Aerodynamics of Tesla Cybertruck using OpenFOAM

Arpit Arvind Jain

Mechanical Engineering, Maulana Azad National Institute of Technology, Bhopal (India)

Abstract

This Case study aims to check the design of Cybertruck from Tesla from the aerodynamic point of view using OpenFOAM CFD Simulation. The Fuel economy of any pickup is directly dependent on the Drag force created by the airflow over it. Conventional Pickup trucks like RAM 1500 & Ford F150 have a Drag Coefficient of 0.357 and 0.40 [8]. Elon Musk, CEO of Tesla Motors Inc, has claimed that the new Cybertruck can achieve a drag Coefficient of 0.30, though it has a box-like design and a flat front. The drag coefficient will be calculated in this Case study. The geometry used is the original design of the vehicle. The structured mesh is generated using blockMesh and snappyHexMesh. Plots for the coefficients of drag are evaluated for different velocities.

Keywords: Cybertruck Aerodynamics, Drag Coefficient, CFD, OpenFOAM

1. Introduction

The Tesla Cybertruck is an all-electric light-duty truck announced by Tesla at the Tesla Design Studio in Los Angeles on 21 November 2019. There are three models, with range estimates of 400–800 km and an estimated 0–100 km/h time of 2.9–6.5 seconds, depending on the model. Elon Musk, CEO of Tesla Motors Inc tweeted "With extreme effort, Cybertruck might hit a 0.30 drag coefficient, which would be insane for a truck. Requires tweaking many small details.".

Figure 1: *Cybertruck*

2. Problem Statement

To obtain the Drag Coefficient of flow over Tesla Cybertruck by CFD simulation by turbulence modelling for speed range of 18 – 27 m/s (40 – 60 mph). Studying the flow over the vehicle. Plotting the coefficient of drag which would be extracted from the Results. A streamline of the flow and a suitable animation would be made.

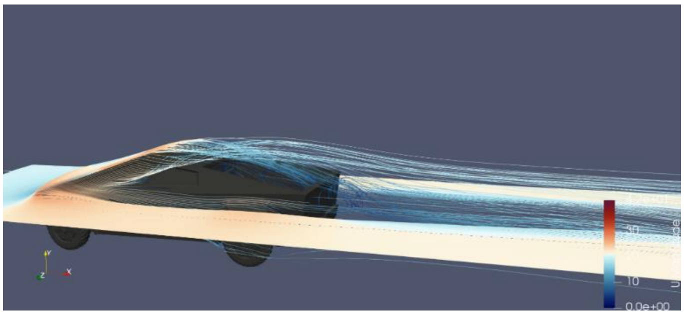


Figure 2: *Flow around the Cybertruck preview*

3. Governing Equations

3.1 RANS Equation

They case study is based on Reynolds-averaged Navier–Stokes's equations (or RANS equations). The Reynolds-averaged Navier–Stokes's equations (or RANS equations) are time-averaged equations of motion for fluid flow. The RANS equations are primarily used to describe turbulent flows. These equations can be used with approximations based on knowledge of the properties of flow turbulence to give approximate time-averaged solutions to the Navier–Stokes's equations. For a stationary flow of an incompressible Newtonian fluid, these equations can be written as [1]:

$$\rho \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \rho \bar{f}_i + \frac{\partial}{\partial x_j}\left[-\bar{p}\delta_{ij} + \mu\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \rho u'_i u'_j\right]$$

3.2 Bernoulli's Equation

Bernoulli's equation defines the various pressure and velocity distribution contours around the vehicle. It is given as:

$$P + \frac{1}{2}\rho v^2 + \rho g h = Constant$$

4. Simulation Procedure

4.1 Geometry and Mesh

The Geometry was retrieved from GrabCAD [2].

The geometry was rescaled to actual dimensions [3]:

Length: 5,885 mm

Width: 2,027 mm

Height: 1,905 mm

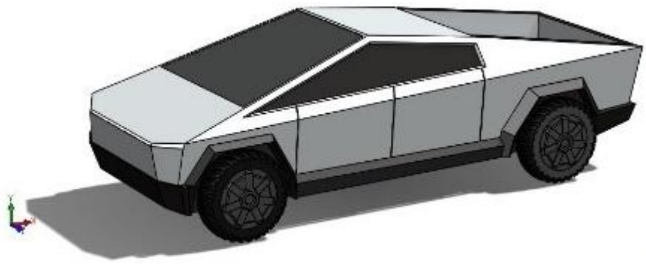


Figure 3: *CAD model of Tesla Cybertruck*

The 3D Mesh generation for the cuboidal domain was done using the blockMesh utility. The snappyHexMesh utility was then used to snap out the geometry and refined the mesh adjacent to the surface with accuracy.

4.1.1 BlockMesh

The Cubical Domain was made using BlockMesh of Dimension:

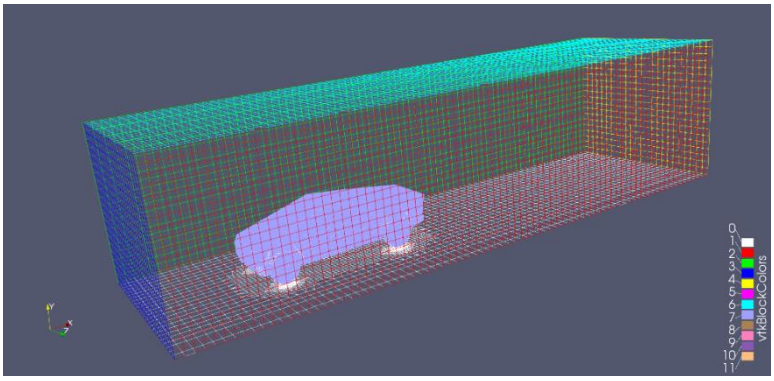


Figure 4: *Domain Box*

BlockMesh	
Bounding Box	(-3 -0.4 -3)
	(17 4.6 3)
Divisions	(80 20 24)

4.1.2 SnappyHexMesh

Before proceeding, the 'surfaceFeatures' command was used to extract a surface definition file(.emesh). The surface definition file(.emesh) is used in snappyHexMesh as a definition of the geometry. The SnappyHexMesh file is defined as:

snappyHexMeshDict	
Edge Refinement Level	4
Surface Refinement Level	(4 4)
No. of Layers	0

checkMesh results	
Domain	(-3 -0.4 -3) (17 4.6 3)
Points	886823
faces	2231844
internal faces	2032139
cells	680042
faces per cell	6.28488
boundary patches	7
Max Skewness	3.21809
Max Aspect ratio	14.8771

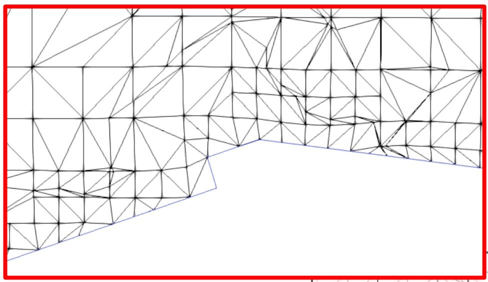


Figure 6: *Cross-sectional mesh closeup*

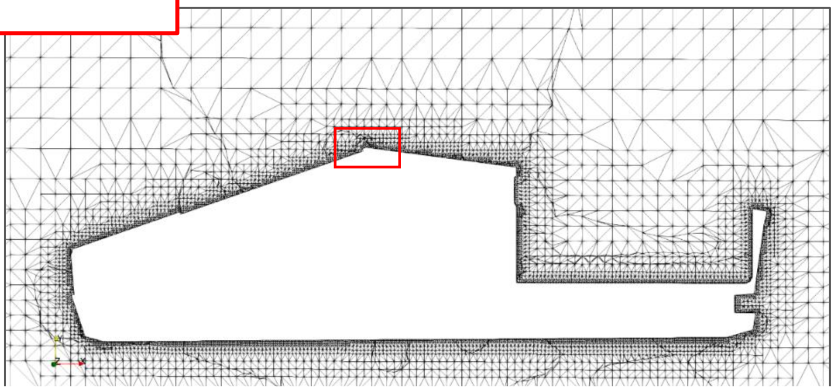


Figure 5: *Cross-section Mesh around Tesla Cybertruck after SnappyHexMesh*

4.2 Initial and Boundary Conditions

Airflow enters with 20 m/s of fixed velocity from the inlet patch, close to 72 km/hr. The outlet is inletOutlet, as flow velocity 20 m/s. The pressure outlet is assigned a fixed value of 0 atm. The front, back, upper and lower patches were defined as No-slip walls. On the Cybertruck, velocity is no-slip and, pressure is zeroGradient.

List of abbreviations used in the list:

1) FV: Fixed Value
2) ZG: Zero Gradient
3) IO: Inlet Outlet

Boundary	U	P
Inlet	FV (20)	ZG
Outlet	IO	FV (0)
FrontAndBack	noSlip	ZG
LowerWall	noSlip	ZG
UpperWall	noSlip	ZG
Tesla	noSlip	ZG

4.3 Turbulence model

The kOmegaSST turbulence model of OpenFOAM is used in this simulation. This model is a combination of k-epsilon (k-ε) and k-omega (k-ω) turbulence model. The following formulae were used in the calculations:

1. Reynolds Number

$$Re = \frac{uL}{\nu} = \frac{\rho u L}{\mu}$$

where,

u - Maximum velocity of the object relative to the fluid

L - Characteristic Linear dimension

ν - Kinematic Viscosity

2. Turbulent Energy

$$k = \frac{3}{2}(UI)^2$$

where,

U – Mean Flow Velocity

I – Turbulent Intensity

3. Specific Turbulent Dissipation Rate

$$\omega = \frac{\sqrt{k}}{l}$$

where,

k - Turbulent Energy

l – Turbulent Length Scale

Boundary	k	omega	nut
Inlet	FV (0.24)	FV(1.78)	calculated
Outlet	IO	IO	calculated
FrontAndBack	kqrWallFunction	omegaWallFunction	nutkWallFunction
LowerWall	kqrWallFunction	omegaWallFunction	nutkWallFunction
UpperWall	kqrWallFunction	omegaWallFunction	nutkWallFunction
Tesla	kqrWallFunction	omegaWallFunction	nutkWallFunction

4.4 Solver

PimpleFoam solver of OpenFOAM was used for this simulation. PimpleFoam is a transient solver for incompressible turbulent flow. PIMPLE algorithm is a combination of the PISO (Pressure Implicit with Splitting of Operator) and SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithm [4]. The controlDict file, which controls the solution time steps and duration, was defined as below:

```
FoamFile
{
   version     2.0;
   format      ascii;
   class       dictionary;
   object      controlDict;
}
application     pimpleFoam;

startFrom       latestTime;

startTime       0;

stopAt          endTime;

endTime         1;

deltaT          5e-5;

writeControl    timeStep;

writeInterval   250;

purgeWrite      0;

writeFormat     binary;

writePrecision  6;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

graphFormat gnuplot;

functions
{
   #include "forceCoeffs"
}
```

The deltaT was 5e-5, and the simulation ran for real-time of 1 second. It also consists of the forceCoeff function, which is discussed later in this report.

5. **R**esults

The simulation was done to study the Aerodynamic characteristics of flow over Tesla Cybertruck. This was done for 4 cases with velocity ranging from 18 to 27 m/s which is 64.8 - 97.2 km/hr. The following characteristics or parameters were calculated and visualized using ParaView:

5.1 Pressure and Velocity Contours

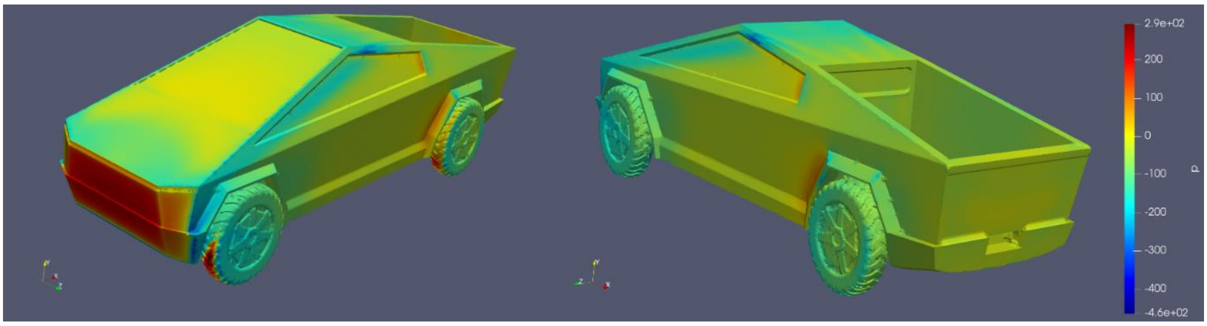


Figure 7: *Pressure contours on the Cybertruck*

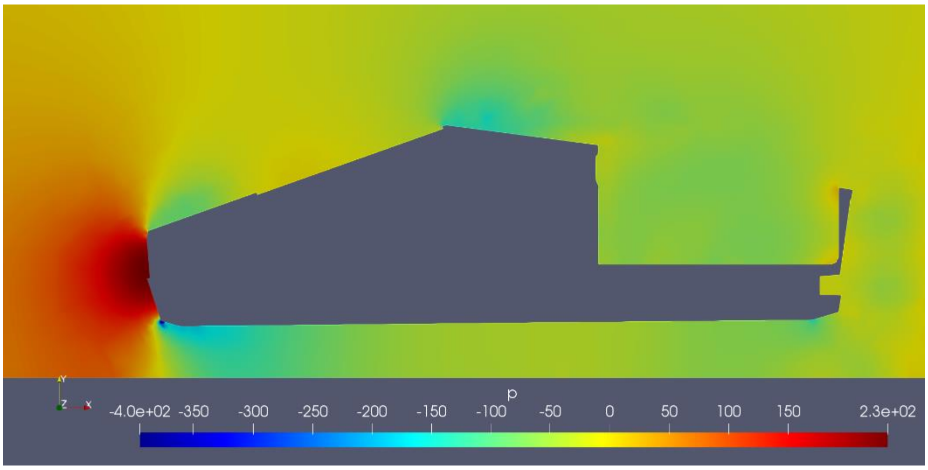
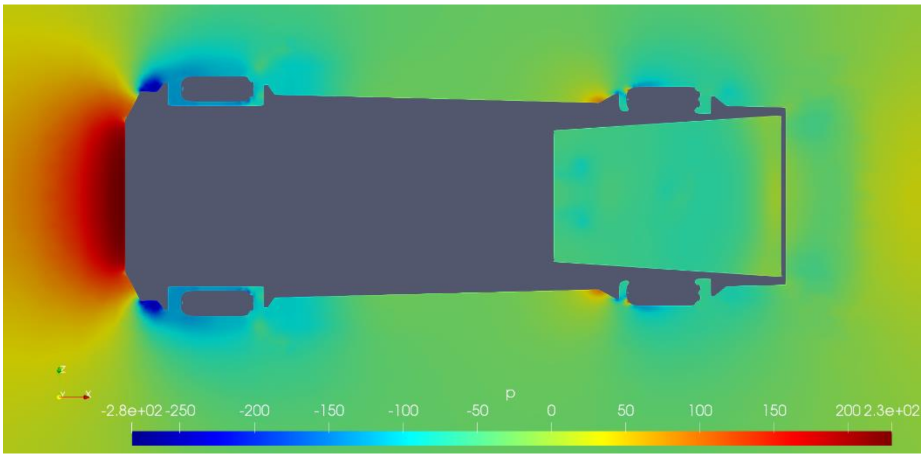


Figure 8: *Pressure contours of vertical cut-section of flow around the Cybertruck*

Figure 9: *Pressure contours of horizontal cut-section of flow around the Cybertruck*



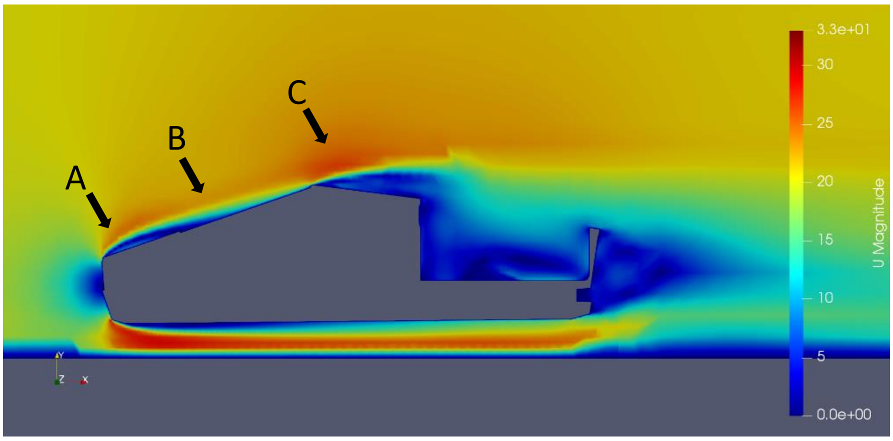


Figure 10: *Velocity contours of vertical cut-section of flow around the Cybertruck. A and C are locations of flow separation, B is the location of flow reattachment*

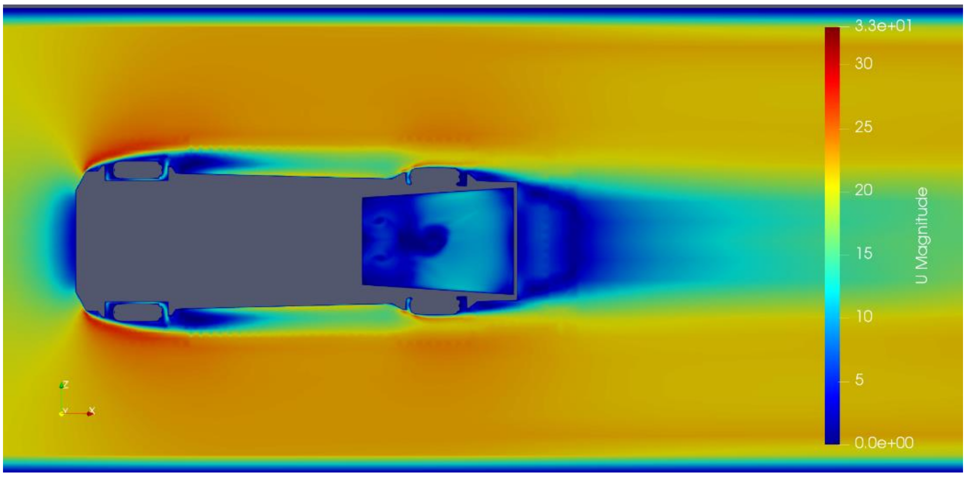


Figure 11*: Velocity contours of vertical cut-section of flow around the Cybertruck*

5.2 Streamlines & Vector Streamlines with arrows

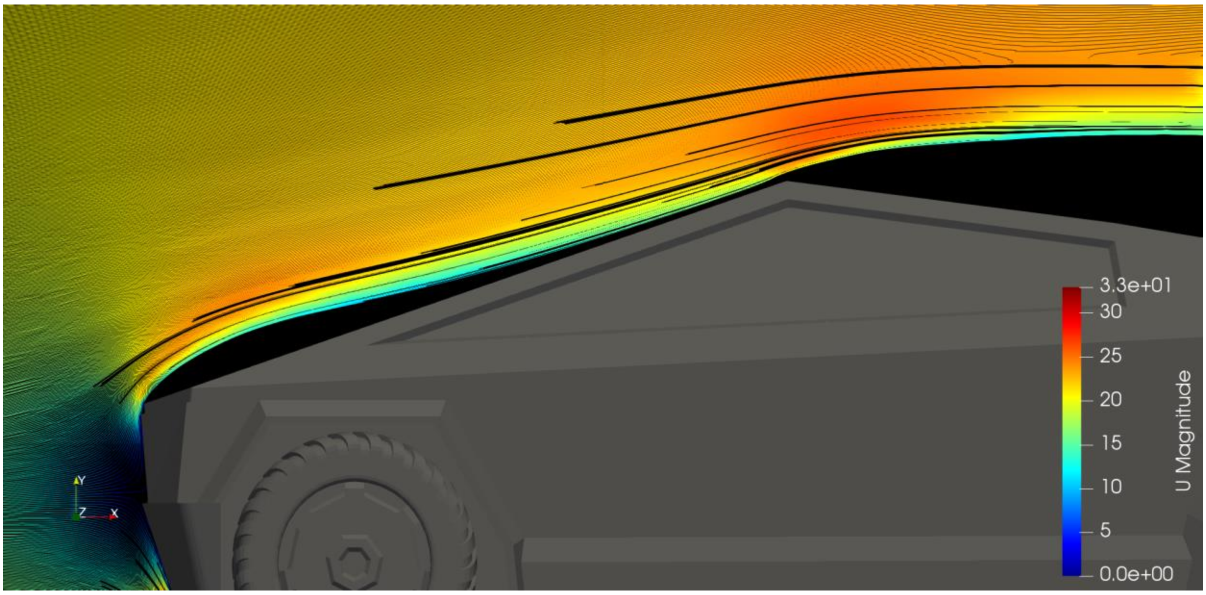
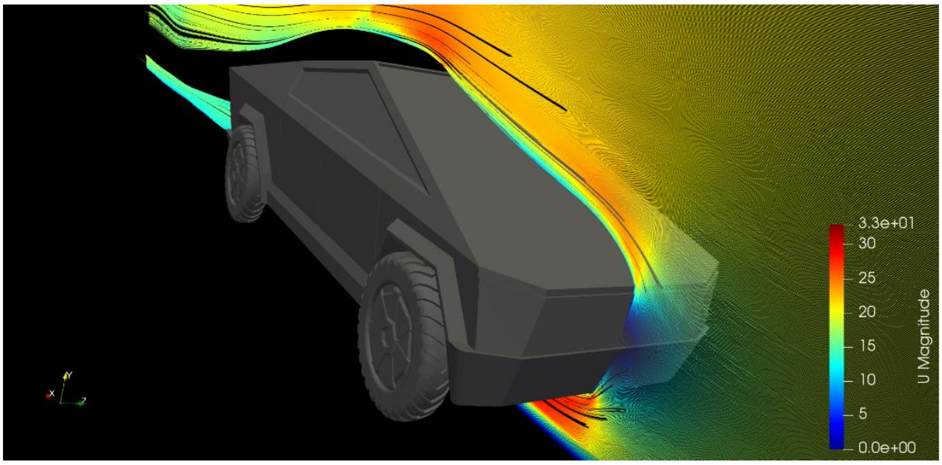


Figure 12: *Velocity Streamlines of vertical plane flow over the Cybertruck (2D view)*

Figure 13: *Velocity Streamlines of vertical plane flow over the Cybertruck (Oblique view)*



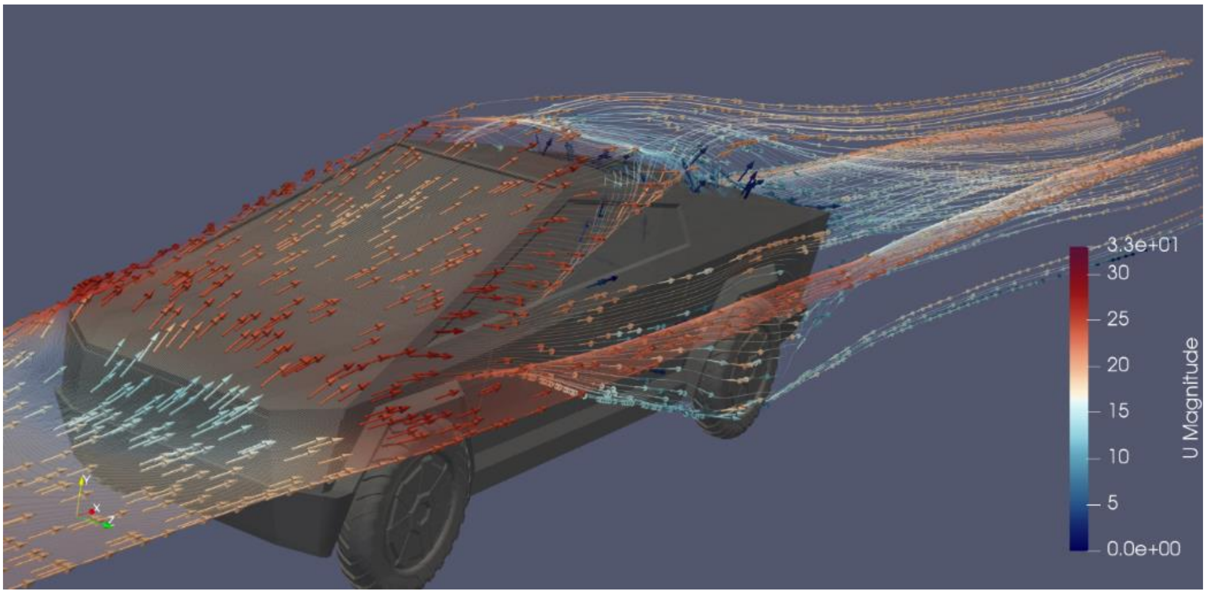


Figure 14: *Vector streamline of flow around the Cybertruck with a line source (Front view)*

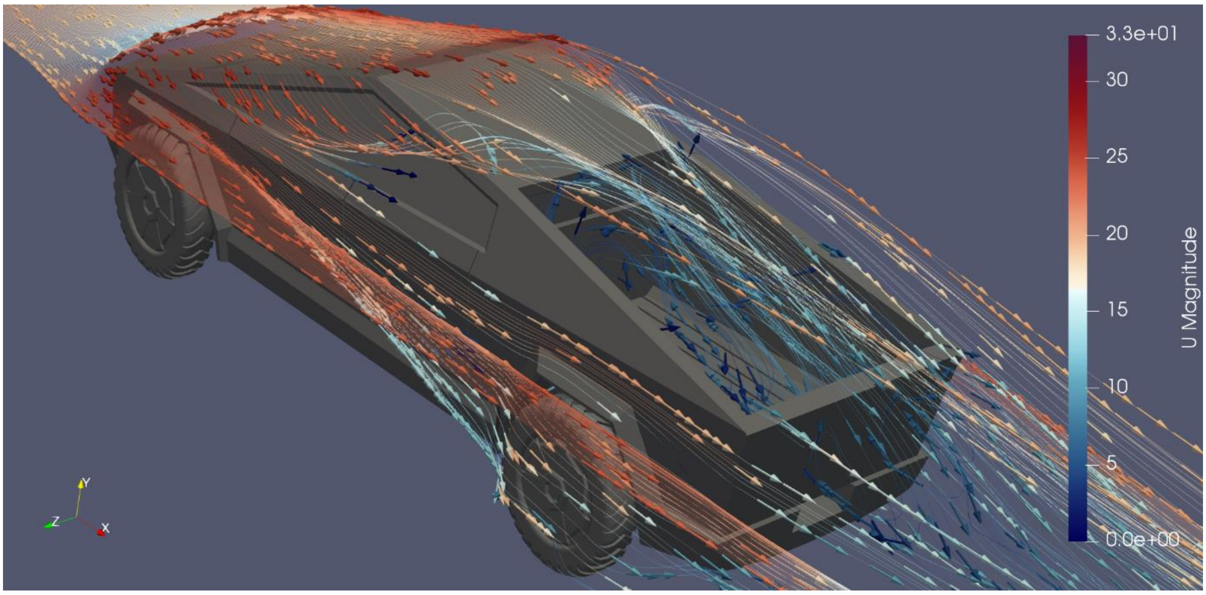


Figure 15: *Vector streamline of flow around the Cybertruck with a line source (Rearview)*

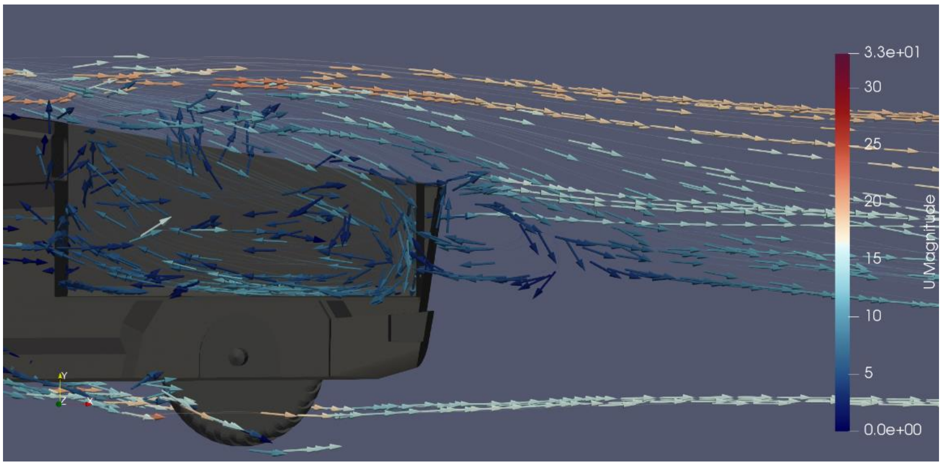
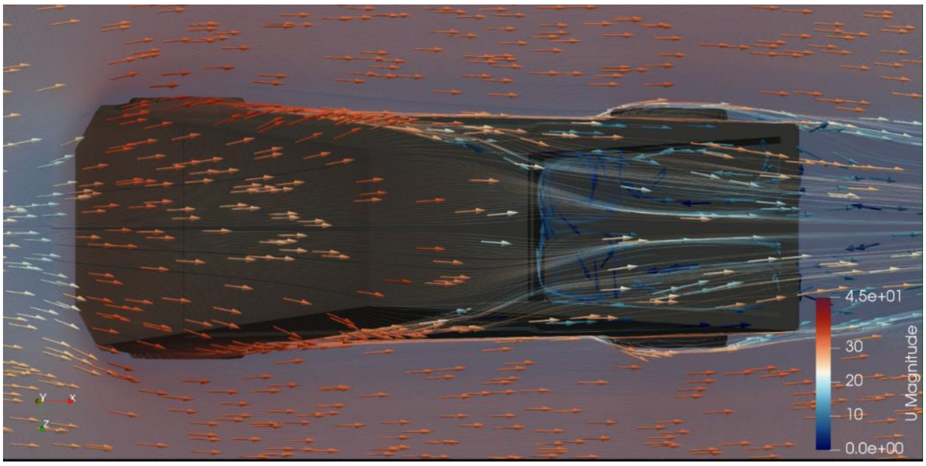


Figure 16: *Vector streamline of flow showing the vortices after flow separation with vehicle cut-section*

Figure 17: *Vector streamline of flow showing the disturbed vortices at the rear section of the vehicle (Top view)*



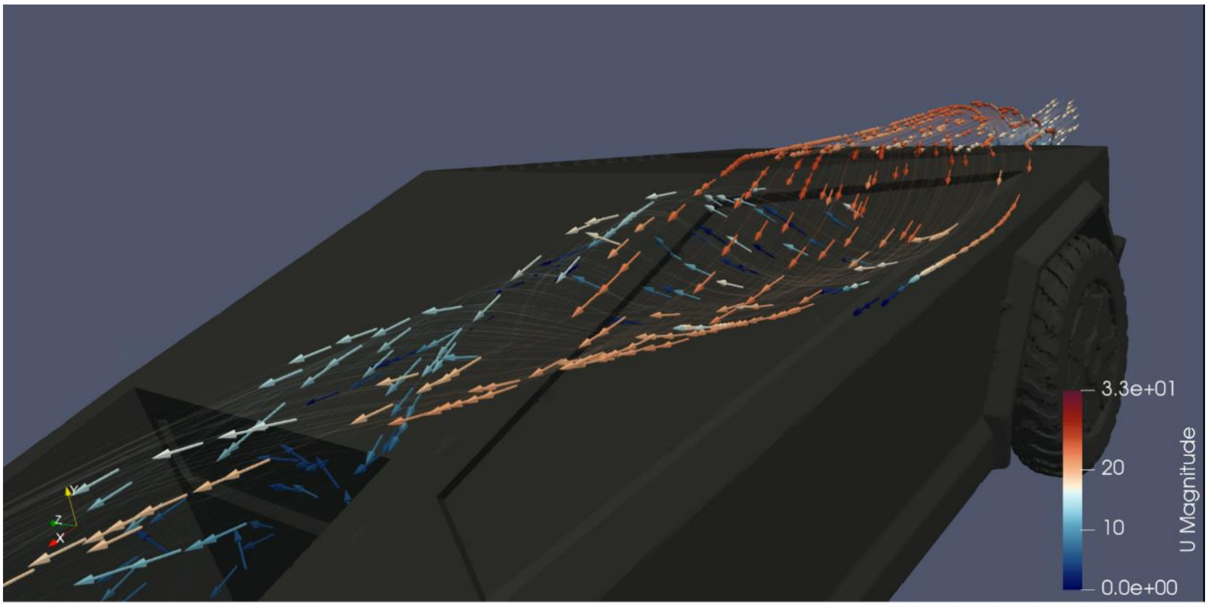


Figure 18: *The Slanted upper front and rear surfaces make this vortex flow near the A-pillars, which is then pulled inwards at the rear of the vehicle* [7]

5.3 Zone clouds

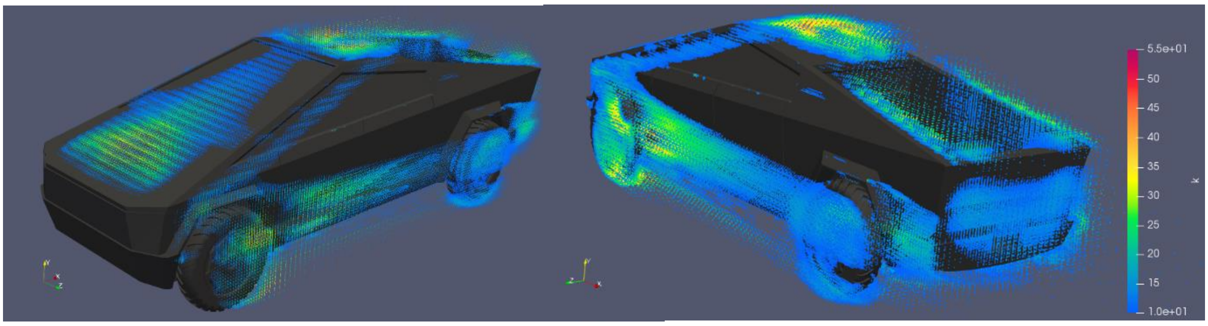


Figure 19: *Turbulence zone clouds (k)*

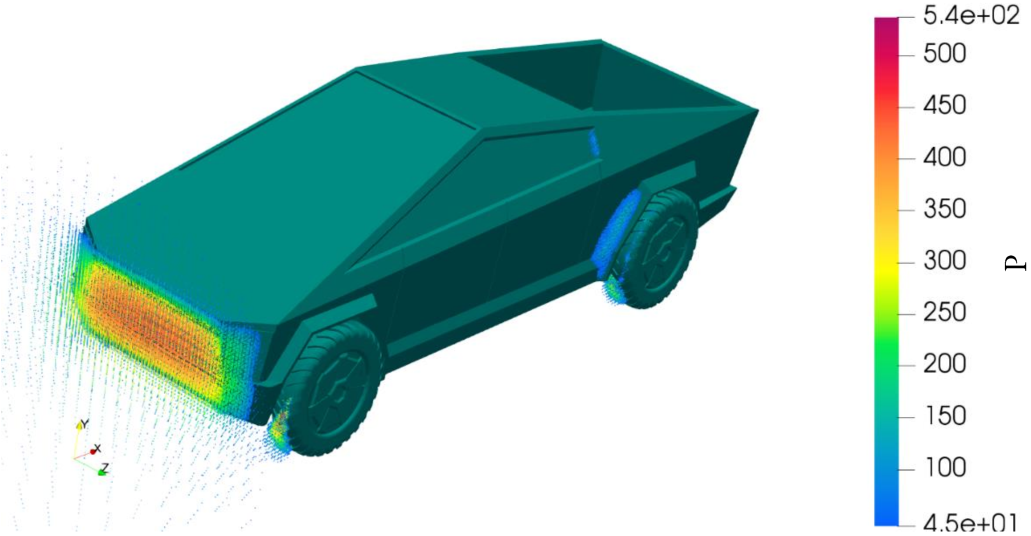
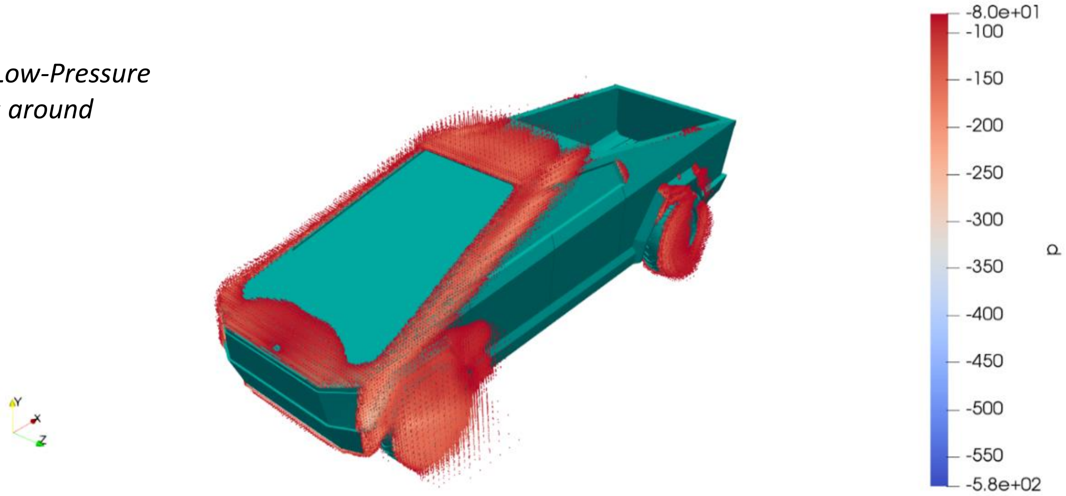


Figure 20: *High-Pressure zone clouds around Cybertruck*

Figure 21: *Low-Pressure zone clouds around Cybertruck*



5.4 Forces and Drag Coefficient (Cd)

The Drag coefficient is calculated using the following formula:

$$C_d = \frac{F_d}{{}^{1}\!/_{2}\,\rho u^2 A}$$

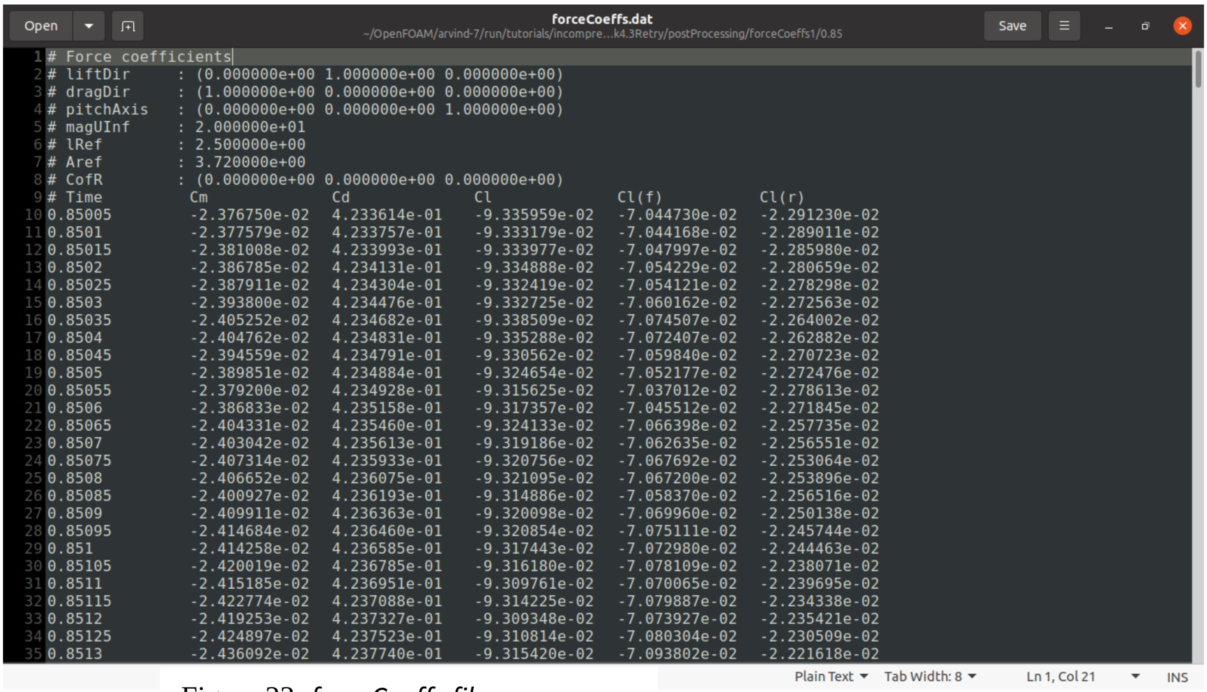
In this case study, the Drag coefficient was calculated by 2 different methods. Both methods gave results in a close range. The methods are as follows:

1) forceCoeffs function utility:

The function was defined and stored in the system folder. Parameters like frontal area, flow velocity and direction were input. The outputs are Cd, Cl and Cm. The function was as follows:

```
forceCoeffs1
{
    type          forceCoeffs;

    libs          ("libforces.so");

    writeControl    timeStep;
    timeInterval    10;

    log           yes;

    patches        (tesla);
    rho            rhoInf;     // Indicates incompressible
    rhoInf         1.225;         // Redundant for incompressible
    liftDir        (0 1 0);
    dragDir        (1 0 0);
    CofR           (0 0 0);  // Axle midpoint on ground
    pitchAxis      (0 0 1);
    magUInf        20;
    lRef           2.5;       // Wheelbase length
    Aref           3.72;       // Estimated
}
```

And the output was after every iteration while the solver ran, and it was:



```
# Force coefficients
# liftDir      : (0.000000e+00 1.000000e+00 0.000000e+00)
# dragDir      : (1.000000e+00 0.000000e+00 0.000000e+00)
# pitchAxis    : (0.000000e+00 0.000000e+00 1.000000e+00)
# magUInf      : 2.000000e+01
# lRef         : 2.500000e+00
# Aref         : 3.720000e+00
# CofR         : (0.000000e+00 0.000000e+00 0.000000e+00)
# Time          Cm              Cd              Cl              Cl(f)           Cl(r)
0.85005         -2.376750e-02   4.233614e-01    -9.335959e-02   -7.044730e-02   -2.291230e-02
0.8501          -2.377579e-02   4.233757e-01    -9.333179e-02   -7.044168e-02   -2.289011e-02
0.85015         -2.381008e-02   4.233993e-01    -9.333977e-02   -7.047997e-02   -2.285980e-02
0.8502          -2.386785e-02   4.234131e-01    -9.334888e-02   -7.054229e-02   -2.280659e-02
0.85025         -2.387911e-02   4.234304e-01    -9.332419e-02   -7.054121e-02   -2.278298e-02
0.8503          -2.393800e-02   4.234476e-01    -9.332725e-02   -7.060162e-02   -2.272563e-02
0.85035         -2.405252e-02   4.234682e-01    -9.338509e-02   -7.074507e-02   -2.264002e-02
0.8504          -2.404762e-02   4.234831e-01    -9.335288e-02   -7.072407e-02   -2.262882e-02
0.85045         -2.394559e-02   4.234791e-01    -9.330562e-02   -7.059840e-02   -2.270723e-02
0.8505          -2.389851e-02   4.234884e-01    -9.324654e-02   -7.052177e-02   -2.272476e-02
0.85055         -2.379200e-02   4.234928e-01    -9.315625e-02   -7.037012e-02   -2.278613e-02
0.8506          -2.386833e-02   4.235158e-01    -9.317357e-02   -7.045512e-02   -2.271845e-02
0.85065         -2.404331e-02   4.235460e-01    -9.324133e-02   -7.066398e-02   -2.257735e-02
0.8507          -2.403042e-02   4.235613e-01    -9.319186e-02   -7.062635e-02   -2.256551e-02
0.85075         -2.407314e-02   4.235933e-01    -9.320756e-02   -7.067692e-02   -2.253064e-02
0.8508          -2.406652e-02   4.236075e-01    -9.321095e-02   -7.067200e-02   -2.253896e-02
0.85085         -2.400927e-02   4.236193e-01    -9.314886e-02   -7.058370e-02   -2.256516e-02
0.8509          -2.409911e-02   4.236363e-01    -9.320098e-02   -7.069960e-02   -2.250138e-02
0.85095         -2.414684e-02   4.236460e-01    -9.320854e-02   -7.075111e-02   -2.245744e-02
0.851           -2.414258e-02   4.236585e-01    -9.317443e-02   -7.072980e-02   -2.244463e-02
0.85105         -2.420019e-02   4.236785e-01    -9.316180e-02   -7.078109e-02   -2.238071e-02
0.8511          -2.415185e-02   4.236951e-01    -9.309761e-02   -7.070065e-02   -2.239695e-02
0.85115         -2.422774e-02   4.237088e-01    -9.314225e-02   -7.079887e-02   -2.234338e-02
0.8512          -2.419253e-02   4.237327e-01    -9.309348e-02   -7.073927e-02   -2.235421e-02
0.85125         -2.424897e-02   4.237523e-01    -9.310814e-02   -7.080304e-02   -2.230509e-02
0.8513          -2.436092e-02   4.237740e-01    -9.315420e-02   -7.093802e-02   -2.221618e-02
```


Figure 22: *forceCoeffs file*

2) Using ParaView calculator Utility:

The Pressure over the Cybertruck (Fig. 7) integrated for all axes using the *IntergrateVariable* function in ParaView gave an output of pressure forces in all 3-axis directions. The drag coefficient equation was input in the calculator of ParaView. Pressure force was used, which is towards the axis of flow. A preview of the method is as shown below:

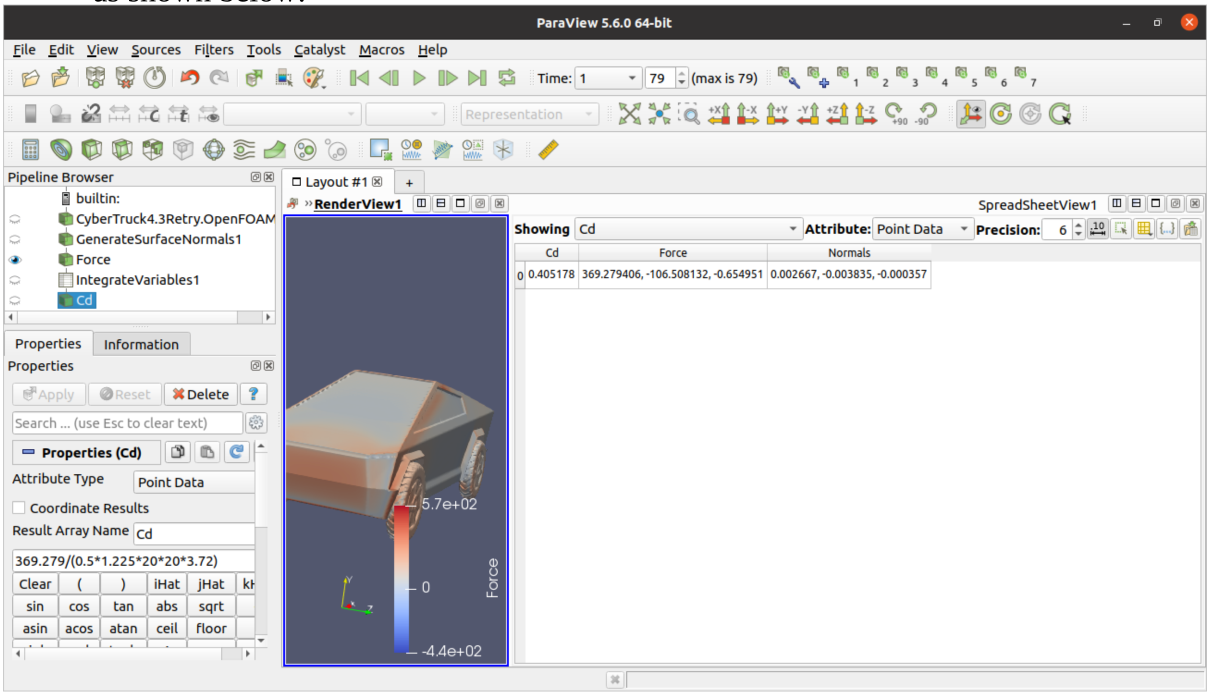


Figure 23: *ParaView preview with results*

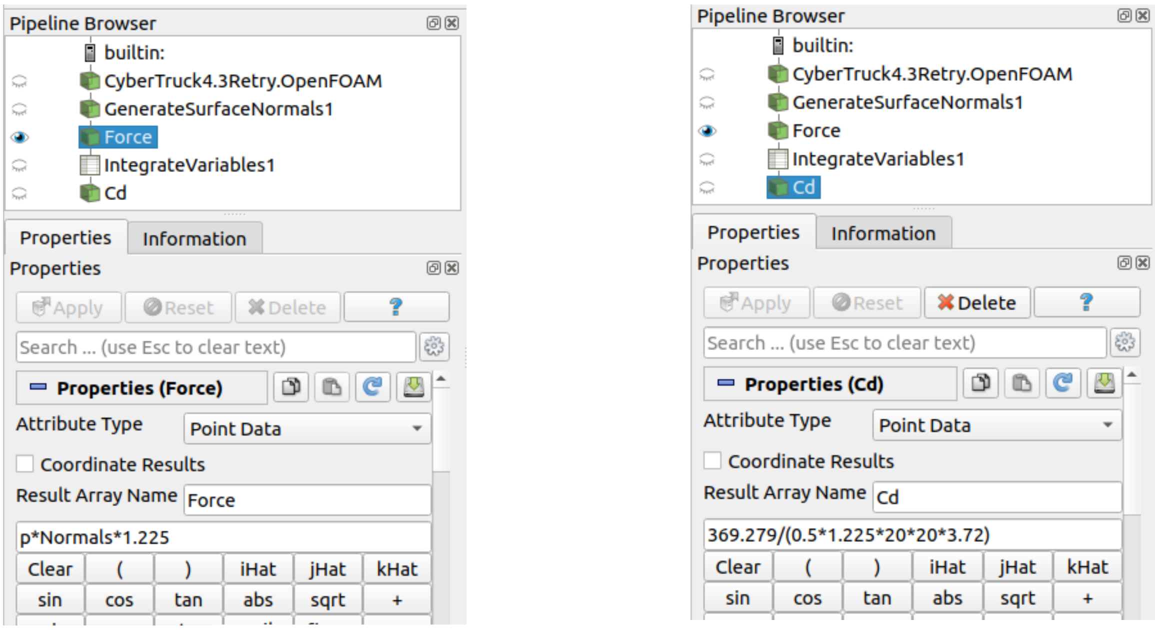


Figure 24: *ParaView calculator with Force and Cd*

The Final Results from both methods were as follows:

Forces - Wind Tunnel coordinate system					
V = 18 m/s	Fx	Fy	Fz	Cd	Cd using forceCoeff
Pressure Force	300.396 N	-79.946 N	2.4394 N	**0.407**	**0.432**

Forces - Wind Tunnel coordinate system					
V = 20 m/s	Fx	Fy	Fz	Cd	Cd using forceCoeff
Pressure Force	369.279 N	-106.508 N	0.605 N	**0.405**	**0.431**

Forces - Wind Tunnel coordinate system					
V = 24 m/s	Fx	Fy	Fz	Cd	Cd using forceCoeff
Pressure Force	516.307 N	-158.867 N	-1.601 N	**0.393**	**0.418**

Forces - Wind Tunnel coordinate system					
V = 27 m/s	Fx	Fy	Fz	Cd	Cd using forceCoeff
Pressure Force	638.667 N	-180.603 N	4.773 N	**0.3845**	**0.409**

lift Coefficients were calculated for v = 20 m/s:

Cl	Cl using forceCoeff
-0.116	**-0.114**

The results were plotted using MATLAB:

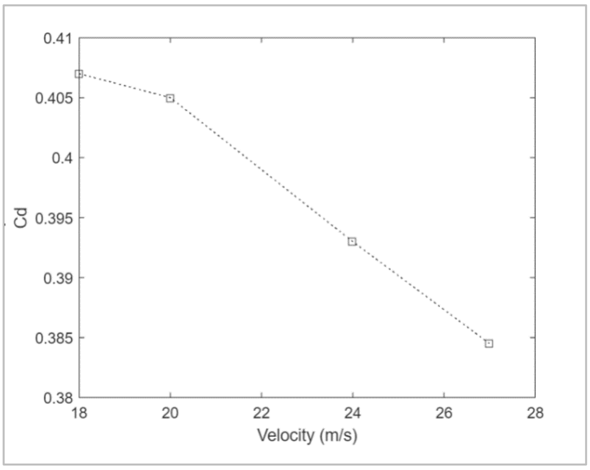


Figure 25: Cd vs Velocity plot

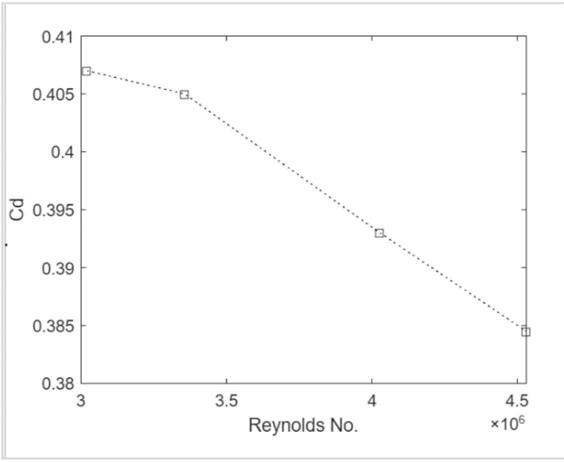


Figure 26: Cd vs Reynolds number plot

5.5 Residuals and Cd converging graph

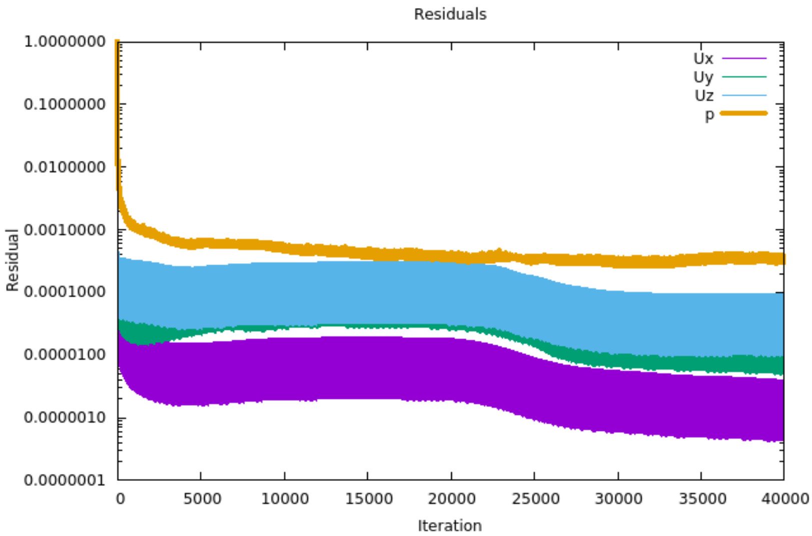


Figure 27: *Residuals Plot*

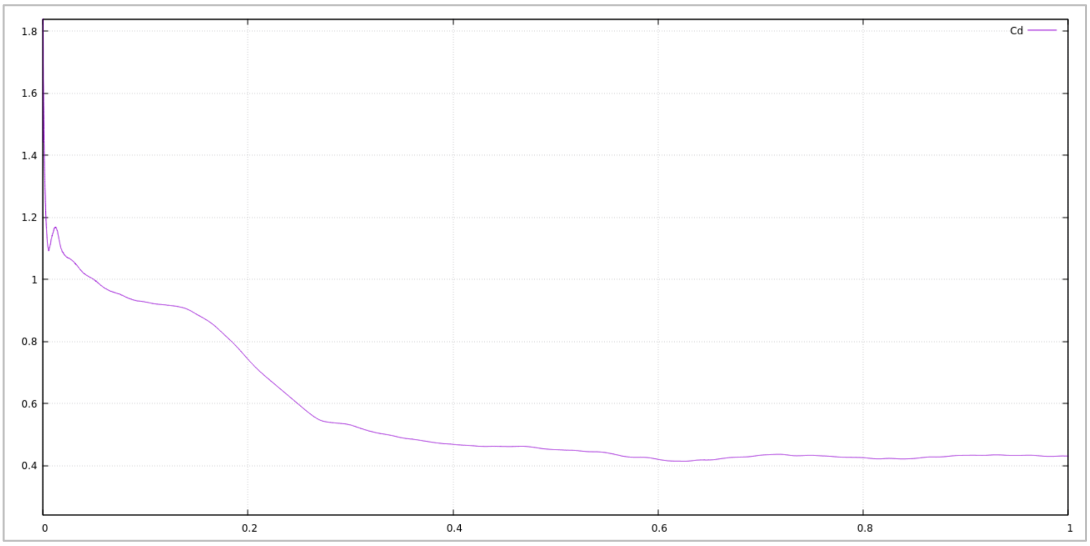


Figure 28: *Cd vs Iteration time*

5. Conclusions

- The Drag Coefficient is calculated and found to be around 0.40. That is close to the results obtained from reference [5] [6]. Though it could be reduced by many strategies and tweaks, as tweeted by Elon Musk, CEO of Tesla Motors Inc.
- The flow separating at the front edge of the bonnet and roof edge as seen in Fig. 10 at locations A and C. These flow separations break the streamline flow over the vehicle and contribute towards the drag.
- The CAD model used has an open trunk that simulates most case scenarios of travel on highways. Whether the trunk is open or close, in both cases, the drag contribution of the rear section by negative pressure is negligibly varying as the flow has separated from the roof edge as seen in Fig. 12.
- There are Air swirls produced by A-pillars, at the side windows [7]. The flow ahead, in most cases, creates vortexes. These are pulled towards the middle and absorbed by the trunk. Hence there are no vortexes seen in the flow aft to the vehicle. This can be seen in Fig. 15 & Fig. 18.
- Considering the two conclusions stated above, in a case where the trunk is closed, these vortexes could help the reattachment of the flow at the rear. Hence helping in reducing the drag coefficient even more and contributing towards a more streamlined flow [9].

6. Future Scope

- A similar study could be done on a Cybertruck geometry with a closed trunk to verify that the flow does reattach at the rear of the vehicle.
- Some new design considerations can be tested for, like the high-pressure zone at the front could be provided with a certain kind of intake or use NACA ducts on the sides to transfer the upcoming airflow to be redirected to bleed it out at locations like locations of flow separation, or low-pressure area. The airflow can also be used for air conditioning or cooling the battery pack and brakes. This kind of strategy is used in aircraft wings by the name of the leading-edge slot to prevent flow separations on the low-pressure zones. However, even vortex generators can be used at the roof edge to control the flow separation.

7. **R**eferences

[1] "Reynolds-Averaged Navier–Stokes Equations." *Wikipedia*, Wikimedia Foundation, 20 Mar. 2021, en.wikipedia.org/wiki/Reynolds-averaged_Navier%E2%80%93Stokes_equations.

[2] "Free CAD Designs, Files & 3D Models: The GrabCAD Community Library." *Free CAD Designs, Files & 3D Models | The GrabCAD Community Library*, www.grabcad.com/library/tag/cybertruck.

[3] "Tesla Cybertruck." *Wikipedia*, Wikimedia Foundation, 27 May 2021, en.wikipedia.org/wiki/Tesla_Cybertruck.

[4] 1318980. "CFD: PIMPLE Algorithm." *SimScale CAE Forum*, 6 Nov. 2020, www.simscale.com/forum/t/cfd-pimple-algorithm/81418.

[5] Ruffo, Gustavo Henrique. "Tesla Cybertruck Aerodynamic Studies Point To Cd Of 0.4: Musk Says 0.3." *InsideEVs*, InsideEVs, 3 Dec. 2019, insideevs.com/news/385856/tesla-cybertruck-aerodynamic-coefficient-03/.

[6] Remmerie, Wouter. "Is the Tesla Cybertruck Aerodynamic?" *AirShaper*, AirShaper, 2 Feb. 2021, airshaper.com/cases/tesla-cybertruck-aerodynamics.

[7] Joseph Katz, PhD. "An Introduction to Automobile Aerodynamics." *Mechanix Illustrated*, 6 Dec. 2018, mechanixillustrated.technicacuriosa.com/2017/03/04/an-introduction-to-automobile-aerodynamics/.

[8] Halvorson, B. (2019, December 5). "Tesla Cybertruck aerodynamics could be better than other trucks, suggests Musk." Green Car Reports. https://www.greencarreports.com/news/1126288_tesla-cybertruck-aerodynamics-could-be-better-than-other-trucks-suggests-musk.

[9] Dow, J., & Jameson Dow Jameson has been driving electric vehicles since 2009. (2019, November 25). "Are the boxy Tesla Cybertruck's aerodynamics any good? An engineer found out." Electrek. https://electrek.co/2019/11/25/tesla-cybertruck-aerodynamics-cfd-rendering/.