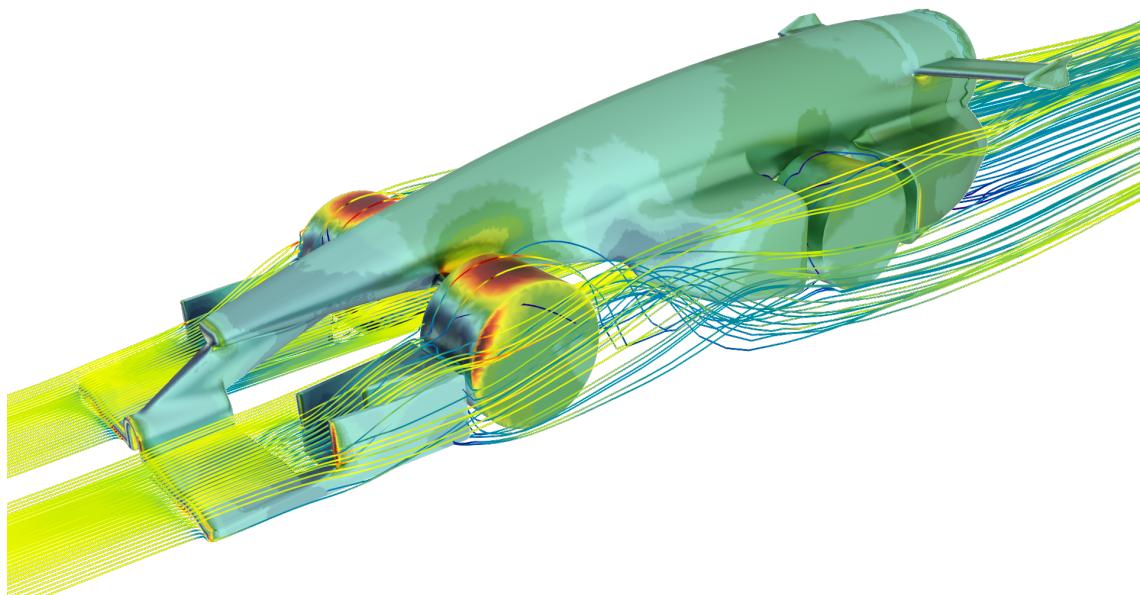


# A Beginner's Guide to Computational Fluid Dynamics

*Computational fluid dynamics theory without advanced mathematics*

Revision 2.1



Nathanael Jenkins

July 3, 2021

## Abstract

This document explains essential computational fluid dynamics (CFD) concepts without the use of advanced mathematics. Specifically, it is targeted at (and includes examples for) students competing in the F1inSchools competition, who are likely to be working at a GCSE level of mathematics. This is **not** a comprehensive guide to CFD, and some concepts are simplified where appropriate.

# Contents

<b>1 Preface</b>	<b>II</b>
<b>2 CFD Basics</b>	<b>1</b>
2.1 Fundamentals . . . . .	1
2.2 Meshing . . . . .	2
2.3 Simulations . . . . .	3
2.4 Results . . . . .	3
2.5 Software . . . . .	4
<b>3 CAD Geometry</b>	<b>4</b>
3.1 Modelling for CFD . . . . .	4
3.2 Exporting Geometry . . . . .	5
<b>4 Meshing</b>	<b>5</b>
<b>5 Boundary Conditions</b>	<b>7</b>
5.1 Advanced boundaries . . . . .	7
<b>6 Physics Models</b>	<b>8</b>
6.1 Turbulence Modelling . . . . .	8
<b>7 Running Simulations</b>	<b>9</b>
7.1 Convergence . . . . .	10
7.2 Problems . . . . .	11
<b>8 Results</b>	<b>13</b>
8.1 Graphic Results . . . . .	13
8.2 Numerical Results . . . . .	14
<b>9 Verification and Validation</b>	<b>14</b>
9.1 Mesh Convergence Studies . . . . .	14
9.2 Validation . . . . .	14
<b>10 Formula 1 in Schools</b>	<b>15</b>
10.1 Top Tips . . . . .	15
<b>References</b>	<b>III</b>
<b>A Appendices</b>	<b>IV</b>
A.1 Glossary . . . . .	IV
A.2 Additional Resources . . . . .	IV

# 1 Preface

Computational fluid dynamics is used to describe the broad range of computer simulation methods used to predict the behaviour of fluids (liquids and gases) in motion. It utilises degree-level mathematical concepts, which students often struggle to understand. This makes CFD inaccessible and confusing to the vast majority of younger engineers with an interest in aerodynamics. However, it is possible to begin to learn about how CFD works, and how to properly use it, without a degree.

For students competing in the F1inSchools competition, many of whom are working towards GCSEs or equivalent qualifications, the academic language and strange mathematical symbols easily cause confusion. This document is written to help such students competently use CFD to evaluate their designs, extracting valid and verifiable data, with an understanding of the fundamental concepts behind the simulations. This is not a 'how to' for any specific software, but an explanation of why CFD programs use certain approaches, and how they can be correctly applied.

The concepts presented in this document apply to most CFD software, although it is advisable to use a reasonably powerful computer with a CFD program that includes a GUI (graphical user interface), such as Autodesk CFD, ANSYS Fluent, Simscape, or StarCCM+.

## 2 CFD Basics

Perhaps the most important place to start a discussion on CFD is why it is used. Fluid dynamics is a broad term describing the interaction of liquids and gases with their surroundings, which can include solids or other fluids. Physical testing can provide the most true-to-life results when evaluating the behaviour of a fluid in or around a component, although it can be expensive and time-consuming, particularly for large projects such as aircraft or wind turbines. Sometimes, physical testing may not be possible at all.

Theoretical approaches are an alternative to physical testing. Using relatively simple maths, it is possible to predict the behaviour of a fluid around simple components. However, this approach often makes assumptions about the flow, and cannot be used on complicated geometries like the flow around an F1inSchools car.

This is where CFD comes in. Computers are excellent at making fast calculations without making mistakes, and we can use them to solve the complicated equations which govern fluid flow. But how do computers do it? To understand that, we first need to discuss some fundamentals of aerodynamics.

### 2.1 Fundamentals

Before running any CFD, it is important to understand exactly what is being investigated. CFD programs handle massive lists of variables, but which ones actually matter? A field is a variable which can have a different value at any point in space [1]. For example, the velocity of wind at the ground is much less than wind at the top of a large mountain. A vector quantity represents both a magnitude and direction, while a scalar can only represent a magnitude. For example, air moves at a certain speed, but also in a certain direction, so velocity is a vector. However, the air pressure acts the same in all directions at any point, so it is a scalar field. Some useful properties are defined in table 1 (this is **not** an exhaustive list). Additional definitions can be found in Fyrillas [2] and Leighton & Leighton [3].

Table 1: Physical quantities

Quantity	Type	Definition
Velocity	Vector field	Direction and rate of fluid flow at any point; the rate of change of position of a small fluid element
Density	Scalar field	The mass per unit volume of a fluid at any point [2]; usually constant for low-speed flows [4]
Pressure	Scalar field	The force per unit area exerted by a fluid on its surroundings due to molecular collisions [5]
Viscosity	Scalar field	A measure of a fluid's resistance to deformation at a certain rate; usually constant for low-speed flows [6]

For the purpose of simplicity, the majority of other quantities will not be discussed in detail in this document. A flow where density and viscosity are constant is called incompressible, usually defined as any flow under 30% of the speed of sound [7]. In air at sea level, this is equivalent to a velocity below  $100 \text{ ms}^{-1}$ . It is important to note that compressible flow is not the same as a gas which can be compressed. Flow around an F1inSchools car should be considered incompressible.

Having defined some quantities, it is now desirable to link them together using equations. The Navier-Stokes equations define fundamental relationships between quantities including velocity and pressure for any fluid volume [8]. They apply to air over the tiny wings of insects and even flow around the biggest skyscrapers and aircraft. However, the Navier-Stokes equations are incredibly complicated. In fact, they are so complicated that there is a \$1 million prize for anyone who proves that they can even be solved [9]! The Navier-Stokes equations represent the concepts of conservation of mass and conservation of momentum for incompressible flow, with conservation of energy added for compressible flow. To avoid complicated mathematical proofs, this document won't go into greater detail, although more information is available online [10] [[Click here](#)].

To compare between tests, aerodynamicists often use dimensionless coefficients. These are quantities defined in a way that means they have no units [11]. For example, dividing two velocities will give you a 'velocity ratio' which has no units of measurement. This may seem confusing, but the important result is that flow with the same dimensionless coefficients will behave in the same way. Dimensionless quantities make it easier to scale up results from a wind tunnel (because a full-size jumbo jet wouldn't fit in a

wind tunnel, for example). One of the most important quantities is the Reynolds number,  $Re$ , given in equation 1 [12].

$$Re = \frac{\rho U D}{\mu} \quad (1)$$

For incompressible flows,  $\rho$  ( $1.225 \text{ kg m}^{-3}$ ) and  $\mu$  ( $1.789 \times 10^{-5} \text{ N s m}^{-2}$ ) are constants, defined by the international standard atmosphere [13].  $U$  is the velocity ( $\approx 20 \text{ ms}^{-1}$  for an F1inSchools car), and  $D$  is a 'characteristic length'. This depends on the size of an assembly, so length or width would be appropriate choices for a car. To compare results between scale models, the Reynolds number and geometry must be identical for both tests. This is because the Reynolds number defines how flow interacts with a body, particularly in relation to turbulence. Testing a scale model of an F1 car in a wind tunnel will only provide accurate results if the Reynolds number for the test is the same as the Reynolds number the car would experience at full-scale [12]; for this reason, Formula 1 wind tunnels have to use higher wind speeds than the car would experience at full scale.

It is assumed that the reader has some understanding of laminar and turbulent flow. To demonstrate the effect of Reynolds number on flow behaviour, consider figure 1. It shows how the wake behind a cylinder varies with Reynolds number. A higher Reynolds number corresponds to more turbulent airflow.

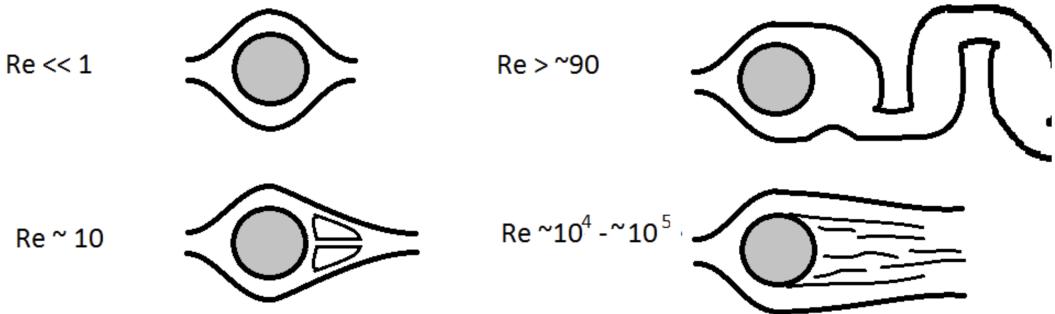


Figure 1: Effect of Reynolds number on flow around a cylinder [14]

## 2.2 Meshing

One teaspoon of air contains  $5 \times 10^{19}$  (or 50,000,000,000,000,000,000) molecules. Even the most advanced supercomputer in the world couldn't simulate this many particles individually [15][16]. Therefore, CFD estimates the behaviour of regions of the fluid based on versions of the Navier-Stokes equations. Rather than thinking of a fluid as a collection of many tiny molecules, CFD uses a 'continuum approach', which treats a fluid as one continuous material [17].

The space occupied by a fluid can be divided into many smaller 'pieces' of fluid called control volumes, each of which contain some amount of fluid. We can solve a version of the Navier-Stokes equations at the center of each control volume; for sufficiently small control volumes, this will accurately predict fluid behaviour [18]. An explanation of how we determine what counts as 'sufficiently small' is given in section 4.

Discretisation is the process of turning a CAD geometry into a collection of control volumes, also called a mesh. The geometry of a component is broken up into many smaller regions called cells. In a structured mesh, these cells are rectangular and grid-like. However, for complicated geometry such as an F1inSchools car, an unstructured mesh is more appropriate [19], consisting of polyhedra (i.e. shapes), usually with 4 (tetrahedra) or 6 (hexahedra) faces, shown in figure 2. The size and shape of the mesh has an effect on the simulation results, discussed in detail in section 4. A simulation with many smaller cells is likely to be more accurate than one with fewer large cells, although with increased computational cost (more cells mean more equations to solve) [19].

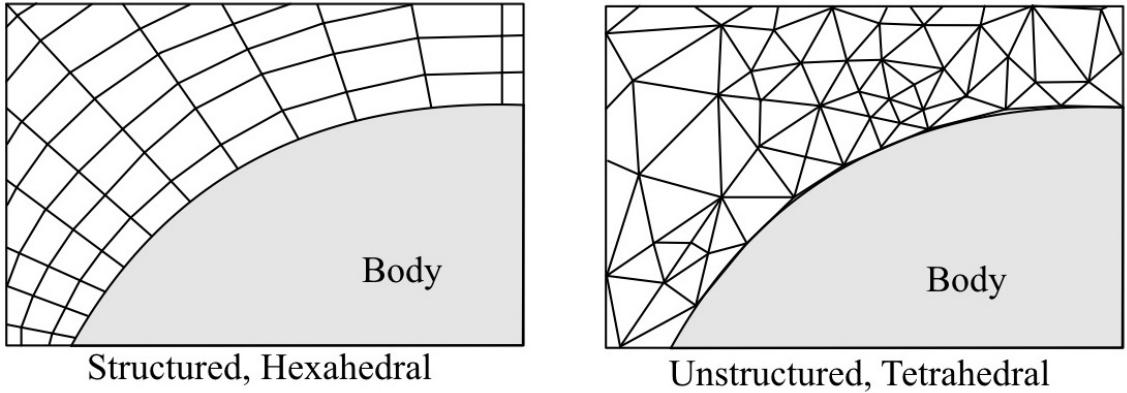


Figure 2: Example structured and unstructured meshes

### 2.3 Simulations

Once a problem has been meshed, so that our geometry is surrounded by control volumes, we need to define exactly which equations to solve. CFD programs often present a range of physics models to choose from, each of which will be valid for different types of flow. For example, an incompressible solver will only be valid for flows below 30% of the speed of sound [7]. The various models are discussed in section 6.

The physics models will either adapt or add to the equations to be solved in each cell. Because of the complexity of the Navier-Stokes equations, almost all CFD programs don't solve the actual equations. The Direct Numerical Simulation (DNS) method is the only approach which solves the Navier-Stokes equations [19], although it is almost exclusively used on supercomputers which have the capacity for such complex calculations.

Most CFD programs instead use a Reynolds-Averaged Navier-Stokes (RANS) approach, which takes average values of certain quantities in order to make calculations faster [19]. The mathematics behind this is complex, but it is most important to simply be aware that CFD is almost never directly solving the Navier-Stokes equations.

Regardless of the simplifications made, the flow equations cannot be solved analytically (i.e. using 'normal' maths). This means that a computer has to make a guess for the flow conditions in every control volume, using the equations to refine it. Repeating this process of guessing and refining hundreds or thousands of times will usually tend towards a stable and accurate set of values. However, repeating the calculations over every cell in a mesh for hundreds of iterations can require powerful computers, limiting the capability of computational fluid dynamics. This is discussed in more detail in section 7.

To summarise the CFD process; the goal is to find a relatively close solution to the Navier-Stokes equations for each control volume in a fluid domain. By discretising the fluid into control volumes, the number of calculations is reduced to a reasonable amount. Using various assumptions, the equations are simplified so that a solution can be found using iterative methods relatively quickly. Once the simulation has finished, fluid behaviour can be analysed and better understood than through practical or theoretical approaches.

### 2.4 Results

It is dangerous to simply throw a CAD model into CFD software and click 'run' without carefully generating a mesh and choosing the correct physics models. Even with careful setup, results are not guaranteed to be accurate. Proper validation or results is always necessary [20], using physical testing or theoretical predictions, as explained in section 9.

Often, issues with mesh generation can cause divergence, where the initial guess does not move closer to the true solution, but further away. This causes inaccurate results, which must not be used.

Students are often tempted to rely exclusively on graphical analysis of results, using streamlines or scalar plots of velocity and pressure. However, CFD is primarily intended for numerical analysis. The value of drag coefficients and similar quantities should not be overlooked in a CFD simulation. Many commercial CFD programs also come with useful analysis tools to help the user get the most out of their results; make sure to find out what your program can do so that you don't miss out.

## 2.5 Software

Hundreds, of CFD programs are available online. For a novice user, a program with a GUI is highly recommended. This allows the user to visualise their mesh and results 'live', with a clear workflow and helpful warnings of potential errors. Excellent software includes Autodesk CFD, ANSYS Fluent, Simscale, and StarCCM+. While many of these are license or subscription based, some companies do offer student licenses for reduced or no cost.

Regardless of the software, it is vital that the user fully understands its purpose, and any assumptions the solver makes. For example, some open-source software is only suitable for incompressible simulations. Make sure you understand what assumptions your software makes, and what the default settings are. You may want to change them.

## 3 CAD Geometry

Sections 3-9 explore specific aspects of CFD simulation in greater detail. These aim to answer any potential questions a new user may have, without requiring advanced mathematical understanding.

### 3.1 Modelling for CFD

Many engineers use CAD to design components and assemblies, and it can be tempting to import these directly into simulation software. For some programs, this may be possible and users should always see which file formats their software requires. However, usually, some additional modelling approaches are required to prepare geometry for simulation.

Flows around a body such as a car are called external (as opposed to internal flow inside a pipe or duct). The air around a car can extend miles away, although it would not make sense to try to model all the air in the world. Therefore, when modelling external flows, a 'virtual wind tunnel' is required, as illustrated in figure 3. The size and shape of the virtual wind tunnel can affect results.

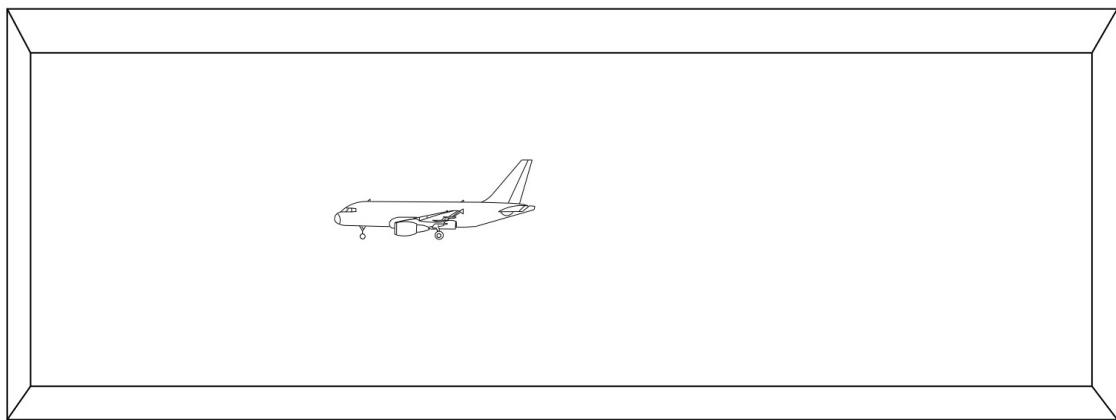


Figure 3: Virtual wind tunnel for CFD analysis of a passenger aircraft

A virtual wind tunnel should be significantly larger than the model being tested, although the actual size is likely to depend on mesh size and boundary conditions. Boundary conditions should be considered at this point (see section 5), particularly if the model is symmetric. By only exporting half of a symmetrical model, a symmetry boundary can be used to reduce simulation time. It is important to be sure that the model is totally symmetric, and divided perfectly at the symmetry plane when generating CAD models for such tests.

It is easiest to handle CAD files in CFD software if only one body is imported. While an F1inSchools car is likely to consist of separate front and rear wings, wheels, and chassis components, a simplification of the CAD file can make meshing much easier. In Fusion360, the 'combine' and 'boundary fill' tools can be particularly useful. Depending on the choice of CFD software, some geometry processing may be possible. Turning an assembly of many components into one body is surprisingly complex, so users should be prepared to spend some time on this.

Depending on the choice of CFD software, it may be necessary to generate a virtual wind tunnel in CAD, rather than generating it in the software. It is important that the virtual tunnel does not

contain any irregularities, such as small fillets, or irregular-shaped faces, as this can cause meshing and simulation issues. In Fusion360, the boundary fill tool can be useful for generating a virtual wind tunnel; the curvature and 'zebra' analysis tools may help to check for surface irregularities.

When generating a mesh from a CAD file, a minimum surface size will usually be specified. If the minimum size is close to the radius of any small components or fillets, this can result in a loss of geometry which will affect results. It is important to either remove or enlarge small features, or decrease the mesh size (see section 4). Any sharp corners on a CAD model are likely to remain sharp when meshed, although this should be monitored as some geometry can result in jagged edges when generating unstructured meshes.

If only one component needs testing, to reduce simulation time it would not be unreasonable to test only the component of interest, without including the whole assembly. However, the effect of nearby components should be considered. For example, if testing a wing for an aircraft, it would not make sense to place the wing in the middle of a virtual wind tunnel, with both ends free. Instead, the root of the wing should extend from one of the tunnel walls, as it would from the aircraft fuselage.

## 3.2 Exporting Geometry

Different programs take different geometry files. .iges are a common input type, which can be exported instantly from Fusion360. Note that some files such as .stl are mesh files, although these meshes are not generated in a way which is suitable for use in CFD, so should not be considered. Usually CFD software will specify a preferred file type.

Make sure when exporting geometry that only the desired body is exported, and always check the axis orientation when it is imported into CFD software. Although axis orientation won't directly lead to invalid results, inconsistent axis orientation can cause confusion when setting up a simulation. Most GUI-based software will show an axis indicator, with x, y, and z pointing in a certain direction. It is recommended that the x-direction be aligned with the flow direction, with the z-axis pointing vertically upwards.

## 4 Meshing

Meshing is an incredibly broad subject, with thousands of softwares and approaches. One important factor relating to accuracy is near-wall meshing. This is important in correctly modelling the aerodynamic boundary layer, although uses some slightly more advanced concepts which are discussed in Appendix A.3.

Generating a 3D mesh based on a given geometry is relatively easy in most CFD softwares. However, it is important to understand and modify any default settings which may need changing. Most importantly is the mesh size (or base size), which defines the approximate length of a cell edge. Choosing a mesh size is a complicated task, although a basic approach can be outlined for a novice user. Firstly, understanding of the correct order of magnitude will be helpful. For an F1inSchools car which is 210mm long, the mesh base size might be close to 1-10mm. By comparison, the mesh size for a 100m long airliner would be closer to 100-1000mm. Mesh size should take into account small features. For example, a wing with a leading edge radius of 1mm will need a mesh smaller than 1mm to accurately represent its curvature. At this point, it is necessary to consider localised mesh refinement.

Considering an F1inSchools car, with some fillets of radius 1mm. At the smallest fillets, the mesh size must be under 1mm. However, if every cell was that small, the mesh would be too large to compute. Instead, the mesh size for the entire flow could be set to 3mm, for example, using suitable refinement methods to decrease the mesh size near smaller components. The refinement options available will vary with different softwares, and some may apply refinement automatically. Areas where turbulence is expected to be prevalent may also need a smaller mesh size, to account for the less predictable patterns of turbulence. Figure 4 illustrates a mesh around a rocket, and the refinements applied to it.

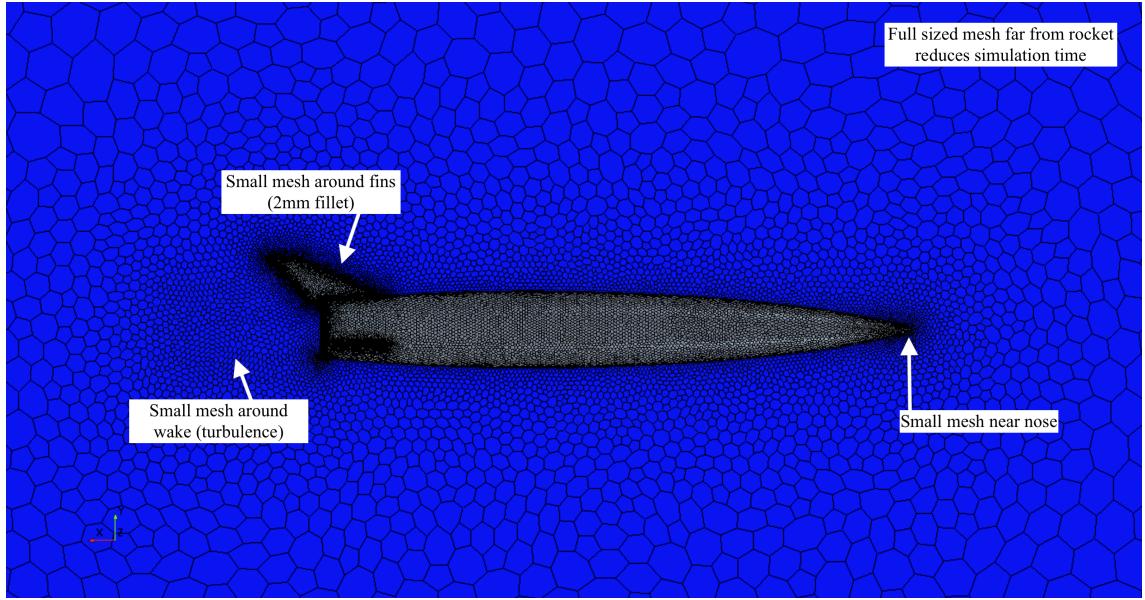
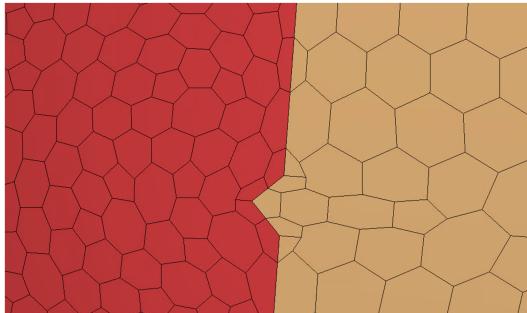
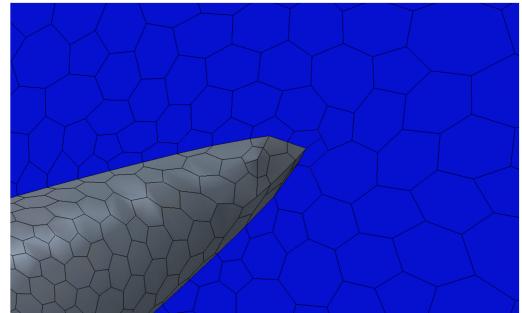


Figure 4: Polyhedral mesh around a model rocket

Ultimately, sizing a mesh is an art. Examining a generated mesh for irregular looking cells, poor geometry, or anything else that looks out of the ordinary will help to save time later, when simulations can be plagued with mesh-related issues. Examples of poor quality meshes are shown in figure 5. Essentially, mesh size should be chosen so that the geometry is preserved correctly and flow features will be easy to analyse, without becoming too fine that the simulation will take too long to run.



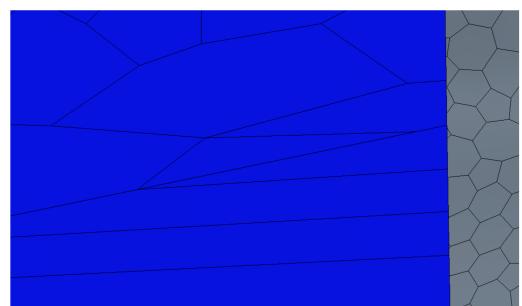
Irregular mesh (on two boundaries)  
*Caused divergence*



Mesh size too large  
*Failed to represent curvature, resulting in invalid results*



Mesh size too small  
*Needs a supercomputer to run*



Stretched, irregular cells  
*Caused divergence*

Figure 5: Signs of poor quality meshes

Another important decision is mesh type. Two of the most common meshes used for external flows, such as F1inSchools cars, are tetrahedral and polyhedral meshes. Tetrahedral meshes are made up of triangular-based pyramid shaped cells, whereas a polyhedral mesh cell has more sides and appears more rounded and almost 'football-like'. Generally, polyhedral meshes are recommended for complex geometry flows (such as an F1inSchools car), as they converge to accurate results faster [21]. As mentioned above, choice of mesh type is a complex field, so this should not be taken as the most appropriate mesh for every scenario, rather a good choice of mesh for a novice aerodynamicist.

Various other mesh controls are likely to be available depending on CFD software and mesh type. This document won't go into detail with all of the possible options, although good advice can often be found on CFD forums.

## 5 Boundary Conditions

It is important that boundary and initial conditions are correctly applied to a simulation. The most common boundary types are; wall (slip and no-slip), inlet (various kinds), outlet (various kinds), freestream, and symmetry.

**Wall** boundaries describe any solid surface. No-slip conditions should be applied to solid walls such as the car or aircraft being tested; this takes in to account surface friction [19]. When using a virtual wind tunnel with external flows, the tunnel walls can be specified using a wall with slip-conditions, provided flow is also incompressible, negating surface friction effects of the tunnel boundaries.

**Inlets** can be specified using a number of conditions [19]. A velocity-defined inlet is suitable for most incompressible flows, and is particularly useful for F1inSchools CFD since the approximate velocity of the car ( $20 \text{ ms}^{-1}$ ) is known. Other inlet types include mass flow, and freestream, although the mass flux and mach number are more difficult to calculate in the case of an F1inSchools car, so should be avoided.

**Outlets** can include mass flux outlets and pressure outlets [19]. As with the mass flux inlet, the rate of mass flow is difficult to calculate for most external flows, so such boundaries should be avoided. Instead, a pressure outlet set to the same pressure as the simulation can be used. Most CFD simulations use a 'relative' pressure approach, meaning that simulation pressure is 0 by default. It is important to understand the chosen pressure value at an outlet, and the choice should be verified when processing results. A correctly configured outlet will allow flow to leave the virtual wind tunnel as expected, without 'sucking' it out (an increase in velocity near the outlet would be evident), or 'pushing' it in (a decrease in velocity near the outlet would be evident).

**Freestream** boundaries are specifically designed to represent external flows far away from the body, for example at the virtual wind tunnel walls. However, most freestream boundaries require a mach number specification, which can cause issues if used alongside a velocity inlet. For this reason, it is not recommended that freestream boundary conditions are used for low-mach simulation.

**Symmetry** boundary conditions can be incredibly helpful in reducing simulation time for symmetric models [19].

### 5.1 Advanced boundaries

When modelling an F1inSchools car, for example, it may be useful to incorporate a moving 'road' and rolling wheels. Most CFD software will allow you to generate motion on a boundary. In the case of the road, it is useful to remember when generating the geometry for such simulations that the car will need to sit on the road as it would in real life, with the wheels touching the ground. Specifying a moving road reduces unwanted effects of treating the road as if it is moving with the car. An alternative to 'moving' the road surface is to make it a wall boundary with slip conditions, which would have a similar effect on incompressible simulations. Additionally, the rotation of wheels impacts flow behaviour, so specifying a rotational movement on the wheels to match the speed of the car is advisable. To calculate the angular velocity of the wheels, based on a car moving at  $20 \text{ ms}^{-1}$ , equation 2 should be used, where  $\omega$  represents angular velocity in **radians** per second, and  $r$  is the radius of the wheel in meters.

$$\omega = \frac{U}{r} \quad (2)$$

## 6 Physics Models

The term 'physics models' describes the collection of CFD methods which can be used to model various flows [19]. By making particular assumptions about flow, unnecessary calculations can be ignored, reducing simulation time. Most models are relatively easy to choose between, given understanding of the expected flow conditions and relatively simple aerodynamics. However, for flows where turbulence is present, including F1inSchools cars, a wide range of turbulence models are available.

Table 1 lists common physics models used by CFD code. Note that one simulation will require multiple physics models. Models appropriate for simulating an F1inSchools car are highlighted in bold.

Table 2: Fundamental physics models and their applications

Model	Assumptions	Applications
Steady	No variation in flow properties with time	Low speed, laminar flow (e.g. pipe flow, $Re < 1800$ )
<b>Unsteady</b>	Flow properties vary with time	Turbulent flows. e.g. wing in stall
Inviscid	High Reynolds number	High $Re$ , slender body. Usually only theoretical
Laminar	Laminar flow only	Low speed flow, e.g. pipe flow, $Re < 1800$
<b>Turbulent</b>	See section 6.2	See section 6.2
Compressible	Compressible flow	Mach number $> 0.3$ . e.g. fighter aircraft
<b>Incompressible</b>	Low speed, no compressibility effects	Mach number $< 0.3$ , e.g. average car
Coupled	Compressible flow	Mach number $> 0.3$ . e.g. fighter aircraft
<b>Segregated</b>	Low speed, no compressibility effects	Mach number $< 0.3$ , e.g. average car

### 6.1 Turbulence Modelling

Selecting physics models becomes complicated when turbulence is present. This is because, unlike laminar models where many assumptions can be made, turbulence is a random motion which is incredibly hard to predict. Without using a supercomputer with DNS code (section 2.3) to directly solve the Navier-Stokes equations, the most common modelling approach is the Reynolds-Averaged Navier-Stokes (RANS) model [19]. It should be noted that the Large Eddy Simulation (LES) approach fits between RANS and DNS in terms of accuracy, although also in terms of computational cost [22]. For a detailed mesh with many cells, as would be expected in an F1inSchools car, DNS and LES are likely to be too slow to be of use. Figure 6 illustrates this.

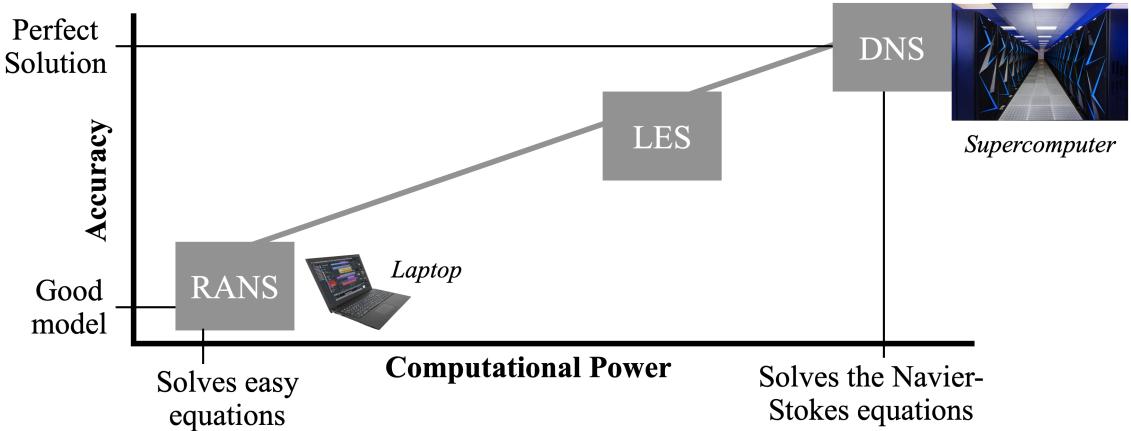


Figure 6: Overview of turbulence modelling approaches

The RANS method simplifies the Navier-Stokes equations by decomposing the seemingly random

turbulence motion into two quantities; a time-averaged quantity, and a fluctuating quantity. The mathematics behind this is highly advanced and there are many different RANS turbulence models [23]. It is vital that the correct RANS model is used when modelling turbulence flow, as each model has unique limitations. This is where an advanced understanding of the mathematics behind CFD is useful, although this document will outline the general characteristics of each approach.

1. The  $k - \epsilon$  model [23][24] calculates two quantities; turbulent kinetic energy,  $k$ , and turbulent dissipation rate,  $\epsilon$ . Those with a knowledge of GCSE physics may be aware that kinetic energy is equal to  $\frac{1}{2}mV^2$ , which means that the turbulent kinetic energy is related to the velocity of turbulent flow. This model is popular because of generally good convergence and relative computational efficiency, meaning that it gives accurate results fast. It is not suitable for simulating flow where separation occurs, or where streamlines are strongly curved, such as flow over a wing in stall. This is because near wall boundary conditions use functions which can be unreliable. To avoid problems caused by these unreliable functions some CFD programs offer 'automatic wall treatment', which adjusts its approach based on various mesh parameters.  $k - \epsilon$  is well-suited for simulating external flow over complex geometry, as expected in an F1inSchools car.
2. The  $k - \omega$  model [23][24] is similar to  $k - \epsilon$ , although solves for specific turbulent dissipation rate  $\omega$  instead of  $\epsilon$ . This is useful in simulations where the  $k - \epsilon$  model is inaccurate, such as internal flows, but it is less convergent. The initial guess is important when using the  $k - \omega$  approach, as it can easily cause divergence if it is incorrect. Often, the  $k - \epsilon$  approach is first used to generate a suitable initial guess. For the case of an F1inSchools car in external flow, the  $k - \omega$  method should not be used.
3. Low Reynolds number  $k - \epsilon$  approaches [24] take a similar approach to the standard  $k - \epsilon$  model, but don't use the unreliable near-wall boundary functions described above. This, however, means that a smaller mesh is needed to gain accurate results. This model can more accurately predict lift and drag forces, as well as flow separation over wings in stall. However, it requires a more advanced understanding of wall functions and should be approached with caution by novice users.
4.  $k - \omega$  SST [23][24] is another popular model. This is actually a combination of  $k - \epsilon$  and  $k - \omega$ , and the SST equations switch between the models for each cell in the mesh, depending on its position. This is essentially the 'best of both' of RANS turbulence models, making it suitable for mildly separated flows, including external flow. The  $k - \omega$  SST model may also be useful for simulating an F1inSchools car. [6]
5. Finally, the Spalart-Allmaras turbulence model [23][24] calculates kinetic eddy turbulent viscosity without calculating turbulent kinetic energy, which is of particular use in aerospace applications such as turbomachinery. It was developed at the same time as the  $k - \omega$  SST model, with the same aims. For most cases, the  $k - \omega$  SST model is preferred, so it would not be advisable to use this model.

It is important to note that all of these models have been refined and changed over the years. There are even more nuanced approaches within each of these models, which the reader is encouraged to explore. Choice of turbulence model cannot be generalised, and requires consideration of geometry, flow conditions, mesh properties, wall boundaries, and the desired result of the simulation. For a novice user, this can be baffling, and the only way to begin to understand these models is to read about them and try using them.

## 7 Running Simulations

Once a simulation is set up, with a mesh and the correct boundary conditions and all the appropriate physics models, it is ready to run a simulation. Finally!

Firstly, most CFD programs have a separate tool for initialisation. This is the term to describe making the initial guess at a solution. Usually, the velocity in each cell is set to the freestream flow velocity, and various other values are computed depending on the choice of physics model. Initialisation shouldn't take very long, less than a few minutes, and is a vital step in attaining valid results. When selecting physics models, some CFD programs may ask for the initial conditions to be specified. It is important to make sure that these are set to the correct values in order to avoid divergence.

Perhaps the easiest part of a CFD simulation is clicking 'run'. Once the simulation has started, it is always advisable to monitor its progress for the first few iterations. Most programs will output a

form of 'residuals' graph, and some will allow for live analysis of the solution, using velocity or pressure contours. This is a useful way to understand how a simulation is progressing, and is often a good indicator of convergence or divergence.

To reach a valid solution, a steady-state simulation will require at least a few hundred iterations, ideally more than 1,000 [25]. A transient (unsteady) simulation may require more iterations in total, as it solves the flow equations for many small time steps, each of which will require 20-50 inner iterations. Unsteady simulations rely on a more accurate initial guess, so it can sometimes be useful to use the result of a steady-state simulation as the starting point for an unsteady simulation. The unsteady simulation is likely to need less iterations per time step, although many time steps may be needed to properly understand the flow conditions.

## 7.1 Convergence

This document has discussed convergence and divergence with respect to various meshing and simulation models. But what do they actually mean? In CFD, good results should converge. This means that, as the guess is refined, it tends towards the correct solution. Divergence is the opposite of this, where the result instead moves away from the correct solution. The causes of convergence and divergence are complex and include meshing, physics models, and flow geometry.

It is also difficult to accurately define what counts as 'converged', and the exact requirements will vary with each simulation. First, let's consider how we can look for convergence. Section 7.2 will then discuss divergence and potential causes.

Many CFD programs generate a 'residuals' graph, showing lots of different quantities for each iteration. These residual values quantify (measure) the error in results, relative to the initial guess. Generally, it is accepted that a solution is likely to have converged if the residuals are no longer changing, and are all below  $10^{-5}$  [26], as demonstrated in the top two graphs in figure 7. However, this is not the only condition required for convergence at an accurate solution, so residuals should be treated with caution. It is recommended that a monitor is generated for any quantities of interest. For example, if investigating the drag coefficient on a body, some programs allow for it to be added to the residual plot. When the drag coefficient stops changing and becomes a horizontal line, the solution is likely to have converged. Figure 7 shows convergence plots for a steady state and a transient (unsteady) simulation in StarCCM+, as well as a diverging and an oscillating simulation. The steady state simulation spans 1000 iterations with one curve, but the transient simulation has many more 'spikes', each of which is under 400 iterations, adding up to more than 2000 iterations in total. Each of these spikes represents a time step. Unlike a steady simulation, which assumes the flow properties are independent of time, the transient simulation has to solve the flow conditions for many small time steps, in order to model the variation in flow with time. This means that it has to generate a new initial guess every time step, based on the previous results.

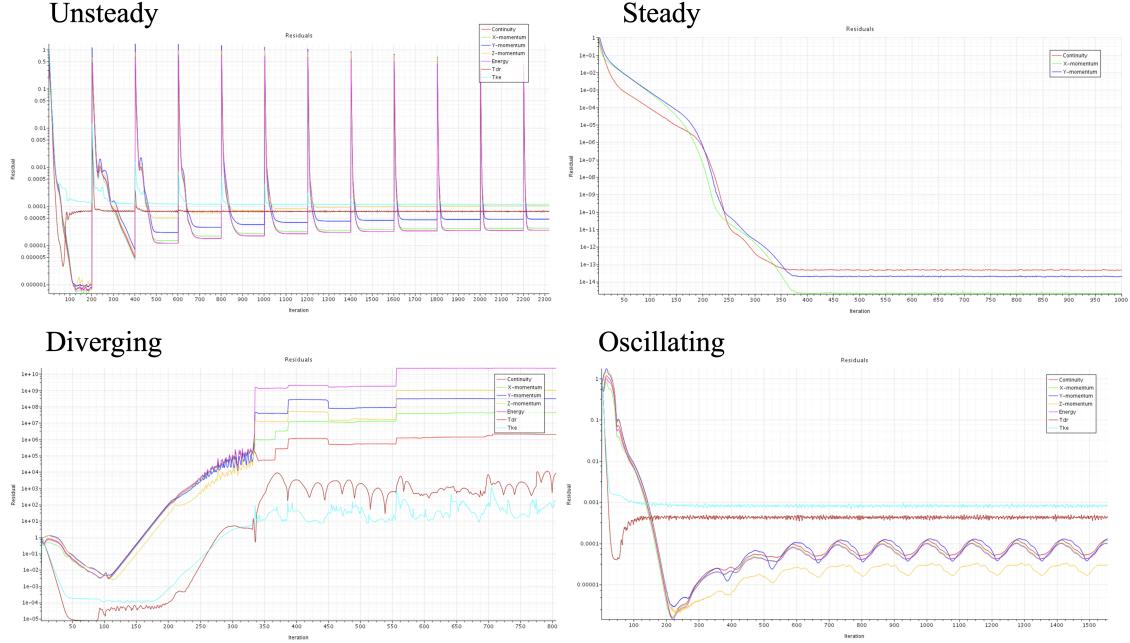


Figure 7: Residual vs iteration plots exhibiting different behaviours

Diverging residuals are easy to identify, and often increase so rapidly that the program stops and returns an error. One important behaviour of residuals is oscillation. For steady-state simulations where the flow properties should vary with time (a transient problem with steady state modelling), the solution is likely to oscillate rather than converge into a straight line. Generally, this can be fixed by changing to an unsteady model.

## 7.2 Problems

Problems usually lead to divergence in residuals. If residual values increase beyond 100, they will often spike suddenly towards infinity. This is a sure sign of an error in the simulation setup. This section will aim to help users diagnose and resolve common issues which can lead to divergence.

One of the most common causes of problems is a poor quality mesh. Figure 5 illustrates some common meshing errors. If these are not diagnosed at the meshing stage, they are likely to cause divergence. Often, this results in the velocity inside a few cells increasing significantly, sometimes by more than 1000 times the initial velocity. Using a scalar plot of velocity, or an isosurface (example in figure 8), it is relatively easy to find these extreme cells. It is then a matter of either changing the geometry, or adjusting the mesh settings to reshape that cell so that it is more regular, or smaller.

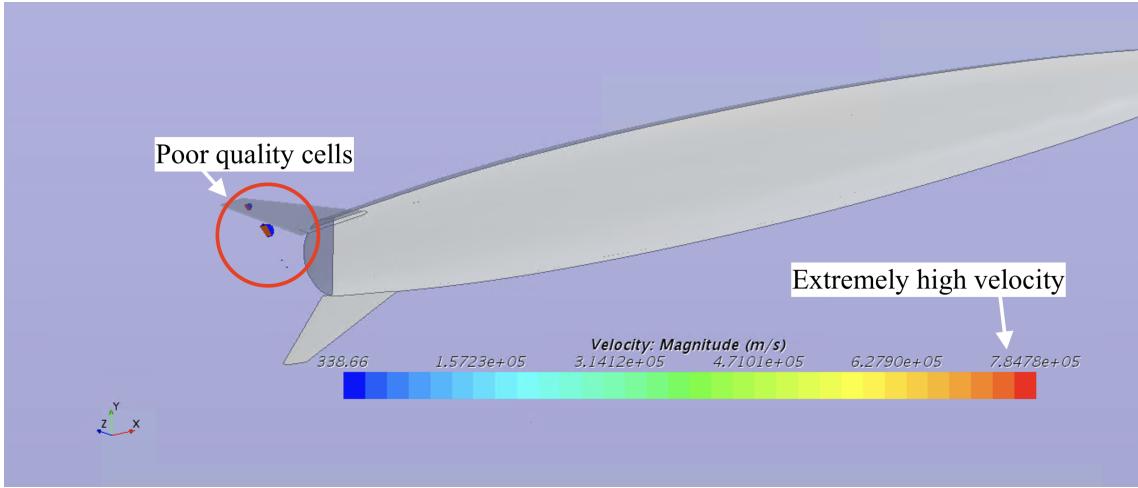


Figure 8: Poor quality cells in a diverged simulation of a rocket at  $500 \text{ ms}^{-1}$

Related to meshing, some divergence issues are a result of incorrect or poor boundary conditions. For example, an F1iSchools simulation using a virtual wind tunnel could select a pressure outlet to represent the tunnel walls (figure 9). From a modelling perspective, this may have worked, although it is not what the pressure outlet is designed for. This led to divergence, and strange flow behaviour near the tunnel walls. It was resolved by instead using a wall, with slip conditions. If divergence appears to occur in cells at boundary faces, it is worth checking that the correct boundary conditions have been applied. This is often helped by colour-coding in many CFD programs.

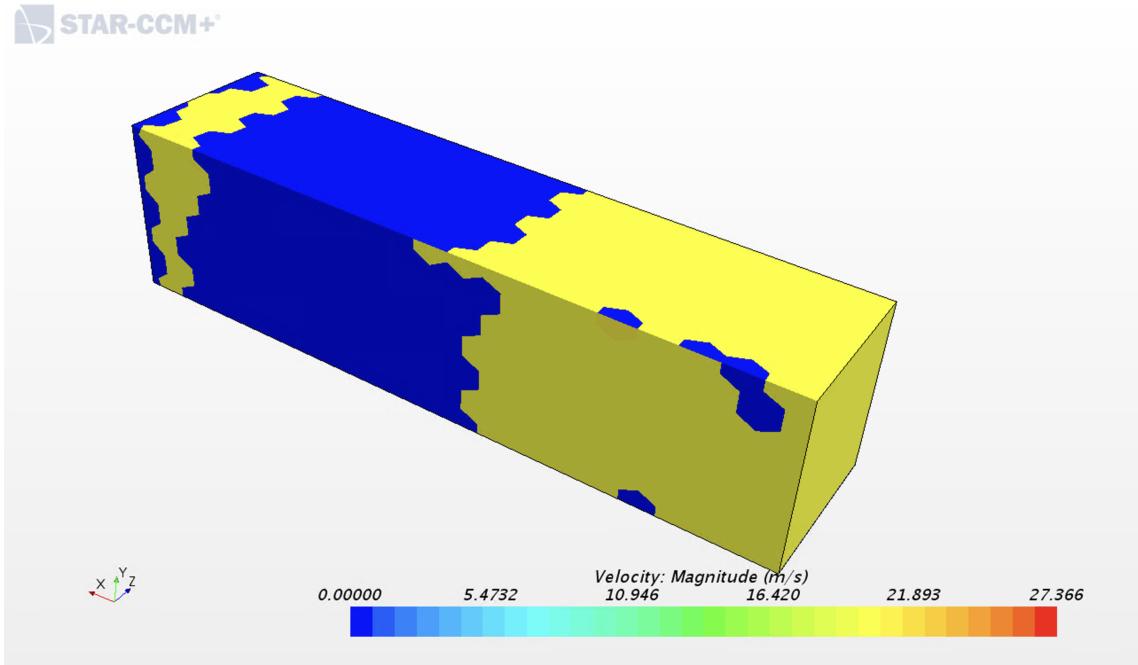


Figure 9: Poor quality boundary conditions causing divergence

A similar issue caused by poor boundary conditions can arise if the velocity inlet or pressure outlet is set incorrectly. For example, if the direction of the velocity at the inlet is specified incorrectly, it can cause, at best, inaccurate results, and at worst, complete divergence.

One other cause of divergence can be incorrect time modelling. If a simulation contains turbulence, but a steady state simulation is used, this can sometimes cause divergence due to the random nature of turbulence. Changing to a transient (unsteady) solver will resolve this issue.

Finally, some problems don't result in divergence. If the mesh size is too small, with too many cells, it is possible that the simulation will simply take too long to run. Even the world's most powerful

supercomputers have a limit, so don't expect to be able to simulate whatever you want. A reasonable guide would be that, if a simulation is taking more than 1 minutes per iteration, a smaller mesh is probably needed. At that rate, 1,000 iterations would take 16.7 hours.

## 8 Results

### 8.1 Graphic Results

Contours, streamlines and isosurfaces are a selection of useful visualisation methods for analysis of CFD results, shown in figure 10. Understanding what these represent is important; CFD isn't just about the pretty pictures!

First, let's consider the important quantities. Velocity is commonly used in analysis; the magnitude of velocity represents the speed of airflow in each cell. Velocity itself is a vector quantity, meaning that it also represents the direction of flow, which can be represented in vector plots using arrows or integral-convolution methods. Pressure is another important quantity to understand; the total pressure in each cell is equal to the sum of static pressure and dynamic pressure. Static pressure in a low speed simulation is likely to be relatively constant, so dynamic pressure is of the most importance as it is a major cause of drag. Note that pressure is a scalar quantity, meaning that it acts the same in all directions. Pressure and velocity are both given as fields in CFD, meaning that every cell will hold a unique pressure and velocity.

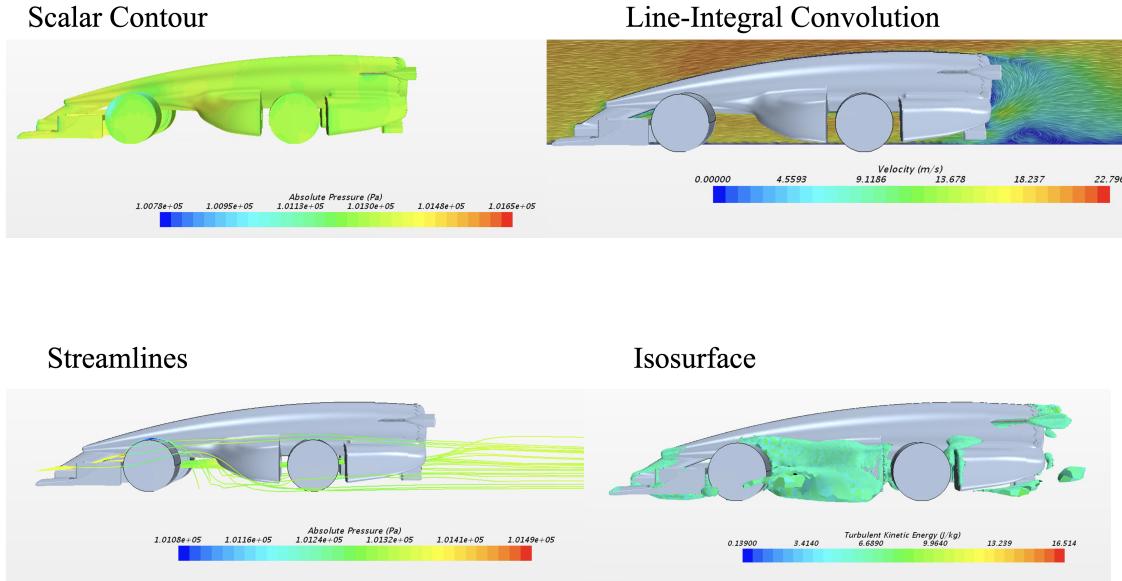


Figure 10: Common visualisation approaches

The first plot many CFD users will generate is a contour plot. Scalar variables can be shown on any surface or section plane using colour-coded scales. Often, pressure contours on the surface of a body will be used to identify regions of increased drag. The most important point to remember when comparing results is that everything must be consistent. If comparing between two contour plots, it is vital that the colour scale is the same in each. If a red colour represents a dynamic pressure of 100 Pa in the first simulation, it must also represent 100 Pa in the second simulation. Velocity contours using section views can also be interesting. It should be noted that generating a velocity contour on the surface or a wall (for example, an F1inSchools car), may only show a velocity of zero. This is because of the no-slip conditions at walls, meaning that the velocity at the surface is zero.

One way to improve on velocity contours is to use a vector approach instead. Rather than a simple colour-based scale, line-integral convolution is a clever-sounding name for adding many small 'streamline' style lines to the contour. This helps to visualise airflow, and can be particularly useful for visualising turbulence and eddies.

Streamlines are similar to line-integral convolution, although calculated slightly differently. They can be useful for analysing airflow over components. However, again, it is important to ensure total

consistency if comparing between test cases. The origin of streamlines will effect their behaviour and appearance.

One often feature often overlooked by novices is iso-surfaces. These generate a surface based on cells of constant value. For example, it can highlight all of the cells where the velocity is  $10 \text{ ms}^{-1}$ . Or more usefully, using the turbulent kinetic energy,  $k$ , iso-surfaces can help to visualise regions of more turbulent flow.

## 8.2 Numerical Results

While graphical methods often get much of the attention in CFD, numerical data is the true purpose for running simulations. The value of pressure, velocity, and drag data from CFD should not be underestimated. However, care should be taken to validate and properly interpret the results.

Different CFD programs will offer different methods to export numerical data, and many will include an option to generate line plots. This can be useful, for example, to plot the variation in pressure across the leading edge of a wing.

Aerodynamicists often focus on the coefficient of drag as a measure of the performance of a model. It is vital to make sure that the program is using the correct quantities to calculate coefficients. Most programs will have an option to set the reference area, reference length, velocity, and density for coefficients; this may be separate from the rest of the simulation setup. It is good to make sure that these are correct, although most importantly, they must be consistent between any simulations which are being compared.

As an example, for testing three changes to the front wing of an F1inSchools car, the following data could be useful; running three simulations of the car, at  $10$ ,  $15$ , and  $20 \text{ ms}^{-1}$ , results can be compared for a range of speeds. The drag coefficient of each test could be calculated, for the whole car, and for only the front wing. Comparing these should show which wings perform best. Additionally, a pressure contour over the wing surface could be useful, provided a consistent colour scale is used, and streamlines might help to show how the front wing causes air to interact with the front wheels.

# 9 Verification and Validation

CFD can converge and produce seemingly correct results, even if it is not accurate at all. This is one of the reasons for verification and validation methods. CFD users use statistics (disgusting, I know!) to verify that their results are accurate, or at the very least agree with experimental data [20].

## 9.1 Mesh Convergence Studies

Most meshes, particularly those which can be run on home computers and laptops, are large enough that they have a small impact on the solution. As the mesh size is decreased (so that number of cells increases), the accuracy of the solution will also increase. Beyond a certain mesh size, it will be so fine that the solution becomes 'mesh independent', meaning that however much the mesh is refined after that point, the solution remains the same. The purpose of a mesh convergence study is to investigate and quantify this [26].

Generally, it is advisable to start with a relatively large mesh. Running a simulation shouldn't take too long - no more than half an hour - and some numerical data may be saved. Then, the mesh size is decreased, and the simulation is run again. This is repeated, ideally until mesh independent solutions are found. This will be evident when the value of quantities such as velocity, pressure, or drag coefficient are the same for every simulation below a certain mesh size. Without access to high performance computers, complex flow problems may never reach a mesh-independent solution, but simply need to be refined as much as possible.

Some CFD programs will run automatic mesh refinement, although it is important to make sure that this is properly understood and implemented to avoid potential inaccuracy.

## 9.2 Validation

Validation is a term used to describe the use of experimental or theoretical data to confirm a computational result. For example, using CFD to simulate airflow through a pipe, a relatively simple equation can be solved for the velocity at any point. This can be compared to the CFD results, to make sure that it is accurate. Even better would be a comparison with experimental data using a physical pipe.

For complex flows such as an F1inSchools car, this is understandably more difficult. Convergence of residuals is a part of validation, although additional methods are explained in NASA [20].

## 10 Formula 1 in Schools

This document has attempted to explain the most basic elements of CFD in a relatively simple way. But how should an F1inSchools team use CFD? That is up to you to work out. While it would be useful to give teams a testing procedure to follow, you won't learn anything following a list of instructions. It's up to the user to experiment, explore, and conduct individual research into CFD and how it should be applied to an F1inSchools car. A useful chapter on CFD 'best practise guidelines' can be found in 'Computational Fluid Dynamics for Engineers' [19].

### 10.1 Top Tips

A few CFD specific tips:

1. If comparing colour-based graphs or contours, use a consistent scale. There is always an option to manually adjust the scale in the postprocessing software.
2. Take screenshots and record as much as you can. Make sure to take a note of the physics models and meshing approaches. This helps with explaining the simulations to judges, as well as meaning that the same setup can be used for all tests.
3. Use common sense! If a number doesn't seem right, it probably isn't. The velocity around an F1inSchools car won't exceed  $50 \text{ ms}^{-1}$ , so it also shouldn't exceed  $50 \text{ ms}^{-1}$  in CFD.
4. Practise makes perfect. CFD is an incredibly broad and complicated topic. An understanding of the 'common sense' in point 3 is only possible through practise.
5. Don't just do CFD for the pretty pictures. Actually use the data to analyse the car. All F1inSchools teams have the same images - give the judges some real data.
6. Don't just use the default results settings. Most teams will have a pressure contour over their car, a velocity contour, and streamlines. Try using isosurfaces or integral-convolution contours to show some ingenuity.
7. Above all, make sure you understand it! If you're confused about a concept, do not guess. Try looking online, or searching CFD forums. If that doesn't work, ask someone for help. If you have looked, but can't find someone willing to help, you can ask your teacher to contact me through my website ([click here](#)).

General competition advice:

1. Make sure you understand the details of everything in the portfolio. If you include a word or concept, judges may question you on it.
2. Read the mark-scheme! They're included in the competition regulations for teams to use as guidance. If it's not in the mark-scheme, don't include it in the portfolio. If it is in the mark-scheme, make sure you do include it in the portfolio.
3. Get expert advice! No judges expect school students to know everything about aerodynamics and engineering. But you are encouraged to reach out to engineering companies for sponsorship and support. All it takes is an email, letter, or phone call.
4. Read the mark-scheme again! It's so important I wrote it twice. Teams often lose out on points because their presentations or portfolios just don't meet the mark-scheme criteria.
5. Most importantly, have fun! F1inSchools is about inspiring you to become engineers; the judges and everyone involved in the competition want to help you to learn about and enjoy engineering. Don't get caught up on what other teams have done, or whether you won a prize. The engineering you are doing here is degree-level. CFD is a topic most engineers won't even use until the third year of their university degree. So be proud of what you've achieved.

## References

- [1] Galbis A, Maestre M. 2012. *Vector Analysis Versus Vector Calculus*. Springer. p.Â 12. ISBNÂ 978-1-4614-2199-3.
- [2] Fyrillas M. *Basic Concepts/ Definitions of Fluid Mechanics*. Frederick Institute of Technology. Available from: [last accessed: 18.06.2021].
- [3] Leighton E, Leighton D. 2015. *A Glossary of Terms for Fluid Mechanics*. University of Notre Dame. Available from: [last accessed: 18.06.2021].
- [4] NASA. 2021. *Bernoulli's Equation*. Available from: [last accessed: 18.06.2021].
- [5] NASA. 2021. *Kinetic Theory of Gases*. Available from: [last accessed: 18.06.2021].
- [6] Wikipedia. 2021. *Viscosity*. Available from: [last accessed: 18.06.2021].
- [7] White F. *Fluid Mechanics*. 6th ed. Available from: [last accessed: 18.06.2021].
- [8] NASA. 2015. *Navier Stokes Equations*. Available from:  
<https://www.grc.nasa.gov/WWW/k-12/airplane/nseqs.html> [last accessed: 09.04.2021].
- [9] Clay Mathematics Institute. 2021. *Navier-Stokes Equation*. Available from:  
<http://www.claymath.org> [last accessed: 09.04.2021].
- [10] Anderson J.D. 2009. *Governing Equations of Fluid Dynamics*. In Wendt, J. Computational Fluid Dynamics, An Introduction. XII, p.216. Available at:  
<https://www.springer.com/gb/book/9783540850557> [Last accessed: 09.04.2021].
- [11] International Standards Organisation. 2008. *International vocabulary of metrology. Basic and general concepts and associated terms*. JCGM:200. Available from: [last accessed: 18.06.2021].
- [12] NASA. 2014. *Reynolds Number*. Available from:  
<https://www.grc.nasa.gov/www/BGH/reynolds.html> [last accessed: 09.04.2021].
- [13] International Standards Organisation. 2007. ISO 2533:1975, Standard Atmosphere.
- [14] Teapeat. 2012. *Reynolds Behaviours*. Wikipedia. Available from:  
[https://commons.wikimedia.org/wiki/File:Reynolds\\_behaviors.png](https://commons.wikimedia.org/wiki/File:Reynolds_behaviors.png) [last accessed: 09.04.2021].
- [15] Burgess M. 2017. *China's supercomputer simulated the Universe with 10 trillion digital particles*. WIRED. Available from: [last accessed: 18.06.2021].
- [16] Markov I. 2014. *Limits on fundamental limits to computation*. Nature 512, 147-154. .
- [17] CFD Online. 2005. *Continuum*. Available from: [last accessed: 18.06.2021].
- [18] Ramezani A, Stipech G, Garcia I. 2016. *Introduction to Computational Fluid Dynamics by the Finite Volume Method*. Basque Center for Applied Mathematics. Available from: [last accessed: 18.06.2021].
- [19] Anderson B, et al. 2011. *Computational Fluid Dynamics for Engineers*. Cambridge University Press, ProQuest Ebook Central. Available from: [Last accessed: 09.04.2021].
- [20] Slater J. 2021. *Overview of CFD Verification and Validation*. NASA NPARC. Available from: [last accessed: 18.06.2021].
- [21] Symscape. 2013. *Polyhedral, Tetrahedral, and Hexahedral Mesh Comparison*. Available from: [last accessed: 18.06.2021].
- [22] Ideal Simulations. 2020. *Turbulence Models In CFD*. Available from: [last accessed: 18.06.2021].
- [23] NASA. 2021. *Turbulence Modelling Resource*. Available from: [last accessed: 18.06.2021].
- [24] Wimhurst A. 2019. *Fluid Mechanics 101*. YouTube. Available from: [last accessed: 18.06.2021].
- [25] Tran L. 2020. *CFD Online*. Available from: [last accessed: 18.06.2021].
- [26] Fogell N. 2020. *Introduction to Computational Fluid Dynamics*. Imperial College London. [last accessed: 18.06.2021].
- [27] Anderson J.D. 2011. *Fundamentals of Aerodynamics*. McGraw-Hill Series in Aeronautical and Aerospace Engineering, New York, 2011:5. ISBN: 978-1-259-01028-6.

## A Appendices

### A.1 Glossary

Fluid	A liquid or gas; substances which cannot resist shearing forces [27]
Aerodynamics	"The dynamics of gases, especially atmospheric interactions with moving objects." [27]
Control Volume	A region of space through which fluid can flow
Conserved	Quantities which cannot be created or destroyed must be conserved
Dimensionless	A quantity which does not have units
Reynolds Number	A dimensionless quantity which describes fluid behaviour [12]
Mesh	A set of points, joined by edges, to form 2D or 3D cells
Discretisation	The process of turning a continuous geometry into a mesh
Iterative	A method to solve equations which must be repeated many times for accuracy
Solver	A computer program which calculates certain flow variables in CFD code
CAD	Computer-Aided Design
Navier-Stokes	The general equations which describe fluid flow [23]
Incompressible	Flow in which density and viscosity do not change [27]
Steady	Flow in which no quantities change over time [27]
GUI	Graphical user interface
Mesh density	The number of cells in a given mesh volume
Boundary	A mesh surface, with certain flow conditions
Wall	A solid boundary, usually with 'no slip' conditions
No slip	Flow velocity is zero at a wall due to friction [27]
Inlet	A boundary through which flow enters the mesh
Outlet	A boundary through which flow exits the mesh
Freestream	External flow far away from a body
Physics model	See 'Solver'
Laminar	Flow which travels smoothly, in predictable ways [27]
Turbulent	Random flow motion involving fluctuations and mixing [27]
Coupled	A physics solver suitable for compressible flows
Supercomputer	A high performance computer with many thousands of processors
RANS	Reynolds-Averaged turbulence models
Initialisation	The process of making an 'initial guess' in CFD
Transient	Unsteady simulations (opposite of 'steady')
Converged	Describes simulations which have reached a valid answer
Residuals	A measure of the potential error in a quantity
Contour	A colour-based plot illustrating variation in a physical quantity across a surface
Scalar	A magnitude of a quantity
Vector	A quantity with both magnitude and direction
Streamline	A line representing the path of a fluid particle in a flow
Isosurface	A surface representing constant values of a certain quantity
Mesh dependent	A simulation where the result varies with mesh density
Validation	The process of verifying a CFD result using experimental or analytical data

### A.2 Additional Resources

Computational Fluid Dynamics for Engineers, Anderson et. al. [19]

Fundamentals of Aerodynamics, Anderson. [27]

Governing Equations of Fluid Dynamics, Anderson. [10]

Fluid Mechanics 101, Wimhurst. [24]