Formula-one 3D Vehicle Aerodynamic Modeling, Geometric Idealization and Mesh Refinement Strategies

MARK LIN and PERIKLIS PAPADOPOULOS
Aerospace Engineering Department
San Jose State University
1 Washington Square, San Jose, CA 95192
USA
Mark.Lin@SJSU.edu

Abstract: - Aerodynamic analysis techniques on a complex 3-D geometry performed using ANSYS Fluent CFD is presented in this paper. Specifically, a formula racecar's full-car model is analyzed where aerodynamic design plays a critical role in the vehicle's performance. CFD computational fluid dynamics is used because it would not be practical to use a theoretical approach due to the geometry's complexity. In order to analyze this vehicle model the solution approach is illustrated in three parts. First, a multi-element front wing assembly is analyzed by using a high-fidelity mesh modeling technique to capture local flow structure through a multitude of flaps and vanes. After the high-fidelity meshing technique has been demonstrated, next, a sequential idealization technique is used to study various combinations of airfoils to determine which configuration gives the highest aerodynamic efficiency. The result shows that an intermediate configuration offer the best lift-to-drag ratio, and that any additional airfoils does not improve the aerodynamic efficiency. Finally, the techniques for high-fidelity meshing and iterative solving are demonstrated on a full-car model to show how to solve aerodynamic problems of a real-world, complex 3D geometry. The paper concludes with a proposal for an oblique-wing geometry which takes advantage of non-symmetric turning-bias of modern race tracks. Comparison results with a conventional-wing car show that an oblique-wing car offer higher turning-downforce and lower drag than the traditional symmetrical version.

Key-Words: - Aerodynamics, Oblique Wing, Sideslip Angle, Rolling Tire, Ground Effect, Downforce

1 Introduction

Ever since postwar 1950, automobile racing has been in continuous running. In the early days the cars had a streamline, cigar-shaped fuselage. Then in 1973 race teams started adding wings to improve the cars' downforce. Now, all race cars are highly engineered with an aerodynamic exterior that maximizes the downforce to help them go around corners faster [1, 2]. As part of the engineering design, analysis is performed on the aerodynamic characteristic of the car. In the early days, the aerodynamic performance was assessed experimentally using wind tunnels; in recent years, with the widespread of personal computing, computational fluid dynamic software has gained popularity because it is cheaper and faster to perform than experimental assessment [3-5].

The race takes place on public roads in the form of an enclosed circuit that cars run a certain number of laps before taking the checkered flag. In modern times, some public roads (street circuits) have been replaced by purposely built race tracks that also have left and right-handed turns, fast and slow corners, and long straights for proving the cars' acceleration, cornering, and braking abilities. The current formula cars are the epitome of automotive technology. Everything we see in our passenger cars today comes from formula racing from years past, including powertrain, suspension, safety features, and aerodynamics [6]. In this paper we look only at the aerodynamic aspect of race cars by deploying computational aerodynamic analysis [7, 8].

To perform computational aerodynamic analysis, first the geometry has to be available in CAD. This computation geometry is then taken into a mesher software to discretize the flowfield and generate a mesh grid for solving the governing equations (idealized Navier-Stokes equations) numerically [9-12]. After the entire mesh is solved many times iteratively and the residual errors are less than a prescribed amount, the converged solution is then output to a post-processor and plotted to visualize the results.

In this paper, a commercial computational fluid dynamics (CFD) software is used to solve the flowfield around a race car outer body. The code used is ANSYS. The mesher routine used is Workbench

Mesher [13] (as opposed to Fluent Mesher). The solver routine used is ANSYS Fluent [14] (alternative to ANSYS CFX). The final data is viewed using CFD-Post, a postprocessor that can display pressure contour, velocity profile, and streamlines across the entire flowfield. In particular, the pressure contour displayed is gauge pressure and not absolute pressure, therefore a negative number means that pressure is below ambient. CFD-Post is further used to extract aggregate (integral) properties such as the forces acting on a body. This is used to determine the downforce and drag force affecting a race car.

2 Overall Problem

The thing one needs to do when doing CFD analysis is to prepare the computational geometry. The problem with many Computer Aided Design (CAD) models is that they may contain a mixture of surfaces and solid bodies. This is the most common problem among many different problems encountered during geometry preparation for fancy CAD models that are found on the internet. For this, the first thing one needs to do is to open up the file in a CAD program such as Solidworks to see if it contains surfaces. If the CAD model contains surfaces and not solid bodies then it cannot be meshed (without much difficulty). The method to do this is described in section 4.

The second problem one may encounter when trying to mesh a car model is that the model might contain many fine features, which if not meshed correctly may not be able to capture all the local flow structures. This has to do with the accuracy of the results. With the front wing that is used on race cars today, the complexity of it has grown over the years: It started out as a single airfoil in 1976; next, a second winglet was added. And today, the front wing is a very complex aerodynamic structure with upwards of 39 individual airfoils. A complicated wing like this is very difficult to analyze and one would need to use CFD. The intricacies of the design and the small air passageways that must also be modeled in order to capture the local flow structure for a high fidelity simulation [15]. The way to analyze this highly complicated 3D aerodynamic structure will be discussed in section 3.

The third problem is when trying to discretize the computational domain the mesher successfully generates a volume mesh but the mesh size is simply too large to solve! This is a real-world problem because in an ideal world, one can have unlimited memory to solve his problem. During initial meshing, especially for geometries that has many small features, the total mesh size can be too large to compute for the given hardware. It would then need subsequent refinements to get the mesh down to a manageable size. For the computers used in this study with 16GB of RAM on each machine, the upper limit of mesh size and hence the maximum problem size is about 4 million cells. For example, the first attempt at meshing the whole car geometry shown in Figure 1 resulted in 31 million cells! While it may be possible to solve a 31 million-cell problem, we would only be able to solve 1/10th of it. It typically takes a couple weeks to clean up the geometry using CAD operations to get it down to a 4 million-cell mesh that's suitable for computation. An illustration of this problem is shown in section 5.

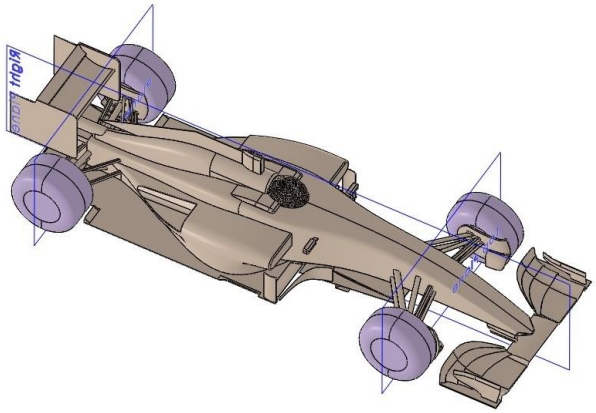

Fig 1. CAD geometry in Solidworks

3 High-Fidelity Meshing Strategy

In the following section, a high-fidelity mesh model is constructed for a complex front wing geometry. The goal of this exercise is to develop a methodology for meshing complex geometries with small gaps to show that local flow structure can be captured [16]. This is the first step in the analysis of a complete race car: one must demonstrate that a complex geometry can be meshed first before it can be taken into the solver. As an example, in this section a complex 3D front wing geometry with fine features will be used.

3.1 Computational Geometry

The Importance of Including the Front Wheels in the Wing Model

In order to perform a computational fluid dynamics study, a fluid domain must be constructed and discretized to form a mesh grid that's used in the computation. While this part of the work may seem trivial, it seldom is. To the contrary, without a mesh the domain cannot be discretized and CFD techniques cannot be used to solve the flowfield. To

construct a computational geometry, Computer Aided Design (CAD) software is typically used. This is the most common method for setting up the problem. While it is possible to construct a mesh from scratch in the mesher software, the method is limited to simple geometries such as 2D airfoils that are only suitable for pedagogical purposes [17-20]; on the other hand, real-world applications have complex 3D shapes. To construct a geometry, an assortment of CAD software can be used. Two are illustrated here: Solidworks by Dassault Systèmes [21] and DesignModeler by ANSYS Inc. [22]. These two software are used collaboratively to construct the geometry because as it will be shown later, each software has a complementary set of tools for geometry manipulation.

This paper describes the method that is developed by the authors to take a 3D CAD model of everyday object (e.g. a car) into the computational domain and make it available for meshing. In this section the geometry of interest is the front-wing of a race car, with many wing elements and small gaps between them. In addition, the front tires need to be "modeled-in" as well since they have direct effect on the wing design, because one of the main functions of the front wing is to direct air away from the tires so they have to be a part of the aerodynamic consideration [23].

3.2 Mesh Generation Strategy

Focusing Attention on Wing Elements with a Fine Mesh to Capture Local Flow Structure

Meshing is a sequential process and very often, much CAD manipulation has to be performed before the meshing software (ANSYS Mesher) can successfully construct a mesh. Very often, in the first few tries mesh generation would fail. This can happen in three ways: first, a surface mesh is generated but cannot be propagated to a volume mesh. Second, the mesher could fail with an error message saying, "a mesh cannot be generated using the current meshing options and settings." Third, the mesher could fail with an error message saying, "one or more entities failed to mesh. The mesh of the body containing these entities may not be up-to-date." In any case, the solution is to go back to the CAD model and simplify it some more or try to delete the problematic geometry and reconstruct it. ANSYS Mesher has a command that will display the problematic geometries when meshing fails. From our experience, mesh generation usually take many tries. For complex 3-D shapes, one must do several rounds of simplification (a.k.a. defeaturing) before a successful mesh can be generated. In the entire workflow, 80% of human work is spent on constructing the geometry and generating the mesh; the actual computation part takes just 20% of human effort.

As shown in Figure 2, it is also worth noting that the winglet area for this mesh uses a much finer grid size so as to more effectively capture surface complexity. This was accomplished via the Face_Sizing command in ANSYS Mesher with an element size of 0.5e-3m (i.e. a 5 mm element size). This resulted in 3.01 million elements for our model. This view illustrates the intricacies of the modern wing design (circa 2017), where multiple winglet surfaces are joined together and thin slots are placed in-between to allow air to flow through. In order to capture this millimeter-scale flow structure, a small element size was chosen to model the fluid flow through these slots.

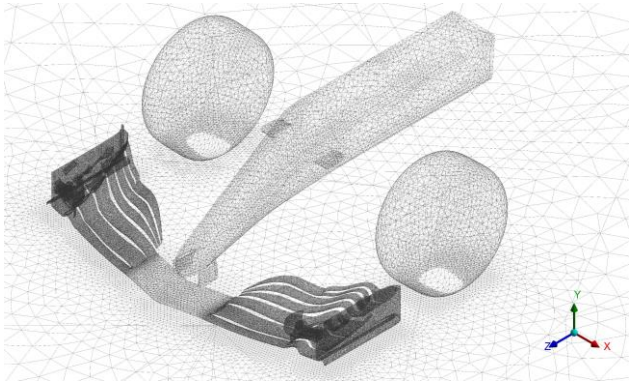

Fig 2. Mesh of first model

3.3 Modelling Approach

3.3.1 Introducing Moving Ground Plane and Pressure Far-field Boundary Condition

The reason for the pressure-far-field boundary condition is so that the problem will solve; when we first tried pressure-outlet boundary condition, the problem didn't solve. While the same types of boundary condition were used from 2.5° to 15°, the boundary condition used for the 0° case was different: for the 2.5° to 15° case (i.e. turning), one of the enclosure walls was specified as inlet (air_inlet) instead of as outlet (pressure far-field) like in the 0° case. Other boundary conditions used include inflation (for boundary layer) on the surface of the car, and a moving ground plane matched to the airflow speed and direction. Additionally, the numerical solver used is a density-based solver, and the turbulence model used is transition SST (Shear Stress Transport) model. Although it is possible to setup the problem as coupled laminar-turbulent using a more complex numerical model, the model used here is a fully-turbulent model in order to keep the problem size manageable. The CFL number is

adjusted for the different β (sideslip angle) cases to help with convergence. We ran each case to 15000 iterations using a computational domain of 55 m long by 30 m wide by 16 m tall.

3.3.2 Using Transition SST Turbulence Model

After the mesh was constructed the problem was run as a steady-state analysis, with an SST turbulence model. Freestream velocity was specified as 41.6 m/s, which is equivalent to 150 km/hr. The inlet boundary conditions were such that the incoming stream had a variable angle dictated by changing the velocity components (U and V) of the freestream. The bottom wall (i.e. ground plane) was specified as a moving wall with velocity and direction the same as the freestream. This is important because in reality, when a car is moving through air, air is stationary with respect to the ground, so there should be no boundary layer. Additionally, during some initial simulation runs we observed that sometimes the temperature would rise and the pressure would dip to levels that are impossible in the physical world. We attribute this to numerical errors stemming from vacuum forming behind the wheels, hence we put a limiter on pressure of 1e7 Pa and a limiter on temperature of 173K to 473K. Finally, we ran the problem to 15000 iterations with a CFL number of 0.1, which on our computer took 4 days per case using 7 cores of an Intel Core i7 processor. Equations 1 and 2 show the SST turbulence model as given by Menter [24]. The variable F_1 is a blending function that is equal to zero away from the surface (i.e. using the κ-ε model), and it is equal to one inside the boundary layer (i.e. using the κ-ω model).

$$\frac{\partial(\rho k)}{\partial t}+\frac{\partial(\rho U_i k)}{\partial x_i}=\tilde{P}_k-\beta^*\rho k\omega + \frac{\partial}{\partial x_i}\left[(\mu+\sigma_k\mu_t)\frac{\partial k}{\partial x_i}\right] \quad (1)$$

$$\frac{\partial(\rho\omega)}{\partial t}+\frac{\partial(\rho U_i\omega)}{\partial x_i}=\alpha\rho S^2-\beta\rho\omega^2+\frac{\partial}{\partial x_i} \left[(\mu+\sigma_\omega\mu_t)\frac{\partial\omega}{\partial x_i}\right]+2(1-F_1)\rho\sigma_{\omega2}\frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial\omega}{\partial x_i} \quad (2)$$

3.4 Results

After the runs were finished, results were viewed using ANSYS CFD-Post postprocessor. Force results were output by ANSYS and plotted in Figures 3 and 4, and the calculated coefficients are listed in Table 1.

Table 1 Calculated coefficients vs. sideslip angle (first model)

β	-L (N)	D (N)	C_l†	C_d‡
0°	660.8	326.0	1.06	0.51
2.5°	654.3	337.7	1.05	0.53
5°	646.6	353.2	1.04	0.55
7.5°	636.9	380.9	1.02	0.60
10°	635.9	411.0	1.02	0.64
12.5°	635.5	441.1	1.02	0.69
15°	617.9	438.7	0.99	0.69

† Reference area using top projection of the wing (0.59 m^2)
‡ Reference area using front projection of the wing (0.60 m^2)

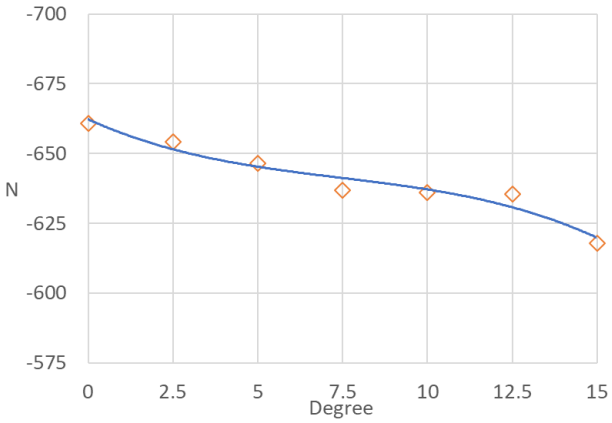

Fig 3. Downforce vs. sideslip angle

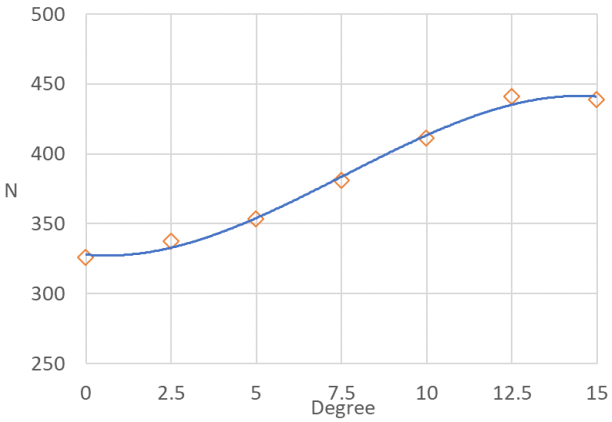

Fig 4. Drag force vs. sideslip angle

Two things are important to mention here: 1) the comparison of C_d and C_l as β is varied from 0° to 15°, and 2) local aerodynamic effects of the wing, for example the flow in-between winglets.

3.4.1 Aerodynamic Forces and Coefficients vs. Sideslip Angle

The results shown here for the front wing are similar to those of a complete car, which will be shown later in section 5. Figure 3 shows that downforce decreases as the car turns; this is due to an increase in the airfoil's swept angle. Figure 4 shows that drag force increases as the car turns; this in turn, is due to an increase in the frontal projection area. Although the increase in drag force is unavoidable, the decrease in downforce is undesirable and should be avoided. Downforce is the normal force N exerted by the car on the ground. Multiply N by the friction coefficient μ of the rubber-to-asphalt interface, and the result is the tangential frictional force that makes the car turn. When turning, we want to maximize the downforce so that tires press as hard as possible into the ground, thus allowing the car to take corners at a higher speed. Ideally, the front wing should have maximum downforce when turning and minimal downforce (in the same way as drag) when going straight. What is shown in Figure 3 is opposite of the desired downforce behavior.

For calculating C_l and C_d, an area of 0.59 m2 was used as the wing surface and an area of 0.60 m2 was used as the frontal projection. Looking at the calculated results in Table 1, the C_l and C_d values show a trend: the C_l (i.e. negative lift) starts out high for 0° sideslip angle but then it decreases as β is increased to 15°. In contrast, the C_d (i.e. drag) is monotonically increasing as the angle of sideslip rises – this is because as the car turns, more frontal area is exposed and more pressure force acts against it. Because this is a significant finding, the same angle analysis was performed on another front wing model shown in Figure 5. The results are included in

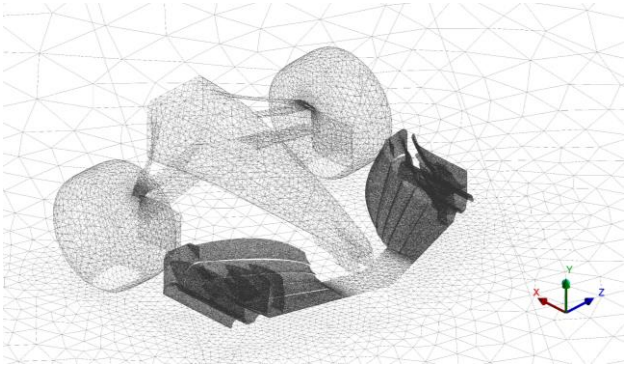

Fig 5. Mesh of second model

Table 2 and Figures 6 and 7. The consistency of behavior corroborates that modern F1 front wings create higher straight-line downforce but lower turning downforce.

Table 2 Calculated coefficients vs. sideslip angle (second model)

β	-L (N)	D (N)	C_l†	C_d‡
0°	1015.2	565.8	1.32	0.68
2.5°	989.5	570.4	1.28	0.69
5°	945.0	584.1	1.23	0.71
7.5°	899.8	596.2	1.17	0.72
10°	865.8	613.0	1.12	0.74
12.5°	820.9	641.8	1.06	0.78
15°	708.4	637.7	0.92	0.77

† Reference area using top projection of the wing (0.73 m^2)
‡ Reference area using front projection of the wing (0.78 m^2)

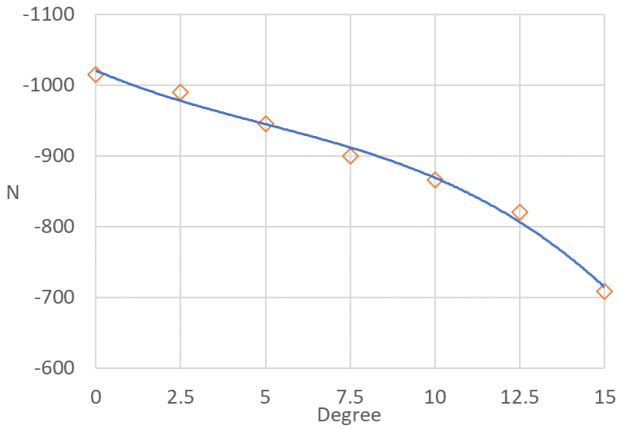

Fig 6. Downforce vs. sideslip angle

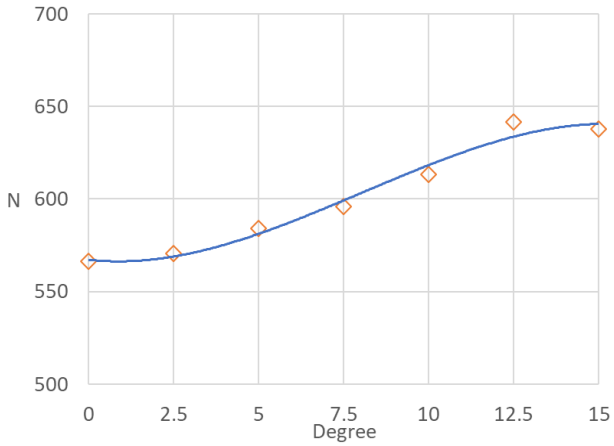

Fig 7. Drag force vs. sideslip angle

With knowledge of the C_l and C_d values, these data can be correlated with experimental data from the wind tunnel. Because of the cost of operating a wind tunnel, typically a scale model of the car is used. In order to correlate CFD results to experiments using a scaled model flow similarity needs to be maintained. In this case because the freestream velocity (41.6 m/s) is below 0.3 Ma, we only need

Reynold's number similarity and not Mach number similarity. A report that contains experimental data from a 3/8-scale car model is shown in [25, 26]. Current-day F1 cars are typically tested at 60% full-scale. By matching the Reynold's number (i.e. by operating the wind tunnel at 1/60%*U = 1.67U), in our case it is 5.1×10^6 (with the width of the front wing at 1.8 m taken as the characteristic length), the lift and drag coefficients would be the same for both the full-scale CFD simulations and the 60% wind tunnel model.

3.4.2 Forward Winglets Local Aerodynamic Effects

When we zoom in to look at the flow structure in more detail, the qualitative simulation result is quite interesting: when we take a cross section through the front wing, we can see the millimeter-scale-flow through the winglet gaps (Figure 8). The winglet gaps bleed high-pressure air from the top-side of the previous winglet onto the bottom-side of the following winglet, allowing faster airflow underneath the winglet to create downforce. Nevertheless, this effect diminishes as the chord length of each successive winglet gets shorter to avoid separation – one may ask whether the benefit gained from additional downforce would out-weight the additional frictional drag from the increased area. Figure 9 shows the right-half of the wing along with a front tire, where in (a) it shows the top view and in (b) it shows the bottom view. In Figure 9 (a) the front 3 winglets make sense, but the next 4 winglets seem to contribute little to the downforce generated (i.e. the first 3 winglets have higher pressure on top but not the last 4 winglets). This information is corroborated in Figure 9 (b) when viewed from the bottom where only the first 3 winglets show a lower pressure (i.e. blue color). Although we would expect the whole front wing to be blue in color (i.e. lower than ambient pressure), we see that only the first 3 winglets are contributing to that; the rear 4 winglets do little to create downforce, while generating significant and undesirable drag. This design should be updated so that all of the area underneath the front wing shows blue.

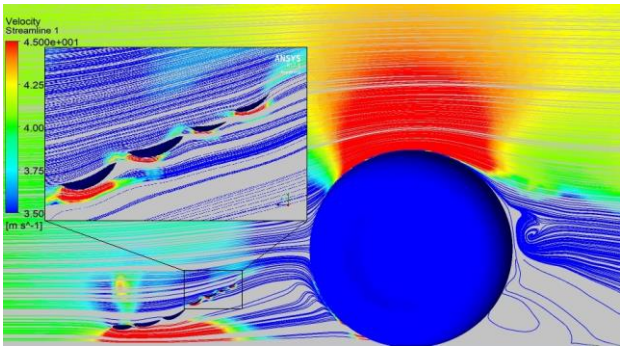

Fig 8. Front wing section view of streamlines colored by velocity

As mentioned earlier, this plot also shows the treatment for the tires' contact with the ground. In our case, the tires are positioned slightly below the ground plane so as to protrude slightly outside the bottom wall of the enclosure. When it is meshed, an empty space remains. This tire treatment has shown to simulate the tires' contact patch quite well.

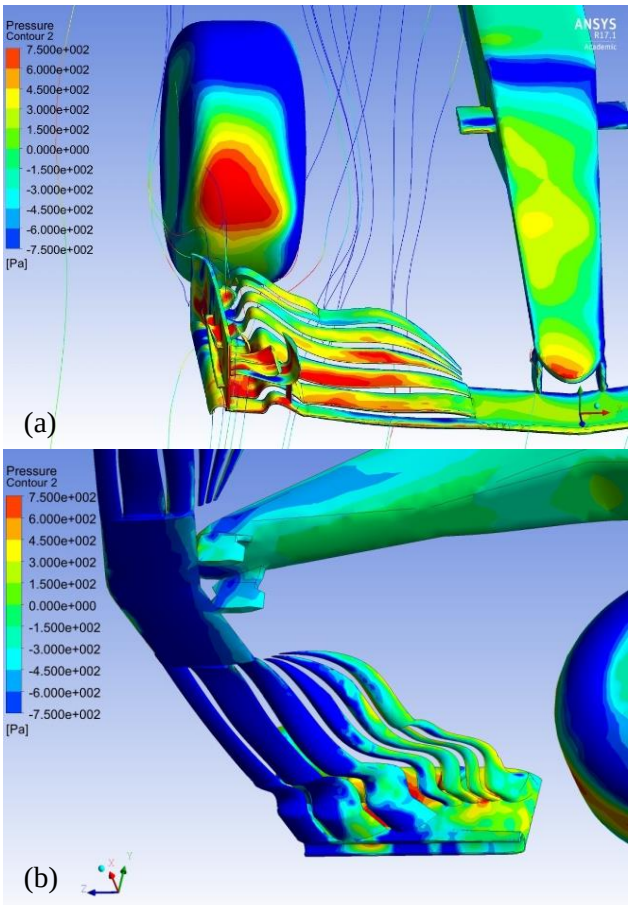

Fig 9. (a) Upper surface pressure distribution and (b) lower surface pressure distribution

Although Figure 9 (a) shows what appear to be an effective channeling of airflow around the front tire (i.e. only a small patch of high pressure area), when we look at the streamlines in Figure 8, it shows that the flow is not deflected sufficiently upward to go around the topside of the tire. Furthermore, the backside of the tire clearly shows separation at about 135°, followed by a region of recirculation. Lastly, in the lower portion of Figure 8, behind the tire there is extreme low pressure with very sparse streamlines: this is a problematic area for aerodynamics, and it may be useful to install a fairing behind the tire in

order to avoid separation and recirculating streamlines. This would go a long way to help reduce the overall pressure drag.

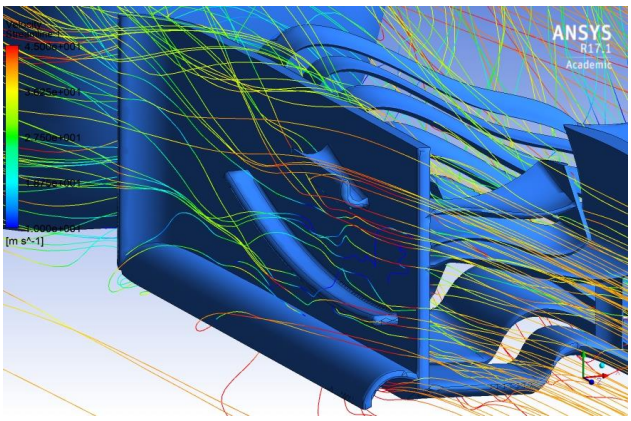

Fig 10. Outboard winglet flow structure

Figure 10 shows the streamlines as they pass over the two canards mounted on the outside of the endplate in front of the right front tire. This figure is from the 0° β simulation so the freestream is coming in parallel to the endplate. Looking at the curved shape of the two canards, the intent appears to be to divert air upwards over the top of the tire; however, as the streamlines clearly show, air spills over the side of the canards, instead of following their curvature. This generates unnecessary vortices and reduces downforce due to vortex-induced lift and also creates drag.

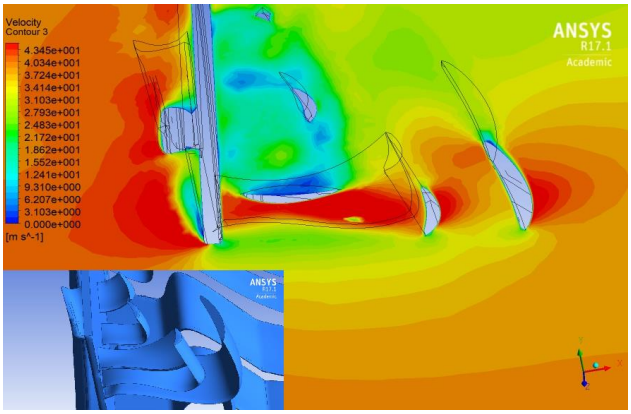

Fig 11. Inboard winglet velocity profile horizontal section-cut

Figure 11 shows yet another portion of the front wing: the front inboard section of the endplate. On the left side, the model shows five endplate-mounted winglets; on the right side, a contour plot shows fluid velocities surrounding this complex geometry. To plot this velocity contour, a horizontal plane was inserted that cuts across the three vertical fins. Remarkably, while the two front fins show a conventional velocity profile over the airfoil shape, the third fin does not – rather, it shows a much slower fluid velocity. Therefore, it seems that this winglet serves no aerodynamic purpose. Through this example, we showed that we can model a complex geometric shape and produce realistic-looking velocity profile, streamlines, and pressure contours for a complex 3D wing. This gives us the confidence to start modifying shapes and comparing results.

4 Design Considerations

In this section, the technique developed previously for meshing a complex geometry is applied to multiple aerodynamic configurations to mesh them individually and then to solve them. The individual results are then compared to determine which aerodynamic configuration is the most efficient [27]. The technique developed here to defeature the geometry illustrates the importance of Computer Aided Design (CAD) tools in geometry preparation for CFD. The goal of this exercise is to show that by sequentially idealizing front wing geometry one can discover the best combination of airfoils that give the most efficient aerodynamic performance.

4.1 Computational Geometry

4.1.1 Sequential Idealization of Seven Geometries

The technique presented in the last sections is used here to create seven variations of the computational geometries for analysis. The seven computational geometries are generated by taking the complete wing geometry as shown in Figure 12, then airfoils are sequentially removed using DesignModeler, and lastly the model is remeshed and recomputed. The seven computational geometry configurations are numbered 1 through 7. Configuration 1 is the base geometry and is the most complete wing. Configurations 2 to 7 are simplified versions of the wing, with configuration 7 being the minimum version.

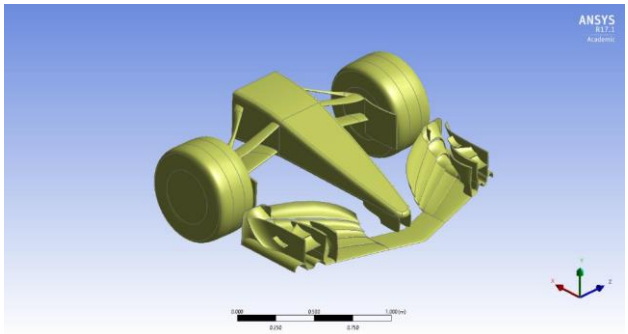

Fig 12. Computational geometry used for design variations as seen in DesignModeler

4.1.2 The Power of Sectioning (the CAD Model) – to Make Sure Solids Stay as Solids

Figure 13 illustrates the various geometry configurations using the same model. Each time, a wing element is removed from the assembly using CAD operations which can be done in either Solidworks or DesignModeler. DesignModeler was chosen because it is an ANSYS native program and is easy to remove unwanted geometry. For example, to generate configuration 2 all that needs to be done is to highlight the faces of the inside vertical winglet, and use the Remove_Face command to eliminate it. When it's removed this way, DesignModeler automatically generates a patch on the remaining wing element to repair the hole that's left when this inside vertical winglet is removed. However, it is still a good idea to check that the new geometry is still a solid body by looking at the cross-section of the geometry.

When performing CAD manipulations in Solidworks and DesignModeler, one should frequently "section" the model to ensure the model

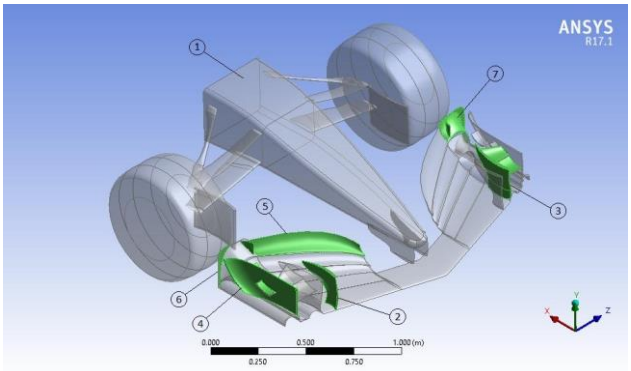

Fig 13. Various configurations used in the analysis model

is still a solid body. This is a standard operation that most CAD software can perform. As mentioned earlier, a surface body with hollow internal space cannot be used for meshing. Therefore, when one starts to work with a CAD model, the first thing to do is to look at the features tree in Solidworks to see if there are any surfaces. Sometimes, when one is manipulating a CAD model and a portion of the model is cut away (to exclude it from analysis), the software would turn a solid body into a surface body and render it unusable. Hence, when working in Solidworks or DesignModeler, it's a good practice to frequently "section" the model to make sure it is still a solid body. Examples of sectioned geometry are shown in Figures 14 and 15.

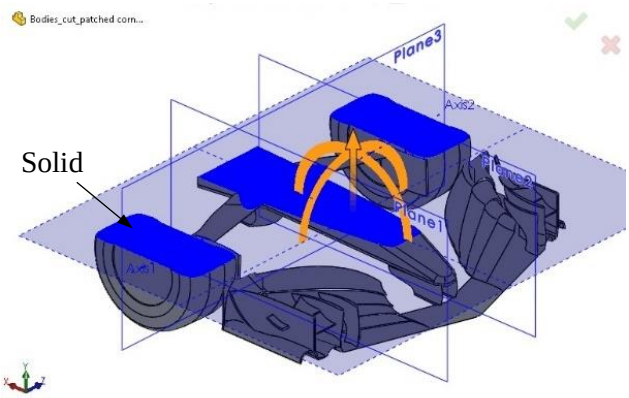

Fig 14. Sectioning approach in Solidworks

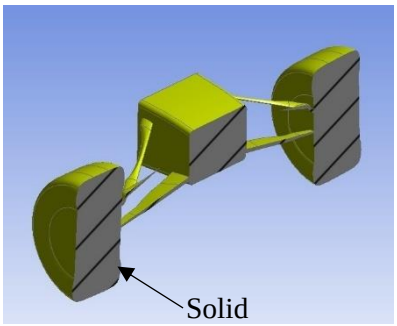

Fig 15. Sectioning approach in DesignModeler

After CAD operations have been performed to clean up the model and reduce its complexity, it is ready to be used for volume extraction from the fluid domain. This part of geometry preparation is different than a Finite Element Analysis (FEA) because in FEA the positive shape represented by the CAD model is what's meshed and analyzed, while in CFD the computational domain is the fluid volume surrounding the object. Hence, a volume enclosure needs to be defined and the object shape needs to be subtracted from the fluid volume. This is a very different concept because in the CFD computational domain, the object of interest (e.g. cars, airplanes, spacecraft) is represented by empty space. An image of the enclosure is shown in Figure 16 (a). Figure 16 (b) zooms-in on the object of interest which is represented by empty space.

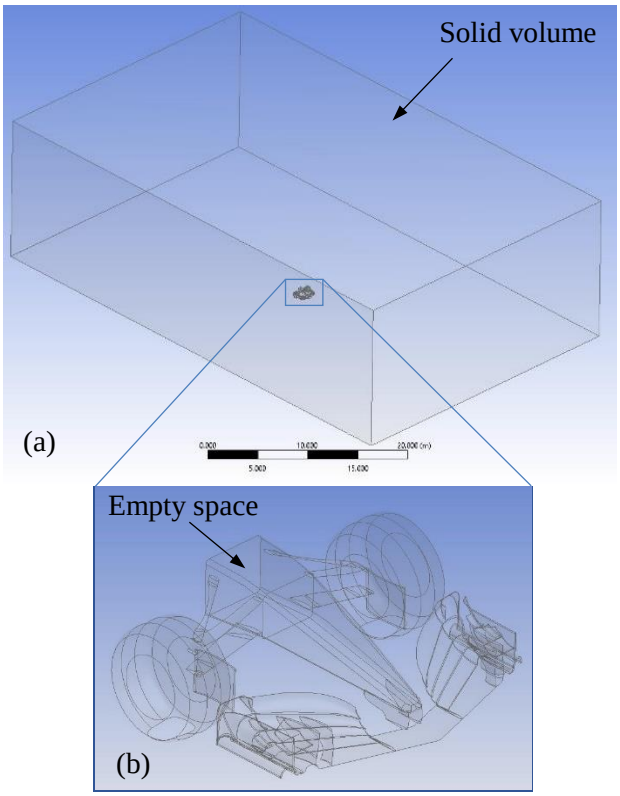

Fig 16. (a) The computational domain and (b) surface geometry wall boundary condition

4.2 Mesh Generation Strategy

4.2.1. Seven Different Geometries – Seven Different Meshes

In this section, the seven geometries are meshed separately. Doing this is computationally expensive but it's necessary, and each mesh needs to be solved individually. Once the geometry is simplified and is ready for meshing, ANSYS Mesher is used to generate a computational grid. Other commercially available meshing software can also be used [28].

Earlier in section 3 Figure 5, it showed the mesh generated for the 3D front wing geometry used here. It comprises of 3.9 million grid points. Looking at this plot, one can see that mesh refinement has been done on the individual winglets because of the high density of grid elements on them. This allows the creation of small gaps for air to pass through, and it allows the intricacies of local flow structure to be captured.

4.3 Modelling Approach

For all seven cases the same solver parameters were used to allow the outputs to be directly comparable. For this design exercise, solver parameters used in the previous section are again used here. These are: density-based solver, implicit formulation, transition SST turbulence model, steady state run, 15000 iterations, and a starting Courant number of 0.5. In addition to the parameter used in section 3, here an additional parameter is introduced to ignore the convergence criteria so the runs would finish 15000 iterations without stopping prematurely – this allows the results to be directly comparable for all seven cases. The only parameter that was varied for each run was the Courant number (i.e. the CFL number) in order to improve the stability when it's solving. For some cases the residuals came down quickly but for others the problem had to be solved slower to allow it to converge. Decreasing the CFL number helps with problem convergence.

4.4 Results

The following seven paragraphs talk about configurations 1 to 7 separately – each one showing a CFD result for one configuration. At the end of this section, a summary plot of aerodynamic efficiency is shown that's calculated from the resultant lift force and drag force predicted by CFD.

4.4.1 Results for Seven Design Configurations

Configuration 1 is the base geometry – this is the most complete aerodynamic configuration. It is a complete Formula 1 front wing from the 2017 season. As can be seen in the CAD image, this is a complicated wing geometry. Using CAD to process this geometry, the method of preparing it using Solidworks and then DesignModeler had been discussed in the previous section. When it is meshed, the minimum mesh size is specified as 5e-3 m. This resulted in a mesh of 4.1 million elements. The downforce (negative lift) extracted from ANSYS CFD-Post is 1015.2 N and the drag force is 565.8 N. There are no (red) high-pressure patches on the front side of the tires, indicating that the front wing is effective at diverting airflow around the tires. A vortex is formed on the inside of the wing assembly adjacent to the cockpit that stabilizes the airflow.

Configuration 2 is when the four inner vertical winglets (two in front and two in the back) are removed from configuration 1. This is done in DesignModeler, and the process is fairly straightforward using Face_Delete command. The model is then remeshed with a minimum cell size of 5e-3 m. This resulted in a mesh with 3.9 million cells. The resultant downforce extracted from CFD-Post is 984.5 N and the drag force is 545.8 N. This is direct output from ANSYS using a fine mesh size and high cycle calculation (15,000 iterations). The residuals have fallen by at least 2 orders of magnitude

comparing to when it started so the solution has converged.

Configuration 3 is a simplified version of configuration 2. In this configuration, the inboard structure that comprises of two horizontal flaps and one vertical fin is removed. Once again, this was done in DesignModeler so we didn't have to go all the way back to Solidworks. After meshing with a minimum cell size of 5e-3 m, the resulting mesh has 3.5 million cells. The predicted downforce is 1014.5 N while the drag force is 568.6 N. Once again, comparing Figure 19 to Figure 18, after the inboard structure is removed the pressure on the winglets become a lot more uniform. Looking at the numerical results they show that both the downforce and the drag force goes up. When the numbers are divided to calculate the airfoil efficiency there doesn't seem to be any aerodynamic benefits this inboard structure offers. The streamlines in Figure 19 show that airflow doesn't change very much when this inboard structure is removed.

Configuration 4 is derived from configuration 3 by removing the outer endplates on the front wing. These two large pieces on the front wing are the vertical fins that mark the end of the wing, and they are easily removed in DesignModeler using the Face_Delete command. On each endplate there are also two small canards protruding from each side – one on the inboard and one on the outboard. After these elements are removed, the wing is meshed in ANSYS Mesher with a cell size of 6e-3 m, and this resulted in a mesh size of 2.2 million elements. After calculation, the downforce predicted is 1033.7 N and the drag is 567.4 N. These numbers are very similar to those obtained for configuration 3, and according to Figure 20 there is no visible effect to the airflow when these large endplates are removed. This indicates that the endplates are not necessary and the wing would still perform equally well.

Configuration 5 is reduced from configuration 4 by removing part of the top winglet so only the outside circular tunnels remain. The portion that is removed is the horizontal slab, therefore a lower downforce is anticipated. This configuration is meshed with a minimum mesh size of 4e-3 m, and it results in an overall mesh of 3.9 million elements. After computation, the predicted downforce (i.e. negative lift) is 678.7 N and the predicted drag force is 449.2 N. Here, the downforce is noticeably reduced when the horizontal slab of the winglet is removed. This is an indication that it is an important aerodynamic surface to provide downforce so it needs to be there for the wing to work properly.

Configuration 6 is a further reduction of configuration 5 by removing the small Gurney flap [29] on the outer edge of the winglet. The Gurney flap is a small vertical strip about 1-inch tall that's mounted on the trailing edge of the airfoil and perpendicular to the streamlines. The same minimum mesh size of 4e-3 m is used, and this resulted in a mesh of 3.5 million elements. The predicted downforce is 599.3 N and the drag force is 445.8 N. Comparing these results to configuration 5, it shows that when the Gurney flap is removed, even though downforce is reduced by about 80 N, the drag force changes only by 4 N. This is an indication that the Gurney flap is an important aerodynamic device that helps generate downforce. This is done by increasing pressure on the pressure side (upper surface), decreasing pressure on the suction side (lower surface), and helping the boundary layer flow stay attached all the way to the trailing edge on the suction side (lower surface) of the airfoil.

Configuration 7 is the simplest of all geometries. It is obtained by removing the two circular tunnels in front of the tires so that the front tires are now exposed to the freestream. It is meshed with a cell size of 4e-3 m, and the resulting mesh has 3.6 million elements. The predicted downforce is only 453.8 N and the drag force is 439.6 N. This is the lowest downforce among all seven configurations. It is not an efficient airfoil because when the downforce is lowered, the drag force did not go down by the same amount. This is partly due to the fact that there is no longer a surface to divert air around the tire, hence a high-pressure zone forms in-front of the tires. This creates a blunt body as opposed to a long slender shape. Therefore the circular tunnels are important in diverting air around the tires and reducing form drag. This also reconfirms the importance to model the front wheels when studying the front wing.

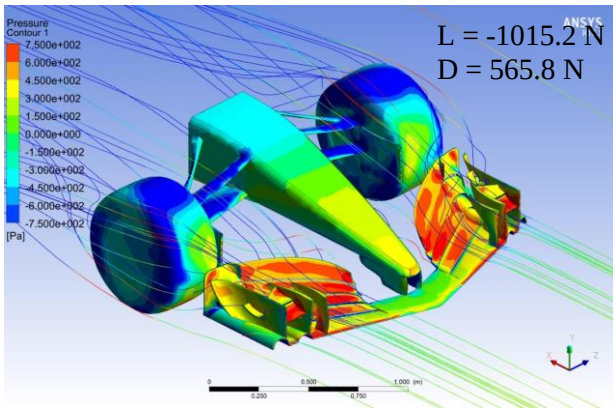

Fig 17. Configuration 1 frontend pressure distribution

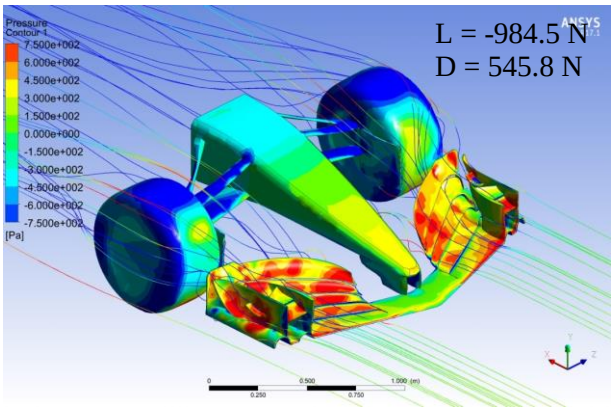

Fig 18. Configuration 2 frontend pressure distribution

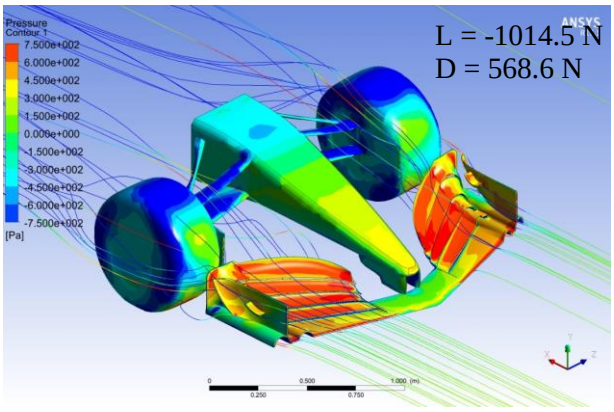

Fig 19. Configuration 3 frontend pressure distribution

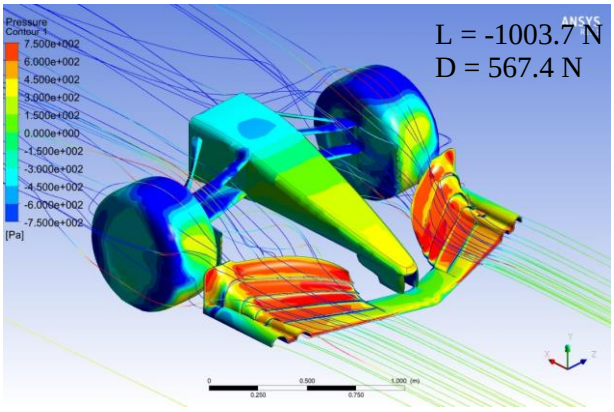

Fig 20. Configuration 4 frontend pressure distribution

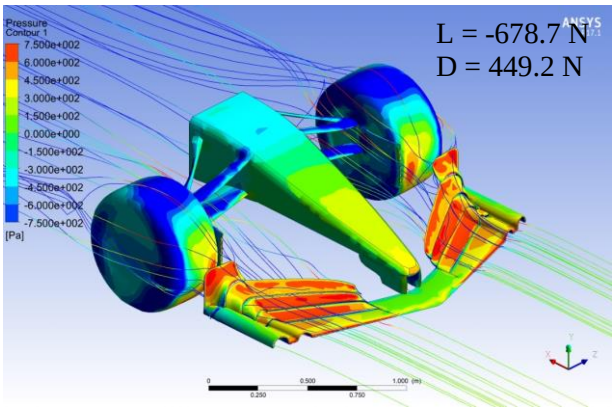

Fig 21. Configuration 5 frontend pressure distribution

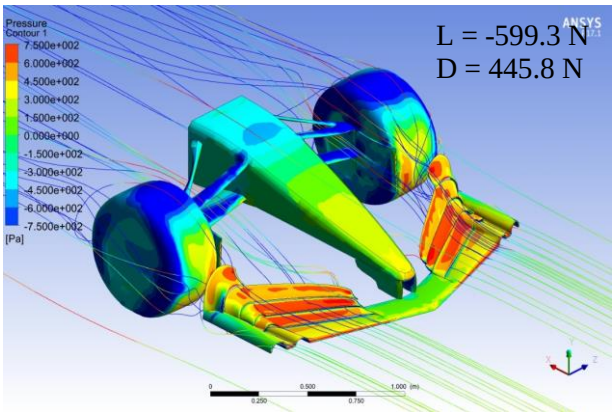

Fig 22. Configuration 6 frontend pressure distribution

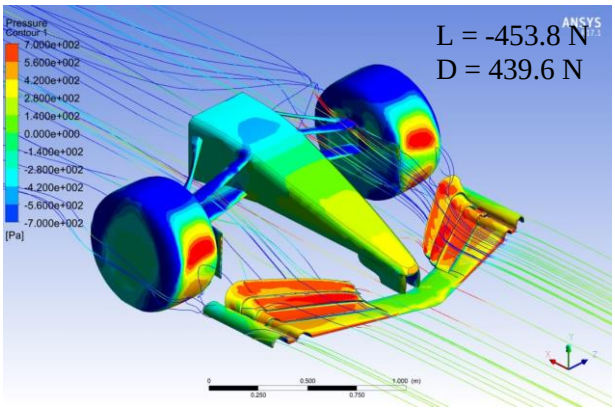

Fig 23. Configuration 7 frontend pressure distribution

After computing all seven geometries using ANSYS Fluent, the aerodynamic efficiency (-L/D) [30] for the different configurations are calculated and tabulated in Table 3. The results are plotted in Figure 22. A trend is clearly visible: as wing elements are removed and the wing is simplified, airfoil efficiency decreases accordingly – this is not surprising. However, the way that efficiency stays nearly constant for configuration 7 down to configuration 4 and then decreases for configuration 3, 2, and 1 is noteworthy. It's saying that anything beyond configuration 4 does not improve the aerodynamic efficiency of the front wing because the ratio of lift to drag stays constant. Therefore, the front wing design should be based on configuration 4: any further addition of aerodynamic surfaces would simply create drag due to skin friction, and any subtractions of aerodynamic surface would reduce downforce. From the table, the front wing aerodynamic efficiency η (-L/D) is 1.8 at best.

Table 3 Resultant forces and calculated aerodynamic efficiency of the seven wing configurations

Configuration	-L (N)	D (N)	η (-L/D)
1	1015.2	565.8	1.79
2	984.5	545.8	1.80
3	1014.5	568.6	1.78
4	1003.7	567.4	1.77
5	678.7	449.2	1.51
6	599.3	445.8	1.34
7	453.8	439.6	1.03

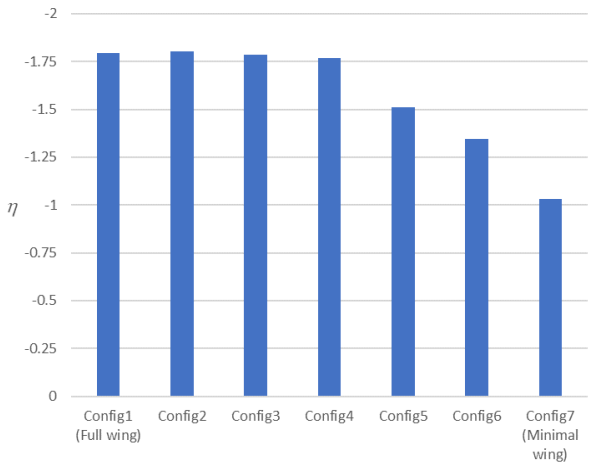

Fig 24. Lift-to-drag ratio of configurations 1 to 7

5 Application – Oblique Geometry

In this section, the analysis methodology developed in the preceding two sections is applied to a whole-car model. The geometry is more complicated, the mesh is bigger, and the solve-time is longer. Once we have demonstrated that we can work with a whole-car model then the design variation strategy employed in the last section will be used to study another configuration of the whole-car model to show how it can be improved [31]. The idea of an oblique wing geometry will be analyzed to illustrate the benefits over the conventional design.

5.1 Complete Vehicle Modeling

First, to demonstrate that a whole-car model can be solved successfully, the same solver parameters were used: density based solver, transition SST turbulence model, same boundary conditions, to 15000 iterations, and with a CFL number of 0.5. For the mesh, much geometry simplification had to be performed to get the grid down to 4 million cells, and in this particular model the geometry cleanup process further involved deleting surface bodies and replacing them with solid bodies. After the mesh was successfully generated, the program is ran and the results are described in the following paragraphs.

In addition to the numerical results listed in Tables 5 and 6, other results are best presented in graphical format to show the qualitative behavior of the flow around the vehicle. Velocity-colored streamlines, as well as pressure contours on both the ground and the vehicle surface, are used to illustrate the downforce and the drag force generated.

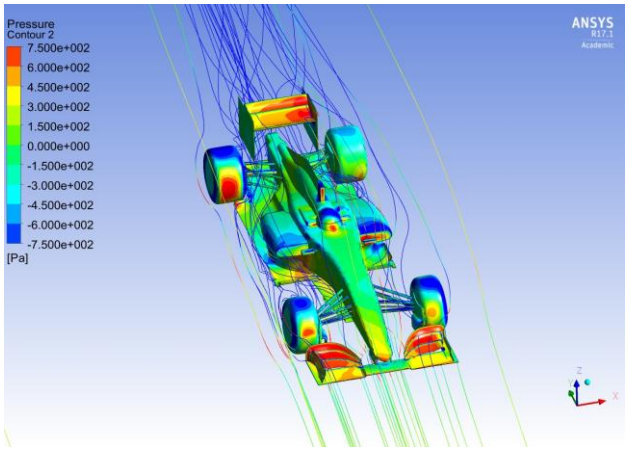

Fig 25. Representative pressure distribution for a 5° sideslip angle case

In Figure 25, a quarter view of the car is shown. It shows the streamlines, the pressure distribution on the surface of the car as well as pressure distribution

on the ground. In this example the freestream is coming in at 5° to the axis of the car, hence the back wing has a skewed pressure profile. In addition, the right-side tires are seeing higher pressure, especially the right rear tire, therefore creating the most form drag (i.e. pressure drag).

$$\text{Profile drag} = \text{Friction drag} + \text{Form drag} \quad (3)$$

Since friction drag and form drag are defined as

$$\text{Friction drag} \doteq \oint_A \tau_w dA = \oint_A C_f \left(\frac{1}{2}\rho_\infty U_\infty^2\right) dA \quad (4)$$

$$\text{Form drag} \doteq \oint_A P dA = P_\infty \cdot A + \oint_A C_p \left(\frac{1}{2}\rho_\infty U_\infty^2\right) dA \quad (5)$$

Therefore, combining (3), (4), and (5) profile drag can be written in terms of the coefficients as

$$\text{Profile drag} = P_\infty \cdot A + \frac{1}{2}\rho_\infty U_\infty^2 \oint_A (C_p + C_f) dA \quad (6)$$

Examining the ground pressure contour, it shows that there's negative pressure underneath the front wing while there's positive pressure on top of the wing, therefore creating downforce. It's also interesting to note that while there's a patch of red on the ground in front of each tire, there's a patch of blue on either sides of each tire, showing that's where air accelerates around the side of the tire and creates a low-pressure region.

Examining the streamlines, there appears to be a lot of turbulence on the right side of the vehicle as they zigzag across much of the bodywork to go between the right rear tire and the rear wing. For the rear wing to produce downforce, air should be directed away from the underside of the rear wing in order to create negative pressure. Perhaps an alternate path for the air is to go around the outside of the rear tires instead of channeling it through the bottom of the rear wing?

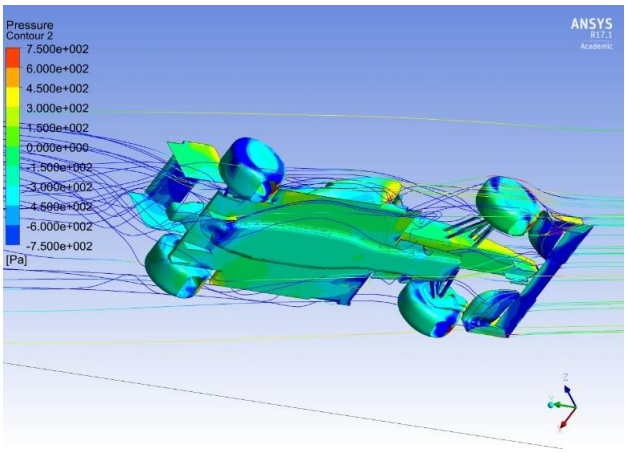

Fig 26. Representative pressure distribution for a 10° sideslip angle case

In Figure 26, a bottom view of the car is shown. From the pressure contour, it clearly shows that both the front wing and rear wing have negative pressure on the bottom surface, therefore creating downforce. Additionally, the under-chassis diffusor at the backend of the car also shows negative pressure, indicating that it is aiding to create downforce. The streamlines behind the rear wing as it exits the vehicle is interesting: the swirling nature of the streamlines shows vortex formation behind the vehicle. Depending on the degree of vortex formation and the dissipation duration, vortex-generated lift may have some effect which should be addressed in a follow-on study. When looking at the bottom of the car, we would also like to point out the treatment of the interface between the tires and the ground: the interface is not modeled as a line-contact but instead it is modeled as a patch where the tires intersect with the bottom wall of the enclosure (i.e. the computational domain). This creates an opening at the bottom of the tires because the inside of the car is hollow. This treatment has proven to work well in our analysis.

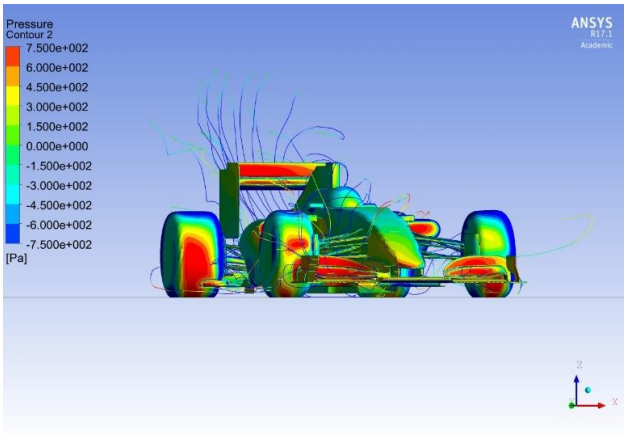

Fig 27. Representative pressure distribution for a 12.5° sideslip angle case

In Figure 28, it shows a sideview of the car. The most prominent feature of this figure is the streamlines after they exit the car – it shows the "rooster tail" flow pattern behind the car and the vortex generated. Viewed from another angle in Figure 27, the "rooster tail" behavior is seen again. This frontside view is looking along the freestream direction. Another thing this figure shows is that the streamlines curve around the top of the nose from left-to-right (in positive x), therefore a more rounded nose design may work better to delay separation for this turning case. One thing to note when looking at the pressure contour on the surface of the car is that while the red patches on the front wing and the rear wing are necessary to create downforce, the red patch on the right rear tire does not create downforce. While the front wing does a good job of minimizing the red patch on the front tires, but in this case of turning 12.5° the rear tires are completely exposed so it behooves to add some sort of aerodynamic covering to minimize the red patch on the rear tires when the car is turning.

A rear view of the backside pressure contour is shown in Figure 29. A high pressure region can be seen on top of the driver's helmet. Closely related to this red spot is that right behind the driver's helmet there is a rapid forming vortex as seen in Figure 30. This vortex does not appear to be intentional in the car's design, so it may warrant a closer look to avoid unintentional air disturbance. One way to address this helmet generated vortex is to change the helmet design so the wake coming off the helmet goes cleanly into the back of the car.

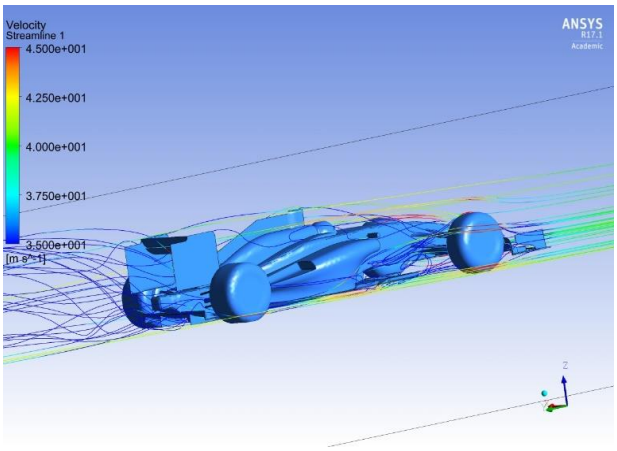

Fig 28. Representative streamlines for a 12.5° sideslip angle case

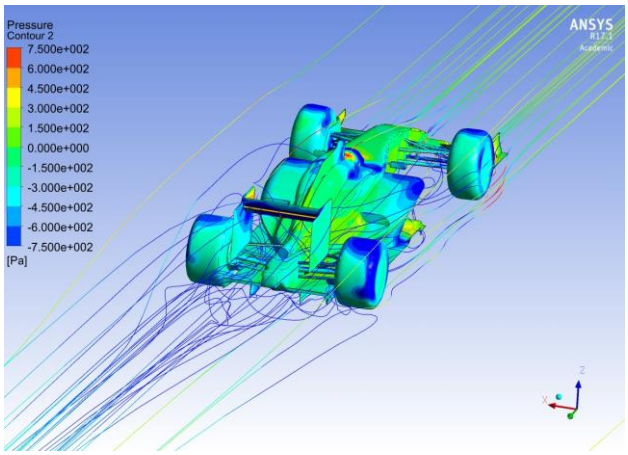

Fig 29. Representative pressure distribution for a 2.5° sideslip angle case

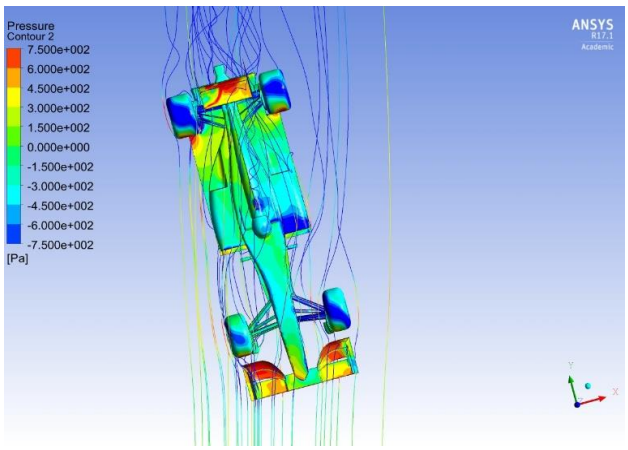

Fig 30. Representative pressure distribution for a 15° sideslip angle case

In this image of 15° sideslip angle, because the freestream is flowing past an asymmetrical shape it is broken up, with more of the flow directed to the left side of the car as shown on this image. Another thing we noticed is that on top of the sidepod to the right of the driver there is a blue patch, showing that it's a low-pressure region. In terms of streamlines, behind the car the flow appears to be very messy. This could be because of a multitude of reasons: turbulence formation, vortex bubble bursting, and flow unsteadiness. When looking at the behavior behind the car perhaps it would be best to perform a transient analysis and look at periodicity, Von Karman street, and vortex shedding because it's unsteady flow.

5.2 Survey of Twenty FIA Circuits

To see how the car will perform, first we need to look at the track it will run on. For aerodynamic performance, tailoring the wing design to the commonly raced track is important because it defines the environment the vehicle must go through. The races typically take place on either open roads (e.g.

Dakar Rally) or closed tracks (a.k.a. circuit racing). Road-courses are used in the Formula 1, 2, and 3 series, and to some extent the IndyCar series when it is not run on an oval track. Example used here are the 20 race tracks that were used during the 2017 Formula 1 racing season. In grand prix racing, a typical track has between 15 and 20 turns that the cars must navigate. When all turns are considered, the direction of turns on a race track is almost never 50/50, meaning it would either have a right-turning bias or a left-turning bias, which in some parts is influenced by whether the race is run in a clockwise or a counter-clockwise direction. Because of this, an oblique wing concept was proposed by the authors that was inspired by NASA's oblique-wing aircraft created by the legendary R.T. Jones [32]. The idea is that by tilting the wing of a race car one way, it can take advantage of the higher percentages of turns while making a sacrifice for the lower percentage of turns.

The 20 race tracks are spread out across 5 continents and 20 different countries. In grand prix racing the number of right-hand turns and the number of left-hand turns are typically different for each track, and whether by design or coincidence it is almost never even. The 20 racetracks shown in Table 4 reveals turning biases ranging from 0.25 (to the left) to 0.75 (to the right) when all 15~20 turns are considered. To illustrate the concept of turning-bias: a drag strip that's used in the Top Fuel series would have a 0.0 turning bias while a circular track that's used in the Sprint Car series would have a 1.0 turning bias. Because of this turning bias if a race car was designed for the predominant direction of turns it would gain advantage over cars that are designed for no bias. Perhaps this is the next level of aerodynamic optimization, when an aero package is designed to not just give different level of downforce but also for different turning bias.

5.3 Turning-angle Analysis from 0° to 15°

In order to take advantage of the turning bias of a race track, one must understand the turning aerodynamics of a race car over varying degrees of turns. This "turning aerodynamics" is simulated in CFD by changing the freestream angle relative to the race car. This is accomplished by changing the boundary condition in the solver but keeping the same mesh. The freestream angle is varied from 0° to 15°, at 2.5° increments. This allows the prediction of lift force and drag force when the car is at an angle to the freestream, simulating various degrees of turn. The simulation results are tabulated in Table 5 and 6. The drag force and the downforce are direct outputs from ANSYS Fluent, while the C_d and C_l were computed using freestream velocity and reference areas as indicated. In particular, C_d was calculated using the windward coordinate system that is defined on the ground plane by a vector parallel to the incoming freestream and a vector perpendicular to the freestream. The drag force is obtained by performing a coordinate transformation from the body coordinate system (i.e. Fx and Fy used in ANSYS) to the windward coordinate system. The drag coefficient C_d is calculated using frontal area projection in the windward coordinate system, so as the sideslip angle β increases the projected area increases as well. The tabulated drag and lift force (negative) are plotted in Figures 31 and 32. As shown by the figure the highest downforce occurs at 0° (i.e. going in a straight line). It then reaches a minimum at 7.5° β before coming back up at 15°. Figure 32 shows the upper bound of drag force D^* (i.e. dynamic pressure multiplied by projected area) as well as the actual drag force for each level of β.

$$C_l = \frac{L}{\frac{1}{2}\rho U^2 A_{wing}} \tag{7}$$

$$C_d = \frac{D}{\frac{1}{2}\rho U^2 A_{proj}} = \frac{D}{D^*} \tag{8}$$

In Table 6, the drag force D is computed as the vector sum of Fx and Fy. For the numerical results shown in Tables 5 and 6 the freestream velocity is 41.6 m/s (150 km/hr) and the air density is 1.2 kg/m3.

From the simulations, the drag force increases each time the car is turned. This came as no surprise since the projected area also increases when the car is turned; however, when the area is divided out to calculate the drag coefficient (i.e. dividing by projected area and dynamic pressure $\frac{1}{2}\rho U^2$), the resulting plot is surprisingly similar to the lift curve. The significant finding here is that downforce is minimum at 7.5° β. This is not a monotonically increasing or decreasing trend but a local minima of the aerodynamic performance.

Table 4 Percentage of right-handed versus left-handed turns from circuits used in the 2017 F1 season

Circuit	# of turns	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	RHT	LHT	RH%	LH%
Albert Park, Australia	16	R	L	R	L	R	R	L	R	R	L	L	R	R	R	L	R								10	6	**0.63**	**0.38**
Shanghai International Circuit, China	16	R	R	L	L	R	R	L	R	L	L	L	R	R	R	-	L								8	7	**0.53**	**0.47**
Bahrain International Circuit	15	R	L	R	R	-	R	L	R	L	L	L	R	R	R	-									8	5	**0.62**	**0.38**
Sochi Autodrom, Russia	19	R	R	L	L	R	R	R	R	L	L	R	-	L	R	L	L	R	R	R					11	7	**0.61**	**0.39**
Catalunya, Spain	16	R	L	R	R	L	-	L	R	R	L	L	R	R	L	R	R								9	6	**0.60**	**0.40**
Monte Carlo, Monaco	19	R	-	L	R	R	L	R	R	R	L	R	L	L	R	R	L	-	R	R					11	6	**0.65**	**0.35**
Circuit Gilles Villeneuve, Canada	13	L	R	R	L	R	L	R	R	L	R	-	R	L											7	5	**0.58**	**0.42**
Baku, Azerbaijan	20	L	L	L	R	L	R	R	L	R	L	R	L	L	-	L	L	-	L	R	R				7	11	**0.39**	**0.61**
Spielberg, Austria	8	R	R	R	L	L	R	R	R																6	2	**0.75**	**0.25**
Silverstone, Great Britain	18	R	L	R	L	L	L	R	R	R	L	R	L	R	L	R	L	R	R						10	8	**0.56**	**0.44**
Hungaroring, Hungary	14	R	L	R	L	R	R	L	L	R	L	R	R	L	R										8	6	**0.57**	**0.43**
Spa-Francorchamps, Belgium	19	R	-	R	L	R	L	R	R	L	L	L	R	L	R	R	L	L	R	L					9	9	**0.50**	**0.50**
Monza, Italy	11	R	L	R	L	R	R	R	L	R	L	R													7	4	**0.64**	**0.36**
Marina Bay, Singapore	23	L	-	L	-	R	-	L	R	L	L	R	L	L	R	-	R	L	L	R	R	L	L	L	7	12	**0.37**	**0.63**
Sepang International Circuit, Malaysia	15	R	L	R	R	L	R	R	R	L	R	R	L	R	R	L									10	5	**0.67**	**0.33**
Suzuka, Japan	18	R	R	L	R	L	R	L	R	R	R	L	R	L	L	L	R	L	R						10	8	**0.56**	**0.44**
COTA, USA	20	L	R	L	R	L	R	L	R	L	-	L	L	R	R	L	R	R	R	L	L				9	10	**0.47**	**0.53**
Autódromo Hermanos Rodríguez, Mexico	17	R	L	R	L	R	R	L	R	L	R	L	R	L	R	-	R	R							10	6	**0.63**	**0.38**
Interlagos, Brazil	15	L	R	L	L	L	R	R	R	L	-	R	L	L	L	-									5	8	**0.38**	**0.62**
Yas Marina, Abu Dhabi	21	L	L	R	-	L	R	L	L	R	-	L	R	L	L	R	R	R	L	L	R	R			9	10	**0.47**	**0.53**

R: right-handed corner L: left-handed corner -: faux corner that is not counted

Table 5 Lift force (i.e. negative downforce) and coefficient for various sideslip angles

β	-L (N)	$C_{l,wing}$†	$C_{l,car}$‡
0°	1483.1	1.46	0.24
2.5°	1353.0	1.33	0.22
5°	1235.5	1.21	0.20
7.5°	933.9	0.92	0.15
10°	993.8	0.98	0.16
12.5°	1136.0	1.12	0.18
15°	1229.0	1.21	0.20

† Reference area using top projection of the front and rear wings only (0.96 m^2)
‡ Reference area using top projection of the entire car (5.9 m^2)

Table 6 Drag force and coefficient for various sideslip angles

β	A_{proj} (m^2)	Fy (N)	Fx (N)	D (N)	D^* (N)	C_d
0°	1.28	1030	5.3	1030	1360	0.76
2.5°	1.38	1011	95	1015	1466	0.69
5°	1.52	982	279	1021	1610	0.63
7.5°	1.59	934	394	1013	1688	0.60
10°	1.64	1023	554	1164	1743	0.67
12.5°	1.70	1067	789	1333	1801	0.74
15°	1.78	1050	1025	1468	1885	0.78

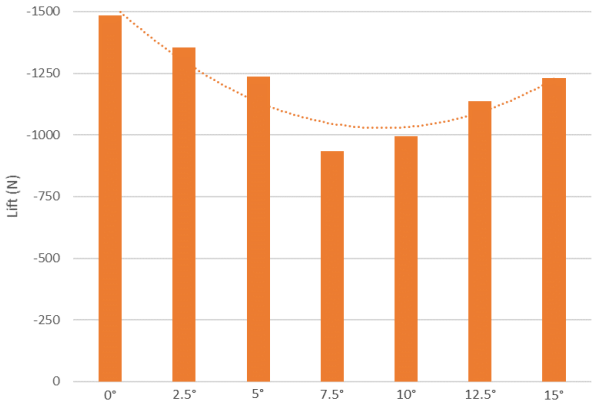

Fig 31. Downforce versus sideslip angle from 0° to 15°

It is also important to note that the shape of the downforce curve is the opposite of what's desired – namely, we see a concave curve with maximum downforce at 0° and minimum downforce at 7.5°, and this is not what we want! While this may be how a wing is designed to provide downforce in a straight line, this is not where downforce is needed. Rather, downforce is needed when a car is turning. Therefore, the optimal shape of the curve should be convex with

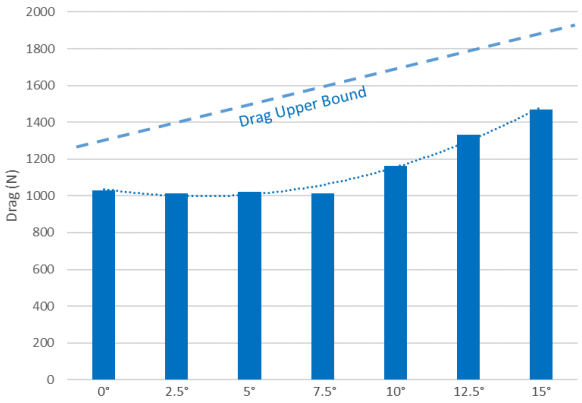

Fig 32. Drag force versus AOSS from 0° to 15°

a lower straight-line (0°) downforce and a higher turning downforce. As a way to improve this, the wing design should aim to maximize the downforce at 7.5° β. Having an uniform downforce across the range of β, the car would be better suited to turning. This is another reason why the car should be designed for 7.5° β instead of 0° β. From the results of the analysis shown here, 7.5° β is where the aerodynamic performance can benefit the most.

From the results presented in this section, it has been shown that the angle for which the wings should be designed is not 0° but rather 7.5°. Furthermore, another consideration is that, for the same reason race tracks are not symmetric, car design need not be symmetric. While we optimize the wing design for a 7.5° turning-angle, it need not be symmetric about the centerline. The authors acknowledge that in certain racing series an asymmetrical wing design may not be allowed, but this paper only looks at the performance aspect and makes no inference to racing regulation which is bound to change from time-to-time. Even though most cars today are designed with left-and-right turn symmetry, we are aware of no reason for this from a mechanical nor aerodynamic standpoint. Case and point: Champcars have the driver position shifted to one side. Moreover, with the fact that many Grand Prix circuits used today typically have a left or a right-turn bias, depending on whether the car is raced in a clockwise or an anti-clockwise direction, an asymmetrical car design may be advantageous. Because of this, the wings should be tilted one way, much like the NASA oblique wing aircraft AD-1 that has a tilted wing (Figure 33). A depiction of this concept for a race car is shown in Figure 34.

Fig 33. NASA AD-1 oblique-wing aircraft

5.4 Computational Geometry

5.4.1 Two Different CAD Models

In this section, the computational geometry is described by two separate CAD models: a conventional car model and an oblique-wing car model. The conventional car model was downloaded from the internet [source: grabcad.com] and the oblique-wing car model was created by modifying the conventional car model. The oblique-wing car model took the conventional car model and rotated the front wing and the rear wing by 7.5°; other than this difference the two car models are identical. However, the models contain surface geometries as described in section 4 that had to be resolved first. Also, the geometry had to defeatured as it contained too many fine features and individual faces. The geometry further contains internal features which had to be deleted because only the outer envelope is used in CFD calculation. Ultimately, what remains is a static outer shell of the car that's used to compute external flow aerodynamics. One additional thing about the 7.5° model is that in order to rotate the front wing, a portion of the nose cone is also rotated 7.5° with it, therefore this model is also referred to as the slanted nose model.

5.5 Mesh Generation Strategy

5.5.1 Both Models are Meshed Separately but Using Similar Mesh Parameters

Because the geometry is different for the two models, each model has to be meshed separately. The resulting mesh size from the automesher for the two models turns out to be fairly similar, therefore their computational time is similar as well. Since the wings on the oblique-wing model are turned 7.5 degrees, when the car goes into a turn the wings would straightened-out in relation to the freestream. Figure 34 shows the meshes that are used in this study: although they appear similar the top mesh (a) is the conventional car model and the bottom mesh (b) is the oblique-wing car model. They are both hybrid meshes with structured boundary layer elements and unstructured tetrahedral volume elements. To analyze the turning aerodynamics, the inlet freestream angle is changed from 0° to 7.5° instead of rotating the model inside the computational domain so we can employ the same mesh. In other words, we kept the same mesh but modified the boundary condition. This way, analyses could be carried out more quickly and the data compared more easily. As a result one idealization was made: the same car model was used for both the straight-line case and the turning case, hence the tires were not turned because that would change the geometry and be a more complicated study.

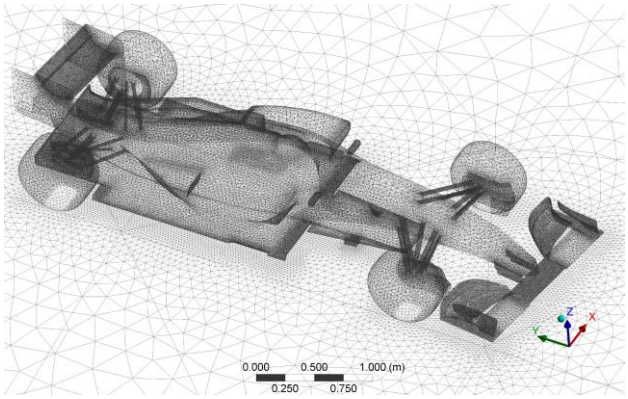

Fig 34. Oblique-wing geometry mesh

5.6 Modelling Approach

5.6.1 Introducing Rolling-tire Boundary Condition

In this full car model another boundary condition was introduced to further add fidelity to the model: the tires are simulated as rotating tires. The way this is done is different from the moving ground plane boundary condition. Because the tire surface is a curved cylindrical surface, a rotational axis needs to be defined for each of the four tires. The moving boundary condition is defined in reference to these axes, and the vehicle linear velocity of 41.6 m/s is converted to an angular velocity for the given tire diameter with a unit of rad/s.

5.6.2 Run Parameters

Other boundary conditions used are also slightly different for the straight-line case and the turning case. For the turning cases, one of the sides of the

enclosure was specified as inlet (velocity-inlet) instead of as outlet (pressure-far-field) like that used in the straight-line case. Other boundary conditions used include boundary layer (inflation) on the surface of the car, and a moving ground plane matched to the airflow speed and direction. Additionally, the numerical solver chosen was a density-based solver, and the turbulence model used was Transition SST (Shear Stress Transport). The CFL number was adjusted for the different cases to help with convergence. We ran each case as a steady state problem to 15000 iterations using a computational domain of 55 m long, by 30 m wide, by 16 m tall.

5.7 Results

From the simulation results of conventional versus oblique-wing cars, they show that the desired result is obtained when the wings are rotated 7.5°. By slanting the wings, the downforce is increased for the turning case and is decreased for when the car is going straight. This comes as no surprise as it has been shown through sideslip angle analysis that the downforce is minimum when the freestream is at 7.5° to the car. Combining this with the result that drag force does not change too much when the wings are tilted, this will allow the oblique-wing car to go faster both on the straight as well as during cornering. This is significant because Formula 1 aerodynamic design today already offer three wing options: low downforce, medium downforce, and high downforce. To go one step further, we would have to look at the asymmetrical nature of the tracks and optimize it for each track's turning bias.

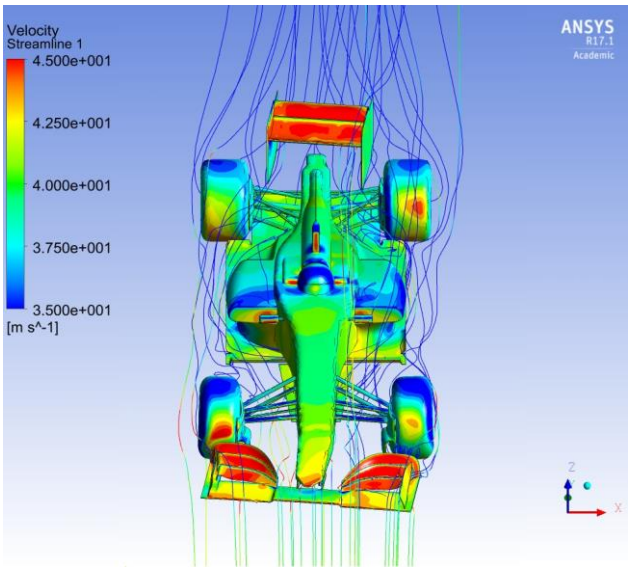

Fig 35. Oblique-wing simulation result

Finally, it is important to note that this aerodynamic advantage only works for turning in one direction, and it behaves oppositely when turning in the other direction. However, if one were to focus on biasing the wing in the direction of predominant turns, as has been shown in the turning-bias analysis of the 20 racetracks, the advantage of the oblique wing geometry will materialize over the course of an entire race. A typical race has approximately 900 turns lasting a little over an hour.

6 Overall Discussion

6.1 Improving Mesh Density for Better Solution Quality

Combining the lessons learned in the preceding three sections of this paper, first a technique was demonstrated for solving a detailed geometry with very fine features. The mesh size used was the maximum allowable; one could argue that it's easier to use better hardware to solve bigger problems, however one should still know the maximum size his hardware can handle so that subsequent runs can fully utilize the hardware's capability. For the cases presented here all calculations were performed with double-precision, and the largest mesh size our hardware could solve was determined to be 4 million grid points that all subsequent runs conformed to.

Additionally, alternative hardware capability using cloud computing was explored. Cloud computing was hosted by Rescale Inc. and Amazon Web Service (AWS). The same problem that was solved in 7 days on our computers was solved in 1/10th of the time in the cloud. The reason for the speed-up was because the number of processor cores requested was 144, instead of 7 on our computers. Despite the fast speed, one still needs to prepare the job to run in batch mode because there's no runtime GUI to display the status. Hence, if one is willing to pay to send the job to the cloud, one would get the results back much quicker.

6.2 Ensuring Model Size fit within Hardware Limitations

When the mesh gets too big for the computer to handle there are some telltale signs. During computation, if one looks at Task Manager inside Windows, one would see that CPU Usage is turning on-and-off instead of a steady percentage when data is written out to the hard drive. The reason for this is because there's not enough RAM to store the entire problem while the solver is calculating. One would

see a period of 100% CPU Usage followed by a period of 0% CPU Usage when the solver has to take a break to write the data out. While it's doing that, the % Disk Usage would become 100%. When the CPU cannot be used to compute the problem continuously, it is a sign that the mesh size is too large for the computer hardware.

When one sees that the mesh size is too big the solution is to go back and relax the default element size. Using the wing geometry of section 4 as an example, if the element size is specified as 5.00e-3m it will result in 4.14 million elements. If it is changed to 5.12e-3m it will reduce to 3.96 million elements. If it is changed to 5.20e-3m it will reduce to 3.85 million elements. As mentioned earlier, a 4 million grid mesh is what our lab's computer can handle. A quick way around this problem is to add more RAM to the computer. Another way is to send the job to the cloud that has seemingly unlimited memory. As we have also showed, the 4 million mesh size is hardware dependent; if the problem is solved in the cloud, the mesh size can be increased. However, the user should still know the limit of that particular hardware in order to fully utilize its capability and minimize billing from a third party. Finally, all of these methods for geometry preparation and meshing take time, and it is an iterative process that requires going back-and-forth between CAD and mesher in order to create a successful mesh for computation.

6.3 Comparing Design Variations by Geometry Manipulation and Remeshing

Seven design iterations were solved and compared using aerodynamic efficiency L/D as the metric. The same solver parameters were used for all seven cases so that the results can be directly compared. The grid density was adjusted so that the mesh was as close to 4 million grid points as possible to fully utilize the hardware's capability. Results of the seven cases were then plotted and one intermediate configuration was found to be the most efficient aerodynamic configuration. In our study, configuration 4 with 11 airfoils is proven to be aerodynamically efficient; any airfoils added after that does not increase aerodynamic efficiency. The study is conducted up to 27 airfoils for configuration 1. Utilizing this iterative design technique, one can determine when the aerodynamic efficiency will not improve simply by adding airfoils.

6.4 Lessons Learned in Full-car Analysis

Since the lowest downforce and the lowest drag coefficient happen at 7.5° sideslip angle, the wing should be turned 7.5° with respect to the car. This way, when the car is going in a straight line, it will experience the lowest drag and the least amount of downforce. This will allow the car to have the highest straight-line speed. When the car turns, the downforce will increase from this minimum point. Having a higher downforce when the car is turning will force the tires onto the pavement and thereby increase the friction force to turn the car. If the circuit is left-turn biased, then the wing should be tilted 7.5° to the left. This is because the parabolic curve in Figure 31 is non-symmetric, therefore it would be more advantageous to move to the side that has the higher downforce (namely the 0°-side). With a higher downforce, the car can travel at a higher speed around the corner and carry that speed onto the following straight. The opposite applies when the circuit is right-turn biased: in this case, the wing should be tilted 7.5° to the right for the same effect.

A new wing design with this oblique angle will allow the car to benefit from turning-biased circuits. For a circuit with more turns in one direction than the other, an oblique wing designed to take advantage of the predominant turning direction, while sacrificing on turns in the less common direction, would result in a faster overall lap-time. As an example, Figure 34 shows a car with its front and rear wings turned to the left to suit a left-turn biased circuit. While one lap may not make a big difference, over the course of a race where some 50 laps are completed, the accumulated effect could be big. With that said, the analysis presented here only lays the foundation: additional considerations such as cornering speed, turning duration, changes in elevation, as well as attitude of the car upon approach, should all be taken into account to truly optimize the oblique angle for any specific circuit.

6.5 CFD Results for New Oblique-wing Model

Finally, things learned in sections 3 and 4 were used in section 5 to explore a new design concept – that of an oblique wing race car [33, 34]. A new CAD model was created to introduce an oblique angle of 7.5° to the front and the rear wings of a race car (Figure 35). Initially, the full-car model created a lot of grid points, but with subsequent geometry cleanup and defeaturing the final mesh size was reduced to 4 million, and the problem was solved in a week's time. As shown in Figure 36 the result showed an increase in downforce when turning and a decrease in downforce when going straight. This is the desired outcome! To solve this problem and generate this result the three-stage process described in sections 3,

4, and 5 is used: high-fidelity mesh generation, sequential design variation, and application to a complete car and its oblique wing variant. Through all this work we've demonstrated that oblique wing geometry is an improvement to the current race car design.

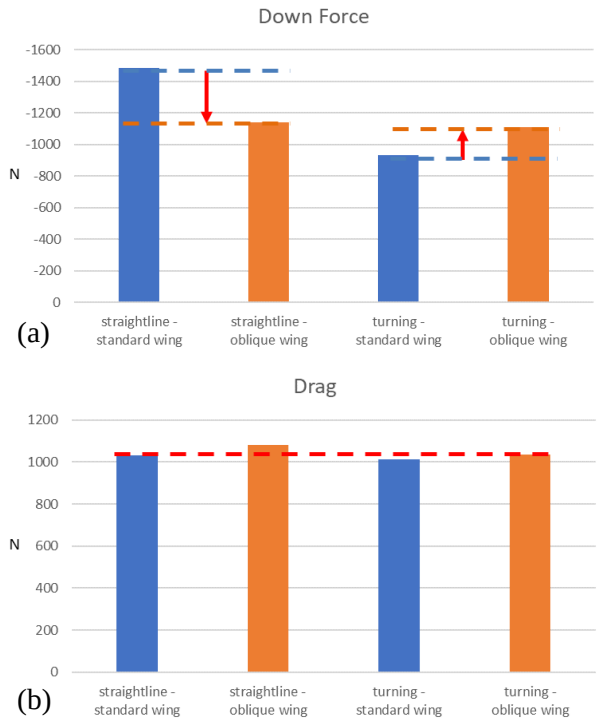

Fig 36. (a) Downforce and (b) drag force for conventional geometry vs. oblique-wing geometry

7 Conclusion

A number of techniques have been demonstrated in this study: CAD model sectioning, geometry cleanup, enclosure creation for computational domain, object shape extraction, mesh refinement for fine gaps, mesh size reduction, moving ground plane boundary condition, rotating-tire boundary condition, tire contact-patch treatment, turning-angle analysis, and cloud computing. These techniques are all valuable in allowing the whole-car model to be solved. Additional solver settings were also implemented: temperature and pressure limiters, reducing the CFL number, deactivating convergence check to allow the CFD solver to successfully run to 15000 iterations. All of these techniques and settings made it possible to generate qualitative pressure and streamline plots as well as quantitative numerical force data that are presented in this study.

This study has shown the benefit of oblique wing design for race cars to improve its cornering ability. By improving downforce during corning as well as reducing downforce when going straight, oblique wing geometry has shown its advantages over symmetrical wing design. The study started out by demonstrating a mesh generation strategy to create a high-fidelity mesh that captures the local flow structure of today's complicated race car front wing. Next, a sequential design methodology was carried out to look at the effects of vehicle aerodynamics when the computational geometry is changed to a different airfoil configuration. Finally, the techniques developed here were applied to a full-car model as well as its oblique-wing variant. The problem was solved successfully and data was produced to support the hypothesis of using oblique wing design to improve race cars' aerodynamic performance.

References:

[1] Katz, J., *Race Cars Aerodynamics*, Bentley Publishers, 2nd Edition, 2006.

[2] McBeath, S., *Competition Car Aerodynamics*, Veloce Publishing, 3rd Edition, 2015.

[3] Toet, W., "Aerodynamics and Aerodynamic Research in Formula 1," *The Aeronautical Journal*, 117(1187), 1-26. doi:10.1017/S0001924000007739

[4] Bienz, C., Larsson, T., Sato, T., Ullbrand, B., "In Front of the Grid - CFD at SAUBER PETRONAS F1 Leading the Aerodynamic Development," *European Automotive CFD Conference*, 2008.

[5] McLaren Racing Limited, "A brief history of computing in Formula 1," *https://www.mclaren.com/formula1/team/abrief-history-of-computing-in-F1-1052199/*, 2016.

[6] Ogawa, A., Yano, S., Mashio, S., Takiguchi, T., Nakamura, S., Shingai, M., "Development Methodologies for Formula One Aerodynamics," *Honda R&D Technical Review, F1 Special (The Third Era Activities)*, 2009, pp. 142-151.

[7] Lomax, H., Pulliam, T., Zingg, D., *Fundamental of Computational Fluid Dynamics*, 2002.

[8] Anderson, J. D., Jr., *Fundamental of Aerodynamics*, 5th Edition, 2011.

[9] Chandra, S., Lee, A., Correll, S., Jensen, G., "CFD Analysis of PACE Formula 1 Car," *Computer-Aided Design & Applications*, PACE (1), 2011, pp. 1-14.

[10] Williams, M. and Shires, A., "Validation of CFD for Racing Car Analysis and Design," *SAE Technical Paper 2002-01-3349*, doi:10.4271/2002-01-3349, 2002.

[11] Ueno, D., Hu, G., Komada, I., Otaki, K., et al., "CFD Analysis in Research and Development of

Racing Car," *SAE Technical Paper 2006-01-3646*, doi:10.4271/2006-01-3646, 2006.
[12] Strachan, R. K., Knowles, K., and Lawson, N. J., "A CFD and Experimental Study of an Ahmed Reference Model," *SAE Technical Paper*, doi:10.4271/2004-01-0442, 2004.
[13] Jones, R. T., "The Oblique Wing – Aircraft Design for Transonic and Low Supersonic Speeds," *Acta Astronautica*, 4, 1977.
[14] ANSYS, Inc., *Introduction to ANSYS Meshing, 15.0 Release, Training Manual*, 1st Edition (Mar. 2014), Inventory #000576, 2014.
[15] ANSYS, Inc., *Introduction to ANSYS Fluent, 15.0 Release, Training Manual*, 1st Edition (Mar. 2014), Inventory #000575, 2014.
[16] Wade, A., "Why I became a fan of Fluent Meshing to create large meshes of complex geometries," *https://www.ansys-blog.com/became-fan-fluent-meshing-create-large-meshes-complex-geometries/*, 2014.
[17] Lin, M. and Papadopoulos, P., "Aerodynamic benefits of oblique wing geometry for race cars by CFD analysis," *Proceedings of the 26th Annual Conference on Computational Fluid Dynamics Society of Canada (CFD2018)*, MS-428, 2018.
[18] Dickerson, M. T., Keil, J. M., and Montague, M. H., "A large subgraph of the minimum weight triangulation," *Discrete & Computa-tional Geometry 18*, 1997, pp. 289-304.
[19] Blelloch, G. E, Miller, G. L., and Talmor D., "Developing a practical projection-based parallel Delaunay algorithm," *Proceedings of the Twelfth Annual Symposium on Computa-tional Geometry*, 1996, pp. 186-195.
[20] Alliez, P., Cohen-Steiner, D., Yvinec, M., and Desbrun, M., "Variational tetrahedral meshing," *ACM Transactions on Graphics 24*, 2005, pp. 617-625.
[21] Su, P. and Drysdale, R. S., "A comparison of sequential Delaunay trian-gulation algorithms," *Proceedings of the Eleventh Annual Symposium on Computational Geometry*, 1995, pp. 61-70.
[22] GoEngineer, *SolidWorks Essentials, SolidWorks 2013 Training*, Dassault Systèmes SolidWorks Corporation, Part #PMT1300-ENG, 2013.
[23] ANSYS, Inc., *Introduction to ANSYS DesignModeler, 15.0 Release, Training Manual*, 1st Edition (Feb. 2014), Inventory #000570, 2014.
[24] Mahdi Al-Obaidi, A., and Sun, L., "Calculation and Optimization of the aerodynamic drag of an open-wheel race car," *Journal of Engineering Science and Technology EURECA*, 2014.
[25] Menter, F. R., Kuntz, M., Langtry, R., "Ten years of industrial experience with the SST turbulence model," *Turbulence, Heat and Mass Transfer*, 4, 2003.
[26] Turner, T. R., "Wind-tunnel investigation of a 3/8-scale automobile model over a moving-belt ground plane," *NASA TN D-4229*, NASA Langley Research Center, 1967.
[27] Desai, M., Channiwala, S., and Nagarsheth, H., "Computational Validation of Experimental Aerodynamic Predictions of a Car," *International Conference on Fluid Mechanics and Aerodynamics*, 2008, pp. 125-128.
[28] Lin, M. and Papadopoulos, P., "Application of computer aided design tools in CFD for computational geometry preparation," *ATINER's Conference Paper Proceedings Series*, MEC2018-0102, ISSN: 2529-167X, 2018.
[29] CFD Online, *https://www.cfd-online.com/Forums/blogs/far/1555-best-mesh-generation-software.html*, 2012.
[30] Wikipedia, *https://en.wikipedia.org/wiki/Gurney_flap*, 2018.
[31] Wikipedia, *https://en.wikipedia.org/wiki/Lift-to-drag_ratio*, 2018.
[32] Lin, M. and Papadopoulos, P., "Oblique CFD analysis of open-wheel race car with circuit-tailoring wing optimization," *Tenth International Conference on Computational Fluid Dynamics (ICCFD10)*, Barcelona Spain, Poster session, 2018.
[33] Larrimer, B. I., *Thinking obliquely: Robert T. Jones, the Oblique Wing, NASAs AD-1 Demonstrator, and its legacy*, Washington, DC: NASA, 2013.
[34] Jones, R. T., "The Oblique Wing – Aircraft Design for Transonic and Low Supersonic Speeds," *Acta Astronautica*, 4, 1977.
[35] Painter, W., "AD-1 oblique wing research aircraft pilot evaluation program," *Aircraft Design, Systems and Technology Meeting*, doi:10.2514/6.1983-2509, 1983.