforces.so" );

    outputControl   timeStep;
    timeInterval    1;

    log             yes;

    patches         ( Huayra );
    rhoName         rhoInf;
    rhoInf          1.2;
    liftDir         (0 0 1);
    dragDir         (-1 0 0);
    CofR            (0.99 0 0);
    pitchAxis       (0 1 0);
    magUInf         27.78;
    lRef            3.5;
    Aref            2;
}

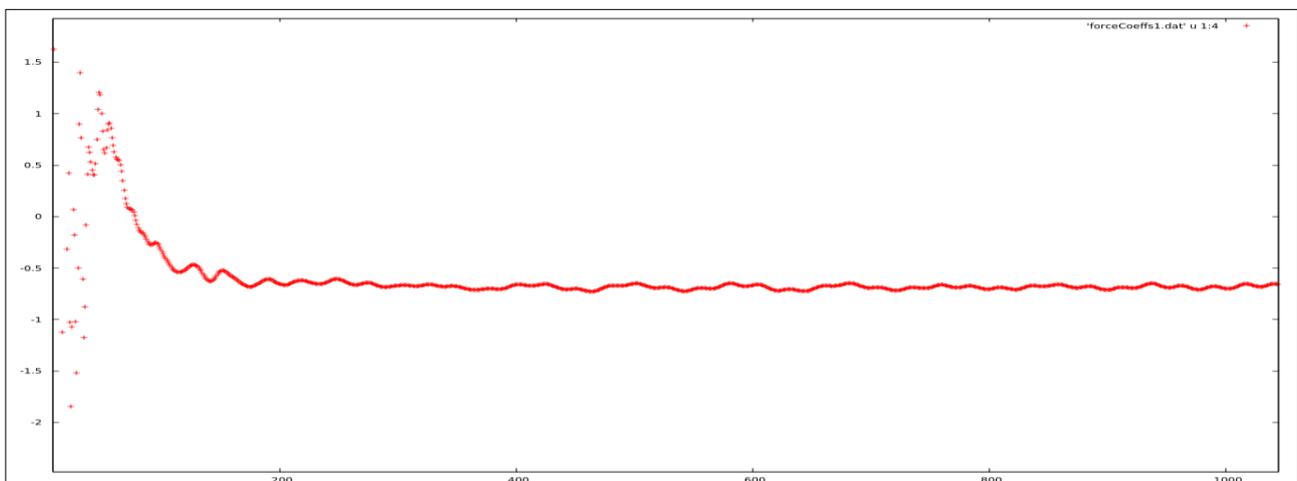
```

Figura 8 - Force coefficient D1ct

## 4 - RESULTS

We analysed six cases, all with the same mesh that gave satisfactory results for the first case which is the 40°-15° degrees angle one. Our goal for every case was to find stable values of  $C_L$  and  $C_D$  that we used at the end of the project to make comparisons. After the simulation every model was checked to verify if the stability of the solution was guaranteed and besides to see if the boundary layers were modelled in a good way.

One way to control the convergence of the solution is plotting  $C_D$  and  $C_L$  as the iteration increases using gnuplot. To be short it's reported only the graph for the first geometry, the other ones were similar: there is an initial zone in which they fluctuate a lot and then they tend to converge to a specific value.

Figura 10 -  $C_L$  with iterations

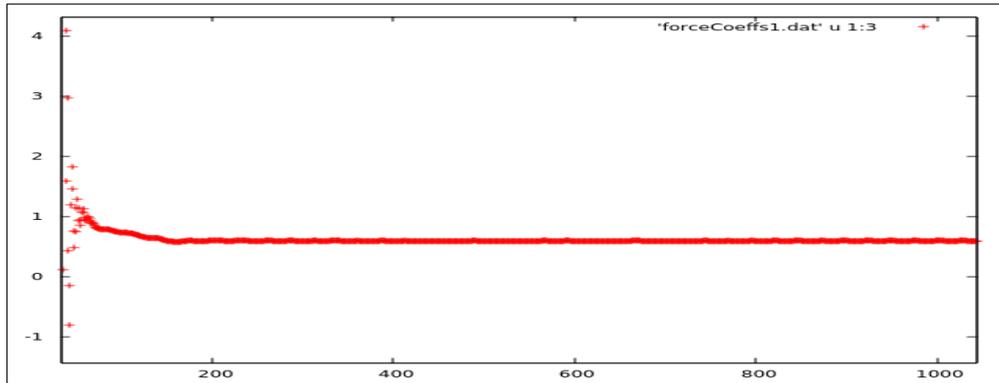
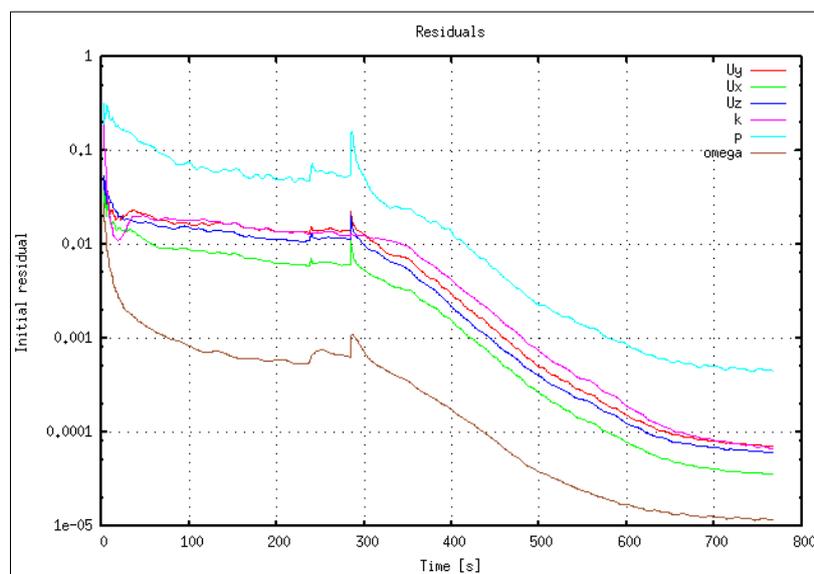


Figura 11 -  $C_D$  with iterations

Another way to check the accuracy of the solution are the residuals which represent the mistakes of it, and they must have an horizontal trend not to have a jumping solution. So we ensured that these values went to a convergence even though the absolute values were a little bit higher than classical values but this is due to the compromise we had to deal with between a huge volume (and mesh) and computer performances.



Moreover a boundary layer optimization was carried out based on the  $y^+$  parameter. This is defined as wall distance units , using the following equation:

$$y^+ = \frac{\rho \times U_\tau \times y}{\mu}$$

Where  $y$  is the distance to the first cell centre normal to the wall, and  $U_\tau$  is the friction velocity and

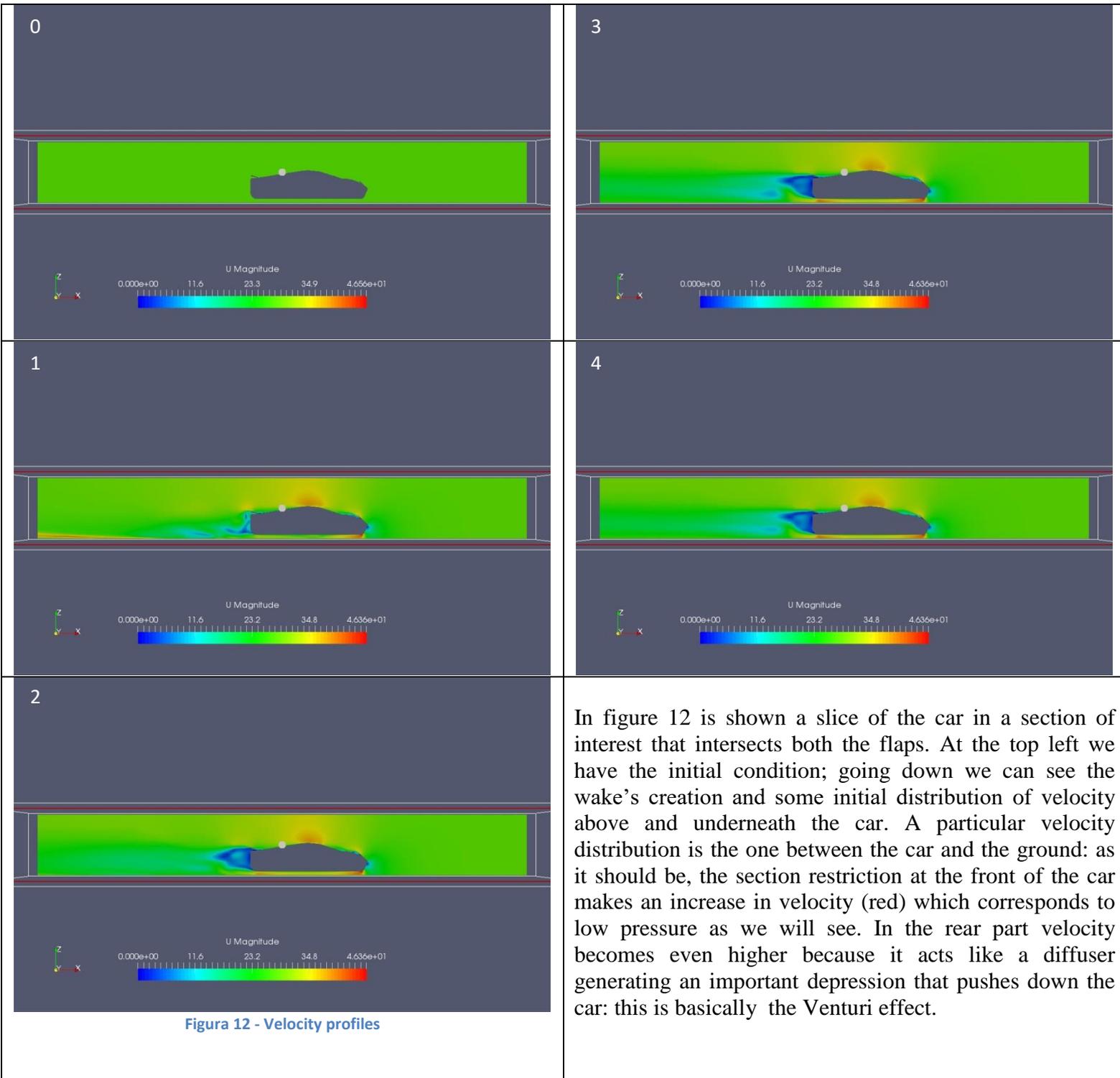
$$U_\tau = \sqrt{\frac{\tau_w}{\rho}}$$

is equal to :

where  $\tau_w$  is the wall shear stresses.  $y^+$  was calculated on the ground and the car: it is normally considered an acceptable value if less than 300. To realize it in sHM the refinement of these areas was increased, making heavy the mesh.

#### 4.1 First geometry: 40/15 degrees

This is the first situation studied, let's begin with the velocity profile. Since simpleFoam is not a transient solver, we can only look at qualitative images at different iterations to see the wake development.



In figure 12 is shown a slice of the car in a section of interest that intersects both the flaps. At the top left we have the initial condition; going down we can see the wake's creation and some initial distribution of velocity above and underneath the car. A particular velocity distribution is the one between the car and the ground: as it should be, the section restriction at the front of the car makes an increase in velocity (red) which corresponds to low pressure as we will see. In the rear part velocity becomes even higher because it acts like a diffuser generating an important depression that pushes down the car: this is basically the Venturi effect.

Now we take a look at the pressure field. The Figure 13 shows the pressure at time=0 i.e. initial condition. The other one is the final distribution around the car. At the front there is the stagnation point. Then we notice the light blue zones which are depression parts corresponding to the high velocity we mentioned before, besides there is a high pressure area nearby the rear flap and therefore, as a result, a downforce arose.

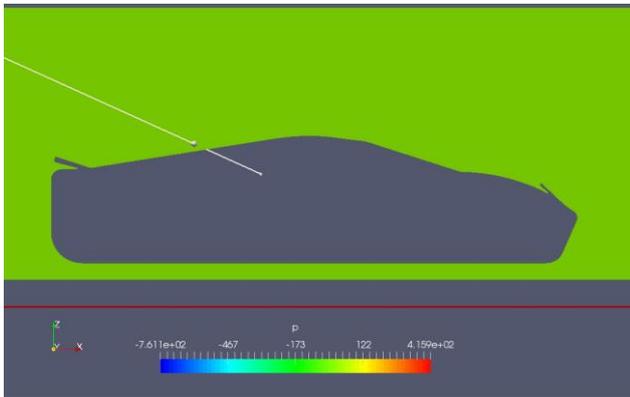


Figura 13 - Pressure field at time 0

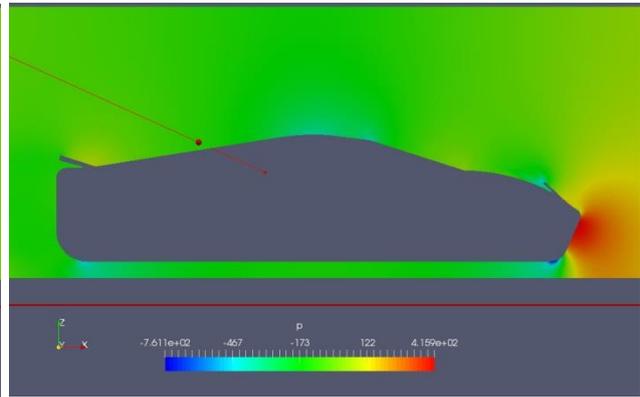


Figura 14 - Pressure field at final time

In Figure 15 we present a 3D view of the entire body pressure distribution at final time.

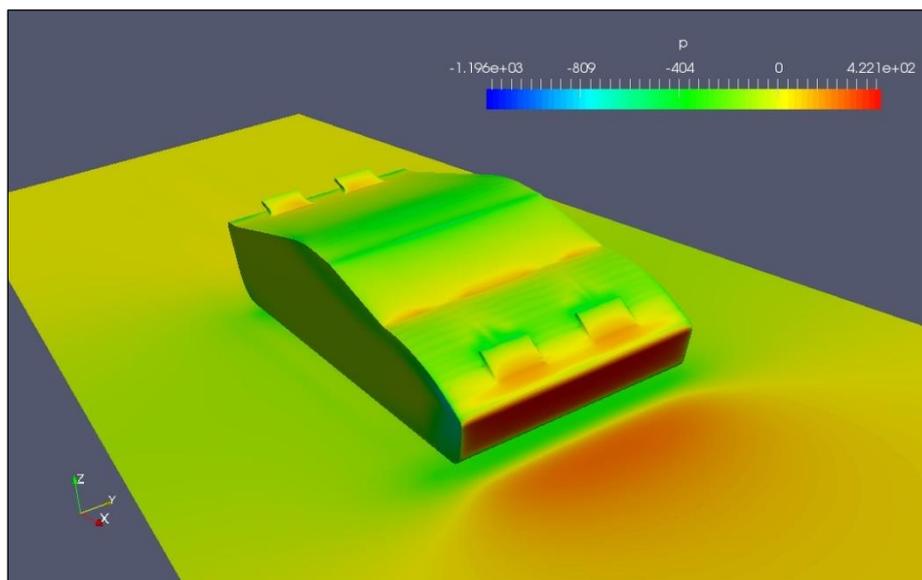


Figura 15 - Pressure field on car

And finally we take advantage of paraFoam to have a direct vision of the yPlus values so we can easily understand where the highest values are and then refine the mesh, if necessary. The optimized mesh has no point above the limit value of 300 as can be seen in Figure 16. Nevertheless the maximum values of yPlus ( $\approx 250$ ) are situated on the side of the car.

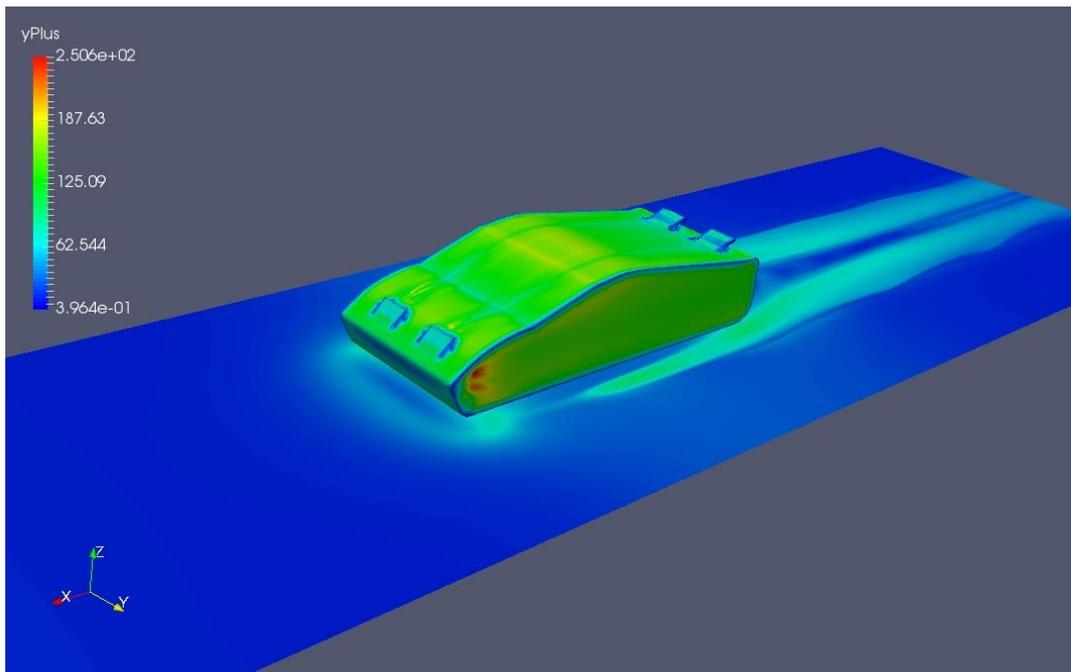


Figura 16 - YPlus distribution

For every single geometry post processing like this has been done. We now quote only a final-time velocity pictures for each geometry to see the changes with the angles and the value of  $C_L$  and  $C_D$ .

For the first geometry we obtained:

$C_L$	$C_D$
-0.67	0.59

#### 4.2 Second geometry: 40/30 degrees

In this second situation the rear flap is tilted more so the wake region is bigger and the pressure on the rear flap is increasing. We expect to find a lower  $C_L$  (more downforce) and a higher  $C_D$  due to the separated flow.

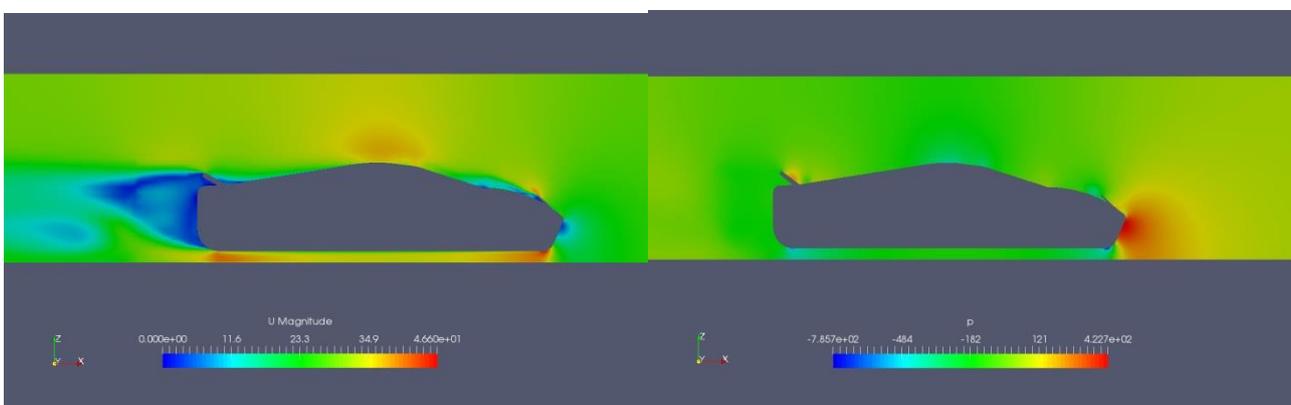


Figure 17 - Velocity and pressure slice profile

To confirm these suggestions we find:

$C_L$	$C_D$
-0.81	0.63

#### 4.3 Third geometry: 40/40 degrees

Going on with the inclination of the rear flap only the trend is highlighted even more and we get:

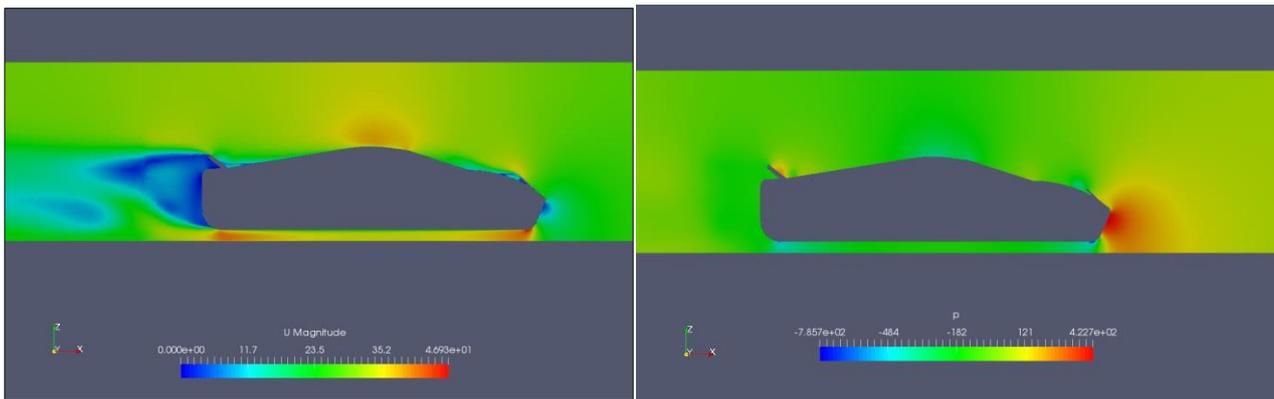


Figure 18 - Velocity and pressure slice profile

$C_L$	$C_D$
-0.93	0.67

#### 4.4 Fourth geometry: 50/15 degrees

Now we increase the front flap to 50° degrees and keeping it fixed we investigate the variation of parameters for the same three rear angles. This is the first one:

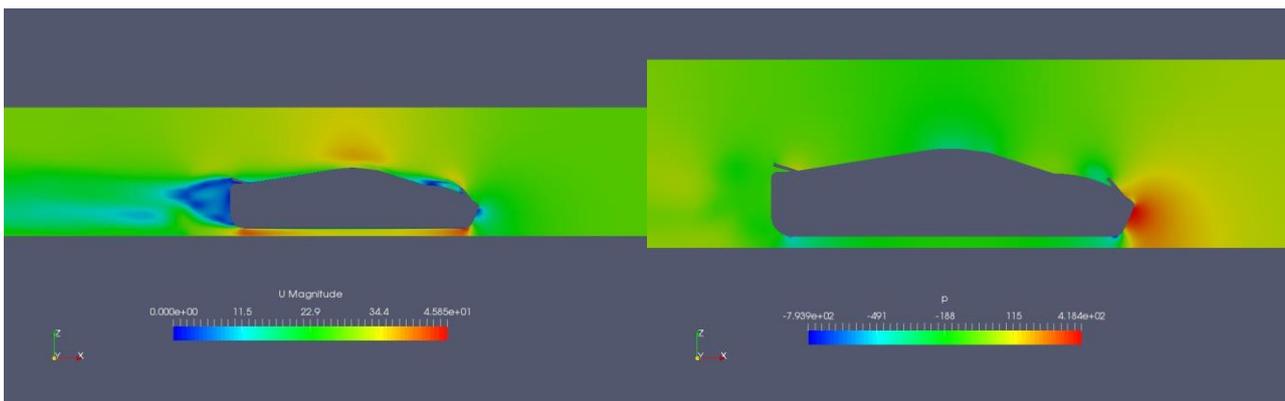


Figure 19 - Velocity and pressure slice profile

In the velocity field we can clearly see a low magnitude area above the front flap that there was not present before. As a result we obtain:

$C_L$	$C_D$
-0.81	0.61

#### 4.5 Fifth geometry: 50/30 degree

Increasing front flap angle we have a real rise in  $C_L$  which exceeds the value 1 in magnitude.

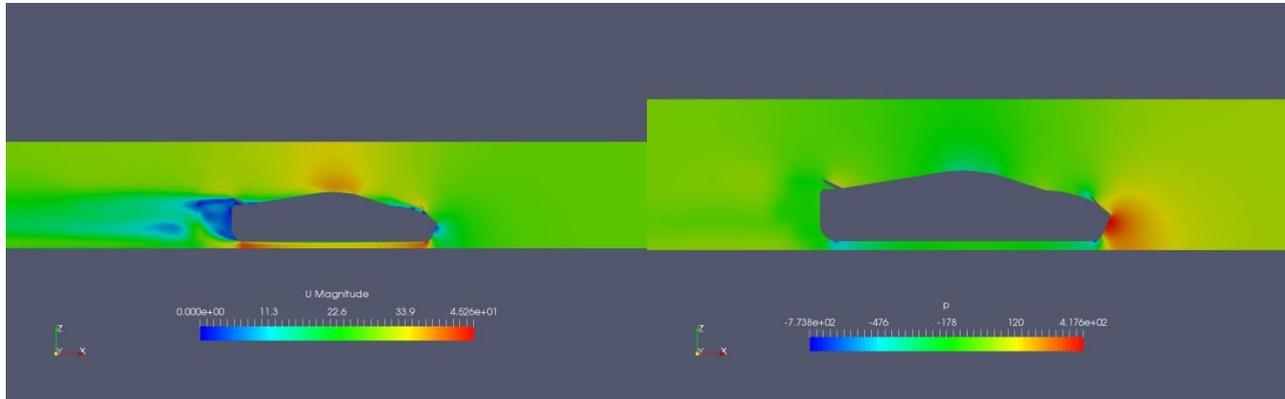


Figure 20 - Velocity and pressure slice profile

For this case we get:

$C_L$	$C_D$
-1.06	0.659

#### 4.6 Sixth geometry: 50/40 degrees

This is the last geometry with both flaps at maximum inclination, then we have:

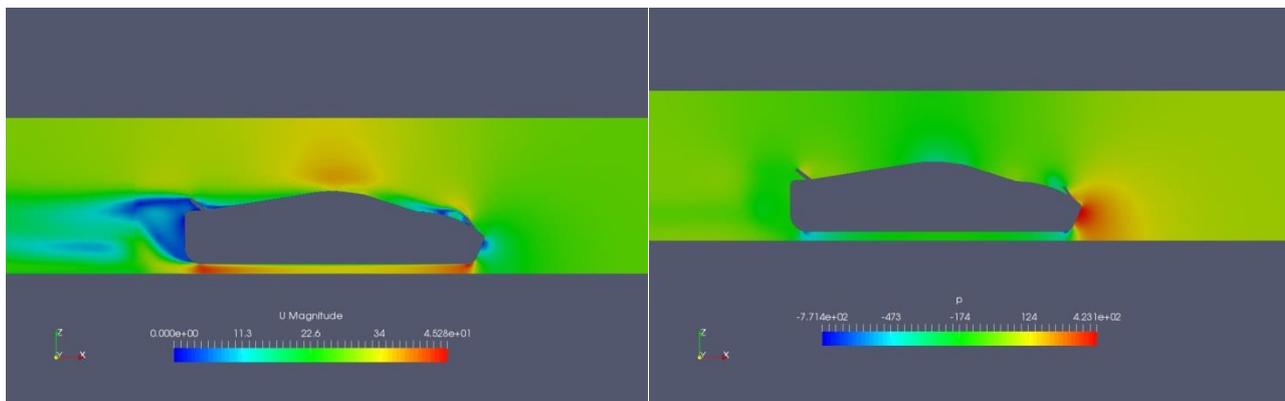


Figure 21 - Velocity and pressure slice profile

$C_L$	$C_D$
-1.1	0.68

#### 4.7 Final considerations

This work was finalised to find out the effect of several angles on a car aerodynamics. Clearly we would like to get the more downforce as possible with the lowest drag to save fuel or to go faster. To choose an ideal angle we should ask what it is required: if the car had a positive  $C_L$  at  $0^\circ$ , with danger of take off, we would look for a maximum value of  $C_L$  to respect the balance:

$$M \cdot g > \rho \cdot V^2 \cdot A \cdot C_L$$

So an additional simulation for  $0^\circ$ - $0^\circ$  degree angles has been done and it turned out that, at this speed, the value of  $C_L$  is already negative (-0.19). So there's no danger of positive lift force and thus every combination studied could be acceptable.

But let's assume that a particular value of  $C_L$  is required: we found out that is more convenient (less  $C_D$ ) to increase inclination of front flaps rather than rear flaps with an equal  $C_L$ . This will happen until the front flap is so inclined that  $C_D$  starts to grow and  $C_L$  to decrease.

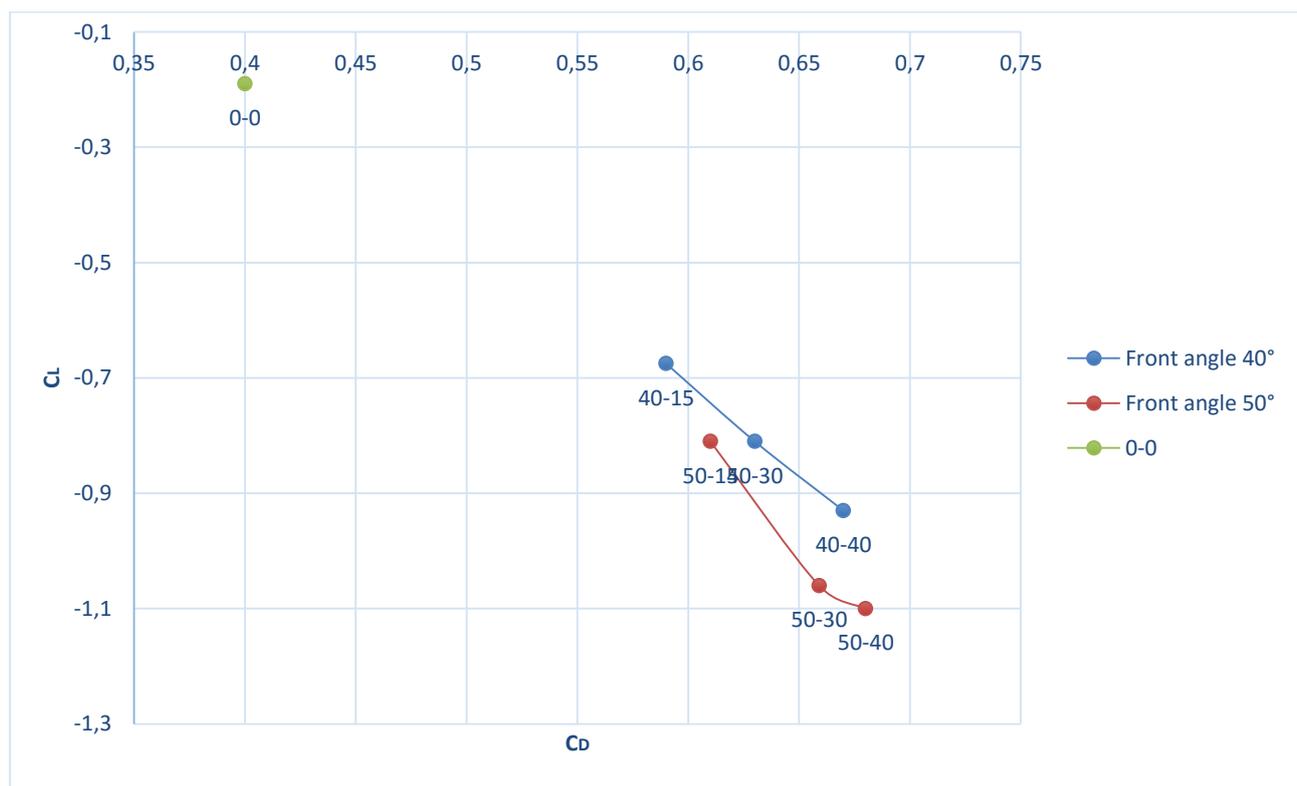


Figure 22 -  $C_L$  vs  $C_D$  plot

#### 4.8 Future development

This was a simplified analysis of the problem as an initial case study: a more complicated one could be carried out certainly with:

- a more detailed geometry and a more refined mesh,
- velocity of the car should be increased till 200 or 300 km/h,
- non-aligned inlet flow should be investigate to simulate cornering that is the main situation in which stability must be provided by the flaps.

#### REFERENCES

1. CFD Online <http://www.cfd-online.com/>
2. Course slide- Joel Guerrero DICAT Unige [www.dicat.unige.it/querrero/](http://www.dicat.unige.it/querrero/)
3. [www.openfoam.com](http://www.openfoam.com)