ATINER's Conference Paper Proceedings Series
MEC2018-0102
Athens, 27 September 2018

Application of Computer Aided Design Tools in CFD for Computational Geometry Preparation

Mark Lin and Periklis Papadopoulos

Athens Institute for Education and Research
8 Valaoritou Street, Kolonaki, 10683 Athens, Greece

ATINER's Conference Paper Proceedings Series
MEC2018-0102
Athens, 27 September 2018
ISSN: 2529-167X

Mark Lin, Graduate Student, San Jose State University, USA
Periklis Papadopoulos, Professor, San Jose State University, USA

Application of Computer Aided Design tools in CFD for Computational Geometry Preparation

ABSTRACT

Computer Aided Design tools (CAD) allow geometries to be constructed in software so they can be used in manufacturing (CAM), analysis (CAE), and measurement (CMM). For performing Computational Fluid Dynamics analysis (CFD), although a mesh can be constructed in certain cases from scratch in a meshing software, more often it is created and imported from a CAD software. Starting from a positive shape geometry, various steps in geometry simplification, negative volume extraction, meshing, and solving are described and illustrated in this paper. The two CAD software used in this study are Solidworks by Dassault Systèmes and DesignModeler by ANSYS Inc. Their usage, file transfer format, and the issue of surfaces vs. solids are illustrated in this paper through a geometry example. Other common problems such as being unable to generate a volume mesh or the mesh size is too large, are also presented and discussed. Finally, a successfully completed mesh is shown, along with its CFD computational results.

Keywords: CFD, Numerical Analysis, Meshing, Aerodynamics, Automotive Racing.

Acknowledgments: Our thanks to Professor N. Mourtos for the use of E164 Engineering Lab to run our CFD code. We also thank the Charles W. Davidson College of Engineering for providing the ANSYS software license.

Introduction

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 CFD problem is not solvable, no matter how advanced the solver algorithm is. 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 that are only suitable for teaching purposes; on the other hand, industrial applications are likely to have products that are described by complex 3D shapes. To construct a geometry, an assortment of CAD software can be used. Two of them are used here: Solidworks by Dassault Systèmes and DesignModeler by ANSYS Inc. 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. Meshing is an iterative process and very often, much CAD manipulation has to be performed before the meshing software (ANSYS Mesher) can successfully construct a mesh. From our experience, a complex 3D geometry such as a car would typically take about 2 weeks of "massaging" the model before it can successfully complete the meshing step without generating an error message. 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.

This paper describes the method that is used and refined by the authors to take a 3D CAD model of everyday object (a car, an airplane, a boat, etc.) into the computational domain and make it available for meshing.

Many different types of problems are encountered during geometry preparation. The most common is when the CAD model contains surfaces that are not solid bodies - this is common in many fancy CAD models that are found on the internet. The first thing one needs to do is to open up the file in Solidworks to see if it contains solid bodies. If the CAD model contains surfaces and not solid bodies then it cannot be meshed (without much difficulty).

Another problem occurs when one tries to mesh the geometry: many-a-times one would get an error message saying that surface mesh has been generated but the volume mesh cannot. Without a volume mesh, the mesh is only half complete and cannot be used to solve. This is a frequently encountered problem for complex geometries that have small gaps in the fluid volume where the fluid needs to pass through.

The third problem is when the mesher successfully generates a volume mesh but the mesh size is simply too large! During initial meshing, especially for geometries that has many small features, the total mesh size can be too large to compute. It would then need subsequent refinements to get the mesh down to a manageable size. For the computers used in this study with the amount of RAM on each machine (16GB), 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 front wing geometry shown in Figure 1 resulted in 21 million cells! It took 2 weeks of CAD operations using both Solidworks and DesignModeler to get it down to a 4 million-cell mesh that's suitable for computation.

Literature Review

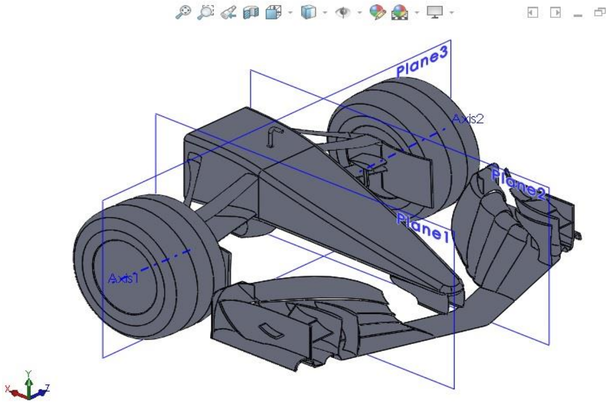
In reviewing literature on the topics of geometry preparation and meshing, three papers are cited here that provide good background information. The first paper is titled *A Large Subgraph of the Minimum Weight Triangulation* (MWT) in Discrete & Computational Geometry which talks about recent advances in solving a polynomial algorithm for the MWT. Another paper titled *Developing a Practical Projection-Based Parallel Delaunay Algorithm* talks about a parallel Delaunay triangulation algorithm (the only one that exists) that works in practice. The third paper titled *Variational Tetrahedral Meshing* and presented at SIGGRAPH in 2005, it describes an algorithm that iterates between performing a smoothing operation and recomputing the Delaunay triangulation, similar to the McCormack method used for solving. All of these papers give good background information on computational geometry construction. However, on a more practical note, the literature that was used extensively in this study are the training manuals from Solidworks and ANSYS on computer-aided design, meshing, and to some extend solver computation. These training manuals are listed in the reference section.

Various problems encountered here in computational geometry preparation and meshing are discussed in online media. For example, the problem of surface versus solid body is discussed on researchgate.net, where the blog posts describe the difficulty of filling a surface geometry to use in ANSYS Workbench. The other problem of having a mesh size that's too big is discussed in an ANSYS blog *Why I became a fan of Fluent Meshing to Create Large Meshes of Complex Geometries* where the author talks about a parallel meshing technique that utilizes many CPUs. That would be useful if one has the hardware to do that; in the study here, geometry simplification is the way to go. That article also has a mesh plot of a racecar’s frontend, which when compared to our mesh plot shown in Figure 6 the mesh in this paper looks better.

Methodology

For the problems described in the preceding sections, this section gives a step-by-step procedure of how to address these problems. Together, with a recipe of how to solve the problems, detailed descriptions of how to get an imported complex 3D geometry into the mesher is included here using an example.

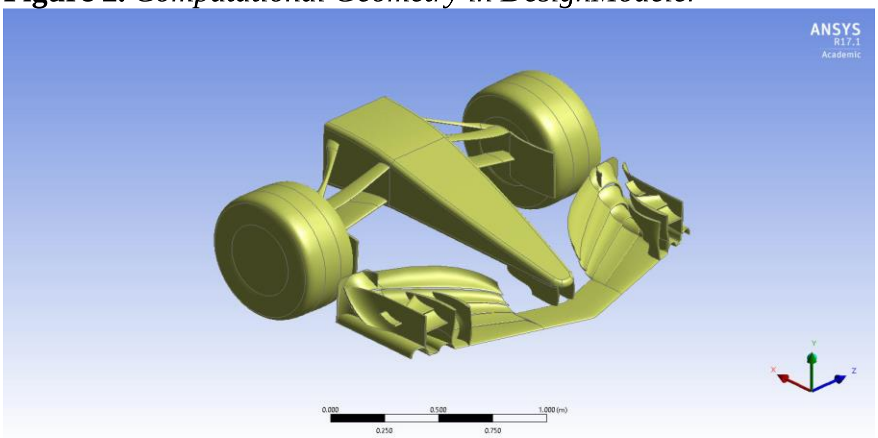
Figure 1. *CAD Geometry in Solidworks*



Source: grabcad.com.

First, a CAD geometry as shown in Figure 1 is downloaded from the internet. This geometry contains all the small features that are not relevant to CFD analysis and hence must be removed using a CAD software. This can be done by using a Solidworks command repeatedly to remove all of the small features - "Delete" in the menu bar *Insert->Face->Delete*... Together with the *Delete* command, *Extruded Cut* and *Extruded Boss/Base* commands are also used to repair the model wherever needed. Next, the model is saved in the ACIS (.sat) format to import into ANSYS DesignModeler. The reason ACIS file format is chosen is because the geometry kernel that DesignModeler is built on uses it. ACIS is better than other types of universal file format such as STEP or IGES because it minimizes file translation errors. Once inside ANSYS workbench, additional CAD operations can be performed using DesignModeler to combine surfaces and remove small features. The DesignModeler commands are *Face_Delete* and *Merge*. DesignModeler is also used to extract a negative volume from the positive shape in the original CAD model. Although Solidworks can perform the same operations, this portion is better done in DesignModeler because it is a native program within ANSYS that'll ensure compatibility when the finished model is taken into ANSYS Mesher.

Figure 2. *Computational Geometry in DesignModeler*



While CAD manipulations are performed in Solidworks and DesignModeler, one should frequently “section” the model to ensure the model is still a solid body. 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. Also, when one first begins to work with a CAD model, sometimes when 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. Sectioned geometry is illustrated in Figures 3 and 4.

Figure 3. *Sectioning in Solidworks*

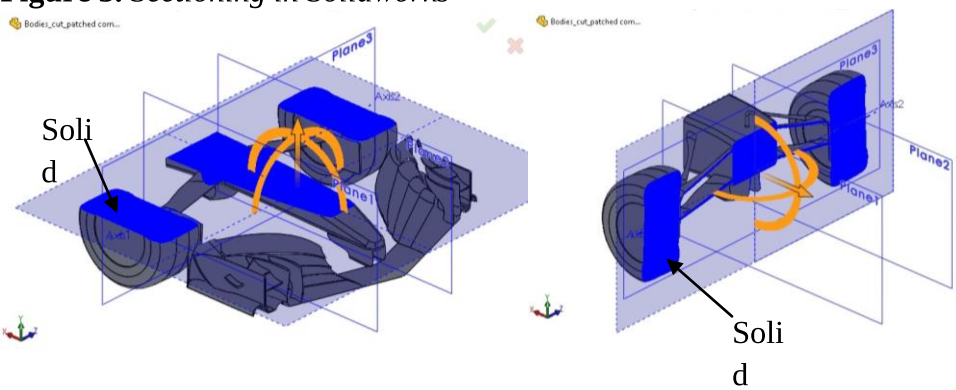
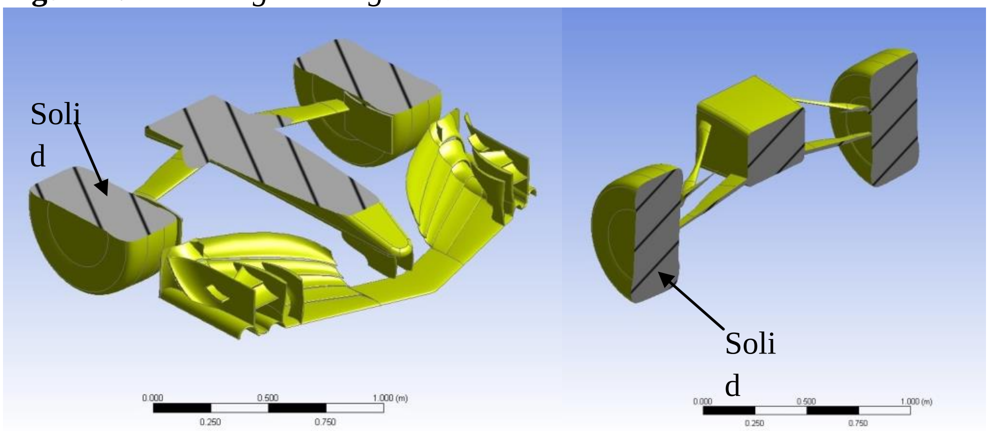
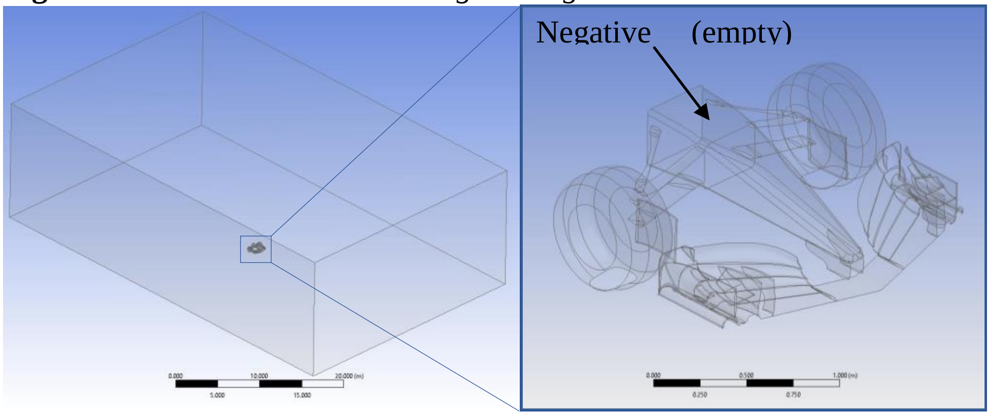


Figure 4. *Sectioning in DesignModeler*



After CAD operations have been performed to clean up the model and reduce its complexity, it is ready to be used to extract a negative volume from the fluid domain. This part of geometry setup is different from a Finite Element Analysis because in FEA the positive shape represented by the CAD model is used for analysis; in CFD, the computational domain is the fluid volume surrounding the object, hence a volume enclosure needs to be defined and the positive volume needs to be deleted from the fluid volume. This part is quite different because in the computational domain, the object of interest (cars, airplanes, etc.) is represented by empty space. Because of this, the fewer surface joints on the CAD model is better so that they won't get transferred onto the fluid volume which would produce a denser mesh. An image of the enclosure is shown in Figure 5-left. A zoomed-in image of the negative volume with the object of interest extracted is shown in Figure 5-right.

Figure 5. *The Enclosure Containing the Negative Volume*



For the next problem of when the mesh gets too big to handle, here are some tell-tale signs: during computation, if one opens up the Task Manager in Windows, one would see that CPU usage is turning on and off instead of a steady percentage while data is written out to the hard drive. This is because there's not enough RAM to store the entire problem while the solver is running. One would see a period of 100% CPU Usage and then there would be ~0% CPU Usage because the solver is writing data out to the hard drive instead of storing everything in RAM. While it's doing that, the %Disk Usage would be running at 100%. When the CPU cannot be used to compute the problem 100% all the time, this is a sign that the mesh is too large for the particular computer hardware. A solution for this problem is to send this job to the cloud which is discussed later in this paper.

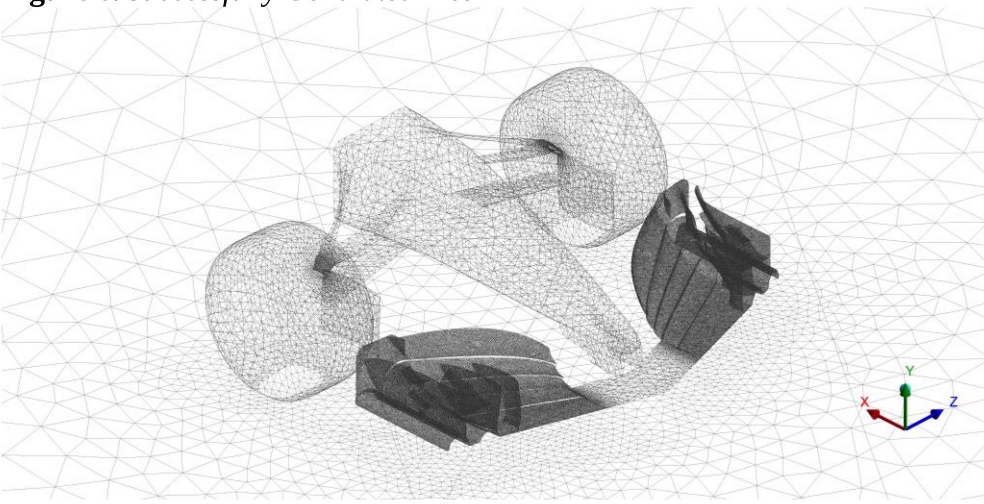
Findings/Results

Once the geometry is simplified and prepared for meshing, ANSYS Mesher is used to generate the mesh. Other commercially available meshing software can also be used to generate a mesh: the popular ones include Hypermesh, Pointwise, Fluent meshing, Chimera overset grid, and Visual-CFD for OpenFOAM.

Very often, in the first few tries the mesh generation would fail. This can happen in 2 ways: first, a surface mesh is generated but cannot be propagated to a volume mesh. Second, error messages may be generated saying "a mesh cannot be generated using the current meshing options and settings," or "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 mesh again. Seldomly is the mesh generation successful on the first try. Although this can happen for simple geometries, for real-world 3-D shapes one must do several rounds of simplification (a.k.a. defeaturing) before a successful mesh can be generated.

Figure 6 shows the mesh generated for the 3D front wing geometry used as an example here. It comprises of 3.9 million elements. Looking at the mesh plot, it is obvious that mesh refinement has been done on the individual winglets on either side because of the high density of grid elements on them. This allows for the creation of thin slots for air to pass through, and allows the intricacies of local flow structure to be captured.

Figure 6. *Successfully Generated Mesh*



In the following sections, we will show the computed results of surface pressure and streamlines on the front wing geometry. This is done using viscous modeling, with a surface boundary layer, and using the transition SST turbulence model. It is run as a steady-state problem, on a density-based solver, implicit formulation, to 15000 iterations, with a Courant number of 0.5.

Discussion

So far, three problems that are commonly encountered when trying to get a geometry to mesh has been identified. The solution to the first problem of surfaces versus solid body is to perform frequent sectioning to make sure it stays as a solid body. In the unfortunate case where the geometry starts out as surfaces, there are ways to turn it into a solid body; however, doing so is not a trivial task, and unless the user is an advanced CAD user it should not be attempted. In any case, one needs to first make sure that the geometry is a solid body, and then perform frequent sectioning in both Solidworks and DesignModeler to make sure it stays as solid.

The solution to the second problem of generating a surface mesh but not a volume mesh is to go back and simplify (i.e. defeature) the CAD model further and try again. In ANSYS Mesher, there is a command that shows the problematic geometry when meshing fails. If necessary, one should go back and delete that geometry if there's no way to fix it, and then remodel it in CAD.

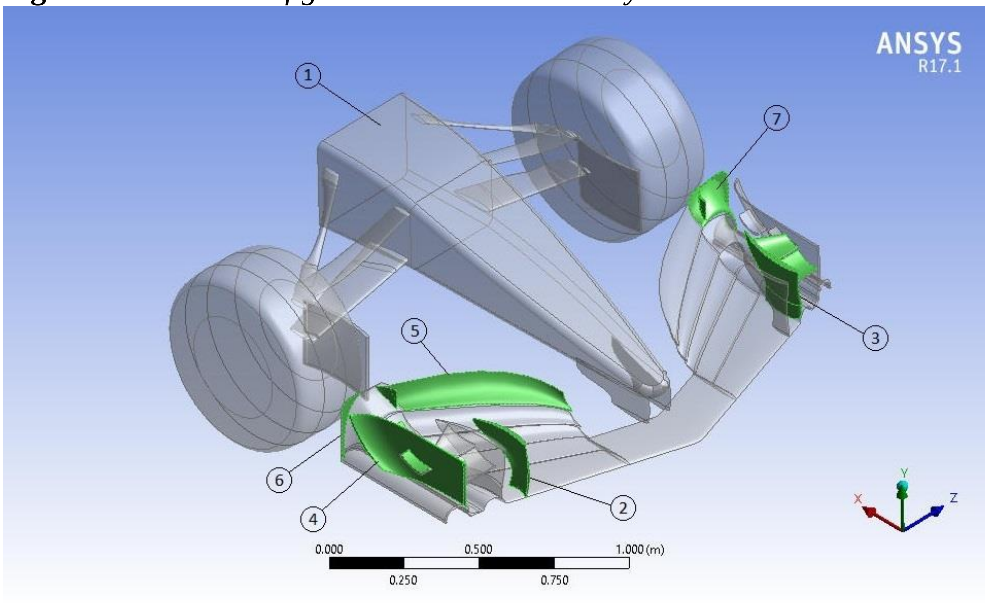
The solution to the third problem of mesh size too big is to go back and relax the default element size. For example in this wing geometry, if the element size is specified as 5.00e-3m it will result in 4.14 million elements. If it is changed to 5.12e-3m it will reduce to 3.96 million elements. If it is changed to 5.20e-3m it will reduce to 3.85 million elements. As mentioned earlier, a 4 million elements mesh is what the computers used in this study can handle. A quick way around this is to add more memory to the computer. Another way is to use cloud computing that sends the job to off-site server clusters with unlimited memory (This will be discussed later in the paper). In summary, all of these solutions for geometry preparation and meshing take time, and it is an iterative process that requires going back and forth in order to come up with a successful mesh.

Application Examples

The technique presented in previous 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 the base model (as shown in Figure 2), then individual panels are removed using DesignModeler, and finally remeshed to compute again. The seven computational geometries are numbered configuration 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 simplest version.

Figure 7 shows the various geometry configurations on the same plot. Each time, a wing element is removed from the assembly using CAD operations, which can be done in either Solidworks or DesignModeler. DesignModeler was used here because it is very easy to remove geometry features. For example, to generate configuration 2 all that needs to be done is to highlight the faces of the element, which in this case is 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 by removing the inside vertical winglet. 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.

Figure 7. *Various Configurations used in the Analysis Model*



When starting from the configuration 1 model that has already been meshed, going back to DesignModeler would require the *Enclosure* and the *Object_Delete* commands be suppressed first. This would bring back the positive geometry which can then be manipulated in DesignModeler to turn it into configuration 2. After that's completed and a check for solids has been performed, then the enclosure can be reestablished to extract the negative volume for meshing.

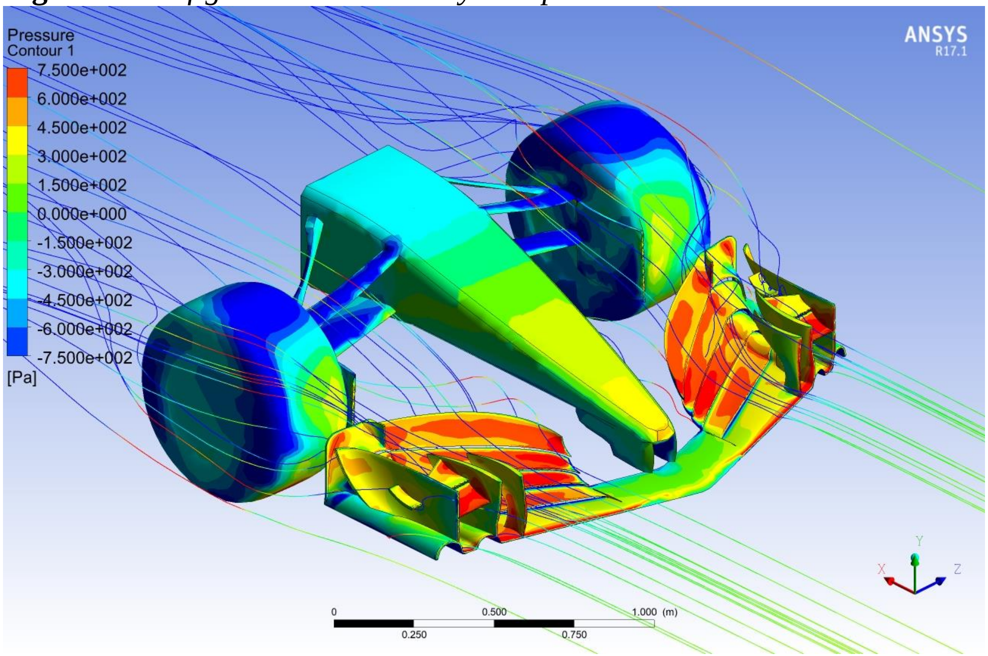
When the new geometry is taken into ANSYS Mesher to mesh, the "Named Selections" features chosen previously in DesignModeler should be checked to make sure that they still refer to valid geometry. This feature of the model can sometimes be invalid when a previously defined element is no longer there. Before meshing, the minimum element size is adjusted to a smaller number to achieve a finer mesh that is just large enough (3~4 million) for the computer to compute. After running ANSYS mesher one should look at "Mesh Statistics" for the final cell count. If the cell count is too small or too large, the minimum cell size is adjusted accordingly.

Sometimes when meshing using a different cell size, the mesher may fail and not be able to generate a valid mesh. When this situation happens, one quick fix is to adjust the cell size to something slightly larger or smaller than the failed cell size. Because ANSYS meshing uses complex algorithms, it is difficult to explain why adjusting the cell size slightly up or down would fix the meshing problem, but from our experience it has shown to work. Finally, only when a mesh-generation run returns no error messages could the mesh be used for computation (warning messages are acceptable).

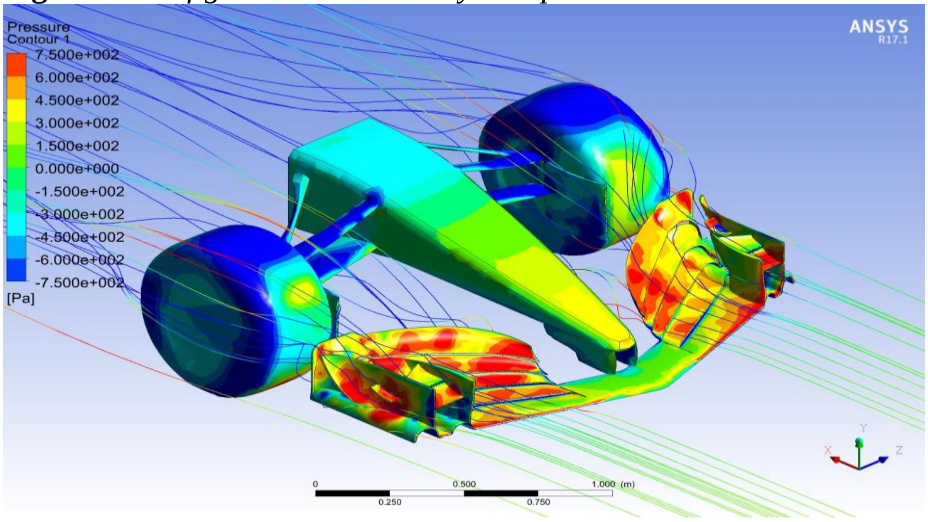
The following seven paragraphs talk about configurations 1 to 7 separately – each shows the CFD computational result for one configuration. At the end, a summary plot of aerodynamic efficiency is shown that's calculated from the resultant lift force and drag force predicted by CFD.

Configuration 1 is the full wing geometry – all of the wing elements are included in this 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 has been discussed earlier. When it is meshed, the minimum mesh size is called out to be 5e-3 m. The resulting mesh has 4.1 million elements. The lift force extracted from ANSYS CFD-Post is -1015.19 N and the drag force is 565.831 N. There are no 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 fuselage that stabilizes the airflow.

Figure 8. *Configuration 1 Geometry Computational Result*

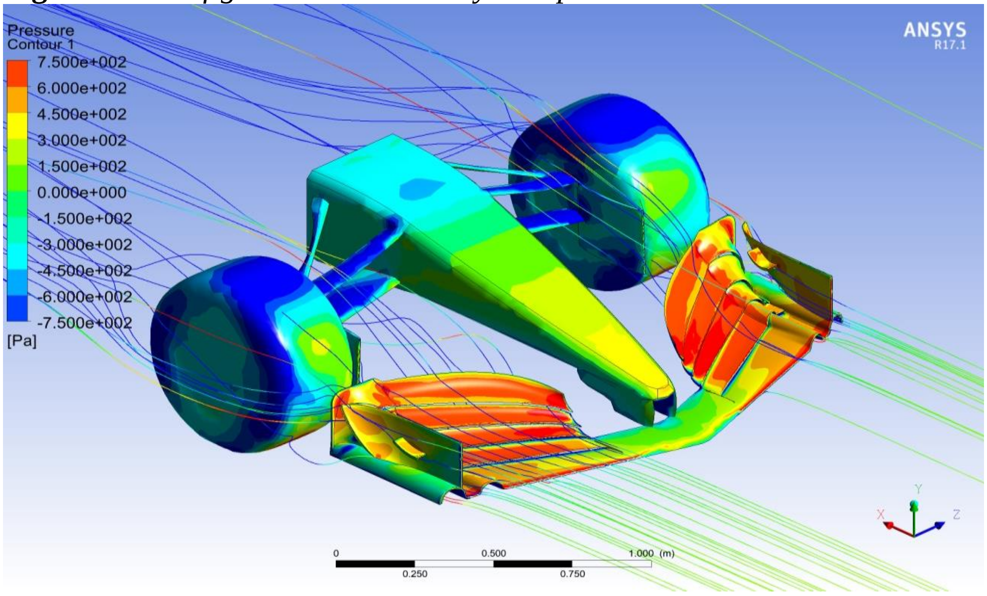


Configuration 2 is created when the four inner vertical winglets are removed from configuration 1. This is done in DesignModeler, and the process is fairly straightforward using the *Face_Delete* command. The model is then remeshed with a minimum mesh size of 5e-3 m. This resulted in a mesh with 3.9 million elements. The resultant lift force is calculated to be -984.523 N and the drag force is 545.759 N. This is the direct computational result 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.

Figure 9. *Configuration 2 Geometry Computational Result*


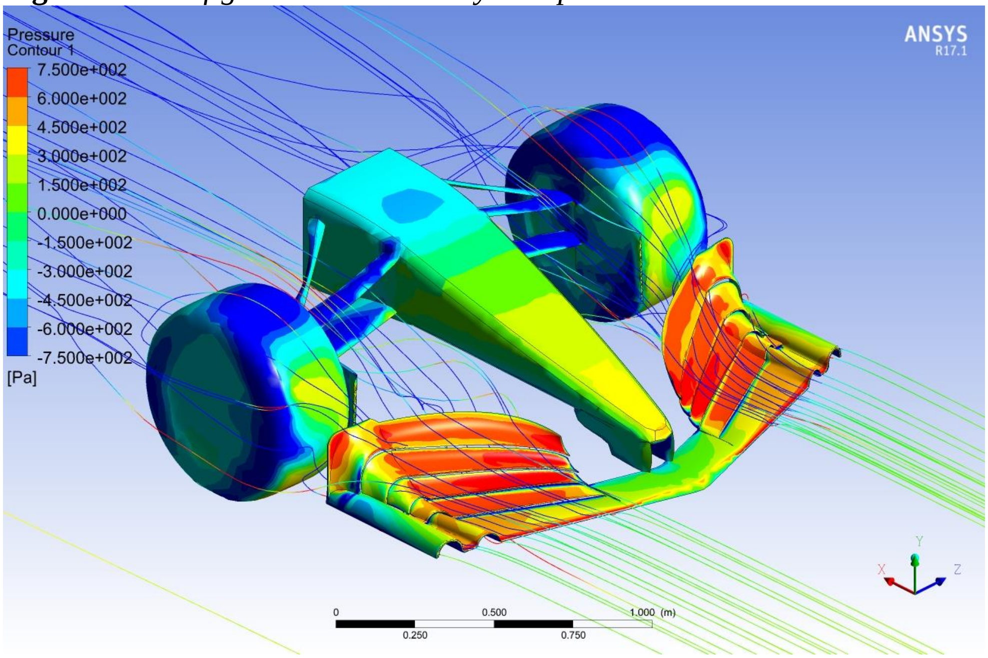
Comparing Figure 9 to Figure 8, the airflow doesn't seem to be too different. This indicates that the vertical fins in configuration 1 are not really needed.

Configuration 3 is a simplified version of configuration 2. In this configuration, the 3 inner winglets from each side (2 horizontal and 1 vertical fins from each side) are removed. Once again, this is done in DesignModeler so we don't have to go back to Solidworks. After meshing using a minimum cell size of 5e-3 m, the resulting mesh has 3.5 million elements. The predicted lift force is -1014.51 N while the drag force is 568.624 N. Once again, comparing Figure 10 to Figure 9, after the inner winglets are removed the effect doesn't seem to be very noticeable. Even though the lift (downforce) goes up, so does the drag force. When they are combined to calculate the airfoil efficiency there doesn't seem to be any aerodynamic benefits these inner winglets offer. Figure 10 shows that airflow doesn't change too much when these winglets are removed.

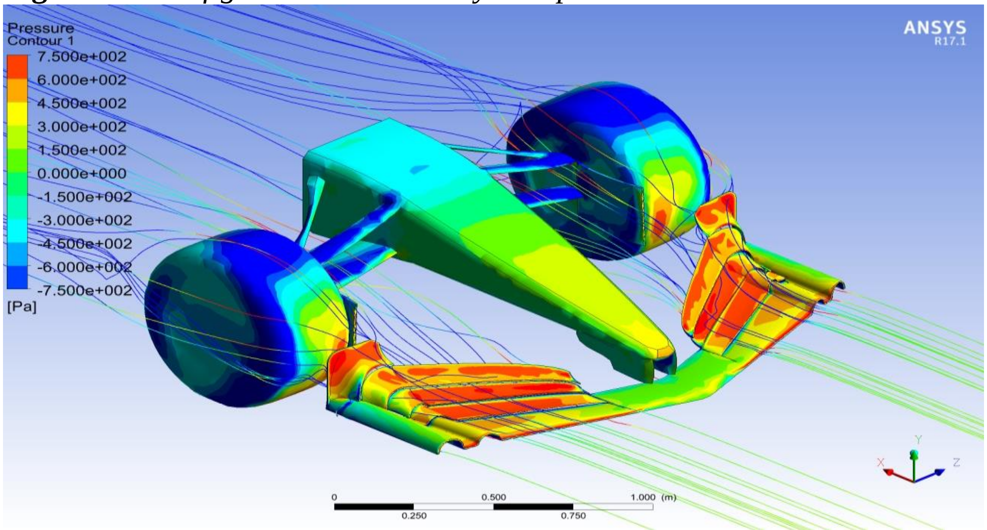
Figure 10. *Configuration 3 Geometry Computational Result*


Configuration 4 is derived from configuration 3 by removing the outer endplates on the front wing. These 2 large pieces on the front wing are easy to identify on the wing structure, and they are easily removed in DesignModeler using the *Face_Delete* command. In addition to these endplates there are also two small canards protruding from the surface - 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 and results in a mesh size of 2.2 million elements. After calculation, the lift force predicted is -1033.66 N and the drag predicted is 567.387 N. These numbers are very similar to configuration 3. There are no visible effects to the airflow when these large endplates are removed. It indicates that they don't need to be there in the first place and the wing structure would still perform equally well.

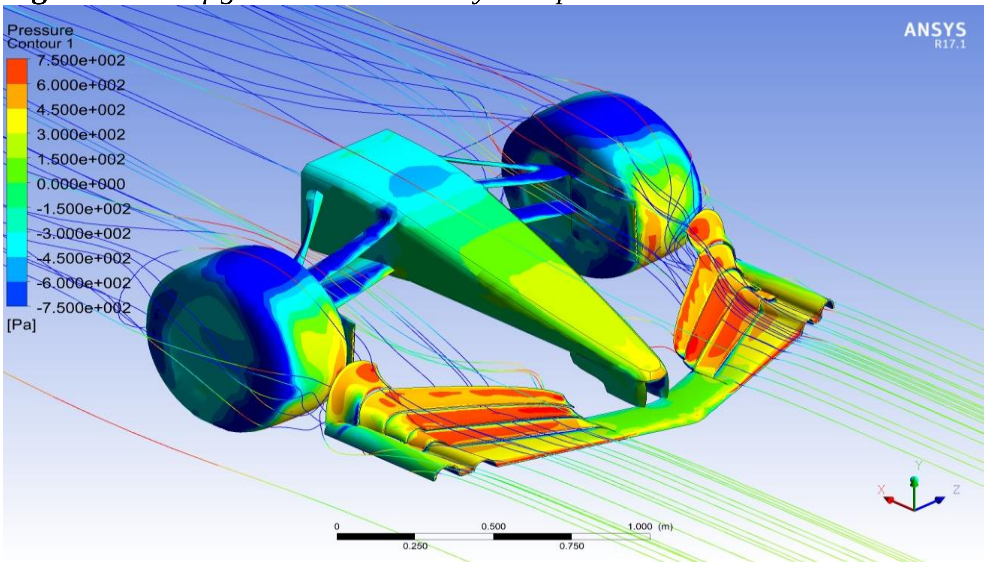
Figure 11. *Configuration 4 Geometry Computational Result*



Configuration 5 is reduced from configuration 4 by removing half of the top winglet so only the outer parts that shield the tires 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 lift force is -678.738 N and the predicted drag force is 449.183 N. Here the downforce is noticeably reduced when the horizontal slab of the winglets is removed. This is an indication that this is an important aerodynamic surface that provides much of the downforce so it need be there for the wing to work properly.

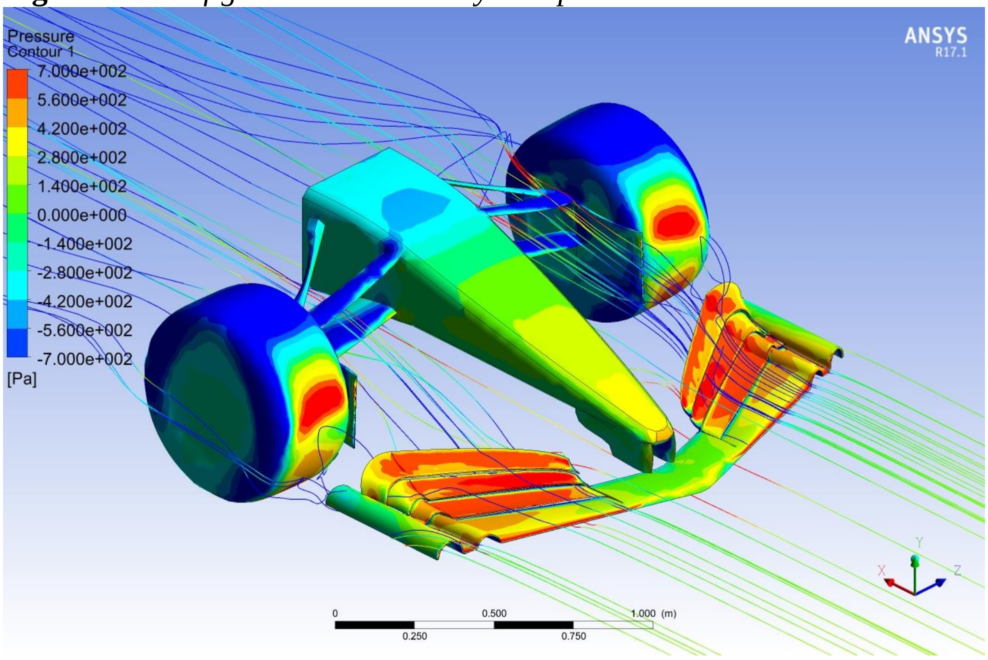
Figure 12. *Configuration 5 Geometry Computational Result*


Configuration 6 is a further reduction of configuration 5 by removing the small Gurney flap on the outer edge of the winglet (i.e. the square edges). This is the small vertical flap that's mounted on the trailing edge of the outboard winglet to further impede airflow. The same minimum mesh size of 4e-3 m is used, and the resultant mesh is 3.5 million elements. The predicted lift force is -599.302 N and the drag force is 445.815 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 create downforce. From Wikipedia, it explains that this is done by increasing pressure on the pressure side (upper), decreasing pressure on the suction side (lower), and helping the boundary layer flow stay attached all the way to the trailing edge on the suction side (lower) of the airfoil.

Figure 13. *Configuration 6 Geometry Computational Result*


Configuration 7 is the simplest of all geometries. It is obtained by removing the two outboard air deflectors in front of the tires so that the wing now appears with a lower profile. It is meshed with a cell size of 4e-3 m, and the resulting mesh has 3.6 million elements. The predicted lift force is only -453.828 N and the drag force is 439.577 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 on the front side of the tires. This creates a bluff body and is not very efficient because it’s not a streamline shape. The effect seen here should not be surprising and this case is presented simply to illustrate.

Figure 14. *Configuration 7 Geometry Computational Result*

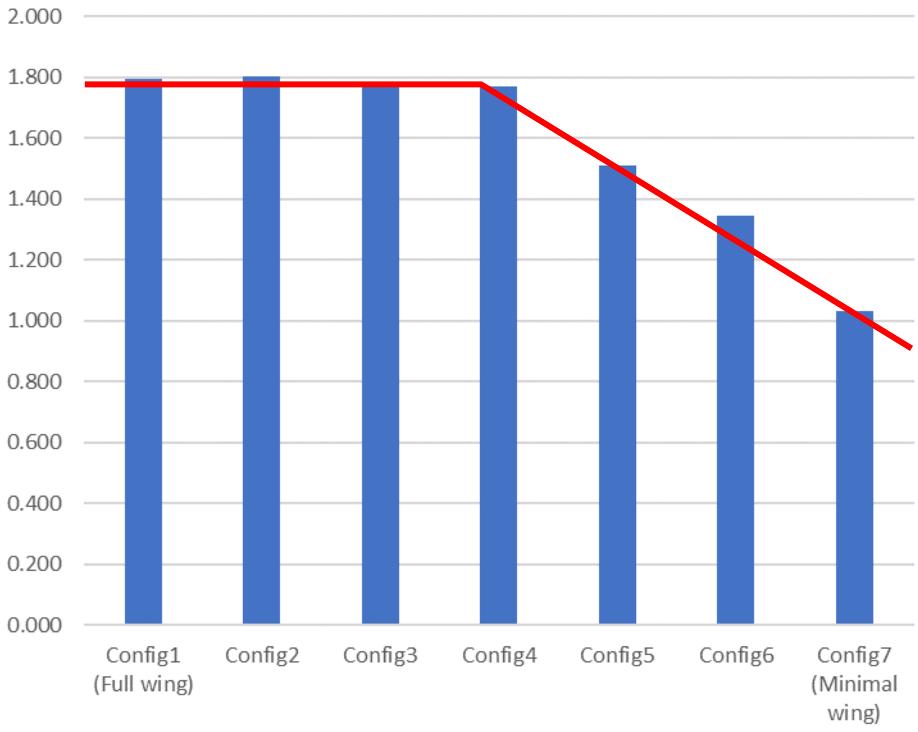


After computing all seven geometries using ANSYS Fluent, the aerodynamic efficiency (-L/D) for the different configurations are tabulated in Table 1 and plotted in Figure 15. A trend is clearly visible: as wing elements are removed and the front wing is simplified, airfoil efficiency decreases accordingly. This part of the result is not surprising; however, the way that it 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 additions of aerodynamic surface 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 (-L/D) is 1.8 at best. When compared to a NACA airfoil or to an actual aircraft, such as the aerodynamic efficient Boeing 777, the lift to drag ratio is 19. This begs the question: what is causing this big difference? If an aircraft can be designed with an L/D of 19, why can't a car be designed with an L/D of 19 also? The answer may lie in the fact that an airplane's wing area is much larger than a car's wing area. In other words, if an airplane uses a wing area that's depicted in these images, it will never fly.

Table 1. *Resultant Forces and Calculated Aerodynamic Efficiency of the Seven Wing Configurations*

Configuration	Lift Force (N)	Drag Force (N)	Aerodynamic Efficiency (-L/D)
1	-1015.19	565.831	1.794
2	-984.523	545.759	1.804
3	-1014.51	568.624	1.784
4	-1003.66	567.387	1.769
5	-678.738	449.183	1.511
6	-599.302	445.815	1.344
7	-453.828	439.577	1.032

Figure 15. *Aerodynamic Efficiency (lift/drag)*



Cloud Computing

So far, all seven cases presented in the previous section have been run on desktop computers in San Jose State University's E164 engineering lab. Each case took approximately 3 days to run, with the exception of the full wing case (configuration 1), which could not finish in one week in the lab. The job was moved to the cloud, running it on Amazon Web Services (AWS) to provide the hardware resources needed for computation. The software interface with Amazon Web Services was provided by Rescale Inc. Because the cloud provides a large network of computers, a large number of cores could be requested to run a job. In this case, 144 cores were requested from an Intel C4 Haswell computer cluster. The large number of cores made the computation much faster. However, these configurations had to be submitted as a batch job which is different from running them on a desktop computer (either in the foreground or background). As such, an ANSYS case file had to be created, along with a journal file which instructed the server machine on how to run the job. This part was not straightforward, but the time it saved from computation made it a worthwhile alternative. When it finished computing, an ANSYS data file was generated and downloaded. Because the data file size was big (in either a .dat or .dat.gz file format), it took upwards of 30 minutes to download from Rescale's website. Once the data file was downloaded, it was read into either Fluent or CFD-Post for post-processing. After the data was read, it followed the same procedure for retrieving resultant forces and calculating the airfoil efficiency. Although the time and effort to prepare a file to send over the internet to the cloud was somewhat involved, Rescale's website helped with moving files to and from the cloud to make computing in the cloud possible. Rescale also provide an ANSYS license if the user does not have his own.

The power of cloud computing has only been explored in this study. In the near future, high-speed internet would be ubiquitous and it will connect all the end-users to the server clusters around the world. No one will need to invest in hardware anymore as these become obsolete too quickly. Everyone can simply send their job to the cloud and get data back in a tenth of the time. From the time-saving realized by running these jobs on the cloud, it has enabled us to solve problems with a bigger mesh size, which was not possible on our lab machines before. Now with 144 cores and a high-performance computing (HPC) environment, our hardware capability is on par with that of a top-tier Formula One team.

There is one caveat: when organizations invest in hardware, all of the computing time afterward is free because they already own the hardware. However, when using a server-leasing service such as Rescale, the pricing scheme is pay-as-you-go, and it doesn't matter if the cloud gives you back good data or not. For example, one of the configurations presented here was sent to both a Haswell and an Obsidian computer cluster; while it took just over 3 hours on Haswell, it took more than 20 hours on Obsidian, which is an inefficient use of the CPU. Someone more versed in remote solving would be needed to make sure this high-performance computing task is done right the first time. As a lesson learned, care must be taken when sending a remote job to the cloud so the time on leased CPU is well utilized; otherwise it can get very expensive.

Conclusions

In this paper the importance of creating a computational geometry and have it successfully mesh is explained: without a mesh, the geometry cannot be analyzed. Two computer-aided design tools have been described, and their use illustrated to solve the common problems associated with preparing a geometry for computation. An example of a racecar front wing has been used throughout this paper to show the various steps for taking a 3D CAD geometry all the way to meshing and solve. CFD results for this computational geometry has also been tabulated. Finally, variations of the front wing geometry point to one intermediate configuration as the most efficient aerodynamic shape.

References

Alliez, P., Cohen-Steiner, D., Yvinec, M., and Desbrun, M. 2005. Variational tetrahedral meshing. *ACM Transactions on Graphics* 24 (2005), 617-625.

ANSYS, Inc. 2014. Introduction to ANSYS DesignModeler, 15.0 Release. *Training Manual, 1st Edition* (Feb. 2014), Inventory #000570.

ANSYS, Inc. 2014. Introduction to ANSYS Meshing, 15.0 Release. *Training Manual, 1st Edition* (Mar. 2014), Inventory #000576.

ANSYS, Inc. 2014. Introduction to ANSYS Fluent, 15.0 Release. *Training Manual, 1st Edition* (Mar. 2014), Inventory #000575.

Blelloch, G. E, Miller, G. L., and Talmor D. 1996. Developing a practical projection-based parallel Delaunay algorithm. *Proceedings of the Twelfth Annual Symposium on Computational Geometry* (May 1996), 186-195.

Dickerson, M. T., Keil, J. M., and Montague, M. H. 1997. A large subgraph of the minimum weight triangulation. *Discrete & Computational Geometry* 18 (Oct. 1997), 289-304.

GoEngineer, Dassault Systèmes SolidWorks Corporation. 2013. SolidWorks Essentials, *SolidWorks 2013 Training*, Part #PMT1300-ENG.

Su, P. and Drysdale, R. S. 1995. A comparison of sequential Delaunay triangulation algorithms. *Proceedings of the Eleventh Annual Symposium on Computational Geometry* (Jun. 1995), 61-70.

Wade, A. 2014. 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/*.

Wikipedia. 2018. *https://en.wikipedia.org/wiki/Gurney_flap*.

Wikipedia. 2018. *https://en.wikipedia.org/wiki/Lift-to-drag_ratio*.