

# 2024 Aerodynamics DDR

Ryan Hopke, Hayden Bell, Bryce Fothergill, Dennis Yermakov, Jennifer Brisno, John Mcalister, Simon Nguyen

## Introduction

Aerodynamic systems are projected to increase the performance of the 2023 specification DFR competition vehicle by 5% in skid pad scenarios according to lap simulation done with an underdeveloped aerodynamics package. A similar percent increase can be extrapolated when analyzing autocross performance with the added lift and drag coefficients. This combined analysis has become the basis for the continued development of the aero package for the 2024 season, as all lap simulations were run with a low performing design. This suggests that if properly developed, the performance potential of the vehicle vastly exceeds the projected 5% increase. While an in depth discussion about the lap simulation analysis will exceed the scope of this report, the results of both one DOF (degree of freedom) and three DOF models are documented in the 06\_VehicleDynamics BOX drive folder, and can also be accessed with this shared link: <https://utdallas.box.com/s/kjqb9wulo7h8wgf7v0b2oakgqigkumo5>. The scope of this report will be limited to the development of the upgraded 2023 aerodynamics package as it currently sits in the 2024 design pipeline as of the 24<sup>th</sup> of July 2023.

## Abstract:

Developing a successful aerodynamic system for a racing car requires more than just an understanding of aerodynamic principles. It requires in-depth knowledge of how that system and the race car will interact. Integral to this understanding is an aerodynamics map, or a series of simulations that position the vehicle and its aerodynamic system in dynamic cases that it will see on track. These simulations allow engineers to make decisions about how to increase the effectiveness of the aero package in cases such as braking, accelerating, crosswind, and cornering.

## I. Development Overview

The overall development plan for 2024 consists of working off the developed 2023 package and creating a sizeable “upgrade” to the existing platform. This was done for two reasons: to ensure the scope of the manufacturing task doesn’t exceed the team’s current capabilities, and to establish a logical path of development to correlate simulation results with. The specific development plan for the CFD analysis was bolstered by large aero map experiments that were ran intermittently between large design iterations. These aero maps consisted of cases that allowed for performance analysis of the system in various ride height cases, yaw angle cases, cornering cases, steering angle cases, and vehicle roll cases. To make the process streamlined and allow the development process to move swiftly, parts were developed exclusively in straight-line cases for a week. Over the course of the following week, parts would be consolidated into a master CAD model for that given week, and a ride height study was run. This block of alternating weeks ran until 3 consecutive ride height maps were run before the focus would

switch to yaw angle analysis. At the time of writing this report, we have finished our ride height studies and are preparing to switch to yaw angle development. This report will outline the results of the past 3 ride height studies and the beginning of the transition to yaw and cornering analysis.

## II. Aero Map Boundary Conditions

All simulations are run as turbulent solutions using the Reynolds Averaged Navier Stokes (RANS) solver in tandem with the  $k-\omega$  SST turbulence model to provide insight into wall interaction. Mesh boundary conditions specify twenty “prism layer” (wall interaction) cells assigned to a growth rate of 1.2. Mesh performance has been optimized for the summer development period, and Figure 1 outlines the wall  $y^+$  performance around the geometry. Most of the CAD will exist in the viscous sub-layer threshold for wall  $y^+$  values, except for tires. This experimental inaccuracy does not exist past the log-law region, and therefore should not skew any final values, however this anomaly will be noted if there are correlation issues with the CFD in experimental testing.

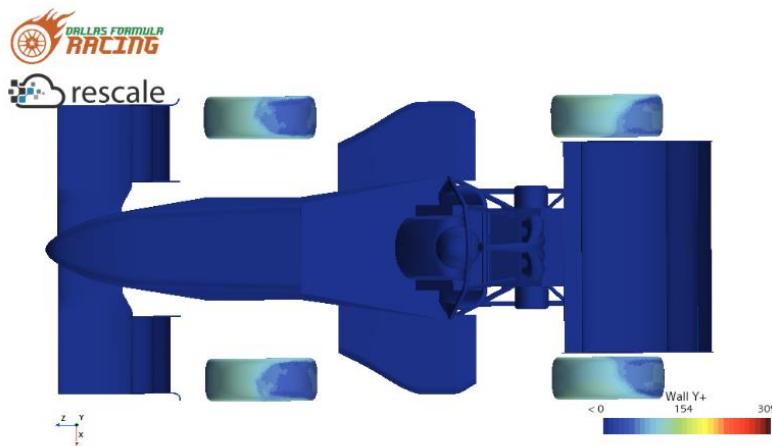


Fig. 1. Wall  $y^+$  scalar to display issues with the tire regions existing close to the log-law region.

The optimization of the first aero map mesh was conducted through an independence study and had to satisfy several criteria. First, it had to work universally with future components such as added beam wings, biplanes, and front wing flaps. Second, it had to be applicable to multiple experiment cases, such as differing ride height positions, crosswind angles, and cornering cases. Finally, the mesh must be economical. The on-demand high performance computing (HPC) power provided to us through Rescale is a luxury we paid for, albeit at a large discount. Regardless, having an efficient mesh allows us to scale our analysis and fit more simulations within our budget and increase the fidelity of our analysis.

Figure 2 shows the first iteration of the updated mesh. While robust and easily satisfied the first two goals, it was hardly economical. Boasting close to 25 million cells for a half-car simulation, diagnosing the mesh locally was close to impossible due to RAM restrictions, and sim output time for a full vehicle sweep was close to 10 hours on 96 HPC-6A and C5-MAX cores on the Rescale system.

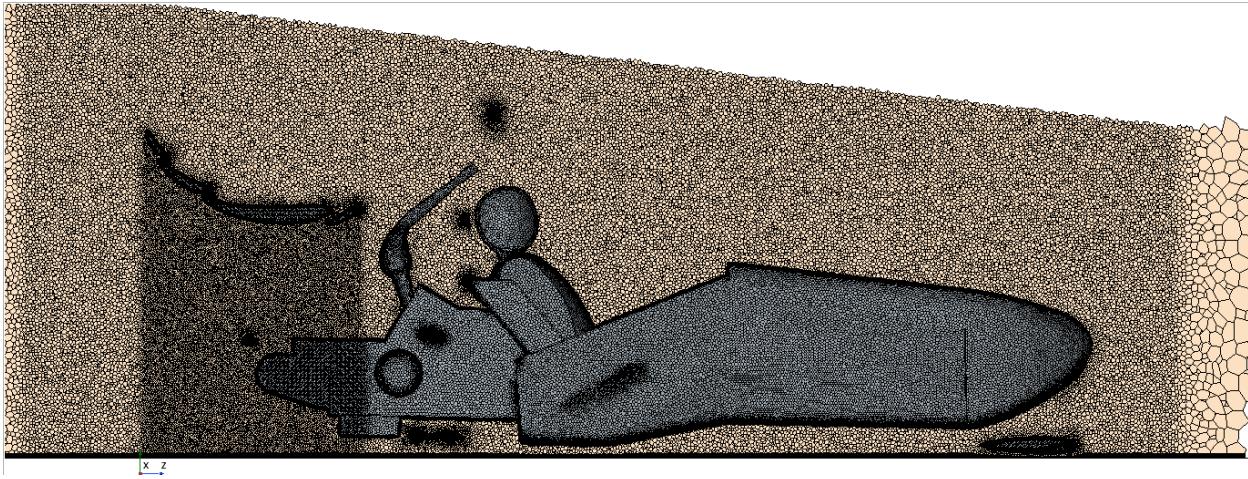


Fig. 2. A cut-plot shows an iteration of the first aero map mesh.

A large wake refinement box with further refinements below the front and rear wings made up the bulk of the cell count and were the regions that were targeted heavily in the second version of the map. Figure 3 displays the visual difference between the original mesh, and the mesh that would appear in all three aero map templates (ride height, yaw angle, cornering).

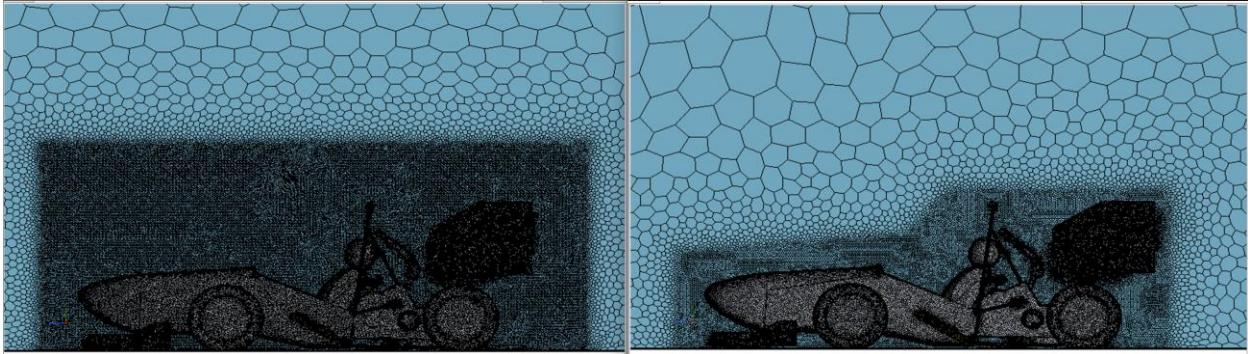


Fig. 3. The baseline/first aero map mesh (left), and the result of the mesh independence study (right).

The final iteration of the second version of the aero map mesh boasted a 26% decrease in cells, and a maximum report value percent difference of no higher than 2.2%. This difference can possibly be attributed to the reduction of mesh fidelity on the pressure side of the wings, and directly being the rear wing where separated flow can propagate, however more analysis on this discrepancy is needed. However, this percent error was traded for greater efficiency in map performance, as ride height simulation time was reduced by roughly 40%. Furthermore, the mesh's flexibility was demonstrated when it was applied to yaw angle and cornering cases and demonstrated convergence on the first attempted calculation. Lastly, the mesh proved to be extremely robust when aero members began to develop their assigned parts, and very few meetings were spent diagnosing mesh issues during the first six weeks of summer design. To view the complete development log of this mesh, refer to this file path in BOX drive:

<https://utdallas.box.com/s/v0dz0gp4b11rqkrsekwygja8gsdptf6u>

### III. Aero Map Setup

#### Ride height map:

The ride height map is a collection of straight-line half-car simulations (assume vehicle symmetry) with parameters designed to move the sprung mass of the vehicle into different ride height positions. The set of angles is set to move through six increments in a range that is limited to one inch up and down from the location of ride height measurement. This is the maximum amount of travel allocated by the suspension, and therefore the limits of rotation around the vehicles pitch center adhere to this principle (the pitch center of the vehicle is at the center of gravity due to the vehicle having infinite instant centers. This is due to a lack of anti-dive geometry built into the design). The extremes of the vehicle can be characterized by terms such as “squat” (most amount of front ride height), and “dive” (least amount of front ride height). Each simulation is run at 18 meters/second, or approximately 40 miles per hour. This number was derived from previous one-dimensional lap simulations where a mean speed was calculated, however this condition is not entirely representative of all track cases and only needs to serve as a realistic baseline value from which further simulation will be developed from. Rolling wheels and a moving ground were included to complete the experiment.

#### Yaw angle map:

The setup for the yaw angle map requires a full CAD model of the car due to the non-symmetrical nature of the flow. For ease of post processing, the domain was rotated around the CAD model and multiple views were established. The domain was rotated around a central rotation point, and the boundary conditions from the ride height simulation were expanded upon for the yaw angle simulation. Tire rotation and ground movement were assigned with respect to the vehicles mean flow. A sweep of angles was run from zero degrees to fifteen degrees off axis from the vehicle’s centerline. The max of this range will represent the worst-case crosswind the vehicle will run in, as slow ground speed with high wind speed tend are few cases experienced on track. Boundary conditions for reports such as the overturning moment and yaw moment were measured around the vehicles roll center and center of gravity respectively.

#### Cornering map:

Building off the full-car yaw angle map, the cornering map is similar yet differs greatly in the solution domain needed to complete the analysis. Known as a “circular domain”, a geometric shape is created in CAD that tracks the car through the corner yet leaves ample room in front and behind the car for the solution to propagate. The domain size is effectively the same as both the ride height and yaw angle simulations, but the measurement of room in front and behind the vehicle CAD is measured in radially instead of linearly. The boundary conditions for vehicle movement are specified through the 2023 competition car’s hypothetical performance through a standard FSAE sized 15.25-meter skid pad(diameter). A critical assumption made is the wheels on the outside track are rotating at the same speed as the inside track. These boundary conditions will be constant throughout the development period as updated 3DOF lap simulations are needed

to update the boundary conditions with relevant data as theoretical aerodynamic performance increases. The parameters that are measured within the cornering sim are the steering angle of the vehicle and the roll angle of the chassis. The steering of the vehicle is accurate to the Ackman created by the steering geometry, which are characterized by 4<sup>th</sup> order polynomials. The exact derivation for the equations cannot be provided due to misplaced files in the BOX drive. The roll of the sprung mass of the vehicle has been derived using the vehicle spec sheet. A coordinate system was positioned in a way such the angle of the Z axis would be aligned to intersect the front and rear roll center heights while keeping the X axis orthogonal to the ZY plane. Much like before, the limits of rotation for the roll of the vehicle correspond to the one inch travel limit of the suspension.

The parameterization of the maps was done using the built in STAR-CCM+ transform functions and parameter creation. These values can be referenced directly through the Java API, and batch scripts can be coded to allow for the automation of the aero map process on an HPC system.

#### IV. Job Scheduling and Post Processing

Refer to the attached document “Process to Run and Post Process Aerodynamics Simulations” for an in-depth look at the creation of the macros, the architecture chosen for the HPC platform, and finally the scripts written to streamline the post processing process.

#### V. Aero Map Results

At the time of writing this report, the development of the 2024 upgrade package has not been complete. This section will include reports and data up to 07/24/23. This is the final ride height driven vehicle iteration that will take place before in-depth yaw analysis and yaw-based iterating begins. Flow control elements such as a beam wing and bullhorn elements are missing from this analysis, and thus there still exists a large discrepancy in front and rear axle loads.

Ride height sensitivities between design iterations were calculated and post processed in multiple different ways. The most common method of monitoring design performance was through a simple bar chart. Due to each map running a different number of simulations, a deeper statistical analysis could not be performed, however each sim did successfully run a minimum position, a maximum position, and greater than 4 simulations between the two extremes. Any extra simulations ran would “pool” together in one of 6 locations along the ride height sweep, making statistical variance for any cases that ran above the minimum 6 simulation requirement negligible, and would remain comparable to previous cases. Figure 4 is the bar chart comparison of the baseline through the third ride height stint, and shows the progress made as development continued throughout this stage.

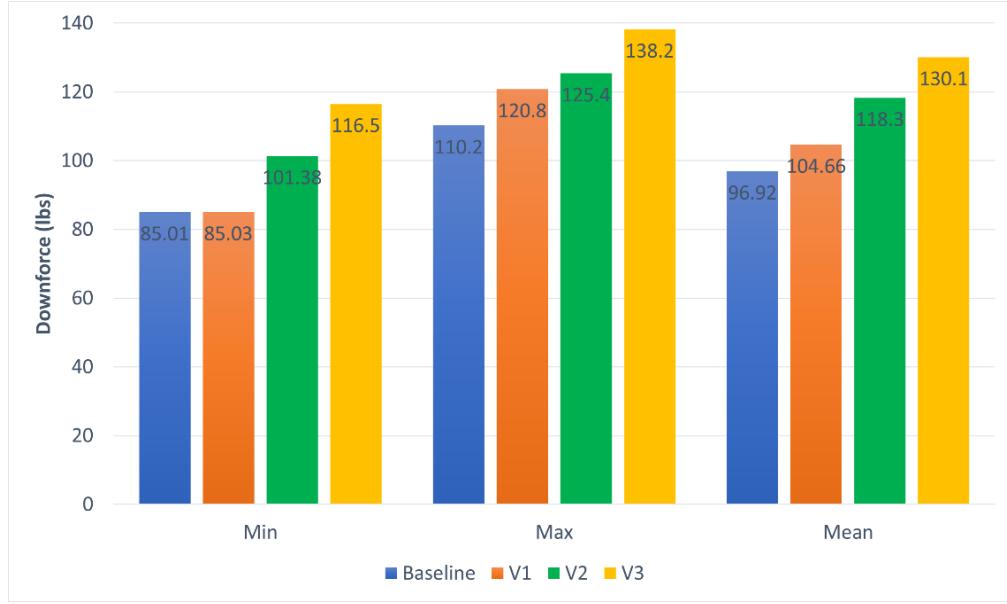


Fig. 4. The complete bar chart that shows the progression of designs and their ride height sensitivities throughout the development stint

More acute analysis can be done on any individual case from the general bar chart analysis. Graphing the most recent design (V3) on ride height interactions, a familiar trend begins to emerge when we observe specific instances of aerodynamic inconsistencies in front axle performance. Figures 5 outlines the results of study 3. The graphs display the relationship of front and rear axle downforce in respect to front ride height.

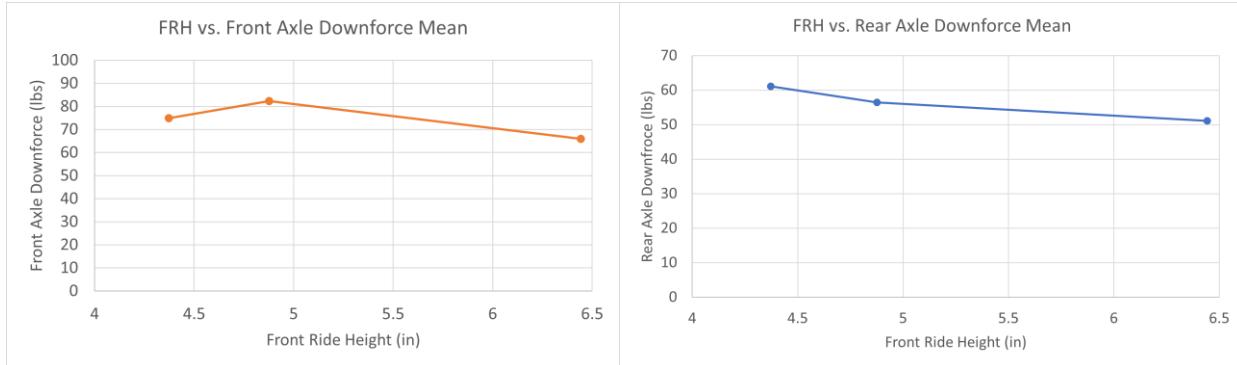


Fig. 5. Front and rear axle load graphs vs. front ride height.

Figure 5 shows the clear relationship between front and rear loads under both extreme cases of squat and dive. The left plot in figure 5 shows that as the front ride height decreases and the car dives, the system loses downforce. The front axle also loses downforce at max squat. This indicates a ride height sensitivity, and therefore a strong indication of ground proximity changing the performance of the front wing as the distance from the suction side of the wing changes distance from the ground. Figures 6, 7, and 8 corroborate this theory by showing the front wing in max squat, max dive, and max downforce respectively. Note the changes in the wall shear stress vector plots.



rescale



Fig. 6. Positioned furthest from the ground. Note the hard separation lines on all the supporting elements after the mainplane.



rescale

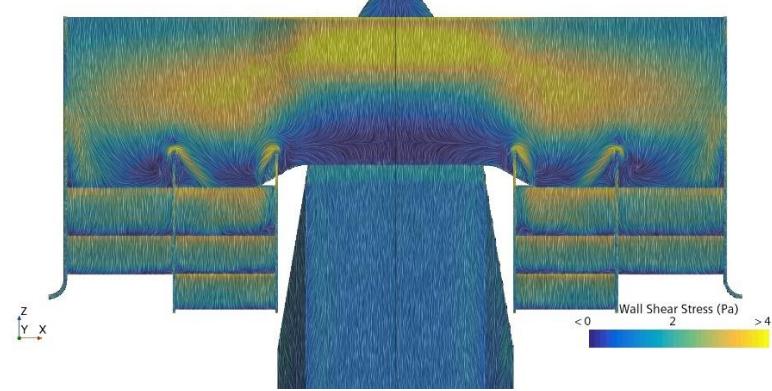


Fig. 7. Positioned closest to the ground. Note the intense mainplane separation.

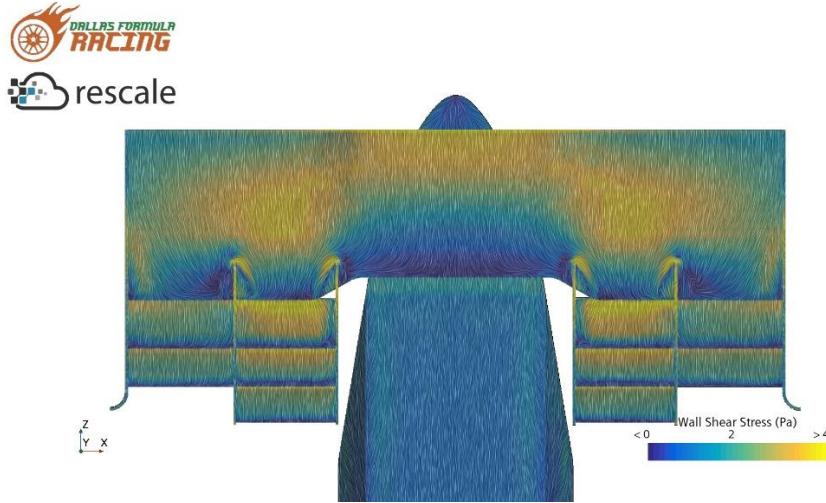


Fig. 8. Ideal ride height scenario.

Having access to this type of analysis allows us to gauge the potential sensitivity of the aerodynamic system, and to monitor design changes throughout the development phase. By checking our progress in this way, we could be confident that that the system we designed would be capable in difficult angle of attack cases.

The end of this design stint has provided us with a system that improves total downforce by 25%, increases the minimum downforce by 37%, and decreases the disparity between potential downforce extremes by 34%. This reduced sensitivity can be attributed to added front and rear wing flaps, and a general increase in endplate size. A greater separation between pressure and suction sides of both the front and rear wing is potentially responsible for the decreases in sensitivity in this analysis. However, an increased side profile has been observed to dramatically increase yaw sensitivity of the package, so a reduction of endplate surface area is a projected change. This will effectively sacrifice ideal ride height performance and total downforce in exchange for reduced yaw and cornering sensitivity. This tradeoff is acceptable due to the primary performance beneficiaries an aero system provides are cornering applications and high yaw angle cases.

## VI. Component Development Documents

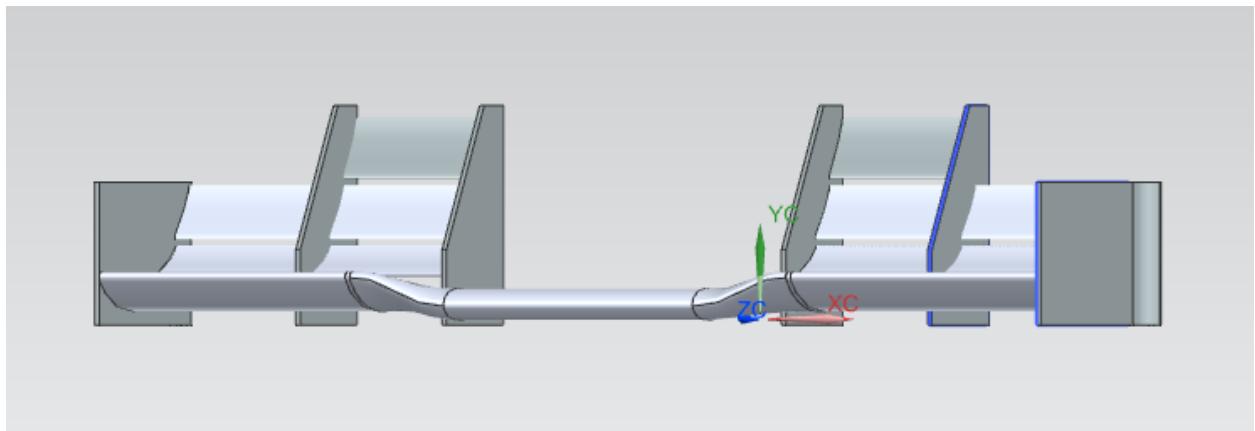
The following documents are a collection of development papers from the team that has developed the aero package to this current state. Their simulation boundary conditions and mesh setup remain consistent across each design, and their simulations are all ran at 18 m/s at the static ride height. Their primary performance metrics are the ClA and the raw downforce mean. The ride height map performance has shown no increases in sensitivity as the development period progresses, and thus no alternate boundary conditions had to be observed or developed around to remedy the potential issue. The documents will follow the order from front to back of the vehicle: front wing, bullhorns, sidepod, rear wing, beam wing. Due to the ongoing nature of the analysis, no standardized format has been required of the team.

# 2024 Front Wing Report

By: Jennifer Briseno

**Design Process:**

The iterative design process of the 2024 front wing I created consists of 5 steps. The first was a change of AOA (Angle of Attack) on the main plane of the original front wing, however, this has been scrapped for 2024 but documented for extra informative analysis of a new AOA on the main plane. The second step was changing the length of the innermost plate (closest to the nose cone) and a slope change on the outer end plate. Next, I added a third flap and a middle plate to the front wing, thus creating a cascade. Afterward, for the fourth step, I decided to elongate the third flap and middle plate to where their placement was in the center of the 1st and 2nd flap of the wing. Lastly, in step 5 the outer end plate was scrapped from the previous steps and adjusted to have a simple rectangular shape.



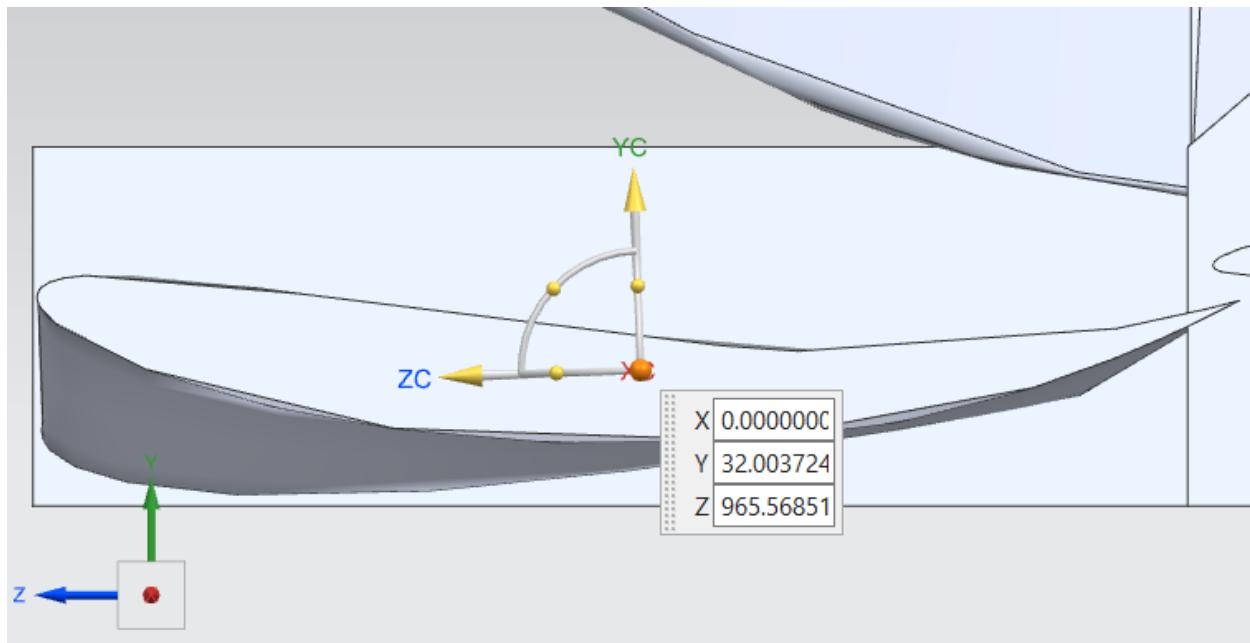
Figures 1 & 2: Finished Front Wing as of Design Stint 3

### **Progression Analysis:**

#### **Change #1: Main Plane - New AOA**

By increasing the AOA of the main plane, it was hoped that it would help direct more airflow to the 1st and 2nd flaps to increase downforce. This was achieved after the first simulation. There was 6.75 lbf increase in front axle downforce and a 4.18 lbf increase in front wing downforce. However, this was scrapped to simplify mounting.

- Original angle of attack: 0.000
  - (0.000, 33.004, 967.569)
- New angle of attack: 2.000
  - (0.000, 32.004, 965.569)



*Figure 3: New AOA View*

Jennifer Front Wing v1 - New Main Plane AOA											
CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downforce (lbf)	Raw Drag (lbf)	Radiator dP (Pa)	Front RH (in)	Rear RH (in)
1.27		54.20	50.40	44.48			104.80				

*Table 1: Main Plane AOA Data*

### Change #2: Inner Plate - Length Increase

The length of the inner plate was adjusted to 273.1244 mm, which would mean the plate has a larger amount of its area placed on top of the main plane. This was done in order to enclose more air in the front wing system instead of airflow being lost by slipping underneath the nosecone. Additionally, having the slope of the end plate begin sooner (at the end of 274.3488 mm) also helped the amount of airflow within the system. The enclosure of more air increased downforce in terms of the front wing and raw.

Jennifer Front Wing v2 - Inner and Outer Front Plate Length Increase												
CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downfoce (lbf)	Raw Drag (lbf)	Radiator dP (Pa)	Front RH (in)	Rear RH (in)	
1.26	0.61	50.56	53.17	40.89	54.02	7.91	103.33	50.10	188.22	4.88	5.55	

Table 2: Inner & Outer Front Plate Length Increase Data

### Change #3: New Cascade - Additional 3rd Flap & Middle plate

Creating a cascade on the front wing by adding a 3rd flap and a middle plate to the system was for the purpose of a dramatic increase of downforce on the front wing alone and overall. Not only that, but the cascade also helped direct airflow in a more direct path.

Jennifer Front Wing v3 - Inner Cascade												
CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downfoce (lbf)	Raw Drag (lbf)	Radiator dP (Pa)	Front RH (in)	Rear RH (in)	
1.38	6.26	70.73	42.48	54.50	52.11	4.45	112.76	51.31	118.84	4.88	5.55	

Table 3: Cascade Creation Data

### Change #4: Cascade Adjustment - Middle Placement

The placement of the cascade was pushed more towards the center in order to increase front wing downforce. More airflow would be in contact with more surface area to then have more flow directed to the rest of the aero package on the car.

Jennifer Front Wing v4 - Inner Cascade Length Increase												
CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downfoce (lbf)	Raw Drag (lbf)	Radiator dP (Pa)	Front RH (in)	Rear RH (in)	
1.34	6.05	73.32	41.51	56.00	52.57	4.46	114.38	49.58	98.91	4.88	5.55	

Table 4: New Cascade Placement Data

**Change #5: New End Plate - Rectangular Shape**

Lastly, to continue following the want for more downforce, changing the end plate to a full rectangle instead of a slanted cut off allows for this. Like the previous changes, this change is enclosing more air in the system. The results for this sim were not done stand alone, instead they were combined with the rear wings last updated iteration.

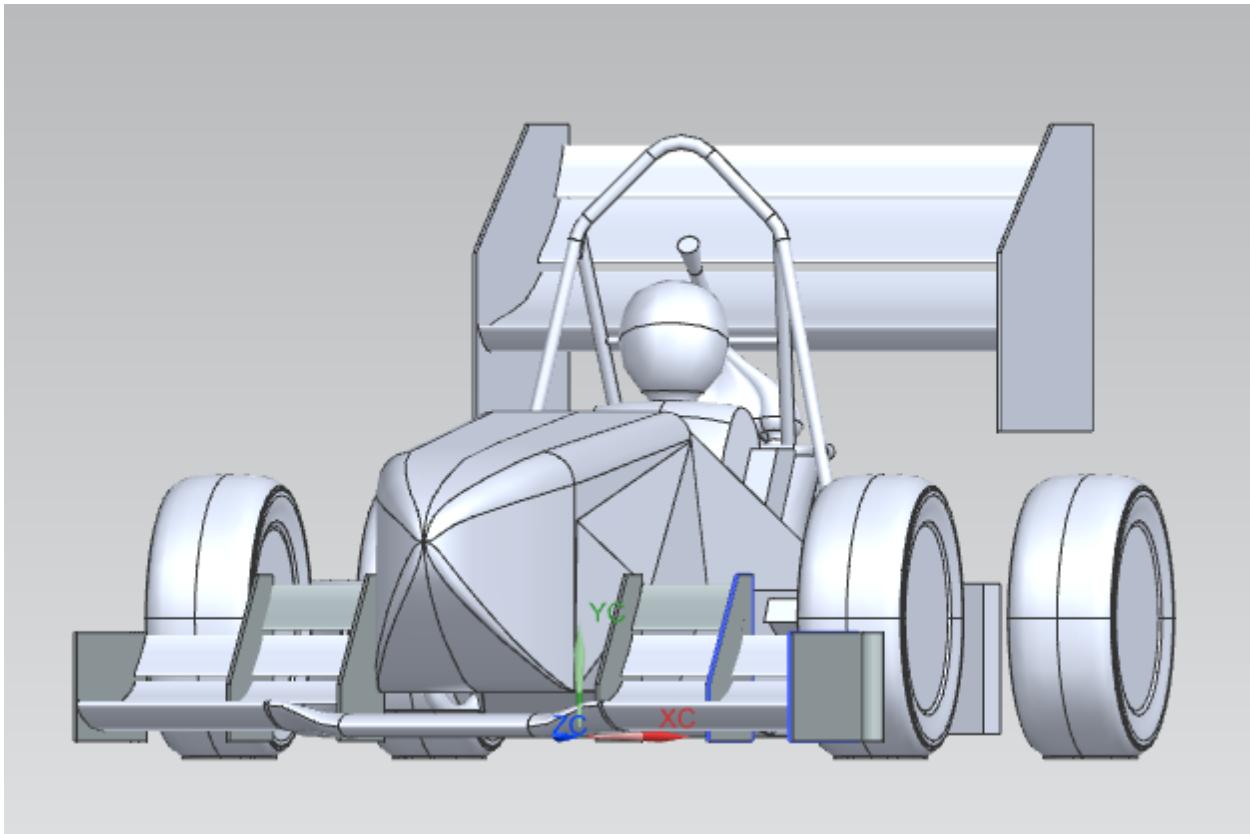
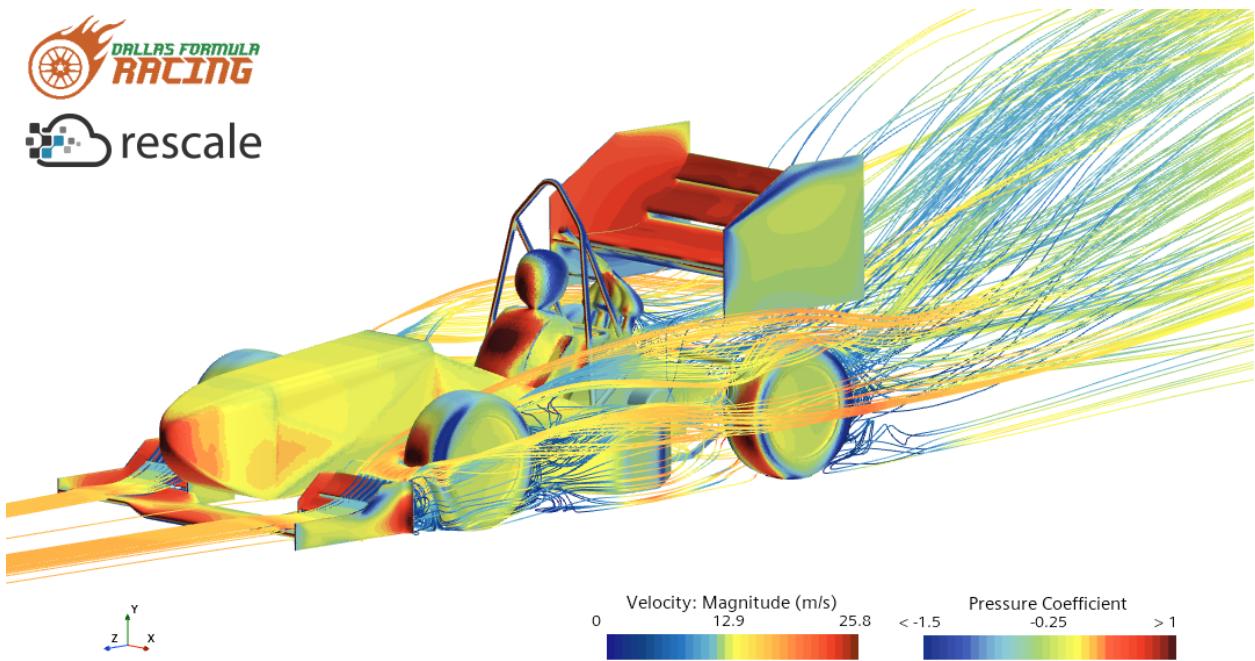
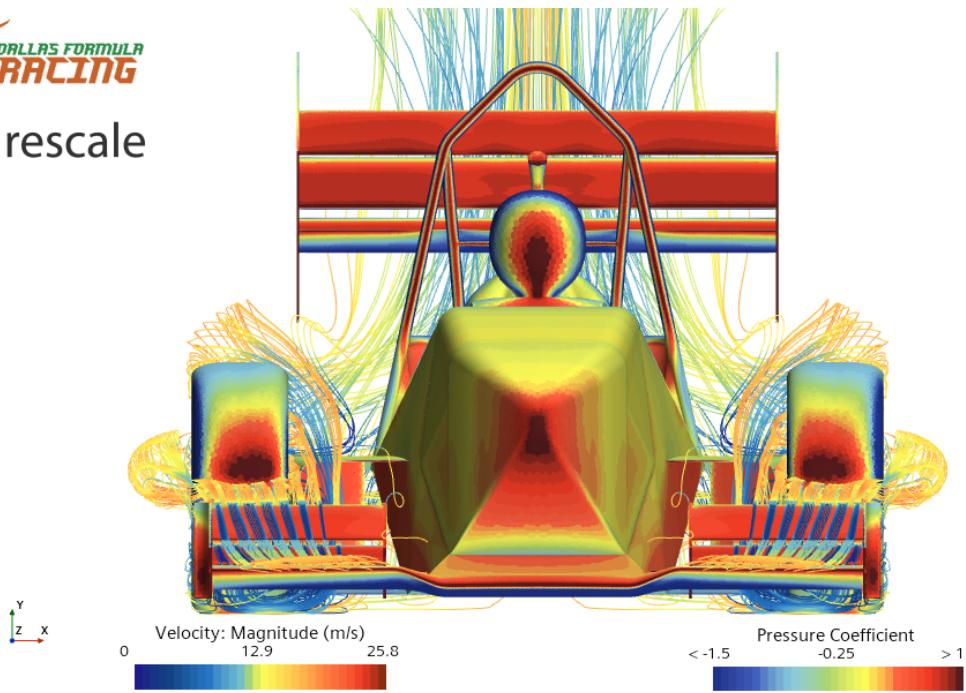
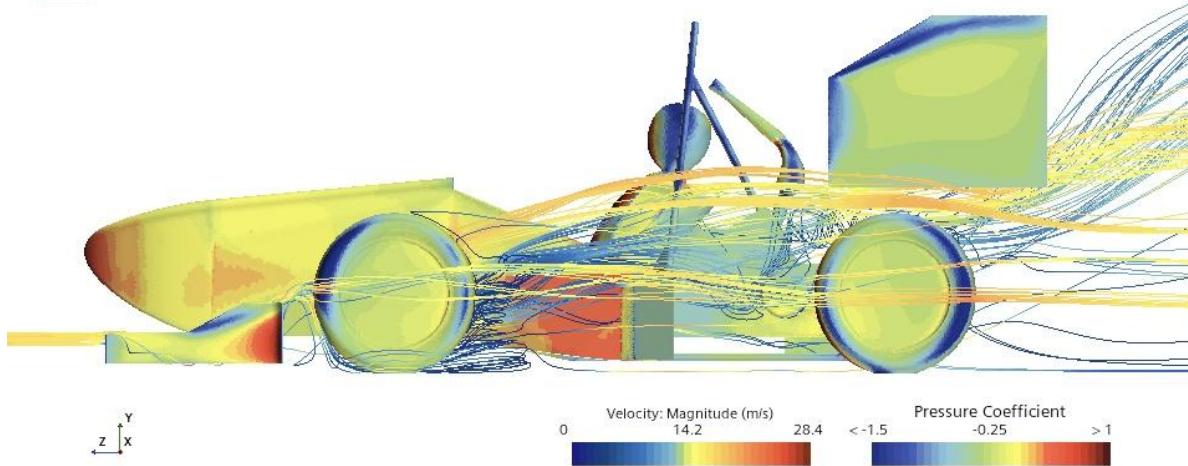
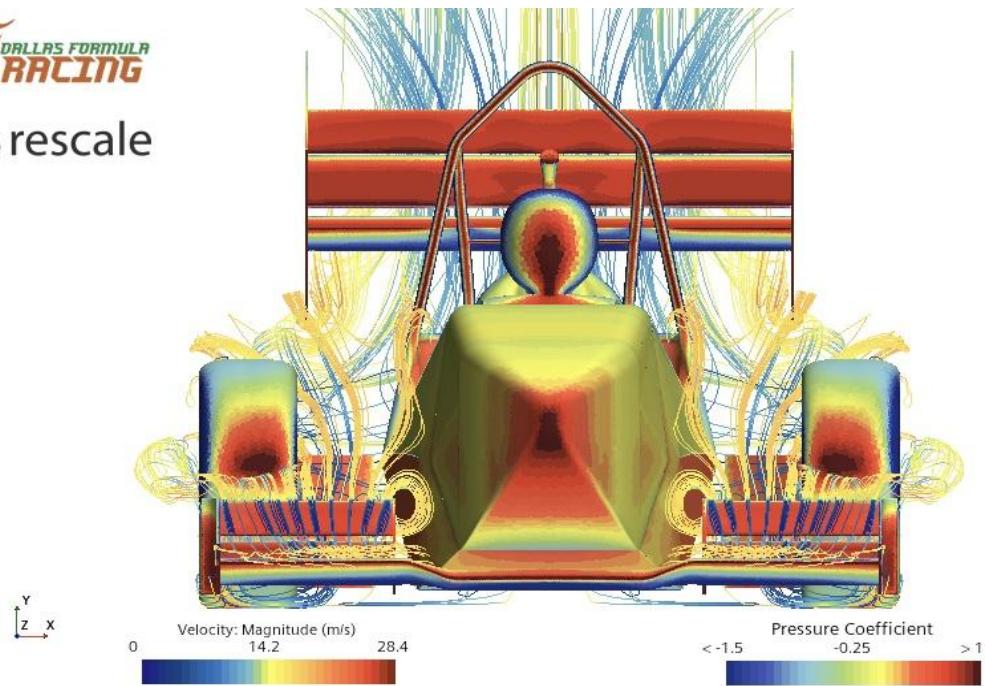
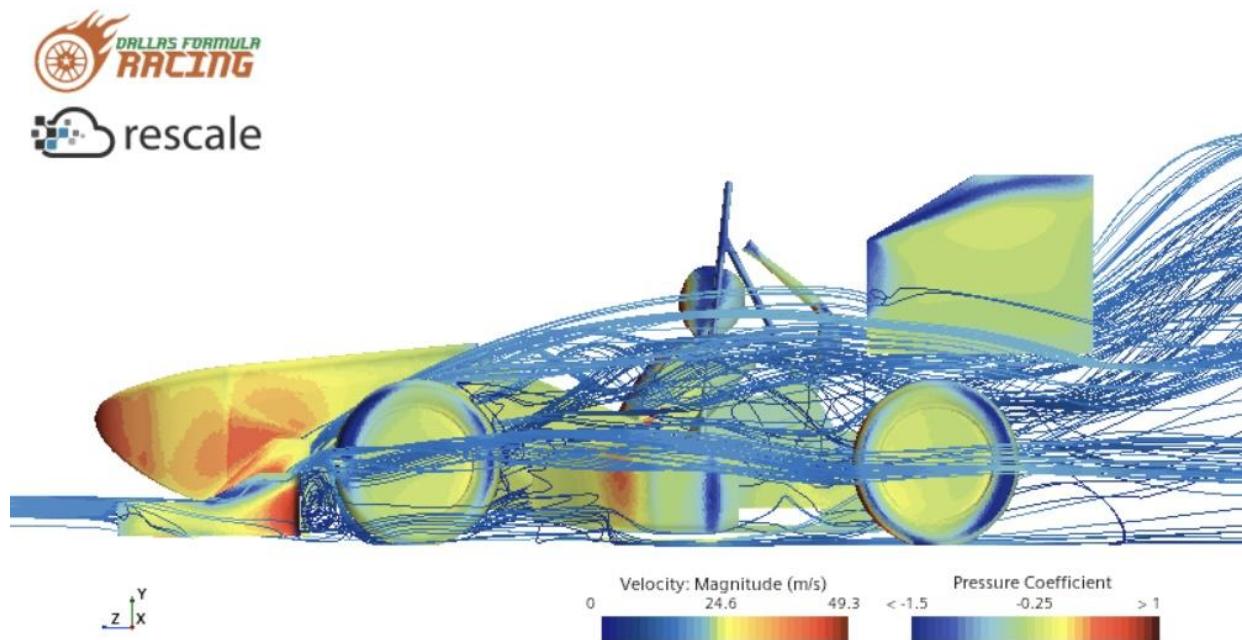
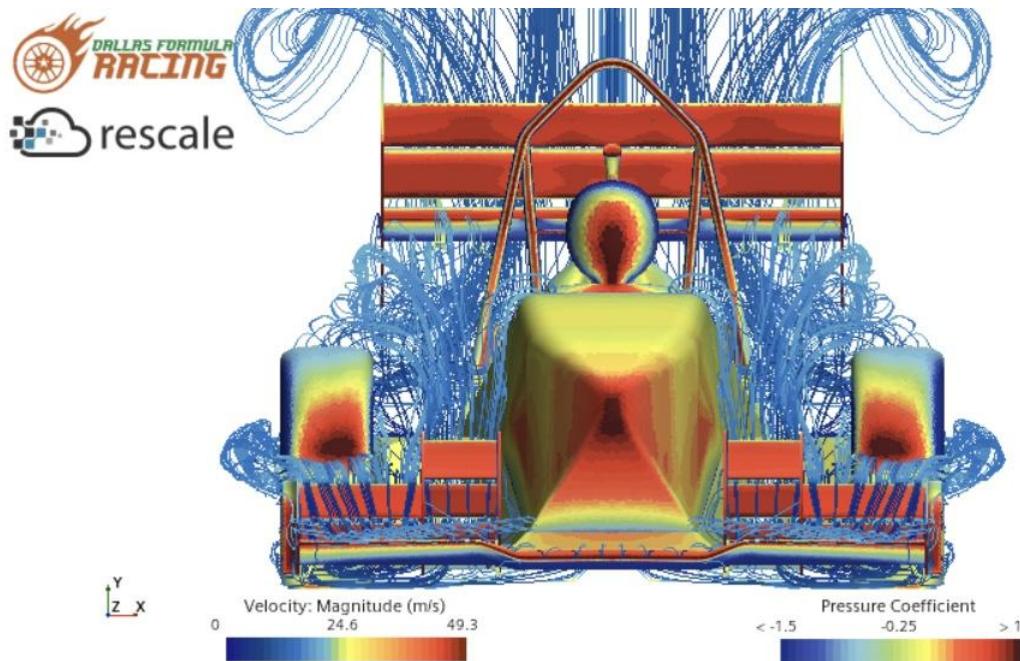


Figure 4: Final Front Wing on Full Car

**Streamlines (w/o final iteration):**





# Bullhorns Ver.1

## Data Results for Ver.1:

Raw Drag	Raw Downforce	Rear Wing Downforce	Front Wing Downforce	Rear Axle Downforce	Front Axle Downforce	Front Wheel Axle Moment	Rear Wheel Axle Moment
59.95885780 4652624	134.6128718 1676056	70.99468819 58776	57.23001772 1896964	73.20161	61.9422	-4026.24271 37950417	4758.104535 131717

## Streamline Data for Ver.1:

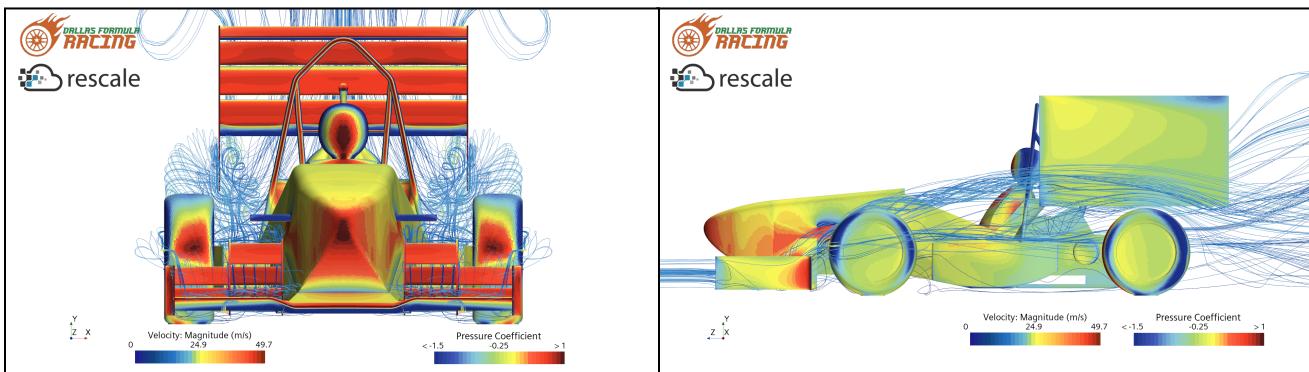


Fig P1-1: Depicts the front view streamlines

Fig P1-2: Depicts the side view streamlines

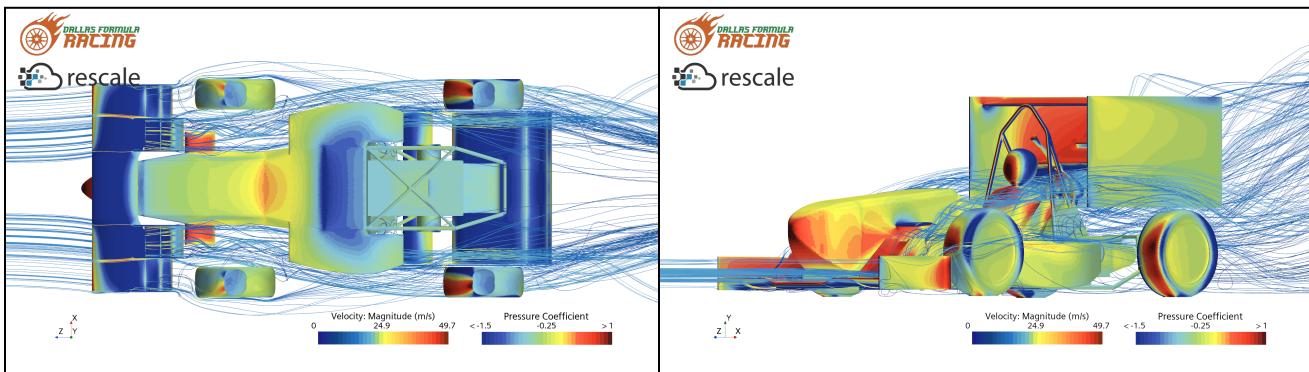


Fig P1-3: Depicts the bottom view streamlines

Fig P1-4: Depicts the 3/4 view streamlines

# Bullhorns Ver.2

## Data Results for Ver.2:

Raw Drag	Raw Downforce	Rear Wing Downforce	Front Wing Downforce	Rear Axle Downforce	Front Axle Downforce	Front Wheel Axle Moment	Rear Wheel Axle Moment
58.01410161 40345	127.0033675 9871627	71.69063774 236882	53.26742776 597585	62.48784	65.01471	-4225.95643 6004012	4061.709721 3535194

## Streamline Data for Ver.2:

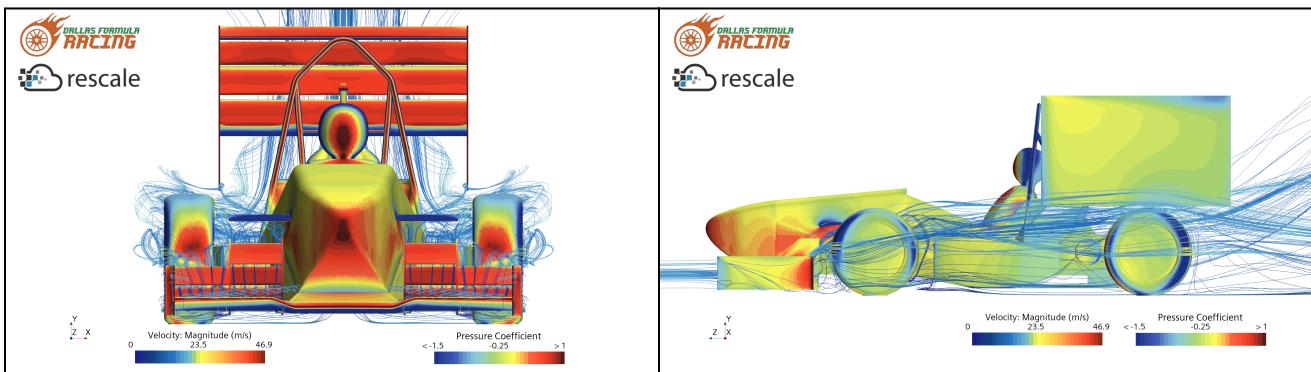


Fig P2-1: Depicts the front view streamlines

Fig P2-2: Depicts the side view streamlines

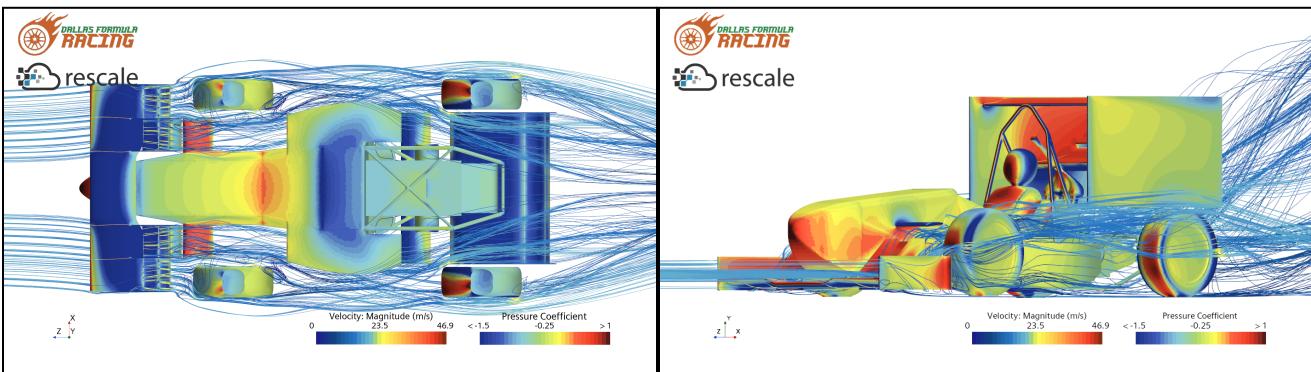


Fig P2-3: Depicts the bottom view streamlines

Fig P2-4: Depicts the 3/4 view streamlines

# Compiled Data (Previous Iteration+Current)

## Data Results for Bullhorn Position #2:

Raw Drag	Raw Downforce	Rear Wing Downforce	Front Wing Downforce	Rear Axle Downforce	Front Axle Downforce	Front Wheel Axle Moment	Rear Wheel Axle Moment
50.26370554 08319	105.6923574 50113	-56.0087084 070243	-50.0921077 571435	50.08324318 86385	56.02190142 12509	-3255.41080 72615	3641.423592 38131

## Data Results for Ver.1:

Raw Drag	Raw Downforce	Rear Wing Downforce	Front Wing Downforce	Rear Axle Downforce	Front Axle Downforce	Front Wheel Axle Moment	Rear Wheel Axle Moment
59.95885780 4652624	134.6128718 1676056	70.99468819 58776	57.23001772 1896964	73.20161	61.9422	-4026.24271 37950417	4758.104535 131717

## Data Results for Ver.2:

Raw Drag	Raw Downforce	Rear Wing Downforce	Front Wing Downforce	Rear Axle Downforce	Front Axle Downforce	Front Wheel Axle Moment	Rear Wheel Axle Moment
58.01410161 40345	127.0033675 9871627	71.69063774 236882	53.26742776 597585	62.48784	65.01471	-4225.95643 6004012	4061.709721 3535194

## Observations

Based on the data presented above, both of the new iterations of the bullhorn design improved in performance. This can be observed by taking a look at the gains made in downforce across the board. This is a desirable outcome but only one of the two new design iterations provides the most optimal performance. Of these two iterations, Ver.2 has the best performance. This is because the forces on the car are fairly balanced at the front and rear of the car. In addition, taking a look at the streamlines, it has far better control in directing messy air away from the rear wing. In comparison, although Ver.1 has better downforce numbers, the balance of the car is heavily skewed to the rear. Furthermore, its ability to direct messy air away from the rear wing is less than stellar. With all of this in consideration, the most optimal design iteration to continue developing is Ver.2.

**Design Goal: Create a sidepod that was easily manufacturable and would interact well with a future diffuser system.**

Part of the goal was to go 'back to the basics' - last year a focus was placed on the midsection of the sidepod, attempting to improve radiator mass flow and calculate each region inside of the sidepod itself.

While this was a good way to look at it, part of the problem was that some focus was lost between that interaction between the sizing of the sidepod's inlet and outlet.

This year is more focused on creating a simpler shape with an easier understanding that primary took into account the difference between the inlet and the outlet, so that more time could be spent on manufacturing and performing live tests of the sidepod to validate CFD simulations.

2024 Baseline Straightline											
CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downforce (lbf)	Raw Drag (lbf)	Radiator dP (Pa)	Front RH (in)	Rear RH (in)
1.24	0.62	47.45	54.24	40.30	55.00	7.46	101.29	53.21	202.70	4.88	5.55

Dennis Sidepod Iteration 1 - Initial Concept, flat top and swept undertray

CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downforce (lbf)	Raw Drag (lbf)	Radiator dP (Pa)	Front RH (in)	Rear RH (in)
1.18	0.62	54.20	50.40	44.48	59.70	1.96	96.85	50.64	151.27	4.88	5.55



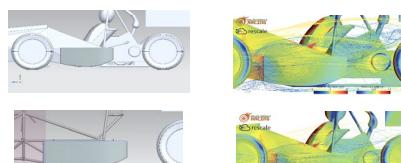
Dennis Sidepod Iteration 2 - Single sketch sidepod CAD, moving CoP closer to CoG

CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downforce (lbf)	Raw Drag (lbf)	Radiator dP (Pa)	Front RH (in)	Rear RH (in)
1.27	0.59	42.59	59.02	41.33	61.12	1.28	104.32	48.64	93.69	4.88	5.55



Dennis Sidepod Iteration 3 - Removing space so Sidepod is behind firewall, increasing AoA

CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downforce (lbf)	Raw Drag (lbf)	Radiator dP (Pa)	Front RH (in)	Rear RH (in)
1.27	0.59	45.45	58.96	41.39	61.11	1.74	104.00	48.47	0.02	4.88	5.55



Dennis Sidepod Iteration 4 - Adding cut to shoulder by the firewall for improved feasibility

CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downforce (lbf)	Raw Drag (lbf)	Radiator dP (Pa)	Front RH (in)	Rear RH (in)
1.19	0.62	44.84	53.12	41.33	56.95	1.96	97.59	50.46	140.67	4.88	5.55



Dennis Sidepod Iteration 5 - Resizing for radiator compatibility, determining potential mounting solutions

CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downforce (lbf)	Raw Drag (lbf)	Radiator dP (Pa)	Front RH (in)	Rear RH (in)
1.20	0.61	45.01	53.95	4.20	58.33	1.96	98.58	50.19	146.77	4.88	5.55



Note 1 about mounting: Powertrain team has agreed to design a mounting system for both the radiator and the sidepod mounting, so current sidepod mounting design has been scrapped.

Note 2 about mounting: sidepod had to be elongated past the firewall in order to properly fit the radiator provided by the powertrain team. benefit of improving the sidepod's tire wake control.

While this will make manufacturing harder, this is necessary to fit the radiator's dual pass system. This will also have the added benefit of improving the sidepod's tire wake control.

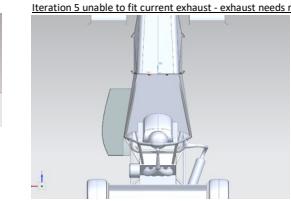
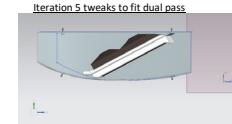
Note 3 about mounting: Sidepod on the right side of the vehicle intersected with the current geometry of the exhaust. Powertrain has agreed to reshape the exhaust based on bounding boxes.

General Note 1: In .sims ran here, the sidepod designs were in comparison with the baseline package from 2023.

General Note 2: Something of interest here is that the new sidepod downforce numbers are considerably lower than the ones from the baseline, and yet when placed into the aeromap analysis, it seems as though the sidepod produces overall higher downforce compared to 2023.

An initial hypothesis is that with the downforce improvements made from the front and rear wing 2024 designs, the sidepod is closer to the ground and has more laminar air fed into it, leading to performance gains as compared to the 2023 sidepod.

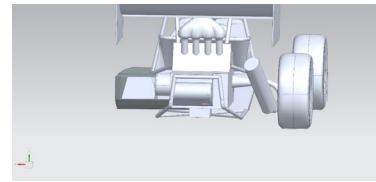
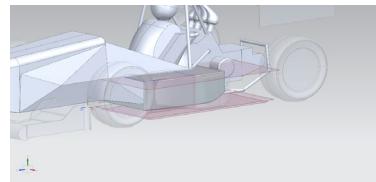
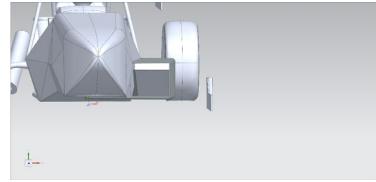
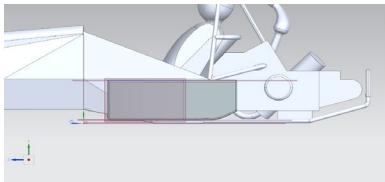
Testing with the 2023 sidepod on the 2024 aero package would be useful for determining why this contrast is seen. It is possible that the 2023 sidepod may still be the better performing choice compared to the 2024 design.



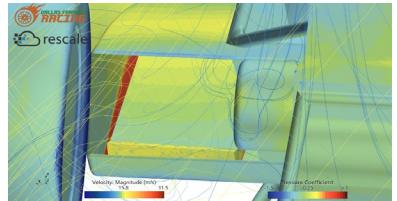
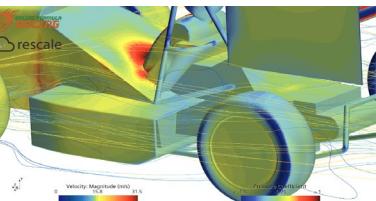
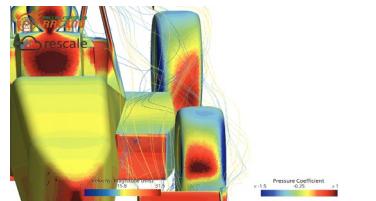
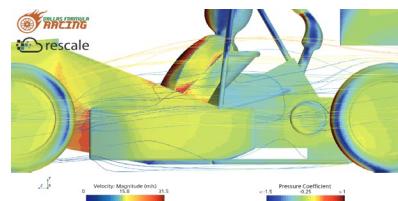
(Note: Panelling was included in all sidepods on the inside of the sidepod by the radiator to inhibit airflow around the radiator.)

#### Iteration 1

##### CAD

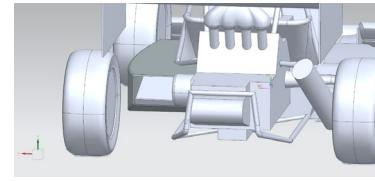
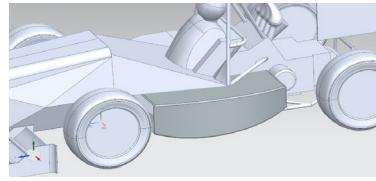
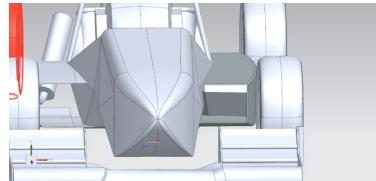
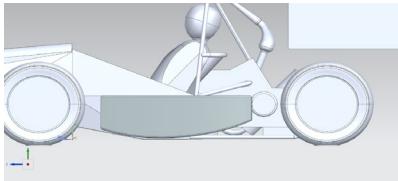


##### CFD

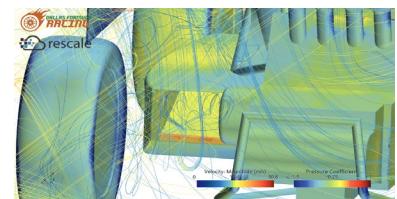
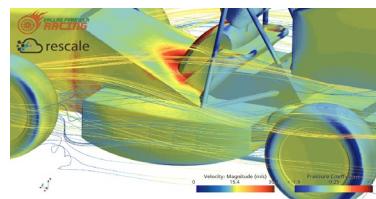
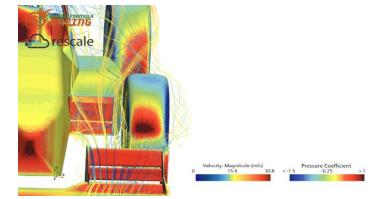
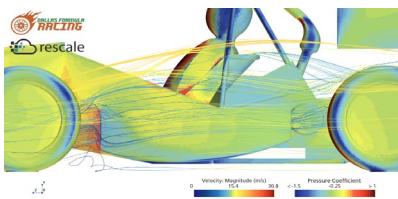


#### Iteration 2

##### CAD

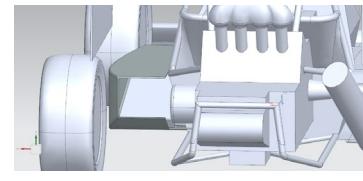
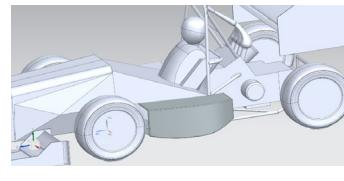
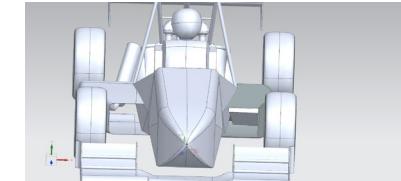
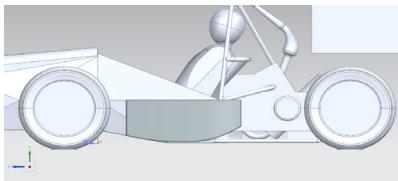


##### CFD

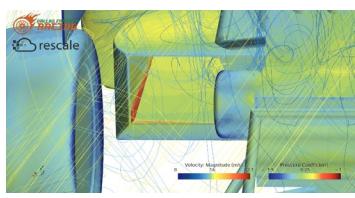
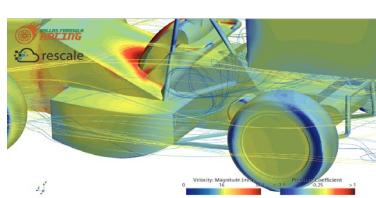
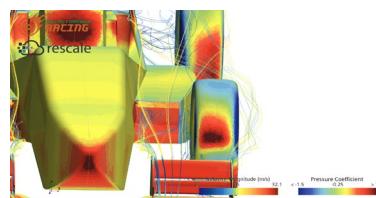
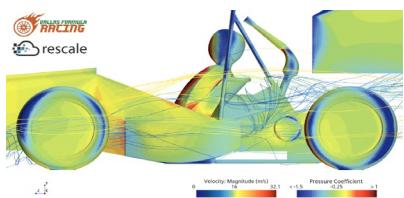


#### Iteration 3

##### CAD

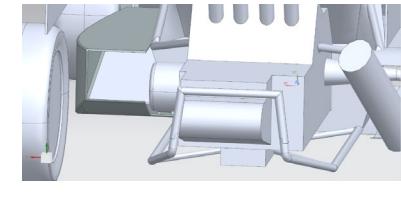
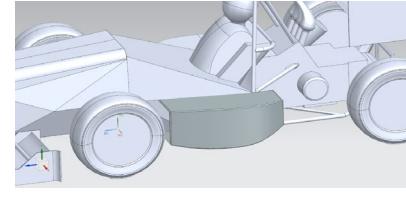
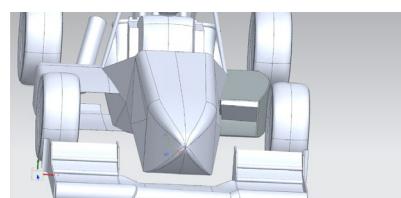
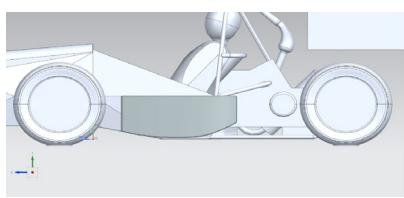


CFD

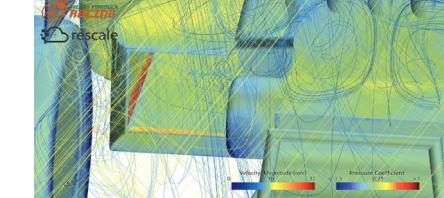
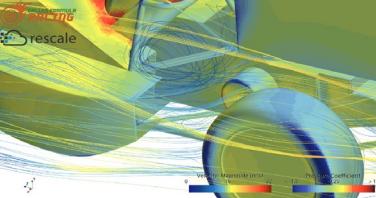
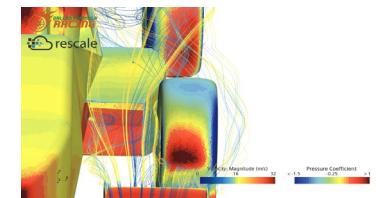
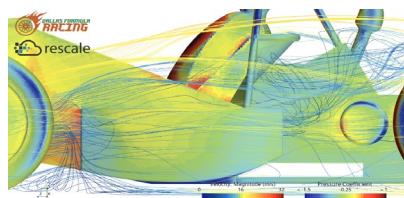


Iteration 4

CAD

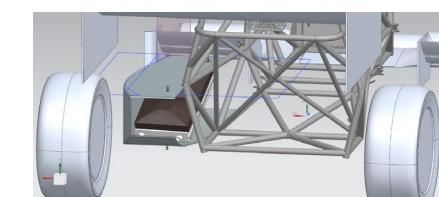
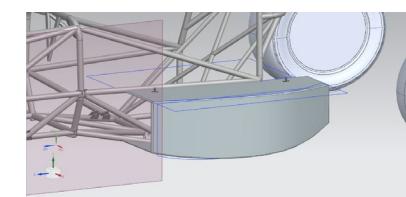
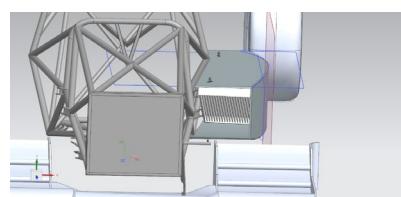
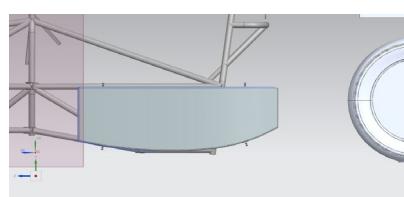


CFD

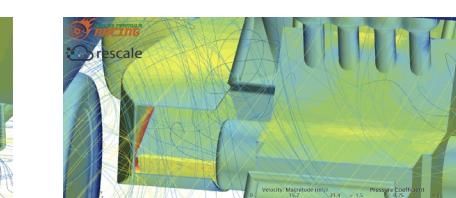
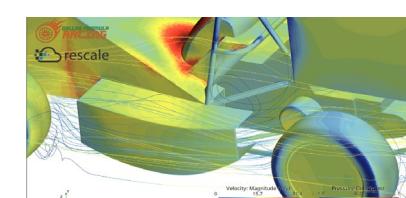
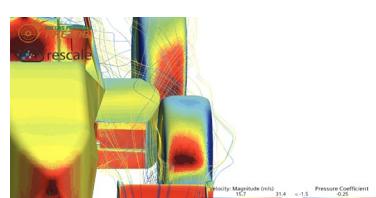
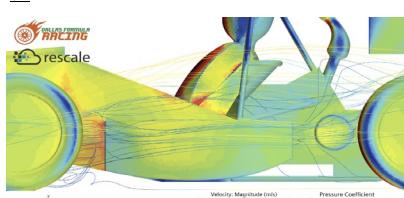


Iteration 5

CAD



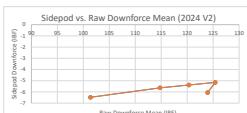
CFD





In recent aerom testing, an inverse relationship between sidepod downforce and raw downforce mean was noticed.

2024 V2					
Front Rideheight (lbf)	Rear Rideheight (lbf)	Sidepod Downforce (lbf)	Raw Downforce Mean (lbf)	Chassis Angle (degrees)	Chassis Heave (degrees)
6.44	4.09	-6.48	101.38	-1.81	-0.14
5.65	4.81	-5.63	114.77	-0.91	-0.14
5.28	5.17	-5.38	126.24	-0.47	-0.14
4.83	5.59	-5.15	125.40	0.06	-0.14
4.37	6.01	-6.06	123.93	0.58	-0.14



The initial hypotheses included:

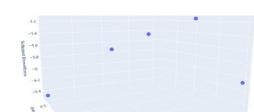
1. The sidepod was creating turbulent air and feeding it into the rear wing, reducing the downforce produced and increasing drag.
2. The sidepod performed better at certain ride heights than the rest of the aero package, so sidepod downforce may have been up when the total downforce produced may have gone down.

Several plots were created to showcase the following relationships:

1. The sidepod's downforce in comparison to the front and rear wing downforce produced.
2. The sidepod's downforce in comparison to the front and rear ride heights.
3. The front wing's downforce in comparison to the front and rear ride heights.
4. The rear wing's downforce in comparison to the front and rear ride heights.

In the future downforce should also be compared between the chassis angle and chassis heave, although more data points are needed (currently all sims ran with chassis heave at -0.1429).

Sidepod Downforce (lbf) vs. Rear Rideheight (in.) vs. Front Rideheight (in.)



Sidepod Downforce (lbf) vs. Rear Wing Downforce (lbf) vs. Front Wing Downforce (lbf)



Front Wing Downforce (lbf) vs. Rear Rideheight (in.) vs. Front Rideheight (in.)



Rear Wing Downforce (lbf) vs. Rear Rideheight (in.) vs. Front Rideheight (in.)



Currently, there is not enough aeromap testing data to give a justified conclusion on this relationship, although it appears as though the relationship is overall positive, not inverse.

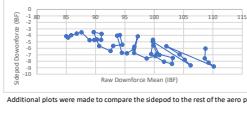
However, based on 2023 sidepod data, a positive linear relationship between sidepod downforce and raw downforce mean was found.

Additionally, plots were created using the same parameters as the previous analysis. These plots confirmed that there is a positive relationship between sidepod downforce and raw downforce mean.

2023

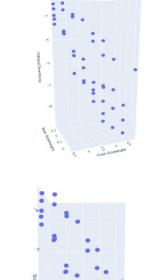
Front Rideheight (lbf)	Rear Rideheight (lbf)	Sidepod Downforce (lbf)	Raw Downforce Mean (lbf)	Chassis Angle (degrees)	Chassis Heave (degrees)
6.44	4.09	-4.13	85.01	-1.81	0.35
6.44	4.09	-4.34	85.31	-1.81	0.00
6.45	4.09	-3.98	85.83	-1.81	0.25
6.45	4.10	-3.72	85.77	-1.81	0.50
6.47	4.11	-3.53	87.59	-1.81	0.75
6.15	4.34	-4.74	88.97	-1.49	0.25
6.15	4.35	-4.72	88.79	-1.49	0.25
6.15	4.36	-4.62	90.27	-1.49	0.25
6.17	4.36	-4.70	90.95	-1.49	0.50
6.18	4.37	-4.77	91.02	-1.49	0.75
6.19	4.38	-5.21	85.57	-1.49	1.00
5.86	4.60	-5.64	90.61	-1.16	0.50
5.87	4.61	-6.40	92.54	-1.16	0.00
5.87	4.61	-6.62	90.00	-1.16	0.00
5.89	4.63	-5.45	94.54	-1.16	0.50
5.90	4.64	-3.95	94.14	-1.16	0.75
5.91	4.65	-4.05	93.67	-1.16	1.00
5.93	4.66	-4.69	94.48	-0.83	0.25
5.98	4.87	-6.77	95.34	-0.83	0.25
5.60	4.88	-6.12	96.53	-0.83	0.25
5.60	4.89	-5.75	96.60	-0.83	0.50
5.62	4.91	-4.16	97.19	-0.83	1.00
5.29	5.12	-6.72	96.57	-0.50	-1.00
5.30	5.13	-7.57	96.71	-0.50	-0.50
5.31	5.14	-7.20	100.19	-0.50	0.00
5.31	5.15	-7.04	100.89	-0.50	0.25
5.33	5.17	-5.00	99.82	-0.50	0.75
5.34	5.17	-4.59	99.79	-0.50	1.00
5.01	5.38	-6.89	99.47	-0.18	-1.00
5.02	5.39	-8.06	101.66	-0.18	-0.50
5.03	5.41	-8.41	103.03	-0.18	0.25
5.04	5.42	-7.49	103.16	-0.18	0.50
5.05	5.43	-6.82	100.49	-0.18	0.50
5.05	5.43	-5.54	99.85	-0.18	0.75
5.06	5.43	-4.37	99.76	-0.18	1.00
4.74	5.67	-8.63	106.19	0.15	0.00
4.75	5.67	-7.75	105.18	0.15	0.25
4.75	5.68	-6.94	104.32	0.15	0.50
4.77	5.69	-5.68	102.07	0.15	0.75
4.46	5.93	-8.80	110.16	0.48	0.00
4.47	5.94	-8.21	109.36	0.48	0.25
4.47	5.94	-7.55	108.60	0.48	0.50
4.21	6.22	-6.04	108.74	0.81	1.00

Sidepod vs. Raw Downforce mean (2023)

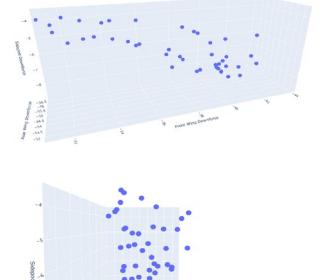


Additional plots were made to compare the sidepod to the rest of the aero package using the same positional arguments as the plots for the 2024 design.

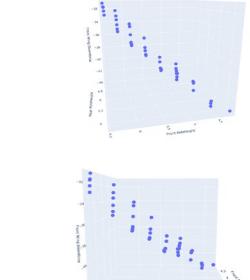
Sidepod Downforce (lbf) vs. Front Rideheight (in.) vs. Rear Rideheight (in.)



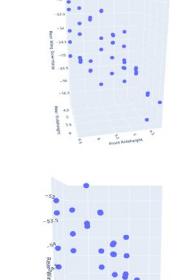
Sidepod Downforce (lbf) vs. Front Wing Downforce (lbf) vs. Rear Wing Downforce (lbf)



Front Wing Downforce (lbf) vs. Rear Rideheight (in.) vs. Front Rideheight (in.)



Rear Wing Downforce (lbf) vs. Rear Rideheight (in.) vs. Front Rideheight (in.)

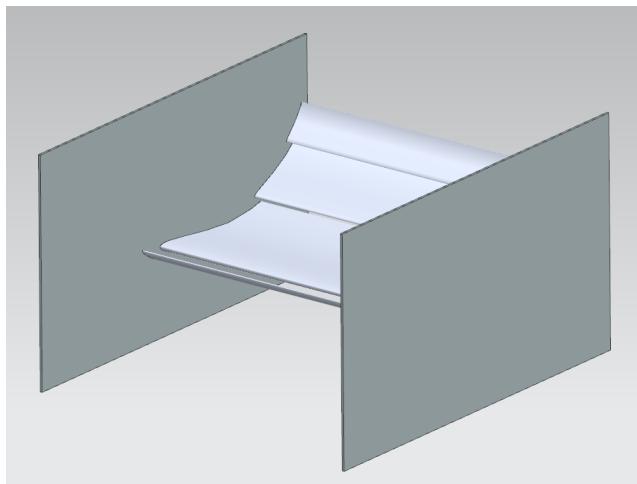


# **2024 REAR WING REPORT**

By: John McAlister

## Overview:

My 2024 rear wing design features an iterative biplane study. My initial step was to figure out areas that could be an issue in performance. My main concern was how the high-pressure region above the main stack of elements will interact with the low-pressure region underneath the biplane system. The second concern was what is the ideal AOA for all elements. After understanding where issues may arise I then started to plan out a step-by-step guide for each version of the rear wing. An important part was only allowing one major change for each simulation so that it is clear that that change caused whatever difference is shown. My iteration guide and results are as follows:

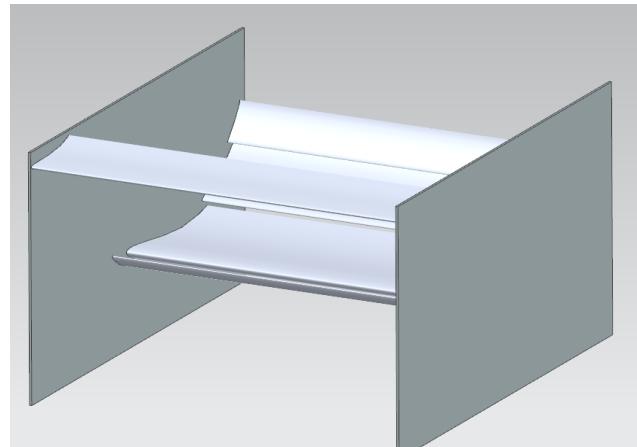


### Iteration 1:

- Run sim with large square endplates and the main elements in a low AOA position. This will set a baseline to see if a biplane actually improves the rear wing's performance

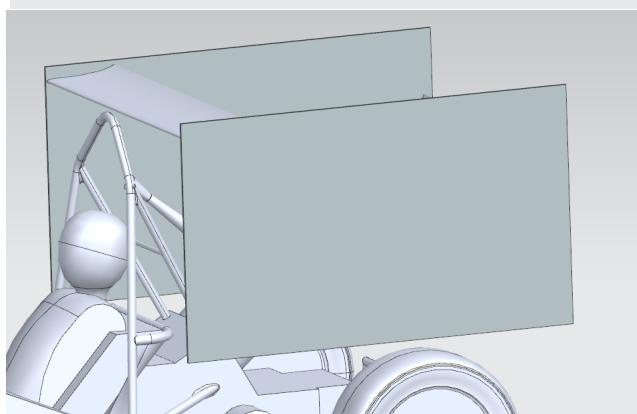
Cd	CdA	Cl	ClA	Drag	DownForce
1.173	0.6004	2.38	1.218	51.5	99.84

I would then use these results as a baseline for if my next iteration with a biplane results in increased performance.



**Iteration 2:** Add an 8" airfoil with limited AOA. Analyze high and low-pressure regions. As well as make a claim on biplane validity.

Cd	CdA	Cl	ClA	Drag	DownForce
1.177	0.627	2.35	1.25	51.38	102.6

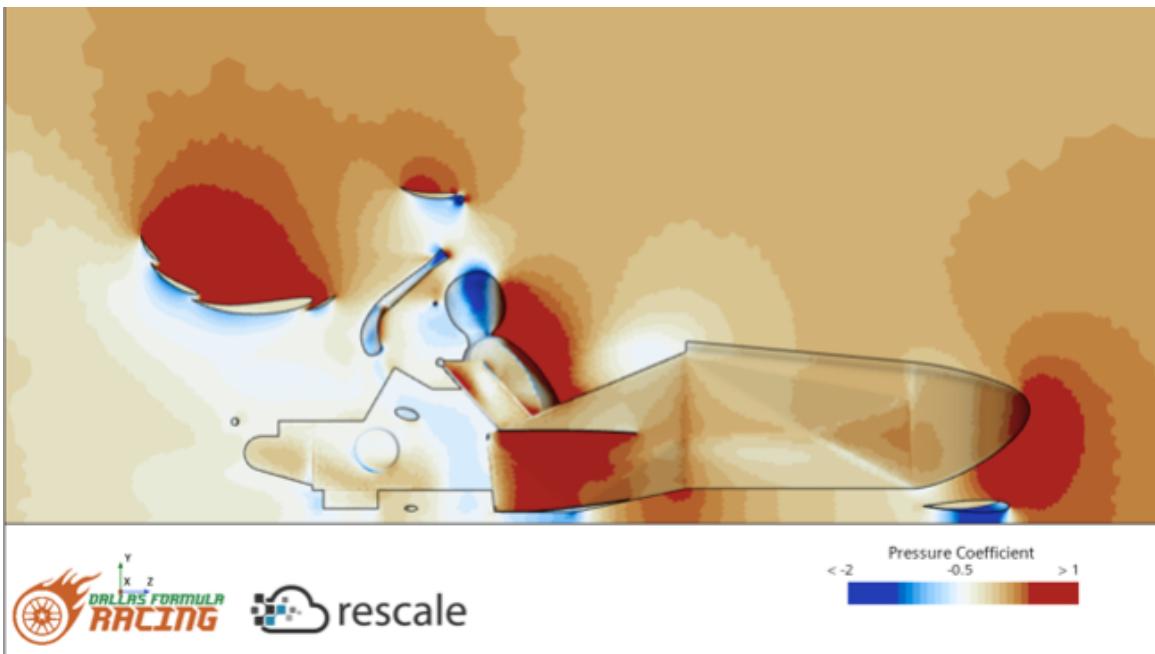


### Analysis:

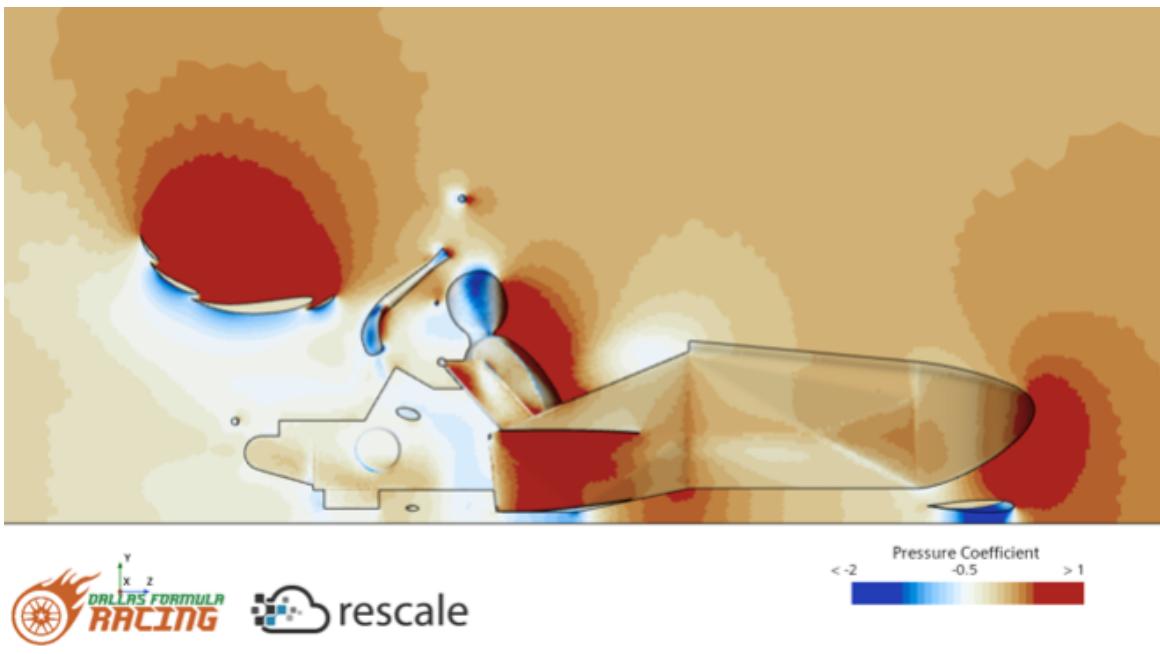
From these results, I see a good increase in downforce which suggests this is a worthwhile change to the rear wing design. This helps provide a reason to follow down a biplane study and see how much extra performance can be gained. An interesting result was also found in the reduction of drag which was not

expected and is something that needs to be analyzed further. With these two simulations, it is also possible to now analyze my first concern mentioned in the overview by using a pressure symplane.

## Pressure Symplane with Biplane

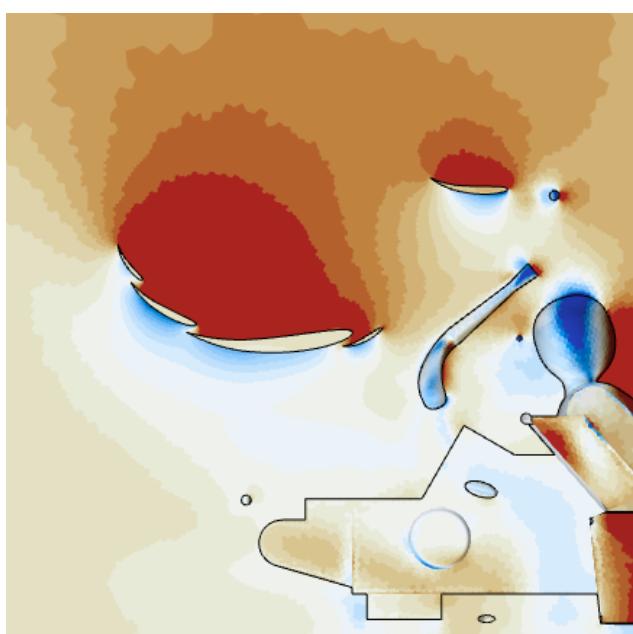
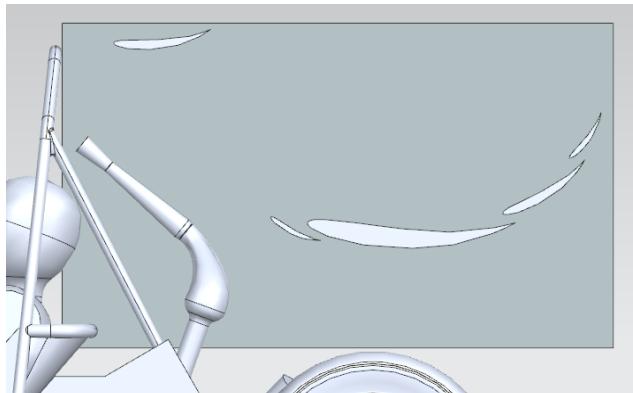


## Pressure Symplane without Biplane



## Pressure Symplane Analysis:

These two pressure scenes provide a lot of information about the interaction between the biplane and the main element systems. There is a larger region of high pressure in the “without biplane” symplane which most likely reduces the amount of downforce the main elements generate. This also means that there is an interaction between the two pressure zones despite the far distance between the systems. Despite this, the addition of the biplane did increase the overall downforce generated. For even more performance a way to reduce the interaction for these pressure zones should be the goal. A possible solution that will be tested in future iterations will be a slot that is cut into the endplate to bleed off high-pressure air before it interacts with the low-pressure region of the biplane.



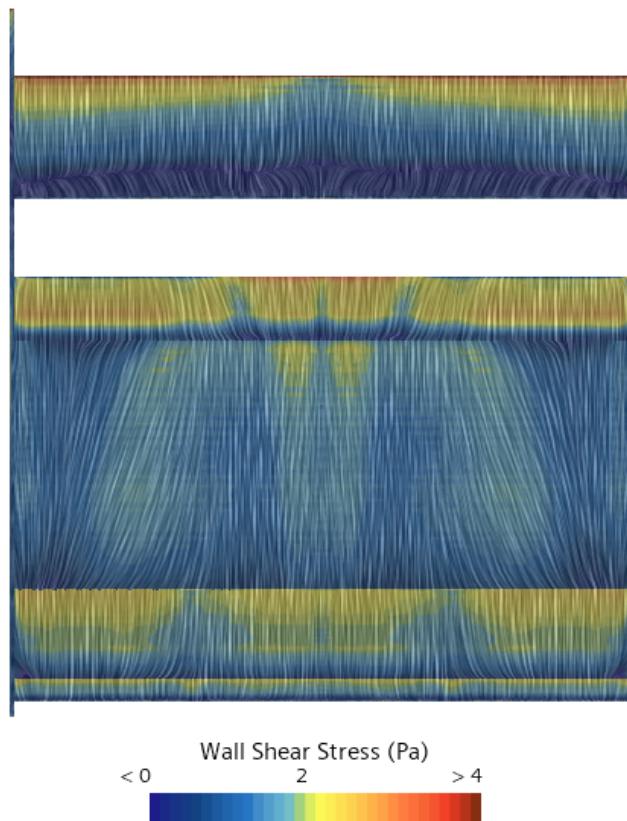
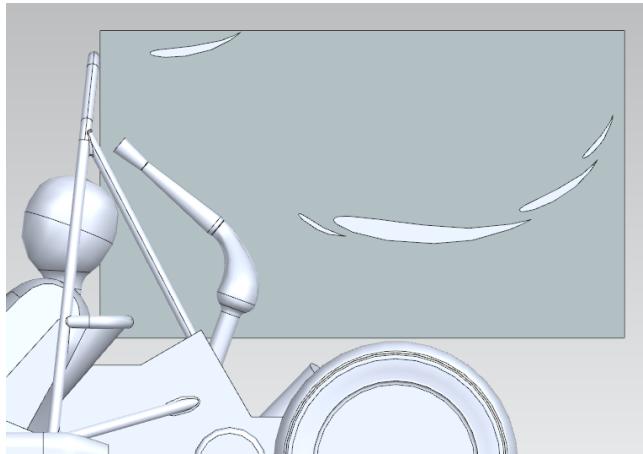
### Iteration 3:

- Before endplate design can be considered first an optimal biplane position should be found. The first step towards improving biplane performance was to figure out how it interacted with the rest of the vehicle. Originally it was very close to the top of the roll hope. For this iteration, it will be moved back to see how it affects pressure zones as well as its overall downforce.

Cd	CdA	Cl	ClA	Drag	DownForce
1.174	0.625	2.35	1.25	51.25	102.6

### Analysis:

Despite moving it closer to the main elements there was little change in the reported forces. Even when analyzing the pressure symplane it is very similar to iteration 2. This lets us know that we can make the rear wing a little smaller without sacrificing performance.



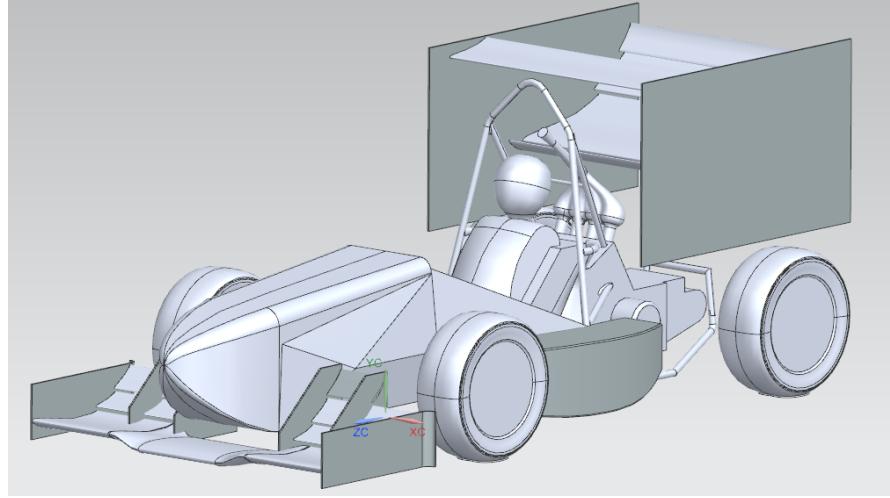
#### Iteration 4:

- With this version I the AOA of the biplane is increased by 5 degrees in the hopes of improved downforce.

Cd	CdA	Cl	ClA	Drag	DownForce
1.177	0.634	2.339	1.26	51.99	103.3

#### Analysis:

This iteration shows a massive reason why there are minimal increases to downforce. This is shown when you analyze the wall shear stress of the rear wing. As you can see in the image to the left there is a definitive dark blue region on the biplane that shows there is a large amount of flow separation happening which is limiting the biplanes performance.



### **Iteration 5:**

- For this iteration, we need to hold off on fixing issues with biplane flow separation and move over to a necessary change with the main elements. The current configuration is a low downforce AOA and due to a lack of rear axle moment, the main elements were switched to their high downforce position. A sensitivity study will be done to see how the increased AOA performs in different positions but for straight line, this is a proven setup. Also, this iteration is the first to include the new Front wing. This will allow for changes to be made with front-wing interaction in mind. The results from this will be the new baseline values.

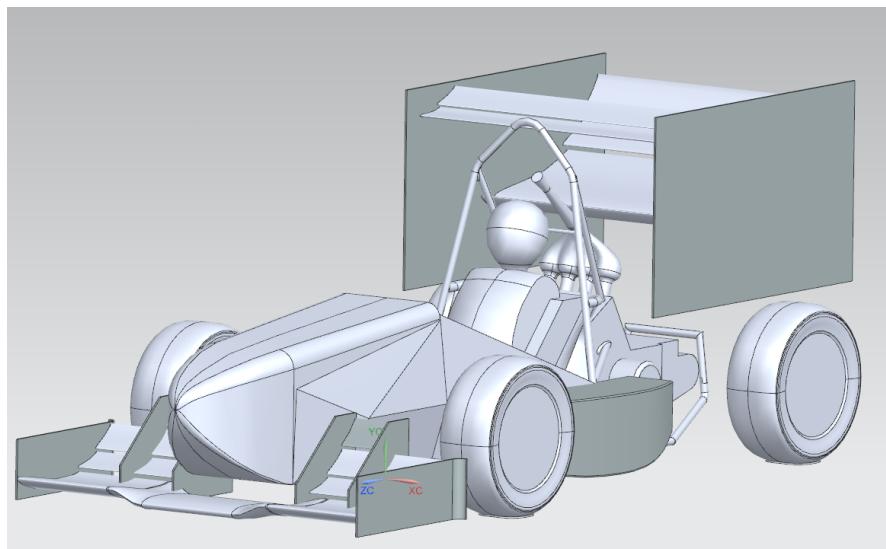
Cd	CdA	Cl	ClA	Drag	DownForce
1.283	0.751	2.693	1.576	61.51	129.1

### **Analysis:**

Overall the entire aero package provides a 29% increase in downforce from iteration 1 to iteration 5. The increased main element AOA, the addition of the biplane, and the new front-wing interaction are all important factors for this increase. However, there is still a lot more on the rear wing that can be improved. For example, the flow separation on the biplane and the endplate design should allow for even better performance.

### **Iterations 6 and Beyond:**

As of right now iteration 5 is the latest simulation. However, there are plans for future changes to be tested. Iteration 6 will change the biplane system to a 2 element system. If positioned correctly smaller elements should increase performance by decreasing the amount of flow separation present. It should look similar to the model below. For future iterations, some fine-tuning of the two-element biplane system might be necessary and then endplate design should be the next consideration after the elements are finalized. Straightline and crosswind simulations will be done to understand what is the optimal endplate design. However, some predictions as of right now will be testing the aforementioned cuts in the endplate as well as varying sizes.



## Beam Wing Element Stint 3 Run

10 Degree Iteration ... Radiator Streamline Position (Low DF)									
CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downforce (lbf)	Raw Drag (lbf)	Radiator dp (Pa)
1.6419	0.7338	80.8589	53.8147	60.1602	66.8245	5.176	134.1454	60.3115	63.2341

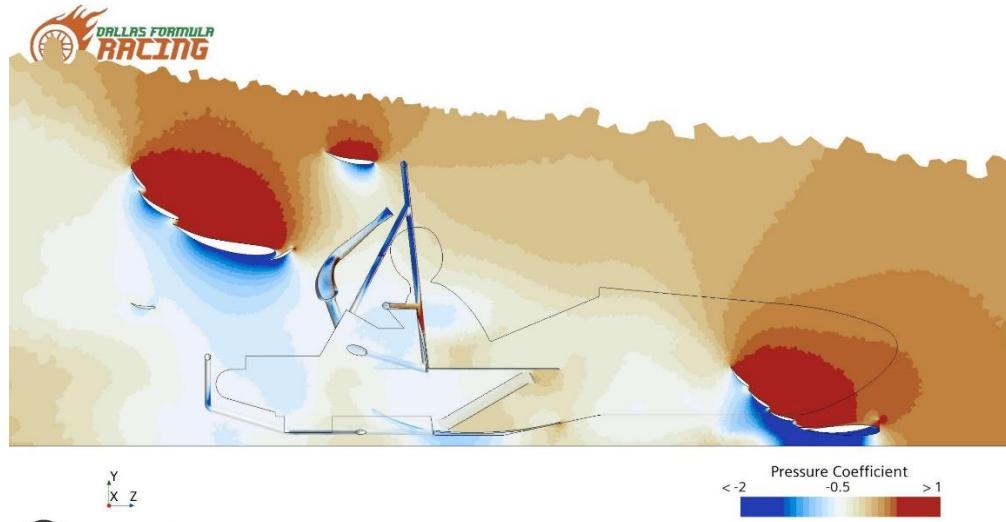
Current Iteration

10 Degree Iteration ... Mid Endplate Position									
CIA	CdA	Front Axle Downforce (lbf)	Rear Axle Downforce (lbf)	Front Wing Downforce (lbf)	Rear Wing Downforce (lbf)	Sidepod Downforce (lbf)	Raw Downforce (lbf)	Raw Drag (lbf)	Radiator dp (Pa)
1.3408	0.6499	68.845	41.9348	52.2366	53.226	6.1701	109.8705	53.2581	88.8545

Old Iteration

### Analysis

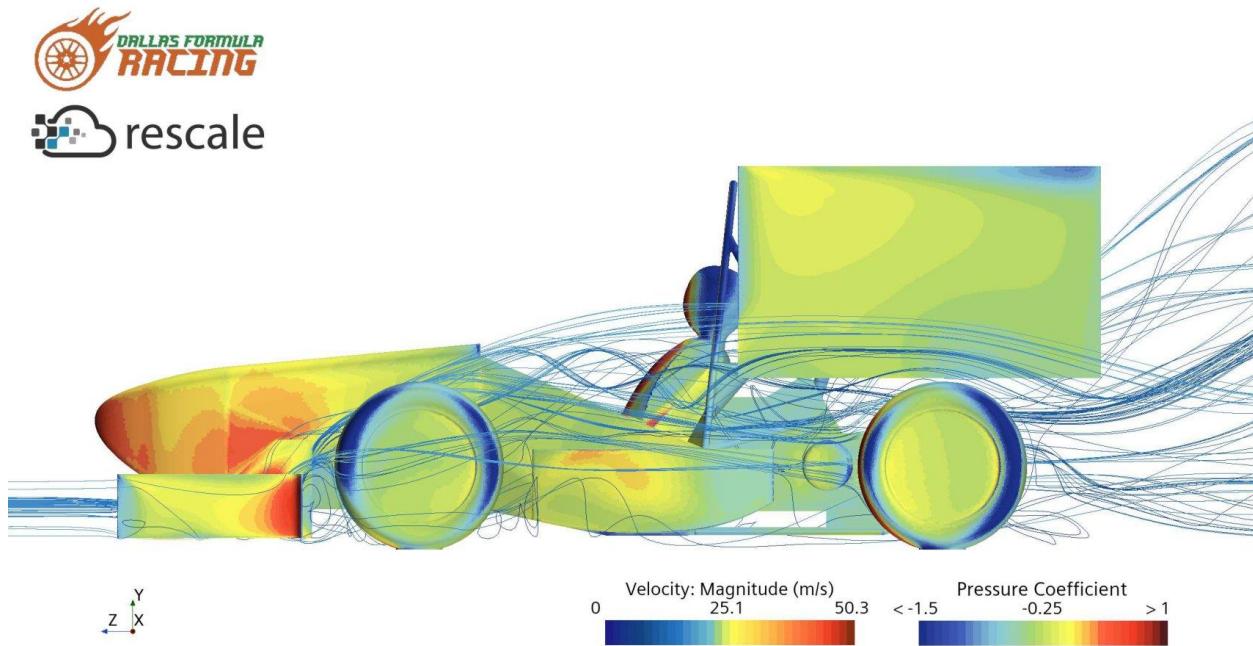
In analysis of the preceding two sims, it becomes evident that positioning the beam wing element on the rear wing system is most efficient within the streamlines of the radiator outflow. We encountered a ~25.5% increase in the total downforce on the rear wing system, as well as a ~28.4% increase on the rear axle downforce. While the rear axle downforce still largely differs from the front axle downforce, the rear wing downforce itself has increased considerably. In order to combat this large difference in downforce, it may be necessary for an additional beam wing or biplane element in order to increase the downforce, possibly fixing this large inconsistency among the axle downforces. It is interesting to note that the rear wing downforce itself is larger than the front wing downforce, although the rear axle still pales in comparison to the front axle downforce. In any case, these should be noted after everyone's parts have been combined to makeup the main system. NOTE: The large difference in axle downforce may be impacted by the excess weight of the transmission being absent, as well as other vital components needed.



Cutplot Image



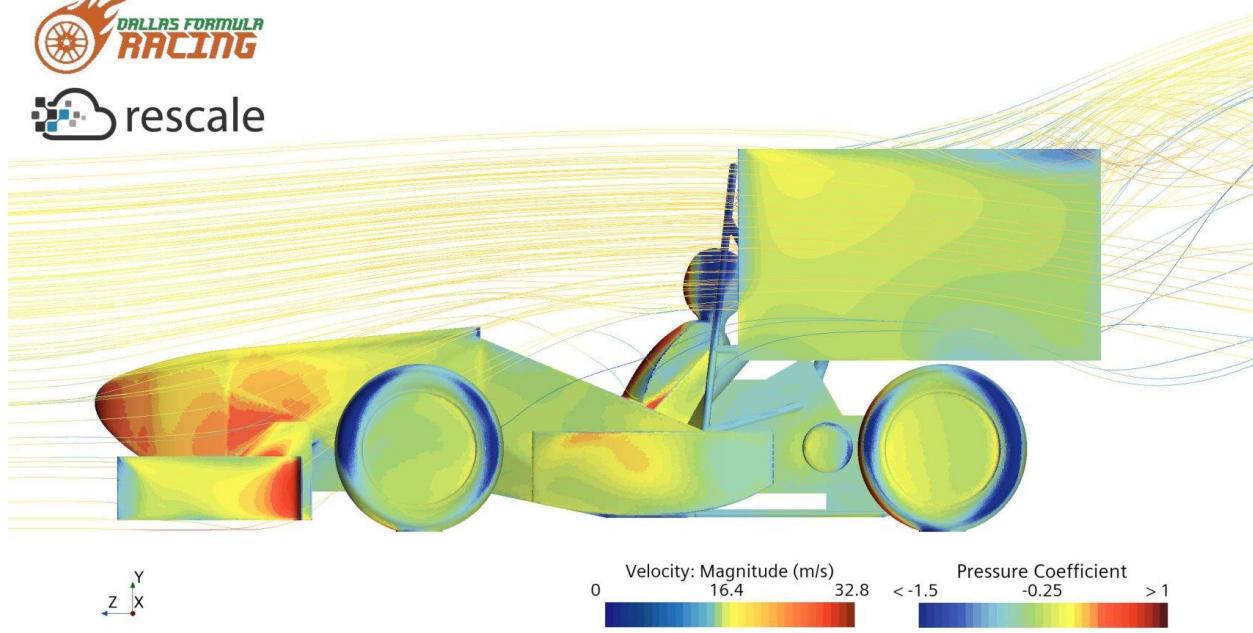
Residuals



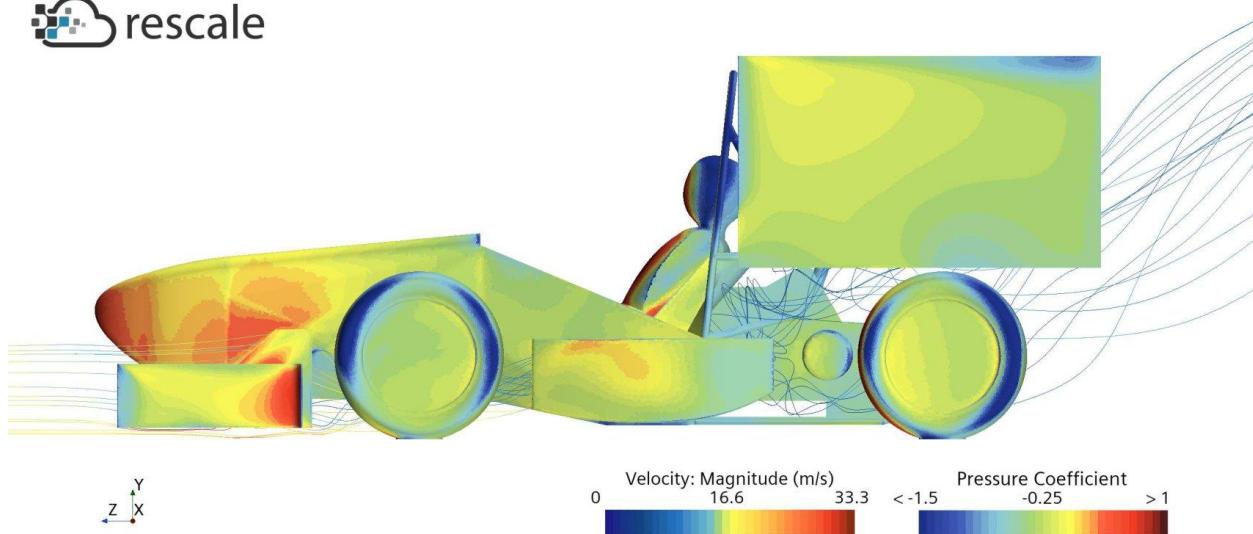
Front Wing Streamlines



rescale



rescale



## Process to Run and Post Process

### Aerodynamics Simulations

#### Creating the Macro

## Design Philosophy

This is by far the most important step in the whole process, so being meticulous and designing for failure are key. It is important to note that StarCCM implements the Java runtime for their macros. With this in mind, the scripts should be written with object-oriented philosophy as well as ensuring as many processes are contained in their own methods. Our scripts are written with two primary goals in mind. One, cause no errors especially logical errors. Two allow for as much reusability and fast customization as possible.

## Deciding on a Job Type

The first step of choosing which method to parameterize the simulations with, falls in line with what our initial goals and timeline are. For instance, are we running an angle sweep only changing one variable, or do we need to iterate through the complete combination matrix of input parameters. Furthermore, we need to decide if we want to run the study in one job or in batches. Running in batches allows for the study to be run as fast as the user wants while also greatly reducing the opportunity cost of a sim failing,

but they do come with extra overhead to setup and postprocess. A single job on the other hand, is mainly used to eliminate overhead and can be used when runtime is not a factor, but the user's time is. The main deciding factor is the access to resources depending on which HPC cluster you want to use.

## Bounds checking

Given how costly each sim is to run, our script needs to able to take out sims that will fail as early in the process as possible. The current method breaks this down into two checks. The first method defines a set of known parameters that will be out of bounds based on past simulations. While this will inevitably exclude a few sims that could be in bounds in a new study, this allows the results to be similar across studies, and the total time saved is much greater than even rerunning a few sims. Method two requires the surface wrapper to run to completion and then using the “Signed Distance” report to check if the distance is less than 0. Although this is not as efficient as the pre checking method, the surface wrapper only takes about 25% of total run time, so potentially saving 75% of the cost is a massive help.

## Error Catching

The number one goal of this script is to cause no errors, so this also means that the code needs to catch all errors to prevent any fatal crashes. The sim itself will throw many different errors during both meshing and iteration if the sim fails. Additionally, saving files will throw another error, so the code needs to have redundant error checking to ensure that we know where and what caused the error and that we prevent the error from

crashing the job. This is implemented using various try, catch statements wrapping around each section we know will throw an exception.

Moreover, we need to prevent any logical errors or runtime exceptions such as index out of bounds. The latter is easy to solve as we can implement the Java collections framework to provide our abstract data types instead of using a normal array. Logical errors are a much harder problem to deal with especially since we are not currently running any unit tests in the code but will in the future. However, we can solve most errors by simply reusing the same scripts that have been tested before. Each of these scripts is written to address a certain job type and to be run on a certain HPC cluster. This allows us to plug and play any new study we want with only needing to change the input parameters thereby eliminating almost all room for error.

## Output

The current method to save the output of each sim is to write the final report data into a csv file and then save each scene that is associated with the sim in Star. The input parameters have to be appended to the csv looping through the values in the report as they are not saved in the report.

## HPC Platform and Architecture

## Rescale

Rescale is our main provider for cloud computing resources. Their website offers a simple user friendly way of sending jobs without any prior knowledge with nearly any type of

hardware you would need. However, the service does cost money and optimizing which hardware to use to minimize total costs is a top priority.

Running sims includes three parts which pose different strains on the computer. The surface wrapper is single threaded meaning that we need to optimize for higher clock speed over core counts. Both the volume mesh and iteration are multi-threaded and can take advantage of all cores we have access to. Rescale tends to have certain issues syncing high core workflows across multiple nodes while also accessing the large amounts of data being processed during each sim. We can prevent the sync errors by unbinding the CPU when we call the StarCCM script and the data access can be helped with faster interconnect speeds though not strictly necessary compared to unbinding the CPU.

The three main concerns we look for when choosing hardware on Rescale are price, CPU clock speed, and interconnect speed. Price is straightforward as we just look for the cheapest option which means we can increase core counts as price is based on per core. Currently the three architecture types listed below are the main ones we have tested and used. We will most likely move to GPU acceleration nodes once the license issues can be worked out.

Type	Core Count	Clock Speed	Interconnect	Price per Core	Total Cost Per Hour
C5n Max	108	3.0 GHz	100.00 Gbps	\$0.0768	\$10.4544
HPC6id	128	3.5 GHz	200.00 Gbps	\$0.0957	\$14.8096
hpc5a-72	144	3.3 GHz	100.00 Gbps	\$0.0499	\$10.0656

Starting from top to bottom, C5n Max is the easiest and standard node type that we use and poses few issues and overall good value. Our current sim settings for half car take almost an hour exactly to run through completely, so a per hour cost of \$10.4544 can also be used as a baseline for per sim cost. HPC6id was tested to optimize heavily meshing bounded sims as it reduces meshing times by half given its faster single threaded performance. This speed allows for the total runtime of the sim to drop to about 75% meaning that the per sim cost drops to \$11.1072 making this a much more enticing node if we are short on time. Hpc5a-72 has become the main architecture that all of our studies on run on currently. We had to avoid this node in the past due to issues with Rescale and StarCCM, but turning off CPU binding has eliminated that problem. In testing, this node has decreased runtime per sim by roughly 3-4 minutes per and total sim cost to roughly \$9.2268 per. Hpc5a also had a 96 core per node version that is even cheaper than this, but we have not tested it yet since it had the same synchronization errors in the past. For future larger scale studies, we could combine HPC6id for meshing and hpc5a for its cost savings though given how optimized the current process is, this might not be time efficient on the user overhead.

## Ganymede

Ganymede is UTD's main HPC cluster that we have access to for free. Given that it is a first come first serve queue, deciding what hardware to use comes down to simply what is available. Along with this, we have not had any hardware related issues on here compared to Rescale. Ganymede allows for interactive and batch jobs meaning that we can much easier open an SSH tunnel and have Star display locally with a GUI while operating off of their computers, or we can queue them using a batch script and have the whole process run automatically. For our

purposes the batch script defines the hardware and other job settings for the Slurm job scheduler to use and the shell command to start StarCCM with our input sim and Java macro. Sim runtime for 108 cores of type 128s takes roughly an hour to run as well.

## Post Processing

### Compiling Results

This step will vary depending on how many sims were ran and the proportion of successes to failures. Every sim that did not have a fatal crash will save a csv file, so the sims that meshed but out of bounds will still save a csv. If the number of sims ran is sufficiently small or the job was run on Rescale, the failed csv files can be excluded manually by looking at the file size in bytes. However, if there are too many sims to do manually, we will compile all csvs into a folder and use a python method to exclude them.

Once the files are saved locally, we will use a python script to do most of the postprocessing. First we need to merge the csv files into one. The merge method depends on the type of job that was run to know which file names to use. For example, a batch job will need to merge csvs with both a batch number and a sim number while a single sweep will only need to track the current sim number. Note this process needs to be wrapped in a catch to avoid any file not found errors from failed sims or otherwise any skipped sim numbers. The compiled data will then be saved into a new csv file.

We utilize the data frame data structure from the Pandas framework to read and manipulate the data. This is an industry standard framework for all data analysis and machine learning done with python and therefore offers many predefined methods and is used by most other visualization and analysis frameworks.

## Cleaning the Data

Then we need to check and clean any data values that might be invalid before continuing.

If we compiled the failed csv files in as well earlier, we can loop through the data frame checking if the raw mean downforce value is 0 and then removing those indices and saving the remaining ones into a new csv. Given that if a csv ran to completion and is still in the data frame, it is very rare that any more cleaning needs to be done. These values can be checked manually or by calling the scatterplot method and looking for any outliers. The only values that have had outliers are raw downforce mean, and front axle downforce mean.

## Visualization

Most plots can be created immediately after the data is compiled using the Plotly.py framework. Since Plotly has integration with the Pandas data types, almost all plots can be made by just specifying the column header to use. For instance, to plot a scatterplot of front ride height vs front axle downforce, you just need to pass in the data frame and use those names in the x and y variables. More detail is listed in the python script that is listed in the github repo.

## Creating the Aeromap

First it is important to specify that our current method of deriving the contour data is dictated by the succeeded data and their input values. For example, the built in methods in Plotly take the z dimension as a M x N array where M and N are the respective lengths of the x and y input series. Meaning that the method heavily relies on input parameters that follow that same schema. Our current method of deriving front and rear ride heights are almost exclusively

proportional to chassis angle; moreover, the method leads to the ride heights being inversely proportional to each other. Therefore, we need to use a different method of creating the contour data in a manner that aligns with those constraints.

The first step is to convert the z dimension representing downforce into a form that can be seen on a 2d graph. The easiest method of doing this is to utilize a 2d histogram with color representing downforce. To do this we copy each point  $f(x,y) = z$  by  $z$  times. This conversion turns the data into a heatmap style where we can plot the contour lines to present 3 dimensions onto 2. We then save this new data frame into another csv.

We can finally call Plotly's 2d histogram method specifying x, y, z as front ride height, rear ride height, and raw mean downforce respectively. The conversion made earlier 'tricks' the method to sum the points  $z$  times to display downforce. Lastly we can update the ranges to more accurately display the new downforce values as they change across each study.

## Further Analysis

After all previous steps are completed, we can now compare the results to the past studies. Here it is crucial that the plots and ranges of input values are the same across all studies. Of course, as we make more or less downforce the range on the plots will change slightly, and in that case we can go back and change those ranges to suit the new one or remember to take the different ranges into account when comparing.

Lastly, we can utilize fundamental practices in machine learning to help determine what parameters are actually impactful towards producing more downforce. This is done using correlation matrixes and their associated scatter matrixes. Given  $n$  random variables, the  $n \times n$  matrix is defined as below provided that their standard deviations are greater than zero.

$$c_{ij} := \text{corr}(X_i, X_j) = \frac{\text{cov}(X_i, X_j)}{\sigma_{X_i} \sigma_{X_j}}$$

We then take the downforce column and sort by greatest to least. Shown below is the output of the correlation matrix using the contour data of 2024 study v2.

Raw Downforce Mean	1.000000
Rear Rideheight	0.940829
Chassis Angle	0.940677
Sidepod Downforce	0.417350
DifferentialPressure	0.301351
Chassis Heave	0.292447
Front Wing Downforce	0.202704
Rear Wing Downforce	0.080492
Front Wheel Axle Moment	0.014034
Rear Axle Downforce	-0.014034
Front Axle Downforce Mean	-0.085708
Cp	-0.097498
signedDistance	-0.111285
Cl	-0.118014
CLA	-0.118014
Raw Downforce	-0.118014
CLA Mean	-0.127245
Rear Axle Downforce Mean	-0.149085
Rear Wheel Axle Moment	-0.173515
Front Axle Downforce	-0.173515
Cd	-0.240174
Raw Drag	-0.240174
CdA	-0.240174
Raw Drag Mean	-0.253116
CdA Mean	-0.253116
Frontal Area	-0.292447
Mass Flow of Radiator	-0.556718
Radiator dP	-0.663671
Front Rideheight	-0.940494

Values closer to 1.0 have a stronger correlation to downforce and positive and negative denote directly vs inversely proportional respectively. Here chassis heave is skewed more than it should be since there was only one heave used in this sweep. Lastly, we could take this matrix and plot

it using a scatter matrix though for our purposes not much can be seen from the plot over the numbers.

## Appendex

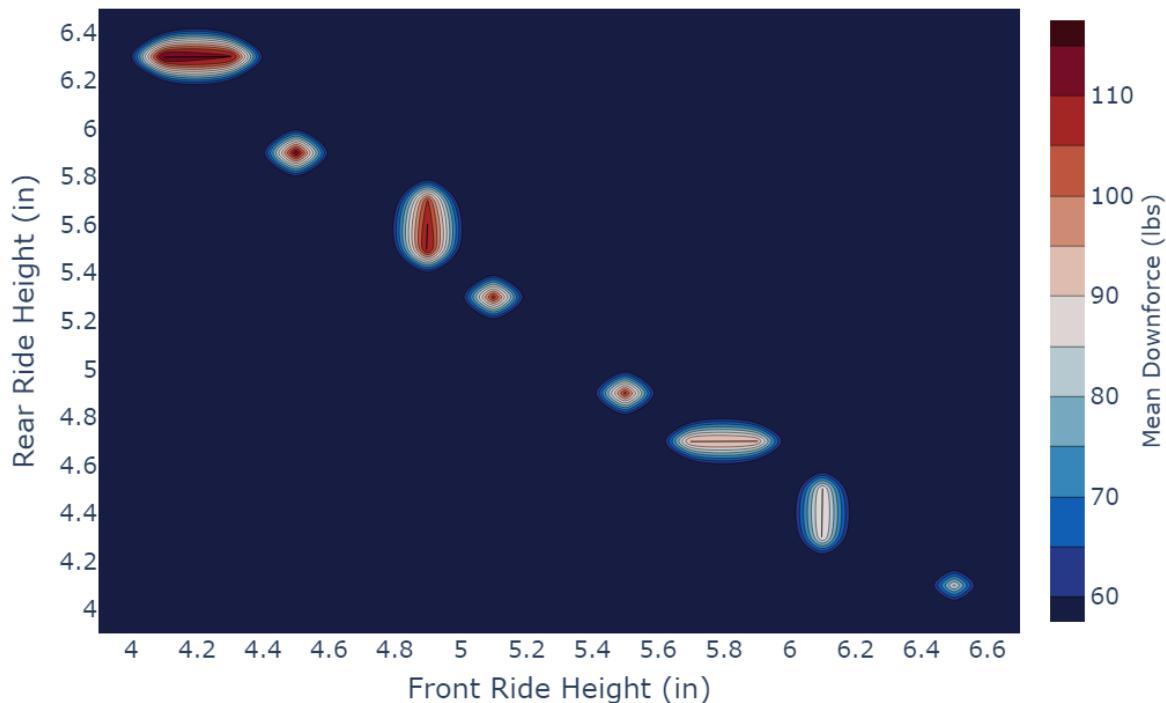
Github repo: <https://github.com/bellh14/DFR-Aeromap>

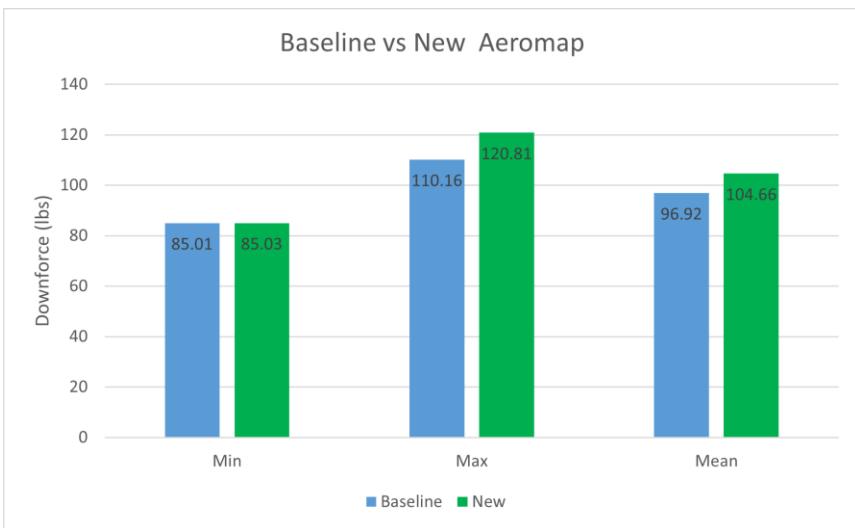
If you want some experience with data analysis or machine learning, hit me up.

Note have some massive reworks planned to move this to a website with plotly's react integration or a Gui since we need a more efficient method of accessing data from every study and especially handling the real world data (hopefully though we know how that be) we will have.

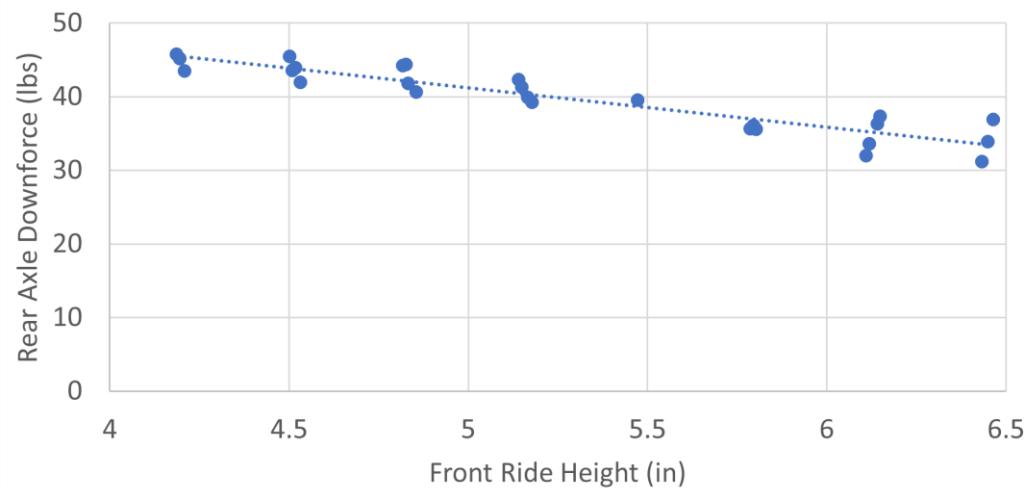
2024 V1

Ride Height vs Downforce



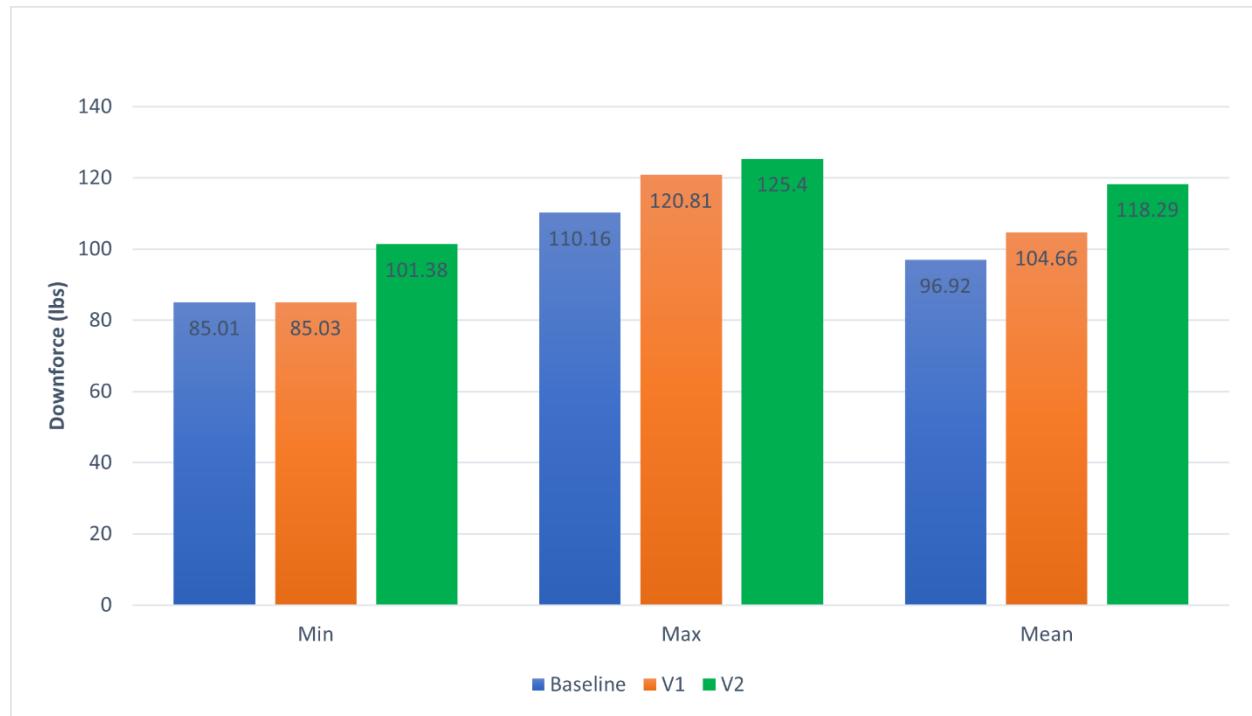
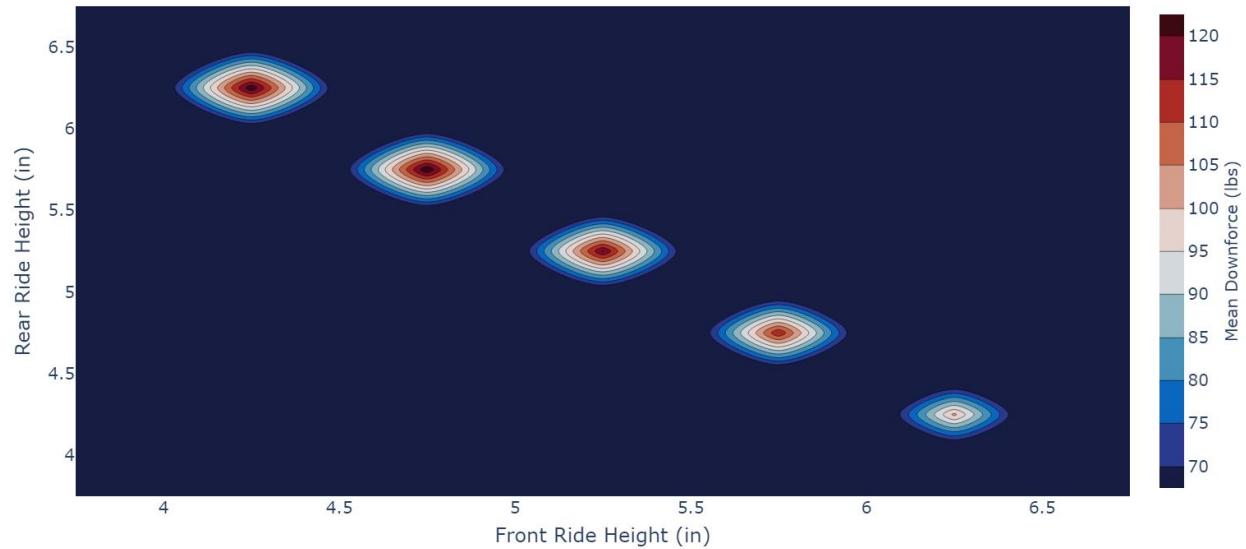


### Front Ride Height vs Rear Axle Downforce

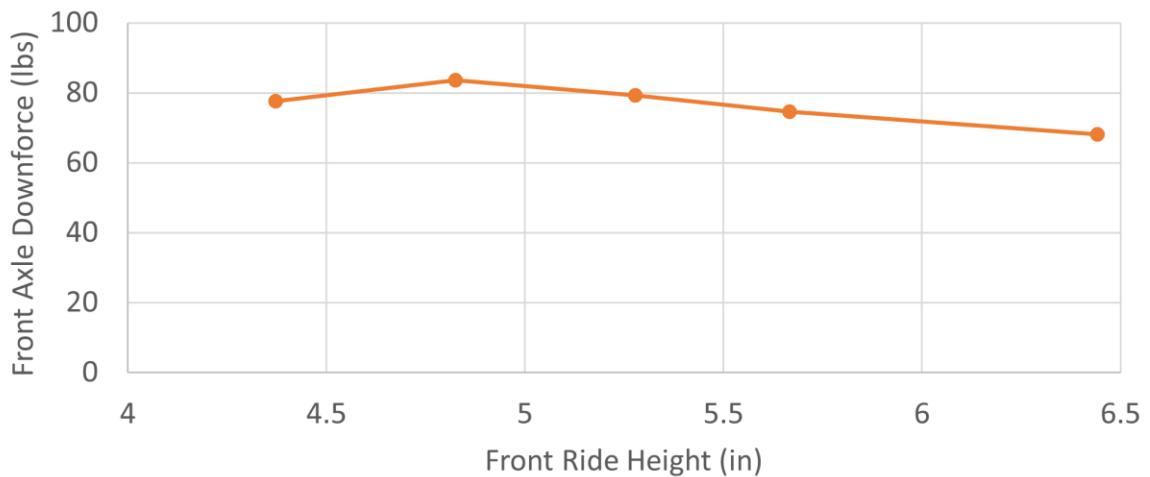


## 2024 V2

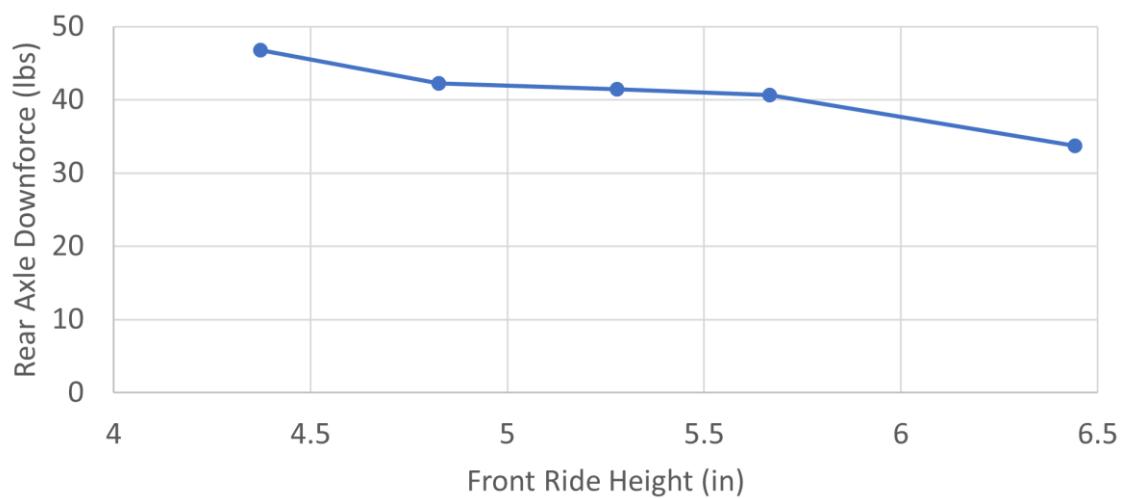
Ride Height vs Downforce



### Front Ride Height vs Front Axle Downforce

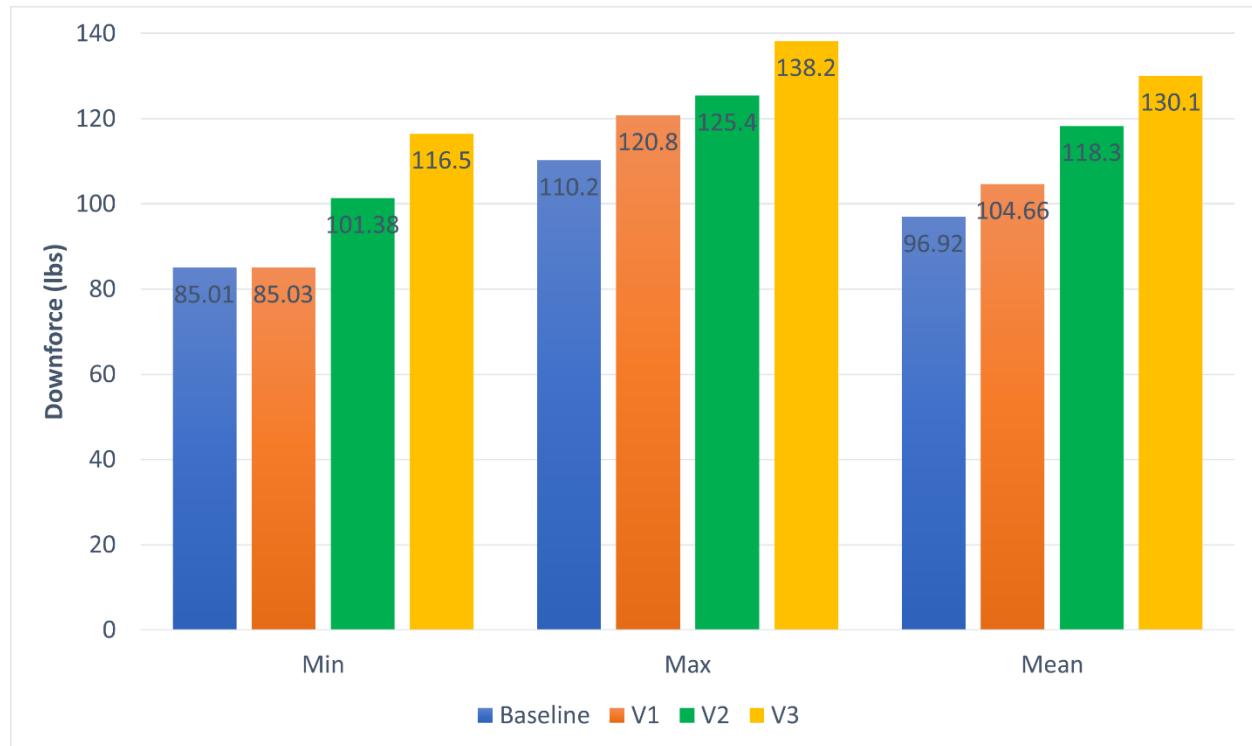
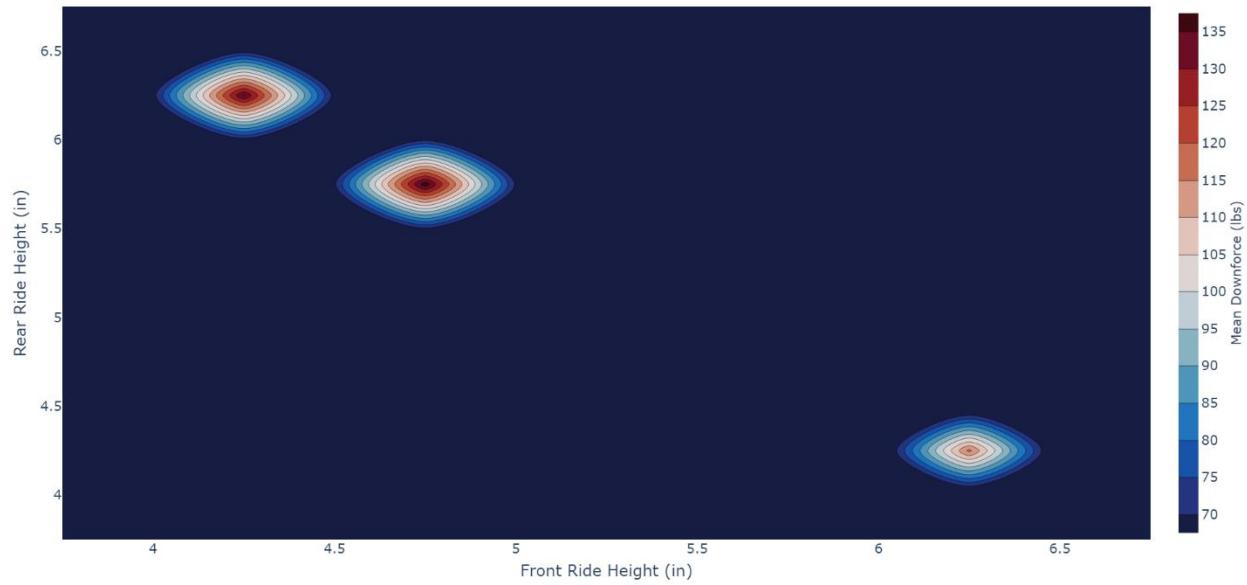


### Front Ride Height vs Rear Axle Downforce

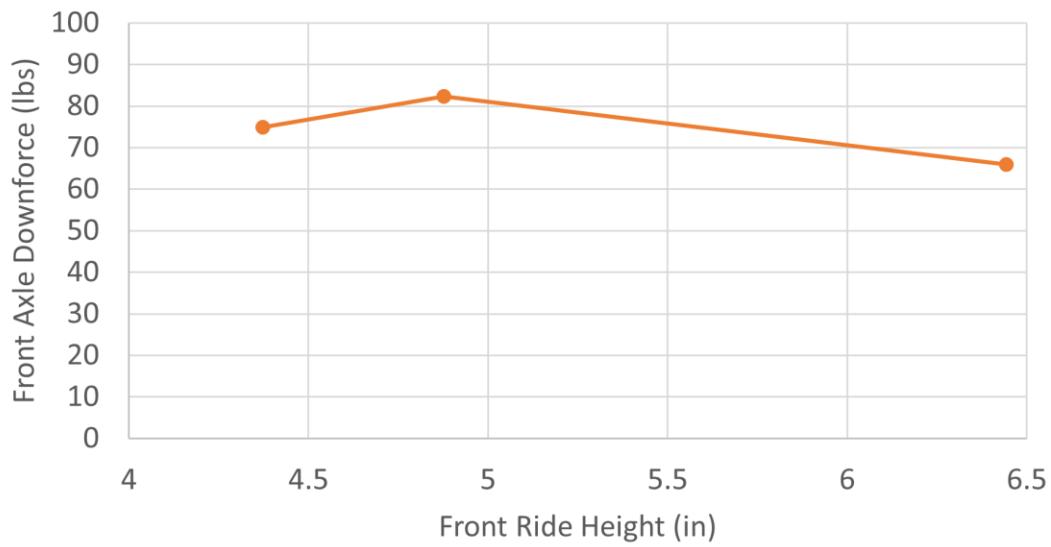


## 2024 V3

Ride Height vs Downforce



FRH vs. Front Axle Downforce Mean



FRH vs. Rear Axle Downforce Mean

