Multidisciplinary Design Optimisation of a Shell Eco-marathon Solar Car Through Coupled Aerodynamic and Solar Panel Performance Analysis

A dissertation submitted to The University of Manchester for the degree of

MEng Mechanical Engineering

in the Faculty of Science and Engineering

Amruth Parakkunnath

Student ID: 11130304

School: School of Engineering

Department: Department of Mechanical and Aerospace Engineering

Year of Submission: 2025

Supervisor: Dr Alessandro De Rosis

Abstract

There is a growing need for solar-powered solutions to power existing transportation systems, including existing electric vehicles. The depletion of conventional energy sources, such as oil, gives reason for developing electric vehicles powered by a renewable source, such as solar power.

To optimise vehicle performance, we must balance aerodynamic efficiency with optimal solar panel surface area, ensuring the outer shell design generates maximum power while minimising drag. This is done by minimising the two key parameters, the drag coefficient and the frontal area. This means making key design decisions weighing various trade-offs, drag force, solar panel efficiency, solar panel area, and weight implications based on the choice of solar pane

An iterative design methodology is used to optimise vehicle performance. To do so, an initial baseline design is to be developed. This was done using CFD tools to analyse flow structures around the geometry, providing information on features such as drag and pressure coefficient, streamlines indicating boundary layer behaviour, separation zones and wake formation. This information dictates the design choices made while adapting the baseline design using CAD software packages.

The resistive forces of drag obtained from the Computational Fluid Dynamics(CFD) simulations were combined with rolling resistance and mechanical inefficiencies to give a total resistive force acting on the vehicle. The results demonstrated that the holistic methodology used, integrating the aerodynamic drag reduction analysis with the solar power generation, helps determine an ideal outer shell configuration. This methodology has wider applications and can be applied to anything having an interplay of aerodynamics and solar power generation, for example, solar-powered drones.

Lay Summary

This study presents a holistic approach to designing the outer shell of a solar-powered vehicle for the Shell Eco-marathon competition. By combining aerodynamic analysis with solar energy modelling, the research ensures that the car's shape not only minimizes air resistance but also maximizes the efficiency of solar power generation. This integrated methodology results in a vehicle design that is both energy-efficient and practical for real-world use. The significance of this approach extends beyond just competition vehicles. The insights gained from balancing aerodynamic performance with optimal solar panel placement can be applied to a wide range of technologies where both factors are important. For example, this methodology could benefit the design of solar-powered drones, boats, and even buildings with integrated solar panels, as well as next-generation electric vehicles. Essentially, any application where the interplay between solar power generation and aerodynamics is critical could leverage these findings to improve efficiency and sustainability. This holistic design strategy therefore offers valuable guidance for advancing sustainable transportation and other green technologies.

Declaration

I declare that this dissertation is my own work and has not been submitted for any other degree or professional qualification. All sources of information and data used in this work have been properly acknowledged.

Intellectual Property Statement

1. The author of this dissertation (including any appendices and/or schedules to this dissertation) owns certain copyright or related rights in it (the "Copyright") and he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

2. Copies of this dissertation, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has entered into. This page must form part of any such copies made.

3. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the dissertation, for example graphs and tables ("Reproductions"), which may be described in this dissertation, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions

4. Further information on the conditions under which disclosure, publication and commercialisation of this dissertation, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy, in any relevant Dissertation restriction declarations deposited in the University Library, and The University Library's regulations.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Dr. Alessandro De Rosis. Working with him has been a truly joyful experience, and his support throughout the project has been invaluable. His insightful guidance, especially in helping me lay out the scope of my work, ensured that the project stayed on track and progressed smoothly. Our discussions have always been enlightening and encouraging, and I am deeply appreciative of his helpfulness and the positive impact he has had on my work.

I would also like to thank the UOM CSF4 computational facility for providing access to their high-performance computing resources. The availability of this HPC infrastructure was essential for conducting my simulations efficiently and effectively, and it played a significant role in the successful completion of my project.

Contents

List of Figures

List of Tables

Abbreviations used

Abbreviation	**Meaning**
CFD	Computational Fluid Dynamics
Cd	Drag Coefficient
CAD	Computer-Aided Design
PVGIS	Photovoltaic Geographical Information System
NACA	National Advisory Committee for Aeronautics
SST	Shear Stress Transport
Re	Reynolds Number
VOC	Open-circuit Voltage
ISC	Short-circuit Current
y^+	Non-dimensional Wall Distance
u^+	Normalised Wall Velocity

1 Introduction

1.1 Background

The initial development of solar cars goes back decades. The first solar car invented was a tiny 15-inch vehicle created by William G. Cobb of General Motors, called the Sunmobile in 1955 (AutomoStory 2025).

Fig. 1: Sunmobile, General Motors 1955

The Tour de Sol, initiated in 1985, marked the beginning of organised solar car competitions. The event aimed to promote solar energy as a sustainable energy source for transportation. The inaugural race spanned 368 km, where 73 vehicles competed in Switzerland, on a track spanning from Romanshorn to Geneva (Enter Museum 2024).

Fig. 2: Car Competing at Tour De Sol 1987

This inspired the concept of solar-powered vehicle competitions, leading to the establishment of the World Solar Challenge in Australia in 1987. Initially, technology was limited, solar panels proved to be very inefficient, and batteries were too heavy. The motors were neither powerful nor efficient enough to make it a viable mode of transportation. However, over time, advances in materials science, battery technology, and aerodynamics have increased the viability of solar cars in experimental situations, especially in smaller-scale vehicles.

With the rise of university student teams engaging in research in this field, due to the emergence of the World Solar Challenge, active research is being conducted in this field. Competition played a pivotal role in shifting the perception of solar cars from a niche, far-fetched curiosity, an actual sophisticated research platform demonstrating the future of renewable mobility.

1.2 Shell Eco-marathon

The motivation behind this research is deeply intertwined with the work of the Aerodynamics sub-team of the Manchester Solar Car Society, which is a collaborative student society founded in September 2024 to compete in the Shell Eco-marathon. The society competes in the Prototype vehicle class and the Battery Electric division. Performance of the vehicle is measured in Km/KWh (Shell Eco-Marathon 2025).

The Shell Eco-marathon is a student-driven global competition aimed at developing highly efficient vehicles that use minimal energy to complete a track. It fosters interdisciplinary learning and innovative problem solving, with a focus on renewable energy. The Shell Eco-marathon began in 1985, officially launched in France (Shell Eco-Marathon n.d.). Initially, it featured a single category for internal combustion engines; in 1991, a subcategory for electric vehicles was introduced.

The society would ideally like to compete in the World Solar Challenge, however, it comes with a great financial and logistical burden that is difficult for a society to deal with upon formation. Which is why it was decided that we would aim to compete in a smaller competition. In comparison to the World Solar Challenge, which is a 3000 km cross-country race across the Australian Outback, the Shell Eco-marathon is held on shorter closed-loop circuits, with cars often completing distances of 1–2 km per lap till all their energy is used. Due to the simplified design requirements and controlled race environment, the Shell Eco-marathon makes it significantly easier to develop a competitive vehicle.

The winners of this competition have been teams that have managed to come up with innovative solutions in aspects related to battery technology and transmission. The application of solar panels on Shell Eco-marathon cars, however, is an unexplored idea

1.3 Motivation

The looming threat of depleting non-renewable energy has encouraged research into finding alternative methods of powering transport. The conversion of solar energy into electricity, using panels, is an alternative source, which has induced the application of solar energy in powering electric vehicles. The aerodynamic properties of the vehicle play a very important role in the performance of and are the guiding factors of the overall efficiency. The aerodynamic drag force acting on the outer shell is a significant factor that increases the energy required to propel the vehicle. This is an important consideration for a low-power vehicle running on solar-assisted systems. The outer shell serves a dual purpose, as it governs the aerodynamic properties of the car and acts as the integrating platform for the solar panels. As a result, the over shape and surface area contribute heavily to overall system performance, and it is necessary to find a balance between either having a flatter surface for more solar panel placement, or having a more streamlined and sharper profile to reduce drag. These are key design tradeoffs to be considered while developing the outer shell.

2 Aims and Objectives

2.1 Aims

The primary aim of this project is to develop and evaluate outer shell designs for a Shell Eco-marathon prototype class vehicle for the Manchester Solar Car Society. The outer shell of the vehicle is the most important aspect of the development of the vehicle, as the entire design process follows a ground-up approach. Which prioritises a main body optimised for minimal drag, determined by the outer shell's configuration. All other components are designed accordingly to complement and integrate with this aerodynamic foundation. This is done by incorporating varying geometrical features, which will then be assessed using computational fluid dynamics simulations to estimate their aerodynamic efficiency. This project also aims to consider the positioning of solar panels mounted on the surface of the outer shell, considering the curvature and estimating power generation under various operating conditions. It is key that an iterative design and analysis process is used to progressively refine the geometry, ensuring improvements in aerodynamic refinement do not compromise the solar panel area. The aim is to ultimately integrate the aerodynamic aspect of the outer shell with the energy generation requirements of the competition, creating a unified framework to evaluate the final design.

2.2 Objectives

1. Undertake review on existing use of solar panels to power vehicles, and aerodynamic analysis done on Shell Eco-marathon cars.

2. Assess the literature review conducted on World Solar Car Challenge vehicles, evaluating the methodologies used to develop the design.

3. Conduct research on various NACA aerofoil profiles as suggested by the methodologies applied in the literature review.

4. Develop a conceptual design based on the selected aerofoil and assess the aerodynamics and power generation.

5. Adapt design concepts from the World Solar Car Challenge vehicles for Shell Eco-marathon cars, developing a range of outer shell designs, each optimised for better aerodynamics, greater solar panel area, or both.

6. Integrate the requirements of solar panel placement, considering the curvature, orientation and operating conditions into the design process, ensuring that energy generation is realistically estimated.

7. Use a CFD tool such as STAR-CCM+ to identify regions exhibiting poor aerodynamic performance.

8. Evaluate the design using CFD simulations to identify the regions of poor aerodynamic performance, analysing the flow patterns and pressure distributions to assess which geometric features contribute to increased drag.
9. Assess the power generated by the solar panels as a function of the solar panel area, and compare it with the power required to overcome resistive forces.
10. Develop a methodology to account for the curvature of the solar panels while calculating the power generated, to obtain a more accurate estimate.
11. Evaluate the set of developed designs to identify the configuration of outer shell that offers the best balance between maximum solar power generation and aerodynamic efficiency, providing a design that is the most efficient overall.

3 Literature Review

3.1 Introduction

This literature has been organised into six distinct sections, each focusing on a specific aspect important for the overall analysis of this research. These sections explore the various concepts and findings that contribute towards developing a deeper understanding of the subject matter to be dealt with.

1. Introduction to aerodynamic drag, and design aspects to consider when designing a solar car outer shell.
2. Important metrics to keep in consideration when developing the design.
3. Relevant flow properties observed in flows over an outer shell.
4. CFD(Computational Fluid Dynamics) as a design tool.
5. Solar Panels and their operational principles, and the types of panels suitable for use on a solar car.
6. Review of the Shell Eco-marathon regulations for the Prototype Class
7. Looking at existing literature for similar analysis done on World Solar Car Challenge Cars, and comparing them with literature from analysis done on Shell Eco-marathon cars.

3.2 Aerodynamic Drag Properties

According to (Carroll 2003), the ideal start to developing the main body of a solar car must begin with selecting an aerofoil shape as its profile. This decision is to be made with the 5 fundamentals of aerodynamic drag:

Flow separation: Unaerodynamic shapes will cause flow separation from the body near the sharp corners in the body shape, creating spinning vortices leading to a turbulent region. This causes pressure-induced drag, which is to be avoided.

Skin Friction: As air flows over a streamlined body, there is friction between the air and the body; this is often the dominant drag force for streamlined bodies. Skin friction drag is proportional to the total surface area of the car, so reducing the surface area will tend to reduce the skin friction drag.

Boundary Layer Pressure Loss: A boundary layer develops as air flows over the body. This is a layer of fluid between the body and the free-stream flow. Air outside the boundary layer is undisturbed and smooth, flowing at the free-stream velocity.

Induced Drag: This is a change in drag on the body as the lift increases in magnitude. Minimum drag occurs when there is zero lift. It is ideal to ensure that the magnitude of the drag of the car is as close to zero as possible.

Interference Drag: This is drag caused by imperfections in the body, leading to mixing of different airflow streamlines.
According to (Tamai 1999) the basic equation to express aerodynamic drag is :

$$D = (C_D A) \cdot \frac{1}{2} \cdot \rho \cdot (V_{car} + V_{head})^2 \tag{3.1}$$

For simplification, V_{head}, the headwind velocity relative to the road is often taken as zero. The form of this drag equation was established by using the Buckingham π theorem for dimensional analysis, identifying non-dimensional groups such as the drag coefficient and Reynolds number. The full equation and values of C_D were later determined through experimental studies.

Many experimental studies to define drag coefficient for streamlined bodies were conducted by the National Advisory Committee at Langley, Virginia, from the late 1930s to 1950s. One such key development was the conceptualisation of 6-series aerofoils. These were designed with the intention of maximising laminar flow and minimising drag.

The 5 fundamental contributors to aerodynamic drag can be represented in an equation as follows (Tamai 1999):

$$D = D_{p,\mathrm{sep}} + (D_{\mathrm{skin}} + D_{p,\mathrm{BL}}) + D_{\mathrm{ind}} + D_{\mathrm{int}} \tag{3.2}$$

The pressure drag term $D_{p,\mathrm{sep}}$ is usually the dominant term of drag for conventional road vehicles. This drag originates when the airflow fails to remain attached to the body throughout the flow and separates when it reaches the end of the car. This creates a region of low pressure at the rear, causing the car to be sucked backwards. Such flow separation is attributed to shapes called bluff bodies. In typical road cars, 50-90 % of aerodynamic drag is separation pressure induced. Although separation drag is a separate entity from skin friction, it occurs due to skin friction. Separation drag depends mainly on the frontal area of the body.

The second term comprised of $(D_{\mathrm{skin}} + D_{p,\mathrm{BL}})$ is the viscous friction on the body. The terms are combined as they both occur due to the interactions with the boundary layer. The boundary layer is a thin region of flow near the body surface, where viscosity effects are significant. The skin friction component, D_{skin}, arises from the viscous shearing of fluid molecules along the surface. For attached flows as seen in streamlined bodies,D_{skin}, is the predominant drag component.

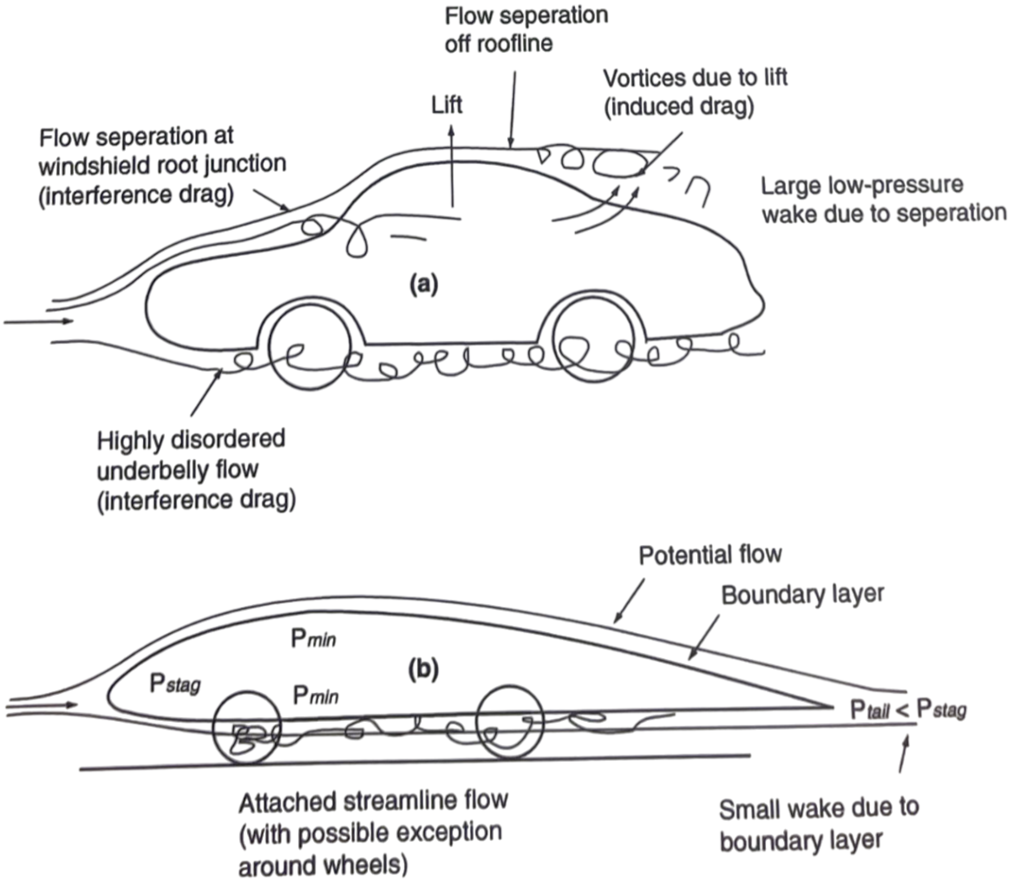


Fig. 3: Separated flow on a production car and attached flow on a streamlined body

The pressure drag due to the boundary layer,$D_{p,\mathrm{BL}}$, is generated due to the pressure gradient formed due to the velocity difference in the fluid, in the flow near the body. This occurs because, as flow progresses along the body, the flow accelerates, and pressure decreases as per Bernoulli's principle in the inviscid region outside the boundary layer. The pressure is at its maximum in the thickest region of the boundary layer, closest to the free stream region. As the flow goes through the body, the pressure tries to recover to the stagnation pressure formed at the beginning of the flow. Throughout this process, the boundary layer is increasing in thickness, and if the pressure gradient becomes too high, it leads to separation.

The third term of the equation induced drag,D_{ind}, is the result of lift acting on the body, due to the pressure difference between the top and bottom of the body, which creates vortices. The contribution of lift can also be used to reduce the aerodynamic drag force of the vehicle, depending on the profile of the vehicle.

The final term interference drag, D_{int}, is a result of the geometric imperfections of the body; common ones for solar cars are misaligned panels and the wheels that protrude from the main body. This is why, often, if a body is an assembly of multiple bodies, of two or more shapes attached, the resulting drag is usually greater than the sum of the individual drag of the bodies.

Whilst designing an ultra-streamlined body, minimisation of the sum of the five aerodynamic drag components are critical, and should be done in the order presented above.

3.3 Drag Area as a Metric

In the previous section, concepts of frontal area and drag coefficient were introduced. The drag area is defined as the product of the drag coefficient and its reference area (Tamai 1999) The chosen referral area makes a big difference in the metric being calculated; more often than not, the maximum cross-sectional area is taken as the frontal area. This analysis works well for a bluff body, since its aerodynamic drag is dominated by separation pressure drag. This is proportional to the base area, which is the area at the point of separation. This point commonly occurs where the cross-sectional area is maximum.

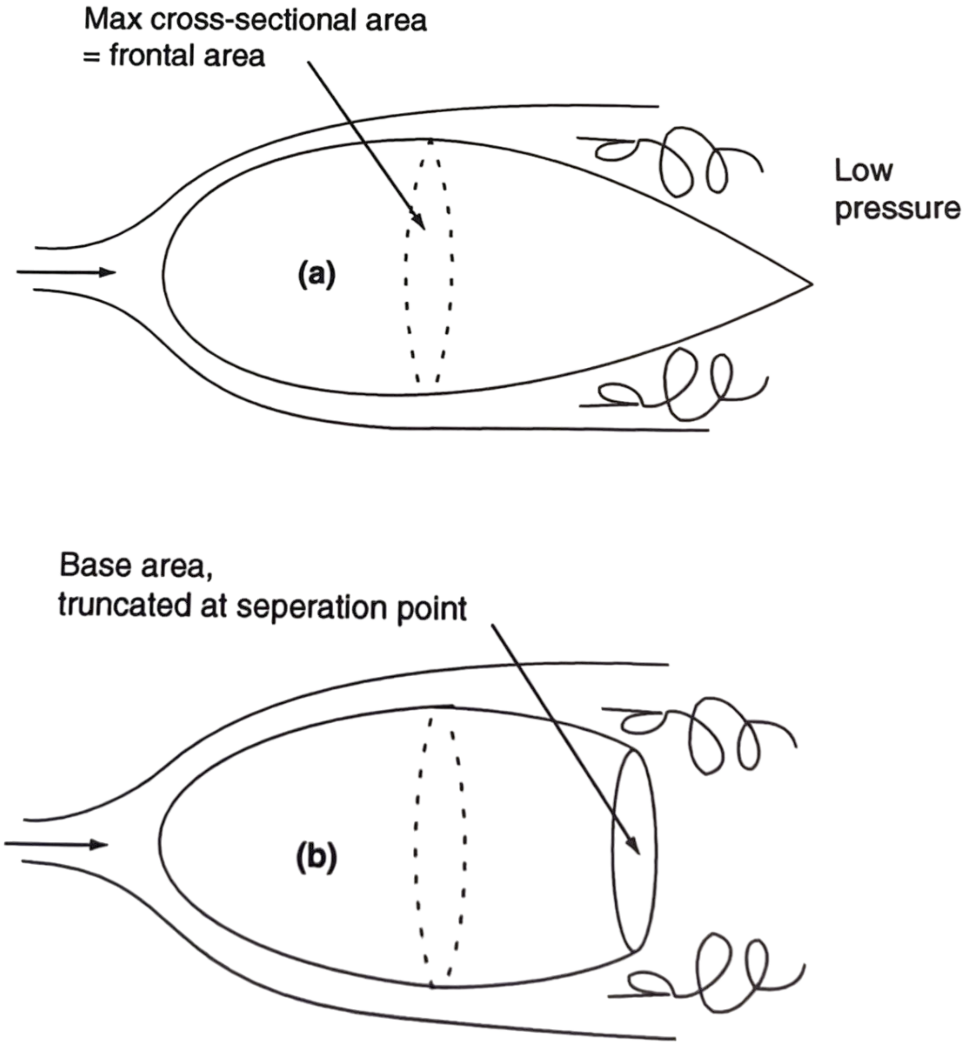


Fig. 4: Position of frontal area

For solar cars with streamlined shape, the aerodynamic drag is predominantly caused by skin friction. An apt reference area for this would be the total wetted area, which is the entire surface area of the body exposed to the fluid. Due to the curved nature of most streamlined shapes, it is difficult to compute this in an efficient manner. Instead planview area is used, which is the area of the body when viewed normal to the flow (Tamai 1999). The drag coefficient with the frontal area as its reference area is a representation mainly of the pressure-induced drag of the body. Whereas the drag coefficient with the planview area as its reference area is a representation of the skin friction drag of the body.

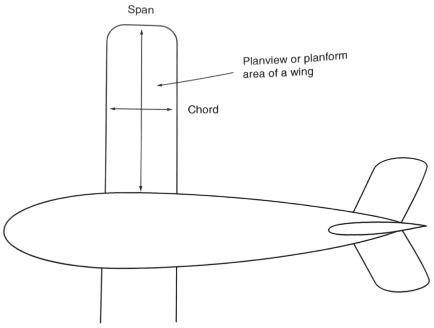


Fig. 5: Planview Area

3.4 Fluid Flow Properties

The Boundary Layer

The boundary layer is the region of fluid flow directly adjacent to the surface, where viscous effects are present. Viscosity represents the fluid's resistance to motion. For a body moving through a fluid like air at a certain relative velocity, known as the free-stream velocityU_{∞}. At the surface of the body, the fluid has zero velocity due to the no-slip condition, as the fluid sticks to the surface. Just above this stationary layer that is only a molecule thick, the velocity of the fluid increases despite being lower than U_{∞}. As you move away from the surface, the fluid velocity gradually increases till it reaches the free stream region. This region, with variation in fluid speed, is the boundary layer (Tamai 1999).

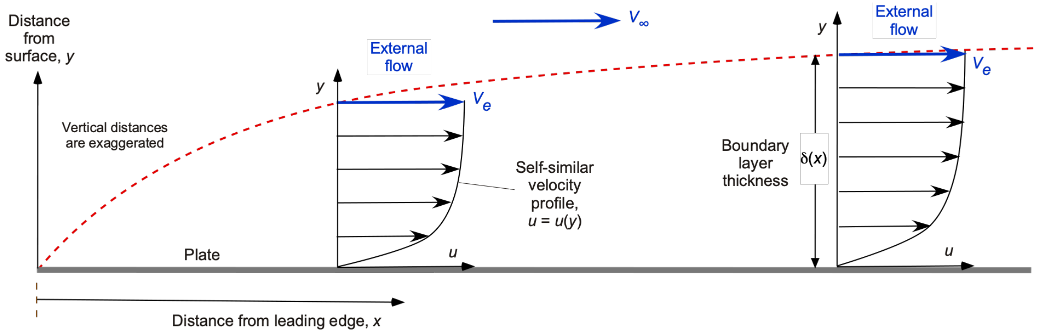


Fig. 6: Laminar Boundary Layer

Boundary layer flows can be characterised as two distinct regimes of boundary layer flows, laminar and turbulent. In laminar flow, there is no intermixing between layers of airflow. Turbulent boundary layer flows have intermixing between layers. Due to this intermixing of air layers requiring more energy, the skin friction from turbulent flows can be greater by a large magnitude, when compared to laminar flows. As the skin friction is proportional to the gradient of the velocity, and in the turbulent flow the velocity is increasing more rapidly near the surface, skin friction increases. Most flows start as laminar flows, then transition into turbulent flows due to naturally occurring instabilities (White 2016).

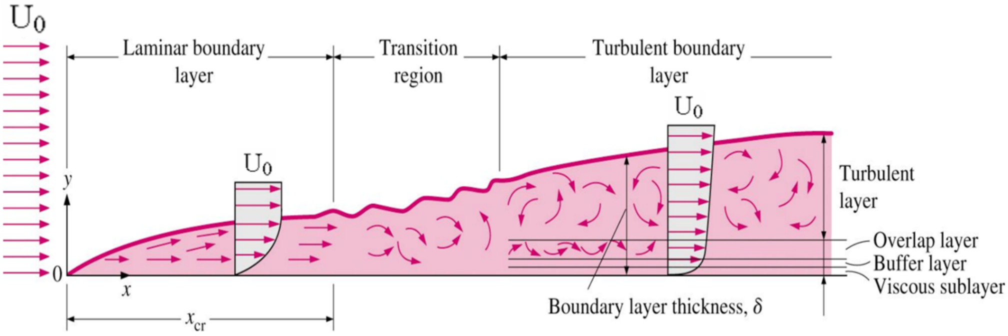


Fig. 7: Boundary Layer Transition

This is why for the design of the outer shell, it is critical that a laminar flow regime is preserved for the majority of the flow.

Boundary layer sublayers and wall distance y^+

The inner region of the boundary layer can be split into three sublayers.

1. Viscous Sublayer
   The fluid layer in cont act with the wall is dominated by viscous effects. The flow velocity here only depends on the fluid density, viscosity,wall distance and wall shear stress.

2. Log layer
   This layer is the turbulent log layer dominated by both viscous and turbulent effects.

3. Buffer layer
   This is the transitional layer between the viscous sublayer and the log layer

Each of the sublayers can be modelled using different approaches; the non-dimensional wall distance y^+ is used to define the extent of the sublayers. The following plot shows the non-dimensional velocity u^+ across the three sublayers.

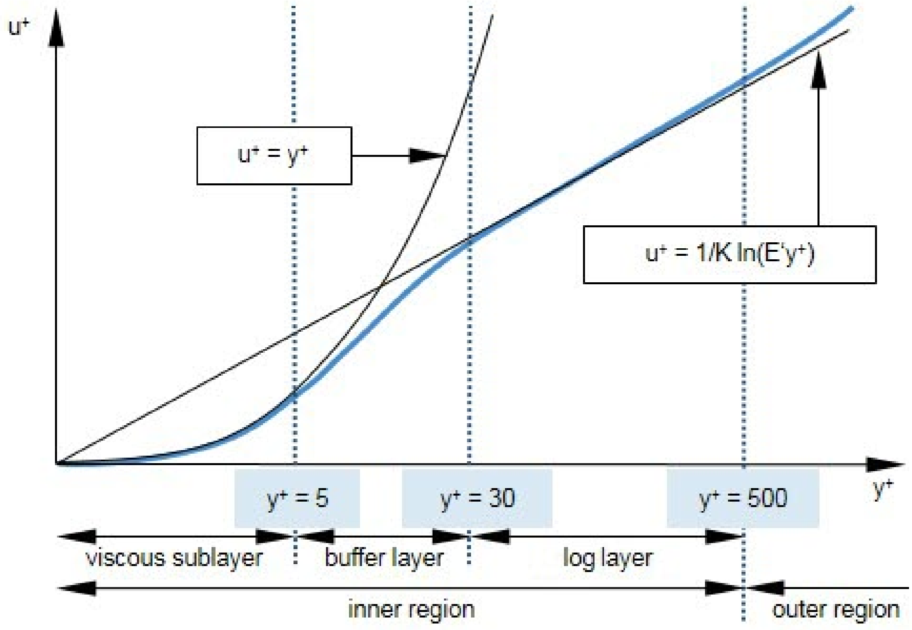


Fig. 8: Change in u^+ across the sublayers

Log-law equation for for u^+:

$$u^+ = \frac{1}{\kappa}\ln(y^+) + C \tag{3.3}$$

where κ is the von Kármán constant (approximately 0.41), and C is a constant depending on wall roughness.

In the viscous sublayer the u^+ and y^+) values are equal. The prediction of drag by the CFD solver is governed by the structure of the boundary layer. If the mesh or turbulence model does not resolve the sublayers correctly, drag predictions will be incorrect. Which is why an ideal wall treatment model is to be selected while using the CFD solver.

Flow Separation

Flow separation occurs when the flow is under an adverse pressure gradient. The higher downstream pressure pushes back the flow near the surface, leading to an inflexion point in the boundary layer. This changes the profile of fluid flow from convex-forward to convex-rearward. With a high enough pressure gradient, the velocity gradient at the surface becomes zero, indicating the point of separation. This phenomenon commonly occurs if a body contour has abrupt curves, not letting the flow stay attached to the surface.

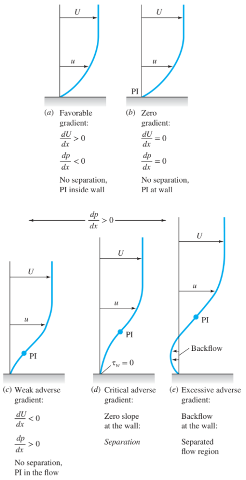


Fig. 9: Effect of pressure gradient on flow separation

3.5 CFD for Analysis

Introduction

Computational Fluid Dynamics is the method by which the governing fluid dynamics differential equations are discretized to given an approximation of expected flow properties.

Governing Equations

The governing differential equations of fluid flows include the Navier-Stokes equations of continuity and momentum.

Continuity:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho U_i}{\partial x_i} = 0 \tag{3.4}$$

Momentum:

$$\frac{\partial \rho U_i}{\partial t} + \frac{\partial \rho U_i U_j}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu \frac{\partial U_i}{\partial x_j}\right) \tag{3.5}$$

To generate numerical solutions to these equations, approximations have to be made to the derivatives and integrals. The method by which these approximations are made in most CFD codes is called the Finite Volume Method.
The Navier-Stokes equations are used to generate a generalised scalar transport equation, which is the equation being discretised in the Finite Volume Method (John D. Anderson 2009).

$$\underbrace{\frac{\partial}{\partial x}(\rho U\phi)+\frac{\partial}{\partial y}(\rho V\phi)}_{\text{Convection terms}}=\underbrace{\frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Gamma\frac{\partial\phi}{\partial y}\right)}_{\text{Diffusion terms}}+\underbrace{S_\phi}_{\text{Source terms}} \tag{3.6}$$

Convection terms: $\frac{\partial}{\partial x}(\rho U\phi)+\frac{\partial}{\partial y}(\rho V\phi)$. These terms represent the transport of the scalar ϕ due to the motion of the fluid. Here, ρ is the fluid density, U and V are the velocity components in the x and y directions respectively, and ϕ is the scalar quantity being transported.

Diffusion terms: $\frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right)+\frac{\partial}{\partial y}\left(\Gamma\frac{\partial\phi}{\partial y}\right)$. These terms account for the spreading of ϕ due to diffusion. Here, Γ is the diffusion coefficient, which is thermal conductivity for heat or dynamic viscosity for momentum, and the gradients $\frac{\partial\phi}{\partial x}$ and $\frac{\partial\phi}{\partial y}$ quantifies how fast ϕ changes in space.

Source term: S_ϕ. This term includes any internal generation or consumption of ϕ within the control volume.
The above transport equation reflects the terms in a two-dimensional setup; for a three-dimensional case, there would simply be extra terms with the z component.

For laminar flows, the above transport equations can be discretised directly using the Finite Volume Method, creating an algebraic set of equations to solve.

Reynolds Averaged Navier Stokes

For turbulent flows, however, there are constant fluctuations with the value of the scalar ϕ, which makes it difficult to directly apply the full Navier-Stokes equations.

A solution for this was proposed by Reynolds, where the fluctuating velocity and pressure fields are split into an average part and a fluctuating part (Reynolds 1883).

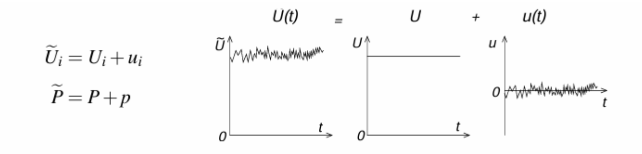


Fig. 10: Decomposition of velocity into averaged and fluctuating parts (Craft 2024b)

The flow is treated as mainly steady, which allows the decomposition to be easily done by the following definition of the averaged value.

$$U_i=\lim_{T\to\infty}\frac{1}{T}\int_0^T \tilde{U}_i\,dt \tag{3.7}$$

These decomposed scalars can be substituted into the Navier-Stokes equations, where instead of solving for the instantaneous and fluctuating velocity field, the averaged profile is solved for.

Continuity:

$$\frac{\partial \rho U_i}{\partial x_i} = 0 \tag{3.8}$$

Momentum:

$$\frac{\partial \rho U_i}{\partial t} + \frac{\partial (\rho U_i U_j)}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu \frac{\partial U_i}{\partial x_j} - \rho \overline{u_i u_j}\right) \tag{3.9}$$

The above set of equations has a Reynolds stress term, $\rho\overline{u_i u_j}$, representing the effects of turbulent eddies that are not being resolved. A model is required to approximate this term. These models are called eddy viscosity models.

These models use the Boussinesq eddy viscosity assumption, which links the Reynolds stress term, $\rho\overline{u_i u_j}$, with a scalar property called the eddy viscosity, μ_t.

$$-\rho\overline{u_i' u_j'} = \nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{2}{3}\frac{\partial U_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\rho k \delta_{ij} \tag{3.10}$$

The turbulent kinetic energy, k, is defined as:

$$k = \frac{1}{2}\left(\overline{u_1^2} + \overline{u_2^2} + \overline{u_3^2}\right)$$

There are various eddy viscosity models, that compute the eddy viscosity in differing manners, ν_t. The most commonly used models used in commercial CFD codes are two-equation models called the $k-\epsilon$, and the $k-\omega$ model.

$k-\epsilon$ **model:**

Turbulent kinetic energy equation:

$$\frac{Dk}{Dt} = P_k - \varepsilon + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \tag{3.11}$$

Dissipation rate equation:

$$\frac{D\varepsilon}{Dt} = C_{\varepsilon 1}\frac{\varepsilon}{k}P_k - C_{\varepsilon 2}\frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] \tag{3.12}$$

Turbulent viscosity:

$$\nu_t = C_\mu \frac{k^2}{\varepsilon} \tag{3.13}$$

$k - \omega$ model:

Turbulent kinetic energy equation:

$$\frac{Dk}{Dt} = P_k - \beta^* k\omega + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_k \nu_t)\frac{\partial k}{\partial x_j}\right] \tag{3.14}$$

Specific dissipation rate equation:

$$\frac{D\omega}{Dt} = \alpha\frac{\omega}{k}P_k - \beta\omega^2 + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_\omega \nu_t)\frac{\partial \omega}{\partial x_j}\right] \tag{3.15}$$

Turbulent viscosity:

$$\nu_t = \frac{k}{\omega} \tag{3.16}$$

The main difference in the two equations are the transported properties, where the $k - \epsilon$ model calculates for k (turbulent kinetic energy) and ϵ (dissipation rate). Whereas, the $k - \omega$ model calculates for ω (specific dissipation rate).The two models have their own drawbacks and advantages depending on the scenario. The $k - \omega$ model is a low Reynolds model, that can be used for flows where the boundary layer is relatively thick and the viscous sub layer easily resolved. The $k - \epsilon$ model uses damping functions to resolve the boundary layer. These functions are not very accurate in the presence of adverse pressure gradients. The $k - \omega$ model on the other hand, does not require functions to calculate flow properties in the near-wall region. This leads to it having better accuracy in predicting flow separation and, dealing with adverse pressure gradients. One of the advantages the $k - \epsilon$ model has over the $k - \omega$ model is that it performs better in free-stream regions away from the walls, and is less sensitive to inlet free stream value, especially in fully turbulent flows (Versteeg and Malalasekera 2007).

$k - \omega$ SST model:

Since the $k - \epsilon$ model shows good results in the free stream region, and the $k - \omega$ model has a good accuracy when resolving the boundary layer region. Combining the advantages of these two models using a blending function gives rise to the $k - \omega$ SST model.
Combining the 2 models gives: The turbulence specific dissipation rate equation is given by:

$$\frac{D}{Dt}(\rho\omega) = \nabla \cdot (\rho D_\omega \nabla\omega) + \rho\gamma G - \frac{2}{3}\rho\omega(\nabla \cdot \mathbf{u}) - \rho\beta\omega^2 - (F_1 - 1)\, CD_{kw} + S_\omega \tag{3.17}$$

and the turbulence kinetic energy by:

$$\frac{D}{Dt}(\rho k) = \nabla \cdot (\rho D_k \nabla k) + \rho G - \frac{2}{3}\rho k(\nabla \cdot \mathbf{u}) - \rho\beta^*\omega k + S_k \tag{3.18}$$

The turbulence viscosity is obtained using:

$$\nu_t = a_1 \frac{k}{\max\left(a_1\omega, b_1 F_{23} S\right)} \tag{3.19}$$

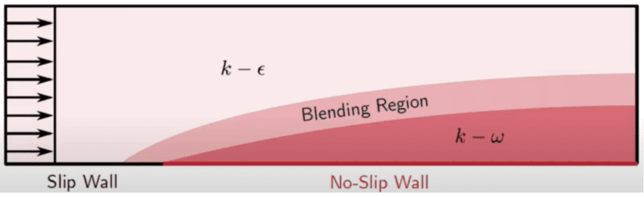
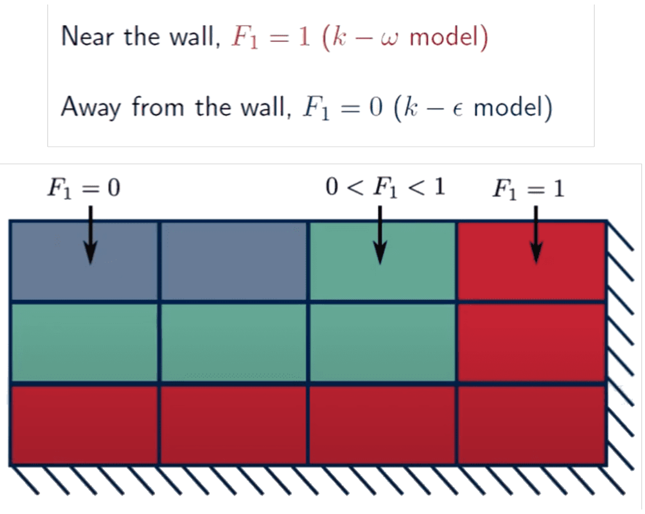


Fig. 11: Blending between $k-\epsilon$ and $k-\omega$ turbulence models in $k-\omega$ SST model (SimScale n.d.)

In the above equations, F_1 is the blending function. If F_1 is zero, equation 3.16 gives ϵ representing the $k-\epsilon$ turbulence model. If F_1 is 1, the additional term equates to 0, and thus the transport equation represents the $k-\omega$ model. The value of F_1 in the blending region is between 1 and 0.



Near the wall, $F_1 = 1$ ($k - \omega$ model)

Away from the wall, $F_1 = 0$ ($k - \epsilon$ model)

Fig. 12: Blending Function F_1 variation with wall distance.(SimScale n.d.)

The above equations are two-equation models, meaning that they solve two extra transport equations apart from the Navier-Stokes equations. This can be computationally challenging, especially for complex geometries and flows. This is due to both models involving the solving of sets of non-linear equations. Due to this, they tend to suffer from slow convergence.

The Spalart Allmaras Model

The Spalart-Allmaras Model is a one-equation model. This leads to lower computational cost than two-equation models. Since there is no coupling or interdependence between the different transport properties, it leads to stable and quicker convergence.

Transport equation for the modified turbulent eddy viscosity $\hat{\nu}$:

$$\frac{\partial \hat{\nu}}{\partial t} + u_j \frac{\partial \hat{\nu}}{\partial x_j} = c_{b1}(1 - f_{t2})S\hat{\nu} - \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right] \left(\frac{\hat{\nu}}{d}\right)^2 + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \hat{\nu})\frac{\partial \hat{\nu}}{\partial x_j}\right) + c_{b2}\frac{\partial \hat{\nu}}{\partial x_i}\frac{\partial \hat{\nu}}{\partial x_i}\right] \tag{3.20}$$

The turbulent eddy viscosity is computed from:

$$\mu_t = \rho \nu f_{v1} \tag{3.21}$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \tag{3.22}$$

$$\chi = \frac{\nu^\star}{\nu} \tag{3.23}$$

and ρ is the density, $\nu = \frac{\mu}{\rho}$ is the molecular kinematic viscosity, and μ is the molecular dynamic viscosity. Additional definitions are given by the following equations:

$$\hat{S} = \Omega + \frac{\nu^\star}{k^2 d^2} f_{v2} \tag{3.24}$$

The turbulent eddy viscosity is computed from:

$$\mu_t = \rho \nu f_{v1} \tag{3.25}$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \tag{3.26}$$

$$\chi = \frac{\nu^\star}{\nu} \tag{3.27}$$

and ρ is the density, $\nu = \frac{\mu}{\rho}$ is the molecular kinematic viscosity, and μ is the molecular dynamic viscosity. Additional definitions are given by the following equations:

$$\hat{S} = \Omega + \frac{\nu^\star}{k^2 d^2} f_{v2} \tag{3.28}$$

The aforementioned turbulence models are critical to the study, and the ideal selection of a model is imperative to have an accurate estimation of the drag force acting on the outer shell.

The γ–Re_θ Transition Model

The γ–Re_θ transition model is a two-equation model to simulate the transition from laminar to turbulent flow in wall-bounded flows. It modifies the turbulent transport equations to account for laminar, transitional, and turbulent states in a fluid flow (ANSYS Inc. 2025).
The model introduces two additional variables:

- **Intermittency** (γ): Represents the portion of time for which the flow is turbulent at a given location ($\gamma = 0$ for fully laminar, $\gamma = 1$ for fully turbulent).
- **Transition momentum thickness Reynolds number** ($Re_{\theta t}$): Used to predict the onset of transition based on local flow conditions.

The following equations are used to:

$$\frac{\partial(\rho\gamma)}{\partial t} + \frac{\partial(\rho U_j \gamma)}{\partial x_j} = P_\gamma - E_\gamma + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_f}{\sigma_f}\right)\frac{\partial \gamma}{\partial x_j}\right] \tag{3.29}$$

$$\frac{\partial(\rho\overline{Re_{\theta t}})}{\partial t} + \frac{\partial(\rho U_j \overline{Re_{\theta t}})}{\partial x_j} = P_{\theta t} + \frac{\partial}{\partial x_j}\left[\sigma_{\theta t}(\mu + \mu_t)\frac{\partial \overline{Re_{\theta t}}}{\partial x_j}\right] \tag{3.30}$$

The γ–Re_θ transition model is very useful in modelling the transition in the boundary layer from laminar to turbulent. This variation in boundary layer properties has major implications for the estimation of the drag force acting on the outer shell by the numerical CFD solver.

Discretisation Schemes

Discretisation refers to the process of splitting the domain, either time, space or physical properties, into smaller grid points to approximate the solution.

The commonly used discretisation method for fluid dynamics analysis is the Finite Volume Method. In this method, the physical domain is divided into smaller control volumes, where the governing equations are integrated over these control volumes.

The Finite Volume Method works on the conservation of fluxes across the faces of each control volume. The fluxes through each face of the control volumes are computed and then used to update the values within each volume.

For the given transport equation:

$$\frac{\partial}{\partial x}(\rho U\phi) + \frac{\partial}{\partial y}(\rho V\phi) = \frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma\frac{\partial\phi}{\partial y}\right) + S_\phi \tag{3.31}$$

The integral form of the above equation is integrated across the control volume:

$$\int_V \frac{\partial}{\partial x}(\rho U\phi)\,dx\,dy + \int_V \frac{\partial}{\partial y}(\rho V\phi)\,dx\,dy = \int_V S_\phi\,dx\,dy + \int_V \frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right)\,dx\,dy + \int_V \frac{\partial}{\partial y}\left(\Gamma\frac{\partial\phi}{\partial y}\right)\,dx\,dy \tag{3.32}$$

The terms are discredited as follows for a two-dimensional flow :

$$\int_V \frac{\partial}{\partial x}(\rho U\phi)\,dx\,dy = \left[\int \rho U\phi\,dy\right]_w^e \tag{3.33}$$

$$\int_V \frac{\partial}{\partial y}(\rho V\phi)\,dx\,dy = \left[\int \rho V\phi\,dx\right]_s^n \tag{3.34}$$

$$\int_V \frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right)\,dx\,dy = \left[\int \Gamma\frac{\partial\phi}{\partial x}\,dx\right]_w^e \tag{3.35}$$

$$\int_V \frac{\partial}{\partial y}\left(\Gamma\frac{\partial\phi}{\partial y}\right)\,dx\,dy = \left[\int \Gamma\frac{\partial\phi}{\partial y}\,dx\right]_s^n \tag{3.36}$$

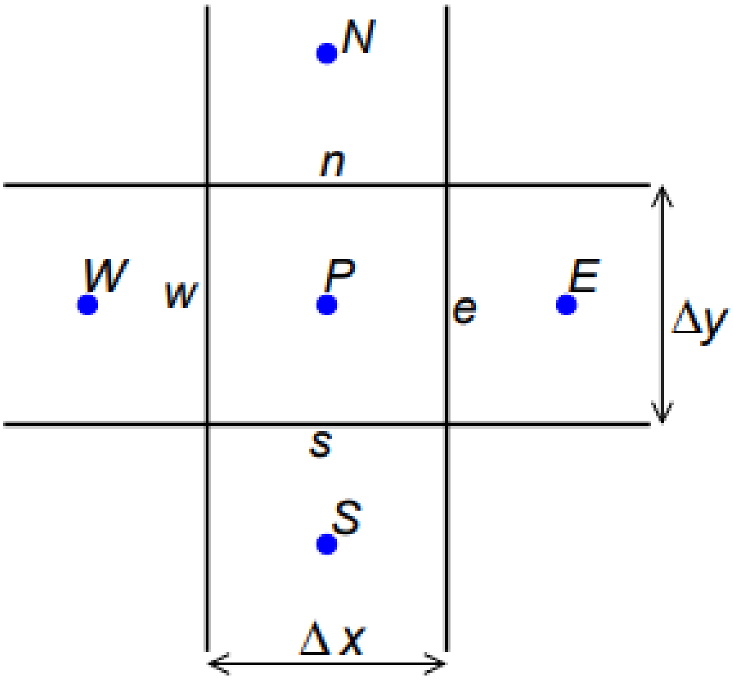


Fig. 13: 2-D Uniform Rectangular Grid (Craft 2024a)

The equations above are for a two-dimensional laminar flow problem; however, this approach is the same for a turbulent flow as well; the RANS equations mentioned above can be discretised similarly, along with the transport equations for the turbulence models.

Second-Order Upwind Scheme

The Second-Order Upwind Scheme is the most commonly used discretising scheme in commercial CFD code(Craft 2024a, p.22).

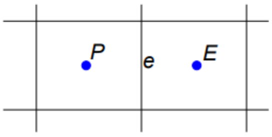


Fig. 14: 1-D Grid (Craft 2024a)

$$\phi_e = \begin{cases} \phi_P + \left(\frac{\Delta x}{2}\right)\left(\frac{\partial \phi}{\partial x}\right)_P & \text{if } C_e \geq 0 \\ \phi_E - \left(\frac{\Delta x}{2}\right)\left(\frac{\partial \phi}{\partial x}\right)_E & \text{if } C_e < 0 \end{cases} \tag{3.37}$$

In the above equation C_e is the mass flux going east, usually in the direction of the flow. This shows how the value of a scalar property ϕ changes as the equation is iteratively solved across the grid.

In order to assess the accuracy of the set of equations, the Taylor series approximation of the

ϕ_e term is done, giving:

$$\phi_e = \phi_P + \left(\frac{\Delta x}{2}\right)\left(\frac{\partial \phi}{\partial x}\right)_P + \frac{\left(\frac{\Delta x}{2}\right)^2}{2!}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_P + \frac{\left(\frac{\Delta x}{2}\right)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_P + O(\Delta x^4) \tag{3.38}$$

Here it can be seen that the approximation is built from the first two terms of the expansion, and the leading error term is proportional Δx^2. Which is why this scheme is second-order accurate. The second-order upwind scheme provides an excellent balance between robustness and accuracy, which is why it is so commonly used.

Solution Strategies to Solve Discretised Equations

The most commonly used strategies to solve the set of discretised equations are the segregated and coupled solver strategies. For the purpose of this research, the Segregated solver will be looked at in more detail, since it is the more computationally efficient and suited for low-speed flows, as expected in the CFD analysis of a solar car.

Segregated Solver

For 2D incompressible, steady, laminar flow, the governing equations are (Craft 2024c):

U-Momentum:

$$\frac{\partial}{\partial x}\left(\rho U^2\right) + \frac{\partial}{\partial y}\left(\rho UV\right) = -\frac{\partial P}{\partial x} + \frac{\partial}{\partial x}\left(\mu \frac{\partial U}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial U}{\partial y}\right) \tag{3.39}$$

V-Momentum:

$$\frac{\partial}{\partial x}\left(\rho UV\right) + \frac{\partial}{\partial y}\left(\rho V^2\right) = -\frac{\partial P}{\partial y} + \frac{\partial}{\partial x}\left(\mu \frac{\partial V}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial V}{\partial y}\right) \tag{3.40}$$

Continuity:

$$\frac{\partial}{\partial x}(\rho U) + \frac{\partial}{\partial y}(\rho V) = 0 \tag{3.41}$$

Discretization

The first two equations (U-Momentum and V-Momentum) can be discretised using the finite volume method, resulting in difference equations at each node:

General Form:

$$a_P\phi_P = a_W\phi_W + a_E\phi_E + a_N\phi_N + a_S\phi_S + S \tag{3.42}$$

Expanded Form (Considering velocities in x and y directions):

$$a_PU_P = a_WU_W + a_EU_E + a_NU_N + a_SU_S + S_U \tag{3.43}$$

$$a_PV_P = a_WV_W + a_EV_E + a_NV_N + a_SV_S + S_V \tag{3.44}$$

Iterative Solution Process

Due to the non-linear terms and coupling between equations, the coefficients a_P, a_W, etc, depend on both U and V. Additionally, pressure enters the source terms S_U and S_V. A segregated solution method solves these equations iteratively as follows:

1. Initial guesses for U, V, and P.
2. Solve the U-momentum equation (assuming V and P are known) to update U.
3. Solve the V-momentum equation (assuming U and P are known) to update V.
4. Solve the pressure equation to update P.
5. Repeat until convergence is achieved.

The above solver strategy shows how a CFD solver solves the set of discretised equations iteratively to achieve converged values. The segregated solver has been used in the CFD analysis of the solar car outer shell in this study, this is because for incompressible flows, pressure is not dependent on density, so decoupling pressure and velocity, as shown above, allows for convergence to be achieved a lot faster.

The SIMPLE algorithm

In combination with the discretisation methods and solver strategies, an algorithm needs to be used to simplify the process and decouple the interdependency of pressure and velocity One of the most commonly used algorithms for this purpose, to iteratively solve the discretised form of the Navier-Stokes equation in commercial CFD codes, is the SIMPLE(Semi-Implicit Method for Pressure-Linked Equations) algorithm. This algorithm is a pressure correction algorithm used to decouple the interdependent velocity and pressure field, solving them iteratively, ensuring mass and momentum conservation (Craft 2024c, p.14).

The steps in the SIMPLE algorithm are as follows:

$$U_P = \sum a_i U_i + D_u(P_w - P_e) + S_u \tag{3.45}$$

$$V_P = \sum a_i V_i + D_v(P_s - P_n) + S_v \tag{3.46}$$

In these equations:

1. U_P and V_P are the velocity components at the current cell.
2. The terms $a_i U_i$ represent the contributions from the neighbouring velocity components.
3. P_w and P_e are the pressures at the west and east faces of the control volume.

4. D_u and D_v are the diffusion coefficients in the x and y directions, respectively.

5. S_u and S_v are the source terms.

6. The gradient of the pressure $\left(\frac{\partial P}{\partial x}\right)$ is approximated as $\frac{P_e - P_w}{\Delta x}$, where Δx is the grid spacing in the x-direction.

The SIMPLE algorithm is an essential part of the CFD solver, improving the computational efficiency of the system by effectively decoupling the pressure and velocity terms, massively reducing the computation time needed for the CFD simulations.

Importance of CFD as a design tool

The importance of CFD as a design tool to assess the drag acting on the surface of the outer shell is critical due to the cost and time efficiency it provides. Using a CFD tool such as STAR-CCM+ allows the application of above mentioned concepts effectively. The main goal of using a CFD tool for the purposes of this study, is to compute the total drag acting on the outer shell, the application of above mentioned concepts does this, as the flow properties through each cell in the mesh is computed, it becomes possible to calculate the force acting on each cell that is on the surface of the outer shell, this is done by considering the pressure element of the force and shear element. Both of these are integrated along the entire surface to give the total force on the body.
Computational Fluid Dynamics (CFD) is an irreplaceable design tool for evaluating drag acting on surfaces or bodies, It offers exceptional efficiency in terms of cost and time compared to traditional methods such as wind tunnel testing.

The primary objective of CFD in this context is to determine the total drag force acting on the outer shell of the solar car. This is done through the above mentioned fluid flow concepts and discretisation schemes, giving a sophisticated computational process resolving flow properties within each cell of a mesh. Calculating the drag force acting on the surface of the outer shell is done by integrating the pressure and shear contributions of force, across the entire surface mesh, providing a detailed drag profile of the outer shell.

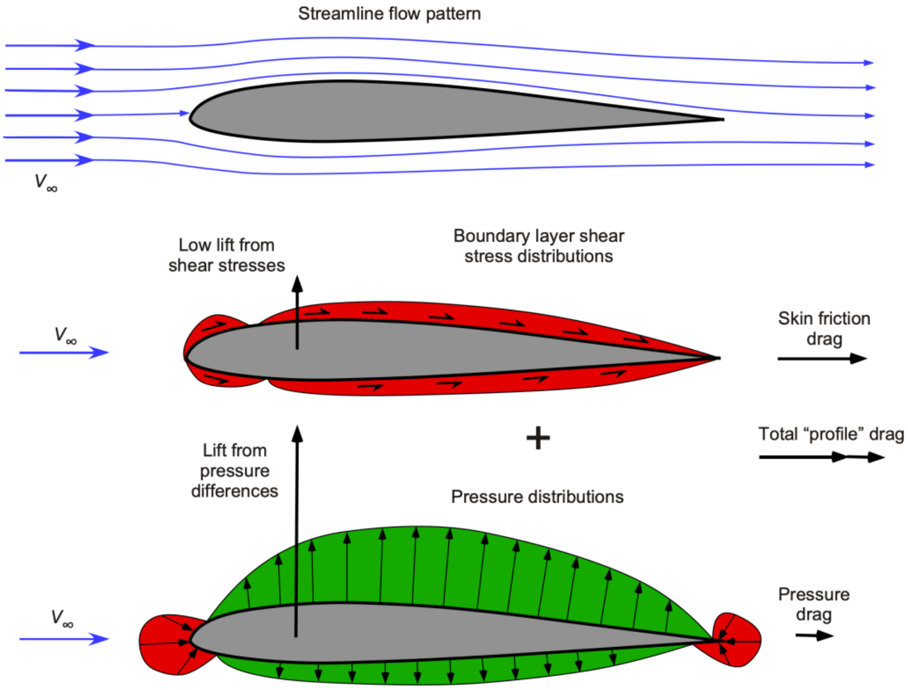


Fig. 15: Split of force into pressure component and shear force in an aerofoil (University 2021)

3.6 Prototype Class Shell Eco-marathon regulations

- **Maximum vehicle height:** less than 1000 mm
- **Minimum track width:** 500 mm (measured between the midpoints where the tyres of the outermost wheels touch the ground)
- **Height-to-track width ratio:** must be less than 1.25
- **Minimum wheelbase:** 1000 mm
- **Maximum total vehicle width:** 1300 mm
- **Maximum total vehicle length:** 3500 mm
- **Maximum vehicle weight (excluding the driver):** 140 kg
- None of the above body dimensions may be achieved by design singularities such as ‘stuck-on’ appendages or cut-outs

The above-listed design considerations are to be kept in mind while developing the outer shell design. The aim of this research is to develop the best possible design, maximising solar panel area and optimising for aerodynamics, whilst staying within the regulations (Shell Eco-Marathon 2025).

3.7 PV Cell Technology and Application in Solar Vehicles

Photovoltaic Cell Operational Principle

A silicon-based photovoltaic cell consists of p-type and n-type silicon layers. The p-type silicon has fewer electrons than silicon, leaving holes. N-type silicon has an extra electron, creating electrons that can freely move. When the two types are combined, electrons from the n-type side move towards the holes in the p-type side, creating a depletion zone. This zone is oppositely charged on both sides, leading to an internal electric field that prevents further electron flow(Society 2013).

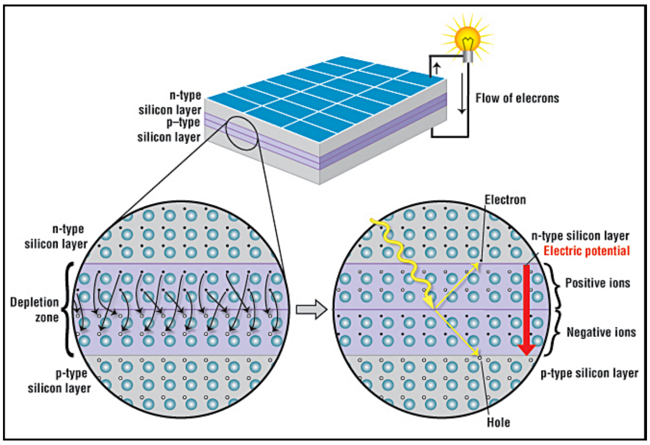


Fig. 16: Schematic representation of a solar cell showing the n-type and p-type layers, with a close-up of the depletion zone at the junction. (Society 2013)

When sun rays hit the photovoltaic cell, it frees electrons, creating extra vacant space. The electric field in the depletion zone pushes electrons to the n-type side and holes to the p-type side, leading to current generation when the two layers are connected through a circuit.

The most common type of solar panels used in solar cars are monocrystalline solar panels. These are the most reliable, efficient and case-friendly solution for solar cars(Technologies 2022). Typically, the efficiency of monocrystalline cells at the highest level ranges from 15-24%. Another silicon-based photovoltaic cell is the polycrystalline cell. Instead of being built from a single silicon crystal, it is built using multiple silicon crystals. The manufacturing process tends to be cheaper; however, it is not as efficient and has an efficiency ranging from 13-16 %. Polycrystalline cells are commonly used in solar fields and large-scale applications where space is not a concern.

During the last 10 years, much research and development have taken place in high-efficiency photovoltaic technologies, due to the push for renewable energy and the World Solar Car Challenge. This has led to an increase in new technologies in the field of photovoltaics, which can be applied to solar cars (Management 2019). This includes flexible monocrystalline cells, GaAs (gallium arsenide) single junction cells, with much higher efficiency, ranging from 39-47% (Yamaguchi 2020), and thin film cells that provide greater ease of integration.

GaAs (Gallium Arsenide) solar cells are at the cutting edge of photovoltaic efficiency; they are extremely efficient and are very thin, with a thickness of just 1-5 μms , making them significantly lightweight in comparison to traditional silicon-based cells. For solar car applications, GaAs cells provide an unmatched power-to-weight ratio and superior performance in high-temperature environments. This is a critical advantage in solar cars, where thermal management systems are not in place to avoid the unwanted weight. They also have great radiation resistance, allowing for prolonged periods of operation.

Thin-film cells are solar cells that use a direct bandgap semiconductor(like CIGS, CdTe, or a-Si:H) that absorb light more efficiently than indirect bandgap materials like crystalline silicon. This allows for much thinner layers and higher optical absorption. Due to the nature of the material used, the panels produced tend to have good flexibility, leading to superior integration with curved surfaces (Chopra, Paulson, and Dutta 2004). The efficiency of thin-film panels commercially available at this moment is on the lower end, from 7-13% (Richardson 2024). There is, however, a lot of research being conducted in this field, with the aim of improving the efficiency of such panels. One of the leading motivations of this research is the flexibility these panels provide in terms of areas of application.

Conclusion

With various solar panels have various disadvantages and advantages, and the complexity of accounting for their operational behaviours under varying conditions, including temperature, shading, radiation resistance, operation time and cooling requirements, makes it more practical, for the simplification of this research, to focus on efficiency percentages. Thereby, ignoring the nuances of performance in certain scenarios.

Technology	Efficiency (%)	Advantages	Disadvantages
Monocrystalline Si	High (20-22%)	High efficiency, durable, long lifespan	Heavy, rigid, poor shading tolerance
Polycrystalline Si	Moderate (15-17%)	Cheaper, decent efficiency, diffuse light	Less efficient, rigid, heat-sensitive
GaAs	Highest (29.1%)	Highest efficiency, lightweight, durable	Expensive, limited availability
Thin Film (CIGS/CdTe)	Low to Moderate (10-12%)	Flexible, lightweight, good in low light	Lower efficiency, less proven

Table 1: Comparison of Solar Cell Technologies

3.8 Literature Review of Similar Analysis on Competition-Based Solar Cars

Introduction

This section aims to look at the various studies conducted in the field of solar car aerodynamics as well as the aerodynamics of Shell Eco-marathon cars. This literature review aims to define the scope of the research area, summarising the key theoretical frameworks identified and the methodologies and findings of the relevant studies.

Dream Solar Car

One of the fundamental studies done in the field of solar car aerodynamics was by (Ozawa, Nishikawa, and Higashida 1998). The study highlights the development of the design whilst considering the aerodynamic properties. It demonstrates the importance of reducing aerodynamic drag to improve efficiency and shows the contribution of the aerodynamic properties of solar cars to the overall performance. It looks at methods to increase the energy generation capacity of the vehicle as a function of the outer shell design. The main metric being analysed is the drag area, which is the product of the drag coefficient and its reference area. The study looks at specific design modifications and how they contribute effectively to drag reduction as well as their impact on stability.

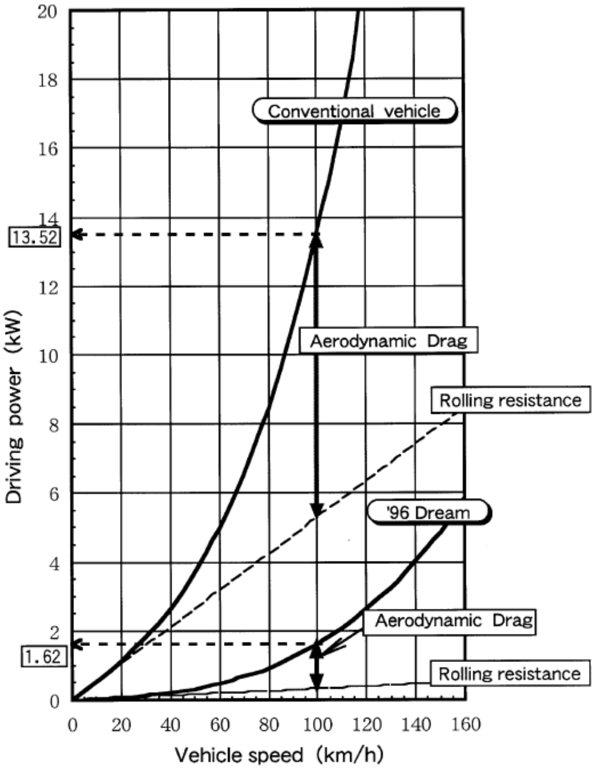


Fig. 17: Driving Power of Dream Solar Car in comparison with a conventional vehicle(Ozawa, Nishikawa, and Higashida 1998)]]

The iterative analysis was done using CFD(Computational Fluid Dynamics) simulations, a well refined mesh with 850,000 cells was used to construct a detailed model, to simulate the incompressible flow around the car by solving the Navier-Stokes equations. Results from the CFD simulations were incorporated into the design process, and refinements were made to the design based on it.(Ozawa, Nishikawa, and Higashida 1998) validates the results from the CFD simulations using wind tunnel testing. The verification and validation of the results showed that, through the iterative design process, the new design achieved an 11 % reduction in the drag area value of C_dA. This was achieved through thorough refinement of the body shape, especially around the wheels, and led to a total resistive power of only 1.62 kw.

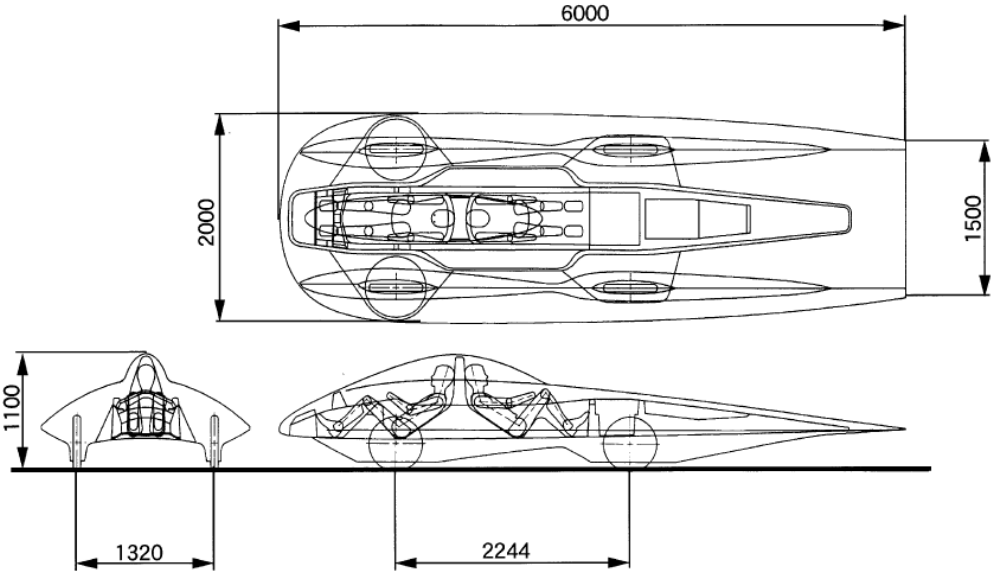


Fig. 18: Final Dream Solar Car dimensions (Ozawa, Nishikawa, and Higashida 1998)

This study is mainly conducted by looking at the constraints provided by the rules of the World Solar Car Challenge, which is why it is not straightforward to directly apply all of the design concepts introduced in the study to a Shell Eco-marathon car. Mainly due to the stark contrast in outer shell shapes seen in Shell Eco-marathon cars and World Solar Car Challenge cars. The study, however, introduces important parameters and an iterative design methodology that can be useful whilst developing a design for a Shell Eco-marathon car, as well as the application of CFD as a tool to aid design.

Shell Eco-marathon drag reduction

Building on the work of previous Shell Eco-marathon cars helps in establishing a foundational framework for developing steps in the design methodology to optimise the outer shell. One such study is by (Cieśliński et al. 2016). This study looks specifically at the fundamentals of drag minimisation, relating to the overall geometry of the body. It looks at the splitting down of drag components into friction and pressure drag. The design framework used integrates aerodynamic performance with vehicle design constraints, which includes driver space and mechanical feasibility for the Shell Eco-marathon cars. It looks at how driver ergonomics needs to be considered whilst developing the outer shell design, and explores how the rules of the Shell Eco-marathon affect the design of the outer shell.

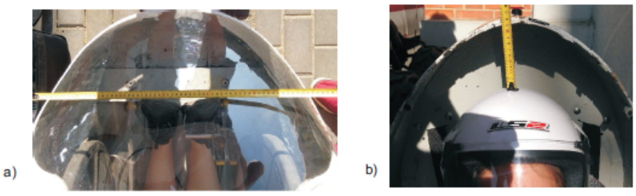


Fig. 19: Measurement of existing car with driver for optimisation (Cieśliński et al. 2016)

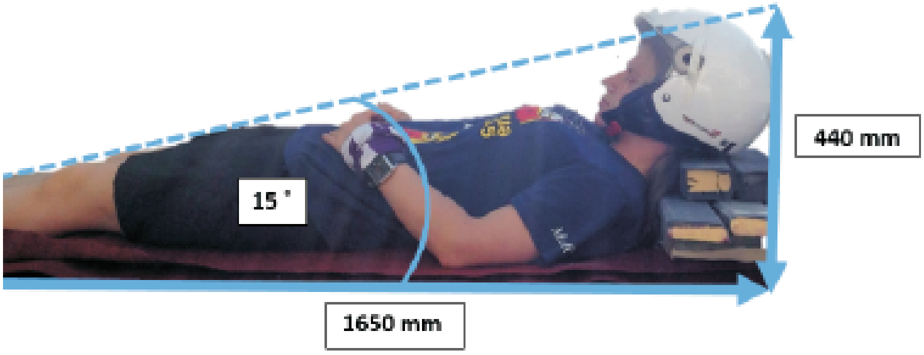


Fig. 20: Angular position of driver's seat to reduce frontal area (Cieśliński et al. 2016)

After having conducted ergonomics studies, the ideal angular position of the driver's seat was found; this configuration allowed for as less as frontal area as possible, whilst keeping in mind the visibility and steering operational constraints of the driver.

With the CAD model developed, a CFD study was conducted on the geometry, using ANSYS Fluent. A tetrahedral mesh was used due to the irregularity of the surface of the vehicle. The mesh was refined in regions closer to the vehicle body to account for the adverse gradients experienced in these regions. The $k - \omega$ SST turbulence model was used, as it provides good results for a diverse range of flow properties.

As a measure of solution quality, the study by (Cieśliński et al. 2016) looks at the reliability of the CFD solution for the flow structure in the boundary layer region. This is done by evaluating the Yplus number y^+. Which is a non-dimensional wall distance for wall-bounded flows. This value depends on the refinement of the mesh in the boundary layer region. It is given by :

$$y^+ = \frac{u_\star y}{\nu} \tag{3.47}$$

Where $u_\star$ is the velocity at the nearest wall,y is the wall distance, and ν is the kinematic viscosity at the wall. With an inlet velocity of 25 km/h, considering the surface of the vehicle as a no-slip surface, and the floor as a moving wall with the same velocity in the opposite direction, realistic conditions were simulated with the use of these boundary conditions.

The simulations looked at various designs, and the results were analysed considering drag area as the main metric. The study concluded that the drag force acting on the vehicle only ranged from 0.69 N to 1.39 N. This is a very low value, suggesting that there could be an error in the methodology or analysis conducted. However, the study provides information on how to develop the design using various tools and methodologies discussed.

PAC Car 2

One of the benchmark studies that looks at Shell Eco-marathon as a whole is the study by (Santin et al. 2007), a team from ETH University. Their work focuses on the development of the team's Eco-marathon car in 2005. The car was engineered to maximise fuel efficiency and showcase innovative design. It set a world record at the time, achieving an equivalent mileage of 5,385 km per litre of petroleum. The car was powered by a hydrogen fuel cell system. The team worked on the body and design of the PAC CAR using an iterative process, developing a teardrop-based geometry, combining the design with wind tunnel trials and CFD analysis. The team developed 4 geometries, improving each design by mainly looking at improving the aerodynamic performance of the wheel fairings and changing the yaw of the vehicle.

Shape number	*Perspective*
1.1	
1.2	
1.3	
2	

Fig. 21: Iterative design process of body shapes (Santin et al. 2007)

The final results obtained from the analysis of shape number 2 are as follows:

Table 2: The aerodynamic characteristics of the PAC-Car II.

Parameter	**Value**	**Unit**
Frontal area (from Shape 2, scale 1:1)	0.254	m^2
c_X (from PAC-Car II wind tunnel trials)	0.075	-
Cruising speed	30	km/h
Vehicle length	2.8	m
Re_L	1.6×10^6	-

The geometry used here has been used as an inspiration for the development of the final design in the study.

Aerodynamic analysis of a Solar Car using STAR-CCM+

Another critical study which looks at the integration of CFD with an iterative design methodology is that done by (Fourie 2016). This study looks at beginning the design process by selecting an aerofoil as a baseline. The author makes use of the extensive NACA database of aerofoils, particularly the NACA 6000 series of aerofoils. These are a set of aerofoils designed to maintain laminar flow throughout the upper and lower regions of the aerofoil. They have a slight camber that generates a bit of lift and assists in reducing the resistance to rolling.

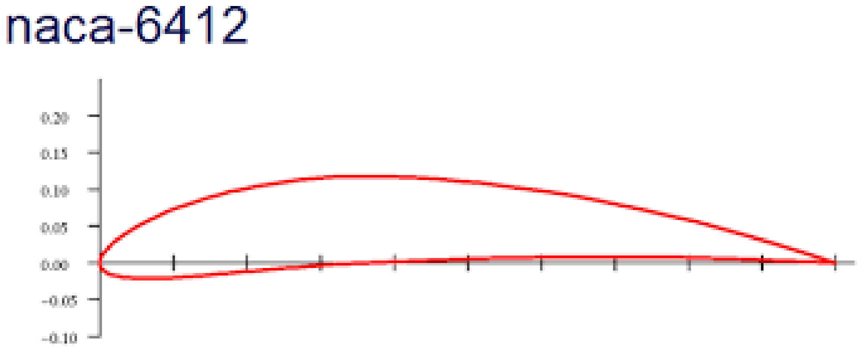


Fig. 22: Example of a 6000 series aerofoil

This paper looks at using XFLR5, a popular analysis tool for 2D aerofoils. It is a low-level tool, used for quick estimates of aerofoil performance. In this study, the author analyses aerofoils that may serve as reference models for the main body of a solar car.

After having found a suitable aerofoil to be used as a baseline for the main body, the author looks at developing a design for the wheel fairings, and subsequently conducts a thorough CFD analysis on the design chosen. The study looks at the combinatory effects of the wheel fairings, and looks at how not only the shape of the wheel fairing affects fluid flow, but also how the orientation and the integration of the fairings change the overall aerodynamics. The CFD analysis was done on STAR-CCM+ using the $k - \omega$ SST turbulence model, along with the $\gamma - Re_\theta$ transition model. Although CFD models struggle at predicting separation in fluid flows, the $\gamma - Re_\theta$ model looked at previously in the literature review is used. This allows for better prediction of separation since the flow velocities result in a Reynolds number that is close to the transitional Reynolds number for external surface flows, as seen in flows over solar car outer shells.

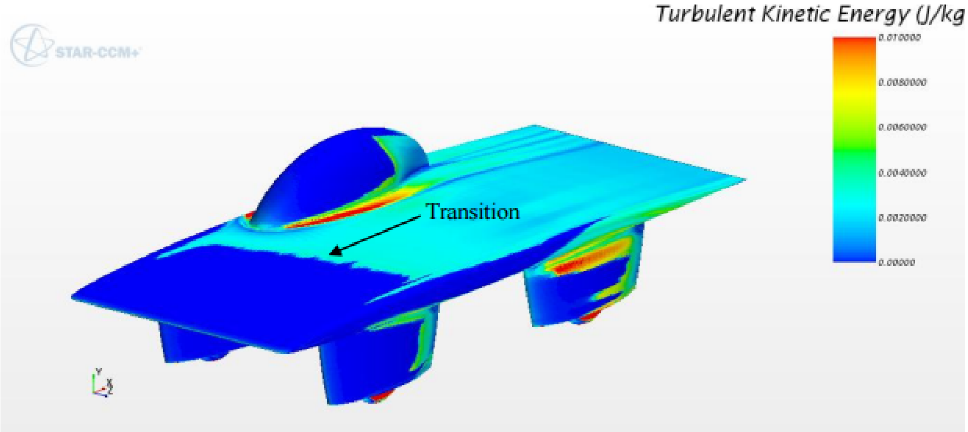


Fig. 23: Prediction of transition depicted by the $\gamma - Re_\theta$ model (Fourie 2016)

In the conclusion, the author states that, through this iterative process, integrating design with CFD analysis led to a reduction of 5 % in the drag coefficient, compared to a previous design used for benchmarking and validation. The study also compares a 3-wheel and 4-wheel design, with results showing that the 3-wheel design was more aerodynamically efficient. The author recommends using a parametric CAD setup for faster analysis. This allows for easy modification of key design parameters such as fillet radii, body length and controlling the curvature of the main body to adjust the profile of the aerofoil. This allows a mathematical control over the geometry, which can be seamlessly integrated into CFD simulations, allowing a systematic and logical workflow for design optimisation.

3.9 Conclusion

The initial sections of this literature review look at establishing a comprehensive theoretical and methodological foundation for the aerodynamic analysis and design process of the outer shell. Particularly focusing on the fluid flow properties that occur and their modelling through the application of Computational Fluid Dynamics(CFD).

The section on aerodynamic drag properties highlights the five main contributors to drag: flow separation, skin friction, boundary layer pressure loss, induced drag, and interference drag. The importance of minimising these components, particularly separation and skin friction drag, has been highlighted. The analysis draws on the foundational works by (Carroll 2003) and (Tamai 1999), advocating for the use of streamlined, aerofoil-based profiles to reduce drag and maintain attached flows. The importance of understanding the interdependence between body shape, flow separation, and the formation of a resulting low-pressure is emphasised.

Drag area as a metric has been introduced to refine the evaluation of aerodynamic efficiency. By considering the drag coefficient and its reference area, the review distinguishes between the use of frontal area and planar area. This approach ensures that the chosen metric accurately reflects the dominant drag mechanism.

A detailed exploration of fluid flow properties, critically important to the drag force acting on the outer shell, has been carried out. Particularly, the behaviour of the boundary layer provides critical insight into the transition from laminar to turbulent flow and the conditions leading to flow separation. The review underscores the significance of maintaining a laminar flow over the majority of the surface, as the formation of turbulent boundary layers leads to an exponential increase in skin friction drag.

The section on CFD as a design tool dives into the numerical and computational models that are used in aerodynamic analysis. The review outlines the use of the Navier-Stokes equations, the Finite Volume Method, the discretisation and solving mechanisms, and the implementation of the various turbulence models. Each model's strengths and limitations are discussed within the context of flows over a solar car outer shell, with particular attention to their ability to predict flow features important for drag force assessment.

The review highlights the critical importance of CFD in the overall iterative methodology. Its ability to provide detailed analysis of pressure and shear forces throughout the geometry, allowing accurate prediction of total drag and identification of design improvements.

The final section of the literature review systematically explores the aerodynamic and design methodologies applied to various competition-based cars. This includes looking at solar cars that have competed in the World Solar Car Challenge, as well as hyper-efficient vehicles competing in the Shell Eco-marathon. The pioneering work on the Dream Solar Car by (Ozawa, Nishikawa, and Higashida 1998) demonstrated the critical role of aerodynamic drag reduction in enhancing efficiency. Through the use of an iterative design methodology integrating CFD with design to ensure a design in the drag area, the study underscored the value of such a computational approach in the design process. Although the constraints of the car are very different from those of the Shell Eco-marathon, the iterative methodology provides a framework to work with.
Building on this, the work done by (Cieśliński et al. 2016) on research specific to Shell Eco-marathon vehicles emphasised the importance of balancing aerodynamic performance with driver ergonomics and competition guidelines. The use of advanced CFD techniques and turbulence modelling enabled the detailed analysis of drag components and provided the ability to make informed design decisions that optimise both performance and practicality.

The review on PAC-Car 2, looking at the study by (Santin et al. 2007) inspired the final geometry developed in this study. The study focuses on the integration of wheel fairings into the design.

Additionally, the work by (Fourie 2016), highlights the benefits of using an aerofoil as a baseline to being the design process. The study integrates parametric CAD modelling with CFD turbulence and transitioning modelling, enabling precise control over geometry and prediction of flow features.
In summary, the literature establishes a robust foundation for the design and optimisation of Shell Eco-marathon cars. It also lays a robust foundation for the subsequent analysis and design of the vehicle, integrating theory, advanced computation, and practical design considerations.

One of the main research gaps identified in the literature review is the lack of integration in the analysis, where both solar panel area and power generation were not considered alongside aerodynamic optimisation of the outer shell design. In most cases, the available solar panel area is determined solely by the aerodynamic considerations, and the power generated is a secondary outcome rather than an integral part of the design process. The methodology proposed in this study aims to address this gap by incorporating solar panel power output as a key parameter during the design phase, ensuring a more holistic and performance-driven design approach.

4 Methodology

4.1 Introduction

This section describes the development of an iterative methodology, which is based on key studies identified during the literature review. The methodology incorporates proven approaches and best practices from previous research to facilitate a systematic and efficient design process. By continuously analysing and incorporating feedback into the design, this approach aims to optimise the vehicle's aerodynamic performance while considering practical constraints identified in the literature review.

4.2 Reviewing Regulation and Review of Previous Works

To start designing the outer shell, it is of utmost importance to consider the dimensional constraints due to the rules and regulations of the Shell Eco-Marathon. Adhering to the rules and regulations from the outset guides the design process and helps avoid costly redesigns that could occur later on. These regulations also influence key aerodynamic decisions, as the outer shell is to be optimised according to the constraints provided.

4.3 Aerofoil Selection and Analysis

As suggested by the various studies, using an aerofoil as a baseline for the profile of the main body is an effective starting point for developing the final geometry. This was done by conducting detailed analysis of various NACA 6000 series aerofoils of 2.75 m length, using XFLR5 which is a tool that uses semi-empirical correlations to predict the laminar-to-turbulent transition location on panel methods. It models the turbulent boundary transition layer using the boundary layer theory, which accounts for the velocity profile and skin friction to account for turbulent effects.

An aerofoil was then selected from the results provided by the XFLR5 data. Similarly, the profile of the wheel fairings were designed based on NACA aerofoil profiles. A review of symmetrical NACA 6000 series aerofoils was conducted to identify options with favourable aerodynamic properties for the design, whilst also considering the geometry and integration with the wheels. Ensuring smooth airflow and minimal drag around the wheel fairings.

4.4 Conceptual Geometry Generation

After selecting an aerofoil using the aforementioned methods, an initial geometry was to be developed on SOLIDWORKS. The first geometry developed took inspiration from World Solar Car Challenge cars, featuring a wide aerofoil-based design and a 3 wheel chassis. As per the study by (Fourie 2016) with an ellipsoidal canopy for the driver to view out of. This outer shell design was developed using inspiration from the design developed by World Solar Car

Challenge teams. The profile of the wheel fairings was also selected based on experimental data conducted on NACA aerofoils. This design focuses on prioritising solar panel area and is more tuned towards the regulations of the World Solar Car Challenge than the Shell Eco-marathon.

4.5 CFD analysis of conceptual outer shell

To look at the aerodynamic properties of the developed outer shell, an initial CFD analysis was conducted using STAR-CCM+. The main aim was to look at the aerodynamic properties of the vehicle at different speeds to obtain values of the drag force acting on the vehicle. The drag values obtained here were to be used as a component of the resistive forces to create a comparison between the power generated by the solar panels and the resistive forces.

When looking at the Reynolds number of the flow to assess if the flow will be turbulent or laminar, it can be seen that for a length scale of 2.75 metres, considering the upper bound of a Reynolds number of 5×10^5 as the turbulent Reynolds number, it is evident that turbulence occurs for velocities as low as 2.67 m/s. Since Shell Eco-marathon cars operate at velocities ranging from 5 m/s to 16 m/s, the flow over the outer shell will be within the turbulent regime. Therefore, turbulent effects will have a significant effect on the aerodynamic behaviour of the vehicle.

Given that the flow is predominantly turbulent, a turbulent flow model is necessary for accurate simulation, therefore, a RANS (Reynolds-Averaged Navier-Stokes) model, as mentioned in the literature review, is used to solve the Navier-Stokes equations, modelling turbulence using the Spalart-Allmaras model mentioned earlier. As the model was to be run through various inlet velocity conditions, leading to multiple simulations, using the Spalart-Allmaras model proved to be more computationally efficient due to it being a one-equation model.

In terms of the meshing strategy used a trimmed cell mesher is used to mesh the domain and outer shell geometry. The trimmed cell mesher generates hexahedral meshes that are shaped and oriented to follow the path of the flow, which increases the computational efficiency of the process. The trimmed cell mesher also allows for meshing of complex geometries, although it could be argued that a tetrahedral or polyhedral mesh works better for curved surfaces, the trimmed cell mesher can conform closely to curved or closed boundaries, ensuring smooth transitions (Siemens Digital Industries Software 2024a).

4.6 Comparison of Resistive Forces with Power Generated

From the above CFD study on the conceptual geometry, drag force values for various operational velocities were obtained. These drag forces were combined with other resistive forces, such as rolling resistance, mechanical inefficiency of the power being generated, to provide a total resistive force acting on the body of the vehicle. The rolling resistance was modelled using the SAE J2452 model (SAE International 2017), which is an empirical model developed to accurately predict the rolling resistance of tyres in real-world scenarios. This is done by accounting for the vehicle speed, tyre load and inflation pressure.

$$F_{rr} = \left(A + B \cdot v + C \cdot v^2\right) \times P^{\alpha} \times L^{\beta} \tag{4.1}$$

where:
F_{rr} = rolling resistance force
v = vehicle speed
P = tyre inflation pressure
L = load on the tyre
A, B, C, α, β = empirical coefficients based on tyre testing.

```python
def rolling_resistance(mass, speed):
    if speed == 0:
        return 0

    load_per_tire = (mass * GRAVITY) / 4
    rolling_resistance_force = (A_COEFF + B_COEFF * speed + C_COEFF *
       speed**2) * \
                               (TIRE_PRESSURE_KPA**ALPHA) * (
                                  load_per_tire**BETA)
    return rolling_resistance_force * 4
```

Listing 1: Rolling Resistance Function from the Python Code

```python
def mechanical_efficiency(speed):
    return ETA_MAX - (ETA_MAX - ETA_MIN) * math.exp(-ALPHA_EFF * speed
       )
```

Listing 2: Mechanical Efficiency Function

This implements the efficiency equation by (Khajepour, Fallah, and Goodarzi 2014):

$$\eta_{mech} = \eta_{max} - \left(\eta_{max} - \eta_{min}\right) \cdot e^{-\alpha v} \tag{4.2}$$

where:

- η_{mech} is the mechanical efficiency
- η_{max} is the maximum efficiency (0.90)
- η_{min} is the minimum efficiency (0.80)
- α is the efficiency coefficient (0.2)
- v is the vehicle speed

This was integrated through a Python script, and the total resistive forces were compared with the power generated by the solar panel area. For the load on the tire, the weight of the vehicle was assumed to be 140 kg, with the outer shell material assumed to be carbon fibre, and using rough estimates of the weight of other components and the driver. Due to the nature of the curvature of the solar panels, certain assumptions had to be made regarding the amount of solar irradiation being incident on the panels. For the purpose of simplification, angle multipliers were used, and the solar panel region was split up into regions inclined at an inclination of 15, 30, 45 and 60 degrees. The temperature dependence of efficiency was ignored and was assumed as a constant at 21%, assuming the usage of mono-crystalline silicon panels. The irradiation from the sun was assumed to be a constant of 1000 W/m^2 directly overhead the car.

The total power needed is calculated by combining the drag force, rolling resistance, and mechanical efficiency:

```python
def compute_power_needed(drag_force, speed, mass):
    F_rolling = rolling_resistance(mass, speed)
    eta_mech = mechanical_efficiency(speed)
    total_resistive_force = drag_force + F_rolling
    power_required = (total_resistive_force * speed) / eta_mech
    return power_required
```

Listing 3: Power Requirement Function

The power requirement is calculated using:

$$P_{required} = \frac{(F_{drag} + F_{rolling}) \cdot v}{\eta_{mech}} \tag{4.3}$$

where:

- $P_{required}$ is the total power required
- F_{drag} is the aerodynamic drag force
- $F_{rolling}$ is the rolling resistance force
- v is the vehicle speed
- η_{mech} is the mechanical efficiency

The total electrical power generated by the solar panels is calculated by considering different panel inclinations and their respective efficiencies:

```python
def maincalc():
    # User input for solar panel area
    total_area = float(input("Enter the total surface area of the solar panels (in m ): "))

    # Distribution of panel areas at different angles
    area_distribution = {}

    # Calculate total electrical power
    total_elec_power = 0.0
    for angle, area in area_distribution.items():
        multiplier = ANGLE_MULTIPLIERS[angle]
        total_elec_power += SOLAR_IRRADIANCE * EFFICIENCY * area * multiplier
```

Listing 4: Solar Power Generation Calculation

The total electrical power generated is calculated using:

$$P_{total} = \sum_i (G \cdot \eta_{PV} \cdot A_i \cdot k_i) \tag{4.4}$$

Where,

- P_{total}: total power output
- G: solar irradiance (W/m^2)
- η_{PV}: photovoltaic (PV) panel efficiency
- A_i: area of the i-th PV panel (m^2)
- k_i: angle correction factor for the i-th panel
- $\sum_i$: summation over all PV panels

Through the use of this Python script, the efficiency of the design was determined for various operational speeds, which allowed the identification of the most efficient operating velocity for subsequent CFD analysis.

4.7 Final Design Development using Surface Modelling

The final design was developed considering the optimised idealised geometry that was developed in the above section. Specific considerations and modifications had to be made to the developed teardrop shape, such that it would conform to the rules and regulations of the Shell Eco-marathon and was developed taking inspiration from the PAC 2 car mentioned in the literature review. The final outer shell was designed to maintain the profile of the aerofoil and teardrop shapes while accommodating the spatial requirements of the various subsystems. Three different geometries were developed, all adhering to regulatory constraints.

The three outer shell designs differ in length and width. The initial geometry acts as a baseline, while the second design has a wider track width, resulting in an increased overall width. The third design has a longer wheelbase, increasing its overall length. Both the second and third outer shell designs provide a larger solar panel area compared to the first design. The goal is to assess whether these design modifications justify the additional power generated.

Design 1

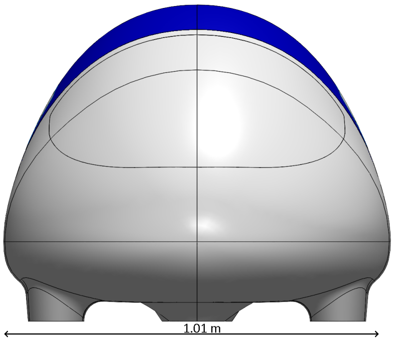


Fig. 24: Design 1 Front View

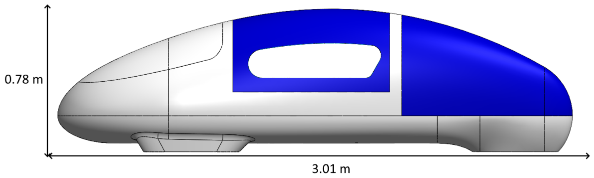


Fig. 25: Design 2 Side View

Design 2

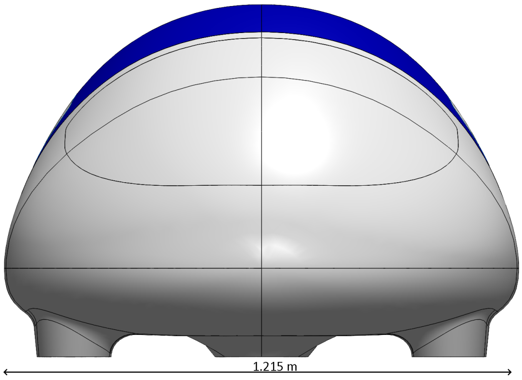


Fig. 26: Design 1 Front View

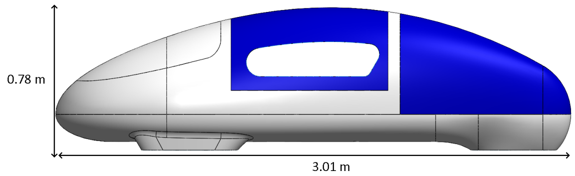


Fig. 27: Design 2 Side View

Design 3

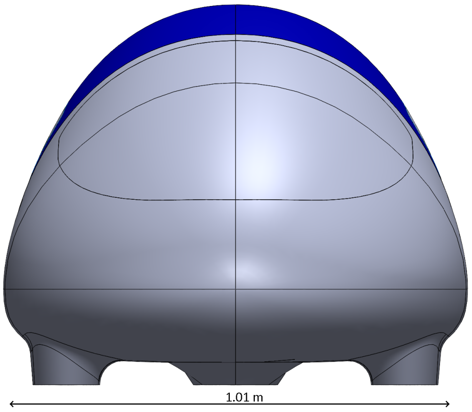


Fig. 28: Design 3 Front View

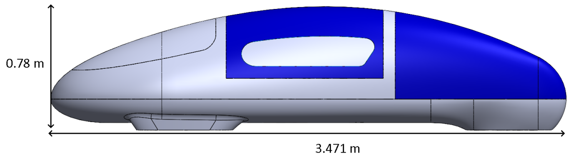


Fig. 29: Design 3 Side View

Table 3: Dimensions of Designs (in metres)

Dimension	**Design 1 (m)**	**Design 2 (m)**	**Design 3 (m)**
Length	3.01	3.01	3.471
Width	1.01	1.215	1.01
Height	0.78	0.78	0.78

4.8 CFD Analysis of Final Designs

CFD model setup:
The 3 CAD models generated were pre-processed to be used by the CFD software, i.e, STAR-CCM+. As a means of simplification and ensuring that the geometry remained watertight, the effects of the wheels have not been taken into consideration. The geometry imported into STAR-CCM+ assumes that the entire wheel is sealed off in the fairings. This allows us to simplify the simulation, ignoring the unsteady turbulent effects caused by the motion of the wheels. A three-dimensional steady-state simulation, with the assumption that the air is incompressible throughout the flow, is conducted. The reason why a steady simulation has been opted for instead of an unsteady simulation is that the key focus of the study is on understanding the drag forces acting on the outer shell. Drag itself is considered as the average resistive force acting on the body when the fluid flowing is at a constant speed. It could be argued that the turbulent flow features that develop about the body are time-dependent, however, these features tend to stabilise after a certain time, allowing the consideration of a steady simulation for the analysis of the outer shell. The incompressibility assumption has been made as the flow velocities are quite low in the simulation, less than a Mach number of 0.3.

Discretisation scheme used:
The discretisation scheme used to discretise the pressure and momentum terms of the RANS equation is the second-order upwind scheme. As mentioned in the literature review, this is a scheme of second-order accuracy, which provides improved resolution (Versteeg and Malalasekera 2007). This scheme ensures that there is no artificial smoothening of gradients, which is significant given that there is a possibility of the formation of adverse gradients in near-wall regions during the simulation of the outer shell.

Solver mechanism applied:
The solver used in the simulations to solve the set of discretised equations is a segregated solver. As mentioned in the literature review, the segregated solver solves governing equations separately and consecutively. For a steady-state incompressible flow simulation, segregated solvers often converge efficiently, which is due to how pressure and momentum are decoupled by the solver. This allows the set of equations to be solved in a computationally efficient manner. The decoupling in the segregated solver is managed by the use of the SIMPLE algorithm, which links pressure and velocity within iterations, by the steps described in the literature review.

Meshing methodology
During the CFD analysis of the conceptual geometry, a trimmed cell mesher was used. However, when the trimmed cell mesher was used to develop hexahedral cells on the set of final shell-based geometries, it struggled to deal with the thinness of the body, leading to an inability to create watertight meshes. The tetrahedral mesher builds the core volume using tetrahedral-

shaped cells. In STAR-CCM+, this is done by using the Delaunay method, which works by iteratively adding points inside the domain to create high-quality tetrahedral elements (Siemens Digital Industries Software 2024b).

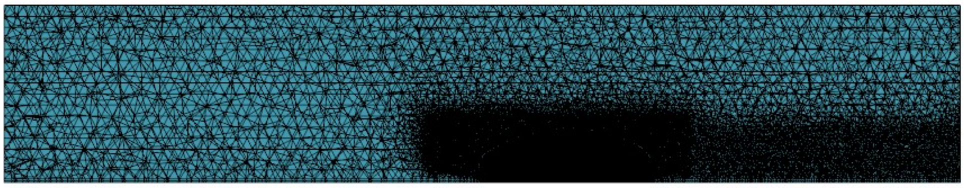


Fig. 30: Mesh Domain

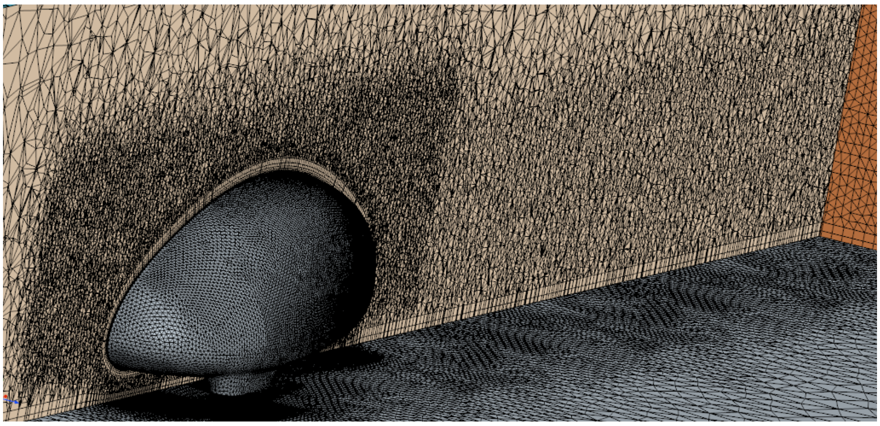


Fig. 31: Mesh refined near the body and wake refinement

The mesh generated closely follows the surface mesh's triangulation at domain boundaries, hence the quality of the tetrahedral volume mesh closely follows that of the surface mesh. The main reason the tetrahedral volume mesher works well on thin surfaces is that, on STAR-CCM+, it can be accompanied by the thin mesher, which identifies areas of the geometry which are thin and generates a prismatic-type mesh, well suited for capturing flows and gradients expected.

The surface remesher tool has been used to improve the overall quality of an existing surface, optimising it for the volume mesh. Prism layers, which are thin, structured layers of the mesh elements stacked normal to the wall surface have been used. Prism layers help in improving the resolution of the boundary layer, allowing the solver to capture the adverse gradients observed in this region. It also ensures that the wall y^+ value is less than 1, which is the critical value needed for accurate turbulence modelling.

A base size of 0.1 m has been used in the CFD studies, this was obtained as a result of the mesh independence study. It is important to note that this base size is only the size of the cells in areas with no mesh control specifications for refinement. The mesh has been refined near the body, allowing the mesh to conform to the complex curved nature of the shape, improving the simulation accuracy, and accounting for better calculation of adverse gradients found closer to the body of the vehicle. Wake refinement has also been applied using STAR-CCM+'s surface refinement option, which allows for greater accuracy in computing the values of parameters observed in the wake region. Improving the density of the mesh in the wake improves the accuracy of drag, due to the drag forces being dependent on flow separation and pressure change, which can be observed in the wake.

Near-Wall modelling

The all y^+ wall treatment has been used to accurately capture the flow behaviour across all regions of the mesh. The all y^+ wall treatment automatically adapts to the mesh resolution by blending low y^+ and high y^+ wall functions, ensuring accurate near-wall modelling irrespective of mesh density. The high y^+ wall treatment assumes that the near-cell wall lies within the log layer of the boundary layer, with a $y^+ > 30$. In this region, the log-law, as mentioned in the literature review, is used to estimate the velocity profile. Whereas in the low y^+ wall treatment, the boundary layer is resolved with a very fine mesh, and no modelling is necessary to predict the flow across the wall boundary.

Boundary Conditions

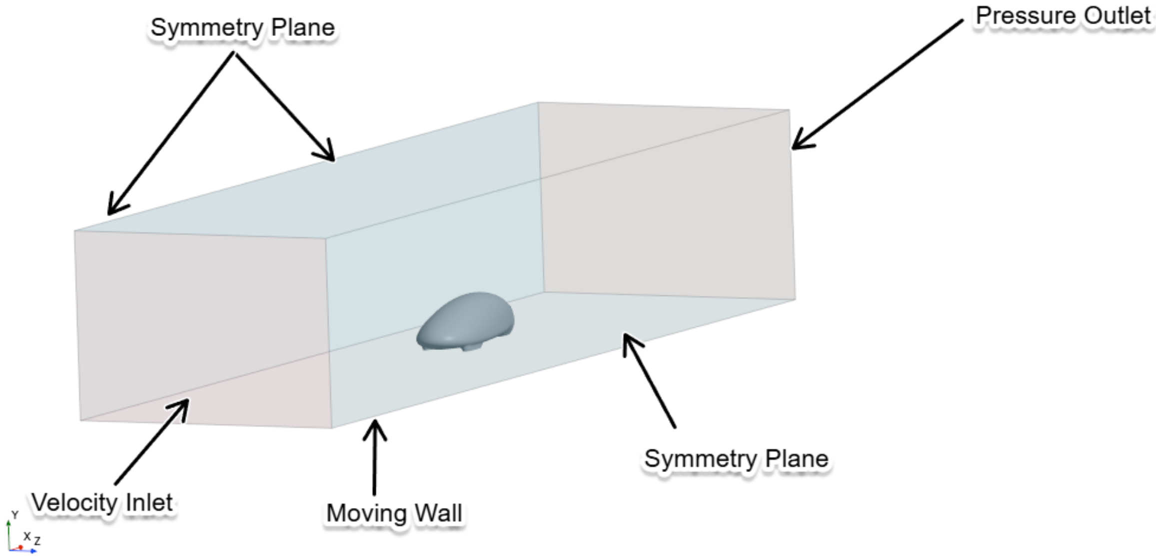


Fig. 32: Computational Domain

The boundary conditions in the simulation were set up to replicate the physical environment and flow behaviour observed in wind tunnel experiments. This means the simulation domain has been split into various parts to closely match the conditions a model would experience in a wind tunnel, ensuring that CFD results are realistic and comparable to experimental data.

The ground was simulated as a moving wall in the STAR-CCM+ simulation. This is to accurately replicate the real-world conditions. This prevents artificial or unrealistic boundary layer development on the ground, accurately capturing the interaction between the car's underbody flow and the road surface. This improves the prediction of aerodynamic forces and flow structures near the ground.

The surface of the car is set as a stationary, non-slip wall. This means that the fluid velocity at the car surface is zero relative to the car, representing the physical conditions that occur when air sticks to the car surface.

The top and side planes have been set as symmetry planes to model the atmosphere, which implies that the flow is uniform and undisturbed far from the vehicle. The symmetry boundary condition means that there is no flow normal to these planes, effectively mimicking an infinite, undisturbed atmosphere.

The inlet boundary is set as a velocity inlet, specifying the incoming flow velocity equal to the vehicle's speed. This effectively replicates the relative motion between the vehicle and incoming air, accurately mimicking real-world conditions. The inlet velocity has been set to 16 m/s, determined from the previous studies conducted in the iterative methodology, as well as from the operational speeds reported for various prototype class Shell Eco-marathon cars. This value reflects the maximum operational speed at which the efficiency of the car remains optimal.

The values used for setting up the boundary conditions are shown in Table 4:

Table 4: Boundary Conditions

Property	**Value**
Air	
Density	1.18415 kgm^{-3}
Dynamic Viscosity	$1.85508 \times 10^{-5}\ Pa \cdot s$
Inlet Velocity	16 m/s
Turbulence Parameters	
Inlet Turbulence Intensity	0.01
Outlet Turbulence Intensity	0.01
Inlet Turbulent Viscosity Ratio	10
Outlet Turbulent Viscosity Ratio	10
Wall Parameters	
Floor Moving Wall Velocity	-16 m/s
Blended Wall Function: E	9.0
Blended Wall Function: Kappa	0.42

Turbulence modelling

In the simulations, STAR-CCM+ uses the Reynolds-Averaged Navier-Stokes(RANS) equations as mentioned in the literature review. The turbulence model used to solve for the Reynolds stresses in the RANS equations is the $k-\omega$ SST turbulence model. As mentioned in the literature review, this model's blended approach provides good results over the entire flow regime, combining the advantages of the $k-\epsilon$ and $k-\omega$ models. The default parameters from STAR-CCM+ have been used for the turbulence model.

In addition to the $k-\omega$ SST model, γ–Re_θ transitional model has also been implemented. Although the Reynolds number indicates that the majority of the flow is expected to be turbulent, the γ–Re_θ model was included to capture possible regions where transitional effects from laminar to turbulent can be witnessed, particularly at leading edges and areas with favourable pressure gradients. This is due to the major effect that even small regions of laminar flow have on overall drag properties (Fourie 2016).

4.9 Comparative analysis of resistive forces and power generated

This section looks at the improvements made to the methodology described in section 4.6 to perform a comparative analysis of resistive forces and power generated. To accurately calculate the power generated by the solar panels, the panel surfaces were meshed using SOLIDWORKS. This meshed surface was then imported into the Python code using an STL library, where each mesh cell was processed to determine its unit normal vector. With the assumption that the sun was directly overhead, the angle of incidence of the solar rays for each cell was calculated. This allows for a more accurate estimation of the irradiation incident on the solar panels. The power generated by each cell was computed and summed up to provide the total power output of the solar panels.

This analysis was conducted on the 3 designs to calculate the power generated by the varying solar panel layouts on each design.

The blue region indicates the solar panel region:

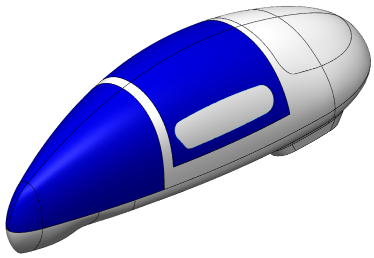


Fig. 33: Isometric view of outer shell

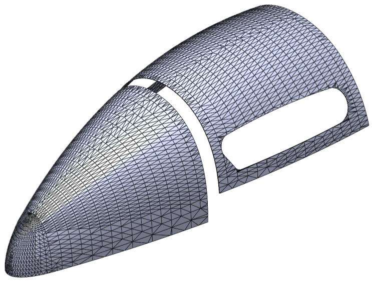


Fig. 34: Meshed solar panel surface

```
    def pv_power(I_solar, area, normal, sun_vec, efficiency):
    """Calculate electrical power for a single triangle."""
    # Ensure normal is a unit vector if not already
    norm_normal = np.linalg.norm(normal)
    if norm_normal == 0: return 0.0 # Avoid division by zero for degenerate triangles
    unit_normal = normal / norm_normal

    angle_cos = np.dot(unit_normal, sun_vec) # sun_vec should also be unit vector

    # Power is generated only if the surface faces the sun (angle_cos > 0)
    if angle_cos <= 0:
        return 0.0

    absorbed = I_solar * ALPHA_PV * TAU_PV * area * angle_cos
    power = absorbed * efficiency
    return power
```

Listing 5: Power calculation function for one cell of the mesh

The solar power generation is calculated using the `pv_power` function:

$$P_{\text{triangle}} = I_{\text{solar}} \cdot \alpha_{\text{PV}} \cdot \tau_{\text{PV}} \cdot A \cdot \max(0, \vec{n} \cdot \vec{s}) \cdot \eta \tag{4.5}$$

The mesh STL file input into the code is to be processed. The unit normal of each triangular cell and its area are calculated using the following function:

```
def read_stl(filename, scale=1.0):
    """Reads STL, scales vertices, returns triangle areas and unit normals."""
    if not os.path.exists(filename):
        print(f"Error: STL file not found at {filename}")
        return None, None
    try:
        your_mesh = mesh.Mesh.from_file(filename)
    except Exception as e:
        print(f"Error reading STL file {filename}: {e}")
        return None, None

    triangles = your_mesh.vectors * scale
    # Use the normals directly from the mesh object, they should be unit normals
    normals = your_mesh.normals
```

```
    areas = []
    for i, tri in enumerate(triangles):
        # Calculate area using cross product for robustness
        v1 = tri[1] - tri[0]
        v2 = tri[2] - tri[0]
        cross_prod = np.cross(v1, v2)
        area = 0.5 * np.linalg.norm(cross_prod)
        areas.append(area)

        # Optional: Check if calculated normal matches mesh normal
           direction
        # calc_normal = cross_prod / np.linalg.norm(cross_prod) if np.
           linalg.norm(cross_prod) != 0 else np.array([0,0,0])
        # if np.dot(calc_normal, normals[i]) < 0.9: # Allow for some
           tolerance
        #     print(f"Warning: Triangle {i} in {filename} might have
           inconsistent normal orientation.")

    return areas, normals
```

Listing 6: Mesh STL processing

The processed mesh data allows for the comprehensive calculation of solar power generated across the entire surface of the solar panels. This methodology is implemented through the section of code shown below.

```
for cell_name, efficiency in PV_CELLS.items():
    total_power_watts = 0.0  # Initialize total power for this cell
       type

    # This loop goes through each triangle in the mesh
    for area, normal in zip(areas, normals):
        # Calculate power for single triangle and add to total
        power = pv_power(SOLAR_IRRADIANCE, area, normal,
           unit_sun_vector, efficiency)
        total_power_watts += power

    # Convert total power to kilowatts
    total_elec_kW = total_power_watts / 1000.0
```

Listing 7: Power summed across the entire mesh using the inputted mesh STL file

This analysis was performed for each type of photovoltaic (PV) cell identified in the literature review, with the efficiency parameter tailored to accurately represent the performance characteristics of each panel technology. Specifically, the following PV cell types and their corresponding efficiencies were considered in the simulations:

- Monocrystalline Si ($\eta = 0.21$)
- Polycrystalline Si ($\eta = 0.16$)
- GaAs ($\eta = 0.291$)
- Thin Film ($\eta = 0.11$)

The efficiency values for each cell type were incorporated into the power generation calculations to ensure a realistic comparison of their potential performance under identical operating conditions.

The generated power was compared with the resistive forces. Drag forces for the 3 designs were obtained through the CFD analysis conducted, combined with the formulations for rolling resistance and mechanical inefficiencies mentioned in subsection 4.6, to obtain the total resistive force on the entire vehicle.

The magnitude of the difference between the generated power and resistive force serves as an indicator of the overall efficiency of the outer shell design. This integrated approach, considering both solar panel power generation and aerodynamic performance, enables informed design decisions that guide the overall development of an optimised outer shell design.

5 Results and Discussion

This section presents the results derived from each stage of the iterative design methodology. All the designs developed were done so with careful adherence to the dimensional constraints specified by the Shell Eco-marathon regulations, ensuring that every design is both compliant with the rules and practically feasible. This approach ensured that the final set of designs is fully optimised, providing a robust foundation for meaningful performance comparison and selection.

5.1 Aerofoil Analysis

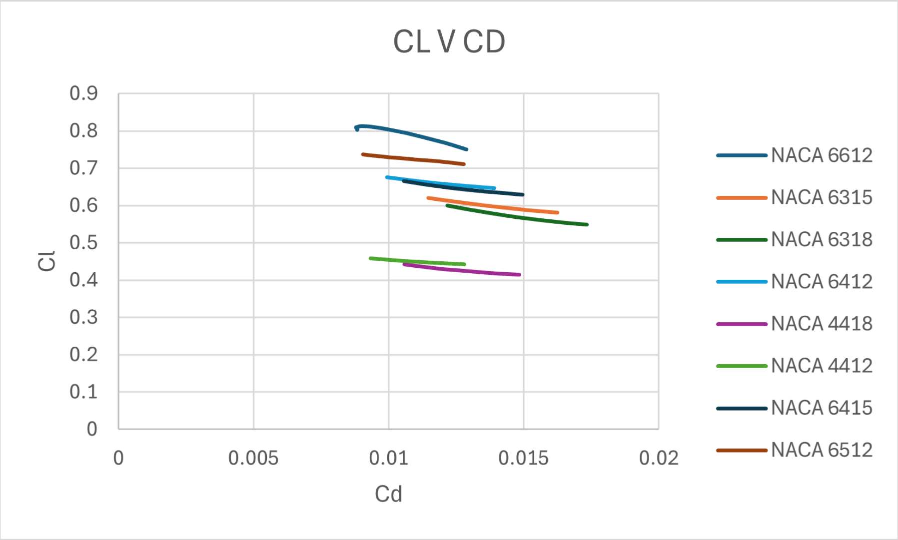


Fig. 35: C_l vs C_d comparison of various aerofoils

The results presented above have been obtained by conducting simulations in XFLR5 on a range of serialised NACA aerofoil profiles for a range of Reynolds' numbers, each of them selected for their low-drag characteristics.

The key parameter to be looked at is C_d(Drag Coefficient); a C_D value as low as possible is most beneficial. In terms of the C_l(Lift Coefficient) value, a value in magnitude as close to zero is ideal. Generating lift is undesirable as it reduces traction, whereas excessive downforce increases drag and reduces efficiency due to the increased rolling resistance. Another key consideration to be made is the geometric profile of the aerofoil itself, in terms of its manufacturability and suitability as a baseline shape for an outer shell.

Based on the results presented in the graph and analysis of selected aerofoils, the **NACA 6412** aerofoil showed the most favourable characteristics for the parameters mentioned above.

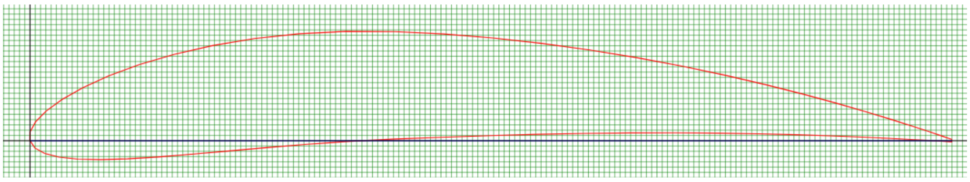


Fig. 36: NACA 6412 Aerofoil (Airfoil Tools 2025b)

For the selection of the wheel fairing aerofoil, a range of options in the NACA 63000 series were considered. These symmetric aerofoils are well-established in aerodynamic literature for their low drag coefficients and favourable flow characteristics, and minimal flow separation. This makes them suitable for applications where both aerodynamic efficiency and stability are crucial. Of the various aerofoils, the NACA 63012 aerofoil was selected as the profile for the wheel fairing, this is due to its comparative increase in thickness towards the leading edge of the aerofoil, making it suitable for it to be integrated into the design of a wheel fairing.

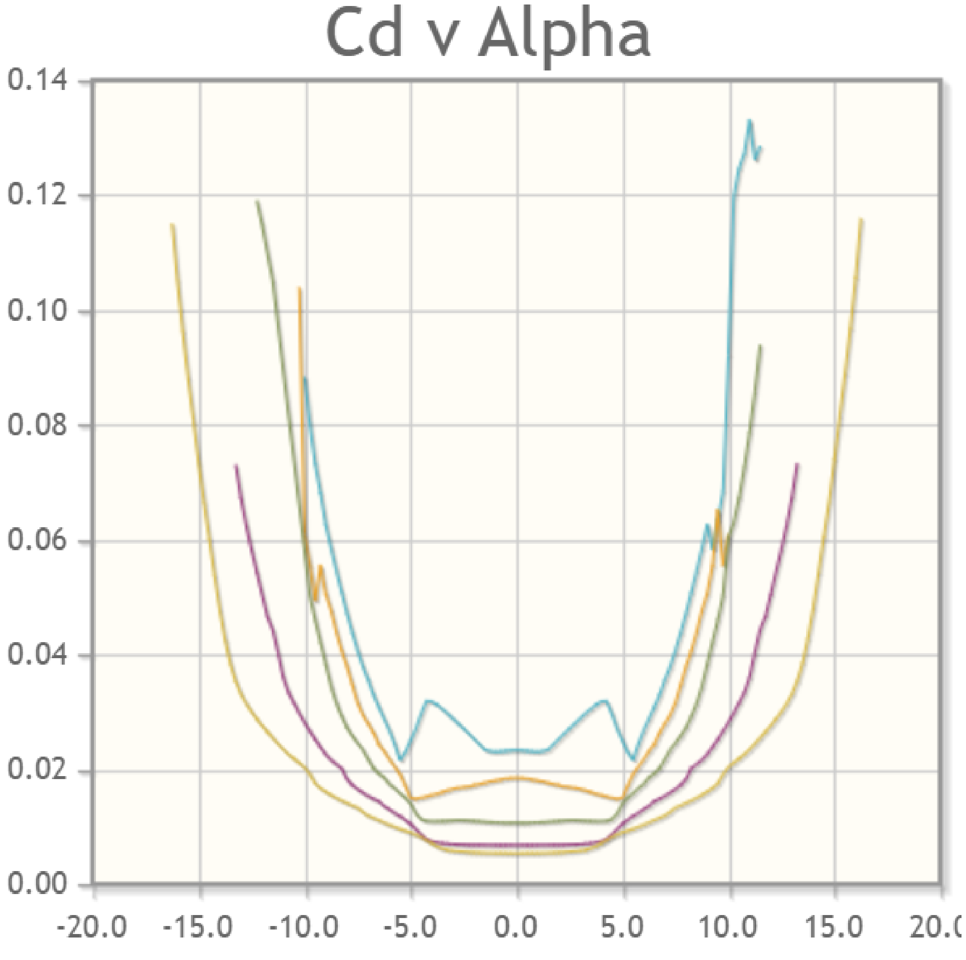


Fig. 37: NACA 63012 C_d at varying angles of attack (Airfoil Tools 2025a)

In Fig. 37 presents the C_d values for varying angles of attack and a range of Reynolds numbers can be seen. For a wheel fairing, the angle of attack is typically 0, at which point the values of C_D are noticeably low, ranging from 0.01 to 0.02. This suggests that the NACA 63012 aerofoil is a suitable choice for the wheel fairing profile, offering excellent aerodynamic efficiency.

5.2 Analysis of initial concept geometry

Building upon the aerofoil selected from the above analysis, a conceptual geometry was developed. This was done by thickening the aerofoil profile height-wise and incorporating a three-wheel fairing design, an approach commonly seen in most Shell Eco-marathon prototype class vehicles, including those highlighted in the literature review.

The dimensions of the outer shell have been set to the maximum allowed by the Shell Eco-marathon prototype class regulations, with a width of 1.2 metres and a length of 3.5 metres. Although the XFLR5 analysis was performed using a length scale of 2.75 metres, the results remain applicable to a design with a length of 3.5 metres.

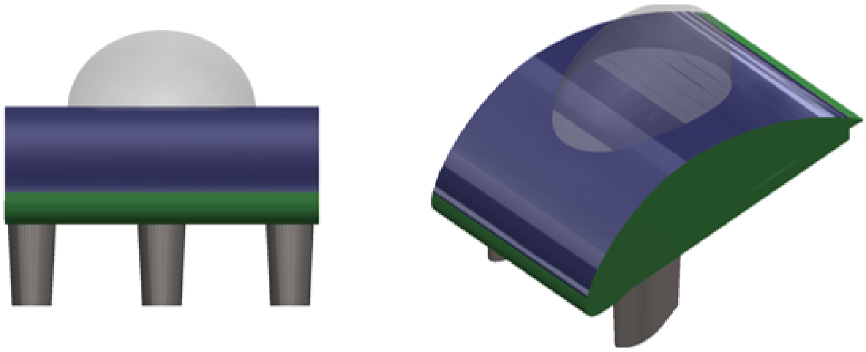


Fig. 38: Conceptual design based on NACA 6412 aerofoil

To assess the aerodynamic performance of the developed geometry, a CFD simulation was conducted to yield values of the drag force acting on the body and drag area.

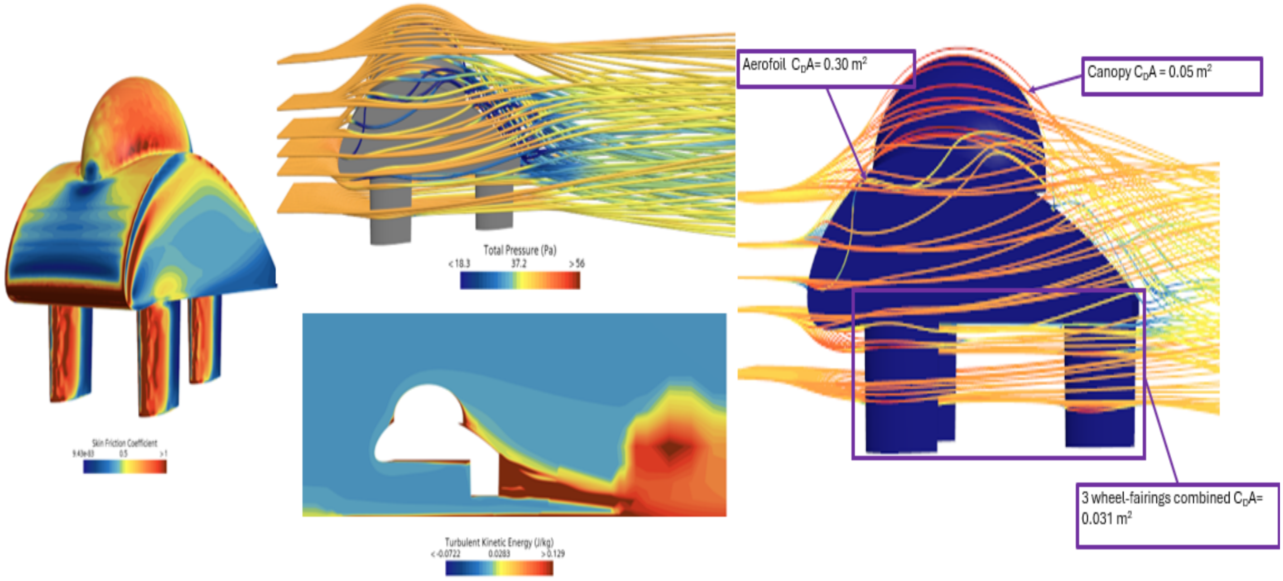


Fig. 39: CFD results from concept design

Table 5: $C_D A_{\text{frontal}}$ Values for Conceptual Geometry Components

Component	$C_D A_{\text{frontal}}$ (m^2)
Aerofoil	0.30
Canopy	0.05
3 wheel-fairings combined	0.031
Total Outer Shell	0.3928

As illustrated in the above CFD output scenes, the initial conceptual geometry is not aerodynamically efficient. It is optimised for maximum solar panel area, leading to a considerable width, which results in a large frontal area. This large frontal area consequently leads to a high drag force. The frontal drag areas are presented as a key metric for analysing the designed conceptual outer shell.

It can be seen that there are several suboptimal design characteristics with this design, due to the presence of sharp edges, a lot of flow separation can be observed, as indicated by the high skin friction coefficient values at the edges, indicating the presence of a turbulent boundary layer. The large frontal area also creates a high frontal pressure, causing drag-induced pressure. Furthermore, the presence of the canopy as an extrusion leads to flow separation and the formation of a turbulent wake. The presence of a turbulent wake leads to a low-pressure region occurring at the wake, further amplifying the pressure-induced drag. The wheel fairings have not been seamlessly integrated with the main body, as evidenced by the absence of fillets to smooth the transition between these components. This lack of aerodynamic refinement at the junction leads to increased interference drag.

The results shown in the above visualisation are from an RANS simulation, where the $k-\omega$ SST turbulence model was used. However, since simulations needed to be conducted across a range of velocities, and taking computational efficiency in mind, the Spalart-Allmaras one-equation turbulence model was used instead to conduct the RANS simulations. Although this model generally struggles with flows with curvature, and tends to underpredict turbulence and flow separation (Cadence-CFD 2024). This means that it also slightly underestimates the drag force acting on the outer shell. Nevertheless, it helps us gain a general estimation of the drag force experienced by the outer shell.

The drag forces obtained for various velocities are integrated into the Python code as shown in section 4.6 to compare the power generated by the solar panels with the resistive power, which includes both aerodynamic drag and rolling resistance, and the power lost due to the mechanical inefficiency of the system.

The results obtained from the Python code are as follows:

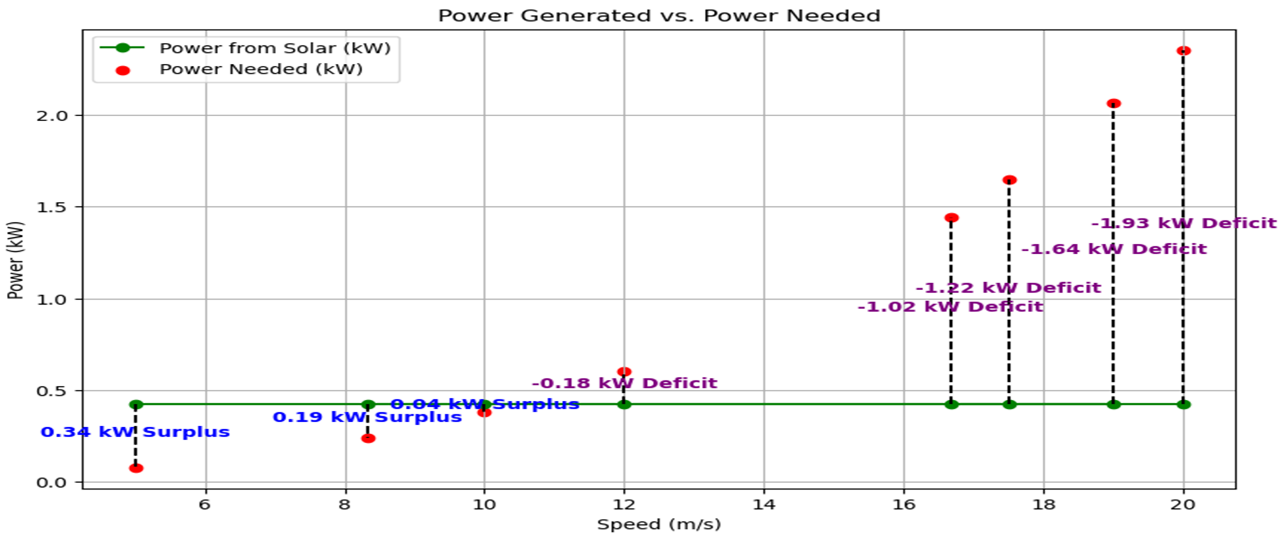


Fig. 40: Results from comparative analysis of resistive forces and power generated on conceptual geometry

It is observed that for velocities ranging from 3-12 m/s, the power generated by the solar panels remains greater than the power needed to propel the vehicle forward. The solar panels produce approximately 0.46 kW of power. However, it is important to note that this value is subject to some uncertainty, as a set of assumptions has been made regarding the orientation of each solar cell, where a certain amount of the surface area of the solar panel has been chosen to be inclined at certain angles.

Nevertheless, this analysis provides valuable insight into the range of optimal velocities from an efficiency viewpoint. By examining the difference between the power generated and the required power to overcome the resistive forces, a comprehensive understanding of the outer shell's efficiency is gained. This aids in integrating solar panel power generation into the aerodynamic analysis.

5.3 Analysis of final surface modelled designs

Transition from conceptual design to final outer shell designs

The conceptual designs were enhanced, incorporating insights from previous teams, as demonstrated by the studies by (Santin et al. 2007) and (Cieśliński et al. 2016). The conceptual designs were modified to 3 specific designs, each of them have been designed keeping the constraints provided by the rules, as well as the design requirements to integrate other subsystems into the design as well. Design 1 represents the initial surface modelled developed, this was done so taking inspiration from the PAC CAR 2, and maintaining the NACA 6412 aerofoil profile from the earlier analysis. Design 2 is a slightly wider variation of this, designed for a greater solar panel area. Whereas in Design 3, a greater solar panel area has been achieved by increasing the length of the outer shell, while keeping the width the same. The height of this design, as specified in Table 3, is governed by ergonomic requirements and has been minimised to achieve the smallest possible frontal area while maintaning an optimal driving position.

The below illustrations shows the results of CFD simulations conducted on these 3 designs based on the methodology mentioned in Section 4.8, these illustrations highlight the specific contours and values that affect the drag force acting on the outer shell. The CFD simulations have yielded the drag forces acting on the outer shells, as well as the corresponding drag areas, as mentioned in the literature review. These metrics will be useful in the subsequent analysis.

CFD results

Parameter	**Design 1**	**Design 2**	**Design 3**
Drag Force (N)	11.490	14.490	10.910
$C_{\mathrm{D,Front}}$	0.113	0.118	0.108
A_{Front} (m^2)	0.669	0.803	0.669
$C_{\mathrm{D,Front}}A_{\mathrm{Front}}$ (m^2)	0.076	0.095	0.072
$C_{\mathrm{D,Planar}}$	0.034	0.035	0.028
A_{Planar} (m^2)	2.244	2.693	2.580
$C_{\mathrm{D,Planar}}A_{\mathrm{Planar}}$ (m^2)	0.076	0.095	0.072

Table 6: Comparison of drag forces and drag area metrics for the three designs

The results presented above summarise the drag force and drag area metrics obtained from CFD simulations for the three designs. Drag force is the most important parameter for the design analysis, as it directly reflects aerodynamic efficiency. The overall energy consumption is highly dependent on the drag force acting on it, since the system needs to overcome the drag force to operate at the given velocity.

Based on the results presented, Design 3 shows the best aerodynamic efficiency, exhibiting the lowest drag force of 10.910 N. Design 2, on the other hand, is the least efficient, with a drag force of 14.490 N acting on it under the same conditions. Design 1 performs comparably to Design 3, with a marginally higher drag force of 11.490 N. These findings indicate that Design 3 offers the most favourable outer shell configuration in terms of minimising aerodynamic drag. The following scalar field visualisations illustrate the spatial variation of key parameters across the entire flow domain, providing insight into how the components of the design contribute to the aforementioned drag values.

CFD results visualisation

Design 1	
Design 2	
Design 3	

Table 7: Side view of C_P contours at the midsection for all designs.

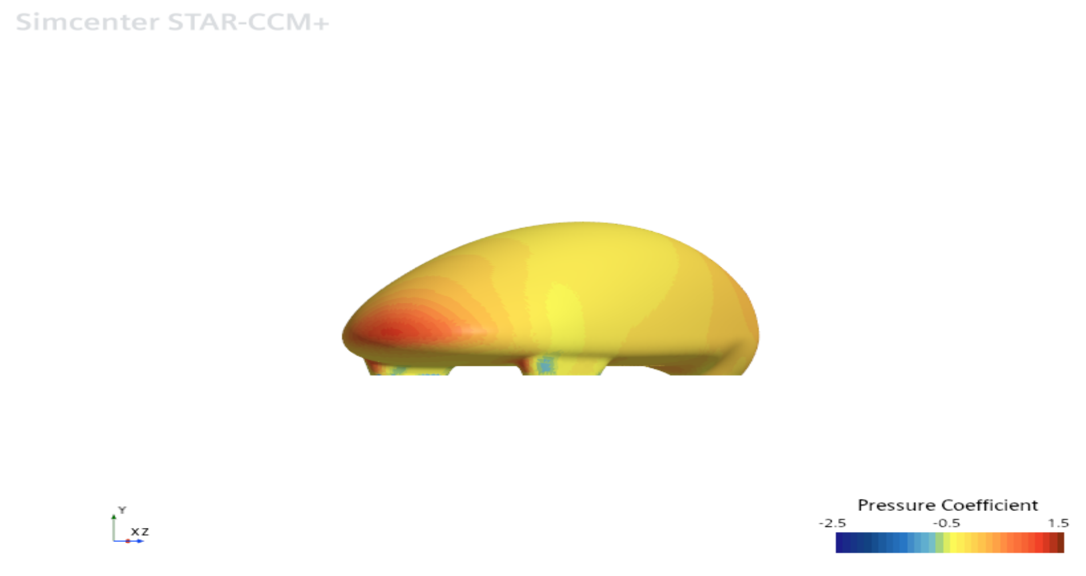
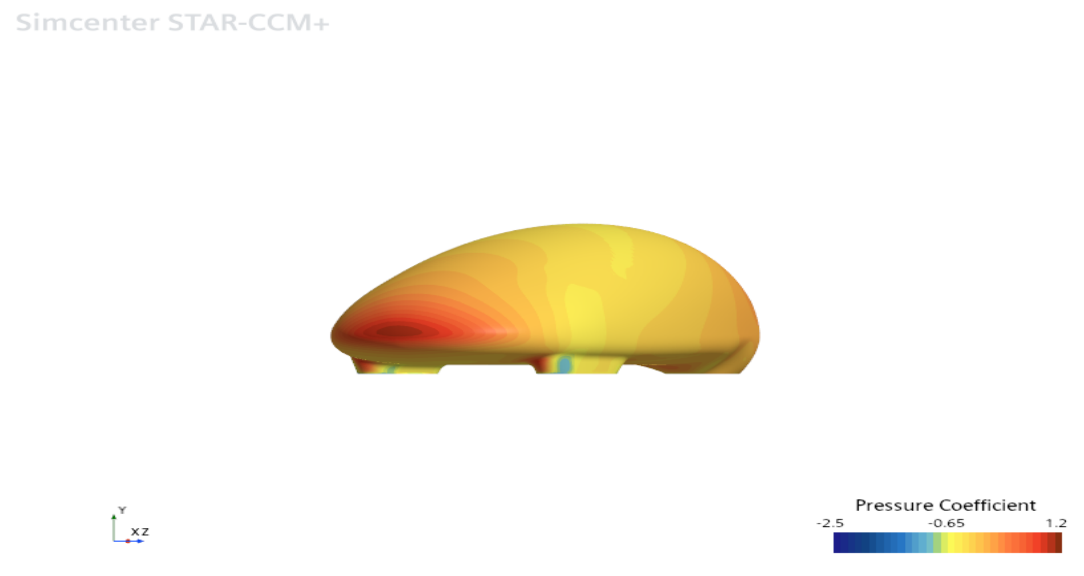
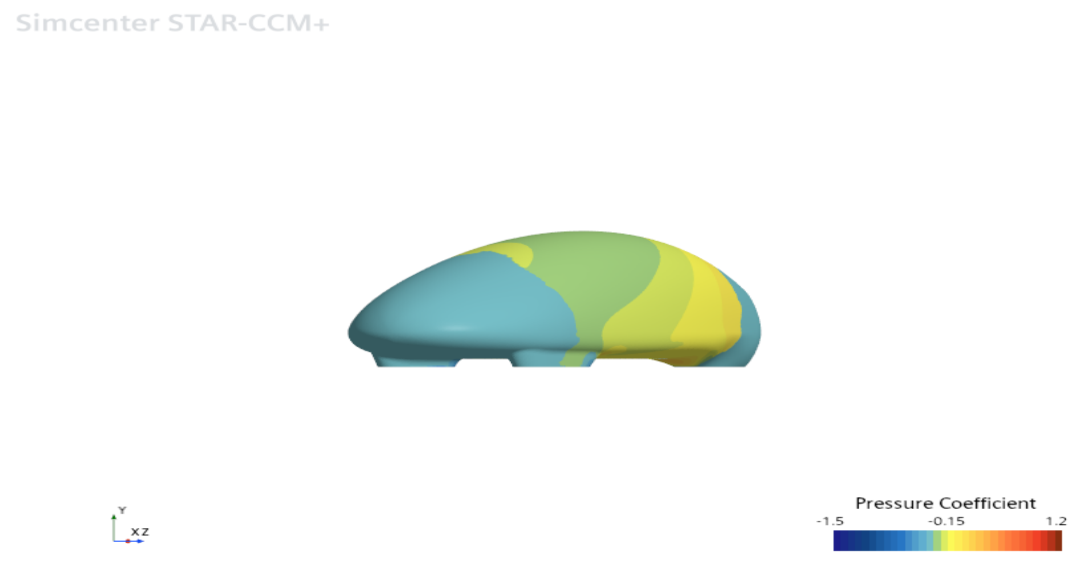
Design 1	
Design 2	
Design 3	

Table 8: Isometric view of C_P contours for all designs.

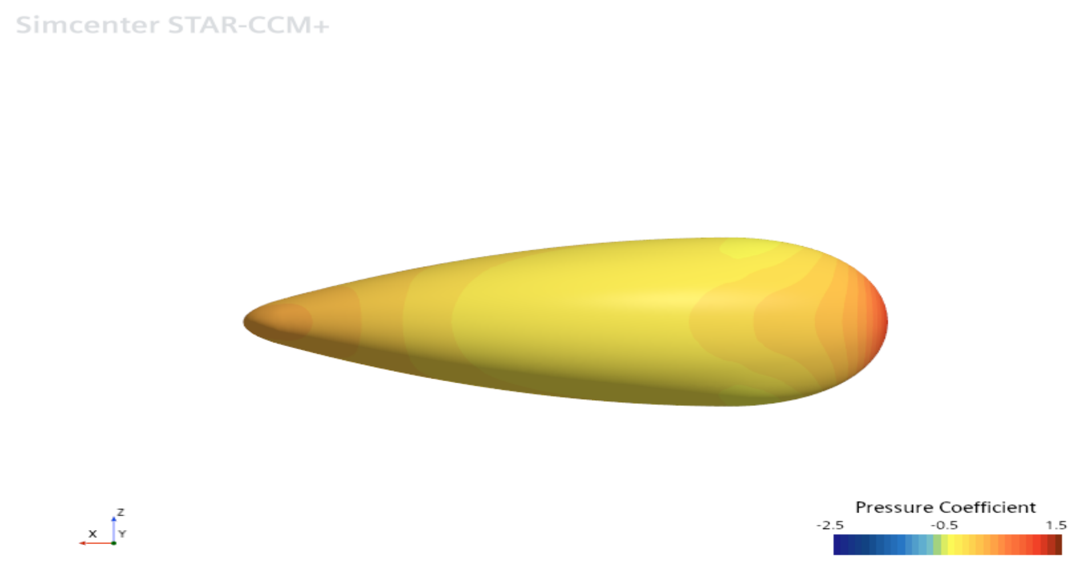
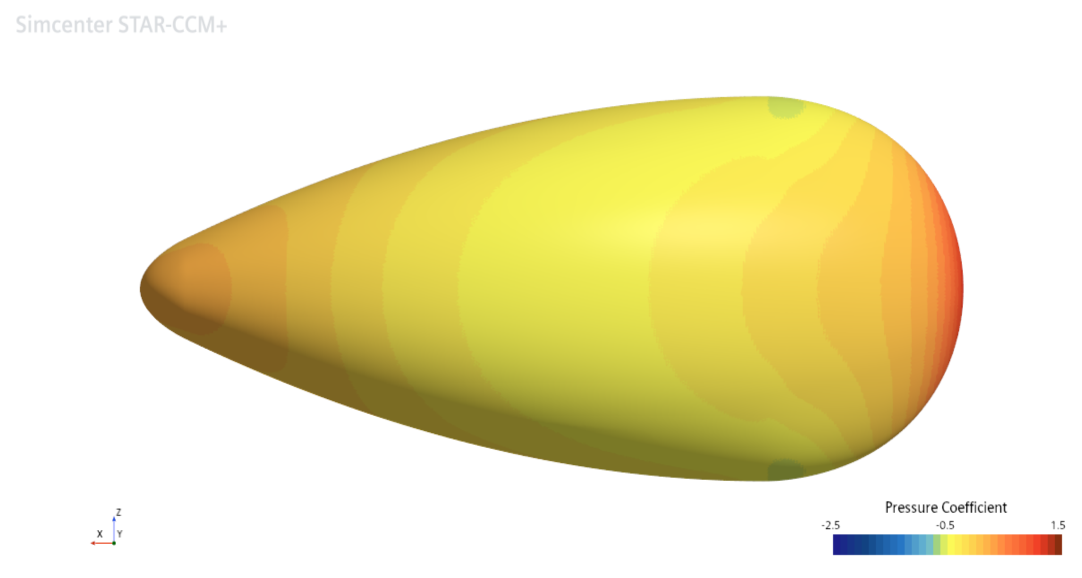
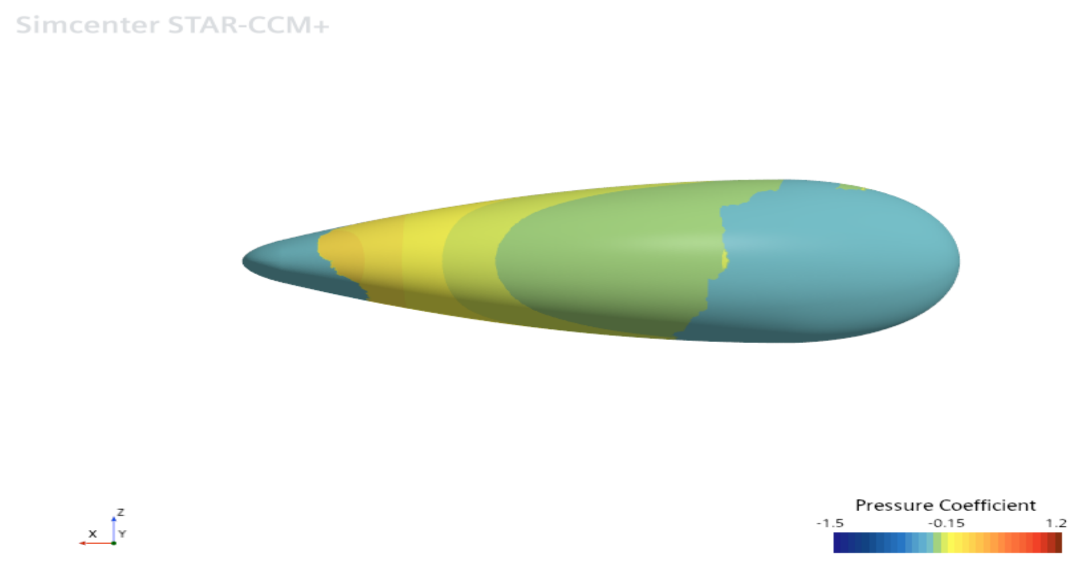
Design 1	
Design 2	
Design 3	

Table 9: Top view of C_P contours for all designs.

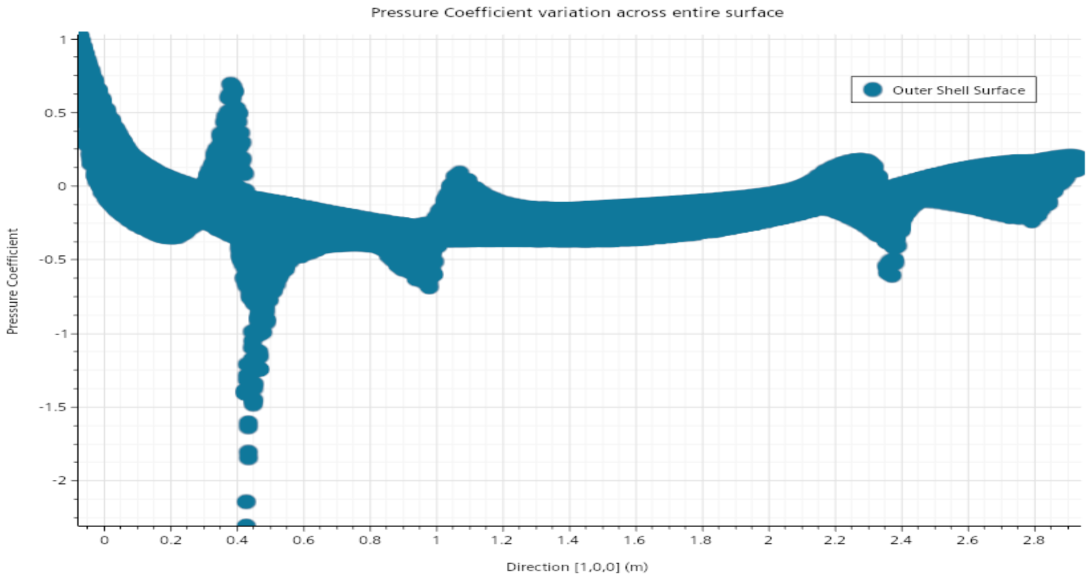
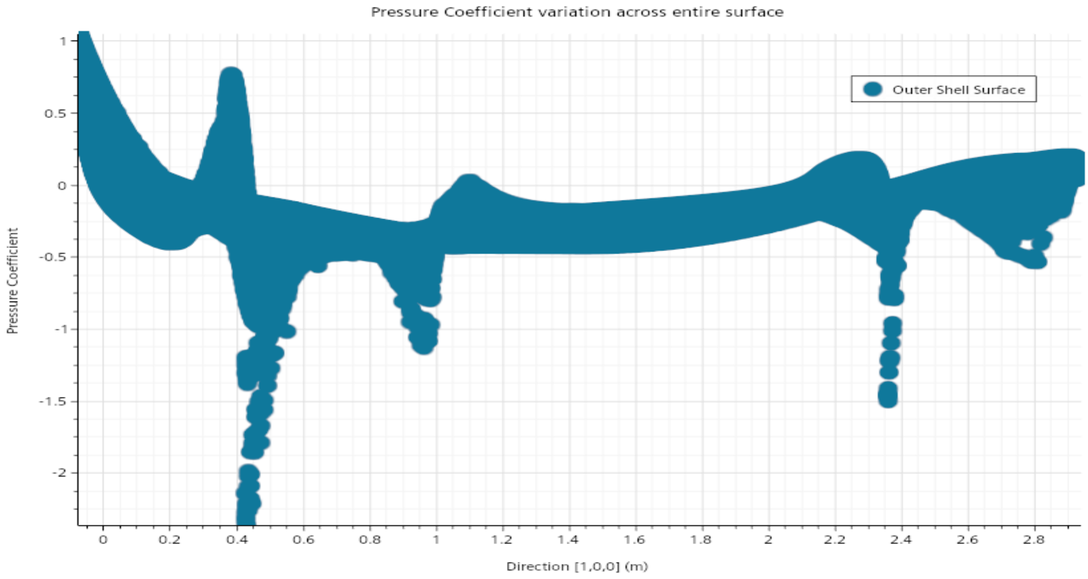
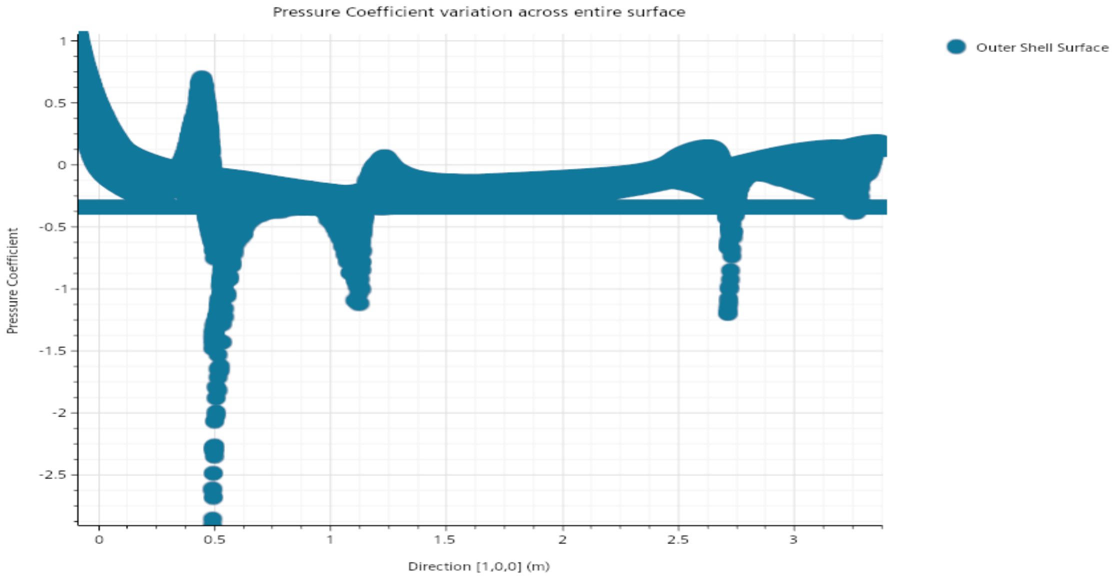
Design 1	
Design 2	
Design 3	

Table 10: C_P distribution across the entire length of the outer shell

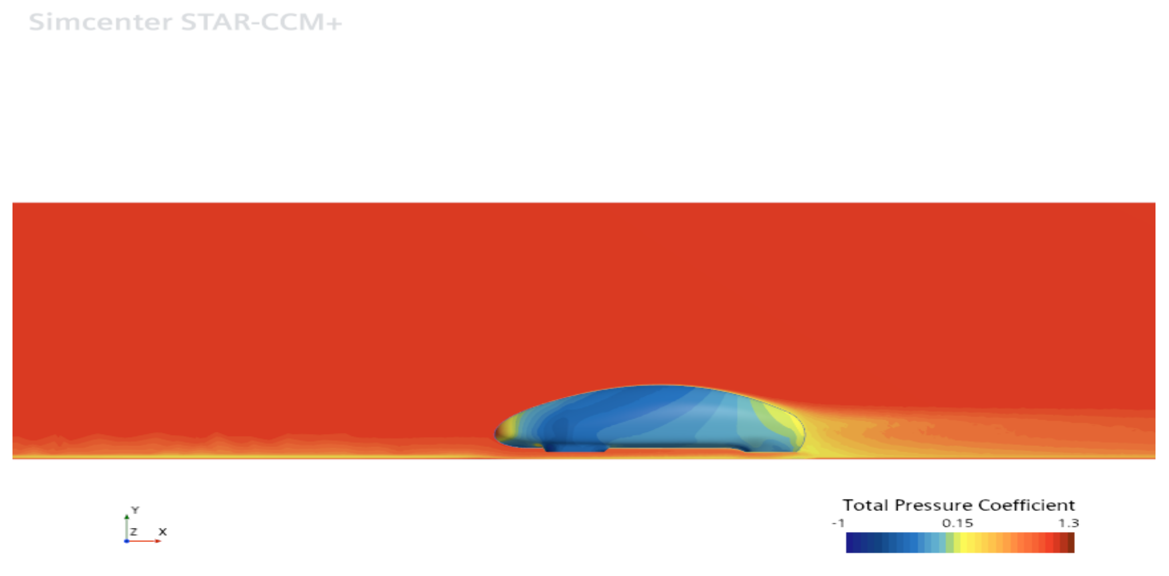
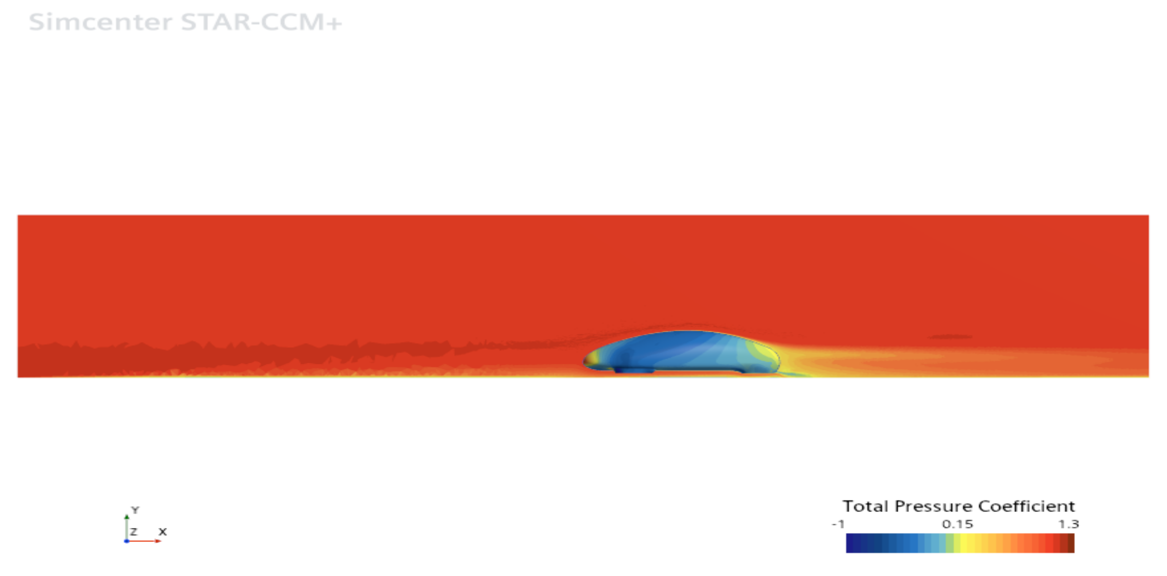
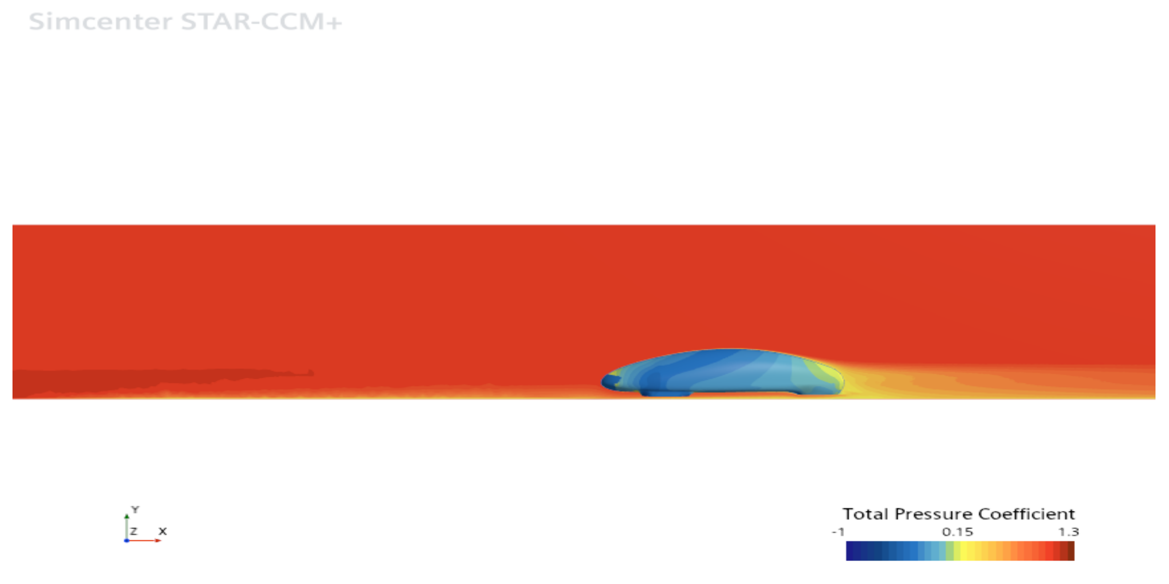
Design 1	
Design 2	
Design 3	

Table 11: Side view of $C_{P,T}$ contours with midsection plane for all designs.

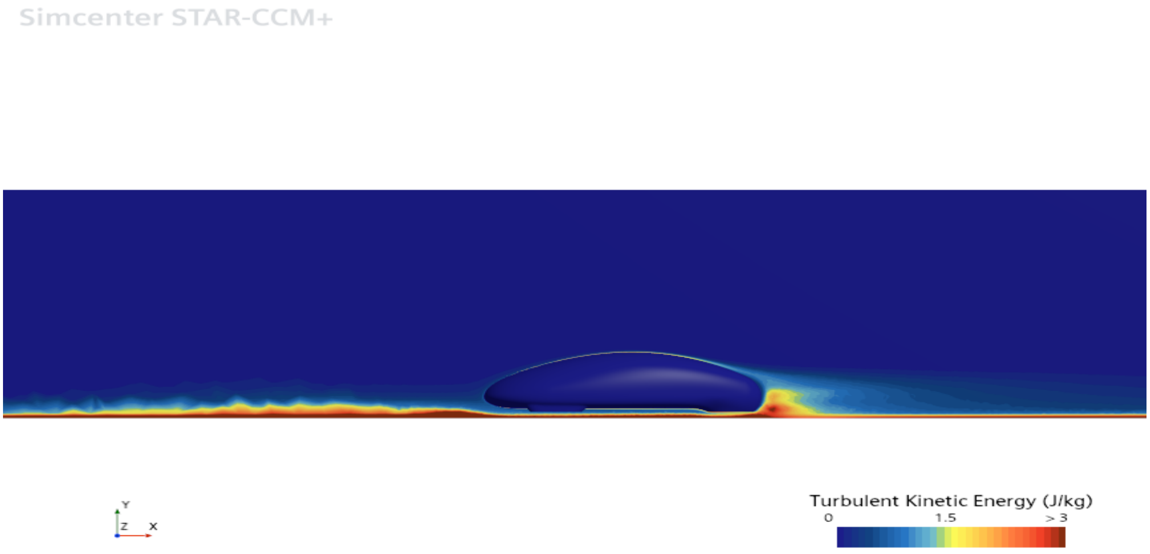
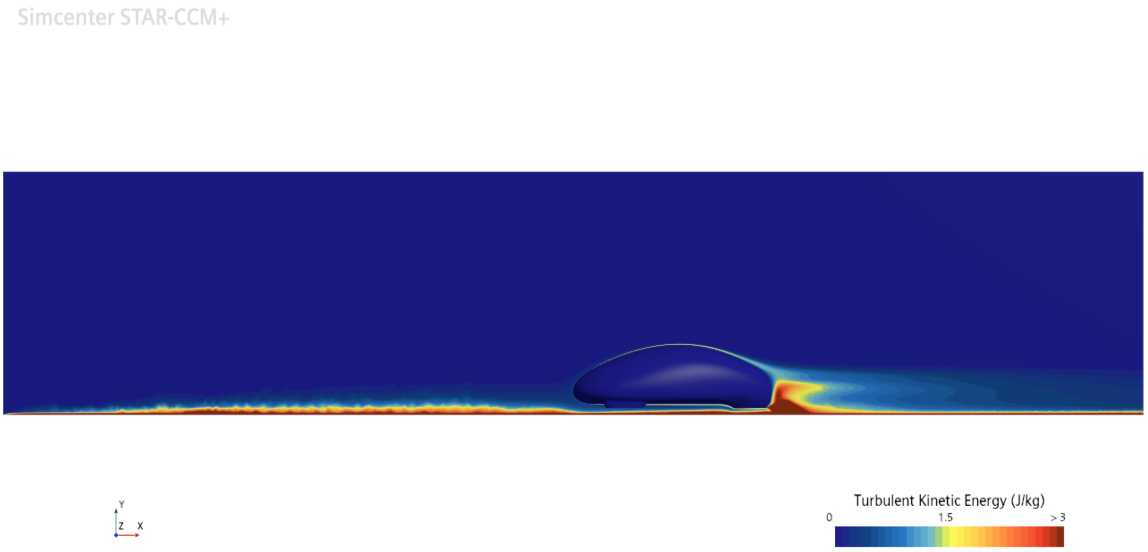
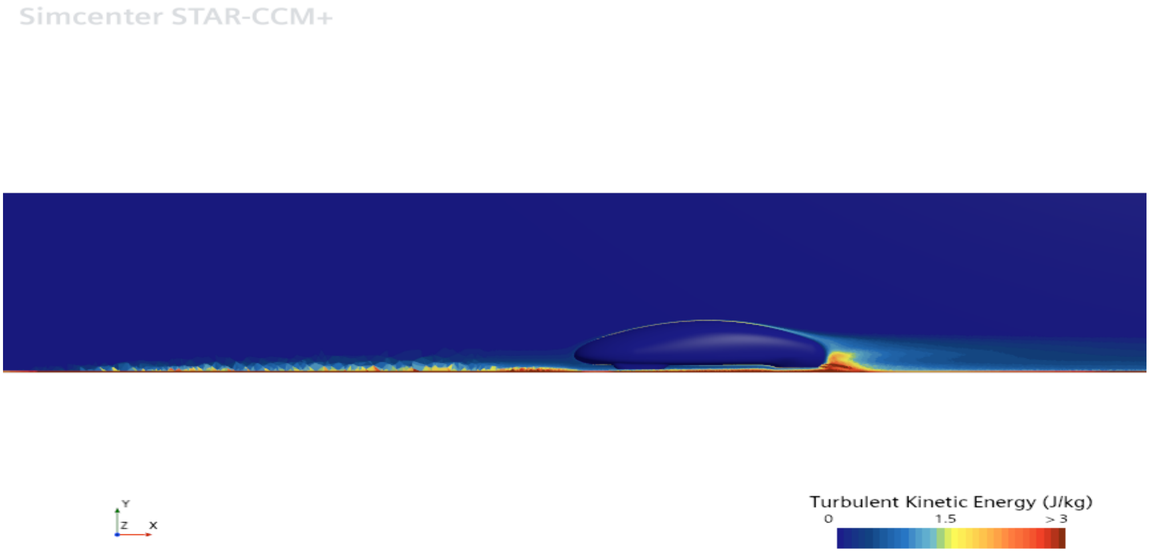
Design 1	
Design 2	
Design 3	

Table 12: Side view of Turbulent Kinetic Energy contours with midsection plane for all designs

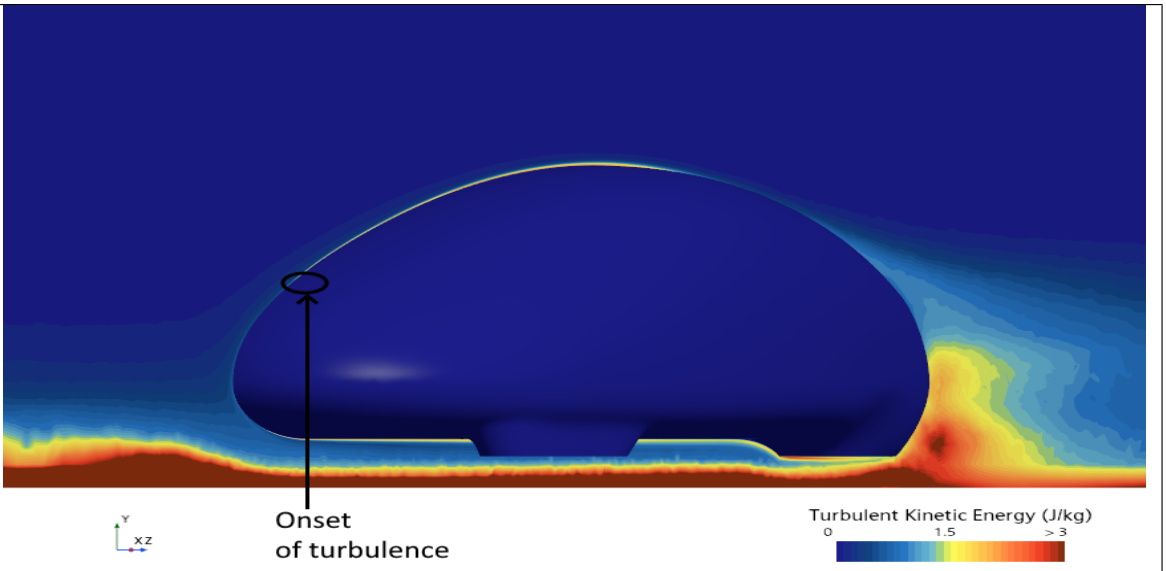
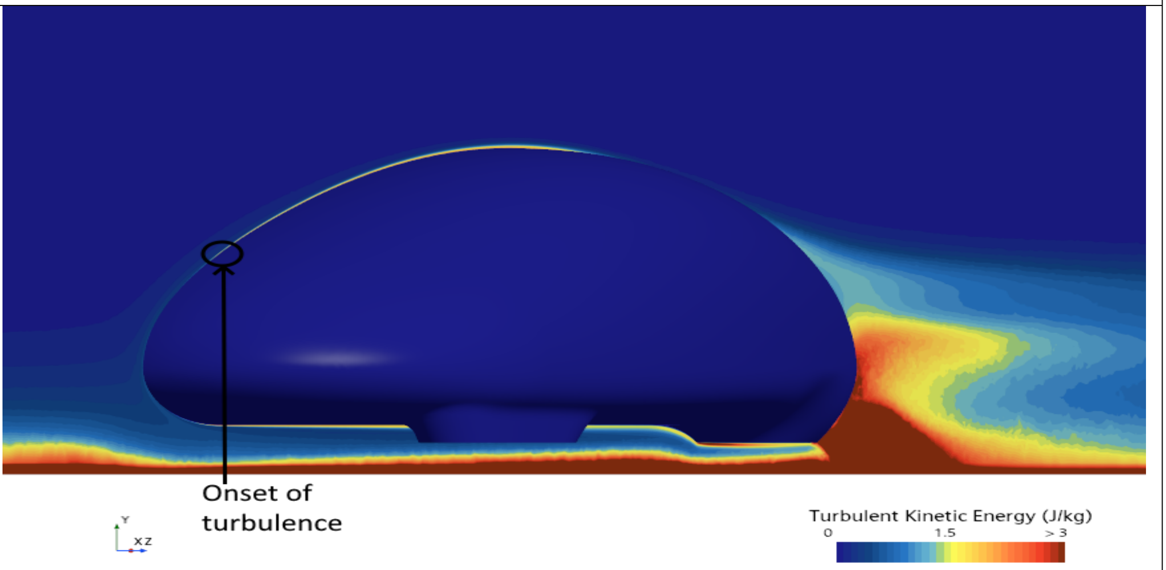
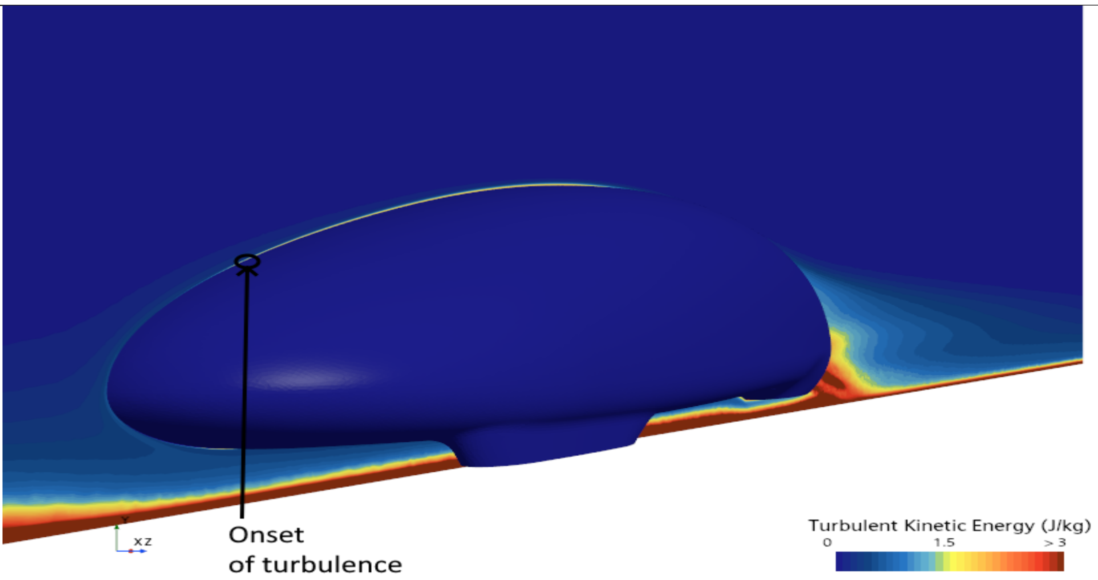
Design 1	
Design 2	
Design 3	

Table 13: Isometric closeup of Turbulent Kinetic Energy contours with midsection plane for all designs

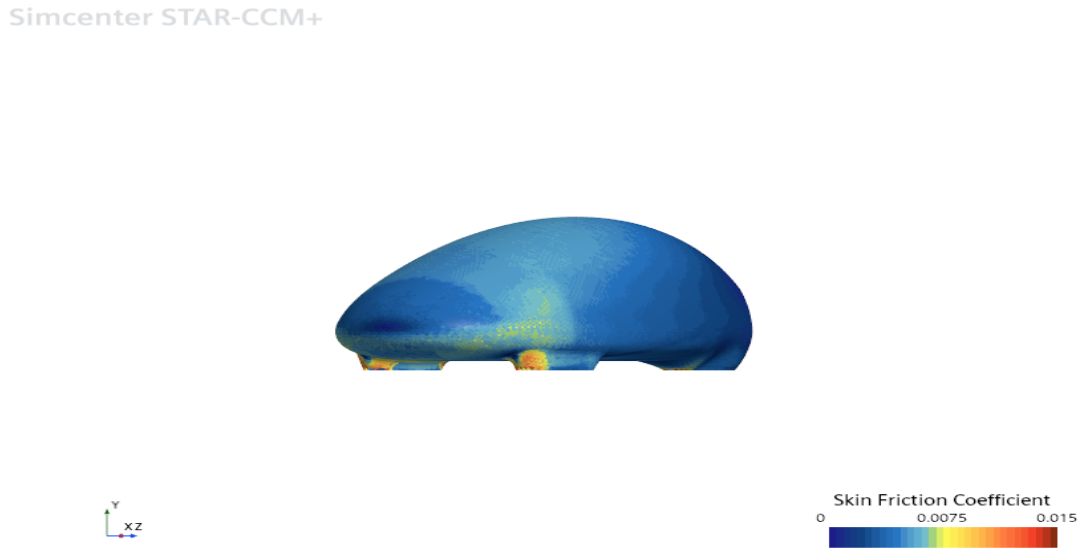
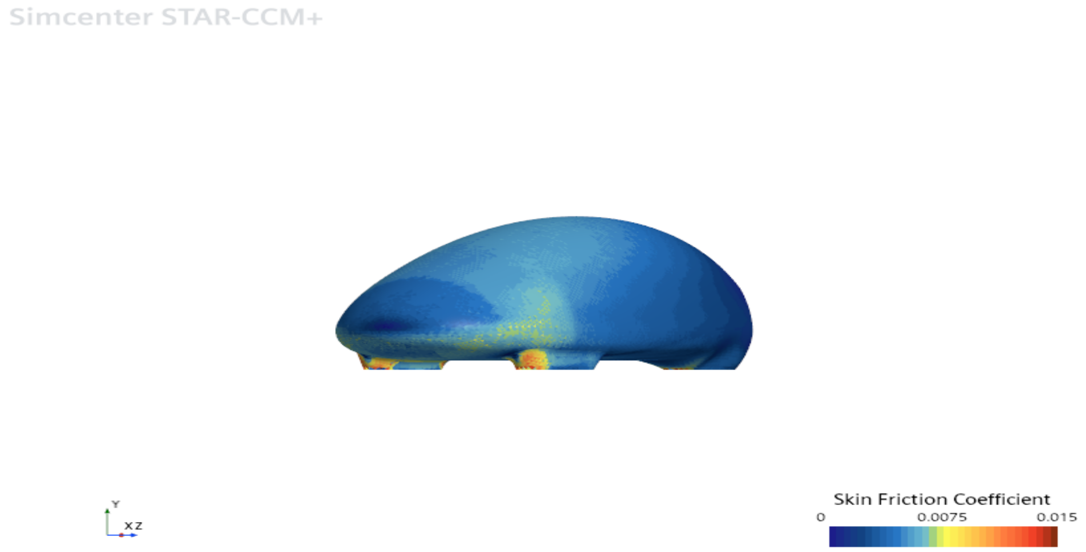
Design 1	
Design 2	
Design 3	

Table 14: Skin friction isometric view

Analysis of C_P results

While comparing the three designs, it is evident that Design 2 exhibits a significantly higher C_P value in the frontal region, as seen in Tables 7 and 8. This is attributed to the presence of its larger frontal area, which promotes the formation of a stagnation zone. Whilst Designs 1 and 3 also possess regions of stagnation, these zones are not as pronounced compared to the region observed in Design 2. Comparing Design 1 and 3, for the flow over the body, they show similar attributes for C_P as shown in Table 7. However, in the isometric view shown in Table 8, Design 3 shows much lower surface values for C_P on the outer shell body. The values are significantly lower than other designs, which may indicate the presence of an error. Therefore, the surface values for all scalar functions on Design 3, should be interpreted with a degree of uncertainty. Table 9 illustrates how the value of C_P varies along the upper surface of the outer shell. The initial rise in C_P at the stagnation region is followed by a subsequent decrease, this is because the fluid accelerates as it moves over the outer shell, resulting in a subsequent decrease in C_P. The relation with flow velocity and C_P is shown in the following equation:

$$C_P = \frac{P - P_\infty}{\frac{1}{2}\rho U_\infty^2} \tag{5.1}$$

Table 10 depicts an X-Y plot showing the variation in the value of C_P across the entire surface of the outer shell ; this appears to be mostly similar across the 3 designs, reflecting how similar the designs are. However, it can be seen that there is an error with the graph in Design 3 where there is a constant value of C_P The use of C_P as a parameter is crucial for understanding the drag force acting on the body, this is because the pressure variation over the surface determines the magnitude of pressure drag, as highlighted in the literature review.

Analysis of $C_{P,T}$ results

The total pressure coefficient,$C_{P,T}$, valuable indicator to observe the effects of turbulence, as it reflects the change in total pressure associated with variations in entropy, which depend on turbulence. These changes are irreversible, so in comparison to C_P where pressure may recover in the wake, $C_{P,T}$ remains low after separation. This occurs due to the reduction in total pressure, which is determined by the fluid's energy, which drops due to viscous effects and turbulence (Anderson 2017).

Table 11 illustrates the variation in $C_{P,T}$ across the surface of the outer shell and the surroundings, this was done using a mid-section depicting the value of $C_{P,T}$. Whilst comparing the three designs, it is apparent that Design 2 has the greatest reduction of $C_{P,T}$ in the wake region, highlighting the presence of a turbulent wake. Design 1 and 3 show losses in $C_{P,T}$ as well, and it can be seen that Design 3 shows a better performance with a lesser drop in the wake region, demonstrating better flow through the wake.

Analysis of Turbulent Kinetic Energy results

Turbulent kinetic energy indicates the intensity of turbulence in the flow (Versteeg and Malalasekera 2007). Turbulence is closely linked with drag, particularly skin friction drag. Increased turbulence enhances the mixing of fluid near the wall, leading to a higher velocity gradient at the surface (Anderson 2017).

As illustrated by Table 12, the moving wall boundary condition leads to the generation of turbulence near the floor. The turbulence continues onto the regions under the body of the car, potentially increasing skin friction drag, due to the chaotic motion at the surface.

Comparing the three designs, Design 2 shows the most turbulence in its wake, which is also a factor of it having the greatest frontal area, promoting greater flow separation, leading to a more turbulent wake. Whereas when looking at Design 1 and 3, Design 3 shows a minute improvement in terms of the turbulence generated in the wake region, this is because Design 3 features a longer outer shell, this means that the airflow can remain attached to the surface for slightly longer before separating. This allows for a more smooth pressure change and lessens flow separations.

Looking at the isometric close-up view illustrated by the figures in Table 13, the point of transition of the flow from laminar to turbulent has been highlighted. It is evident that for Design 1 and 2, the location of transition from laminar to turbulent occurs at a similar point, whereas in Design 3 it occurs slightly later on in the flow. This indicates that for a larger portion of the outer shell surface, the flow remains laminar. A laminar boundary layer results in lower skin friction compared to a turbulent boundary.As a result, the overall resistance to the flow is reduced, decreasing the drag acting on the outer shell.

Analysis of Skin Friction Coefficient

The skin friction coefficient is a good representation of the nature of the boundary layer flow. For values ranging from 0.003 to 0.001, the flow indicates a laminar region, whereas for values greater than 0.003, it indicates turbulence or a transition to turbulence. This is an important parameter to consider since the increase in turbulence near the boundary layer leads to an increase in skin friction. As illustrated in Table 14, Design 3 shows excellent and near 0 values for the skin friction coefficient in the frontal zones, and gradually increases as the flow progresses. However, as mentioned before, there seems to be an error persistent with the surface-specific values obtained from the simulation on Design 3, therefore, these results have to be treated with a sense of uncertainty. Designs 1 and 2 show an increased skin friction coefficient in the regions closer to the fairings, with values ranging from 0.008 to 0.01, suggesting the presence of a turbulent boundary layer there. This could be improved by better integration of the fairings with the main body itself.

Conclusion

The above analysis and usage of CFD on the 3 designs help in gaining an understanding on the aerodynamic performance of the 3 outer shells, and the scalar field visualisations help in gaining an understanding on the various design aspects that lead to an aerodynamically efficient design. In Design 3, the reduced frontal area minimises the surface area exposed to the flow, leading to lesser stagnation, thereby decreasing the pressure-induced drag. Also, due to the smoother transitions in the overall curvature of the body, flow separation is delayed, and the formation of turbulent wake regions is somewhat reduced.

Uncertainty analysis of CFD results

One of the main sources of uncertainty from the above CFD simulations on the 3 designs comes from the simplification made to the pre-processed geometry. By ignoring the wheels and sealing of the wheel fairings, many transient turbulent effects that originate from the interactions between the wheels and the fairings are not captured. This leads to an underestimation of the numerically calculated drag values. Moreover, the input inlet velocity conditions operate under the assumption that the airflow through the vehicle is directly against the direction of motion. This assumption is ,however, not true in a real-world scenario where the vehicle runs on a track with varying airflow due to the orientation of the vehicle constantly changing.

Since the designs have been developed as part of the study, there are no verified experimental results that can be used to validate the accuracy of the CFD simulations. Therefore, the best way to quantify numerical uncertainty is by conducting a grid independence test to ensure that the CFD solution obtained is one that is grid independent and accurate enough.

Mesh independence results

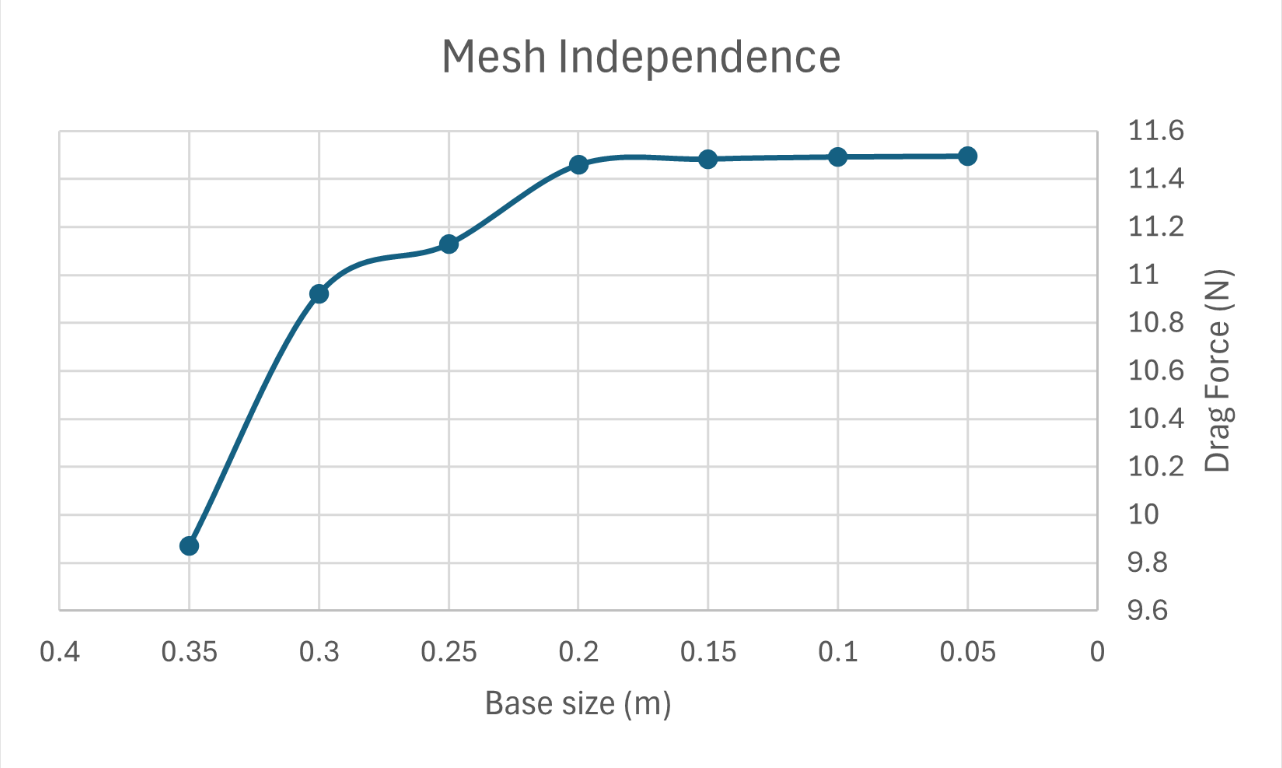


Fig. 41: Refinement in base size compared with drag force acting on Design 1

Base size (m)	0.35	0.3	0.25	0.2	0.15	0.1	0.05
Force (N)	9.871	10.921	11.128	11.458	11.482	11.491	11.494

The results from the mesh independence study show that for the mesh refinement conducted on the domain for the CFD study of Design 1, refining it past a base size of 0.1 m led to negligible improvements in the resolution of the solution. This suggests that a base size of 0.1 m leads to a grid-independent solution.

5.4 Results from integrated analysis with solar power generated

Introduction

The drag forces provided from the results above have been input into Python code to provide a comparative analysis of the resistive forces and power generated, as mentioned in the methodology in section 4.9. The power needed is obtained by multiplying the resistive forces with the velocity, and taking efficiency as a consideration, as shown in the equation below:

$$P_{\text{required}} = \frac{\left(F_{\text{drag}} + F_{\text{rr}}\right) v}{\eta_{\text{mech}}} \tag{5.2}$$

F_{drag} = aerodynamic drag force (N),
F_{rr} = rolling resistance force (N),
v = vehicle speed (m/s),
η_{mech} = drivetrain mechanical efficiency (0–1).

The below table provides detailed results using the mesh file of the solar panel surfaces and the drag forces evaluated to provide a comprehensive integrated analysis on the efficiency of the outer shell.

Design 1		
Panel Area	$4.32\ \text{m}^2$	
Drag Force	11.49 N	
Power Needed	0.45 kW	
Power Generation by Cell Type:		
Cell Type	**Power Generated (kW)**	**Surplus/Deficit (kW)**
Monocrystalline Si	0.28	−0.17
Polycrystalline Si	0.22	−0.24
GaAs	0.39	−0.06
Thin Film (CIGS/CdTe)	0.15	−0.30
Design 2		
Panel Area	$4.74\ \text{m}^2$	
Drag Force	14.49 N	
Power Needed	0.51 kW	
Power Generation by Cell Type:		
Cell Type	**Power Generated (kW)**	**Surplus/Deficit (kW)**
Monocrystalline Si	0.28	−0.22
Polycrystalline Si	0.22	−0.29
GaAs	0.39	−0.11
Thin Film (CIGS/CdTe)	0.15	−0.36
Design 3		
Panel Area	$4.93\ \text{m}^2$	
Drag Force	10.91 N	
Power Needed	0.44 kW	
Power Generation by Cell Type:		
Cell Type	**Power Generated (kW)**	**Surplus/Deficit (kW)**
Monocrystalline Si	0.33	−0.12
Polycrystalline Si	0.25	−0.19
GaAs	0.45	0.01
Thin Film (CIGS/CdTe)	0.17	−0.27

Table 15: Power Generation and Surplus/Deficit for Each Geometry and Cell Type

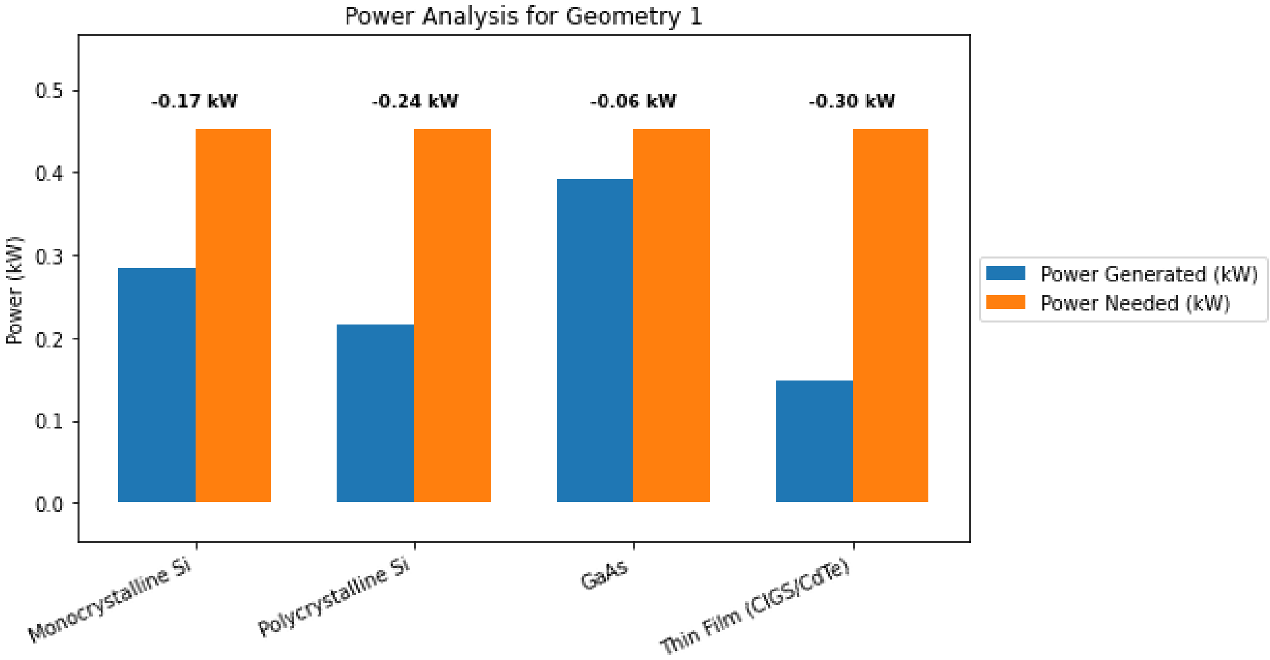


Fig. 42: Power generated vs Power needed comparison for Design 1

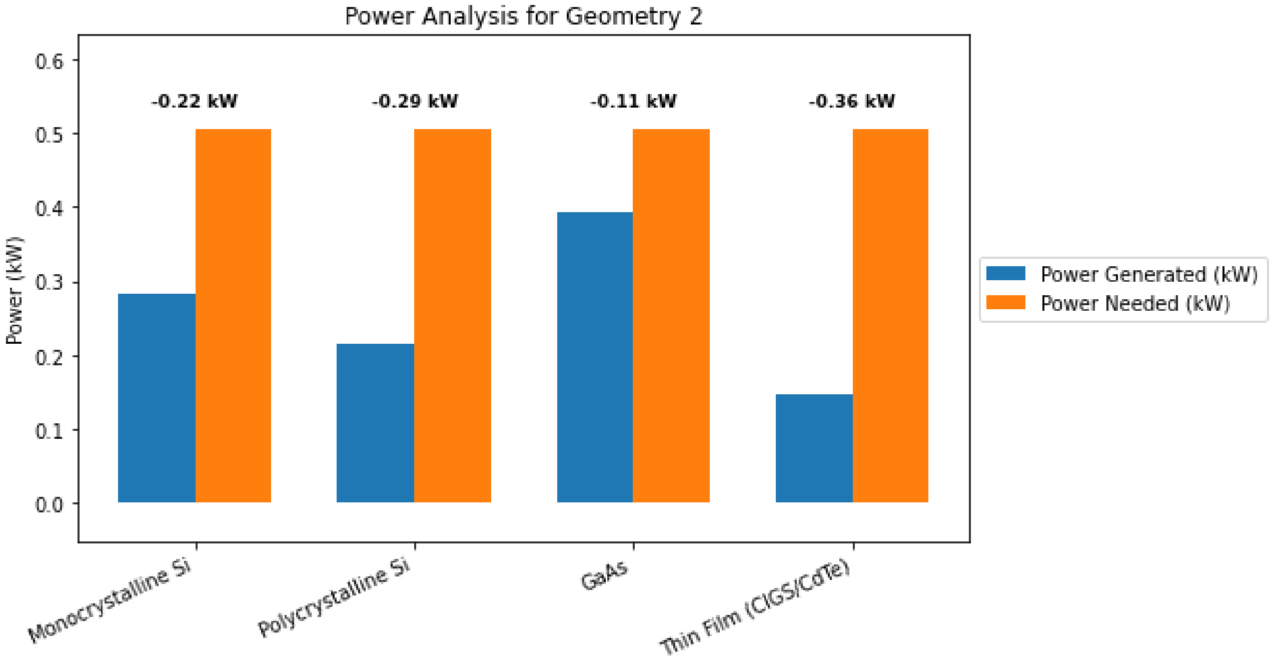


Fig. 43: Power generated vs Power needed comparison for Design 2

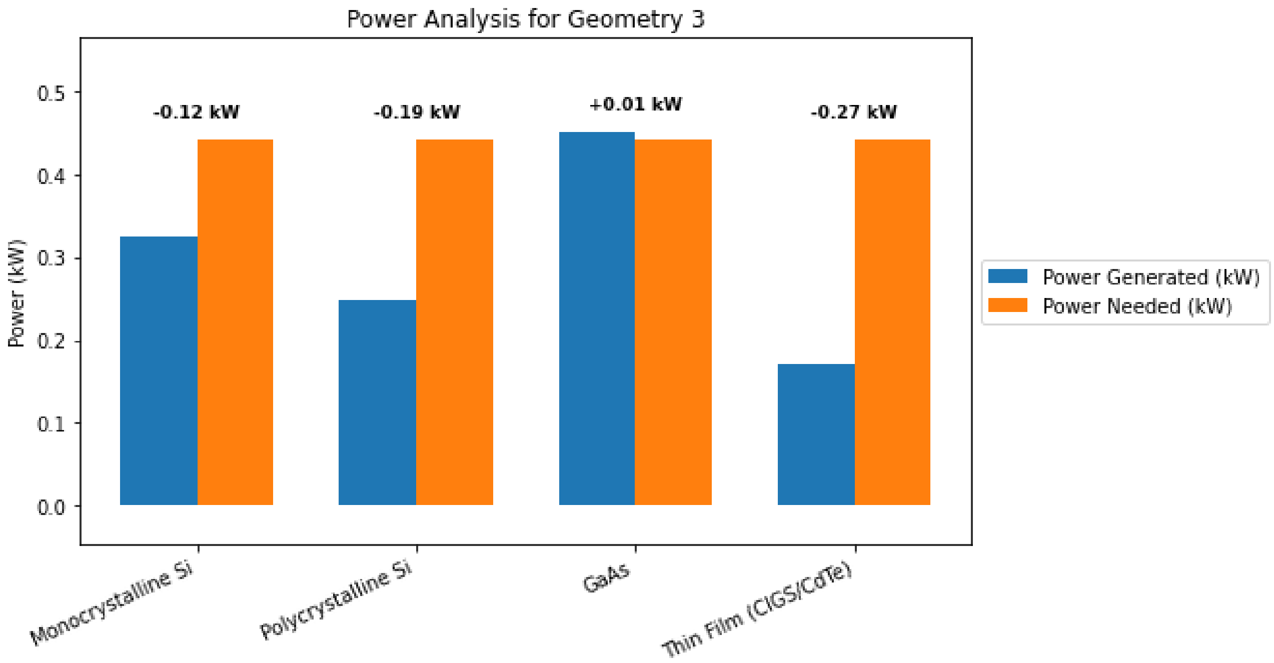


Fig. 44: Power generated vs Power needed comparison for Design 3

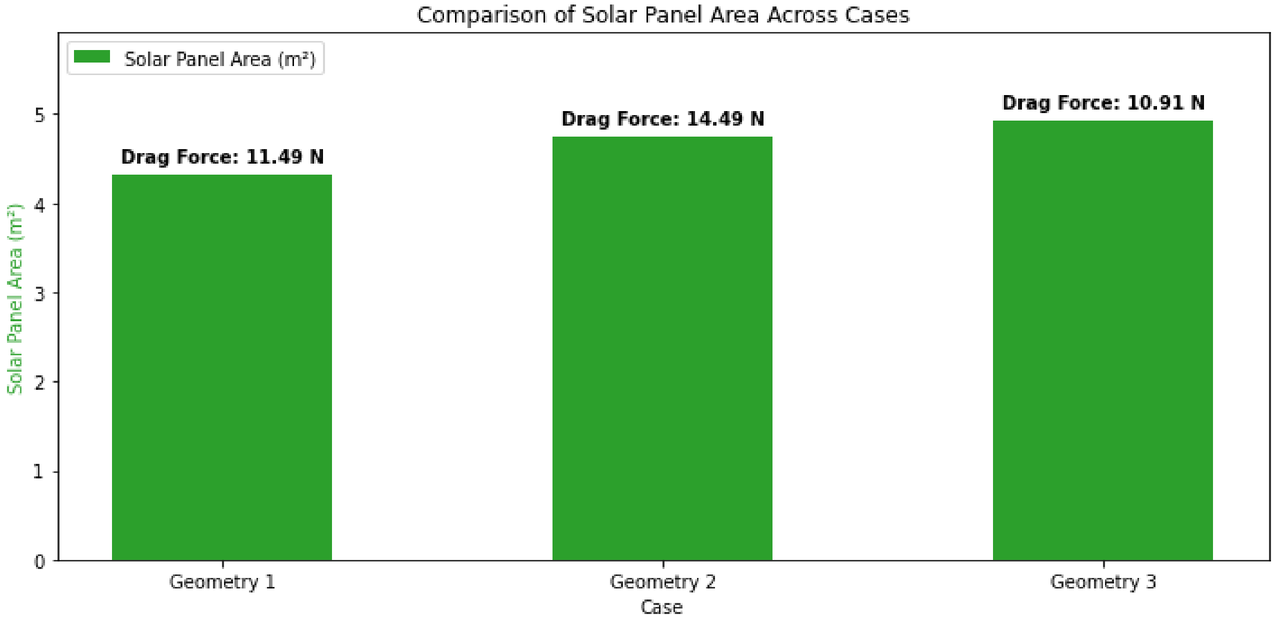
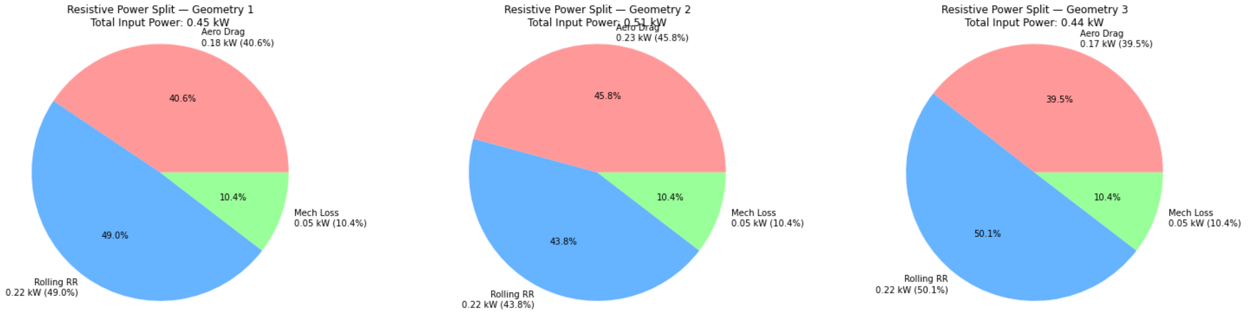


Fig. 45: Solar panel and drag force comparison across designs

Discussion

The results evaluate three different solar panel configurations, each corresponding to the respective outer shell design. Their performance has been evaluated using four different types of solar cells: Monocrystalline Si, Polycrystalline Si, GaAs, and Thin Film (CIGS/CdTe). The code uses the drag force for each design as mentioned in the above section, integrated with the other resistive forces to provide the following metrics: panel area, drag force, power needed, and the power generated by each cell type. The power surplus or deficit between the two is the main parameter to be assessed as the overall benchmark for the efficiency of the outer shell.

Table 15 illustrates that the total resistive power requirements for all three outer shells are nearly identical. This arises due to rolling resistance constituting the major share of the resistive power as shown in Fig. 46. Since the aerodynamic drag forces differ only marginally, their combined resistive powers varies very minutely.



((a)) Resistive power split for Design 1

((b)) Resistive power split for Design 2

((c)) Resistive power split for Design 3

Fig. 46: Comparison of resistive power split across the three designs

Looking at each design specifically, for Design 1 in , all cell types lead to a power deficit, with the resistive power being greater than the power generated. For the GaAs cell type, the power generated comes close with a deficit of only 0.06 kW. Design 2 requires the most power due to it having the most drag force acting on it and all cell types seem to have a significant power deficit. As expected GaAs is the best performing cell type with its high efficiency of 29.1 %, it still however, is not enough with a power deficit of 0.11 kW. Design 3 is the design requiring the least power, only 0.44 kW and has the lowest drag force acting on it. Despite this, all cell types barring GaAs, lead to a power deficit, with GaAs being the only cell type to produce a slight power surplus of 0.01 kW.

Looking at the Monocrystalline Si cell type, they prove to provide a considerable amount of energy, generating between 0.28 kW and 0.33 kW of energy across the three designs as shown in Fig. 42-44, which is almost more than half the power needed to oppose the resistive forces. For example, in Design 3, Mono Si cells generated 0.33 kW of power against the power requirement of 0.44 kW, resulting in a power deficit of -0.12 kW. Since this is an established and currently used cell technology in the field of solar cars, it can be integrated well into the Shell Eco-marathon cars as an additional power source. If paired with a sufficiently large battery to compensate for the power deficit, the system can ensure a reliable energy supply throughout the competition.

Despite the advantages in flexibility and ease of integration with curved surfaces for thin film solar cells, the results indicate that their lower efficiency limits their ability to meet the power requirements of the vehicle. Thin film cells generated only 0.15 to 0.17 kW across the designs, leading to large power deficits: -0.30 kW in Design 1, -0.36 kW in Design 2, and -0.27 kW in

Design 3. Across the three outer shell designs evaluated, thin film cells consistently resulted in the largest power deficit.

Despite Polycrystalline Si cells being included in the holistic analysis, traditionally, the solar panels are encased in rigid structures like glass (EcoFlow Blog 2023). There are limited applications on curved surfaces; however, in comparison to Monocrystalline Si cells, the efficiency is not high enough to be a viable choice. For instance, Polycrystalline Si cells generate 0.22–0.25 kW, resulting in power deficits of -0.24 kW (Design 1), -0.29 kW (Design 2), and -0.19 kW (Design 3), as depicted by the results in Fig. 42 to Fig. 44.

It can be seen that in terms of increasing solar panel area, increasing the length of the outer shell proves to be a better method. This is illustrated in Table 15, comparing Design 2 and 3; both designs increase their solar panel areas in different ways, Design 2 by increasing width and Design 3 by increasing length. The results indicate that increasing the area lengthwise is more effective since it gives greater solar panel area as well as better aerodynamic properties, as highlighted in section 5.4.

Conclusion

Based on the results discussed, it can be concluded that the holistic methodology as discussed in section 4.9, which integrates solar panel energy yield with aerodynamic performance analysis, provides a comprehensive framework for evaluating the performance of the outer shell. This novel approach not only applies to Shell Eco-marathon cars but could also be transferable to the wider context of other solar-powered cars in other competitions.

Through the use of this analysis, it can be concluded that of the three designs, **Design 3** proves to be the most efficient design. Not only does it show the least aerodynamic resistance, but it also has the greatest amount of solar power generation. Looking at the various cell types that can be used, GaAs cells show the best performance due to their high efficiency; however, as mentioned in the literature review, this is a newly developing technology, especially in terms of GaAs cells that are flexible. This is why they are not yet a practical option for implementation, but they represent a highly promising avenue for future applications.

Uncertainty analysis of the integrated methodology used

A number of assumptions were made in the analysis of the power generated by the solar panels, which were made to simplify the study. In reality, the efficiency of the solar panels and the solar irradiation incident on the panels are not constant values, rather, they depend on various operating conditions.

The efficiency of the solar cells is not constant; rather, it varies with temperature. For this study, the temperature dependence has not been included. This decision was made as most of the existing literature predominantly addresses the temperature effects of solar cells specifically within the context of commercial photovoltaic applications, for example, in solar fields or installations in homes. The temperature dependence is modelled through experimental data, which has not been conducted thoroughly for the application of solar cells in solar cars. Moreover, since the study looks at the use of various photovoltaic cells, the temperature dependence equations that have been mainly developed for monocrystalline or polycrystalline silicon cells cannot be directly applied (Hassan, Omer, and Mofdal 2025). This is because different photovoltaic technologies can exhibit different temperature coefficients and performance characteristics.

Another simplifying assumption used in the study was that solar irradiation incident on the solar panels on the surface of the car remains constant in both magnitude and direction throughout the analysis. This idealisation allows for easier modelling of the process in the Python code; however, it does not capture the variability of real-world conditions that are expected as the car goes around the circuit. In reality, the magnitude and the incident angle of the solar irradiation are constantly changing, resulting in a time-dependent power generation.

6 Conclusions and Future Work

6.1 Conclusions

The study has investigated the optimisation of a solar car's outer shell, balancing aerodynamic drag and solar panel area through a holistic methodology integrating CFD and photovoltaic energy generation. This work advances the topic area, demonstrating that for Shell Eco-marathon prototype vehicles, prioritising aerodynamic refinement, particularly by increasing the length of the outer shell, leads to aerodynamic drag reduction and an increase in panel area. The methodology developed couples CFD drag analysis and a mesh-based solar power estimation to accurately model the incident solar irradiation on curved surfaces. Providing a robust framework for future studies on solar vehicle design optimisation. This area of study can be improved by incorporating real-world factors such as the variation in the irradiation of the sun, the effect of the airflow over the solar surface, and the aerodynamic effects due to the interactions between the wheels, fairings and the vehicle body. Based on the methodology applied in this study, Design 3 was selected as the final outer shell design from the three developed options. The design will be used by the Manchester Solar Car Society in future Shell Eco-marathon competitions.

Beyond its direct application in the study conducted, this research enables related work in the wider field of energy-efficient solar-powered vehicle design. This simulation-driven workflow can be readily adapted by other teams and researchers working on similar projects, providing a template for integrating aerodynamics and energy generation considerations from early design stages. The CFD aspects of the methodology can be extended towards the optimisation of sub-systems of the vehicle, especially looking at the cooling of the internal electric components, integrating the effect of cooling with the overall holistic analysis of the efficiency of the system.

This work also has implications outside its immediate research area, offering valuable insights into the design of solar-powered drones or planes, and could be applied to any scenario where the trade-off between aerodynamics and solar energy generation is relevant. The mesh-based approach to quantifying solar power generation on complex, curved surfaces is transferable to any scenario where there is non-planar solar panel integration. This is particularly useful in light of the current push towards sustainability.

6.2 Future Work

In light of the findings made in this study, several improvements can be made. In terms of the design itself, future designs should favour increasing the length of the outer shell and overall chassis to increase panel area to minimise drag. The adoption of high-efficiency lightweight flexible photovoltaic cells, such as GaAs cells must be prioritised, despite their current cost and feasibility limitations.

The methodology should look to integrate real-world effects. For the case of power generation, the transient effects of solar irradiation can be modelled using existing data on global solar irradiation. This can be done by integrating data from a tool such as PVGIS, which provides historical data of the solar irradiance of a specific region (European Commission, Joint Research Centre 2025). This, along with the nature of the track the vehicle is competing on, can be integrated to provide an accurate estimate of the incident irradiation on the vehicle's solar panels.

Another aspect that can be improved upon is the dependence of the efficiency of the photovoltaic cells on external factors. This includes the temperature dependence of the efficiency of the solar cells. For silicon-based cells, this relationship is summarised as follows:

$$\text{Efficiency} \propto \frac{V_{oc} \times I_{sc}}{\text{Incident Power}}$$

Where V_{OC} and I_{SC} are the open-circuit voltage and short-circuit current. The use of the above analysis however, also has its shortcomings, since it does not consider the cooling effect due to the aerodynamics and flow over the panels. There has, however, been some groundbreaking research in this field, specifically, the paper by (Vinnichenko et al. 2014) investigates how the aerodynamic design of a solar car affects the cooling and efficiency of its photovoltaic modules. This was done using experimental and simulations to show the optimal shapes that enhance heat removal and power generation. Future work in the field should look at integrating this approach into the overall holistic methodology, providing an accurate analysis of the real-world operational conditions of a solar car.

References

Airfoil Tools (2025a). *NACA 63012A (n63012a-il) Airfoil.* `http://airfoiltools.com/airfoil/details?airfoil=n63012a-il`. Accessed: 2025-05-02. URL: `http://airfoiltools.com/airfoil/details?airfoil=n63012a-il`.

– (2025b). *NACA 6412 (naca6412-il) Airfoil.* `http://airfoiltools.com/airfoil/details?airfoil=naca6412-il`. Accessed: 2025-05-02. URL: `http://airfoiltools.com/airfoil/details?airfoil=naca6412-il`.

Anderson, John D. (2017). *Fundamentals of Aerodynamics.* 6th. New York: McGraw-Hill Education. ISBN: 9781259129919.

ANSYS Inc. (2025). *ANSYS Fluent Theory Guide: Transport Equations for the Transition SST Model.* Accessed: 2025-04-17. URL: `https://www.afs.enea.it/project/neptunius/docs/fluent/html/th/node74.htm`.

AutomoStory (2025). *First Solar Car Invented - History - AutomoStory.* Accessed: 2025-04-05. URL: `https://www.automostory.com/first-solar-car.htm`.

Cadence-CFD (2024). *What Is the Spalart-Allmaras Turbulence Model?* Accessed: 2025-04-27. Cadence Design Systems. URL: `https://resources.system-analysis.cadence.com/blog/msa2024-what-is-the-spalart-allmaras-turbulence-model`.

Carroll, Douglas R. (2003). *The Winning Solar Car: A Design Guide for Solar Race Car Teams.* Accessed via ProQuest Ebook Central. Warrendale, PA: SAE International. ISBN: 0-7680-1131-0. URL: `http://ebookcentral.proquest.com/lib/manchester/detail.action?docID=28983911`.

Chopra, K. L., P. D. Paulson, and V. Dutta (2004). "Thin-Film Solar Cells: An Overview". In: *Progress in Photovoltaics: Research and Applications* 12.2-3, pp. 69–92. DOI: `10.1002/pip.541`.

Cieśliński, Artur et al. (2016). "Investigation on Aerodynamics of Super-Effective Car for Drag Reduction". In: *Mechanics and Mechanical Engineering* 20.3, pp. 295–308.

Craft, T. J. (2024a). *Basic Finite Volume Methods.* `https://www.manchester.ac.uk`. Accessed: 2025-04-13.

– (2024b). *Further Flow Modelling – Turbulence.* `https://www.blackboard.manchester.ac.uk` (access restricted). Modelling & Simulation 3: CFD Lecture Notes, Dept. of Mechanical and Aerospace Engineering, University of Manchester.

– (2024c). *Segregated Flow Solver Strategy.* `https://www.manchester.ac.uk`. Accessed: 2025-04-13.

EcoFlow Blog (2023). *Flexible Solar Panels: The Ultimate Guide.* Accessed: 2025-05-01. URL: `https://blog.ecoflow.com/us/flexible-solar-panels-ultimate-guide/`.

Enter Museum (2024). *Tour de Sol: Pioneer Era of Solar Vehicles.* Accessed: 2025-04-10. Enter.ch. URL: `https://enter.ch/en/news/tour-de-sol-pioneer-era-of-solar-vehicles/`.

European Commission, Joint Research Centre (2025). *Photovoltaic Geographical Information System (PVGIS)*. `https://joint-research-centre.ec.europa.eu/photovoltaic-geographical-information-system-pvgis_en`. Accessed: 2025-05-01.

Fourie, A.Z. (May 2016). “Aerodynamic analysis and design of a solar powered vehicle using CFD”. Supervisor: Dr J.J. Bosman. Magister Dissertation. Potchefstroom, South Africa: North-West University, Potchefstroom Campus.

Hassan, Siham Muhammed, Manahil Mohammed Baher Eldin Omer, and Manahil E. E. Mofdal (2025). “Determining the Efficiency of Solar Cells at Different Temperatures”. In: *Journal of Power and Energy Engineering* 13.4, pp. 23–32. DOI: `10.4236/jpee.2025.134003`. URL: `https://www.scirp.org/pdf/jpee2025134_31771203.pdf`.

John D. Anderson, Jr. (2009). “Governing Equations of Fluid Dynamics”. In: *Computational Fluid Dynamics*. Ed. by J.F. Wendt. 3rd. Springer-Verlag Berlin Heidelberg. Chap. 2, pp. 15–51. ISBN: 978-3-540-85055-7. URL: `http://www.springer.com/978-3-540-85055-7`.

Khajepour, Amir, M. Saber Fallah, and Avesta Goodarzi (2014). *Modeling and Simulation of Electric and Hybrid Vehicles*. John Wiley & Sons. ISBN: 9781118403105.

Management, Solar Power (2019). “Alta and Stanford to Race GaAs-Powered Solar Car”. In: *Solar Power Management*. Accessed: 2025-04-14. URL: `https://solarpowermanagement.net/article/108990/Alta_and_Stanford_to_race_GaAs-powered_Solar_Car`.

Ozawa, Hiroyuki, Sumio Nishikawa, and Dai Higashida (1998). “Development of aerodynamics for a solar race car”. In: *JSAE Review* 19, pp. 343–349. DOI: `10.1016/S0389-4304(98)00019-8`.

Reynolds, Osborne (1883). “On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion”. In: *Philosophical Transactions of the Royal Society of London* 174, pp. 935–982. DOI: `10.1098/rstl.1883.0029`.

Richardson, Janet (Sept. 2024). *Thin Film Solar Panels in the UK*. Reviewed by Richard Burdett-Gardiner, Updated on Sep 29, 2024. URL: `https://www.renewableenergyhub.co.uk/main/solar-panels/thin-film-solar-panels`.

SAE International (2017). *Stepwise Coastdown Methodology for Measuring Tire Rolling Resistance*. Latest revision, July 2017. URL: `https://www.sae.org/standards/content/j2452_201707/`.

Santin, Jean-Jacques et al. (2007). *The World’s Most Fuel Efficient Vehicle: Design and Development of PAC-Car II*. ETH Zurich, Institut für Mess- und Regeltechnik. Zurich, Switzerland: vdf Hochschulverlag AG. ISBN: 978-3-7281-3134-8. URL: `http://www.vdf.ethz.ch`.

Shell Eco-Marathon (2025). *Shell Eco-Marathon 2025 Official Rules. Chapter I.* `https://www.shellecomarathon.com/about/global-rules/_jcr_content/root/main/section/simple_copy_copy_143/link_list/links/item0.stream/1725262844182/58f0026a158591aab5df47 shell-eco-marathon-2025-official-rules-chapter-i.pdf`. Accessed: 05 April 2025.

– (n.d.). *About Shell Eco-marathon: Energy Efficiency Competitions*. `https://www.shellecomarathon.com/about.html`. Accessed: 05 April 2025.

Siemens Digital Industries Software (2024a). *How to generate flow-aligned meshes.* Accessed: 2025-04-19. Siemens. URL: `https://community.sw.siemens.com/s/article/How-to-generate-a-flow-aligned-meshes`.

– (2024b). *Tetrahedral Mesher.* Simcenter STAR-CCM+ 2406 User Guide. Siemens. URL: `https://docs.plm.sw.siemens.com/en-US/starccmplus/2406/help/index.html?contextID=meshing_tetmesh` (visited on 04/20/2025).

SimScale (n.d.). *k-omega SST Model — Simulation Setup.* Accessed: 2025-04-13. URL: `https://www.simscale.com/docs/simulation-setup/global-settings/k-omega-sst/`.

Society, American Chemical (2013). *How a Solar Cell Works.* Accessed: 2025-04-14. URL: `https://www.acs.org/education/chemmatters/past-issues/archive-2013-2014/how-a-solar-cell-works.html`.

Tamai, Goro (1999). *The Leading Edge: Aerodynamic Design of Ultra-Streamlined Land Vehicles.* Accessed: 2025-04-10. Warrendale, PA: SAE International. ISBN: 9780837608605. URL: `https://www.amazon.co.uk/Leading-Edge-Aerodynamic-Ultra-streamlined-Engineering/dp/0837608600`.

Technologies, Solar Capture (2022). *What Are Monocrystalline Solar Panels?* Accessed: 2025-04-14. URL: `https://www.solarcapturetechnologies.com/what-are-monocrystalline-solar-panels/`.

University, Embry-Riddle Aeronautical (2021). *Aerodynamics of Airfoil Sections.* Accessed: 2025-04-14. URL: `https://eaglepubs.erau.edu/introductiontoaerospaceflightvehicles/chapter/airfoil-characteristics/`.

Versteeg, H.K. and W. Malalasekera (2007). *An Introduction to Computational Fluid Dynamics: The Finite Volume Method.* 2nd. Harlow, England: Pearson Education. ISBN: 9780131274983.

Vinnichenko, Nikolay A. et al. (2014). "Solar car aerodynamic design for optimal cooling and high efficiency". In: *Solar Energy* 103, pp. 183–190. DOI: `10.1016/j.solener.2014.02.019`. URL: `https://www.sciencedirect.com/science/article/pii/S0038092X14000613`.

White, Frank M. (2016). *Fluid Mechanics.* 8th. New York: McGraw-Hill Education. ISBN: 9780073398273.

Yamaguchi, Masafumi (2020). "High-Efficiency GaAs-Based Solar Cells". In: *Post-Transition Metals.* Ed. by Mohammed Muzibur Rahman et al. IntechOpen. DOI: `10.5772/intechopen.94365`. URL: `https://doi.org/10.5772/intechopen.94365`.

Appendices

Project Management

This appendix provides a detailed account of how the project plan evolved throughout the project. It discusses the reasons behind the necessary changes to the plans, taking into account the various circumstances at each stage, offering reflections on the progress of the project. The section begins by outlining the plan, which was created in week 2 of the first semester. Followed by a revised plan created in week 12 of the first semester. Finally, the last revision of the project plan developed in week 3 of the second semester is presented.

Initial Project Plan

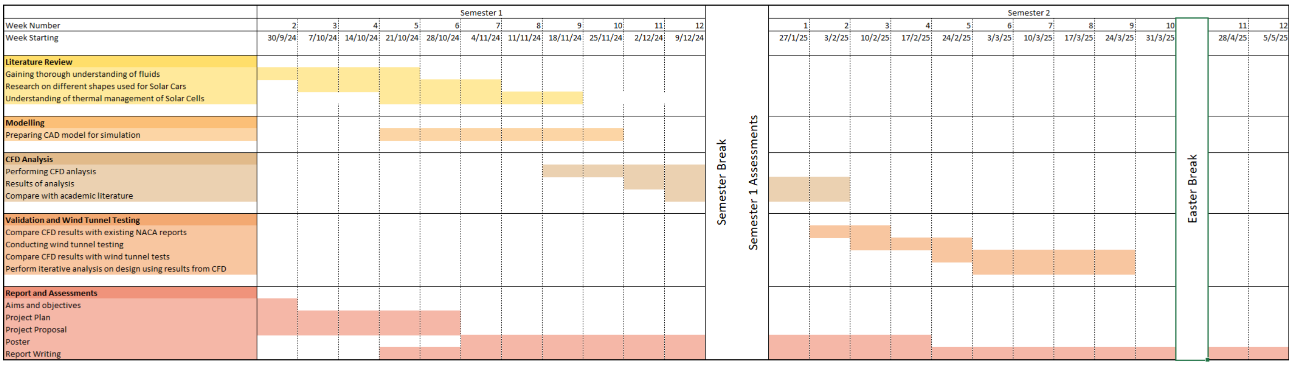
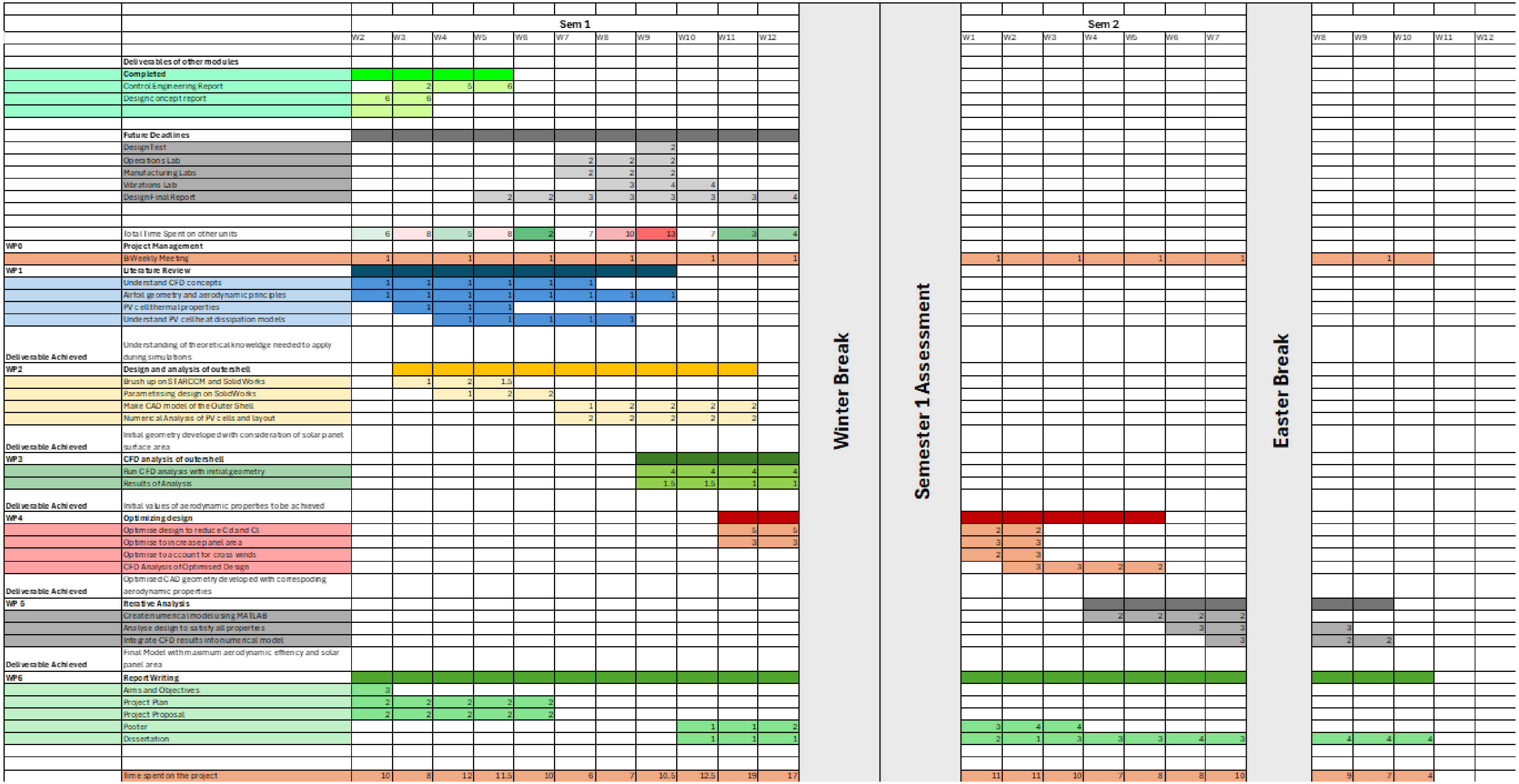


Fig. 47: Initially developed Gantt Chart

The initial project plan was developed in week 2, with very little knowledge of the project and its scope. The initial plan involved integrating the temperature dependence and cooling effects of the airflow. Initially, the aim was to develop a singular design, conduct CFD simulations on it and validate it using experimental wind tunnel tests. It, however, became apparent that the scope of the project was too large. Integrating both aerodynamic and heat transfer analysis into the CFD simulations proved to be quite a challenging task given the time frame. On top of that, adding an experimental aspect to it with wind tunnel testing made the initial plans unfeasible.

Revised Project Plan

Considering these challenges, it was decided to solely focus on the aerodynamic aspect, ignoring the thermal effects of the solar panels. Instead, a holistic methodology was chosen, integrating solar power generation as a consideration in the design methodology. The revised project Gantt chart features the work packages with the aim of meeting the above goals



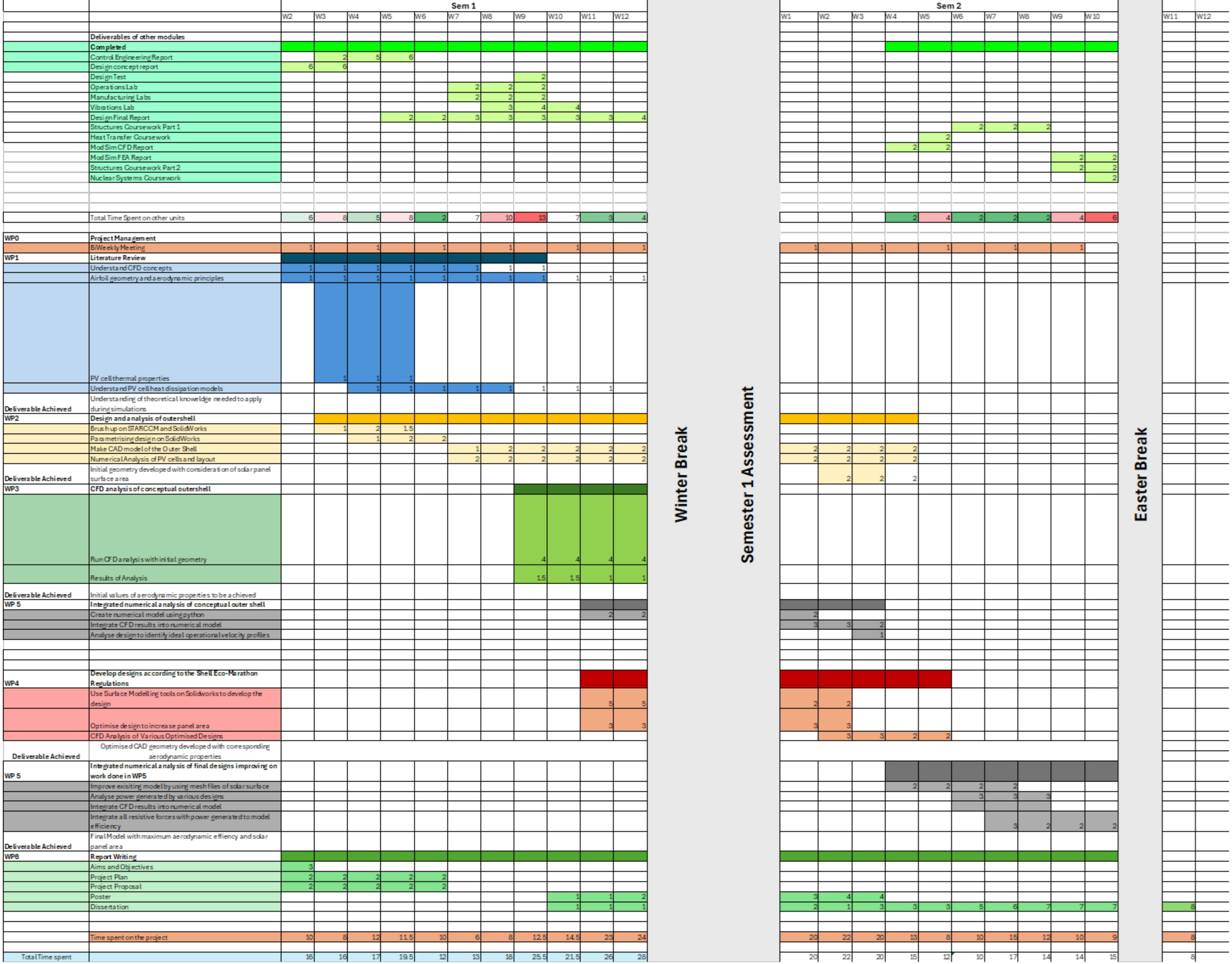
		Sem 1											Winter Break	Semester 1 Assessment	Sem 2							Easter Break					
		W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12			W1	W2	W3	W4	W5	W6	W7		W8	W9	W10	W11	W12
	Deliverables of other modules																										
	Completed	■	■	■	■																						
	Control Engineering Report		2	5	6																						
	Design concept report	6	6																								
	Future Deadlines	■	■	■	■	■	■	■	■	■	■	■															
	Design Test								2																		
	Operations Lab						2	2	2																		
	Manufacturing Labs						2	2	2																		
	Vibrations Lab							3	4	4																	
	Design Final Report				2	2	3	3	3	3	3	4															
	Total Time Spent on other units	6	8	5	8	2	7	10	13	7	3	4															
WP0	**Project Management**																										
	BiWeekly Meeting	1		1		1		1		1		1			1		1		1		1			1			
WP1	**Literature Review**	■	■	■	■	■	■	■	■																		
	Understand CFD concepts	1	1	1	1	1	1																				
	Airfoil geometry and aerodynamic principles	1	1	1	1	1	1	1	1																		
	PV cell thermal properties		1	1	1																						
	Understand PV cell heat dissipation models			1	1	1	1	1																			
Deliverable Achieved	Understanding of theoretical knoweldge needed to apply during simulations																										
WP2	**Design and analysis of outer shell**		■	■	■	■	■	■	■	■	■																
	Brush up on STARCCM and SolidWorks		1	2	1.5																						
	Parametrising design on SolidWorks			1	2	2																					
	Make CAD model of the Outer Shell						1	2	2	2	2																
	Numerical Analysis of PV cells and layout						2	2	2	2	2																
Deliverable Achieved	Initial geometry developed with consideration of solar panel surface area																										
WP3	**CFD analysis of outer shell**								■	■	■	■															
	Run CFD analysis with initial geometry								4	4	4	4															
	Results of Analysis								1.5	1.5	1	1															
Deliverable Achieved	Initial values of aerodynamic properties to be achieved																										
WP4	**Optimizing design**										■	■			■	■	■	■	■								
	Optimise design to reduce Cd and Cl										5	5			2	2											
	Optimise to increase panel area										3	3			3	3											
	Optimise to account for cross winds														2	3											
	CFD Analysis of Optimised Design															3	3	2	2								
Deliverable Achieved	Optimised CAD geometry developed with correspoding aerodynamic properties																										
WP 5	**Iterative Analysis**																	■	■	■	■		■	■			
	Create numerical model using MATLAB																	2	2	2	2						
	Analyse design to satisfy all properties																			3	3		3				
	Integrate CFD results into numerical model																				3		2	2			
Deliverable Achieved	Final Model with maximum aerodynamic effiency and solar panel area																										
WP6	**Report Writing**	■	■	■	■	■	■	■	■	■	■	■			■	■	■	■	■	■	■		■	■	■		
	Aims and Objectives	3																									
	Project Plan	2	2	2	2	2																					
	Project Proposal	2	2	2	2	2																					
	Poster									1	1	2			3	4	4										
	Dissertation									1	1	1			2	1	3	3	3	4	3		4	4	4		
	Time spent on the project	10	8	12	11.5	10	6	7	10.5	12.5	19	17			11	11	10	7	8	8	10		9	7	4		

Fig. 48: Revised Gantt Chart

Final Project Plan

The revised project plan mentions conducting an iterative process, looking to constantly refine the geometry of the developed outer shell. Since this was a very time-intensive process, having to constantly update the CAD to be input into the CFD solver, it was decided that only 3 final designs were to be developed and then compared. Analysing the behaviour of the designs under cross-winds and complex wind loading scenarios was also decided to be ignored, to save time.

The final project plan also includes the time that was spent on all the modules throughout the year. It can be seen that a lot more time than expected was spent on writing the final dissertation. This was mainly because of the workload from various other coursework deadlines from other modules.



		Sem 1												Sem 2												
		W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	Winter Break Semester 1 Assessment	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	Easter Break	W11	W12
	Deliverables of other modules																									
	Completed																									
	Control Engineering Report		2	5	6																					
	Design concept report	6	6																							
	Design Test								2																	
	Operations Lab						2	2	2																	
	Manufacturing Labs						2	2	2																	
	Vibrations Lab							3	4	4																
	Design Final Report				2	2	3	3	3	3	3	4														
	Structures Coursework Part 1																		2	2	2					
	Heat Transfer Coursework																	2								
	Mod Sim CFD Report																2	2								
	Mod Sim FEA Report																					2	2			
	Structures Coursework Part 2																					2	2			
	Nuclear Systems Coursework																						2			
	Total Time Spent on other units	6	8	5	8	2	7	10	13	7	3	4					2	4	2	2	2	4	6			
WP0	**Project Management**																									
	BiWeekly Meeting	1		1		1		1		1		1		1		1		1		1		1				
WP1	**Literature Review**																									
	Understand CFD concepts	1	1	1	1	1	1	1	1																	
	Airfoil geometry and aerodynamic principles	1	1	1	1	1	1	1	1	1	1	1														
	PV cell thermal properties		1	1	1																					
	Understand PV cell heat dissipation models			1	1	1	1	1	1	1	1															
Deliverable Achieved	Understanding of theoretical knoweldge needed to apply during simulations																									
WP2	**Design and analysis of outershell**																									
	Brush up on STARCCM and SolidWorks		1	2	1.5																					
	Parametrising design on SolidWorks			1	2	2																				
	Make CAD model of the Outer Shell						1	2	2	2	2	2		2	2	2	2									
	Numerical Analysis of PV cells and layout						2	2	2	2	2	2		2	2	2	2									
Deliverable Achieved	Initial geometry developed with consideration of solar panel surface area														2	2	2									
WP3	**CFD analysis of conceptual outershell**																									
	Run CFD analysis with initial geometry								4	4	4	4														
	Results of Analysis								1.5	1.5	1	1														
Deliverable Achieved	Initial values of aerodynamic properties to be achieved																									
WP 5	**Integrated numerical analysis of conceptual outer shell**																									
	Create numerical model using python										2	2		2												
	Integrate CFD results into numerical model													3	3	2										
	Analyse design to identify ideal operational velocity profiles															1										
WP4	**Develop designs according to the Shell Eco-Marathon Regulations**																									
	Use Surface Modelling tools on Solidworks to develop the design										5	5		2	2											
	Optimise design to increase panel area										3	3		3	3											
	CFD Analysis of Various Optimised Designs														3	3	2	2								
Deliverable Achieved	Optimised CAD geometry developed with corresponding aerodynamic properties																									
WP 5	**Integrated numerical analysis of final designs improving on work done in WP5**																									
	Improve exisiting model by using mesh files of solar surface																2	2	2	2						
	Analyse power generated by various designs																		3	3	3					
	Integrate CFD results into numerical model																									
	Integrate all resistive forces with power generated to model efficiency																			3	2	2	2			
Deliverable Achieved	Final Model with maximum aerodynamic effiency and solar panel area																									
WP6	**Report Writing**																									
	Aims and Objectives	3																								
	Project Plan	2	2	2	2	2																				
	Project Proposal	2	2	2	2	2																				
	Poster									1	1	2		3	4	4										
	Dissertation									1	1	1		2	1	3	3	3	5	6	7	7	7		8	
	Time spent on the project	10	8	12	11.5	10	6	8	12.5	14.5	23	24		20	22	20	13	8	10	15	12	10	9		8	
Total Time spent		16	16	17	19.5	12	13	18	25.5	21.5	26	28		20	22	20	15	12	10	17	14	14	15		8	

Fig. 49: Final Gantt Chart

Reflection

Tackling ambiguous engineering problems in a student-led initiative.

The project provided valuable experience in addressing an open-ended engineering challenge within a collaborative student society. The process of splitting complex problems into achievable tasks and applying engineering principles to make informed assumptions, even in the absence of complete information, has established a holistic and robust framework for future work. These strategies are expected to enhance the effectiveness and efficiency of problem-solving in subsequent engineering projects.

Adapting to a constantly changing project scope

The need to respond to the frequent changes in the project scope highlighted the importance of flexibility and adaptability within engineering problem-solving. Operating a student society has underscored the need for dynamic project management. The ability to reassess and review plans proved essential for project continuity.

Wider implication of the research

Due to the methodology developed and applied, integrating CFD with mesh-based solar power estimation, extending the relevance of the study well beyond the context of just solar cars. By providing a robust, holistic framework for balancing aerodynamic performance with energy generation on complex curved surfaces, this approach can be used to make informed decisions on a broad range of research, for example, solar-powered drones, and systems with interplay between aerodynamics and solar panel power generation. As such, the work contributes a versatile toolset to the wider field of sustainable engineering.

Risk 1: Steep Learning Curve Associated with CFD Software

The complexity of CFD software, which encompasses a broad array of models for simulating real-world fluid flows, presents a considerable learning challenge. Mastery of such tools often requires more time than initially anticipated, posing a risk of project delays.

Mitigation:

To mitigate this risk, it is advisable to allocate additional time for software familiarisation during the initial project planning phase. Should further time be required, timely adjustments to the project schedule should be implemented to accommodate the learning process.

Risk 2: Conflicts with Concurrent Academic Commitments

Undertaking the project alongside other academic responsibilities introduces the potential for overlapping deadlines, which may adversely affect the project timeline and resource allocation.

Mitigation:

This risk can be effectively managed by incorporating the deliverables and key milestones of other academic units into the overall project plan. Proactive scheduling and resource management, as well as anticipating potential conflicts where specific deadlines are not yet defined, can help minimise disruptions.

Risk 3: Insufficient Time for Completion of the Initial Project Scope

The initial project scope was ambitious, encompassing a wide range of objectives. As the project progressed, it became evident that the available time was insufficient to address all planned activities, resulting in the removal of certain tasks that were ultimately deemed redundant or non-essential.

Mitigation:

To address this risk, it is essential to establish a realistic and prioritised project scope at the outset, informed by available resources and time constraints. Regular review and adjustment of the project scope throughout its duration can ensure that efforts remain focused on the most impactful and feasible objectives.

Code Used

Python code used on conceptual geometry

```python
import math
import matplotlib.pyplot as plt

# Constants for thermal calculations
AMBIENT_TEMP = 298.15        # Ambient temperature in Kelvin (25°C)
REFLECTANCE = 0.05           # Fraction of incident light reflected
CONVECTIVE_COEFF = 10.0      # Convective heat transfer coefficient (W/m²·K)
EMISSIVITY = 0.90            # Emissivity of the PV surface
STEFAN_BOLTZMANN = 5.67e-8   # Stefan-Boltzmann constant (W/m²·K)

# Constants for solar power generation
SOLAR_IRRADIANCE = 1000.0    # Direct normal irradiance in W/m²
EFFICIENCY = 0.21      # PV conversion efficiency (21%)

# Efficiency Multipliers Based on Inclination Levels
ANGLE_MULTIPLIERS = {
    15: 0.95,
    30: 0.85,
    45: 0.70,
    60: 0.55
}

# Constants for rolling resistance
GRAVITY = 9.81
TIRE_PRESSURE_KPA = 96.53
COEFFICIENT_ROLLING_RESISTANCE = 0.01
MECHANICAL_LOSSES = 0.10

# SAE J2452 Rolling Resistance Model Coefficients
ALPHA = -0.5
BETA = 1.2
A_COEFF = 0.005
B_COEFF = 0.0003
C_COEFF = 0.00001

# Mechanical Efficiency Parameters
ETA_MIN = 0.80
ETA_MAX = 0.90
```

```
ALPHA_EFF = 0.2

def rolling_resistance(mass, speed):
    if speed == 0:
        return 0

    load_per_tire = (mass * GRAVITY) / 4

    rolling_resistance_force = (A_COEFF + B_COEFF * speed + C_COEFF * speed**2) * (TIRE

    return rolling_resistance_force * 4

def mechanical_efficiency(speed):
    return ETA_MAX - (ETA_MAX - ETA_MIN) * math.exp(-ALPHA_EFF * speed)

def compute_power_needed(drag_force, speed, mass):
    F_rolling = rolling_resistance(mass, speed)
    eta_mech = mechanical_efficiency(speed)
    total_resistive_force = drag_force + F_rolling
    power_required = (total_resistive_force * speed) / eta_mech
    return power_required

def maincalc():
    try:
        total_area = float(input("Enter the total surface area of the solar panels (in
    except ValueError:
        print("Invalid input. Please enter a numerical value for area.")
        return

    area_distribution = {}
    total_percent = 0.0

    print("\nEnter the percentage of total area for each inclination level.")
    for angle in ANGLE_MULTIPLIERS.keys():
        try:
            percent = float(input(f"Percentage of area for {angle}° inclination (0-100)
            if percent < 0 or percent > 100:
                print("Invalid input. Enter a percentage between 0 and 100.")
                return
            area_distribution[angle] = total_area * (percent / 100)
```

```
            total_percent += percent
        except ValueError:
            print("Invalid input. Please enter a numerical value for the percentage.")
            return

    if not math.isclose(total_percent, 100.0, rel_tol=1e-6):
        print("Error: The sum of percentages must equal 100%.")
        return

    total_elec_power = 0.0
    for angle, area in area_distribution.items():
        multiplier = ANGLE_MULTIPLIERS[angle]
        total_elec_power += SOLAR_IRRADIANCE * EFFICIENCY * area * multiplier

    total_elec_kW = total_elec_power / 1000.0
    print(total_elec_kW)

    try:
        n_speeds = int(input("Enter the number of speed scenarios to analyze: "))
        if n_speeds < 1:
            print("Enter at least one speed scenario.")
            return
    except ValueError:
        print("Invalid input. Please enter an integer value for the number of speeds.")
        return

    speeds = []
    power_needed_list = []
    power_deficit_list = []

    for i in range(n_speeds):
        try:
            speed = float(input(f"Enter speed #{i+1} in m/s: "))
            drag_force = float(input(f"Enter drag force for speed #{i+1} (in N): "))
        except ValueError:
            print("Invalid input. Please enter numerical values for speed and drag forc
            return

        speeds.append(speed)

```

```
        power_needed = compute_power_needed(drag_force, speed, 300) / 1000.0
        power_needed_list.append(power_needed)

        power_deficit = total_elec_kW - power_needed
        power_deficit_list.append(power_deficit)

    plt.figure(figsize=(10, 6))

    # Plot power generated as a connected green line
    plt.plot(speeds, [total_elec_kW] * len(speeds), color='green', linestyle='-', marke




    # Plot power needed as red dots
    plt.scatter(speeds, power_needed_list, color='red', label="Power Needed (kW)")

    for i in range(len(speeds)):
        plt.vlines(x=speeds[i], ymin=total_elec_kW, ymax=power_needed_list[i], colors='

        label_color = "purple" if power_needed_list[i] > total_elec_kW else "blue"
        label_text = f"{power_deficit_list[i]:.2f} kW Deficit" if power_needed_list[i]

        plt.text(speeds[i], (total_elec_kW + power_needed_list[i]) / 2, label_text,
                 fontsize=10, color=label_color, ha='center', fontweight='bold')

    plt.xlabel("Speed (m/s)")
    plt.ylabel("Power (kW)")
    plt.title("Power Generated vs. Power Needed")
    plt.grid(True)
    plt.legend()
    plt.show()

if __name__ == "__main__":
    maincalc()

```

Python code used for final integrated analysis of 3 final designs

```
# -*- coding: utf-8 -*-
"""
```

```
Created on Fri Apr 25 21:52:40 2025

@author: amrut
"""

import math
import numpy as np
from stl import mesh
import matplotlib.pyplot as plt
import os # Added for checking file existence

CASES = [
    {
        "name": "Geometry 1",
        "panel_stl": "Shell_Eco_Car_2015_just_mesh.stl",
        "drag_force": 11.49
    },
    {
        "name": "Geometry 2",
        "panel_stl": "Shell Eco Car_2015_geom_2_mesh.stl",
        "drag_force": 14.49
    },
    {
        "name": "Geometry 3",
        "panel_stl": "Shell Eco Car 2015_geom_3_mesh.stl",
        "drag_force": 10.91
    }
]

# --- Constants & Parameters ---
MASS = 300.0     # Vehicle mass in kg (used for rolling resistance calculation)
SPEED = 16     # Vehicle speed in m/s
SOLAR_IRRADIANCE = 1000.0     # W/m²
ALPHA_PV = 0.95     # Absorptivity
TAU_PV = 1.0     # Transmissivity
SUN_VECTOR = np.array([0, 0, 1])  # Sun directly overhead (Z-axis)
STL_SCALE = 0.001 # Scale factor for STL units (e.g., 0.001 if mm, 1.0 if m)

# Rolling resistance and mechanical efficiency parameters
GRAVITY = 9.81
```

```
TIRE_PRESSURE_KPA = 96.53
A_COEFF = 0.005
B_COEFF = 0.0003
C_COEFF = 0.00001
ALPHA = -0.5
BETA = 1.2
ETA_MIN = 0.80
ETA_MAX = 0.90
ALPHA_EFF = 0.2

# PV Cell Types and Efficiencies
PV_CELLS = {
    'Monocrystalline Si': 0.21,
    'Polycrystalline Si': 0.16,
    'GaAs': 0.291,
    'Thin Film (CIGS/CdTe)': 0.11
}
COMPARISON_CELL_TYPE = 'GaAs' # Cell type used for the final comparison plot

# --- Functions ---

def pv_power(I_solar, area, normal, sun_vec, efficiency):
    """Calculate electrical power for a single triangle."""

    norm_normal = np.linalg.norm(normal)
    if norm_normal == 0: return 0.0
    unit_normal = normal / norm_normal

    angle_cos = np.dot(unit_normal, sun_vec)


    if angle_cos <= 0:
        return 0.0

    absorbed = I_solar * ALPHA_PV * TAU_PV * area * angle_cos
    power = absorbed * efficiency
    return power

def rolling_resistance(mass, speed):
    if speed == 0:
```

```
        return 0
    load_per_tire = (mass * GRAVITY) / 4
    # Clamp speed for stability if needed, though coefficients might handle it
    # speed = max(speed, 0.1) # Example if needed
    try:
        rolling_resistance_force = (A_COEFF + B_COEFF * speed + C_COEFF * speed**2) * (
    except ValueError: # Catch potential issues with negative bases to fractional powe
         print(f"Warning: Potential math domain error in rolling resistance calculation
         # Provide a fallback or default value if necessary
         rolling_resistance_force = A_COEFF * (TIRE_PRESSURE_KPA**ALPHA) * (load_per_ti

    return rolling_resistance_force * 4

def mechanical_efficiency(speed):
    return ETA_MAX - (ETA_MAX - ETA_MIN) * math.exp(-ALPHA_EFF * max(0, speed)) # Ensur

def compute_power_needed(drag_force, speed, mass):
    F_rolling = rolling_resistance(mass, speed)
    eta_mech = mechanical_efficiency(speed)
    if eta_mech == 0: # Avoid division by zero
        print("Warning: Mechanical efficiency is zero. Cannot compute power needed.")
        return float('inf') # Or handle as appropriate
    total_resistive_force = drag_force + F_rolling
    power_required = (total_resistive_force * speed) / eta_mech
    return power_required # Return in Watts

def read_stl(filename, scale=1.0):
    """Reads STL, scales vertices, returns triangle areas and unit normals."""
    if not os.path.exists(filename):
        print(f"Error: STL file not found at {filename}")
        return None, None
    try:
        your_mesh = mesh.Mesh.from_file(filename)
    except Exception as e:
        print(f"Error reading STL file {filename}: {e}")
        return None, None

    triangles = your_mesh.vectors * scale
    # Use the normals directly from the mesh object, they should be unit normals
    normals = your_mesh.normals
```

```python
    areas = []
    for i, tri in enumerate(triangles):
        # Calculate area using cross product for robustness
        v1 = tri[1] - tri[0]
        v2 = tri[2] - tri[0]
        cross_prod = np.cross(v1, v2)
        area = 0.5 * np.linalg.norm(cross_prod)
        areas.append(area)

        # Optional: Check if calculated normal matches mesh normal direction
        # calc_normal = cross_prod / np.linalg.norm(cross_prod) if np.linalg.norm(cros
        # if np.dot(calc_normal, normals[i]) < 0.9: # Allow for some tolerance
        #     print(f"Warning: Triangle {i} in {filename} might have inconsistent nor

    return areas, normals

def calculate_stl_total_area(filename, scale=1.0):
    """Reads STL and calculates the total surface area."""
    areas, _ = read_stl(filename, scale)
    if areas is None:
        return 0.0 # Return 0 area if file reading failed
    return sum(areas)

def plot_individual_case(case_name, results, power_needed_kw):
    """Generates the power generated vs needed plot for a single case."""
    labels = [r['Cell'] for r in results]
    power_gen_kw = [r['Electrical Power (kW)'] for r in results]
    power_need_kw_list = [power_needed_kw for _ in results]
    diff_kw = [p - power_needed_kw for p in power_gen_kw]

    x = np.arange(len(labels))
    width = 0.35

    fig, ax = plt.subplots(figsize=(11, 5))
    bars1 = ax.bar(x - width/2, power_gen_kw, width, label='Power Generated (kW)', colo
    bars2 = ax.bar(x + width/2, power_need_kw_list, width, label='Power Needed (kW)', c

    # Adjust y-limit based on data range
    max_val = max(max(power_gen_kw, default=0), max(power_need_kw_list, default=0))
    min_val = min(min(power_gen_kw, default=0), min(power_need_kw_list, default=0))
```

```
    ax.set_ylim(min(0, min_val) - max_val*0.1, max_val * 1.25) # Ensure 0 is visible,

    # Annotate differences
    for i, d in enumerate(diff_kw):
        y_pos = max(power_gen_kw[i], power_need_kw_list[i]) + max_val * 0.05
        ax.text(x[i], y_pos, f"{d:+.2f} kW", ha='center', va='bottom', fontsize=9, font

    ax.set_xticks(x)
    ax.set_xticklabels(labels, rotation=25, ha='right')
    ax.set_ylabel('Power (kW)')
    ax.set_title(f'Power Analysis for {case_name}')
    ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
    plt.tight_layout(rect=[0, 0, 0.85, 1]) # Adjust layout to make room for legend
    plt.show()

def plot_comparison(comparison_data):
    import matplotlib.pyplot as plt
    import numpy as np

    case_names = [d['name'] for d in comparison_data]
    panel_areas = [d['panel_area'] for d in comparison_data]
    drag_forces = [d['drag_force'] for d in comparison_data]

    x = np.arange(len(case_names))
    width = 0.5

    fig, ax = plt.subplots(figsize=(10, 5))

    # Bar chart for Solar Panel Area
    bars = ax.bar(x, panel_areas, width, label='Solar Panel Area (m²)', color='tab:gree
    ax.set_xlabel('Case')
    ax.set_ylabel('Solar Panel Area (m²)', color='tab:green')
    ax.set_xticks(x)
    ax.set_xticklabels(case_names)
    ax.set_ylim(0, max(panel_areas) * 1.2)
    ax.set_title('Comparison of Solar Panel Area Across Cases')

    # Annotate each bar with the drag force value
    for i, bar in enumerate(bars):
        height = bar.get_height()
```

```
        ax.annotate(f"Drag Force: {drag_forces[i]:.2f} N",
                    xy=(bar.get_x() + bar.get_width() / 2, height),
                    xytext=(0, 5),  # 5 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom', fontsize=10, color='black', fontweight='b

    ax.legend(loc='upper left')
    plt.tight_layout()
    plt.show()
# --- Main Execution Logic ---
def main():
    comparison_results = [] # To store data for the final comparison plot
    # Normalize sun vector once
    unit_sun_vector = SUN_VECTOR / np.linalg.norm(SUN_VECTOR)

    for case in CASES:
        print(f"\n--- Processing {case['name']} ---")
        print(f"Panel STL: {case['panel_stl']}, Drag Force: {case['drag_force']} N")

        # 1. Calculate Solar Panel Area
        panel_area_total = calculate_stl_total_area(case['panel_stl'], scale=STL_SCALE)
        if panel_area_total == 0:
             # Check if file exists before warning about zero area
             if not os.path.exists(case['panel_stl']):
                 print(f"Error: Panel STL file not found: {case['panel_stl']}. Skipping
                 continue # Skip to the next case
             else:
                 print(f"Warning: Calculated 0 area for panel file {case['panel_stl']}.
        print(f"Total Solar Panel Area: {panel_area_total:.4f} m²")

        # 2. Calculate Power Needed
        power_needed_watts = compute_power_needed(case['drag_force'], SPEED, MASS)
        power_needed_kw = power_needed_watts / 1000.0
        print(f"Power Needed: {power_needed_kw:.3f} kW")

        # 3. Calculate Power Generated for each PV type using the panel STL
        areas, normals = read_stl(case['panel_stl'], scale=STL_SCALE)
        if areas is None or normals is None:
            print(f"Error reading panel STL {case['panel_stl']}. Skipping power generat
            # Still add placeholder data to comparison if needed, or skip fully
```

```
            comparison_results.append({
                'name': case['name'],
                'panel_area': panel_area_total,
                'generated_power_kw': 0, # Placeholder
                'drag_force': case['drag_force']
            })
            continue # Skip rest of the loop for this case

        case_pv_results = []
        generated_power_for_comparison = 0.0 # Track power for the specific cell type

        for cell_name, efficiency in PV_CELLS.items():
            total_power_watts = 0.0
            for area, normal in zip(areas, normals):
                # Pass the unit sun vector to pv_power
                power = pv_power(SOLAR_IRRADIANCE, area, normal, unit_sun_vector, effic
                total_power_watts += power

            total_elec_kW = total_power_watts / 1000.0
            surplus = total_elec_kW - power_needed_kw
            case_pv_results.append({
                'Cell': cell_name,
                'Efficiency': efficiency,
                'Electrical Power (kW)': total_elec_kW,
                'Surplus/Deficit (kW)': surplus
            })
            print(f"  {cell_name}: Generated={total_elec_kW:.3f} kW, Surplus/Deficit={s

            # Store the power generated by the designated cell type for the compariso
            if cell_name == COMPARISON_CELL_TYPE:
                generated_power_for_comparison = total_elec_kW

        # 4. Store results for the final comparison plot
        comparison_results.append({
            'name': case['name'],
            'panel_area': panel_area_total,
            'generated_power_kw': generated_power_for_comparison,
            'drag_force': case['drag_force']
        })

```

```
        # 5. Generate the individual plot for this case
        if case_pv_results: # Only plot if results were generated
            plot_individual_case(case['name'], case_pv_results, power_needed_kw)
        else:
            print(f"Skipping individual plot for {case['name']} due to missing PV resul

    # Print summary of results
    print("\n=== SUMMARY OF RESULTS ===")
    print("-" * 80)
    print(f"{'Design':<15} {'Cell Type':<20} {'Power Generated (kW)':<20} {'Power Neede
    print("-" * 80)

    for case in comparison_results:
        case_name = case['name']
        panel_area = case['panel_area']
        drag_force = case['drag_force']
        power_needed = compute_power_needed(drag_force, SPEED, MASS) / 1000.0  # Conver

        print(f"\n{case_name}")
        print(f"Panel Area: {panel_area:.2f} m²")
        print(f"Drag Force: {drag_force:.2f} N")
        print(f"Power Needed: {power_needed:.2f} kW")
        print("\nPower Generation by Cell Type:")
        print("-" * 40)

        areas, normals = read_stl(CASES[comparison_results.index(case)]['panel_stl'], s
        if areas is not None and normals is not None:
            for cell_name, efficiency in PV_CELLS.items():
                total_power_watts = sum(pv_power(SOLAR_IRRADIANCE, area, normal, unit_s
                                      for area, normal in zip(areas, normals))
                total_power_kw = total_power_watts / 1000.0
                surplus = total_power_kw - power_needed
                print(f"{cell_name:<20}: {total_power_kw:>6.2f} kW (Surplus/Deficit: {s
        print("-" * 40)

    print("\n=== END OF SUMMARY ===\n")


    # 6. Generate the final comparison plot
    if comparison_results: # Only plot if there's data
```

```
        print("\n--- Generating Comparison Plot ---")
        plot_comparison(comparison_results)
    else:
        print("\nNo data available to generate comparison plot.")

if __name__ == "__main__":
    main()
```

STAR-CCM+ Summary Report

The following summary report lists the STAR-CCM+ settings used during the 3-D CFD evaluation of the solar-powered vehicle. It is based on the settings used for the CFD analysis of the final 3 designs.

Summary Report: geom1@30000

Session Summary	
Date	1 May 2025, 11:21:40
Simulation	C:\Users\amrut\OneDrive - The University of Manchester\Thesis\geom1@30000.sim
File size	4.8e+02 MB
Number of Partitions	3
Number of Restored Partitions	4
Software Summary	
Version	BuildArch: win64 BuildEnv: clang15.0vc14.2 PresentationVersion: 2406.0001 ReleaseDate: Tue Jul 16 14:45:25 UTC 2024 ReleaseNumber: 19.04.009
MPI Version	MS MPI-10.1.12498.16
Hardware Summary	
Hosts	Number Processes: 3 Rank[0]: AmruthLaptop Rank[1]: AmruthLaptop Rank[2]: AmruthLaptop

Simulation Properties

```
1 geom1@30000
+-1 Continua
| `-1 Physics 1
```

Continua	1
Regions	[Region]
Interfaces	[]
Point Sets	[]
Active	true
Motion Always Active	false
Tags	[]

```
|   +-1 Models
|   | +-1 All y+ Wall Treatment
```

Iterative Ustar	false

```
|   | +-2 Constant Density
|   | +-3 Gamma-ReTheta Transition
```

Sigma_f	1.0
ca2	0.06
ce2	50.0
Intermittency Minimum	1.0E-10
Secondary Gradients	On
Convection	2nd-order
Cross-Flow Term	false
Correlation Method	Suluksna-Juntasaro
Sigma_ReTheta	2.0
ca1	2.0
ce1	1.0
cTheta_t	0.03
s1	2.0
Conset1	2.193
ReThetaT Minimum	20.0

```
|   | +-4 Gas
|   | | `-1 Air
```

Database Material	Air (Air) [Standard/Gases]
Tags	[]

```
|    | |    `-1 Material
Properties
|    | |       +-1 Density           Method                                  Constant
|    | |       | `-1 Constant        Value                                   1.18415 kg/m^3
|    | |       `-2 Dynamic           Method                                  Constant
Viscosity
|    | |         `-1 Constant        Value                                   1.85508E-5 Pa-s
|    | +-5 Gradients                 Boundary Cell Eigenvalues Ratio         0.06
                                     Tolerance
|    | |                             Limit GradVar By Beta                   true
|    | |                             Limiter Method                          Venkatakrishnan
|    | |                             Custom Accuracy Level Selector          2.0
|    | |                             Maximum Reconstruction Coefficient      1.0
|    | |                             Two Pass Velocity Gradient              false
|    | |                             Use TVB Gradient Limiting               false
|    | |                             Acceptable Field Variation (Factor)     0.05
|    | +-6 K-Omega Turbulence
|    | +-7 Reynolds-Averaged
Navier-Stokes
|    | +-8 Segregated Flow           Flux Update Dissipation Factor          0.5
|    | |                             Positivity Rate Limit                   0.2
|    | |                             Flow Boundary Diffusion                 true
|    | |                             Unsteady Flux Dissipation Corrections   false
|    | |                             Limit Acoustic-CFL Option               Per-Model
|    | |                             Secondary Gradients                     On
|    | |                             Convection                              2nd-order
|    | |                             Delta-V Dissipation                     Off
|    | +-9 Solution Interpolation    Per Part Mapping                        false
|    | |                             Interpolation Method                    Nearest neighbor
|    | |                             Legacy Method                           false
|    | |                             Conservation Correction                 Disable
|    | +-10 SST (Menter) K-          Curvature Correction Option             Off
Omega
|    | | |                           Realizability Option                    Durbin Scale Limiter
|    | | |                           Compressibility Correction              true
|    | | |                           Low Re Damping Modification             false
|    | | |                           Convection                              2nd-order
|    | | |                           Normal Stress Term                      false
|    | | |                           Tke Minimum                             1.0E-10
|    | | |                           Sdr Minimum                             1.0E-10
|    | | |                           Secondary Gradients                     On
|    | | |                           Kappa                                   0.41
|    | | |                           BetaStar                                0.09
|    | | |                           Beta1                                   0.075
|    | | |                           Sigma_k1                                0.85
|    | | |                           Sigma_w1                                0.5
|    | | |                           Beta2                                   0.0828
|    | | |                           Sigma_k2                                1.0
|    | | |                           Sigma_w2                                0.856
|    | | |                           Constitutive Option                     Linear
|    | | |                           a1                                      0.31
|    | | +-1 Compressibility         Zeta_Star                               1.5
Parameters
|    | | `-2 Realizability           Realizability Coefficient               0.6000000238418579
Coefficient
|    | +-11 Steady                   Continuum Iteration                     30004
|    | +-12 Three Dimensional
|    | +-13 Turbulent
|    | `-14 Wall Distance            Wall Distance Method                    Implicit Tree
|    +-2 Reference Values
|    | +-1 Minimum Allowable         Value                                   1.0E-6 m
Wall Distance
```

| | +-2 Maximum Allowable Absolute Pressure	Value	1.0E8 Pa
| | +-3 Minimum Allowable Absolute Pressure	Value	1000.0 Pa
| | `-4 Reference Pressure	Value	101325.0 Pa
| `-3 Initial Conditions		
| +-1 Intermittency	Method	Constant
| | `-1 Constant	Value	1.0
| +-2 Pressure	Method	Constant
| | `-1 Constant	Value	0.0 Pa
| +-3 Turbulence Intensity	Method	Constant
| | `-1 Constant	Value	0.01
| +-4 Turbulence Specification	Method	Intensity + Viscosity Ratio
| +-5 Turbulent Velocity Scale	Method	Constant
| | `-1 Constant	Value	1.0 m/s
| +-6 Turbulent Viscosity Ratio	Method	Constant
| | `-1 Constant	Value	10.0
| `-7 Velocity	Method	Constant
| |	Coordinate System	Laboratory
| `-1 Constant	Value	[16.0, 0.0, 0.0] m/s
+-2 Regions	Part Selection Priority	[Region]
| |	Regions	1
| `-1 **Region**	Index	7
| |	Allow Per-Part Values	false
| |	Physics Continuum	[Physics 1]
| |	Parts	[Subtract]
| |	Type	Fluid Region
| |	Topology	VOLUME
| |	Tags	[]
| +-1 Boundaries	Part Surface Selection Priority	[Region: Subtract.Body 1.Default, Region: Subtract.Domain.Floor, Region: Subtract.Domain.Symm Plane, Region: Subtract.Domain.Symm Plane 2, Region: Subtract.Domain.Symm Plane 3, Region: Subtract.Domain.Outlet, Region: Subtract.Domain.Inlet, Region: Subtract.Body 1.Default 2]
| | |	Boundaries	8
| | +-1 **Subtract.Body 1.Default**	Index	51
| | | |	Interfaces	
| | | |	Part Surfaces	[Subtract.Body 1.Default]
| | | |	Type	Wall
| | | |	Allow Per-Surface Values	false
| | | |	Topology	SURFACE
| | | |	Tags	[]
| | | +-1 Physics Conditions		
| | | | +-1 Reference Frame Specification	Option	Region Reference Frame
| | | | +-2 Shear Stress Specification	Method	No-Slip
| | | | +-3 Tangential Velocity Specification	Method	Fixed
| | | | `-4 Wall Surface Specification	Method	Smooth
| | | `-2 Physics Values		
| | | `-1 Blended Wall Function	E	9.0
| | |	Kappa	0.42

```
 |     | +-2 Subtract.Body          Index                        58
1.Default 2
 |     | |                          Interfaces
 |     | |                          Part Surfaces                [Subtract.Body 1.Default 2]
 |     | |                          Type                         Symmetry Plane
 |     | |                          Allow Per-Surface Values     false
 |     | |                          Topology                     SURFACE
 |     | |                          Tags                         []
 |     |                            Index                        52
 +-3 Subtract.Domain.Floor
 |     | | |                        Interfaces
 |     | | |                        Part Surfaces                [Subtract.Domain.Floor]
 |     | | |                        Type                         Wall
 |     | | |                        Allow Per-Surface Values     false
 |     | | |                        Topology                     SURFACE
 |     | | |                        Tags                         []
 |     | | +-1 Physics Conditions
 |     | | | +-1 Reference Frame    Option                       Region Reference Frame
Specification
 |     | | | +-2 Shear Stress       Method                       No-Slip
Specification
 |     | | | +-3 Tangential         Method                       Vector
Velocity Specification
 |     | | | `-4 Wall Surface       Method                       Smooth
Specification
 |     | | `-2 Physics Values
 |     | |     +-1 Blended Wall     E                            9.0
Function
 |     | |     |                    Kappa                        0.42
 |     | |     `-2 Relative Velocity Method                      Constant
 |     | |         |                Coordinate System            Laboratory
 |     | |         `-1 Constant     Value                        [-16.0, 0.0, 0.0] m/s
 |     |                            Index                        57
 +-4 Subtract.Domain.Inlet
 |     | | |                        Interfaces
 |     | | |                        Part Surfaces                [Subtract.Domain.Inlet]
 |     | | |                        Type                         Velocity Inlet
 |     | | |                        Allow Per-Surface Values     false
 |     | | |                        Topology                     SURFACE
 |     | | |                        Tags                         []
 |     | | +-1 Physics Conditions
 |     | | | +-1 Flow Direction     Method                       Boundary-Normal
Specification
 |     | | | +-2 Reference Frame    Option                       Lab Frame
Specification
 |     | | | +-3 Turbulence         Method                       Intensity + Viscosity Ratio
Specification
 |     | | | `-4 Velocity           Method                       Magnitude + Direction
Specification
 |     | | `-2 Physics Values
 |     | |     +-1 Turbulence       Method                       Constant
Intensity
 |     | |     | `-1 Constant       Value                        0.01
 |     | |     +-2 Turbulent        Method                       Constant
Viscosity Ratio
 |     | |     | `-1 Constant       Value                        10.0
 |     | |     `-3 Velocity         Method                       Constant
Magnitude
 |     | |         `-1 Constant     Value                        16.0 m/s
 |     |                            Index                        56
 +-5 Subtract.Domain.Outlet
 |     | | |                        Interfaces
 |     | | |                        Part Surfaces                [Subtract.Domain.Outlet]
```

```
|    | | |                          Type                        Pressure Outlet
|    | | |                          Allow Per-Surface Values    false
|    | | |                          Topology                    SURFACE
|    | | |                          Tags                        []
|    | | +-1 Physics Conditions
|    | | | +-1 Backflow             Direction                   Boundary-Normal
Specification
|    | | | |                        Pressure                    Environmental
|    | | | |                        Scalars                     Specified
|    | | | +-2 Pressure Outlet      Option                      None
Option
|    | | | +-3 Reference Frame      Option                      Lab Frame
Specification
|    | | | `-4 Turbulence           Method                      Intensity + Viscosity Ratio
Specification
|    | | `-2 Physics Values
|    | |     +-1 Pressure           Method                      Constant
|    | |     | `-1 Constant         Value                       0.0 Pa
|    | |     +-2 Turbulence         Method                      Constant
Intensity
|    | |     | `-1 Constant         Value                       0.01
|    | |     `-3 Turbulent          Method                      Constant
Viscosity Ratio
|    | |       `-1 Constant         Value                       10.0
|    |                              Index                       53
 +-6 Subtract.Domain.Symm
Plane
|    | |                            Interfaces
|    | |                            Part Surfaces               [Subtract.Domain.Symm Plane]
|    | |                            Type                        Symmetry Plane
|    | |                            Allow Per-Surface Values    false
|    | |                            Topology                    SURFACE
|    | |                            Tags                        []
|    |                              Index                       54
 +-7 Subtract.Domain.Symm
Plane 2
|    | |                            Interfaces
|    | |                            Part Surfaces               [Subtract.Domain.Symm Plane 2]
|    | |                            Type                        Symmetry Plane
|    | |                            Allow Per-Surface Values    false
|    | |                            Topology                    SURFACE
|    | |                            Tags                        []
|    |                              Index                       55
 `-8 Subtract.Domain.Symm
Plane 3
|    |                              Interfaces
|    |                              Part Surfaces               [Subtract.Domain.Symm Plane 3]
|    |                              Type                        Symmetry Plane
|    |                              Allow Per-Surface Values    false
|    |                              Topology                    SURFACE
|    |                              Tags                        []
|    +-2 Physics Conditions
|    | +-1 Initial Condition Option Option                      Use Continuum Values
|    | +-2 Mass Source Option       Mass Source Option          false
|    | +-3 Momentum Source          Momentum Source Option      None
Option
|    | +-4 Motion Specification     Option                      Motion Specification
Option
|    | `-5 Turbulence Source        Turbulence Source Option    None
Option
|    `-3 Physics Values
|      +-1 Axis                     Coordinate System           Laboratory
|      |                            Origin                      [0.0, 0.0, 0.0] m
```

```
 |       |                          Direction                [0.0, 0.0, 1.0]
 |       `-2 Motion Specification   Motion                   Stationary
 |                                  Reference Frame          Lab Reference Frame
 +-3 Representations
 | +-1 Latest Surface/Volume        Representation           Volume Mesh
 | |                                Tags                     []
 | +-2 Geometry                     Tags                     []
 | | +-1 Automated                  Tags                     []
Mesh.Remesh
 | | +-2 Input Surface              Tags                     []
 | | `-3 Latest Surface             Tags                     []
 | `-3 Volume Mesh                  Cells                    2975699
 |   |                              Interior Faces           6367616
 |   |                              Vertices                 824409
 |   |                              Tags                     []
 |   +-1 Finite Volume Regions
 |   | `-1 Region                   Cells                    2975699
 |   |   |                          Interior Faces           6367616
 |   |   |                          Vertices                 824409
 |   |   |                          Edges                    0
 |   |   `-1 Finite Volume
Boundaries
 |   |       +-1 Subtract.Body      Faces                    37600
1.Default
 |   |       +-2 Subtract.Body      Faces                    37600
1.Default 2
 |   |                              Faces                    27233
   +-3 Subtract.Domain.Floor
 |   |                              Faces                    806
   +-4 Subtract.Domain.Inlet
 |   |                              Faces                    3257
   +-5 Subtract.Domain.Outlet
 |   |                              Faces                    2913
   +-6 Subtract.Domain.Symm
Plane
 |   |                              Faces                    2915
   +-7 Subtract.Domain.Symm
Plane 2
 |   |                              Faces                    4104
   `-8 Subtract.Domain.Symm
Plane 3
 |   `-2 Cell Sets
 +-4 Automation
 | +-1 Parameters
 | +-2 Field Functions
 | +-3 Simulation Operations        Selected                 []
 | +-4 Filters
 | +-5 Tags
 | +-6 Stages                       Active Stage             []
 | +-7 Update Events                Event Count              0
 | |                                Event Names
 | `-8 Time Scales
 +-5 Contacts
 | +-1 Subtract/Subtract            Part 1                   [Subtract]
 | | |                              Part 2                   [Subtract]
 | | |                              Index                    1
 | | `-1 Body 1.Default/Body        Part Surface 1           Subtract.Body 1.Default
1.Default
 | |                                Part Surface 2           Subtract.Body 1.Default 2
 | |                                Part Surface 2           Subtract.Body 1.Default 2
 | |                                Conformality             STRONG_CONTACT
 | |                                Show Baffle Proxy Surface false
 | |                                Metadata                 {}
```

```
| |                              Index                      1
| |                              Interface                  []
| |                              Tags                       []
| `-2 Subtract 2/Subtract 2      Part 1                     [Subtract 2]
|    |                           Part 2                     [Subtract 2]
|    |                           Index                      2
|    `-1 Body 1.Default/Body     Part Surface 1             Subtract 2.Body 1.Default
1.Default
|                                Part Surface 2             Subtract 2.Body 1.Default 2
|                                Part Surface 2             Subtract 2.Body 1.Default 2
|                                Conformality               STRONG_CONTACT
|                                Show Baffle Proxy Surface  false
|                                Metadata                   {}
|                                Index                      2
|                                Interface                  []
|                                Tags                       []
+-6 Parts
| +-1 Block                      Metadata                   {}
| | |                            Index                      12
| | |                            Color                      java.awt.Color[r=112,g=128,b=144]
| | |                            Is Shell                   false
| | |                            Region                     []
| | |                            Contacts                   []
| | |                            Descriptions               [Root]
| | |                            Face Count                 12
| | |                            Coordinate System          Laboratory
| | |                            Corner 1                   [-1.0361710812055307,
                                                            -0.11000028252601624,
                                                            -0.5099959969520569] m,m,m
| | |                            Corner 2                   [3.671794238432124,
                                                            0.9859394670044237,
                                                            0.5099824070930481] m,m,m
| | |                            Tags                       []
| | +-1 Surfaces
| | | `-1 Block Surface          Index                      27
| | |                            Metadata                   {}
| | |                            Boundary                   []
| | |                            Color                      java.awt.Color[r=112,g=128,b=144]
| | |                            Tags                       []
| | `-2 Curves
| |     `-1 Block Curve          Index                      19
| |                              Tags                       []
| +-2 Body 1                     Metadata                   {}
| | |                            Index                      1
| | |                            Color                      java.awt.Color[r=255,g=255,b=255]
| | |                            Is Shell                   false
| | |                            Region                     []
| | |                            Contacts                   []
| | |                            Descriptions               [Root]
| | |                            Face Count                 18471
| | |                            Tags                       []
| | +-1 Surfaces
| | | `-1 Default                Index                      1
| | |                            Metadata                   {}
| | |                            Boundary                   []
| | |                            Color                      java.awt.Color[r=112,g=128,b=144]
| | |                            Tags                       []
| | `-2 Curves
| |     `-1 Default              Index                      1
| |                              Tags                       []
| +-3 Domain                     Metadata                   {}
```

```
| | |                    Index               10
| | |                    Color               java.awt.Color[r=112,g=128,b=144]
| | |                    Is Shell            false
| | |                    Region              []
| | |                    Contacts            []
| | |                    Descriptions        [Root]
| | |                    Face Count          12
| | |                    Coordinate System   Laboratory
| | |                    Corner 1            [-8.711754888758001,
                                             -0.2002169616140057,
                                             -2.23271560969431] m,m,m
| | |                    Corner 2            [8.571657842142198, 3.0,
                                             2.477455827612582] m,m,m
| | |                    Tags                []
| | +-1 Surfaces
| | | +-1 Floor          Index               13
| | | | |                Metadata            {}
| | | | |                Boundary            []
| | | | |                Color               java.awt.Color[r=112,g=128,b=144]
| | | | |                Tags                []
| | | +-2 Inlet          Index               18
| | | | |                Metadata            {}
| | | | |                Boundary            []
| | | | |                Color               java.awt.Color[r=112,g=128,b=144]
| | | | |                Tags                []
| | | +-3 Outlet         Index               17
| | | | |                Metadata            {}
| | | | |                Boundary            []
| | | | |                Color               java.awt.Color[r=112,g=128,b=144]
| | | | |                Tags                []
| | | +-4 Symm Plane     Index               14
| | | | |                Metadata            {}
| | | | |                Boundary            []
| | | | |                Color               java.awt.Color[r=112,g=128,b=144]
| | | | |                Tags                []
| | | +-5 Symm Plane 2   Index               15
| | | | |                Metadata            {}
| | | | |                Boundary            []
| | | | |                Color               java.awt.Color[r=112,g=128,b=144]
| | | | |                Tags                []
| | | `-6 Symm Plane 3   Index               16
| | |                    Metadata            {}
| | |                    Boundary            []
| | |                    Color               java.awt.Color[r=112,g=128,b=144]
| | |                    Tags                []
| | `-2 Curves
| |   `-1 Block Curve    Index               14
| |                      Tags                []
| +-4 Subtract           Metadata            {}
| | |                    Index               11
| | |                    Color               java.awt.Color[r=112,g=128,b=144]
| | |                    Is Shell            false
| | |                    Region              [Region]
| | |                    Contacts            [Subtract]
| | |                    Descriptions        [Root, Automated Mesh.Remesh]
| | |                    Face Count          18483
| | |                    Tags                []
| | +-1 Surfaces
| | | +-1 Body 1.Default Index               19
| | | | |                Metadata            {}
| | | | |                Boundary            [Region: Subtract.Body 1.Default]
```

```
| | | |                         Color                              java.awt.Color[r=255,g=255,b=255]
| | | |                         Tags                               []
| | | +-2 Domain.Floor          Index                              20
| | | |                         Metadata                           {}
| | | |                         Boundary                           [Region: Subtract.Domain.Floor]
| | | |                         Color                              java.awt.Color[r=112,g=128,b=144]
| | | |                         Tags                               []
| | | +-3 Domain.Inlet          Index                              25
| | | |                         Metadata                           {}
| | | |                         Boundary                           [Region: Subtract.Domain.Inlet]
| | | |                         Color                              java.awt.Color[r=112,g=128,b=144]
| | | |                         Tags                               []
| | | +-4 Domain.Outlet         Index                              24
| | | |                         Metadata                           {}
| | | |                         Boundary                           [Region: Subtract.Domain.Outlet]
| | | |                         Color                              java.awt.Color[r=112,g=128,b=144]
| | | |                         Tags                               []
| | | +-5 Domain.Symm Plane     Index                              21
| | | |                         Metadata                           {}
| | | |                         Boundary                           [Region: Subtract.Domain.Symm Plane]
| | | |                         Color                              java.awt.Color[r=112,g=128,b=144]
| | | |                         Tags                               []
| | | +-6 Domain.Symm Plane     Index                              22
2
| | | |                         Metadata                           {}
| | | |                         Boundary                           [Region: Subtract.Domain.Symm Plane
                                                                   2]
| | | |                         Color                              java.awt.Color[r=112,g=128,b=144]
| | | |                         Tags                               []
| | | `-7 Domain.Symm Plane     Index                              23
3
| | |                           Metadata                           {}
| | |                           Boundary                           [Region: Subtract.Domain.Symm Plane
                                                                   3]
| | |                           Color                              java.awt.Color[r=112,g=128,b=144]
| | |                           Tags                               []
| | `-2 Curves
| |   +-1 Body 1.Default        Index                              15
| |   |                         Tags                               []
| |   `-2 Domain.Block Curve    Index                              17
| |                             Tags                               []
| `-5 Subtract 2                Metadata                           {}
|   |                           Index                              13
|   |                           Color                              java.awt.Color[r=112,g=128,b=144]
|   |                           Is Shell                           false
|   |                           Region                             []
|   |                           Contacts                           [Subtract 2]
|   |                           Descriptions                       [Root]
|   |                           Face Count                         18483
|   |                           Operation                          Subtract
|   |                           Tags                               []
+-7 3D-CAD Models
| `-1 3D-CAD Model 1            Distinguish Bodies Color Palette   High Contrast Color Palette
|   |                           Part Update Method                 UPDATE_GEOMETRY
|   |                           Tags                               []
|   +-1 Body Groups
|   | +-1 Shell Eco Car 2015    Name                               Shell Eco Car 2015
|   | |                         IsAssignedColor                    False
|   | |                         Color                              java.awt.Color[r=128,g=128,b=128]
|   | |                         Opacity                            1.0
|   | |                         Display Resolution                 VERY_COARSE
```

```
|     | |                          Multi-Instanced       false
|     | |                          Tags                  []
|     | +-2 Shell Eco Car 2015 2   Name                  Shell Eco Car 2015 2
|     | |                          IsAssignedColor       False
|     | |                          Color                 java.awt.Color[r=128,g=128,b=128]
|     | |                          Opacity               1.0
|     | |                          Display Resolution    VERY_COARSE
|     | |                          Multi-Instanced       false
|     | |                          Tags                  []
|     | +-3 Shell Eco Car 2015 3   Name                  Shell Eco Car 2015 3
|     | |                          IsAssignedColor       False
|     | |                          Color                 java.awt.Color[r=128,g=128,b=128]
|     | |                          Opacity               1.0
|     | |                          Display Resolution    VERY_COARSE
|     | |                          Multi-Instanced       false
|     | |                          Tags                  []
|     | +-4 Shell Eco Car 2015 4   Name                  Shell Eco Car 2015 4
|     | |                          IsAssignedColor       False
|     | |                          Color                 java.awt.Color[r=128,g=128,b=128]
|     | |                          Opacity               1.0
|     | |                          Display Resolution    VERY_COARSE
|     | |                          Multi-Instanced       false
|     | |                          Tags                  []
|     | +-5 Shell Eco Car 2015 5   Name                  Shell Eco Car 2015 5
|     | |                          IsAssignedColor       False
|     | |                          Color                 java.awt.Color[r=128,g=128,b=128]
|     | |                          Opacity               1.0
|     | |                          Display Resolution    VERY_COARSE
|     | |                          Multi-Instanced       false
|     | |                          Tags                  []
|     | +-6 Shell Eco Car 2015 6   Name                  Shell Eco Car 2015 6
|     | |                          IsAssignedColor       False
|     | |                          Color                 java.awt.Color[r=128,g=128,b=128]
|     | |                          Opacity               1.0
|     | |                          Display Resolution    VERY_COARSE
|     | |                          Multi-Instanced       false
|     | |                          Tags                  []
|     | +-7 Body 1                 Name                  Body 1
|     | |                          IsAssignedColor       False
|     | |                          Color                 java.awt.Color[r=128,g=128,b=128]
|     | |                          Opacity               1.0
|     | |                          Display Resolution    VERY_COARSE
|     | |                          Multi-Instanced       false
|     | |                          Tags                  []
|     | +-8 Body 2                 Name                  Body 2
|     | |                          IsAssignedColor       False
|     | |                          Color                 java.awt.Color[r=128,g=128,b=128]
|     | |                          Opacity               1.0
|     | |                          Display Resolution    VERY_COARSE
|     | |                          Multi-Instanced       false
|     | |                          Tags                  []
|     | `-9 Body 3                 Name                  Body 3
|     |                            IsAssignedColor       False
|     |                            Color                 java.awt.Color[r=128,g=128,b=128]
|     |                            Opacity               1.0
|     |                            Display Resolution    VERY_COARSE
|     |                            Multi-Instanced       false
|     |                            Tags                  []
|     +-2 Features
|     | +-1 XY                     Error Message
|     | |                          Origin                [0.0, 0.0, 0.0] m
```

```
 |    | |                         X-Axis                          [1.0, 0.0, 0.0]
 |    | |                         Y-Axis                          [0.0, 1.0, 0.0]
 |    | |                         Tags                            []
 |    | +-2  YZ                   Error Message
 |    | |                         Origin                          [0.0, 0.0, 0.0] m
 |    | |                         X-Axis                          [0.0, 1.0, 0.0]
 |    | |                         Y-Axis                          [0.0, 0.0, 1.0]
 |    | |                         Tags                            []
 |    | +-3  ZX                   Error Message
 |    | |                         Origin                          [0.0, 0.0, 0.0] m
 |    | |                         X-Axis                          [0.0, 0.0, 1.0]
 |    | |                         Y-Axis                          [1.0, 0.0, 0.0]
 |    | |                         Tags                            []
 |    | +-4  Global Origin        Error Message
 |    | |                         Position                        [0.0, 0.0, 0.0]
 |    | |                         Tags                            []
 |    | +-5  Lab Coordinate       Error Message
System
 |    | |                         Origin                          [0.0, 0.0, 0.0]
 |    | |                         X-axis Direction                [1.0, 0.0, 0.0]
 |    | |                         Y-axis Direction                [0.0, 1.0, 0.0]
 |    | |                         Tags                            []
 |    | `-6  ImportCad 1          Error Message
 |    |                           Exchange Settings               {NX=1, STEP=1, SE=1, CGR=1, SW=1,
                                                                  RHINO=1, JT=1, ACIS=1, IFC=1,
                                                                  CATIAV5=1, IGES=1, CATIAV4=1,
                                                                  3DXML=1, CREO=1, INV=1}
 |    |                           File Name                       C:\Users\amrut\OneDrive - The
                                                                  University of Manchester\Thesis\Shell
                                                                  Eco Car 2015.x_t
 |    |                           Import Multi-instanced Bodies   false
 |    |                           Tags                            []
 |    +-3  Design Filters
 |    `-4  Design Parameters
 +-8  Operations
 |  +-1  Subtract                 Link Output Part Name           false
 |  | |                           Input Parts                     [Body 1, Domain]
 |  | |                           Target Part                     [Domain]
 |  | |                           Perform CAD Boolean             true
 |  | |                           Output Parts                    Manager
 |  | |                           Tags                            []
 |  | +-1  Tessellation Options   Tessellation Options            Medium
 |  | `-2  Tolerance Type         Tolerance Type                  Precise
 |  |                             Tolerance                       1.0E-5 m
 |  `-2  Automated Mesh           Per-Part Meshing                false
 |    |                           Mesher Execution Mode           Serial
 |    |                           Input Parts                     [Subtract]
 |    |                           Perform Local Meshing           false
 |    |                           Preserve Surface Perimeters     None
 |    |                           Verbose Output                  false
 |    |                           Tags                            []
 |    +-1  Meshers
 |    | +-1  Surface Remesher     Meshing Method                  Triangle
 |    | |                         Perform Curvature Refinement    true
 |    | |                         Perform Proximity Refinement    true
 |    | |                         Perform Compatibility Refinement false
 |    | |                         Create Aligned Meshes           true
 |    | |                         Minimum Face Quality            0.05
 |    | |                         Field Function based Refinement []
 |    | +-2  Automatic Surface    Connected Surface Count Limit   None
Repair
 |    | |                         Connected Surface Size Limit(s) None
```

```
|     | |                          Minimum Face Quality                  0.05
|     | +-3 Geometric Sensitivity
|     | +-4 Tetrahedral Mesher      Field Function based Refinement       []
|     | |                          Run Post Mesh Optimizer               true
|     | |                          Conformal To Existing Mesh            false
|     | `-5 Prism Layer Mesher      Stretching Function                   Geometric Progression
|     |                            Distribution Mode                     Stretch Factor
|     +-2 Default Controls
|     | +-1 Base Size               Base Size                             0.2 m
|     | |                          Base Size                             0.2 m
|     | +-2 CAD Projection          Project to CAD                        true
|     | +-3 Target Surface Size     Size Type                             Relative to base
|     | |                          Percentage of Base                    100.0
|     | |                          Absolute Size                         0.2 m
|     | +-4 Minimum Surface Size    Size Type                             Relative to base
|     | |                          Percentage of Base                    10.0
|     | |                          Absolute Size                         0.02 m
|     | +-5 Surface Curvature       Enable Curvature Deviation Distance   false
|     | |                          # Pts/circle                          36.0
|     | |                          Max # Pts/circle                      200.0
|     | |                          Curvature Deviation Distance          0.01 m
|     | +-6 Surface Proximity       Search Floor                          0.0 m
|     | |                          # Points in gap                       2.0
|     | |                          Enable Search Ceiling                 false
|     | |                          Search Ceiling                        1.0E10 m
|     | |                          Search Direction                      INSIDE
|     | +-7 Surface Growth Rate     Surface Growth Rate                   DEFAULT
|     | |                          User Specified Value                  1.3
|     | +-8 Auto-Repair Minimum     Minimum Proximity                     0.01
Proximity
|     | +-9 Geometric Sensitivity   Parameters                            []
|     | |                          Relative Step Size                    0.001
|     | |                          Minimum Step Size                     0.001
|     | +-10 Generate Standard      Generate Only Standard Prismatic Cells false
Prismatic Cells
|     | +-11 Volume Growth Rate     Volume Growth Rate                    1.2
|     | +-12 Maximum Tet Size       Size Type                             Relative to base
|     | |                          Percentage of Base                    10000.0
|     | |                          Absolute Size                         20.0 m
|     | +-13 Core Mesh              Optimization Cycles                   1
Optimization
|     | |                          Quality Threshold                     0.4
|     | `-14 Post Mesh              Optimize Cell Topology                false
Optimization
|     |                            Optimize Boundary Vertices            false
|     `-3 Custom Controls
|       +-1 Surface Control         Enable Control                        true
|       | |                        Controls Display Mode                 All
|       | |                        Part Surfaces                         [Subtract.Body 1.Default]
|       | |                        Apply Only to Contacting Area         false
|       | |                        Tags                                  []
|       | +-1 Controls
|       | | +-1 Target Surface      Target Surface Size                   Custom
Size
|       | | +-2 Minimum Surface     Minimum Surface Size                  Parent
Size
|       | | +-3 Surface             Curvature                             Parent
Curvature
|       | | +-4 Surface Proximity   Proximity                             Parent
|       | | +-5 Edge Proximity      Proximity                             Parent
|       | | +-6 Surface Growth      Surface Growth Rate                   Parent
Rate
```

| | | +-7 Surface Remeshing	Surface Remeshing	Parent
| | | +-8 Meshing Method	Meshing Method	Parent
| | | +-9 Prism Layers	Prism Layers	Custom
| | | | `-1 Customize	Customize Number of Layers	true
| | | |	Customize Total Thickness	false
| | | |	Customize Distribution	false
| | | |	Override Boundary Defaults	false
| | | |	Customize Limit Aspect Ratio	false
| | | |	Customize Minimum Thickness Percentage	false
| | | |	Customize Boundary March Angle	false
| | | |	Customize Concave Angle Limit	false
| | | |	Customize Convex Angle Limit	false
| | | `-10 Wake Refinement	Specify wake refinement options	true
| | `-2 Values		
| | +-1 Target Surface Size	Size Type	Relative to base
| | |	Percentage of Base	10.0
| | |	Absolute Size	0.02 m
| | +-2 Custom Prism Values		
| | | `-1 Number of Prism Layers	Number of Prism Layers	24
| | `-3 Wake Refinement	Size type	Relative to base
| | |	Distance	10.0 m
| | |	Coordinate System	Laboratory
| | |	Direction	[1.0, 0.0, 0.0]
| | |	Spread Angle	0.0 radian
| | +-1 Isotropic Size	Size Type	Relative to base
| | |	Percentage of Base	25.0
| | |	Absolute Size	0.05 m
| | `-2 Wake Refinement	Growth Rate	1.3
| `-2 **Volumetric Control**	Enable Control	true
| |	Controls Display Mode	All
| |	Parts	[Block]
| |	Tags	[]
| +-1 Controls		
| | +-1 Surface Remesher	Customize Size	true
| | +-2 Prism Layer Mesher	Customize Number of Layers	true
| | |	Customize Total Thickness	false
| | |	Customize Stretching	false
| | |	Customize Limit Aspect Ratio	false
| | |	Customize Minimum Thickness Percentage	false
| | |	Customize Boundary March Angle	false
| | |	Customize Concave Angle Limit	false
| | |	Customize Convex Angle Limit	false
| | `-3 Tetrahedral Mesher	Customize Tetrahedral Mesher	true
| `-2 Values		
| +-1 Custom Size	Size Type	Relative to base
| |	Percentage of Base	15.0
| |	Absolute Size	0.03 m
| `-2 Custom Prism Values		
| `-1 Number of Prism Layers	Number of Prism Layers	24

<table>
<tr><td>+-9 Descriptions</td><td>Number of Children</td><td>5</td></tr>
<tr><td>| +-1 Root</td><td>Described Parts</td><td>[Body 1, Domain, Subtract, Block, Subtract 2]</td></tr>
<tr><td>| +-2 Automated Mesh.Remesh</td><td>Described Parts</td><td>[Subtract]</td></tr>
<tr><td>| |</td><td>Faces</td><td>78396</td></tr>
<tr><td>| |</td><td>Vertices</td><td>39217</td></tr>
<tr><td>| +-3 Input Surface</td><td></td><td></td></tr>
<tr><td>| +-4 Latest Surface</td><td>Described Parts</td><td>[Body 1, Domain, Subtract, Block, Subtract 2]</td></tr>
<tr><td>| |</td><td>Faces</td><td>115374</td></tr>
<tr><td>| |</td><td>Vertices</td><td>57731</td></tr>
<tr><td>| |</td><td>Preview Mesh Operation Parts</td><td>false</td></tr>
<tr><td>| `-5 Latest Surface/Volume</td><td>Described Parts</td><td>[Body 1, Domain, Subtract, Block, Subtract 2]</td></tr>
<tr><td>+-10 Coordinate Systems</td><td></td><td></td></tr>
<tr><td>| `-1 Laboratory</td><td>Tags</td><td>[]</td></tr>
<tr><td>| `-1 Local Coordinate Systems</td><td></td><td></td></tr>
<tr><td>+-11 Parameterizations</td><td></td><td></td></tr>
<tr><td>+-12 Tables</td><td>Tables</td><td>1</td></tr>
<tr><td>| `-1 Accumulated Force Table</td><td>Extracted</td><td>[]</td></tr>
<tr><td>| |</td><td>Parts</td><td>[]</td></tr>
<tr><td>| |</td><td>Representation</td><td>Volume Mesh</td></tr>
<tr><td>| |</td><td>Tags</td><td>[]</td></tr>
<tr><td>| +-1 Force Histogram</td><td>Bins</td><td>10</td></tr>
<tr><td>| |</td><td>Force Option</td><td>PRESSURE_AND_SHEAR</td></tr>
<tr><td>| |</td><td>Coordinate System</td><td>Laboratory</td></tr>
<tr><td>| |</td><td>Bin Direction</td><td>[1.0, 0.0, 0.0]</td></tr>
<tr><td>| |</td><td>Force Direction</td><td>[1.0, 0.0, 0.0]</td></tr>
<tr><td>| |</td><td>Profile Direction</td><td>[0.0, 1.0, 0.0]</td></tr>
<tr><td>| |</td><td>Normalization Mode</td><td>NONE</td></tr>
<tr><td>| |</td><td>Reference Pressure</td><td>0.0 Pa</td></tr>
<tr><td>| |</td><td>Range Mode</td><td>AUTO</td></tr>
<tr><td>| |</td><td>Range</td><td>[0.0, 100.0]</td></tr>
<tr><td>| |</td><td>Spacing</td><td>10.0</td></tr>
<tr><td>| |</td><td>Tags</td><td>[]</td></tr>
<tr><td>| `-2 Update</td><td>Enabled</td><td>true</td></tr>
<tr><td>|</td><td>Auto Extract</td><td>false</td></tr>
<tr><td>|</td><td>Trigger</td><td>None</td></tr>
<tr><td>|</td><td>Save To File</td><td>false</td></tr>
<tr><td>|</td><td>Output Directory</td><td></td></tr>
<tr><td>|</td><td>Base Filename Append Tag</td><td>table</td></tr>
<tr><td>+-13 Units</td><td>Preferred System</td><td>Systeme International</td></tr>
<tr><td>+-14 Custom Trees</td><td>Initial Tree View</td><td>[]</td></tr>
<tr><td>+-15 Volume Shapes</td><td></td><td></td></tr>
<tr><td>+-16 Idealizations</td><td>Region selection priority</td><td>[Region Symmetry Idealization 1]</td></tr>
<tr><td>| |</td><td>Idealizations</td><td>1</td></tr>
<tr><td>| `-1 Region Symmetry Idealization 1</td><td>Regions</td><td>[Region]</td></tr>
<tr><td>|</td><td>Generating Region</td><td>[Region]</td></tr>
<tr><td>|</td><td>Electric Fields</td><td>[]</td></tr>
<tr><td>|</td><td>Magnetic Fields</td><td>[]</td></tr>
<tr><td>|</td><td>Tags</td><td>[]</td></tr>
<tr><td>+-17 Color Palettes</td><td></td><td></td></tr>
<tr><td>| +-1 High Contrast Color Palette</td><td>Number of Colors</td><td>28</td></tr>
<tr><td>| |</td><td>Swatches</td><td>[java.awt.Color[r=255,g=0,b=0],
java.awt.Color[r=0,g=255,b=0],
java.awt.Color[r=0,g=0,b=255],
java.awt.Color[r=255,g=255,b=0],
java.awt.Color[r=255,g=128,b=0],
java.awt.Color[r=160,g=32,b=240],</td></tr>
</table>

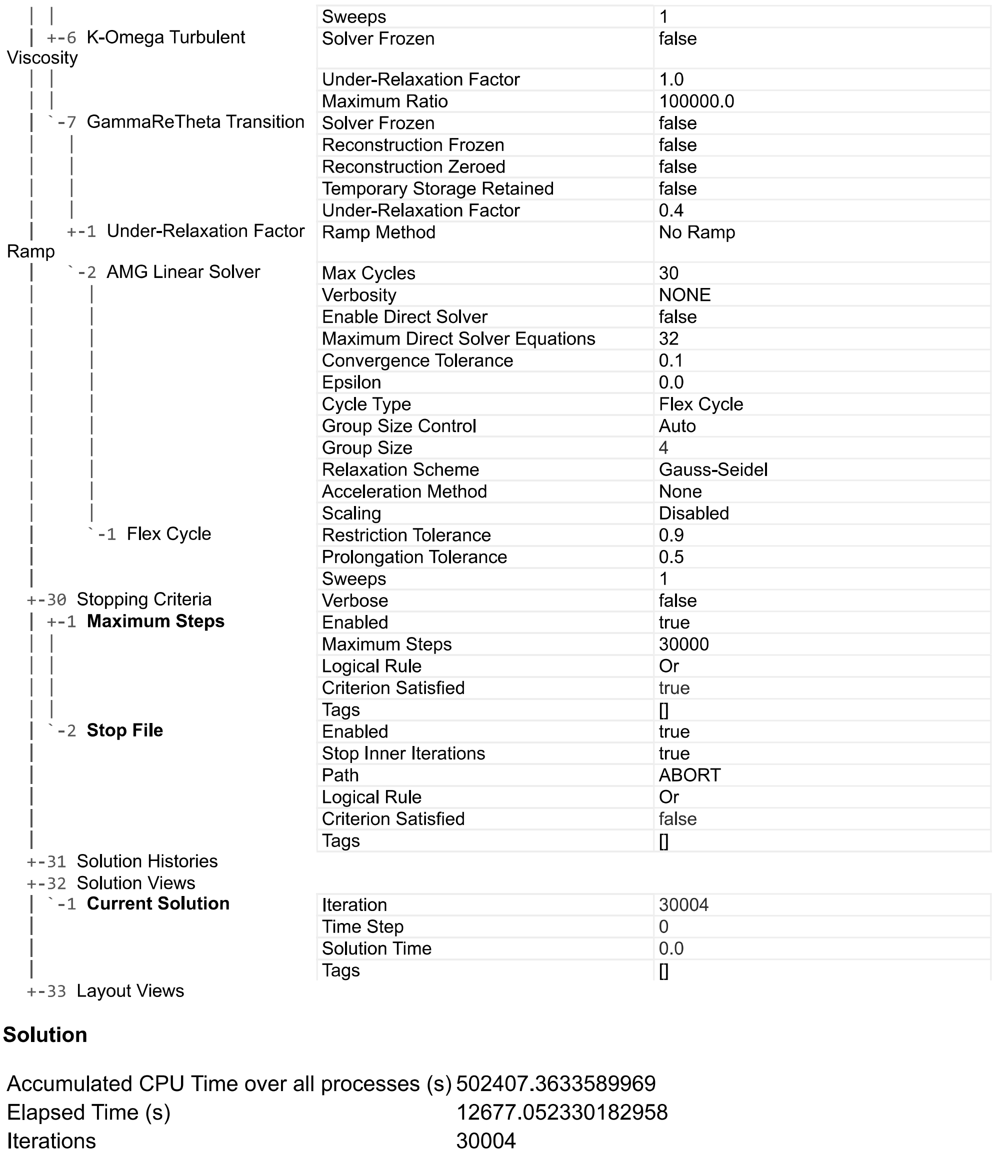
<table>
<tr><th></th><th></th><th></th></tr>
<tr><td></td><td></td><td>java.awt.Color[r=255,g=255,b=255],
java.awt.Color[r=255,g=192,b=203],
java.awt.Color[r=189,g=252,b=201],
java.awt.Color[r=175,g=238,b=238],
java.awt.Color[r=240,g=230,b=140],
java.awt.Color[r=255,g=228,b=181],
java.awt.Color[r=221,g=160,b=221],
java.awt.Color[r=192,g=192,b=192],
java.awt.Color[r=219,g=112,b=147],
java.awt.Color[r=0,g=201,b=87],
java.awt.Color[r=0,g=255,b=255],
java.awt.Color[r=255,g=227,b=3],
java.awt.Color[r=255,g=176,b=15],
java.awt.Color[r=186,g=85,b=211],
java.awt.Color[r=128,g=128,b=105],
java.awt.Color[r=176,g=48,b=96],
java.awt.Color[r=34,g=139,b=34],
java.awt.Color[r=95,g=158,b=160],
java.awt.Color[r=255,g=215,b=0],
java.awt.Color[r=255,g=97,b=3],
java.awt.Color[r=143,g=94,b=153],
java.awt.Color[r=115,g=74,b=18]]</td></tr>
<tr><td>| |</td><td>Tags</td><td>[]</td></tr>
<tr><td>| +-2 Legacy Plot Color Palette</td><td>Number of Colors</td><td>13</td></tr>
<tr><td>| |</td><td>Swatches</td><td>[java.awt.Color[r=255,g=0,b=0],
java.awt.Color[r=0,g=255,b=0],
java.awt.Color[r=0,g=0,b=255],
java.awt.Color[r=255,g=200,b=0],
java.awt.Color[r=0,g=255,b=255],
java.awt.Color[r=0,g=0,b=0],
java.awt.Color[r=178,g=0,b=0],
java.awt.Color[r=0,g=178,b=0],
java.awt.Color[r=0,g=0,b=178],
java.awt.Color[r=8,g=46,b=84],
java.awt.Color[r=178,g=140,b=0],
java.awt.Color[r=160,g=32,b=240],
java.awt.Color[r=92,g=36,b=110]]</td></tr>
<tr><td>| |</td><td>Tags</td><td>[]</td></tr>
<tr><td>| `-3 Siemens Color Palette</td><td>Number of Colors</td><td>30</td></tr>
<tr><td>|</td><td>Swatches</td><td>[java.awt.Color[r=15,g=120,b=155],
java.awt.Color[r=229,g=192,b=76],
java.awt.Color[r=219,g=83,b=90],
java.awt.Color[r=96,g=106,b=117],
java.awt.Color[r=226,g=137,b=77],
java.awt.Color[r=127,g=70,b=100],
java.awt.Color[r=104,g=153,b=98],
java.awt.Color[r=232,g=170,b=184],
java.awt.Color[r=165,g=143,b=111],
java.awt.Color[r=127,g=178,b=172],
java.awt.Color[r=0,g=85,b=125],
java.awt.Color[r=168,g=133,b=45],
java.awt.Color[r=153,g=49,b=64],
java.awt.Color[r=50,g=57,b=63],
java.awt.Color[r=170,g=97,b=48],
java.awt.Color[r=81,g=45,b=67],
java.awt.Color[r=71,g=102,b=66],
java.awt.Color[r=173,g=104,b=127],
java.awt.Color[r=112,g=94,b=75],
java.awt.Color[r=77,g=124,b=115],
java.awt.Color[r=70,g=170,b=193],
java.awt.Color[r=234,g=208,b=150],
java.awt.Color[r=234,g=172,b=185],
java.awt.Color[r=170,g=180,b=188],
java.awt.Color[r=239,g=194,b=163],
java.awt.Color[r=186,g=131,b=165],</td></tr>
</table>

```
                                                             java.awt.Color[r=165,g=198,b=158],
                                                             java.awt.Color[r=239,g=203,b=217],
                                                             java.awt.Color[r=204,g=188,b=168],
                                                             java.awt.Color[r=180,g=214,b=208]]
|                              Tags                          []
+-18 Data Set Functions        Data Directory                function_data
|                              In-core surface FFTs          false
+-19 User Code
+-20 Data Focus
+-21 Data Mappers              Use Legacy Volume Mapper      false
+-22 Motions                   Visual Motion Transforms      []
| |                            Motion Preview Time           0.0 s
| |                            Motion Preview Parts          []
| `-1 Stationary               Tags                          []
+-23 Reference Frames
| `-1 Lab Reference Frame      Tags                          []
+-24 Screenplays
+-25 Derived Parts             Derived Parts                 4
| +-1 Isosurface               Parts                         [Body 1]
| | |                          Scalar Field                  Absolute Total Pressure
| | |                          Mode                          ISOVALUE_SINGLE
| | |                          Tags                          []
| | `-1 Value                  Isovalue                      10135.0 Pa
| +-2 Plane Section            Origin                        [0.0, 0.0, 0.0] m,m,m
| | |                          Coordinate System             Laboratory
| | |                          Normal                        [0.0, 0.0, 1.0] m,m,m
| | |                          Parts                         []
| | |                          Section Mode                  SINGLE
| | |                          Displayed Index               -1
| | |                          Tags                          []
| | `-1 Single section         Offset                        0.0 m
| +-3 Plane Section 2          Origin                        [0.0, 0.0, 0.0] m,m,m
| | |                          Coordinate System             Laboratory
| | |                          Normal                        [0.0, 0.0, 1.0] m,m,m
| | |                          Parts                         [Region]
| | |                          Section Mode                  SINGLE
| | |                          Displayed Index               -1
| | |                          Tags                          []
| | `-1 Single section         Offset                        0.0 m
| `-4 Streamline               Parts                         [Region]
|     |                        Seed Type                     PART
|     |                        Rotation Scale                1.0
|     |                        Vector Field                  Velocity
|     |                        Integration Solver            RK2
|     |                        Wall Treatment                ON
|     |                        Tags                          []
|     +-1 Source Seed          Seed Parts                    [Region: Subtract.Domain.Inlet]
|     |                        On Ratio                      1
|     |                        Randomize                     false
|     |                        N Grid Points                 [20, 40]
|     `-2 2nd Order Integrator Initial Integration Step      0.5
|                              Maximum Propagation           34.56682546180041
|                              Max Steps                     2000
|                              Integration Direction         FORWARD
+-26 Summaries
+-27 Monitors                  Monitors                      13
| |                            Monitors To Print             [Continuity, X-momentum, Y-momentum,
                                                             Z-momentum, Tke, Sdr, Intermittency,
                                                             ReTheta_t, Drag Force Monitor, Cd
                                                             Frontal Monitor, Cd Planar Monitor]
| |                            Output Direction              Horizontal
```

```
| |                            Heading Print Frequency    10
| +-1 Cd Frontal Monitor       Report                     [Cd Frontal]
| | |                          Enabled                    true
| | |                          Value Type                 Total Value
| | |                          Trigger                    Iteration
| | |                          Normalization Option       Off
| | |                          Maximum Plot Samples       5000
| | |                          Tags                       []
| | `-1 Iteration Frequency    Iteration Frequency        1
| |                            Start Iteration            0
| |                            Enable Stop                false
| |                            Stop Iteration             0
| +-2 Cd Planar Monitor        Report                     [Cd Planar]
| | |                          Enabled                    true
| | |                          Value Type                 Total Value
| | |                          Trigger                    Iteration
| | |                          Normalization Option       Off
| | |                          Maximum Plot Samples       5000
| | |                          Tags                       []
| | `-1 Iteration Frequency    Iteration Frequency        1
| |                            Start Iteration            0
| |                            Enable Stop                false
| |                            Stop Iteration             0
| +-3 Drag Force Monitor       Report                     [Drag Force]
| | |                          Enabled                    true
| | |                          Value Type                 Total Value
| | |                          Trigger                    Iteration
| | |                          Normalization Option       Off
| | |                          Maximum Plot Samples       5000
| | |                          Tags                       []
| | `-1 Iteration Frequency    Iteration Frequency        1
| |                            Start Iteration            0
| |                            Enable Stop                false
| |                            Stop Iteration             0
| +-4 Iteration                Maximum Plot Samples       5000
| |                            Tags                       []
| +-5 Physical Time            Maximum Plot Samples       5000
| |                            Tags                       []
+-28 Reports                   Reports                    5
| +-1 Cd Frontal               Units
| |                            Coordinate System          Laboratory
| |                            Direction                  [1.0, 0.0, 0.0]
| |                            Force Option               Pressure + Shear
| |                            Reference Pressure         0.0 Pa
| |                            Pressure Option            SP_UC
| |                            Number of Bands            0
| |                            Reference Density          1.18415 kg/m^3
| |                            Reference Velocity         16.0 m/s
| |                            Reference Area             ${Frontal Area}
| |                            Parts                      [Region: Subtract.Body 1.Default]
| |                            Representation             Volume Mesh
| |                            Smooth Values              false
| |                            Account for Idealization   false
| |                            Tags                       []
| +-2 Cd Planar                Units
| |                            Coordinate System          Laboratory
| |                            Direction                  [1.0, 0.0, 0.0]
| |                            Force Option               Pressure + Shear
| |                            Reference Pressure         0.0 Pa
| |                            Pressure Option            SP_UC
| |                            Number of Bands            0
```

```
| |                             Reference Density                      1.18415 kg/m^3
| |                             Reference Velocity                     16.0 m/s
| |                             Reference Area                         ${Planar Area}
| |                             Parts                                  [Region: Subtract.Body 1.Default]
| |                             Representation                         Volume Mesh
| |                             Smooth Values                          false
| |                             Account for Idealization               false
| |                             Tags                                   []
| +-3 Drag Force                Units                                  N
| |                             Coordinate System                      Laboratory
| |                             Direction                              [1.0, 0.0, 0.0]
| |                             Force Option                           Pressure + Shear
| |                             Reference Pressure                     0.0 Pa
| |                             Pressure Option                        SP_UC
| |                             Number of Bands                        0
| |                             Parts                                  [Region: Subtract.Body 1.Default]
| |                             Representation                         Volume Mesh
| |                             Smooth Values                          false
| |                             Account for Idealization               false
| |                             Tags                                   []
| +-4 Frontal Area              Units                                  m^2
| |                             View Up                                [0.0, 1.0, 0.0] m,m,m
| |                             Normal                                 [1.0, 0.0, 0.0] m,m,m
| |                             Coordinate System                      Laboratory
| |                             Parts                                  [Region: Subtract.Body 1.Default]
| |                             Representation                         Volume Mesh
| |                             Tags                                   []
| `-5 Planar Area               Units                                  m^2
|                               View Up                                [0.0, 0.0, 1.0] m,m,m
|                               Normal                                 [0.0, 1.0, 0.0] m,m,m
|                               Coordinate System                      Laboratory
|                               Parts                                  [Body 1, Region: Subtract.Body
                                                                       1.Default]
|                               Representation                         Geometry
|                               Tags                                   []
+-29 Solvers                    Cache Solver Settings                  true
| +-1 Steady
| | `-1 Stopping Criteria       Verbose                                false
| +-2 Partitioning              Solver Frozen                          false
| |                             Partitioning Method                    Per-Continuum
| +-3 Wall Distance             Solver Frozen                          false
| |                             Verbosity                              0
| |                             Minimum Tree Size Threshold            500000
| +-4 Segregated Flow           Pressure Gradient Inverse Distance     true
                                Weighting
| | |                           Implicit Scheme                        SIMPLE
| | |                           Freeze Flow                            false
| | |                           Reconstruction Frozen                  false
| | |                           Reconstruction Zeroed                  false
| | |                           Enable Enhanced Stability Treatment    false
| | |                           Temporary Storage Retained             true
| | |                           Pressure Corrections: Bad Cell         0.8
                                Minimum Scaling
| | |                           Pressure Corrections: Acceptable Cell  0.001
                                Volume Change
| | |                           Velocity Corrections: Maximum          20.0 m/s
                                Unlimited Velocity
| | |                           Velocity Corrections: Acceptable       0.15
                                Velocity Increase Rate [<1]
| | |                           Continuity Initialization              false
| | +-1 Velocity                Under-Relaxation Factor                0.2
```

```
| | | |                                  Dynamic Local Under-Relaxation          false
| | | +-1 Under-Relaxation Factor Ramp   Ramp Method                             No Ramp
| | | `-2 AMG Linear Solver              Max Cycles                              30
| | |     |                              Verbosity                               NONE
| | |     |                              Enable Direct Solver                    false
| | |     |                              Maximum Direct Solver Equations         32
| | |     |                              Convergence Tolerance                   0.1
| | |     |                              Epsilon                                 0.0
| | |     |                              Cycle Type                              Flex Cycle
| | |     |                              Group Size Control                      Auto
| | |     |                              Group Size                              4
| | |     |                              Relaxation Scheme                       Gauss-Seidel
| | |     |                              Acceleration Method                     None
| | |     |                              Scaling                                 Disabled
| | |     `-1 Flex Cycle                 Restriction Tolerance                   0.9
| | |                                    Prolongation Tolerance                  0.5
| | |                                    Sweeps                                  1
| | `-2 Pressure                         Number of Non-Orthogonality Correctors  0
| |     |                                Under-Relaxation Factor                 0.1
| |     |                                Pressure Reference Location             Automatic Selection
| |     +-1 Under-Relaxation Factor Ramp Ramp Method                             No Ramp
| |     `-2 AMG Linear Solver            Max Cycles                              30
| |         |                            Verbosity                               NONE
| |         |                            Enable Direct Solver                    false
| |         |                            Maximum Direct Solver Equations         32
| |         |                            Convergence Tolerance                   0.1
| |         |                            Epsilon                                 0.0
| |         |                            Cycle Type                              V Cycle
| |         |                            Group Size Control                      Auto
| |         |                            Group Size                              4
| |         |                            Relaxation Scheme                       Gauss-Seidel
| |         |                            Acceleration Method                     Conjugate Gradient
| |         |                            Scaling                                 Auto
| |         `-1 V Cycle                  Pre-Sweeps                              1
| |                                      Post-Sweeps                             1
| |                                      Max Levels                              50
| +-5 K-Omega Turbulence                 Solver Frozen                           false
| | |                                    Reconstruction Frozen                   false
| | |                                    Reconstruction Zeroed                   false
| | |                                    Temporary Storage Retained              true
| | |                                    Under-Relaxation Factor                 0.1
| | |                                    Boundary Layer Initialization           false
| | +-1 Under-Relaxation Factor Ramp     Ramp Method                             No Ramp
| | `-2 AMG Linear Solver                Max Cycles                              30
| |     |                                Verbosity                               NONE
| |     |                                Enable Direct Solver                    false
| |     |                                Maximum Direct Solver Equations         32
| |     |                                Convergence Tolerance                   10.0
| |     |                                Epsilon                                 0.0
| |     |                                Cycle Type                              Flex Cycle
| |     |                                Group Size Control                      Auto
| |     |                                Group Size                              4
| |     |                                Relaxation Scheme                       Gauss-Seidel
| |     |                                Acceleration Method                     None
| |     |                                Scaling                                 Disabled
| |     `-1 Flex Cycle                   Restriction Tolerance                   0.9
| |                                      Prolongation Tolerance                  0.5
```



```
| |                                  Sweeps                          1
| +-6 K-Omega Turbulent              Solver Frozen                   false
Viscosity
| |                                  Under-Relaxation Factor         1.0
| |                                  Maximum Ratio                   100000.0
| `-7 GammaReTheta Transition        Solver Frozen                   false
|   |                                Reconstruction Frozen           false
|   |                                Reconstruction Zeroed           false
|   |                                Temporary Storage Retained      false
|   |                                Under-Relaxation Factor         0.4
|   +-1 Under-Relaxation Factor      Ramp Method                     No Ramp
Ramp
|   `-2 AMG Linear Solver            Max Cycles                      30
|     |                              Verbosity                       NONE
|     |                              Enable Direct Solver            false
|     |                              Maximum Direct Solver Equations 32
|     |                              Convergence Tolerance           0.1
|     |                              Epsilon                         0.0
|     |                              Cycle Type                      Flex Cycle
|     |                              Group Size Control              Auto
|     |                              Group Size                      4
|     |                              Relaxation Scheme               Gauss-Seidel
|     |                              Acceleration Method             None
|     |                              Scaling                         Disabled
|     `-1 Flex Cycle                 Restriction Tolerance           0.9
|                                    Prolongation Tolerance          0.5
|                                    Sweeps                          1
+-30 Stopping Criteria               Verbose                         false
| +-1 Maximum Steps                  Enabled                         true
| |                                  Maximum Steps                   30000
| |                                  Logical Rule                    Or
| |                                  Criterion Satisfied             true
| |                                  Tags                            []
| `-2 Stop File                      Enabled                         true
|                                    Stop Inner Iterations           true
|                                    Path                            ABORT
|                                    Logical Rule                    Or
|                                    Criterion Satisfied             false
|                                    Tags                            []
+-31 Solution Histories
+-32 Solution Views
| `-1 Current Solution               Iteration                       30004
|                                    Time Step                       0
|                                    Solution Time                   0.0
|                                    Tags                            []
+-33 Layout Views
```

Solution

Accumulated CPU Time over all processes (s)	502407.3633589969
Elapsed Time (s)	12677.052330182958
Iterations	30004