Individual Report – I2

ECMM102

Title: Virtual Wind Tunnel

Date of submission: 01/05/2014

Student Name: Daniel Nima

Student number: 600017181

Candidate number: 033197

Supervisor: Dr Gavin Tabor

**Abstract**

Mesh generation is a critically important step in producing stable, fast converging and accurate solutions to fluid dynamic problems using computational simulations. However, it is a process that is often misunderstood, and its significance underestimated, due to the apparent ease of automatic mesh generation in many modern software packages. The process becomes much more involved when component designs are given to the analysis department with little regard for what their work involves, due to incomplete or overly complex descriptions of the geometries. This group project was focussed upon creating an integrated design process for automotive prototypes to challenge the dated industry standard. It would assimilate the production of complete CAD designs with accurate computational modelling of the same design, improved and validated by experimental testing of 3D printed prototypes in a purpose built wind tunnel. This individual project concentrated on the role of CAD import and mesh generation. It performed an in-depth background study into the field of mesh generation and identified common meshing strategies, sources of difficulty in meshing and standards that must be adhered to in order to produce a high quality mesh. In tandem with the Design sub-group, simulations were used to refine a basic automotive geometry towards a more realistic design with a lower drag coefficient. A study was then undertaken to evaluate the effects of different meshing schemes on mesh quality and solution accuracy, using research on an Ahmed body for validation of results. The patch-independent, automatic tetrahedral technique was found to be the fastest, most robust and most accurate method compared to the other three schemes. It was then applied to the final car design, producing results within 1.4% of the experimental value. It is therefore recommended that this meshing process be incorporated into modern automotive aerodynamic simulation procedures.

Keywords: CFD, Meshing, Optimisation, Fluent, Automotive Design.

**Table of contents**

[1. Introduction 1](#_Toc386645529)

[1.1 Background 1](#_Toc386645530)

[1.2 Project Aims 1](#_Toc386645531)

[1.3 Report Structure 2](#_Toc386645532)

[2. Background and Theoretical Understanding 2](#_Toc386645533)

[2.1 Introduction 2](#_Toc386645534)

[2.2 The Role of Mesh Creation in CFD 2](#_Toc386645535)

[2.3 The Components of a Mesh 4](#_Toc386645536)

[2.4 Types of Mesh 5](#_Toc386645537)

[2.4.1 Introduction 5](#_Toc386645538)

[2.4.2 Structured Mesh Varieties 5](#_Toc386645539)

[2.4.3 Unstructured Meshes 9](#_Toc386645540)

[2.4.4 Comparison of Structured and Unstructured Meshes 9](#_Toc386645541)

[2.4.5 Hybrid Meshes 10](#_Toc386645542)

[2.5 Grid Generation Techniques 11](#_Toc386645543)

[2.5.1 Introduction 11](#_Toc386645544)

[2.5.2 A Word on Data Structures 12](#_Toc386645545)

[2.5.4 Surface Mesh generation 12](#_Toc386645546)

[2.5.5 Input Geometry Considerations 14](#_Toc386645547)

[2.5.6 Volume Meshes – Structured Approaches 14](#_Toc386645548)

[2.5.6 Volume Meshes – Unstructured Approaches 16](#_Toc386645549)

[2.5.7 Conclusions of Grid Generation Techniques 21](#_Toc386645550)

[2.6 Mesh Quality and Evaluation 21](#_Toc386645551)

[2.6.1 Introduction 21](#_Toc386645552)

[2.6.2 A Brief Evaluation of the Importance of Mesh Quality 22](#_Toc386645553)

[2.6.3 Measures of Mesh Quality 22](#_Toc386645554)

[2.6.4 Conclusions 23](#_Toc386645555)

[2.7 Mesh Adaptation 24](#_Toc386645556)

[3. Methodology 24](#_Toc386645557)

[3.1 Design Process Methodology 24](#_Toc386645558)

[3.2 Meshing Research Strategy 25](#_Toc386645559)

[3.2 Health and Safety 25](#_Toc386645560)

[4. Design and Experimentation 25](#_Toc386645561)

[4.1 Problem Definition 25](#_Toc386645562)

[4.2 Preliminary Work 25](#_Toc386645563)

[4.3 Meshing Comparison 28](#_Toc386645564)

[4.3.1 Introduction 28](#_Toc386645565)

[4.3.2 Simulation Details 29](#_Toc386645566)

[4.4 Concept Car Meshing and Simulation 32](#_Toc386645567)

[5. Project Analysis 35](#_Toc386645568)

[5.1 Analysis of Design Process 35](#_Toc386645569)

[5.2 Project Management Analysis 36](#_Toc386645570)

[5.2 Important Findings and Implications for Group Project 37](#_Toc386645571)

[6. Sustainability 38](#_Toc386645572)

[6.1 Economic Project Considerations 38](#_Toc386645573)

[6.2 Socio-Environmental Impacts 38](#_Toc386645574)

[6.3 Life-Cycle Analysis 39](#_Toc386645575)

[7. Conclusions and Future Work 39](#_Toc386645576)

[7.1 Project Conclusions 39](#_Toc386645577)

[7.2 Future Work 40](#_Toc386645578)

[References 41](#_Toc386645579)

[Appendix A – Further Information 47](#_Toc386645580)

[Appendix B - Additional Details from Background Sections 48](#_Toc386645581)

[Equations of Mesh Quality Metrics 48](#_Toc386645582)

[Aspect Ratio (Applies to All Cells) 48](#_Toc386645583)

[Equiangular Skewness (Applies to All Cells) 49](#_Toc386645584)

[Orthogonal Quality (Applies to All Cells) 49](#_Toc386645585)

[Fillet Study Information 50](#_Toc386645586)

[Appendix C – Health and safety risk assessment 60](#_Toc386645587)

[Appendix D – Project Management (Gantt Chart) 61](#_Toc386645588)

**Figures and Tabl**

[Figure 1: Depiction of the most commonly used element types in 2D and 3D simulations (10). 4](#_Toc386660068)

[Figure 2: Diagram representing common terms relating to computational grids, in this case a symmetrical hexahedral grid. 5](#_Toc386660069)

[Figure 3: Example 2D grid showing 'i' and 'j' grid edge orientations that do not conform to the (x,y) coordinates of the domain in which the grid lies. The shaded cell has coordinates (i = 4, j = 3) (2). 6](#_Toc386660070)

[Figure 4: Cartesian grid for simulation of flow over cylinder and stepwise approximation of mesh to smooth curve of geometry (16). Note wasted cells within cylinder’s interior that do not form part of simulation, waste computational resources and require removal prior to solving. 6](#_Toc386660071)

[Figure 5: Orthagonal curvilinear mesh fitted around an aerofoil profile in 2D (16). 7](#_Toc386660072)

[Figure 6: Body-fitted, non-orthogonal grid applied to cylinder flow example from Figure 4 (16). 7](#_Toc386660073)

[Figure 7: Example of a 3D multi-block grid, showing surface mesh topology, where interfaces are misaligned (15). 8](#_Toc386660074)

[Figure 8: An example of a body-fitted grid, surrounding a spaceship, which overlaps an existing background grid (15). 8](#_Toc386660075)

[Figure 9: Two dimensional examples of (a) tri and (b) quad unstructured meshes using the same interval spacing as Figure 3 (2). 9](#_Toc386660076)

[Figure 10: Annotated diagram of a hybrid grid formed on a simple car geometry, showing the transition from hex to tet elements on the symmetrical cut-plane. The diagram also gives a 3D representation of some of the surface hex elements (in red) (10). 11](#_Toc386660077)

[Figure 11: Overview of underlying (yet slightly dated) meshing algorithms used in variety of research and industrial fields (31). 11](#_Toc386660078)

[Figure 12: An example of the 2D parameterisation of a 3D surface and stretched triangle generation upon it that leads to generally equilateral mapping in 3D (31). 13](#_Toc386660079)

[Figure 13: Direct 3D meshing approach to triangulating a curved input geometry via an advancing front technique (31). 13](#_Toc386660080)

[Figure 14: Example of the mapping of a mesh from the computational to the physical domain, where TFI provides the singular framework that dictates the mesh structure by the function *X(ξ,η,ζ)* (44)*.* 15](#_Toc386660081)

[Figure 15: The shaded region displays the mapping of a simple cuboidal block to form one section of a (multi-block) cylinder, whereupon its internal mesh has been generated (47). 16](#_Toc386660082)

[Figure 16: 2D example of a Delaunay triangulation where no circumcircle of a set of three vertices that make up a triangle contains the vertices of any other triangle (31). 16](#_Toc386660083)

[Figure 17: 2D example of the filling of a domain using the advancing front technique. The front is said to be ‘empty’ when there remains no space to be meshed (26). 17](#_Toc386660084)

[Figure 18: Refinement around a small feature in a simple 2D quadtree example with hanging nodes clearly visible in refinement region (15). 18](#_Toc386660085)

[Figure 19: An example of quadtree refinement around an aerofoil profile with the ‘staircase’ boundary treatment employed (15). 18](#_Toc386660086)

[Figure 20: 2D example of IBM around a simplified car model, after 'solid' (interior) cell removal (54). 19](#_Toc386660087)

[Figure 21: A simple 2D case of an embedded geometry within a fluid domain where the cells in the Cartesian grid cut by the boundary are shaded. One scheme may entail joining the remaining segments of the darkest cell (*p* and *q*), whose centroid is within the solid, to the adjacent cut-cell (*l)* and interpolating between all its faces and those of the surrounding cells to determine flow variables (15). 20](#_Toc386660088)

[Figure 22: OpenFOAM simple 2D example of a staircase refinement followed by snapping of nearby cell vertices to model edges which results in polyhedral cells. A hexahedral extrusion has not yet been performed here (68). 20](#_Toc386660089)

[Figure 23: Change in drag coefficient (Cd) with increasing mesh density showing convergence around 400,000 elements. 27](#_Toc386660090)

[Figure 24: Decrease in drag coefficient associated with increasing model edge fillet radius where the curve displays diminishing returns on this decrease as the fillet radius grows proportionate to geometry dimensions. CAD model dimensions were: Length x Width x Height = 8 x 3.17 x 3.01cm. 28](#_Toc386660091)

[Figure 25: Graph showing mesh convergence on value of drag coefficient for four meshing schemes, compared with experimental. PD = Patch-dependent, PI = Patch-independent. 29](#_Toc386660092)

[Figure 26: Graph showing convergence times (seconds) against mesh density for four meshing schemes. 30](#_Toc386660093)

[Figure 27: Annotated wireframe view of wind tunnel and concept car set-up in Solidworks, all dimensions in mm. 32](#_Toc386660094)

[Figure 28: 2/8 of the results from the hole repair search in Design Modeler. All holes occurred around the wheels. 33](#_Toc386660095)

[Figure 29: Locations of cells with skewness around 0.999 in initially successful mesh of 600,000 cells. Locations of poor elements coincide with very sharp regions, such as the contact area between the wheel and the ground and the corners of the rear diffuser, as was expected. 33](#_Toc386660096)

[Figure 30: Zoom on ZY symmetry plane cut of mesh showing tetrahedral refinement around body and intricate features of concept car. 35](#_Toc386660097)

[Figure 31: Velocity pathline plot, on wind tunnel ZY symmetry plane, coloured by fluid velocity. Note correct inlet velocity of roughly 45m/s at test section inlet. 35](#_Toc386660098)

[Figure 32: H-type, structured, curvilinear grid around a 2D aerofoil (15). 48](#_Toc386660099)

[Figure 33: Illustration and general equation for calculating the aspect ratio of a cell, in this case a hex, between the lines joining its centroid to a node and the centroid to that of a cell face (74). 48](#_Toc386660100)

[Figure 34: Diagram depicting relevant vectors used to calculate orthogonal quality, in this case for a tetrahedral (74). 50](#_Toc386660101)

[Figure 35: Wireframe visualisation of unfilleted model within domain, showing symmetrical cut-plane of model on near-facing wall and buffer wall distances to minimise near-wall effects. 50](#_Toc386660102)

[Figure 36: Side-on view of 6mm filleted car model, within Fluent, demonstrating very thin face at right of model created by encroachment of filleted edges into originally flat side of car . 51](#_Toc386660103)

[Figure 37: Corresponding mesh for 6mm fillet CAD model, notice concentration of elements at front of car, caused by interaction between very thin model face and applied sizing there. 52](#_Toc386660104)

[Figure 38: Close-up of aforementioned area showing denser tetrahedral element distribution in area where face edges meet in more detail and around mouth of stilt. Due to the effective lengthening of the stilts where they meet faces at a higher point, the small element sizings here and the concentration in the thin-face section, the element count for the 4.5mm and 6mm models significantly higher than the rest. 52](#_Toc386660105)

[Figure 39: Velocity contours (m/s) on initial, bluff-body, model shown on a plane that does not bisect the stilts. 52](#_Toc386660106)

[Figure 40: Velocity contours (m/s) on 6mm fillet model shown on a plane that bisects the stilts. 53](#_Toc386660107)

[Figure 41: Intermediate PD tetrahedral mesh with symmetry plane isometric view, 1,003,046 elements, no wake regions specified. 53](#_Toc386660108)

[Figure 42: Local refinement using box to capture wake around Ahmed body using a PD tetrahderal mesh (2million elements). 53](#_Toc386660109)

[Figure 43: Velocity contours (m/s) on final, 4 million element mesh in PD tetrahedral convergence study. 54](#_Toc386660110)

[Figure 44: View of cells in hybrid grid with aspect ratio of roughly 12.5, note that the inflation region is comprised of wedges, not hexahedra, as hexahedral mesh generation is a more involved and less automatic process unless using cut-cell technique. 54](#_Toc386660111)

[Figure 45: Close-up of lower left corner of 4.2m element PD hybrid mesh, viewed from YZ plane. 54](#_Toc386660112)

[Figure 46: YX cut plane on same mesh, mid-way through Ahmed body. It shows regular inflation layers aligned with flow on Ahmed surface, floor surface and the tetrahedral volume between them. 55](#_Toc386660113)

[Figure 47: Close-up of refinement in lower right hand corner curved section of 3.27m element cut-cell mesh viewed from YZ plane. 55](#_Toc386660114)

[Figure 48: YX plane slice through 3.89m element PI tetrahedral mesh showing, whole elements and refinement towards Ahmed body. 55](#_Toc386660115)

[Figure 49: Velocity path line plot for final PI tetrahedral, 3.98m element, mesh. 56](#_Toc386660116)

[Figure 50: ZY symmetry plane cut showing whole-cell layout of final PI tetrahedral concept car mesh. 56](#_Toc386660117)

[Figure 51: Residual convergence plot and text dialogue, showing number of iterations and various variables including drag coefficient (Cd-1) for final concept car mesh. 57](#_Toc386660118)

[Figure 52: Scale of initial tetrahedral element layers on car surface. Height of cell centroid would be seen to be around 0.0002m, very close to that required to generate the optimum y+ value of 30, required for accurate implementation of the NEWF’s 57](#_Toc386660119)

Y

[Table 1: Common terminology used when referring to a computational mesh. 5](#_Toc386660120)

[Table 2: Weighted matrix analysis of four meshing schemes against various criteria. 31](#_Toc386660121)

[Table 3: Table listing leading providers or CFD meshing, and in some cases solving, software upon time of writing. 47](#_Toc386660122)

[Table 4: Table indicating qualitative meanings of varying skewness metric values, where “Excellent” means equilateral and “unacceptable” means degenerate (103). 49](#_Toc386660123)

[Table 5: Quality spectrum for orthogonal quality of mesh elements, where "Unacceptable" means degenerate and "Excellent" means perfectly orthogonal (103). 49](#_Toc386660124)

[Table 6: Mesh creation specifics and simulation statistics for basic, unfilleted geometry mesh convergence study. 51](#_Toc386660125)

[Table 7: Result of fillet study, displaying changing Cd and other metrics for each fillet radius value. 51](#_Toc386660126)

[Table 8: Table showing various quantities recorded during initial PD tetrahedral mesh convergence study on Ahmed body for later comparison. 58](#_Toc386660127)

[Table 9: Table showing various quantities recorded during PD hybrid mesh convergence study on Ahmed body for later comparison. 58](#_Toc386660128)

[Table 10: Table showing various quantities recorded during PI cut-cell mesh convergence study on Ahmed body for later comparison. 59](#_Toc386660129)

[Table 11: Table showing various quantities recorded during PI tetrahedral mesh convergence study on Ahmed body for later comparison. 59](#_Toc386660130)

[Table 12: Table showing various quantities for final PI tetrahedral concept car mesh. 59](#_Toc386660131)

Unless otherwise stated, all figures and tables have been created by the author.

# 1. Introduction

## 1.1 Background

Many industries make use of computational modelling to create simulations of how a product may function before it is physically produced. In this way, these businesses are able to save considerable sums of money that would otherwise have to be diverted towards manufacturing physical prototypes. The automotive sector is one such field which has seen an increasing uptake in the use of simulations in areas such as combustion modelling within the internal combustion engine, structural analysis of the chassis’ response to impact and, most relevant to this project, to decide how a vehicle might behave in terms of aerodynamic performance. This particular information can be used to determine how the exterior could be optimised to increase down-force and therefore increase acceleration for high speed vehicles, how air flow through the engine bay may best be used to passively cool it to reduce the vehicle’s energy requirements and which features produce the most drag and so might need refining. The role of Computational Fluid Dynamics (CFD) within an integrated design process forms the basis for the work of this group, and individual, project.

## 1.2 Project Aims

This group project aimed to investigate modern methods for integrated automotive design, combining contemporary Computer Aided Design (CAD) techniques with high performance CFD analysis, validated against testing of a physical prototype created via a state-of-the-art rapid prototyping process. Specifically, this individual report concentrated on the integration of CAD designs with the fluid simulation stage by developing a thorough understanding of how mesh creation is fundamental to the attainment of accurate simulation results. In industry, this stage often takes a disproportionate amount of the project time scale and is thus known as a “bottleneck” in production (1). Therefore, the aims of this project were:

1. To research the role of mesh creation within CFD.
2. To determine how meshes are structured in terms of the units that comprise them and the structures created by these units.
3. To examine mesh generation techniques, how these are implemented in common CFD software platforms and determine the strengths and weaknesses of these techniques when considering their use in automotive design.
4. To research how mesh quality is evaluated and whether these metrics are useful and if not which alternatives might be more valid. Use this information to advise the HPC group on best practice quality evaluation.
5. To produce some concept car designs and work with the other Design group members to examine how changing geometrical features impacted its aerodynamic performance, using Ansys Fluent. The High Performance Computing (HPC) group would also use these results during their studies.
6. To produce a final concept car design with the other Design group members, have it created through a rapid prototyping process and present this to the Experimental group for testing.
7. To apply some of the previously researched meshing strategies using a Reynolds Averaged Numerical Simulation (RANS) fluid modelling technique, using the simulation platform Ansys Fluent, on a standard automotive validation case and determine these strategies’ effects on solution accuracy.
8. To apply the most accurate strategies on the final concept car design to assess the accuracy of this relatively rapid technique as compared to the high performance methods and make recommendations regarding which techniques might be most appropriate for the automotive industry at large.

## 1.3 Report Structure

The report will be organised into chapters, split into sections by content. Firstly, the ‘Background and Theoretical Understanding’ section will explain the concept of meshing and its relevance to CFD and also draw conclusions relevant to the aims of the group project. Then, the *modus operandi* will be described and a Health & Safety study will be undertaken in the ‘Methodology’ section. Next, the ‘Design and Experimentation’ section will explain the procedures that were undertaken to achieve the previously mentioned aims and the ‘Project Analysis’ chapter will scrutinise not only the results generated, but also the undertaking of the project, in order to draw valid conclusions. The ‘Sustainability’ section will contain elements related to how the project intends to align itself with recent, global, environmental impact mitigation measures and, lastly, ‘Conclusions’ aims to summarise the main outcomes of the project and how work in this field might continue.

# 2. Background and Theoretical Understanding

## 2.1 Introduction

The following section aims to give a comprehensive overview of why careful mesh construction is paramount to simulation accuracy; what meshes are composed of; what the different types of mesh available are; how meshes (or ‘grids’) are generated; how to determine if a mesh is of good quality and finally how to improve the quality of a mesh in order to generate a more accurate solution. This information will later be used for investigating the effects of different types of mesh on simulation accuracy to determine how appropriate certain mesh types are for incorporation into the CFD of the final concept car.

## 2.2 The Role of Mesh Creation in CFD

Meshing is the term used to describe the partitioning of a region of interest (a ‘domain’) within a simulation into smaller volumes wherein the governing equations of the system, which would be the conservation equations in fluid dynamics, can be discretised & solved. The entire solving process requires that a systematic sequence of actions are undertaken, namely (2):

1. The nature of the bounding faces of the three dimensional domain is specified (for example as being a flow inlet, outlet or solid wall).
2. Properties of the fluid (such as material composition, density, temperature and so forth) are defined.
3. Solution algorithms and their numerical parameters are selected.
4. Initial values for all variables associated with the flow field are specified to give a starting point for the iterative solving process.
5. The continuous partial differential transport equations (PDE’s) are discretised by being replaced with a system of algebraic difference equations. This is undertaken by transferring the PDE’s to their more fundamental, integral, form which is then represented over a set of discrete cells where the solution variables are considered to be constant over the cell (3). Examples of some basic discretisation techniques in relation to fluid dynamics can be found by following the references provided (4) (5).The aforementioned initial values are then used as a starting point to perform an iterative solving process. This continues until the ‘residual’ (the sum if all of the terms of the Navier-Stokes Equation were put on one side, equalling zero if the exact solution were to be found) of each transport equation drops to a predetermined value. Since the residual is another way of displaying how much the solution to a transport equation differs from the exact answer, then, based on the requirements of the analysis, its value can be lowered to provide a more precise result.
6. By paying close attention to the residuals’ values, it is possible to ascertain when convergence has been achieved. Once this has occurred, variables associated with the flow, for instance pressure and velocity, can be displayed and analysed graphically by using the ‘postprocessors’ that are available with many CFD packages.
7. Following on from this, integral properties (for instance forces like lift and drag) and global properties (such as pressure drop) can be extracted from the converged solution flow data. Being able to monitor these quantities while the solution is running can also give an awareness of convergence as their values should become constant over time.

By evaluating the results of the whole array of elements, also known as ‘cells’ or ‘control volumes’, solutions for the system at large can be generated. This process, if used correctly, can be utilised within CFD, Finite Element Analysis (FEA) and other simulation software to create an accurate representation of a given situation, if it were to be physically envisaged. The advantages come from not having to expend the time, money and effort of actually constructing the scenario and also from being able to make significant modifications that would, in real life, otherwise incur potentially massive expenditures (6).

This report will only consider the Finite Volume Method (FVM) in the context of meshing henceforth as it is the most widely used domain discretisation process in the CFD industry (7) and that which will be used within all the simulation undertaken by VWT project members (8) (9). The basic principles of the FVM are similar to those defined in point 5., above:

* A meshing process is used to divide the domain into a number of ‘control volumes’, where the important variable, here a numerical fluid flux such as mass or momentum, is located at a computational ‘node’ at each volume’s centroid.
* The differential versions of the governing conservation equations that contain a divergence term are then integrated over every control volume by using the divergence formula (alternatively, ‘divergence theorem’).
* The values of the variables of interest at the faces bounding the control volumes are determined by a specified interpolation scheme (e.g. ‘First Order Upwind’ etc.) where the choice of interpolation scheme depends on the anticipated flow conditions.
* This results in a set of linear, algebraic equations for each control volume which express the conservation principle for the variable at the centroid. These can be solved either simultaneously or through iteration to ultimately provide a picture of the flow conditions throughout the domain, at a given instant (10).

Some of the advantages of the FVM are that the local conservation of numerical fluxes from one discretised volume to the next is ideal when attempting to compute problems dealing with such quantities as momentum and mass in CFD and also that the technique can be readily applied to arbitrary geometries using a variety of mesh types, leading to robust solving schemes (11). More detailed information on the practical implementation of this method to fluid dynamic problems can be found through the provided references (11) (12).

The danger inherent in modern CFD usage is that an individual can create what they believe to be a comprehensive set of conditions under which to model their intended scenario, applying boundary conditions, creating a mesh and specifying turbulence models with relative ease, but without any real knowledge of fluid mechanics or mesh generation. Since the limitations of these codes and inputs are unknown to the user, erroneous results are more than likely to emerge, therefore it is paramount that the individual has a good understanding of these fields so that they may discern whether a CFD solution makes physical sense or otherwise (13). Producing a colourful diagram certainly does not equate to realistically modelling a situation.

With this in mind, it is all the more important to generate a high quality grid, especially when considering the following factors:

* The mesh describes the geometry to be modelled and so must be precise in such a way as to capture all the relevant geometrical features which may impact the fluid flow around it.
* The mesh also describes the domain in which the geometry resides and so must have an appropriate resolution in particular areas to capture the flow effects which might arise here. Since there is a length scale at which the impact of turbulent eddies no longer significantly affects the flow, infinitesimally small resolution is not required therefore it is up to the user to define how small to form elements, with computational resources in mind, and therefore how accurate a solution they require.
* Many other variables come into play during a simulation, such as: element shape, element type, resolution of elements within boundary layers and alignment of cells with flow (10). These will be expanded upon in later sections.

## 2.3 The Components of a Mesh

As mentioned, in order to simulate a problem, the domain of interest must be broken down into many small elements. Although these elements can take various forms, those most commonly used are given in Figure 1:

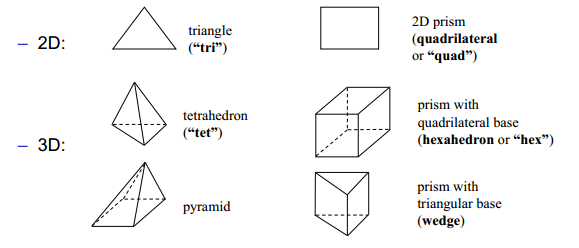


Figure 1: Depiction of the most commonly used element types in 2D and 3D simulations (10).

While tri’s and quad’s are apt for the construction of grids for solving two dimensional problems, they cannot be used for 3D flows. Here, tet’s are commonly employed to form the curved edges of irregular structures, such as the rounded bumper of a car, while hex’s are primarily used to mesh more regular geometries, as in the case of a wall-mounted cube often used as a validating case between fluid dynamics experiments and simulations (14). Another benefit of using hex’s is that fewer cells are required to mesh a given volume, compared to tet’s. This can lead to a lower computational resource requirement for the user if they are able to create a high quality structure in this way. Lastly, pyramids and wedges are used to transition between areas that either contain tet’s or hex’s due to their mix of three and four sided faces. It is obvious that as the size of the elements used decreases, the ability to resolve finer geometrical details increases.

Common terminology when referring to a computational grid are given in Figure 2 and Table 1:

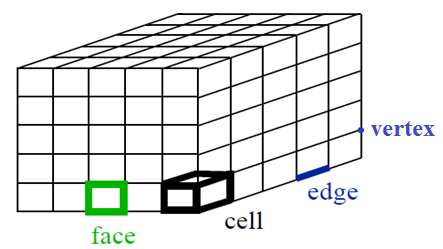


Figure 2: Diagram representing common terms relating to computational grids, in this case a symmetrical hexahedral grid.

Table 1: Common terminology used when referring to a computational mesh.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Cell | The domain is split into these control volumes. |
| Node | The centroid of a cell. |
| Vertex | The point where cell edges meet. |
| Edge | The border of a face. |
| Face | The border of a cell |
| Zone | An assemblage of cells, faces and vertices (e.g. wall boundary zone or fluid cell zone). |
| Domain | A grouping of vertices, face and cell zones in which to model a scenario. |

## 2.4 Types of Mesh

### 2.4.1 Introduction

Mesh generation is always specific to the problem at hand, depending on numerous factors, from geometrical intricacy to boundary conditions and available computational resources. For this reason, many meshing structures exist to allow a user to best apply the discretisation schemes to their individual project. This section aims to introduce some of the main types employed within modern CFD.

### 2.4.2 Structured Mesh Varieties

The underlying principle of all structured 3D mesh varieties is that they consist of a grid of planar cells with six faces, which may be distorted from rectangular, each numbered using indices (i,j,k) that are not required to correspond to a set of (x,y,z) coordinates. A simplified, two dimensional, example of this is provided in Figure 3 below:



Figure 3: Example 2D grid showing 'i' and 'j' grid edge orientations that do not conform to the (x,y) coordinates of the domain in which the grid lies. The shaded cell has coordinates (i = 4, j = 3) (2).

The creation of a grid involves specifying a set of vertices on the boundary faces within the domain in three dimensions, which are connected one-to-one to form the internal mesh and where each cell is uniquely identified by an index triplet (i,j,k) that corresponds to the numbered intervals between the vertices. Alternatively, some CFD codes use the vertices as index coordinates instead (2). It is therefore straightforward for these points to be mapped onto a matrix, whereupon the solver may easily draw upon this cell location data as it performs the iterative solution process. These meshes are comprised of quad’s (2D) and hex’s (3D).

The ‘Cartesian distribution’ is a form of mesh structure that leads to the highest possible accuracy of the discretised formulas (15), however it is only convenient where the domain geometry to be meshed is very regular and does not contain curves. Cartesian grids may also be non-uniform as cell size varies from one region to another to accommodate some form of refinement, as in the Figure 4 below. The same figure shows an attempt to fit the mesh structure to a curve, which requires the implementation of a stepwise approximation. The generation of this kind of approximation is difficult and requires cell deletion from unnecessary regions, which amounts to wasted computer memory storage and resources. Furthermore, the coarse, stepping, transition which attempts to describe the curve induces errors in boundary-layer related flow calculations, and therefore the final solution. Efforts to increase cell resolution in order to account for this again lead to wasted resources due to refinement within unrequired regions. This means that Cartesian grids find limited applications within industry where complex geometry, for instance an aerodynamically optimised car shape in the VWT, is commonplace. However, their inclusion within regions devoid of complexity, such as some parts of the open space between a car and the bounding walls of the domain, may well prove useful, but more on this topic later.

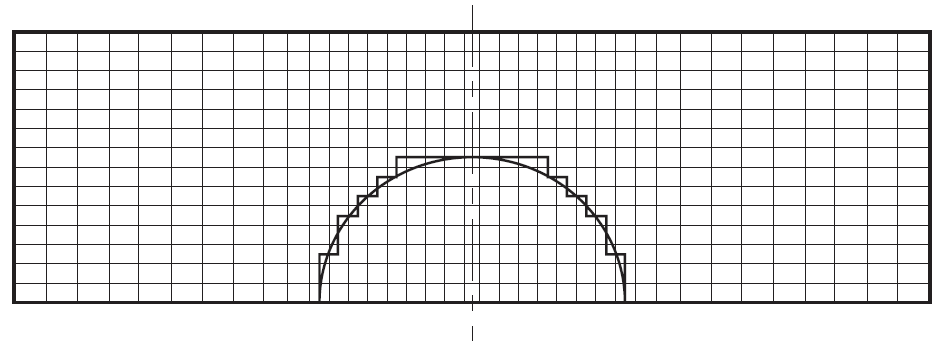


Figure 4: Cartesian grid for simulation of flow over cylinder and stepwise approximation of mesh to smooth curve of geometry (16). Note wasted cells within cylinder’s interior that do not form part of simulation, waste computational resources and require removal prior to solving.

The next level of structured meshing is known as a ‘structured curvilinear’or ‘body-fitted’ grid. This allows the flow domain to be mapped to the computational domain for relatively simple shapes but becomes much more involved when trying to describe complex geometries. There are two types; the first is the ‘orthogonal curvilinear mesh’in which all grid lines meet at right angles and which has been used to successfully model the flow over curved boundaries such as that of an aerofoil (17), as in Figure 5. From this, it is evident that a major disadvantage of the method is the wastage of computational resources due to the global nature of any cell refinement performed; here the cells downstream of the aerofoil being unnecessarily small. One way to improve on this is by using special fitting topologies where the grid conforms to the geometry in such a way as to accentuate refinement in particular areas without using an overly dense mesh in non-critical regions. Terminology relates to the letter which the grid most resembles with ‘I’,’C’,’O’ and ‘H’ being the most common. An example of the latter is provided in appendix B, Figure 32. The second variety is the ‘non-orthogonal body-fitted grid’, where the lines in the grid do not intersect at right angles and this has been applied to the cylinder flow example in Figure 6:

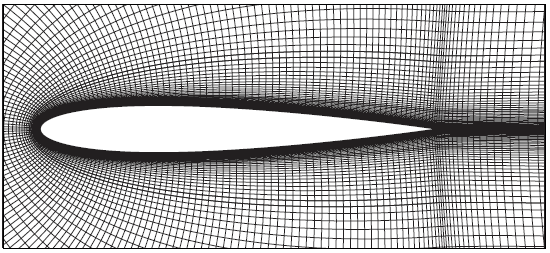


Figure 5: Orthagonal curvilinear mesh fitted around an aerofoil profile in 2D (16).

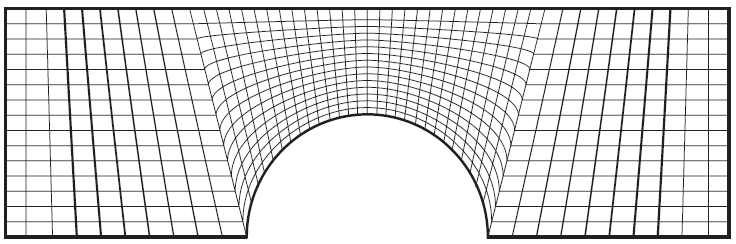


Figure 6: Body-fitted, non-orthogonal grid applied to cylinder flow example from Figure 4 (16).

Though these types of grids may seem like an ideal way of encompassing an irregular structure with a relatively uniform mesh, they do suffer from a number of downfalls:

* They are usually heavily involved and tedious to create.
* Grid refinement is not usually confined to specific areas; it has a global impact.
* The solving schemes associated with curvilinear grids are particularly complex, due to the geometrical flexibility afforded by this type of mesh (more detailed information on the modifications that must be made to the governing equations to ensure their suitability is available through the references (18)).
* An inability to easily map the solution domain into a rectangular solid body, where each face is a parallelogram (for 3D structures), can create unnecessary local variations and skewed grid lines.
* Unrequired grid resolutions in some areas can lead to difficulty in mapping.
* Where complicated geometries exist, such as those with internal configurations, mapping may be impossible.

Where the previous examples employed ‘single block’ configurations, a more advanced form exists, known as the ‘block structured’(alternatively ‘multi-block’ or ‘composite’) type. Here, structured blocks with differing topologies are applied to subsets of the domain. This avoids the restrictions associated with single-block grids, whereby, adding a vertex to generate a family of lines through it, and therefore cause refinement in that area, also causes these lines to propagate throughout the rest of the grid. Creating the right interplay between these new lines and complex geometrical features so that the result is a good quality grid can become very cumbersome. Additional flexibility can arise through specifying ‘non-matching block interfaces’, where it is not made compulsory for the vertices of one block to align with those of its neighbour. The consequence of this action is that the flow solver must use sophisticated interpolation procedures to allow for an accurate transfer of information through these misaligned interfaces. Figure 7 shows a 3D multi-block grid without matching interfaces:

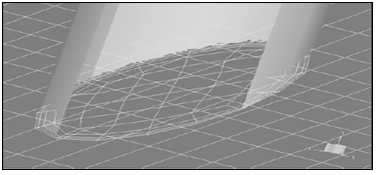


Figure 7: Example of a 3D multi-block grid, showing surface mesh topology, where interfaces are misaligned (15).

One further option that can be used to improve mesh-geometry flexibility is to implement an ‘overset grid’ (otherwise known as the ‘chimera technique’). Here, grids are generated independently, around stationary or dynamic bodies, that overlap an existing background grid. This system is usually applied where there are multiple bodies moving relative to each other and where each body requires an attached mesh to resolve the flow around it during the specified time period. The technique is intensely difficult due to the need to perform accurate interpolations between these overlapping grids in three dimensions, while their proximities continually vary. Figure 8 and its inset gives an example of a typical configuration. Recent developments concerning this process are found through the provided references (19) (20), though this technique will not be applied within the scope of the VWT project.

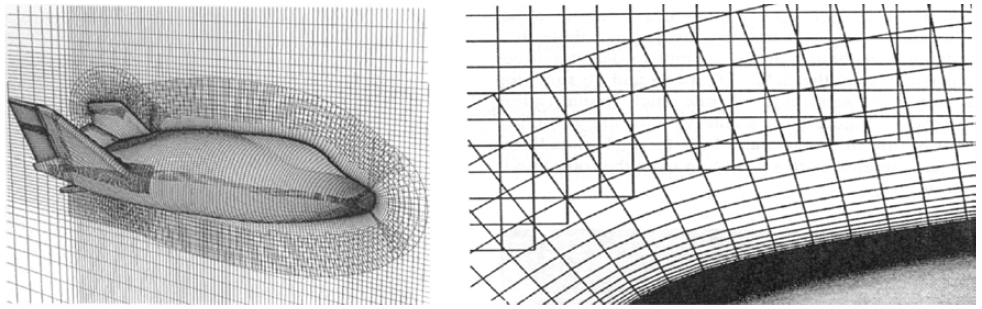


Figure 8: An example of a body-fitted grid, surrounding a spaceship, which overlaps an existing background grid (15).

### 2.4.3 Unstructured Meshes

Unstructured meshes are made from cells of differing shapes, but most commonly these are: tri’s and quad’s (2D) and tet’s and hex’s (3D). In three dimensions, cells are not identifiable through the (i,j,k) notation previously seen in structured grids, but instead the cells have numbers assigned to them by some process specific to the CFD solver. Cell-centred methods, which are those associated with the FVM, store the flow data relevant to each cell at its centroid. Since a control volume will always have more vertices than centroids, this method requires fractionally less computing memory than vertex based methods used with other solving schemes (16).

More recently, researchers have begun to implement meshes containing non-standard polyhedra, such as those with an arbitrary number of faces like the dodecahedron, within the FVM. These methods often concentrate on the amalgamation of existing groups of pre-optimised tetrahedra into large polyhedra (or simply hexahedra (21) that can represent the geometry as accurately as the pre-formed tetrahedral mesh had (22)) and robust solvers have been created that can accurately deal not only with simple flow situations like that over a flat plate, but highly complex scenarios resulting from hypersonic fluid speeds (23). Though there are many reported benefits, such as an incredible flexibility in mesh generation around highly complex geometries while also partitioning the domain optimally in order to reduce the necessary cell count (24) (25), the technique and its successful integration into CFD solving schemes is relatively new. Therefore, as it is a fledgling field, it may take some time before it is widely verified and integrated into most commercial CFD programs and utilised extensively by industry and others. Upon acceptance, however, it is feasible to assume that it will lead to great decreases in the time required to create high quality meshes around complex geometry, and therefore the project time dedicated to CFD, especially when automated generation techniques have been optimised.

### 2.4.4 Comparison of Structured and Unstructured Meshes

Figure 9 (a) and (b) show 2D representations of two unstructured, monolithic meshes, one comprised of tri’s and the other from quad’s, and which use exactly the same grid spacing as that of Figure 3 above.

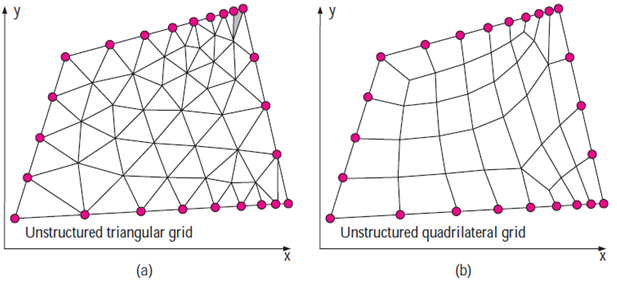


Figure 9: Two dimensional examples of (a) tri and (b) quad unstructured meshes using the same interval spacing as Figure 3 (2).

If it were assumed that the decreasing grid spacing to the right of the near-horizontal axis (of all three grids) corresponded to an attempted cell resolution increase near a fluid boundary layer, then it is obvious that the structured grid in Figure 3 achieves this to a much greater degree than either (a) or (b) and with fewer overall cells; 32 for the structured grid compared with 76 for (a) and 38 for (b). This effect is also applicable to 3D grids and means that a well-defined structured grid will allow a much finer resolution near areas of interest than unstructured varieties, for the same number of cells, due to its regular one-to-one connectivity.

Though this may be the case, the generation of structured surface meshes, which guide the volume mesh that fills the domain, for complex geometries is very time consuming as complex iterative smoothing schemes are required to attempt alignment of cell boundaries with those of the physical domain. Multi-block schemes, where the iterative schemes are not applied globally but instead are individually set for each sub-domain, can make the process less arduous but generally, unstructured tetrahedrals are superior. This is because they can be thought of as the limiting case of a multi-block grid, where each cell is itself an individual block, and so is not confined by constraints applying to any other cells. It is therefore a most attractive method for meshing within industry, where product development deadlines are tight and companies will choose the ability to form complex geometry-matching meshes quickly, at the expense of gaining particularly accurate results that may otherwise have been produced by a structured mesh which took months to generate.

On the other hand, the storage of grid data, i.e. which vertex is neighbour to which, requires significantly more memory in unstructured than for structured meshes and therefore makes parallelisation of solving codes across PC memory cores more difficult and so the whole calculation becomes more time consuming (26). However, advances in processor partitioning for various meshing procedures (27) (28) are enabling this aspect of the unstructured CFD process to become more competitive with their structured counterparts.

### 2.4.5 Hybrid Meshes

Though unstructured meshes may be more versatile for meshing complex geometries, in many applications, and most significantly (in relation to this project) the automotive industry, the ability to determine fluid behaviour very close to the wall is of paramount importance when attempting to calculate a quantity such as the drag on a vehicle (29) (9). For this reason, most simulators will create a number of initial layers away from a vehicle’s surface (using a triangulated surface mesh representation of the geometry as a starting point) using an automatic, hexahedral structured mesh to encapsulate the boundary layer and resolve it accurately before filling the remaining domain volume with an automatically generated, unstructured, tetrahedral mesh. This is one example of a ‘hybrid mesh’ and the region between the rectangular/ square-faced structured elements and the triangular faced tet’s is bridged by pyramids or prisms, sometimes themselves known as ‘semi-unstructured’ elements (see Figure 10). Therefore the essence of these grid varieties is to use the virtues of each mesh type, through a mix of cells, to, for example, provide fine resolution in important areas at a lower cell count without having to use the same resolution away from here and waste computational resources in the process. Another use is found within ‘Hexcore’ meshes; a volume mesh with a tetrahedral outer structure and a hexahedral core that minimises cell usage by using hex cells in large, empty regions. Figure 10 gives a sample representation of a hybrid grid generated on a simplified car geometry:

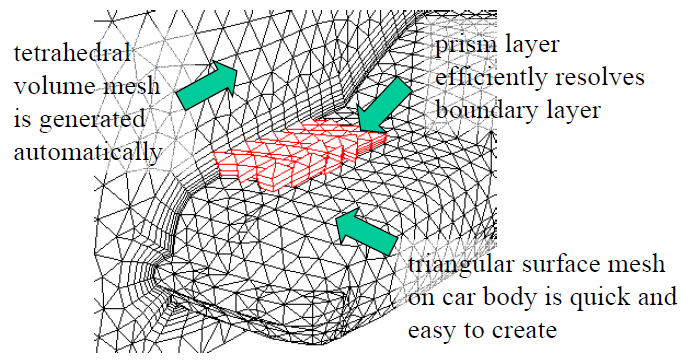


Figure 10: Annotated diagram of a hybrid grid formed on a simple car geometry, showing the transition from hex to tet elements on the symmetrical cut-plane. The diagram also gives a 3D representation of some of the surface hex elements (in red) (10).

## 2.5 Grid Generation Techniques

### 2.5.1 Introduction

Grid generation is an incredibly diverse, multidisciplinary field which is developing at an ever accelerating rate. Meshing strategies must be conceived to partition problem domains for almost any kind of simulation that must be undertaken; from the fluid dynamics surrounding a space shuttle’s re-entry, to the buckling mechanisms of a skyscraper’s internal support structure and even predicting the availability of electromagnetic signals to electronic components embedded in consumer products (30). Figure 11 gives an impression of the breadth of the field, as it stood in 2005, and the following source will give the reader an overview of each type (31), though this level of coverage is not within the scope of this project. Instead, this report will be limited to describing those mesh generation algorithms which predominate in Fluent and, to some extents, within Pointwise and OpenFOAM. In this way, it is hoped that the triviality which is sometimes associated with mesh generation by the unassuming observer will be dispelled and replaced with a deeper understanding of the complexities and importance of this step in the CFD process.

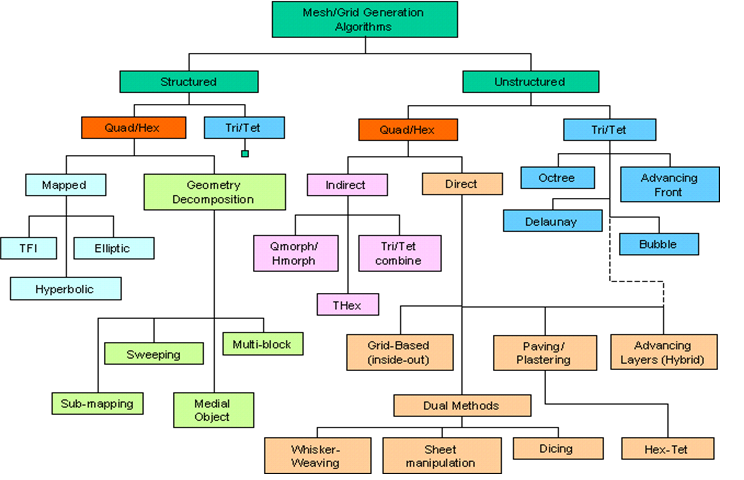


Figure 11: Overview of underlying (yet slightly dated) meshing algorithms used in variety of research and industrial fields (31).

### 2.5.2 A Word on Data Structures

While the aforementioned indexing scheme (section 2.4.2) inherent within structured meshes makes for a relatively easily searchable database from which a computer can draw data, the case is not so straightforward for unstructured mesh varieties. The sheer range of grid generation techniques is accompanied by an equally numerable set of practices for organising how this grid information is stored and accessed. These are necessary so that certain queries can be answered, such as; provide the list of mesh faces connected to a given vertex or determine all the vertices residing within a particular portion of physical space. In the latter case, this is an example of a ‘range search operation’, where vertex coordinate data that is poorly organized will result in a looping over all of the mesh’s vertices to arrive at the answer. The required time for this search operation is then proportional to the number of vertices, *v*, and it is usually preferable to refer to this situation by saying the algorithm is of order *v*, or *O(v)*. It is possible to reduce the number of operations for that type of query by orders of magnitude through efficient data organisation, which leads to significant computational effort savings when *v* is large.

Furthermore, while these data structures must efficiently encode adjacency and/or the incidence relations between mesh entities, they must also do so in a flexible manner so that local topological modifications are possible without sacrificing the ability to navigate the grid efficiently using these local incidence relations. Therefore, there exists a tradeoff between being able to rapidly locate data stored within a mesh area by encoding more incidence data and attempting to minimise the storage space required for this incidence information. Additionally, data storage programs are often optimised for particular cases, e.g. the use of homogenous meshes comprising singularly of tet’s, and so suffer temporal losses when attempting to deal with more varied cases. Research in this field is ongoing and providing promising results with regards to multiphysics simulations and multicomponent geometries (32) (33), which are also applicable to more standard CFD cases, and more general information on data structure synthesis is available through the following (34).

### 2.5.4 Surface Mesh generation

Prior to creating a mesh to fill the volume of a domain, it is usually necessary to generate a mesh on the surface of the geometry, such as the exterior faces that comprise a CAD model of a car, which may act as a starting point to generate the remaining mesh. Just as volume grid generation techniques vary based on the type of mesh being used, so too do the surface meshes; structured and unstructured meshes each utilise a range of techniques to effectively map the model’s surface geometry to ensure that they capture the required details of the model being used.

In industrial applications, where a design team has previously put together a CAD model to accurately describe the product, the technique begins by importing this model into the relevant piece of CFD software, most commonly as an Initial Graphics Exchange Specification (.IGES) or STereoLithography (.STL) file, due to the compatibility of these geometrical representations with the meshing systems. Some CFD packages include a utility, such as Ansys Fluent’s ‘Design Modeler’, which enables geometry creation within the program itself, however the sketching tools are usually too simplistic and non-native to designers to be a viable option. The CAD data will be represented by spline composite curves and tensor-product surfaces such as Ferguson, Bezier or Non-Uniform Rotational B-Splines (NURBS) which provide the meshing software with an adequate mathematical description of the geometry (35). In the case of unstructured meshes, the surface representation consists of an approximation of the geometry by using a set of planar, triangular facets. Therefore the discretisation of a surface involves the accurate positioning of a set of vertices across it and the definition of the relationships between a vertex and its neighbours. Two main techniques exist: ‘Parametric space meshing’ and ‘Direct 3D meshing’. The first is only possible if a parameterisation of the three dimensional surface (i.e. a two dimensional mapping of the 3D region) is available and the most effective technique is to use a form of anisotropic triangle mapping. Here, the triangles appear warped in 2D but are generally equilateral in 3D. The 2D triangle stretching process is based upon a field of surface derivatives. These ensure distances within the parametric space are measured as a function of direction, and location on the target surface, and standardised triangle generation algorithms are used to produce the initial 2D mesh. Figure 12 gives a simplified diagram to help explain the process. More information on this technique is available through the following source (36).

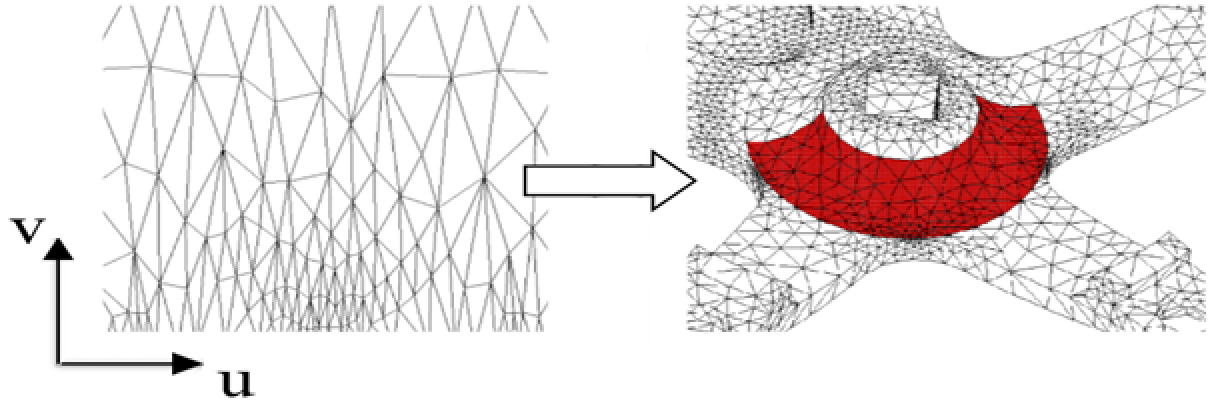


Figure 12: An example of the 2D parameterisation of a 3D surface and stretched triangle generation upon it that leads to generally equilateral mapping in 3D (31).

In contrast, the direct 3D meshing approach (see Figure 13) creates an advancing front of triangles immediately upon the three dimensional surface through a systematic set of steps: first a line is formed between two vertices at the advancing front (line AB), then the tangent plane at this front is formed by averaging the normals at these two vertices (NC), which is followed by the definition of a new point (D) that forms an ideal triangle with these two vertices which is then projected to the surface by finding the point of closest contact. This form of mapping is often slower than the parametric approach due to the requirements of determining overlapping or intersecting triangles in three dimensions and also the heavy use of geometry evaluators to create normals and projections. It does, however, usually result in higher quality triangles and avoids the issues resulting from poor surface parameterisations that are typical of CAD files (31). On the other hand, parameterisation can be more robust as there is no requirement to calculate 3D intersections.

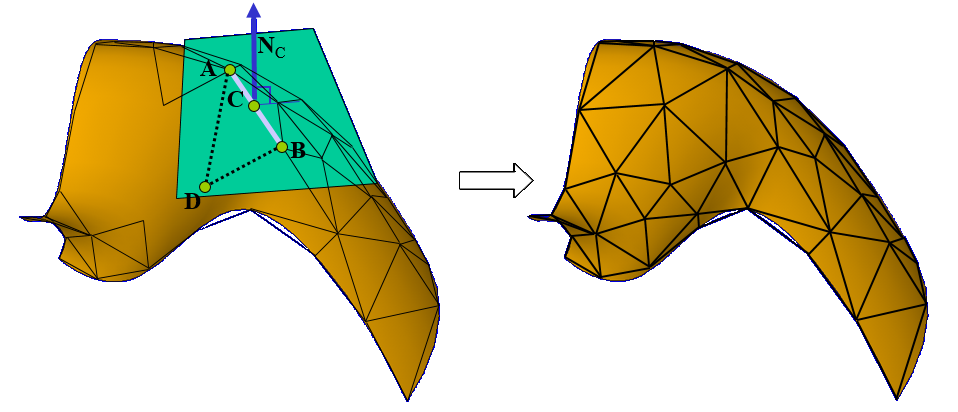


Figure 13: Direct 3D meshing approach to triangulating a curved input geometry via an advancing front technique (31).

The generation of structured surface grids is considerably more involved due to the need to satisfy various element quality metrics such as cell aspect ratio, smoothness, skewness etc., all of which will be detailed in later sections. Though certain 3D shapes, such as spheres, missile geometries and even complete wings, can be relatively easily described through mathematical formulae and mapped with a high quality structured mesh, doing the same for complex geometries is very work intensive. Because of the one-to-one connectivity of a structured surface mesh with its bounding domain, the successful generation of a surface mesh essentially entails the generation of a volume mesh also. For this reason, it will be left to the following sections to describe a selection of the processes available for doing so.

### 2.5.5 Input Geometry Considerations

As was outlined in section 2.5.4, the mathematical format that the CAD geometry arrives in is an important aspect in determining how it will be meshed. It is this importation stage that often results in most of the problems that occur down the line when attempting to mesh a CAD model. This is partially due to the divide that exists between industrial design and CFD departments within organisations that results in the analogy of throwing the model ‘over the wall’ from design to analysis. Issues ensue as the designers are not aware of the level of complexity that is required in a model for it to be accurately modelled; for instance including many interior components, such as steering wheel and seats, and very small exterior components, for instance screws, that not only have an indiscernible impact on the flow regime but require immense computational effort to capture accurately within a mesh. Other issues are cause by ill-defined features such as those that overlap or where gaps exist between surfaces, which make little difference to the design department and may even be all but imperceptible, but provide a significant obstacle for the meshing algorithms. This is where the term “Dirty CAD” arises from (37), which leads to the CFD engineers then having to strip down all these unnecessary components before beginning the meshing process and fixing surfaces to create “watertight” geometries (37) that altogether can seriously hinder the progress of a project. More recently, the aforementioned divide seems to be closing in industry, with the role of designer and analyst often merging into one, as steps are implemented to guide the designer in performing accurate simulations (38), often within integrated design and modelling software platforms to encourage faster product throughput due to familiarity with the user interface (39).

There are many ways to perform ‘CAD cleanup’, dependent on the meshing software being used. Most programs contain functions that detect gaps in surfaces, for which the minimum size tolerance is adjustable. These locations can then be noted and the surfaces can be rejoined with extra planes by editing the original CAD file. Fluent contains such a function and also allows for automatic patching of holes and fixing of surface overlaps within the mesher using their surface ‘wrapping’ tool, after defining the maximum gap size to search for (40). Fluent also contains a ‘defeaturing tolerance’ option where a minimum size can be set so that any feature with dimensions smaller than this specified value is ignored by the mesher and effectively deleted from the model (41). This is particularly useful when trying to easily delete non-relevant features such as very slightly embossed writing on a model part that would otherwise create problems for the mesher. Alternatively, Pointwise’s “Examine Boundary Proximity” and “Grid Merge” functions allow for the rapid identification and merging of CAD faults (42). Also, the ‘Quilting’ function makes for straightforward unification of surface patches into single quilts that simplify the overall mesh generation process as they join partitions that would otherwise have to be meshed separately, potentially leading to low quality, unnecessary, elements (43). Fluent also has a similar function known as ‘Virtual Topology’ which can achieve the same result (41). The tools for reducing the bottleneck that occurs in the meshing stage between CAD model creation and solution initialisation are becoming faster and more automated. However, this is still a largely human labour-intensive area as the definition of a fault is left at a user’s discretion due to the unique requirements of each simulation.

### 2.5.6 Volume Meshes – Structured Approaches

As has been previously outlined, structured meshes consist of regular hexagonal elements connected to form three dimensional grids where the one-to-one connectivity within them ensure that all the geometry’s surface mesh vertices are matched by an equal number of vertices on the boundaries of the domain. Furthermore, accessing specific data, such as the flux computations across a particular cell’s faces, is straightforward due to the (i,j,k) indexing scheme that conforms to the coordinate system of the domain. As shown in Figure 11, there are two distinct families of meshing based upon the underlying algorithms which generate the mesh; those that are ‘mapped’ and those that rely upon ‘geometry decomposition’.

Mapped schemes are either ‘Elliptic’, ‘Hyperbolic’, ’Algebraic’ or more likely some combination thereof. A standard mapping process involves using six-sided blocks, whose faces are described by parametric coordinate planes, where the eight corners of a block are ‘mapped’ to the corresponding points of the physical topology in order to describe it. The mapping process allows for rotations, translations, scalings and potentially even non-uniform projections in order to match the given topology whilst maintaining relatively high quality elements. After the boundary curves of the block (i.e. its bounding edges) have been decomposed to the correct size, a TransFinite Interpolation (TFI) algorithm (such as Lagrange or Hermite (26)) is used to determine the ideal placement of interior vertices within the block which will form the desired hexahedral mesh. Figure 14 Figure 15 are given to help illustrate this concept:

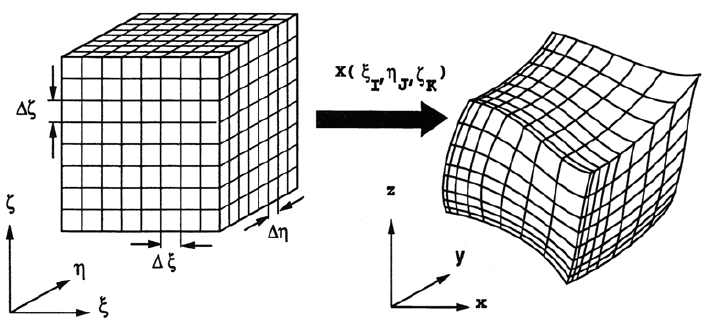


Figure 14: Example of the mapping of a mesh from the computational to the physical domain, where TFI provides the singular framework that dictates the mesh structure by the function *X(ξ,η,ζ)* (44)*.*

While these traditional mapping schemes attempt to map hexahedra to a single block, in order to reduce the complexity of the mapping functions (and ensure element quality), it is common practice to use multiple blocks (the ‘multi-block’ method mentioned in section 2.4.2) to better resolve the topology of the target geometry. The multi-block technique belongs to the geometry decomposition branch of the meshing field since it makes use of a number of solids with cuboid topologies, each of which are separately mapped to a region of the target’s surface and where the mesh structure of each section is consistent with that of its neighbours to allow for uniform grid connectivity throughout the domain (see Figure 15). Multi-blocking algorithms are the most popular and commercially practicable technique for generating high quality, structured hexahedral meshes upon complex geometries. Indeed, they are the primary method utilised by Pointwise for such applications (45), and appear within the technical capabilities of most leading meshing software (see appendix A, Table 3).

Both techniques are highly labour intensive when dealing with very complex geometries, especially when attempting to control mesh metrics such as grid spacing and orthogonality. Though structured meshes often allow a simulation to be solved faster and more accurately than by using an unstructured mesh, it is the time-constrained nature of modern industrial applications that often prevents structured approaches from being used. Detailed information about the specific mapping schemes and TFI algorithms are available through the sources mentioned in this section as well as in the following (46).

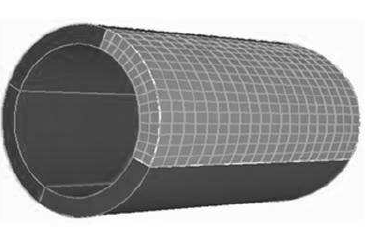


Figure 15: The shaded region displays the mapping of a simple cuboidal block to form one section of a (multi-block) cylinder, whereupon its internal mesh has been generated (47).

### 2.5.6 Volume Meshes – Unstructured Approaches

Unstructured meshes are characterised by their lack of uniform structure, which often leads to the creation of elements that would be deemed unsatisfactory by standard mesh quality evaluation tools. However, they are useful since they allow for rapid resolution of complex geometries and therefore permit the fluid dynamic governing equations to be applied to highly complicated scenarios, such as those useful for industry level simulations. The approaches are mainly divisible by the type of element in place.

Where the creation of unstructured tetrahedral meshes is concerned, there are two main methods. The first is underpinned by the ‘Delaunay Triangulation’ method which is based on the principle that any cell vertex within the domain must not be contained within the circumsphere of any tetrahedra within the mesh, where a circumsphere is the sphere whose outer surface passes through all four vertices of a tetrahedron. Figure 16 illustrates this concept in 2D:



Figure 16: 2D example of a Delaunay triangulation where no circumcircle of a set of three vertices that make up a triangle contains the vertices of any other triangle (31).

The Delaunay method is not itself a way of creating a mesh, but rather a way for a set of vertices to be triangulated. Therefore a technique for point generation within a volume is required. Notable examples are the ‘Bowyer-Watson’ and ‘Tanemura-Ogawa-Ogita’ algorithms and more detailed explanations of these can be found through the following references (26) (48). From a superficial perspective, a standard approach is to first create a set of vertices on the boundaries of the target geometry (which may be straightforward given the correct CAD input file type) and then apply the Delaunay criterion to triangulate these vertices. Following this, an algorithm incrementally inserts vertices into the volume of the domain, while refining tetrahedra locally to maintain the Delaunay criterion, until the entire domain has been successfully meshed.

The second most well-known unstructured tetrahedral generation technique is called the ‘Advancing Front’ method (49). Essentially, tetrahedra are progressively formed inward from the existing triangulated surfaces; in three dimensions this would constitute the filling of the volume between the triangulated surface of the CAD model and the surfaces of the domain in which it resides. The active ‘front’ is the area where new tetrahedra are continually formed and for each triangular facet upon a tetrahedron on this front, the optimum location of a fourth vertex to create a new tetrahedron is computed. Alternatively, should an adjacent vertex on the advancing front match this criterion, then this may be used to create the tetrahedron instead. The choice between these two options is made in favour of that which will result in the highest quality element being produced. Intersection checks run simultaneously to ensure that tetrahedra do not overlap as opposing fronts advance towards one another and sizing functions can be employed to specify element resolution. Figure 17 gives a 2D example of this technique. Delauney triangulation methods are usually preferable, as searching for appropriate nearby vertices and performing complex continual intersection checks in 3D make the advancing front method relatively slow (26).

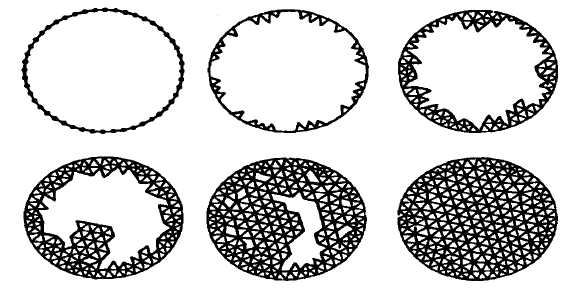


Figure 17: 2D example of the filling of a domain using the advancing front technique. The front is said to be ‘empty’ when there remains no space to be meshed (26).

The ‘Advanced Layering’ method is a variation on the advancing front that instead extrudes hexahedra from a triangulated surface into the domain. The number of layers, and other metrics of the hexahedra, are pre-specified and this provides an effective way of composing elements to accurately resolve the boundary layer surrounding an object. After the desired layer formation has been achieved, an unstructured tet scheme, such as one of those mentioned, would be applied to fill the remaining volume, as seen in Figure 10. This is a common form of the ‘Hybrid’ grid, as mentioned in section 2.4.5, and is often used in automotive design in order to take advantage of the benefits of both tet’s and hex’s; hexahedral layers on the surface capture near body effects accurately as the flow is aligned with the cells but their generally unstructured arrangement, and that of the remaining volume tet’s, allows great flexibility for meshing intricate designs.

There is one final family of methods which is becoming more commonly used as research continues to reinforce their adaptability for complex modelling, including flow simulations, and these are known ‘Octree’ (in 3D and ‘Quadtree’ in 2D) techniques. These methods allow for local mesh refinement, for example near to a wall, within a non-uniform Cartesian grid (as mentioned in section 2.4.2). These grids are otherwise known as ‘non-conformal’ due to the production of ‘hanging nodes’, i.e. lines that divide cells which do not propagate throughout the whole grid and therefore produce recognisable grid structures, as displayed in Figure 18. Though these grids are technically structured upon initial creation, they are able to handle the presence of curved and complex geometries within them and the alterations that enable this mean that they lose the ‘structured’ definition. The refinement used to capture sloped surfaces is the result of a systematic process; a cuboid is first generated whose surfaces are those of the whole domain. This is then recursively divided at the boundary, while intersection calculations between the grid and geometry are carried out, until the user defined cell resolution has been reached. The surface of the geometry itself does not have a mesh applied to it, but rather, is defined by the interaction with the surrounding grid. These schemes differ through their treatment of the interplay between the grid and the boundaries of these geometries.

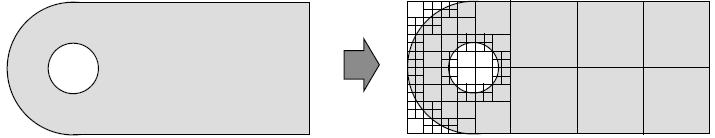


Figure 18: Refinement around a small feature in a simple 2D quadtree example with hanging nodes clearly visible in refinement region (15).

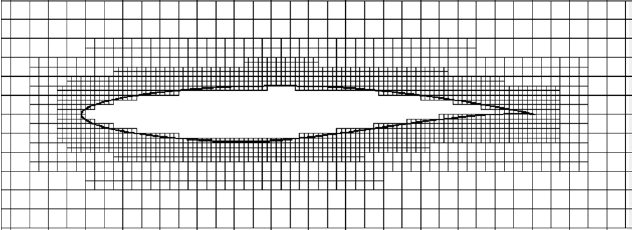


Figure 19: An example of quadtree refinement around an aerofoil profile with the ‘staircase’ boundary treatment employed (15).

The simplest example of Octree is that which was illustrated in section 2.4.2: local refinement of cells around a curved surface that results in a stepped effect after cells interior to the geometry are removed, better known as the ‘staircase’ variety. This is most commonly used where near wall behaviour is of little relevance to the simulation and so computational effort is not directed towards resolving these areas accurately. Staircasing might alternatively be used where computational expense is less important, therefore a very dense mesh can be generated to accurately capture the geometry and near-wall effects. Both of these scenarios are rare and so the staircase technique is not commonly found in the literature. Figure 19, above, gives a coarse representation of a 2D aerofoil profile with this method.

The second scheme is called the ‘Immersed Boundary Method’ (IBM). Here, the same recursive refinement method can be employed to capture the geometry in significant detail and a simple ray-tracing scheme uses a ‘tag function’ to classify cells as being either inside (‘solid’), outside (‘fluid’) or on the immersed boundary (‘interface cells’) (50). Solid cells are discarded and a force (or alternatively ‘source’) term, known as the ‘forcing function’, is added to the fluid momentum equations in the solver in order to mimic the effects of the boundary. IBM’s differ in whether they incorporate the forcing function into the governing equations before (the ‘continuous forcing function’) or after (the ‘discrete forcing function’) they have been discretised for the domain. The discrete forcing function is more appropriate where a definite (rather than elastic) boundary exists and for high Reynolds number flows, as in the case of flow past a vehicle (51). More information on this technique and mathematical considerations for its use are found by following the provided source (52). The main benefits of the IBM are that it does not require painstaking work to resolve the geometry and so can handle ‘dirty’ CAD files, and the simple Cartesian grid requires relatively little computational effort to generate and later solve (53). Figure 20 gives a simple representation of a geometry immersed in an IBM Cartesian grid:

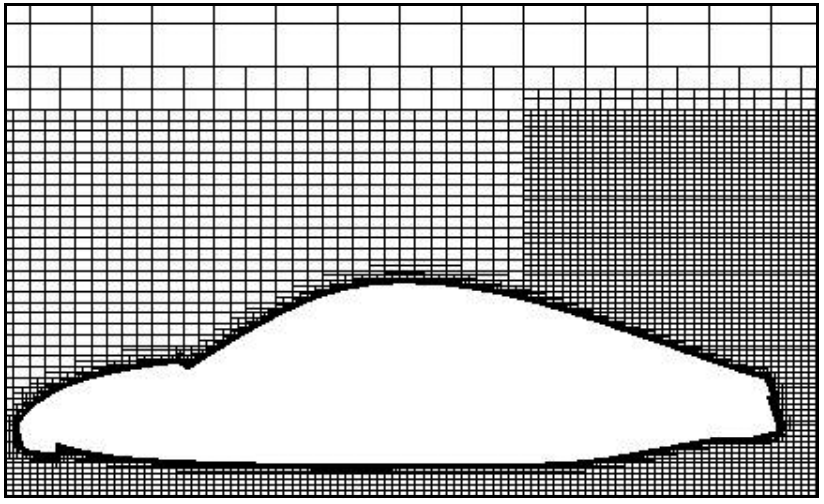


Figure 20: 2D example of IBM around a simplified car model, after 'solid' (interior) cell removal (54).

The IBM is an interesting case as, though it is relatively simple to implement, there is conflicting information over its validity in solving practical scenarios. Research has shown the IBM to be relatively accurate for simulating aerodynamic flows past complex geometries (55), for instance, one paper finds the IBM combined with a Reynolds Averaged Navier Stokes (RANS), second-order implicit fluid solver to gain results within 5% of those generated by a traditional body-fitted grid for the flow past a Chevrolet Tahoe SUV (56). However, these results were generated for a CAD geometry that had been pre-cleaned and so the technique’s effectiveness for a situation where one of the major benefits of its use has been mitigated is called into question.

Other simulations, such as those by the U.S. military of an object in free fall through a fluid (57), have demonstrated this technique’s validity but as it is still relatively novel, it has not seen wide scale uptake within commercial CFD packages. Fluent introduced an IBM incorporated solver in 2009 (54) as a specialised add-on and though they claim it is a useful method to be used in the kind of prototyping environment that necessitates the rapid creation of many models for testing, it seems to be little-known and generally under-utilised, with the company themselves later criticising the method for poorly modelling certain scenarios, such as wall-bounded flows (58). Overall, it is the author’s opinion that the technique may already be viable and in use by some organisations, including military CFD units, but that the larger commercial organisations may not have fully capable versions of the software and so are waiting to do so before saturating the market with related information and advertisements. The alternative is that since they have significant vested interests in their current solvers which provide most of their revenue and which they are not ready to move from, they discourage new entrants to the market, who exhibit fast and efficient IBM methods, since the innovators might undermine their monopoly.

The final Octree approach to be mentioned is the ‘Cut-cell’ method. Here, those cells which intersect the boundaries of the geometry are identified and the way in which the cells are intersected is also calculated (see Figure 21). Many methods exist with regards to dealing with the misshapen cells (59) but a common one is ‘Reshaping’; reshaping occurs when the boundary-crossing portions of cells whose centroid lies in the fluid are discarded and for cells whose centroid lies in the solid, the portions residing in the fluid are amalgamated with nearby fluid cells. This cell merging process presents problems in 3D as cell amalgamation algorithms find the creation of random polyhedrals challenging and because fluxes between diagonally adjacent, instead of simply adjacent, cells must also be computed (60). There are also numerous ways for dealing with the discretisation of the governing equations to account for the presence of cut cells and the interpolation of flow variables between cells. Most have so far shown second-order accuracy throughout all domain cells, both at the boundary and far from it, for simulating benchmark validation cases such as flow past a cylinder and wall-driven flow in an inclined box (61) (62).

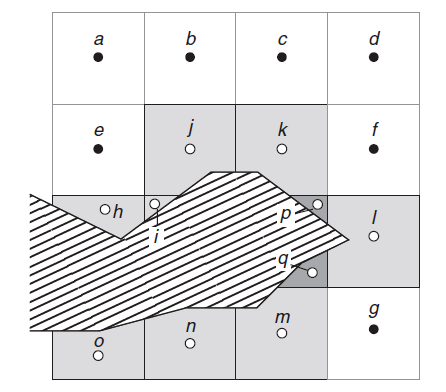


Figure 21: A simple 2D case of an embedded geometry within a fluid domain where the cells in the Cartesian grid cut by the boundary are shaded. One scheme may entail joining the remaining segments of the darkest cell (*p* and *q*), whose centroid is within the solid, to the adjacent cut-cell (*l)* and interpolating between all its faces and those of the surrounding cells to determine flow variables (15).

Though cut-cell is most currently limited to two dimensional scenarios, some applications express the ability to be extended to 3D with relative ease (63), although having to deal with a greater variety of potential cut-plane shapes in three dimensions will add to the complexity of the modelling. There is some evidence that this technique has been used to simulate 3D cases, the first being the modelling of hairpin vortex formation over a hemispherical protuberance due to a laminar incoming boundary layer (64). The second being the extension of a model used first in 2D to simulate cylindrical pipe flow that showed second order accuracy and was later extended to model the heat transfer between concentric spheres where it showed relatively large errors compared to experimental results (maximum error of 13.77%) (65). However, these were relatively simple models and therefore the extrapolation of their results to complex aerodynamic flows is limited. CD-adapco claims to have a 3D ‘Trim cell’ method for modelling flows such as that around military ground vehicles (66) which may well utilise some of these techniques but their website gives very little in the way of technical details. This is most likely because they fear competitor imitation, which might then reduce their competitive edge, and so it is hard to verify whether the method is in widespread use or how it operates.

Finally, for completeness, OpenFOAM’s grid generation process within snappyHexMesh is a primarily Octree based procedure consisting of a staircase-type initial refinement around the geometry, and concluded with a rare method of surface treatment, called ‘Snapping to Surfaces’ by OpenFOAM, where the vertices of the refined cells in closest proximity to the geometry’s surface have their vertices moved onto the surface of the imported .STL file. In the field of grid generation, this is known as the ‘projection method’ (67) and though this technique can often result in low quality elements being produced, the extrusion of a hexahedral boundary layer that usually follows on from the snapping eliminates these cells. This provides a very fast and robust method for meshing complex geometries and a simple 2D case is given in Figure 22 below:

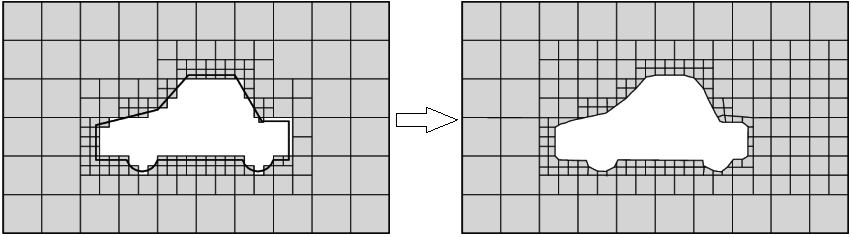


Figure 22: OpenFOAM simple 2D example of a staircase refinement followed by snapping of nearby cell vertices to model edges which results in polyhedral cells. A hexahedral extrusion has not yet been performed here (68).

### 2.5.7 Conclusions of Grid Generation Techniques

The spectrum of research in this field is incredibly large and diverse and though this report aims to give some idea of present developments in this area, it has focused on methods appropriate to the VWT, namely, viscous incompressible flows modelled with the FVM. Scientists and engineers continue to gain ground in utilising newly developed incrementations on existing theories in order to more efficiently and accurately model all manner of physical scenarios. For instance, one method combines the cut-cell and IBM methods and has been used to accurately model fluid flow around a cylinder (69). Alternatively, Fluent uses a combination of the Advancing front and Delaunay methods for tetrahedral volume generation, so that the former creates a desirable point distribution while the latter efficiently triangulates it (70). However, ultimately it will undoubtedly fall to the large CFD vendors that steer market forces to decide where to place money into R&D and which methods to incorporate and promote. In terms of the VWT project, this section has brought about some considerations:

1. Where dirty CAD is used as an input for the meshing process, an effective surface wrapping tool should be employed to greatly reduce cleanup time.
2. While structured meshes may converge upon an accurate solution more quickly than unstructured varieties, the actual volumetric meshing is very time consuming for complex geometries and should only be used where high solution accuracy is critical. Otherwise, unstructured, hybrid, automatic meshing should be employed as it allows for rapid mesh creation with good solution accuracy and so is useful for generating and deciding between early prototypes in a design study, as might be the case when using a VWT in an industrial project.
3. Though Octree-based methods have the potential to greatly reduce the time dedicated to the meshing stage, therefore eliminating the bottleneck here, and have been proven to be accurate for flows relevant to this project, they are still relatively unknown and have seen considerably less investment than other meshing schemes. Therefore, the tools which are most easily available, in this case commercial solvers such as Ansys Fluent, are more useful for the VWT. Firstly, they are well-integrated into existing CAD packages such as SolidWorks. Secondly, they are relatively intuitive to those with a working knowledge of fluid dynamics, and so can produce accurate results. Lastly, they are provided with an extensive customer support base should the user have specific problems.
4. Due to concerns over protecting intellectual property, it is difficult to verify exactly what types of meshing algorithms underpin the software of well-known CFD vendors, unless they are purchased, and therefore making valid comparisons is expensive. So, while researchers create new methods and variations upon old ones to expand the applications and capabilities of this discipline, those schemes that are truly effective are quickly recognised by corporate R&D departments, their rights bought and the schemes incorporated in order to extract revenue from them before many in the CFD community will have the chance to see them. Though this is unfortunate, it is the same technique applied by many large organisations in order to retain market leadership and so the alternative might be to look to open source software such as OpenFOAM, where software codes are freely available and a large community contributes to the program’s development. Though these platforms might not be as initially user-friendly as commercial versions, they are not accompanied by the often heavy price-tag of their marketed counterparts and benefit from a helpful online network that is usually responsive to bugs and specific problems.

## 2.6 Mesh Quality and Evaluation

### 2.6.1 Introduction

This section aims to introduce some of the most commonly applied evaluation metrics when attempting to assess the quality of a mesh. It will be restricted to the main tools that are available within OpenFOAM, Fluent and Pointwise and it aims to give some relevance about how these mesh metrics affect the end solution and where their limitations lie.

### 2.6.2 A Brief Evaluation of the Importance of Mesh Quality

The notion that a ‘bad mesh’ is one which may cause the solver to crash at some stage during the solution process is widely known, and feared, among analysts, especially when a simulation could take weeks to run with some solving schemes. Therefore, measuring the quality of a mesh from this standpoint seems to be a black or white issue; the mesh simply has to allow the solver to run. However, if this was the case, then it is likely that commercial CFD vendors would merely develop functions for providing a binary answer –“yes” or “no” – to the question, “Will the solution run to conclusion?” Yet, this is not the case, as every modern simulation program, both those that are confined to mesh generation and those that include this step as an integrated part of the simulation process, contains a suite of tools claiming to allow simple and clear inspection of various measures pertaining to ‘mesh quality’.

The reason these tools exist is that the degree to which a particular type of element varies from its optimum shape, and how elements are aligned with respect to their neighbours, can influence the outcome of a solution. Mesh quality metrics are often given within a specified range, e.g. 0 – 1, where 0 might equate to a perfect element and 1 would mean ‘degenerate’, or completely incompatible with the solver. A mesh will often then be analysed to determine the distribution of cells’ quality in the mesh using this range. As a cell quality value tends towards the negative end of the spectrum, its detrimental impact on the solution may also increase (71) and so, by discerning how many cells are of a certain quality and where they reside within a mesh, efforts can be made to resolve the issues they might create through grid adaptation.

When applying CFD to a new field, it is imperative that validation studies, in the form of experiments with which the models can be compared, be performed to determine the effect of various properties of the study on the simulation outcome. Therefore, the effect that a given turbulence model or discretisation scheme has on the final solution can be compared to physical results and so the applicability of those schemes, or lack thereof, may be extrapolated to similar problems. From the perspective of mesh generation, it is well known that having a high quality grid is essential in reducing errors, which arise due to the discretisation of the NSE’s across cells, that act to create flow solutions divergent from actual results. Errors are likely to arise where specific cell structures lead to unexpected flow variable interpolation between neighbours and the most common measures for identifying problem cells will be outlined in the following section. When examining the mesh for causes of error, the effects of other sources of error (such as those associated with the iterative solving process; those related to poor mathematical descriptions of the flow; user error through accident or lack of fluid dynamics knowledge) must be controlled for by minimising them as much as possible. Only then is it possible to truly attribute changes in solutions generated to mesh topology modifications.

### 2.6.3 Measures of Mesh Quality

The most important criteria in defining a high quality mesh are: creating an appropriate grid resolution to resolve all the prescient characteristics of the flow in certain regions (e.g. potential areas of flow separation from a body); ensuring that the mesh is created in such a way as to capture all the elements of the input geometry that will have a significant impact on the flow and alignment of cells with flow direction where possible (72). The next most important factors are: firstly, ensuring that the expansion rate (smoothness) of cells from one layer to the next does not exceed 20% (73) as otherwise, this can significantly reduce the order of accuracy of the solver. Secondly, ensuring the aspect ratio of cells – that being the ratio of longest to shortest edge lengths in a cell – is close to 1 where flow is multi-directional (15). This is because the discretised transport equations assume these criteria to be true when solving, however higher aspect ratios are acceptable in regions of developed (laminar) flow as the flow’s uni-directionality means that crossflow terms are very small and so have little impact on the solution (10). It is for this reason that long and thin hex cells are aligned with the flow in boundary layers as the flow here satisfies this requirement.

Of the remaining many possible metrics used to evaluate a mesh, those that have the greatest potential to cause solver instability or reduce rate of convergence are orthogonality and skewness. In Fluent, orthogonal quality is measured as the ratio between the vector connecting a cell centroid to the centre of one of its faces and the face-normal vector for the same face. Alternatively, it is measured between a face-normal vector and the vector connecting the centroid of that cell to that of the cell which lies adjacent to that face (see appendix B, Figure 34) For a given cell, it is the minimum calculated for any of the faces and the range is 0 – 1, where 1 is best and 0 is worst (74). Cell equiangle skew is applicable to all element types and is a measure of how deformed an element is from its ideal, equilateral, shape. The range is from 0 – 1, where 0 denotes a perfect cell and 1 equates to degenerate i.e. cell vertices are coplanar. Highly skewed and non-orthogonal cells are undesirable as they create larger truncation errors in the interpolation of flow variable values between them (71). The equation for skewness is given in the appendices, as are some guidelines in terms of how the value corresponds to the quality of the mesh. Also provided are the equations for aspect ratio, as well as orthogonal quality and some diagrams to help illustrate these concepts.

### 2.6.4 Conclusions

The ability to make *a-priori* assessments of the accuracy of a solution based solely on the quality of a mesh is a sensitive issue within the CFD community. There are those who believe that using a mesh which contains various poorly formed elements poses few problems to the solver and can even lead to reduced discretisation error (75). However, they are usually in the minority and it may well be possible that this only holds true for this specific case, or that errors are cancelling out, resulting in a solution closer to the actual one. Interestingly, this author suggests that a very effective method for decreasing the error caused by deformed cells is to analyse the mesh based on truncation error magnitude (that associated with discretisation of the transport equations) and improve cells that show the largest error. However, this method has yet to be applied to complex PDE’s such as the NSE’s. As has been mentioned, though cell quality is likely to affect the magnitude of errors within the solution, even with a ‘high quality’ mesh, it cannot be known whether a user is converging upon a correct solution or one that is unrealistic without experimental comparison. In this situation, *a-priori* measurements are of limited value until accurate, fast, efficient and widespread solving techniques exist. One work intensive example to be optimised is Direct Numerical Simulation, or DNS (which solve the transport equations directly without discretisation and so provide a ‘true’ solution). Another could be the creation of universally applicable and totally accurate turbulence models.

It is ultimately difficult to make comparisons between meshes based on the quality metrics provided by different companies as, not only do the types of metric they provide differ, but sometimes the formulation of the metric and what constitutes ‘good’ or ‘bad’ differs also. The field of CFD is badly in need of a unanimously agreed upon set of evaluation tools whose impact on flow solution convergence, stability, and potentially even accuracy, has been widely verified. This would allow valid comparisons to be made between the effects of varying the mesh structure upon the solution, possibly even between differing software platforms and turbulence models.

Alternatively, time might be better spent on developing solvers that are both more accurate, due to higher order discretisation of the transport equations, yet stable; therefore ensuring that an imperfect mesh has little impact on a solution so that meshing is no longer seen as a hindrance to project progress and the burden of solution accuracy instead falls squarely on the solvers. For many engineering problems, it is the integral properties such as drag and lift that are important, as opposed to the distributed data in the form of specific velocity contours, and on which the inaccurate results of a few deformed cells will have very little impact. Therefore, with regards to the VWT, quality metrics should be used as a confidence check that the solution will not crash and will reach an acceptable level of convergence in as little time as possible.

## 2.7 Mesh Adaptation

After applying the metrics mentioned in section 2.6 and finding cells, or groups, which appear likely to cause issues for the solver, measures must be taken to improve the local topology to minimise the risk of solution inaccuracy, solver instability or lack of convergence. Mesh adaptation contains two broad categories: mesh movement or repositioning (‘r-adaptation’) and mesh enrichment (‘h-adaptation’) (76). The former aims to reposition vertices without changing the point connectivity, while the latter concentrates on selective coarsening or refinement of grid cells. R-adaptation finds most use in structured grid applications while h-adaptation is most useful in mainly unstructured grids (75). Within Fluent, adaptation can be easily achieved by adjusting the allowable skewness and aspect ratio of cells created by the mesh algorithms through the ‘smoothing’ function within the mesh settings. Algorithms are applied, based on user input, which automatically swap tetrahedral edges within groups of elements and/or move vertex positions in an iterative fashion to achieve the baseline quality criteria. This technique is known as ‘dynamic adaptation’ since the meshing software will attempt to achieve these criteria over the course of a number of user-specified iterations.

Pointwise and OpenFOAM also provide their own functions for mesh improvement which are mainly based on the same concepts as above, albeit with some variation in terms of how much is specifiable through user input and how much is left to automation. Within OpenFOAM, snappyHexMesh contains a range of specific controls which allow a high degree of regulation over the mesh generation, such as ‘maxInternalSkewness’ which allows a user to set a limit on the angle created between a cell’s faces. Pointwise has a more intuitive approach as, for example, its graphical interface permits dragging of cell vertices to more appropriate locations to adjust cells. Also, grid spacings can be individually set for cells clustered around given edges.

Some further common options include: adjusting the sizing functions used to specify the cell sizes around given geometrical features to increase refinement in these areas; reducing the cell growth rate value to achieve a ‘smoother’ grid and, if a very sharp feature is present that would require very intricate cells to resolve it and does not significantly impact the flow, this edge can be filleted within the original CAD model to simplify the meshing process. Also, the elimination of small concave geometrical features can help, since the projected inflation vectors for cell growth may sometimes cross here, leading to the creation of overlapping, degenerate cells. Furthermore, if using a surface-extruded boundary layer, the stage in the meshing process at which this is created, i.e. before or after the tetrahedral cell volume creation, can be specified which can improve the mesh quality. And lastly, ‘feature-based adaptation’ is the term given to the automatic restructuring (e.g. coarsening or refinement) of a mesh, as the solution is processed, to increase the resolution and alignment in areas experiencing high flow gradients. This relatively modern technique can increase the accuracy of the simulation automatically and therefore saves human effort. It should be used where possible in the VWT.

# 3. Methodology

## 3.1 Design Process Methodology

In Stuart Pugh’s 1990 book ‘Total Design: Integrated Methods for Successful Product Engineering’, he outlined the ‘Total Design’ process; a systematic method of producing products which starts with identifying the user or market’s requirement, then concept designs and detailed designs are produced and finally manufacture and market release mark the completion of the project. The technique emphasises iteration between stages so that problem identification leads to quick resolution (77). While the end result of this project was never intended to be a marketable product, the interlinked and iterative process Pugh described is highly important to a group project such as the VWT and has been used throughout.

Specifically, within the Design group, the intent was never to create a new vehicle design but instead to create a relatively realistic CAD geometry, after some aerodynamic optimisation, on which tests could be performed by the rest of the project team. For instance, the HPC group would aim to utilise the CAD model when applying the CFD techniques they had refined and this would be validated against results gained by the Experimental group who were to perform tests upon the 3D-printed model. The project was to investigate how effectively all these elements could be combined in order to create a more modern, integrated, automotive design process to challenge the industry standard. From the standpoint of this individual project, the intention was to aid in the design of the concept car and investigate the importance of meshing within the design process so that recommendations could be made to other group members, and in general, to increase simulation accuracy.

## 3.2 Meshing Research Strategy

Initially the aims of the project were to compare the meshing strategies of Pointwise, OpenFOAM and Fluent in order to determine how each handled dirty CAD geometry and how these issues might be overcome. However, these aims were scaled back as the project progressed, and more was learnt about the complexities of meshing and how *a-priori* mesh checks could not be used solely to verify the accuracy of a model. This action was compounded by witnessing how much work both Hamilton and Bolt were devoting to learning Pointwise and OpenFOAM for their purposes and so becoming accomplished with, and creating multiple accurate simulations in, all of these programs in the given timescale was unrealistic. Instead the project was to focus mainly on developing a sound theoretical basis for making recommendations in terms of best practice grid generation such that simulation errors could be minimised. This was achieved through consulting a range of academic and industrial sources: from grid generation textbooks to papers from international roundtables on mesh generation and also reference material from commercial grid software vendors. There would also be an element of attempting to compare the accuracy of various meshing strategies within Fluent, while varying some parameters, to gain a practical insight into the effects they have on the solution.

## 3.2 Health and Safety

Though the Design sub-group, and this individual project, were not actively involved in any actions or construction that might pose a threat to the members’ health, a health and safety check was nevertheless carried out to conform with good practice and this is provided as appendix C.

# 4. Design and Experimentation

## 4.1 Problem Definition

The Design team’s goals were to produce a CAD concept car. Then, after some refinement, to generate a relatively aerodynamic shape, it would be necessary to have this 3D printed for use within the newly constructed wind tunnel. These experimental results would be used to validate HPC group modelling on the same geometry using the CAD file provided by the Design group. Alongside this, each member of the Design group would be looking at a separate aspect of virtual design in terms of its ability to be integrated into a modern industrial design process.

## 4.2 Preliminary Work

Initially, it was suggested that a scaled model of the Ahmed body (a simplified representation of a modern passenger car which exhibits a similar aerodynamic profile and so is used in experimentation for CFD validation purposes) be created for wind tunnel testing but it was deemed unnecessarily complicated as a simpler model, such as a cylinder in cross flow, could be used for collecting data and validating CFD models instead. In order to produce the final concept car, a basic representation of a car was generated in SolidWorks with easily variable features including: the front and rear windscreen slant angles and the radius of fillets applied to its edges. These three variables were each investigated by a member of the Design team, where the author was responsible for the fillet radii, to determine their effect on the drag coefficient (a numerical indicator of the aerodynamic resistance of a body to a given flow, further described in the author’s I1 report (29)). The model began as a bluff body (fillet radius 0mm) and models were produced with successively larger fillet radii, from 0mm to 6mm in discrete steps of 1.5mm. A new set of coordinate axes was specified at the mid-plane of the model, behind it and at ground level for later use in Fluent. The simplest way of ensuring that this set of axes carry over to the Fluent simulation was to save all CAD models in Parasolid format (.x\_t file extension). While STL files are also compatible, changing the position of the axes’ origin was not as simple and so they were not utilised, though the mathematical description of bodies and surfaces appear similar for both file types once imported into Fluent.

The following simulation settings were a combination of those from a document on best practice guidelines for automotive external aerodynamics in Fluent (78), advice from Hamilton, Bolt and personal experience:

- RANS solver.  
- Using the parallel processing option with 3/4 of the pc's threads.  
- Pressure-based, steady-state solver.  
- Realisable k-epsilon model with non-equilibrium wall functions (NEWF’s).  
- Inlet turbulence intensity ratio = 1%, outlet = 5%.  
- Turbulence viscosity ratio 10% for both inlet and outlet.  
- Coupled solving scheme

- 1st order spacial discretisation for the first 100-200 iterations (with turbulent kinetic energy = 0.8) and then switch to 2nd order (with turbulent KE changed to = 0.95) for remainder.  
- Explicit relaxation factors of 0.25 for momentum and pressure.  
- Hybrid solution initialisation.

To reduce unnecessary calculation time, only half of the model was simulated along its longitudinal axis of symmetry. This model was placed in a domain that extended five car lengths in front, behind, above and out to the side of the vehicle (shown as Figure 35 in appendix B) to dampen the effects of inlet conditions on the boundary layer formed with the ground and to make near-wall effects negligible. Here, the creation of a new set of axes level with the ground in SolidWorks was important as it meant buffer layer (i.e. the domain which acted as a VWT) creation below the body stopped exactly where the stilts on which it stood ended. This ensured that no very small, low quality elements could be formed in a gap that might occur between the stilts and floor – or the wheels and floor in future simulations. The front face of the domain was set as a ‘velocity inlet’, the rear to ‘pressure outlet’, the floor and vehicle faces as ‘no-slip’ and the remaining walls as symmetry planes. An inlet speed of 45m/s was specified as this would be the speed used in the Experimental group’s wind tunnel (79).

Prior to simulating the effects of changing the fillet size, a mesh convergence study was performed on the unfilleted model to identify what level of refinement gave answers independent of the mesh density so that this level of refinement could be applied to future models in the study. After producing an initial unstructured tet mesh, using Fluent’s default settings, with 29,863 elements, specific sizings were attributed to the vehicle’s ‘legs’ and exterior body faces. The size of each of these was gradually decreased, therefore better resolving the areas responsible for generating bodily drag and simultaneously aiming to increase the model’s accuracy. It is noteworthy that as the fillet size increased, the element concentration was seen to be high where many face edges met (see appendix B, Figure 37 and Figure 38). This resolution is not necessary for there are no very small features to resolve therefore, in future, it would be useful to attempt to unify these faces if possible.

A number of variables were recorded as each model was modelled within Fluent’s simulator utility including the average and maximum element skewness within each model, iterations required to reach a continuity convergence level of 1x10-4 (and 1x10-3 for all other variables) and the drag coefficient value for each iteration (see appendix B, Table 6). The results of the mesh convergence study in terms of change in Cd are shown in Figure 23 where the model reaches an acceptable level of convergence after the 9th iteration with a mesh density of 404,183 elements. Henceforth, the variance of Cd between meshes did not exceed 2%, which can be considered an acceptable level of accuracy. The same sizings were then applied to models with fillets of successively larger radii and these were simulated using the same boundary conditions as were utilised in the mesh convergence study.

Figure 24 (and Table 7 in appendix B) provide the results of the analysis which varied the fillets’ radii. The results indicate that increasing fillet radius (starting from a bluff-body shape) decreases the drag coefficient and the decrease is proportional to how large a fillet is in relation to the size of the geometry aspect of which it is a part, which can be seen by the flattening of the curve at larger fillet radii. Velocity profiles on a cut plane of the 0mm and 6mm filleted models can be found in the appendices (Figure 39 andFigure 40).

Figure 23: Change in drag coefficient (Cd) with increasing mesh density showing convergence around 400,000 elements.

Figure 24: Decrease in drag coefficient associated with increasing model edge fillet radius where the curve displays diminishing returns on this decrease as the fillet radius grows proportionate to geometry dimensions. CAD model dimensions were: Length x Width x Height = 8 x 3.17 x 3.01cm.

These results provided good justification for using very rounded edges and sloped surfaces within the final model, reinforced by the results of Crinion and Browne’s studies that demonstrated that shallower front and rear windscreen angles gave rise to lower drag coefficients (80) (81), as intuition would lead one to expect.

In relation to the wider project, this fillet study helped to determine the increment spacing of change in fillet size for an Ahmed body, within Bolt’s surrogate model design study. Here, the similar flattening of the Cd curve was also observed (82).

## 4.3 Meshing Comparison

### 4.3.1 Introduction

To examine the effects of different meshing schemes on solver accuracy, a number of different schemes were employed. A standard Ahmed body was used for validation since it has often been verified in the literature experimentally (83) (84) and the body shape produces aerodynamic effects characteristic of modern passenger cars. A mesh domain was created based on best practices (78) and the information provided in the Ahmed boy experimental study conducted by Meile et.al. (85), which the HPC group were also using for validation of their models. This also allowed for consultation with the HPC group on recommended solver settings to improve accuracy. The purpose was to investigate the performance of a number of commercial meshing algorithms on solution accuracy, while holding solver settings constant. The mesh schemes used included two patch-dependent algorithms (those that relied specifically on the CAD files’ surface descriptions to generate surface and volume meshes) and two patch-independent, or assembly, meshes. The assembly meshes can tolerate dirty CAD since they produce a cut-cell mesh in the volume initially, which makes associations about positions of surfaces based upon where it meets them, and then creates its own triangulated surface mesh independently of the input file. This means it can tolerate gaps and overlapping surfaces, making it much more robust than patch-dependent schemes. The mesh types being considered were: an automatically generated fully tetrahedral patch-dependant mesh, the standard patch-dependant hybrid mesh available within Fluent (using a mixed inflation layer comprised of wedges, pyramids and prisms with a tetrahedral volume fill), Fluent’s Cut-cell (mainly hexahedral) mesh and also its patch independent tetrahedral mesh.

### 4.3.2 Simulation Details

Whilst the majority of the solver settings used were the same as those outlined in the bullet points from section 4.2, only 1st order-upwind schemes were employed in most of the simulations. This was due to these schemes’ known stability, at the cost of some solution accuracy, which would be useful when introducing many varied meshing schemes with varying quality. A mesh convergence study was run for each meshing scheme, where very similar face sizings were applied and reduced in the same way for each type of mesh and a wake region was created around the body, for which the same sizings were used also. The dimensions of the wake region (half a car length in front, above and to the side of the body and a full car length behind) were created based on best practice. For each study there were nine iterations with element counts ranging from 23,000 to 4.2 million. For each mesh in each study, recordings were made of skewness (average and maximum), orthogonal quality (average and minimum), aspect ratio (average and maximum), iterations to convergence (to a level of 0.0001 for continuity and 0.001 for all other variables), simulation time to convergence and drag coefficient. From these results, the percentage change between successive values of Cd and the percentage variance of Cd from the experimental value of 0.299 were also calculated. These tables are all provided within appendix B, Table 8 -Table 11, the graphs summarising their results are given as Figure 25 andFigure 26. Furthermore, some pictures illustrating the meshes created and the resulting flows are given within appendix B, Figure 41 - Figure 49:

Figure 25: Graph showing mesh convergence on value of drag coefficient for four meshing schemes, compared with experimental. PD = Patch-dependent, PI = Patch-independent.

Figure 26: Graph showing convergence times (seconds) against mesh density for four meshing schemes.

At one stage during the PD tetrahedral mesh convergence study, second order interpolation was experimented with to determine its effect on the solution. Using a two million element mesh that had already shown convergence around this mesh density (as can be seen from Figure 25), the interpolation schemes were switched to second-order upwind after 100 iterations. After a further 188 iterations the solution had converged and though there was an initial, expected, jump after switching, the solution of Cd=0.267 was not only was less accurate than the first order scheme, but also took longer to converge in more iterations. Furthermore, it was suspected that this value would continue to fall with increasing mesh density, and this was in part confirmed when experimenting with the cut-cell method. After simulating the default cut-cell mesh, further refinement using the same refinement principles as had been used for the other studies would not produce a convergent solution. It was not an issue of mesh quality as these metrics had remained relatively unchanged. Instead, switching to second-order upwind (seen as the dip in the cut-cell values in Figure 25) provided a method of producing convergence in successive meshes at a cost of decreased accuracy. For this reason, the cut-cell method and second-order upwind schemes were seen as unfavourable choices for further simulations.

Another technique experimented with was the changing of the exterior wall conditions (not including the floor and symmetry plane) to ‘pressure-outlet’ as advised by Bolt (82) due to it having the effect of reducing average drag coefficients, and therefore increasing accuracy, by an average of 7.5% in OpenFOAM. This was implemented for a seemingly stable tetrahedral PD case in Fluent with 1.61 million elements using 1st order interpolation schemes. However, it would not converge, even after 500 iterations where the standard equivalent had converged in 91 iterations, and became quite volatile. The final Cd was around 0.8, almost 2.7 times as large as for the same mesh using symmetry conditions, so evidently the interplay between the solvers, boundary conditions and mesh within Fluent behaves differently, and with less stability, than their equivalents in OpenFOAM. Therefore, this modification was abandoned for all further modelling of aerodynamic flows.

Lastly, upon examination of the PD hybrid mesh using Fluent’s analysis tools, it was noticed that the five layer thick inflation layer was in fact formed from wedges, not hexahedra as assumed (see appendix B, Figure 44). However, analysis of a cut plane of the mesh (appendix B, Figure 46) verified that the elements were composed regularly and in alignment with the flow at the body’s surface and so should still convey the benefits associated with standard inflation layers.

The following matrix analysis was performed to assess the suitability of the meshing schemes previously mentioned, in terms of which could be most appropriate for the concept car analysis, where criteria deemed more important were given a higher weighting than the others to amplify their effects on the final results:

Table 2: Weighted matrix analysis of four meshing schemes against various criteria.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Score (/10) x Weighting** | | | |
| **Criterion** | **Weighting** | **Meaning of Score** | **Tet (PD)** | **Hybrid (PD)** | **Cut-Cell** | **Tet (PI)** |
| Simulation accuracy / no. elements | 2 | High = High accuracy for fewer elements | 16 | 14 | 10 | 18 |
| Convergence time / no. elements | 2 | High = Low convergence time for fewer elements | 12 | 14 | 10 | 16 |
| Ability to resolve geometry / no. elements | 1 | High = Able to resolve geometry with fewer elements | 9 | 7 | 7 | 9 |
| Changes in mesh quality / no. elements | 1 | High = Little degradation of quality with more elements | 8 | 8 | 5 | 9 |
| Ability to easily handle complex or dirty CAD | 2 | High = Dirty CAD poses few issues for mesh scheme | 10 | 10 | 18 | 18 |
| Intuitivity of mesh refinement process | 1 | High = Refinement process easy to determine | 6 | 6 | 5 | 4 |
|  |  | **Total** | **61** | **59** | **55** | **74** |

Simulation accuracy was relatively similar between all three schemes - the patch-dependent tetrahedral and hybrid schemes, and the patch-independent tetrahedral scheme - although for the same number of elements the latter proved to be more accurate at every instance close to convergence. The final PI tetrahedral mesh provided a drag coefficient result 3.73% above the experimental result. The cut-cell method was by far the least accurate of the four tested, potentially because of issues related to the interpolation schemes used between cells when dealing with hanging nodes. Furthermore, the cut-cell method was seen to take the longest time to converge for a similar number of elements as the other schemes, seen by the gradient of the line in Figure 26. The PI tetrahedral method actually took, on average, the least time which makes it an attractive candidate for further use. The ability of all the schemes to resolve geometry, based on a given number of elements, was largely similar. However, since the Ahmed body is a relatively simple shape, further comparisons between the schemes would be useful on more complex geometries. Mesh quality variance, judged using the specified three metrics, was often very weakly correlated with increasing mesh density. At times, mesh quality was lowest during the initial meshes generated which is likely due to the formation of large, irregular elements to resolve complex geometrical features. As element size falls, resolution increases and so elements do not have to form undesirable shapes in an attempt to describe the curves on which they lie. There seemed to be little impact on the solver by the production of cells with adverse quality values, the worst of these being a skewness and orthogonal quality of 0.995 and 0.139 respectively from the cut-cell mesh, and an aspect ratio of 46.618 from the patch dependent hybrid grid. This lends favour to some of the ideas raised in section 2.6.4 and, as long as a solver is robust enough to handle these issues, supports the idea of using automated mesh generation to reduce time wasted on perfecting grids around complex entities. While both the cut-cell and PI tet meshing (assembly) schemes are supposedly able to handle dirty CAD geometry due to their unique mesh generation techniques, this ability was not readily tested in this mesh refinement study and so will require further analysis. Lastly, while the settings of the sizings that changed associated element sizes and mesh densities were very similar for most of the schemes, the PI tet refinement method took much longer to comprehend and so this provided some impediment to progress.

From the results generated and recorded in Table 8 to Table 11 it was possible to make some educated decisions on the suitability of the tested meshing schemes for application to the concept car design. The matrix analysis showed that, based on evaluation by a number of criteria, the PI tet scheme was the most suitable for application to the concept car and so it is this method which was applied and tested further.

## 4.4 Concept Car Meshing and Simulation

Upon deciding on an appropriate meshing scheme, the next stage in the process was to try to simulate the concept car using much the same approach as had been employed previously. The goal would be to perfect a meshing technique for this particular geometry and then perform a mesh convergence study to converge upon a solution most like the result produced by the Experimental group. The first step was to create a CAD representation of the experimental wind tunnel configuration in SolidWorks. Here, the dimensions specified by Blades (86) were used to create the fluid volume equivalent of the space within the wind tunnel, in which the concept car was positioned as it had been for testing. The concept car model (solid) was subtracted from the wind tunnel volume (solid) and this became the effective simulation domain. Figure 27 provides this arrangement:

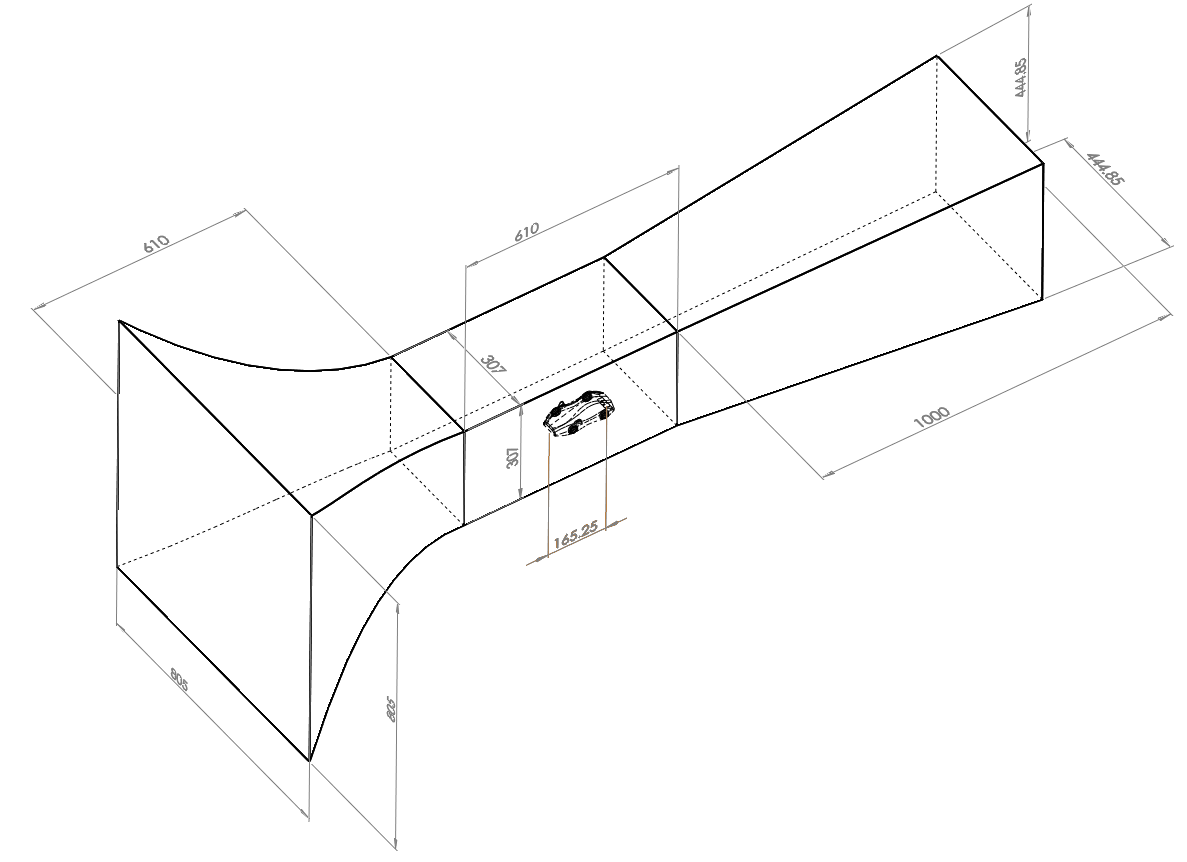


Figure 27: Annotated wireframe view of wind tunnel and concept car set-up in Solidworks, all dimensions in mm.

The part was saved as a Parasolid file and then imported directly into Ansys whereupon the geometry was examined for defects by using the tools within Design Modeler. Several holes were detected (as shown in Figure 28) but a clean-up was not undertaken due to the known time it could take to do so, and the author having little experience with this process, as well as there being the goal of testing the capabilities of the PI tet scheme on these sorts of issues.

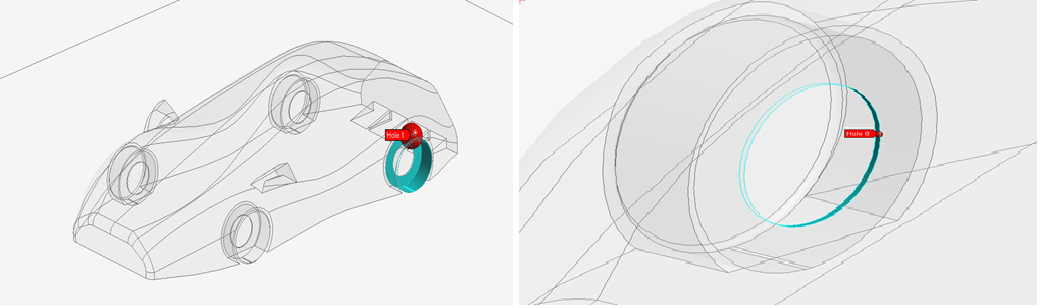


Figure 28: 2/8 of the results from the hole repair search in Design Modeler. All holes occurred around the wheels.

The geometry was then imported into Fluent Meshing, after which the various regions of the domain, such as inlet, outlet, walls and concept car, were defined as named selections. The concept car alone consisted of 79 individual faces which would have undoubtedly caused a great deal of unnecessary refinement on face edges if it had been meshed using the PD tet scheme, as was seen in the earlier fillet study. A number of initial attempts were made to mesh the geometry but which failed due to the interactions between sizing tolerances and the meshing algorithms. Eventually a first successful mesh was created with roughly 600,000 cells. This mesh was evaluated using the known metrics and was found to have cells with skewness as high as 0.999 and orthogonal quality as low as 0.073, where their locations are shown in Figure 29, and which would undoubtedly be classed as degenerate (see appendix B, Table 4 andTable 5).

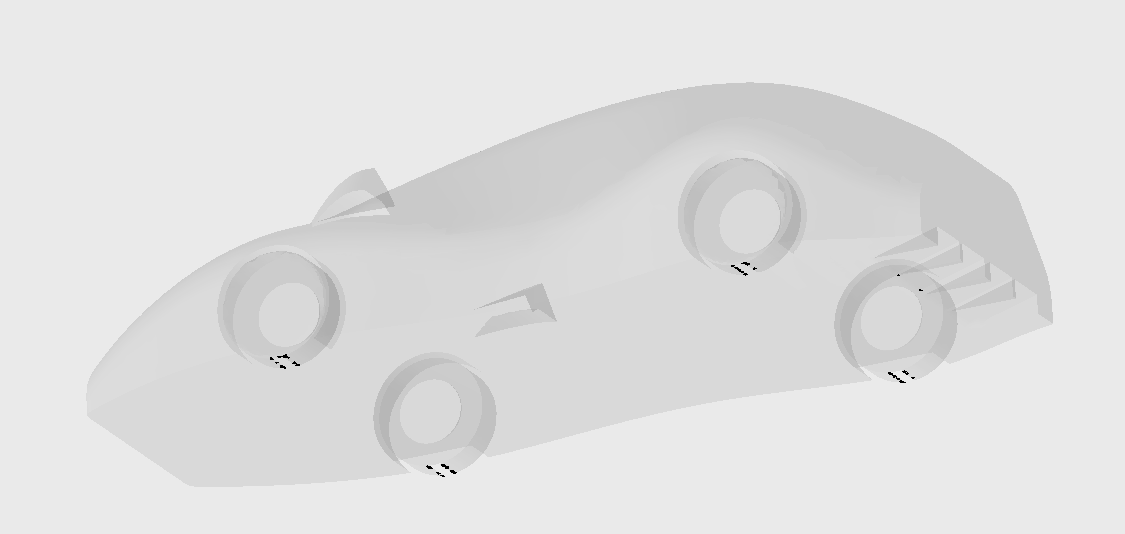


Figure 29: Locations of cells with skewness around 0.999 in initially successful mesh of 600,000 cells. Locations of poor elements coincide with very sharp regions, such as the contact area between the wheel and the ground and the corners of the rear diffuser, as was expected.

Due to experiences with the robustness of Fluent, it was decided that an initial attempt at solving the simulation would be made to see the extent to which the software could deal with poorly formed elements. Using the same solver simulation settings as within the mesh convergence study, a case was set up with a projected area of 0.002933m2, determined by Docherty (87), and a reference length that of the concept car (see Figure 27). It was allowed to run and though it became stable after 200 iterations, it would not converge. Furthermore, the drag coefficient was found to be around 35, almost 100 times bigger than might be expected. After viewing a velocity contour plot for this flow, it was seen that, although the correct velocity had been specified at the inlet, at the point of reaching the car it had increased to almost 260 m/s due to mass flow rate continuity between the inlet of the main inlet and that of the actual test section. Using this same principle, the wind tunnel inlet speed that would create an average flow velocity of 45 m/s at the test section inlet was calculated, using the following equation for volumetric flow rate (*Q*) continuity (88):

[4.4.1]

If *U1* is the mean velocity at the wind tunnel (m/s) inlet, *A1* (m2) is its cross sectional area and *U2* and *A2* are the equivalent values at the test section inlet then this equation can be rearranged to provide the necessary wind tunnel inlet velocity to give the desired test section inlet velocity:

[4.4.2]

This velocity was specified at the inlet for all further calculations. The simulation was then rerun to check the case and proceeded for 300 iterations with 1st order upwind interpolation and though the solution stabilised after around 100 iterations, it still did not reach convergence. The interpolation schemes were then changed to 2nd order as the case had become stable and after a further 600 iterations no convergence was seen. Also, the residuals were less stable than previously and were further from convergence. Because 1st order schemes seemed to entail higher stabilities than their 2nd order equivalents , if only for this style of simulation, all further simulations were undertaken using only 1st order interpolation schemes.

Due to the potentially negative effects of these very low quality cells on the solution, the next stage was to attempt mesh improvement. With the highly automated and specialised mesh generation schemes employed in the PI tet assembly meshing process, little manual input is possible for specific mesh refinement. This restricted the measures that might normally have been taken to improve local areas of the mesh. Ultimately, a combination of setting changes allowed for the creation of a mesh whose quality metrics were borderline acceptable, based upon what had worked in the meshes used in the Ahmed convergence studies. Firstly, the relevance centre (a method for specifying global mesh coarsening or refinement) was set to ‘coarse’, and ‘high’ smoothing (automatic vertex repositioning that increased overall mesh quality but especially so in intricate regions) was specified. Furthermore, face sizing enforcement for sizings on the concept car’s surface were set to ‘soft’ to enable smoothing to function more effectively and minimum element sizes were consecutively decreased. The final mesh had 4.01 million elements, maximum skewness and orthogonal quality of 0.9853 and 17.896 respectively and a minimum orthogonal quality of 0.1177. This mesh was imported into the solver and initiated with the most appropriate settings agreed upon from the preceding concept car and Ahmed body studies. The case eventually converged in 696 iterations, and after 20,865 seconds (5.8 hours), with a drag coefficient of 0.168 (3 s.f.). This compares very favourably with the average drag coefficient calculated by Docherty of 0.17±0.15 (87). Figure 30 andFigure 31 provide a close up representation of the mesh used for the simulation and a visualisation of the fluid path lines over the concept car. Figure 50 andFigure 51 in appendix B show the convergence history of all residuals, as well as the ZY symmetry plane cut view of the entire domain mesh.

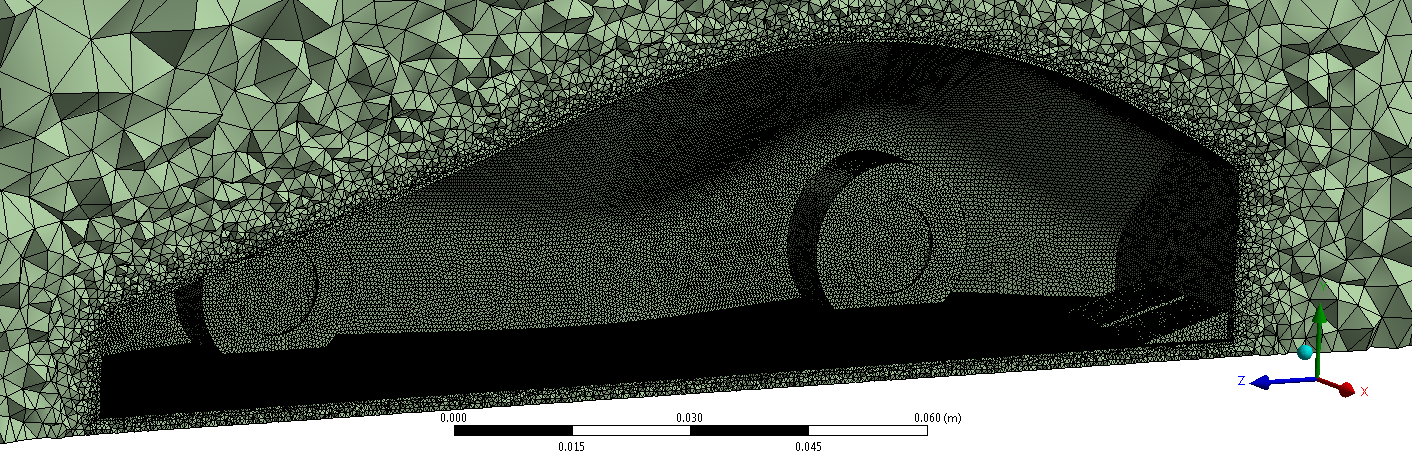


Figure 30: Zoom on ZY symmetry plane cut of mesh showing tetrahedral refinement around body and intricate features of concept car.

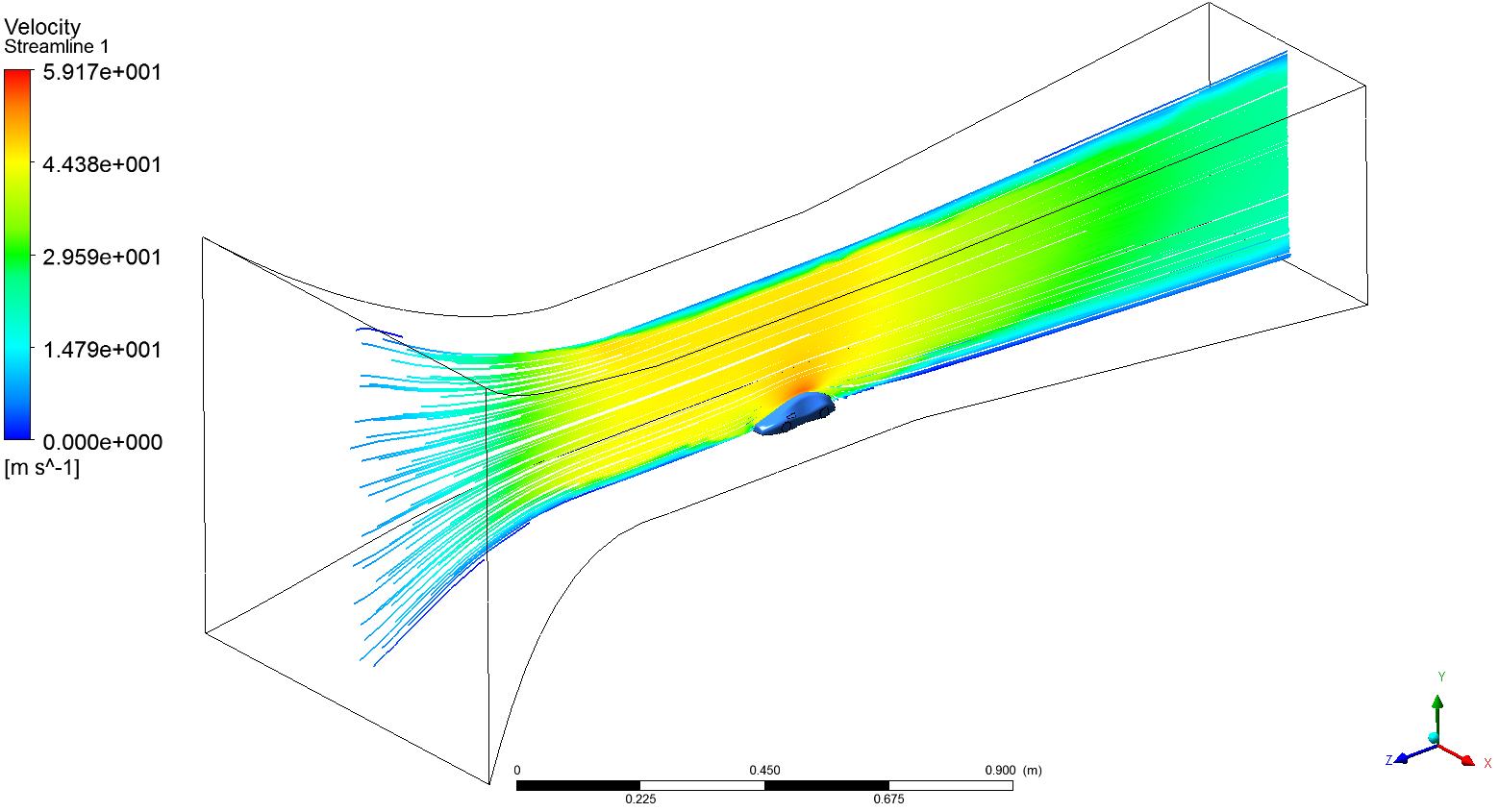


Figure 31: Velocity pathline plot, on wind tunnel ZY symmetry plane, coloured by fluid velocity. Note correct inlet velocity of roughly 45m/s at test section inlet.

# 5. Project Analysis

## 5.1 Analysis of Design Process

With regards to the early fillet radius variance study, the validity of the results may be called into question as there were no experimental bases for comparison purposes since the wind tunnel had not been completed in order to verify results. Also, the simulation methods undertaken by the three Design group members may have differed due to inconsistent CFD knowledge. However, common sense would dictate that the results would be what were witnessed and the results of the fillet study were in part reinforced by Bolt’s more complex, further simulations that produced similar results (82).

The Ahmed body meshing study generally utilised valid modelling techniques and the results were compared to well-documented experimental results. However, comparisons made between the cut-cell and remaining techniques may be less valid since the only way to ensure convergence was to implement 2nd order interpolation schemes, where the other methods had used 1st order. The cause of instability when attempting to produce solutions for the cut-cell method was unknown but may be due to sensitivity to orthogonal quality, as this dropped below 0.30 when convergence became an issue whereas all the other measures of quality had increased (see appendix B, Table 10). Though wall functions in close proximity to the concept car were utilised instead of distant symmetry conditions (in the Ahmed study), overall, the Ahmed study results provided a good foundation for drawing conclusions for which may be the most appropriate meshing strategy, since the boundary conditions were largely similar.

In relation to the concept car, the application of flow rate continuity to replicate the test section inlet’s velocity was an accurate assumption since the physical wind tunnel makes use of the same assumptions in order to do the same. This means that if the remaining solver settings were valid, the flow conditions arising could be expected to be largely similar to those in the experiment which may partially account for the very accurate result. However, the drag coefficient generated by the experimental group took into account the rolling resistance of the (bearing) wheels of another, similar car but not the actual concept car since it had been accidentally destroyed by the cleaners. While the wheels of the similar car spun quite freely, those of the concept car did not (but did roll very smoothly) and so this could have led to an increase in the actual drag coefficient. Another factor that might impact the simulation accuracy may be the production of large truncation errors through the very skewed cells seen in the mesh. In future, instead of trying to edit mesh controls to improve quality, it may be much easier to modify the CAD geometry to, for example, increase the contact area between the wheels and road and so eliminate the production of almost coplanar cells in regions like this. This would be a realistic modification since an actual car’s wheels would sag slightly under its own weight and so would not meet the road at a perfect tangent. One further point to note was that y+values (those that specify the recommended cell centroid distance from a boundary in order for the NEWF’s to effectively model the near-wall region) were not specifically taken into account, during this investigation. This was due to difficulties in controlling the resolution of the surface mesh elements in each mesh type. However, the average distance of centroid of a boundary layer cell for the final concept car simulation mesh was shown to be roughly 0.0002m (see appendix B, ), this being the optimum length to correspond to a y+ value of 30, as recommended by Hamilton (89). In PI tetrahedral meshing it was discovered that this is likely controlled by the ‘Proximity min. size’ function and so the relation to the desired y+ value lends good accuracy to the final result.

More generally, it would be useful if Fluent gave readouts for the total time used to mesh a geometry as this often took a significant amount of time and so would be helpful for comparing meshing schemes in terms of suitability for a project. Also, the behaviour of 1st and 2nd order schemes was in part analysed and showed that 1st order were in general more favourable as they led to faster converging and more accurate results (for the specific flow conditions modelled). However, the cut-cell technique differed as it required 2nd order interpolation (after 100 initial 1st order iterations) for it to converge and so it would be useful to experiment with better quality meshes with this scheme to determine if this affects which interpolation procedure could be used.

## 5.2 Project Management Analysis

The I1 report provided what, in hindsight, was a very optimistic Gantt chart. This plan was based upon the assumption that the field of meshing was not as complex as it has been found to be and so learning how to use various pieces of software to implement different schemes would be fairly straightforward. As time has gone on, however, and a great deal has been learned about the intricacies of this branch of CFD, the goals had to be revised to reflect more realistic targets. This was accompanied by the realisation that simulation accuracy cannot be predicted simply by mesh quality metrics and so some simulation was required for validation of meshing strategies. Another assumption was that the importation of CAD models into CFD software would be a major step when in fact this is relatively simple (given the right file type), and the major impediments to project progress are actually the cleanup and mesh generation stages. In relation to the latter, developing a “reliable, but fairly basic, protocol for…meshing” (29) severely underestimated the need for mesh generation to be problem specific, rather than a ‘one size fits all’ solution. Although automated CAD cleanup and unstructured mesh generation are simplifying the challenge, the task is far from trivial. At the time of writing the I1 report, little was known of the many approaches towards meshing but this report has defined a number of the most widely used and also explored their capabilities within the scope of the VWT project.

An updated Gantt chart was made after revisiting and revising the initial goals and this is given as appendix D. In terms of specific project management, the group were initially using a system where chair and secretary were rotated weekly, as suggested by the project supervisor, in order for everyone to experience these roles. Half-way through the second term, the author realised that although this had been somewhat useful, it was not contributing well towards overall project organisation with respect to the group’s goals. Therefore, it was suggested that one member be in charge of chairing meetings and coordinating information flow between members and the author took on this role. This involved organsing the online filing system for increased functionality and usability, overseeing weekly meetings and coordinating group activities through social media and in person. Although this helped the group to be more organised to some extent, this action should have been undertaken from the beginning for it to have been most effective, but this realisation only came upon reflection. Overall, the first lesson learned was that a driven and well organised leader is needed from the start of a project to best realise its goals and that secondly, the magnitude of a project can sometimes be vastly underestimated which means that a great deal of background research should be undertaken in the required field before setting its goals. The problem with this, however, is that it is not always possible to do so, given the time restrictions of a project, and so an even compromise must be made. This is especially so within CFD as there is a vast amount to learn within a variety of fields before any kind of valid model can be created. Unfortunately it is often taken for granted that the ease of use, and point-and-click nature, of the software and interfaces entails that setting up a case is just as straightforward, leading to inaccurate results and wasted time and resources. A layman-style handbook for setting up common cases using a high quality mesh and appropriate solver settings would be indispensable, but would undoubtedly undermine professionals in the field who might lose out due to the reduced reliance on CFD-dedicated businesses for these analytical tasks.

## 5.2 Important Findings and Implications for Group Project

After performing mesh generation on a relatively complex geometry, it is evident that it may take weeks/ months to generate a high quality structured mesh on a full automotive geometry (for instance, including the engine block and all its components in the engine bay) and so explains why industry CFD analysts may favour tetrahedral-based schemes. Given more time, it would have been useful to become more adept with the specifics of each meshing scheme mentioned in the study so more control could have been had over mesh quality, and potentially solution stability, convergence rate and accuracy. When determining which type of mesh generation technique to use, there are two main options. The first is to use a (patch-independent) fast-forming, automatically-generated mesh that is likely to contain very low quality cells around intricate features and so produce less stable or lower accuracy results, which then requires manipulation of mesh sizing controls to improve it. The second is to use a more manual-based editing (patch-dependent) approach where the user must spend a long time understanding the given geometry before painstakingly creating a high quality mesh which may greatly improve the behaviour of the solver, if the solver’s settings are valid. An effective compromise might be to ensure the design and CFD departments are more highly integrated. This would ensure that designs handed from one department to the other have already had small/ unnecessary features removed or coarsened so that larger elements can resolve them and that surfaces are joined correctly. This would be relatively trivial for the design team who are already adept with CAD software and this simple step could save significant time and effort in the life of an average project.

Other important findings included that, for the types of flows investigated, PI tetrahedral mesh generation proved the fastest, most robust and most accurate technique compared to the other schemes and so should be used in automotive simulations where possible, especially when there is not enough time for CAD clean-up. Furthermore, this meshing strategy and the solver settings utilised, compared well with both experimental results and more complex (and work-intensive) structured mesh-based strategies implemented by the HPC group. This can be expected since it was performed using high quality, and expensive, commercial software but also lends credence to the idea that automated unstructured mesh generation can produce results that compare favourably to structured meshes at a lower human-labour expense. This is important in the automotive sector where aerodynamic simulation result accuracy may be less critical than, for instance, military applications and so this time saving may contribute to market leadership.

In hindsight, it would have been much more productive if every group member had initially had the kind of CFD understanding gained by those who devoted their projects to it. However, as this took a long time to develop, a more realistic solution might have been for members to know more about other members’ work and so could have volunteered relevant information that might have reduced time spent researching by non-specialised individuals. Also, in future, it would be most efficient for the group to consist of a number of individuals already specialised in relevant fields, which could have reduced crossover of work-functions between members of the group and the potentially inaccurate results that may have arisen through tasks being carried out by those with inadequate knowledge of the subject.

# 6. Sustainability

## 6.1 Economic Project Considerations

Any organisation that is able to move away from having to perform experiments in order to validate their simulations and simply perform computational modelling due to the accuracy of their models stands to save a considerable portion of their design and R&D budget. The building of physical models that must be adjusted, sometimes in significant ways, is by no means inexpensive. This issue is exacerbated when precision is paramount due to the increased machining skill required to meet tight tolerances. Take, for instance, the example of wind-tunnel testing where a full scale automotive testing facility may cost £30m to build (89). Even if it is rented the costs will be high, especially when factoring in the cost of producing a prototype (which is significant in itself as the model has not been mass produced and so does not entail the price savings associated with this), costs of vehicle transport to the testing facilities and many other charges. Compare this to the few clicks of a mouse at a computer that it takes to make changes to a CAD model and the savings in man and machine hours, as well as money are obvious. Furthermore, the amount of data that can be collected from physical testing is limited to the number (and cost) of sensors placed in the wind tunnel. On the other hand, the amount of data that a simulation can provide, and at what scale, is limited merely by the domain size and grid resolution (and potentially computational resources). In addition, a simulation could also model multi-physics simulations, such as the combined effects of a crash and wet weather on the structural stability and electrical conductivity of sections of a vehicle design – something near-impossible to replicate through physical testing.

## 6.2 Socio-Environmental Impacts

As mentioned in section 6.1, physical testing facilities are expensive to build, but also their creation takes a toll on the environment in the form of the resources, such as steel and concrete, that must be mined, refined and transported to construct them, in addition to the energy requirement associated with fabricating the structures themselves and the harmful emissions released through all these processes. Though CFD facilities also entail resources to build the hardware and energy to power them, this is almost negligible in comparison to the energy required to build a wind tunnel complex. Whilst there are limitations to the accuracy of CFD simulations due to incomplete turbulence models, these models are being continually refined and revised by researchers. Also, in future, machine learning algorithms with access to vast published journal archives may be able to find links between seemingly unrelated fields of work that could spawn a unified, fully accurate turbulence model or mesh assessment tools for assessing simulation accuracy *a-priori*, and much more besides (90). On the other hand, the production of computer hardware involves the use of rare earth metals (91), itself a controversial topic due to the often forced labour of women and children to do so by rebel forces in less economically developed countries (LEDC’s) such as the DRC (92). The trade-off is not something the author cares to finalise upon. From a wider perspective, although the power requirements may be high, supercomputers are able to simulate whole weather systems and so increases in the accuracy of these large-scale, multi-phase fluid dynamic models may help to predict the effects of climate change and make informed decisions on policy and spending decisions to reduce its negative impacts. These models would be impossible to recreate any other way and so investment in research should be a top priority as the negative implications of a volatile climate in the future would likely present humanity with challenges on a scale it is increasingly unprepared to face.

## 6.3 Life-Cycle Analysis

The life-cycle of computer hardware is fairly easily to predict; due to the modular design of desktop (and higher capability) machines, upgrading is relatively straightforward and so where parts are no longer useful for a given application, they may be recycled into another machine with less demanding requirements or even given to a charity which builds computers for users in LEDC’s (93). Also, due to modern Waste Electrical and Electronic Equipment (WEEE) guidelines, manufacturers must ensure the products they build are, at least in part, recyclable and the UK government stipulates regulations on how these products are handled and recycled at the end of their usable lives (94). The matter of software is less simple; there are no issues with disposal and recycling here but the life of modern CFD software is far from trouble-free for the user. There are a small number of large vendors who supply to most of the market and who have therefore monopolised it through aggressively acquiring intellectual property and marketing it effectively. Due to lack of competition, prices for software packages are often very high; as much as £15,000 for a single commercial license (96) before even considering annual licensing fees, training and hardware costs. For many, this makes it a prohibitive outlay and so the author sees the future of CFD moving toward a greater variety of open source options such as OpenFOAM and more so when these programs are built to function on widely used operating systems with user-friendly graphical interfaces.

# 7. Conclusions and Future Work

## 7.1 Project Conclusions

This project set out to aid in the creation of a concept car for use within the VWT project, and also to explore the role of mesh generation in an integrated, modern, automotive industrial design process. What resulted was a comprehensive literature survey that examined every stage in the process of generating a mesh for a CFD application. It described a number of common structured and unstructured meshing schemes, including some state-of-the-art processes that are yet to see widespread validation. This section concluded that unstructured meshes, due to the complex CAD designs utilised but the automotive industry and geometric flexibility this technique affords, are the most suitable. Also, though Octree approaches hold much promise in terms of automation of high quality meshes, there is much validation work to be done before they will be widely accepted for commercial applications. This report also evaluated common mesh quality measures used and resolved that the two most important considerations when generating a mesh are correct resolution to capture important geometrical features or areas with high flow gradients, and alignment of cells with anticipate flow where possible. Less important are metric measures such as skewness, aspect ratio and orthogonal quality, although the production of degenerate cells (as defined by these metrics) can be a good indicator of potential solution instability. Therefore, where possible measures should be taken to improve areas of the mesh where degenerate cells are observed. Furthermore, it is difficult to make comparisons between the apparent qualities of meshes produced through different software platforms as definitions of mesh quality differ between software vendors. Therefore, a set of universally agreed upon quality evaluation metrics would be very beneficial to the field. In relation to this, investment in solver stability with regards to mesh quality could ensure that the burden of work is shifted from mesh generation to solver activity which could dramatically reduce the human-labour input for such a project.

A number of studies were undertaken to test the theory outlined and the results support the statements made earlier about mesh quality being relatively disconnected from solver stability, potentially because Fluent likely contains a very stable solver. The major study undertaken compared the effects of implementing four different meshing strategies (patch-dependent tetrahedral and hybrid schemes as well as patch-independent tetrahedral and cut-cell schemes) on solution accuracy for a test case well validated by experiments in the literature. This was performed while holding solver settings almost constant between schemes and found that the patch-independent tetrahedral method was the most robust meshing strategy and led to the fastest and most stable solution convergence, with the final mesh producing a result 3.73% greater than the experimental value. Conversely, the cut-cell method was found to perform the worst by these measures at almost every instance in its mesh convergence study. A matrix analysis with a range of relevant criteria was applied to assess the suitability of each scheme for use on the concept car and the patch-independent tetrahedral scheme was found to be the most appropriate.

An iterative solving process was utilised to produce a final mesh that converged in 696 iterations and took 20,865 seconds to do so. The drag coefficient produced was found to be in very good agreement with the identical experimental results recorded in the wind tunnel built by the Experimental group; 0.167 compared to the physical result of 0.17±0.15. This provides good evidence for the use of this mesh generation technique within the automotive industry when exploring similar flows. Lastly, aspects of the design process and overall group project management were qualitatively assessed and some conclusions about the sustainability of the process and of its results were drawn.

Overall, the use of physical wind tunnel testing cannot be excluded from a typical commercial design process as turbulence models are not yet accurate enough to stand by themselves without experimental validation. The use of CFD is altogether a very useful process for whittling down a large number of potential designs, based on their aerodynamic behaviour and the requirements of the project, and back up intuitive assumptions, as long as there is valid experimental data to reinforce the major assumptions.

## 7.2 Future Work

To explore more thoroughly the results of this project, a number of studies should be undertaken. These might include the exploration of patch-independent meshes and their modification on solution accuracy. For instance, combining the tetrahedral and cut-cell schemes with inflation layers to determine this step’s effect on mesh quality and solution accuracy. Also, the ability of the different meshing schemes to capture intricate features, and changes in mesh quality, when employing different input files (e.g. STEP, Parasolid and STL) for the same geometry. To further validate the final result of the project it would be helpful to create a number of meshes with similar element counts and quality to the patch-dependent equivalent and run a mesh convergence study. This could also be used to assess the mesh density at which the change in solution no longer varies by a significant result, where potentially a more coarse mesh might suffice for a given application, therefore reducing computational expense. Lastly, experiments could be conducted to specifically measure the impact of 2nd order interpolation schemes on solution convergence rate and accuracy for a variety of mesh types and resolutions.

# References

1. **W.H.Hul, K.Xu.** *Computational Fluid Dynamics Based on the Unified Coordinates.* Beijing : Science Press Beijing, 2012. pp. 104-107.

2. **Y.A.Cengel, J.M.Cimbala.** *Fluid Mechanics: Fundamentals and Applications.* New York : McGraw-Hill, 2006. pp. 818-826.

3. **Z.Xie.** *Applications of CFD: Curve/ Surface Modelling & Grid Generation.* Southampton : The University of Southampton, 2010.

4. **C.T.Shaw.** *Using Computational fluid Dynamics.* Hemel Hempstead : Prentice Hall, 1992. pp. 44-58.

5. **T.Cebeci, F. Kafyeke, J.P.Shao, E. Laurendeau.** *Computational Fluid Dynamics for Engineers.* Long Beach : Horizons Publishing, 2005. pp. 157-164.

6. **R.Lohner.** *Applied Computational Fluid Dynamics Techniques: An Introduction based on Finite Element Methoods.* Chichester : John wiley & Sons Ltd, 2008. pp. 1-6.

7. **S.Lee.** *Encyclopedia of Chemical Processing.* 1. New York : Marcel Dekker, 2005. p. 511.

8. **L.Hamilton.** *I1 Individual Report.* Exeter : The University of Exeter, 2013. p. 7.

9. **H.Bolt.** *Individual Report - I1.* Exeter : The University of Exeter, 2013.

10. **A.Bakker.** *Computational Fluid Dynamics: Course Materials and Lectures.* [Online] 2006. [Cited: 15 03, 2014.] http://www.bakker.org/dartmouth06/engs150/.

11. **R.Eymard, T.Gallouet, R.Herbin.** *Finite Volume Methods.* Marseille : Handbook of Numerical Analysis, 2003. Vol. 7, 1, pp. 713-1020.

12. **H.K.Versteeg, W.Malasekera.** 2 *An Introduction to Computational Fluid Dynamics: The Finite Volume Method.* Harlow : Pearson Education Limited, 2007. pp. 115-281.

13. —. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method.* 2. Harlow : Pearson Education Limited, 2007. pp. 18-19.

14. **A.Yakhot, T.Anor, H.Liu, N.Nikitin.** *Direct numerical simulation of turbulent flow around a wall-mounted cube: spatio-temporal evolution of large-scale vortices.* Moscow : Journal of Fluid Mechanics, 2006. Vol. 566, 1, pp. 1-9.

15. **C.Hirsch.** *Numerical Computation of Internal and External Flows.* 2. Oxford : Butterworth-Heinemann, 2007. pp. 251-277.

16. **H.K.Versteeg, W.Malasekera.** *An Introduction to Computational Fluid Dynamics: The Finite Volume Method.* Harlow : Pearson Education Limited, 2007. pp. 304-315.

17. **K.C.Karki, S.V.Patankar.** *Calculation Procedure for Viscous Incompressible Flows in Complex Geometries.* s.l. : Numerical Heat Transfer, 1988. Vol. 14, pp. 295-307.

18. **J.H.Ferziger, M.Peric.** *Computational Methods for Fluid Dynamics.* New York : Springer-Verlag, 2001. Vol. 3.

19. **E.Kultajev, C.Benoit, S.Peron, A.Lerat.** *Improvement of a Bidimensional Overset Structured Mesh Generation and Adaptation Method, Based on a Near-Body/ Off-Body Partitioning.* Paris : Arts et Metiers ParisTech, 2011.

20. **G.Zagaris, M.T.Cambell, D.J.Bodony, E.Shaffer, M.D.Brandyberry.** *A Toolkit for Parallel Overset Grid Assembly Targeting Large-Scale Moving Body Aerodynamic Simulations.* Urbana-Champaign : University of Illinois, 2010.

21. **J.F.Remacle, F.Henrotte, T.C.Baudouin, C.Geuzaine, E.Bechet, T.Mouton, E.Marchandise.** *A Frontal Delauney Quad Mesh Generator Using the L norm.* Paris : International Meshing Roundtable, 2011.

22. **R.V.Garimella, J.Kim, M.Berndt.** *Polyhedral Mesh Generation and Optimisation for Non-manifold Domains.* Orlando : International Meshing Roundtable, 2013.

23. **Z.Kang, C.Yan.** *Accurate and Robust CFD Algorithms Applied to Arbitrary Polyhedral Grids.* Beijing : Procedia Engineering, 2012. Vol. 31, 1, pp. 9-15.

24. **K.Lipnikov.** *Mimetic ﬁnite difference method for solving PDEs on polygonal and polyhedral meshes.* [Online] Los Alamos National Laboratory, 2010. [Cited: 04 04, 2014.] http://www-dimat.unipv.it/3indampv/SLIDES/lipnikov.pdf.

25. —. *On Shape-regularity of Polyhedral Meshes for Solving PDEs.* Orlando : International Meshing Roundtable, 2013.

26. **T.Cebeci, F. Kafyeke, J.P.Shao, E. Laurendeau.** *Computational Fluid Dynamics for Engineers.* Long Beach : Horizons Publishing, 2005. pp. 263-294.

27. **D.Martineau, J.Gould, J.Papper.** *Towards an Efficient Distributed Geometry for Parallel Mesh Generation.* Orlando : International Meshing Roundtable, 2013.

28. **R.Lohner.** *A 2nd Generation Parallel Advancing Front Grid Generator.* San Jose : International Meshing Roundtable, 2012.

29. **D.Nima.** *Individual Report - I1.* Exeter : The University of Exeter, 2013.

30. **L.Daniel.** *Simulation and Modeling Techniques for Signal Integrity and Electromagnetic Interference on High Frequency Electronic Systems.* Berkeley, Ph.D. thesis : The University of California, Berkeley, 2003.

31. **S.Owen.** *An Introduction to Mesh Generation Algorithms.* [Online] 2005. [Cited: 04 05, 2014.] http://www.imr.sandia.gov/14imr/.

32. **M.Kremer, D.Bommes, L.Kobbelt.** *OpenVolumeMesh - A Versatile Index-Based Data Structure for 3D Polytopal Complexes.* San Jose : International Meshing Roundtable, 2012.

33. **V.Dyedov, N.Ray, D.Einstein, X.Jiao, T.J.Tautges.** *AHF: Array-based Half-Facet Data Structure for Mixed-Dimensional and Non-manifold Meshes.* Orlando : International Meshing Roundtabe, 2013.

34. **J.F.Thompson, B.K.Soni, N.P.Weatherill.** *Handbook of Grid Generation.* New York : CRC Press LLC, 1999. pp. 418-437.

35. —. *Handbook of Grid Generation.* New York : CRC Press LLC, 1999. pp. 525-543.

36. **E.Marchandise, J.Remacle, C.Geuzaine.** *Quality Surface Meshing Using Discrete Parameterisations.* Paris : International Mesing Roundtable, 2011.

37. **DEStech Publications, Inc.** *Parallel Computational Fluid Dynamics: Recent Advances and Future Directions.* Lancaster : DEStech Publications, Inc., 2010. pp. 44-47.

38. **P.J.Waterman.** *Closing the CAD to CAE Gap.* [Online] 2014. [Cited: 08 04, 2014.] http://www.deskeng.com/de/closing-the-cad-to-cae-gap/.

39. **SolidWorks.** *Computational Fluid Dynamics (CFD).* [Online] 2014. [Cited: 08 04, 2014.] http://www.solidworks.co.uk/sw/products/simulation/computational-fluid-dynamics.htm.

40. **Ansys .** *Fast, Robust, Automatic Mesh Creation for CFD.* [Online] 2014. [Cited: 08 04, 2014.] http://www.ansys.com/Products/ANSYS+15.0+Release+Highlights/Fluids/Fast,+Robust,+Automatic+Mesh+Creation+for+CFD+15-0.

41. **Ansys.** *Meshing Controls: Geometry- and Mesh-Based Defeaturing.* [Online] 2014. [Cited: 08 04, 2014.] http://www.ansys.com/Products/Workflow+Technology/ANSYS+Workbench+Platform/ANSYS+Meshing/Features/Meshing+Controls:+Geometry-+and+Mesh-Based+Defeaturing.

42. **Pointwise.** *Fault-Tolerant Meshing Eliminates CAD Healing.* [Online] 2014. [Cited: 08 04, 2014.] http://www.pointwise.com/pw/cad.shtml.

43. —. *Pointwise is a Quantum Jump Forward in Meshing.* [Online] 2012. [Cited: 08 04, 2014.] http://www.pointwise.com/theconnector/November-2012/Quantum-Jump-Forward-in-Meshing.shtml.

44. **J.F.Thompson, B.K.Soni, N.P.Weatherill.** *Handbook of Grid Generation.* New York : CRC Press LLC, 1999. pp. 109-124.

45. **Pointwise.** *Mesh Confidently Using Pointwise for CFD.* [Online] 2014. [Cited: 10 04, 2014.] http://www.pointwise.com/pw/.

46. **J.F.Thompson, Z.U.A.Warsi, C.W.Mastin.** *Numerical grid generation: foundations and applications.* Ann Arbor : North-Holland, 1985.

47. **J.F.Shepherd.** *Topologic and Geometric Constraint-Based Hexahedral Mesh Generation.* Salt Lake City, Ph.D Thesis : The University of Utah, 2007.

48. **J.F.Thompson, B.K.Soni, N.P.Weatherill.** *Handbook of Grid Generation.* New York : CRC Press LLC, 1999. pp. 461-471.

49. **S.J.Owen.** *A Survey of Unstructured Mesh Generation Technology.* Dearborn : International Meshing Rondtable, 1998.

50. **A.Kuzmin.** *Computational Fluid Dynamics 2010: Proceedings of the Sixth International Conference on Computational Fluid Dynamics.* St. Petersburg : Springer, 2010. pp. 599-606.

51. **J.Tu, G.Yeoh, C.Liu.** *Computational Fluid Dynamics: A Practical Approach.* 2nd. Oxford : Butterworth-Heinemann, 2012. pp. 364-367.

52. **R.Mittal, G.Iaccarino.** *Immersed Boundary Methods.* Washington : Annual Review of Fluid Mechanics, 2005. Vol. 37, pp. 239-261.

53. **W.P.Breugem.** *Lecture 1: A first introduction to Immersed Boundary Methods.* [Online] 2010. [Cited: 11 04, 2014.] http://www.flow.kth.se/sites/flow.kth.se/files/slides\_ibm.pdf.

54. **Ansys.** *The Immersed Boundary Approach to Fluid Flow Simulation.* [Online] 2009. [Cited: 11 04, 2014.] http://www.cadit.com.sg/imagestore/userfiles/image/industry/Automotive/AA-V3-I2-The-Immersed-Boundary-Approach.pdf.

55. **R.Mittal, H.Dong, M.Bozkurrtas, F.M.Najjar, A.Vargas, A.vonLeobbecke.** *A Versatile Sharp Interface Immersed Boundary Method for Incompressible Flows with Complex Boundaries.* Washington : Journal of Computational Physics, 2008. Vol. 227, pp. 4825-4852.

56. **B.Khalinghi, S.Jindal, J.P.Johnson, K.H.Chen, G.Iaccarino.** *Validation of the Immersed Boundary CFD Approach for Complex Aerodynamic Flows.* s.l. : Lecture Notes in Applied and Computational Mechanics, 2009. Vol. 41, pp. 21-38.

57. **T.T.Myers, C.Liang, R.Mittal, F.M.Najjar, A.vonLeobbecke, J.Craley.** *Free Fall Analysis and Simulation Tool (FAST).* Seattle : 20th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar, 2009.

58. **J.Chawner.** *Immersed Boundary Methods Take It On The Chin.* [Online] 2012. [Cited: 11 04, 2014.] http://blog.pointwise.com/2012/10/26/this-week-in-cfd-75/.

59. **D.M.Ingram, D.MCauson, C.G.Mingham.** *Developments in Cartesian Cut Cell Methods.* Manchester : Mathematics and Computers in Simulation, 2003. Vol. 61, pp. 561-572.

60. **H.Bandringa.** *Immersed Boundary Methods.* Groningen, Master Thesis in Applied Mathematics : The University of Groningen, 2010. pp. 33-37.

61. **P.G.Tucker, Z.Pan.** *A Cartesian cut cell method for incompressible viscous flow.* Warwick : Applied Mathematical Modelling, 1999. Vol. 24, pp. 591-606.

62. **X.L.Luo, K.B.Lei, Z.L.Gu, S.Wang, K.Kase.** *A Novel Cartesian Cut Cell Solver for Incompressible Viscous Flow in Irregular Domains.* Xi'an Jiaotong : International Journal for Numerical Methods in Fluids, 2011. Vol. 67, 2, pp. 289-313.

63. **M.W.Johnson.** *A Novel Cartesian Cut Cell Approach.* Liverpool : Computer & Fluids, 2013. Vol. 79, pp. 105-119.

64. **J.Dietiker.** *Implementation of Cartesian Cut-Cell Technique into the Multiphase Flow solver MFIX.* [Online] 2009. [Cited: 12 04, 2014.] http://www.netl.doe.gov/publications/proceedings/09/mfs/12%20%20J%20Dietiker%2042209.pdf.

65. **X.L.Luo, K.B.Lei, Z.L.Gu, S.Wang, K.Kase.** *A Three-Dimensional Cartesian Cut Cell Method for Incompressible Viscous Flow with Irregular Domains.* Xi'an Jiaotong : International Journal for Numerical Methods in Fluids, 2012. Vol. 69, pp. 1939-1959.

66. **CD-adapco.** *Reduce geometry preparation and meshing time from months to hours.* [Online] 2014. [Cited: 10 04, 2014.] http://goo.gl/XfyUwQ.

67. **J.F.Thompson, B.K.Soni, N.P.Weatherill.** *Handbook of Grid Generation.* New York : CRC Press LLC, 1999. pp. 580-591.

68. **OpenFOAM.** *5.4 Mesh generation with the snappyHexMesh utility.* [Online] 2014. [Cited: 13 04, 2014.] http://www.openfoam.org/docs/user/snappyHexMesh.php.

69. **O.Botella, Y.Cheny.** *On the Treatment of complex Geometries in a Cartesian Grid Flow Solver with the Level Set Method.* Delft : European Conference on Computational Fluid Dynamics, 2006.

70. **Ansys .** *Meshing Methods: Tetrahedral.* [Online] 2014. [Cited: 13 04, 2014.] http://www.ansys.com/Products/Workflow+Technology/ANSYS+Workbench+Platform/ANSYS+Meshing/Features/Meshing+Methods:+Tetrahedral.

71. **F.Juretic.** *Error Analysis in Finite Volume CFD.* London, Ph.D Thesis : Imperial College London, 2004.

72. **Pointwise.** *Accuracy, Convergence and Mesh Quality.* [Online] 2012. [Cited: 15 04, 2014.] http://www.pointwise.com/theconnector/May-2012/Mesh-Quality.shtml.

73. **T.J.Craft.** *Trust and Quality in CFD.* [Online] 2012. [Cited: 15 04, 2014.] http://cfd.mace.manchester.ac.uk/twiki/pub/Main/TimCraftNotes\_All\_Access/cfd2-trust-quality.pdf.

74. **Ansys.** *Meshing User's Guide.* [Software Help] s.l. : Ansys, 2014.

75. **C.J.Roy.** *Strategies for Driving Mesh Adaptation in CFD.* Orlando : American Institute of Aeronautics and Astronautics, 2009.

76. **R.Lohner.** *Applied Computational Fluid Dynamics Techniques: An Introduction based on Finite Element Methoods.* Chichester : John wiley & Sons Ltd, 2008. pp. 278-286.

77. **S.Pugh.** *Total Design: Integrated Methods for Successful Product Engineering.* New Jersey : Addison-Wesley Educational Publishers Inc, 1990.

78. **Fluent Deutchland.** *Best Practice Guidelines for Handling Automotive External Aerodynamics with FLUENT.* Darmstadt : Fluent Deutschland, 2005.

79. **D.Docherty.** *Individual Report - I1.* Exeter : The University of Exeter, 2013.

80. **F.Browne.** *Individual Report - I2.* Exeter : The University of Exeter, 2014.

81. **E.Crinion.** *Individual Report - I2.* Exeter : The University of Exeter, 2014.

82. **H.Bolt.** *Individual Report - I2.* Exeter : The University of Exeter, 2014.

83. **S.R.Ahmed, G.Ramm, G.Faltin.** *Some Salient Features of the Time-Averaged Ground Vehicle Wake.* Braunschweig : SAE, 1984.

84. **H.Lienhart, C.Stoots, S.Becker.** *Flow and Turbulence Structures in the Wake of a Simplified Car Model (Ahmed Modell).* s.l. : Notes on Numerical Fluid Mechanics, 2002. Vol. 77, pp. 323-330.

85. **W.Meile, G.Brenn, A.Reppenhagen, B.Lechner, A.Fuchs.** *Experiments and Numerical Simulations on the Aerodynamics of the Ahmed Body.* Graz : CFD letters, 2011. Vol. 3, 1, pp. 32-39.

86. **L.Blades.** *Individual Report - I2.* Exeter : The University of Exeter, 2014.

87. **D.Docherty.** *Individual Report - I2.* Exeter : The University of Exeter, 2014.

88. **EngineeringToolbox.** *Equation for continuity.* [Online] 2014. [Cited: 28 04, 2014.] http://www.engineeringtoolbox.com/equation-continuity-d\_180.html.

89. **K.Varbanov.** *Inside wind tunnels.* [Online] 2012. [Cited: 18 04, 2014.] http://f1framework.blogspot.co.uk/2012/12/inside-wind-tunnels.html.

90. **S.Arbesman.** *Arc 1.3 / Afterparty Overdrive.* s.l. : Reed Business Information, Ltd., 2012. pp. 9-13.

91. **Namibia Rare Earths Inc.** *How Are Rare Earths Used?* [Online] 2014. [Cited: 18 04, 2014.] http://www.namibiarareearths.com/rare-earths-industry.asp.

92. **D.Snow.** *DR Congo: Cursed by its natural wealth.* [Online] 2013. [Cited: 18 04, 2014.] http://www.bbc.co.uk/news/magazine-24396390.

93. **Computers for Charities.** *Working Laptops & PC's Needed. Donate your redundant Computer & IT equipment to benefit Charities across the UK and abroad.* [Online] 2014. [Cited: 18 04, 2014.] http://www.computersforcharities.org/donate.

94. **Health and Safety Executive.** *Waste Electrical and Electronic Equipment recycling (WEEE).* [Online] 2014. [Cited: 18 04, 2014.] http://www.hse.gov.uk/waste/waste-electrical.htm.

95. **sharc.** *Harpoon Key benefits.* [Online] 2014. [Cited: 10 04, 2014.] http://www.sharc.co.uk/html/key\_benefits.htm.

96. **OpenFOAM.** *Chapter 5 Mesh generation and conversion.* [Online] 2014. [Cited: 10 04, 2014.] http://www.openfoam.org/docs/user/mesh.php.

97. **Ansys.** *Computational Fluid Dynamics (CFD) Software.* [Online] 2014. [Cited: 10 04, 2014.] http://www.ansys.com/Products/Simulation+Technology/Fluid+Dynamics.

98. **Numeca International.** *AutoMesh™: The first comprehensive grid generation toolset covering all range of applications.* [Online] 2014. [Cited: 10 04, 2014.] http://www.numeca.be/en/products/automeshtm.

99. **Cambridge Flow Solutions.** *BOXERMesh.* [Online] 2014. [Cited: 10 04, 2014.] http://www.cambridgeflowsolutions.com/en/products/boxer-mesh/.

100. **XYZ Scientific Applications, Inc.** *A Mesh Generator and Pre-Processor for FEA and CFD Analysis.* [Online] 2014. [Cited: 10 04, 2014.] http://www.truegrid.com/INDEX.HTML.

101. **csimsoft.** *Trelis CFD™ - Computational Fluid Dynamics Pre-processor.* [Online] 2014. [Cited: 10 04, 2014.] http://www.csimsoft.com/trelis-cfd.jsp.

102. **KARALIT.** *KARALIT FAQ.* [Online] 2014. [Cited: 11 04, 2014.] http://www.karalit.com/index.php?option=com\_content&view=article&id=70&Itemid=42.

103. **Ansys.** *Introduction to Meshing: 14.5 Release.* Cecil Township : Ansys, 2012.

# Appendix A – Further Information

Table 3: Table listing leading providers or CFD meshing, and in some cases solving, software upon time of writing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Company Name** | **CFD Relevant Product(s)** | **Primary Mesh Generation Methods** | **Type** | **More Information (sources)** |
| Pointwise | Pointwise/ Gridgen | - Multi-block structured  - Semi-structured hex & tet hybrid | Commercial | (45) |
| Sharc | Harpoon | - Automatic body-fitted Cartesian | Commercial | (95) |
| OpenFOAM | blockMesh & snappyHexMesh | - Automatic hex, split-hex and polyhedral | Non-Commercial | (96) |
| Ansys | GAMBIT and ICEM | - Unstructured tet  - Hybrid tet, hex, prism & pyramid  - (Semi)-auto multi-block un/structured hex | Commercial | (97) |
| Numeca International | AutoMesh | - Automatic full-hex structured multi-block  - Unstructured full- hex  - Hybrid unstructured tet, hex, prism & pyramid | Commercial | (98) |
| Cambridge Flow Solutions | BOXERMesh | - Auto-unstructured hybrid tet, hex, prism & pyramid | Commercial | (99) |
| XYZ Scientific Applications Inc. | TrueGrid | - Multi-block structured hex | Commercial | (100) |
| CD-adapco | STAR-CCM+ | - Full semi-structured polyhedral  - Hybrid hex, tet, prism & pyramid  - (Auto) octree  Unstructured tet and polyhdral | Commercial | (66) |
| Csim- soft | Trelis CFD | - Unstructured full (hex or tet) and hybrid (hex, tet, prisms & pyramids) | Commercial | (101) |
| KARALIT | Direct CFD | - Fast Cartesian IBM-based meshing and solving | Commercial | (102) |

# Appendix B - Additional Details from Background Sections

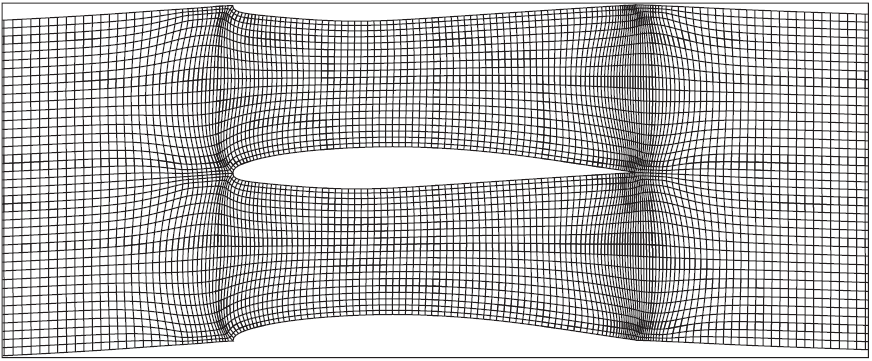
°

Figure 32: H-type, structured, curvilinear grid around a 2D aerofoil (15).

## Equations of Mesh Quality Metrics

### Aspect Ratio (Applies to All Cells)

It is a measure of the stretching of a cell computed as the ratio of the maximum value to minimum value of any of the following distances: normal distance from cell to face centroid and distance between any of the nodes and the cell centroid. Figure 33 gives a simple example for a hexahedral but the equation holds for any polyhedral:



Figure 33: Illustration and general equation for calculating the aspect ratio of a cell, in this case a hex, between the lines joining its centroid to a node and the centroid to that of a cell face (74).

### Equiangular Skewness (Applies to All Cells)

In general, skewness is defined as the maximum value of either of the following (74):

Where:

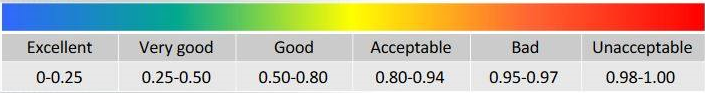
is the largest angle in the face or cell.

is the smallest angle in the face or cell.

is the angle for an equiangular face or cell (i.e. 60° for a triangle, 90° for a quad etc).

For an ideal pyramid or wedge (skewness = zero), ideal equates to all triangular and quadrilateral faces that are equiangular and equilateral. The following table provides a reference for the skewness metric quality standards:

Table 4: Table indicating qualitative meanings of varying skewness metric values, where “Excellent” means equilateral and “unacceptable” means degenerate (103).



### Orthogonal Quality (Applies to All Cells)

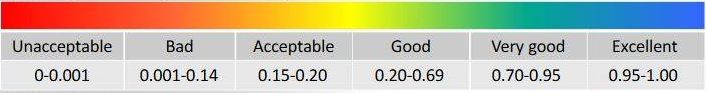
The orthogonal quality is calculated using the following qualities for each face, *i*:

Which calculates a value using the dot product of the face vector () and the vector from the centroid of the cell to the centroid of that face ().

Alternately:

Which calculates a value based on the dot product of the face vector () and the vector from the centroid of that cell to that of the cell which shares that face (). The minimum resulting figure from either or is defined as the orthogonal quality for that cell, within a range of 0 – 1, where 0 is the worst and 1 is the best. The following table better describes this spectrum. Also, Figure 34 illustrates the vectors used throughout this explanation:

Table 5: Quality spectrum for orthogonal quality of mesh elements, where "Unacceptable" means degenerate and "Excellent" means perfectly orthogonal (103).



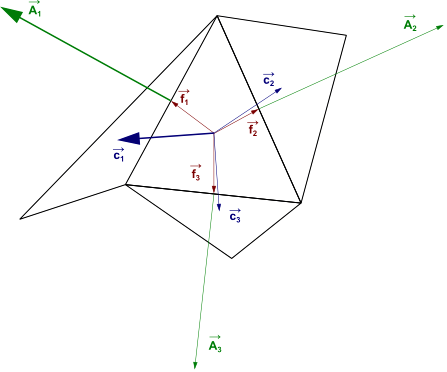


Figure 34: Diagram depicting relevant vectors used to calculate orthogonal quality, in this case for a tetrahedral (74).

## Fillet Study Information



Figure 35: Wireframe visualisation of unfilleted model within domain, showing symmetrical cut-plane of model on near-facing wall and buffer wall distances to minimise near-wall effects.

Table 6: Mesh creation specifics and simulation statistics for basic, unfilleted geometry mesh convergence study.

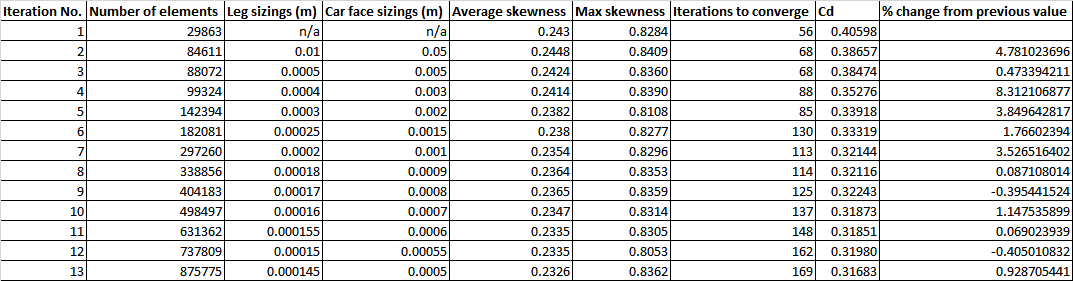
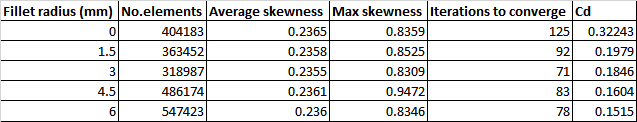


Table 7: Result of fillet study, displaying changing Cd and other metrics for each fillet radius value.



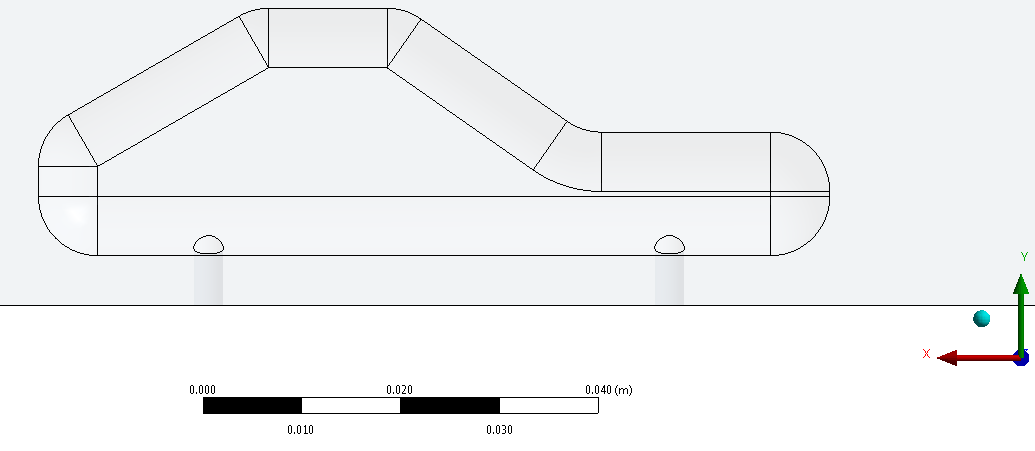


Figure 36: Side-on view of 6mm filleted car model, within Fluent, demonstrating very thin face at right of model created by encroachment of filleted edges into originally flat side of car .

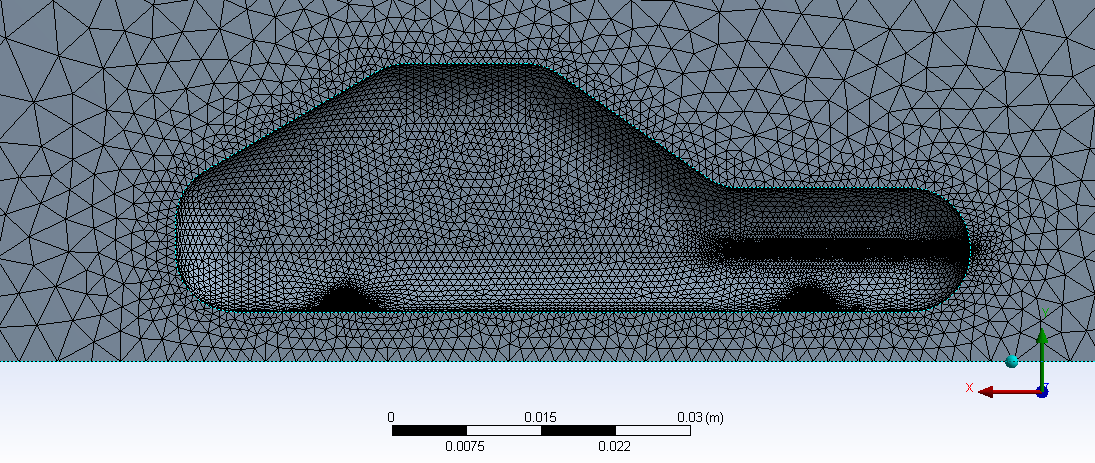


Figure 37: Corresponding mesh for 6mm fillet CAD model, notice concentration of elements at front of car, caused by interaction between very thin model face and applied sizing there.

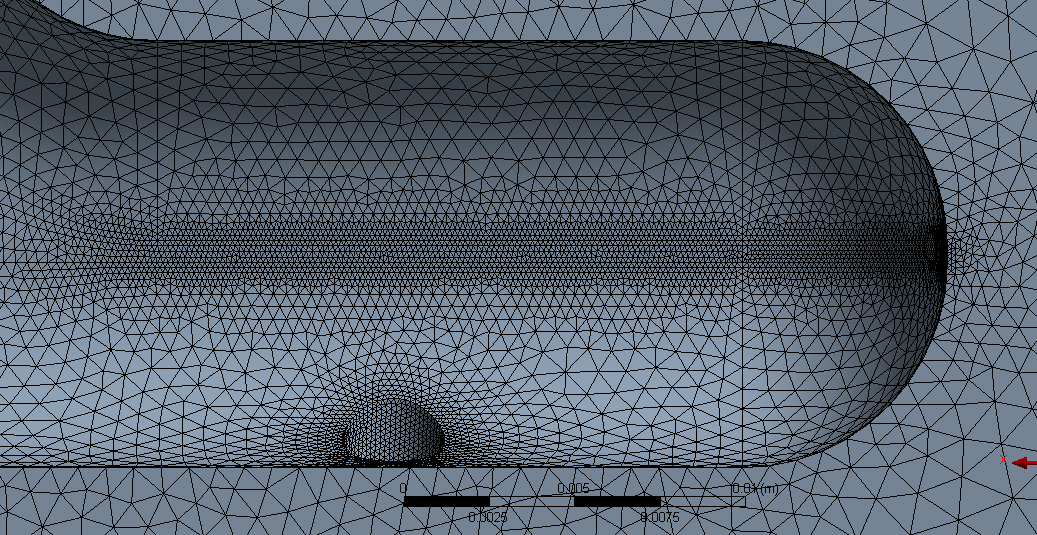


Figure 38: Close-up of aforementioned area showing denser tetrahedral element distribution in area where face edges meet in more detail and around mouth of stilt. Due to the effective lengthening of the stilts where they meet faces at a higher point, the small element sizings here and the concentration in the thin-face section, the element count for the 4.5mm and 6mm models significantly higher than the rest.

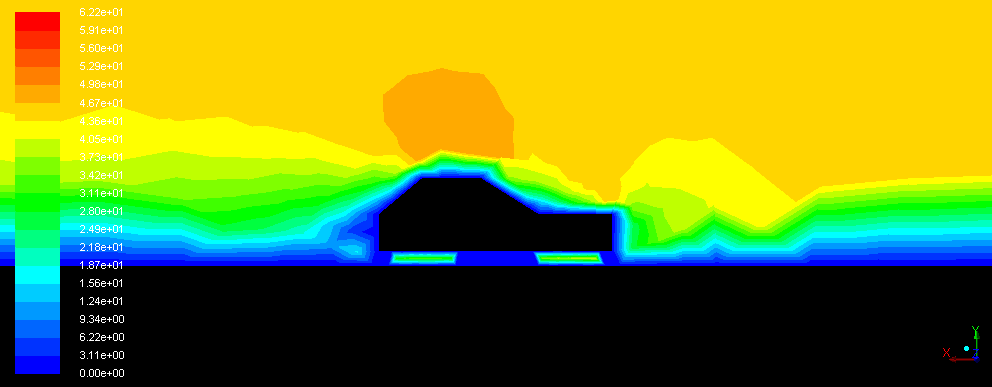


Figure 39: Velocity contours (m/s) on initial, bluff-body, model shown on a plane that does not bisect the stilts.

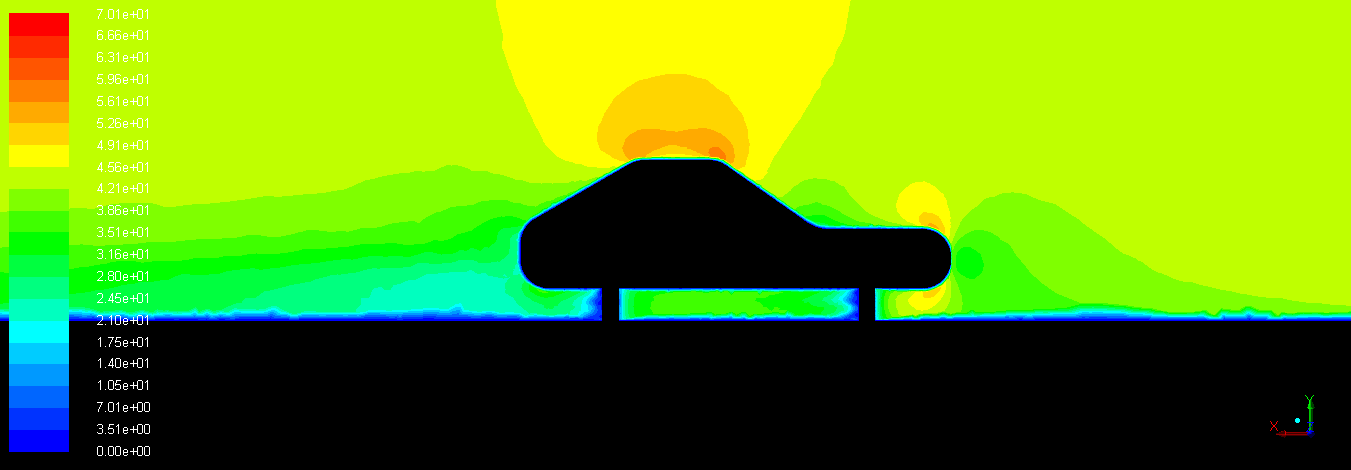


Figure 40: Velocity contours (m/s) on 6mm fillet model shown on a plane that bisects the stilts.

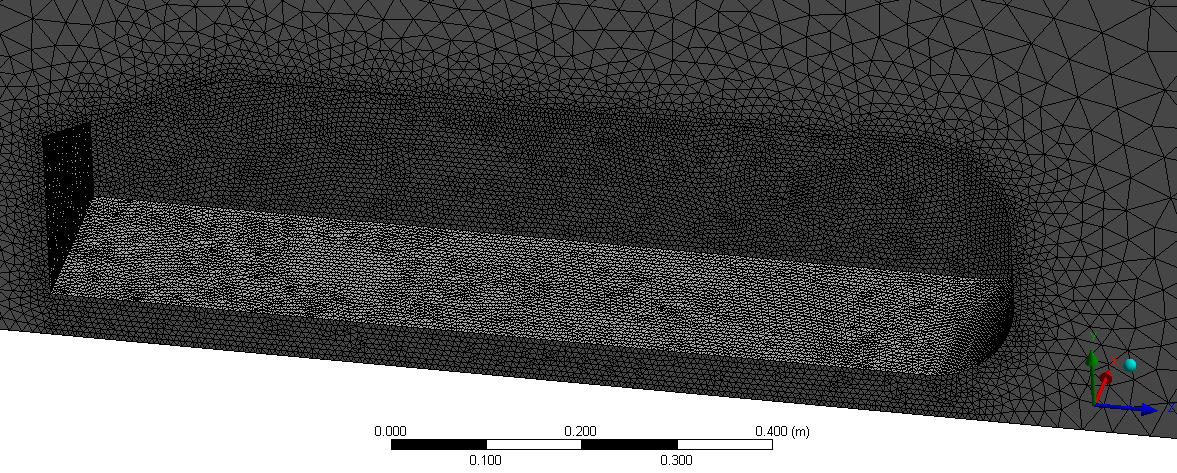


Figure 41: Intermediate PD tetrahedral mesh with symmetry plane isometric view, 1,003,046 elements, no wake regions specified.

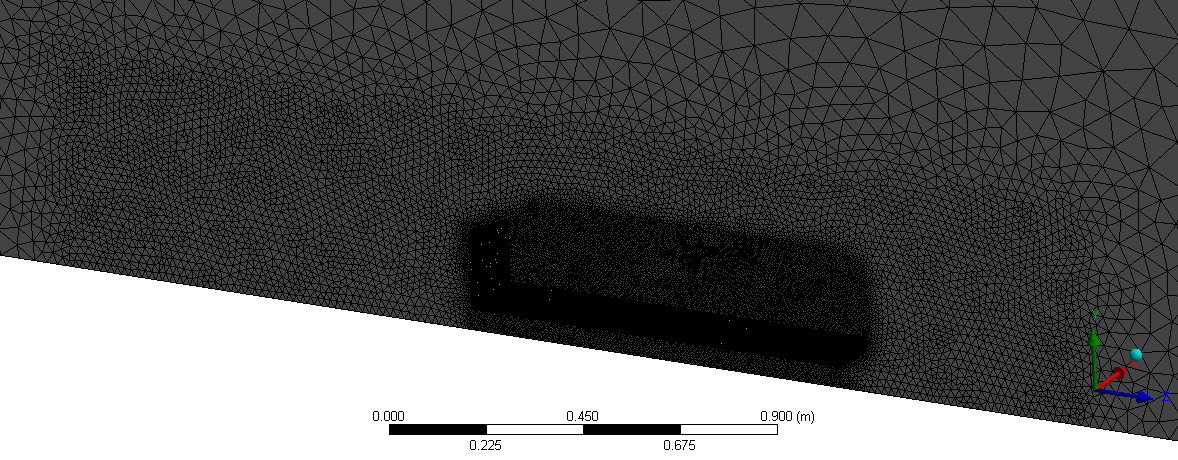


Figure 42: Local refinement using box to capture wake around Ahmed body using a PD tetrahderal mesh (2million elements).

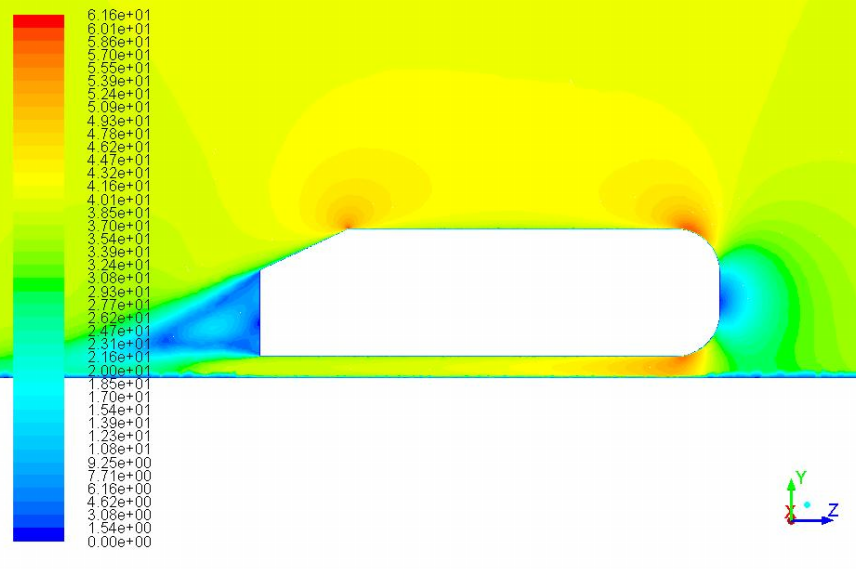


Figure 43: Velocity contours (m/s) on final, 4 million element mesh in PD tetrahedral convergence study.

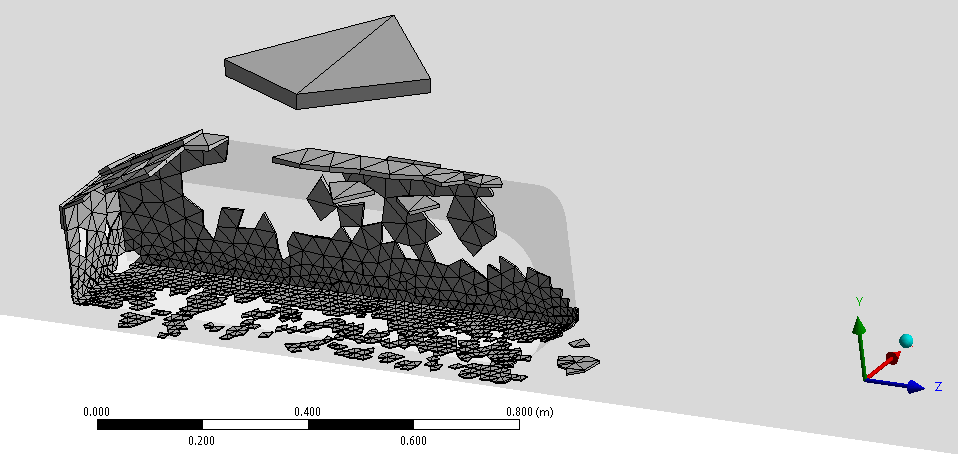


Figure 44: View of cells in hybrid grid with aspect ratio of roughly 12.5, note that the inflation region is comprised of wedges, not hexahedra, as hexahedral mesh generation is a more involved and less automatic process unless using cut-cell technique.

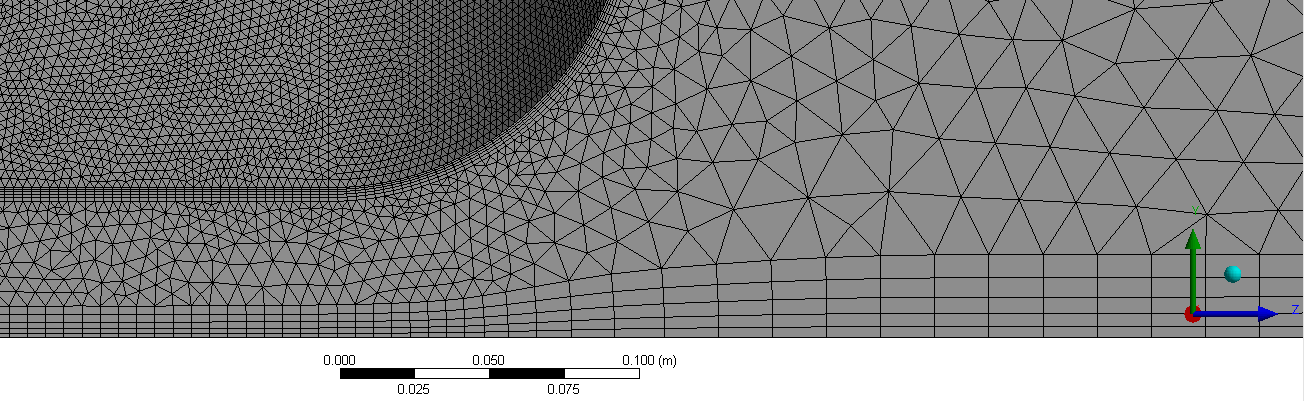


Figure 45: Close-up of lower left corner of 4.2m element PD hybrid mesh, viewed from YZ plane.

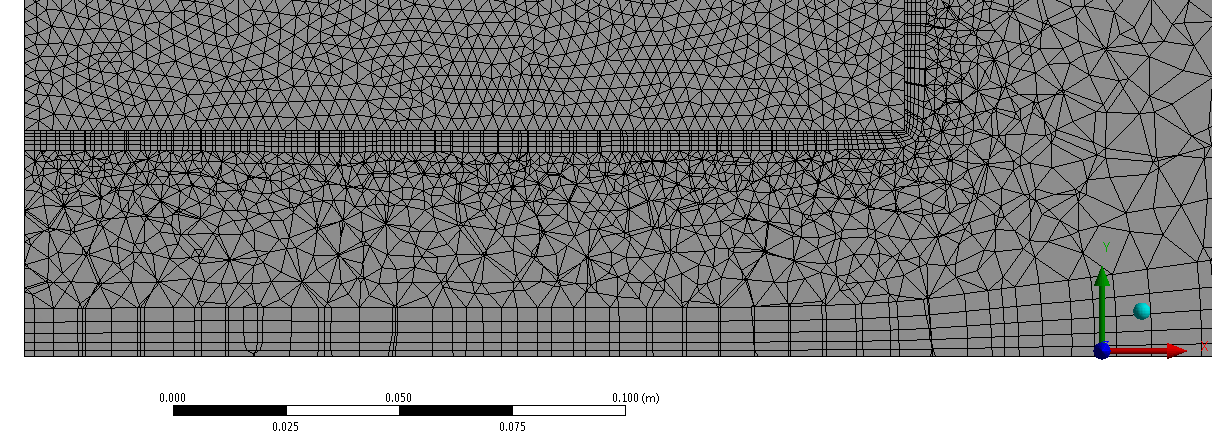


Figure 46: YX cut plane on same mesh, mid-way through Ahmed body. It shows regular inflation layers aligned with flow on Ahmed surface, floor surface and the tetrahedral volume between them.

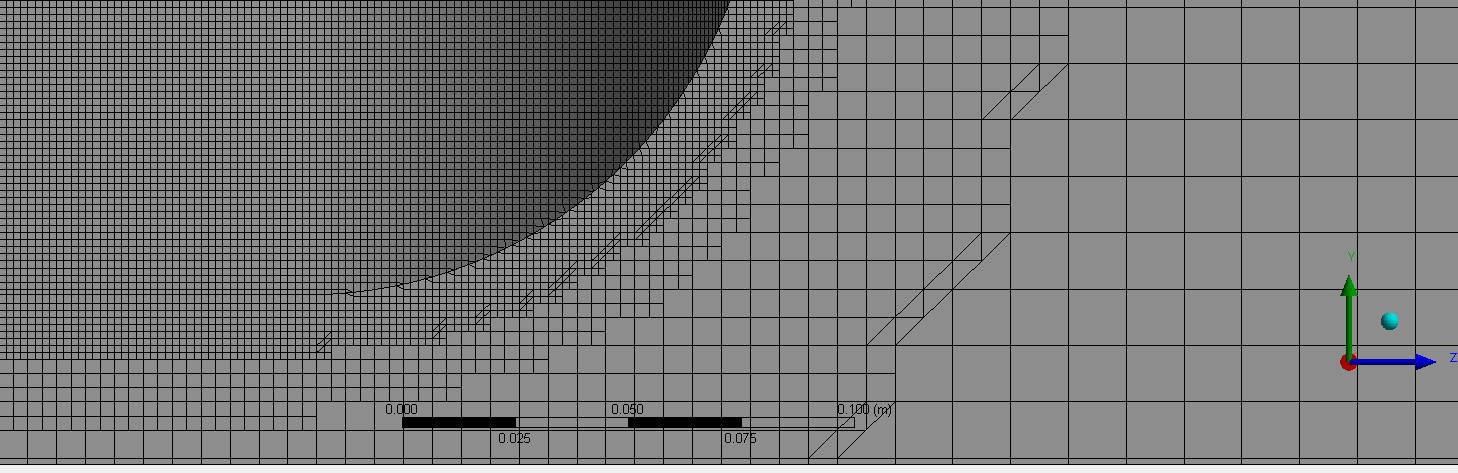


Figure 47: Close-up of refinement in lower right hand corner curved section of 3.27m element cut-cell mesh viewed from YZ plane.

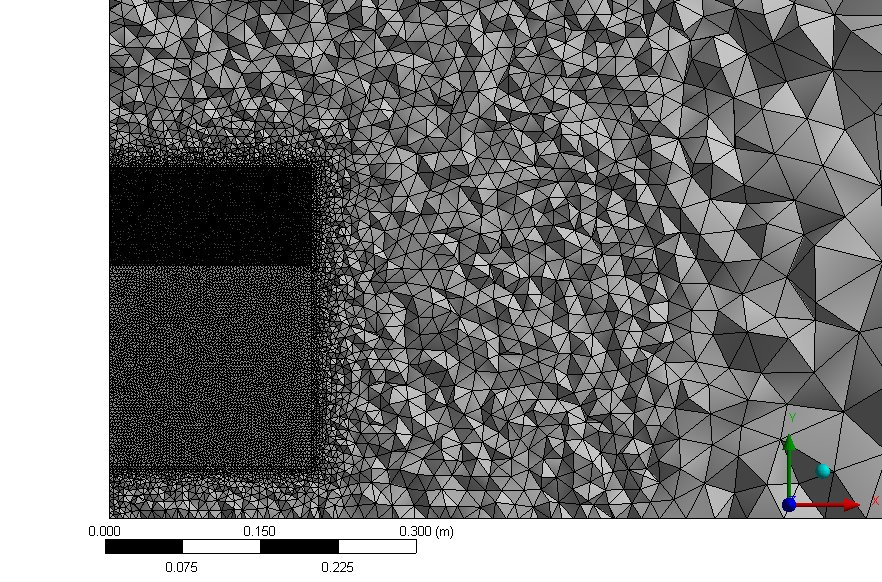


Figure 48: YX plane slice through 3.89m element PI tetrahedral mesh showing, whole elements and refinement towards Ahmed body.

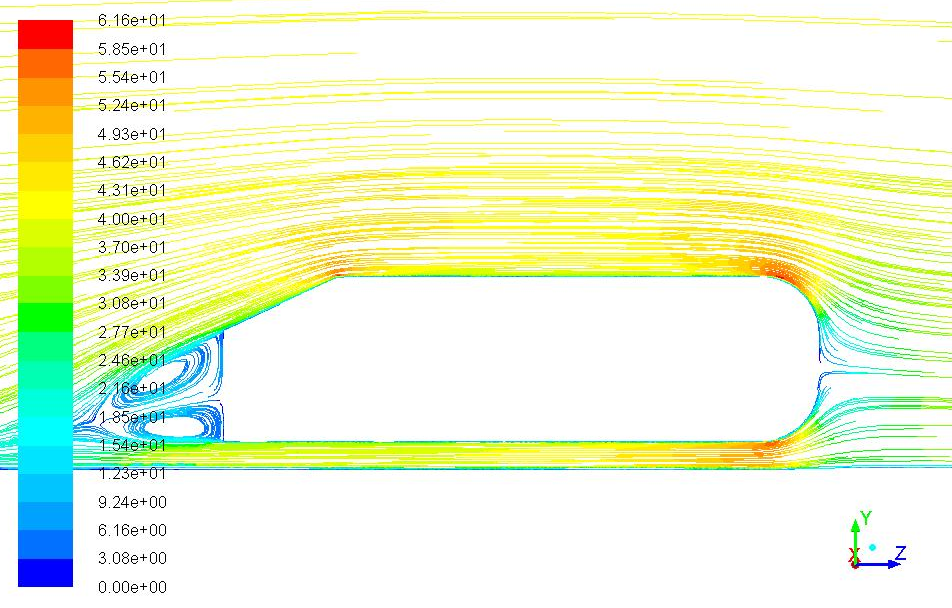


Figure 49: Velocity path line plot for final PI tetrahedral, 3.98m element, mesh.

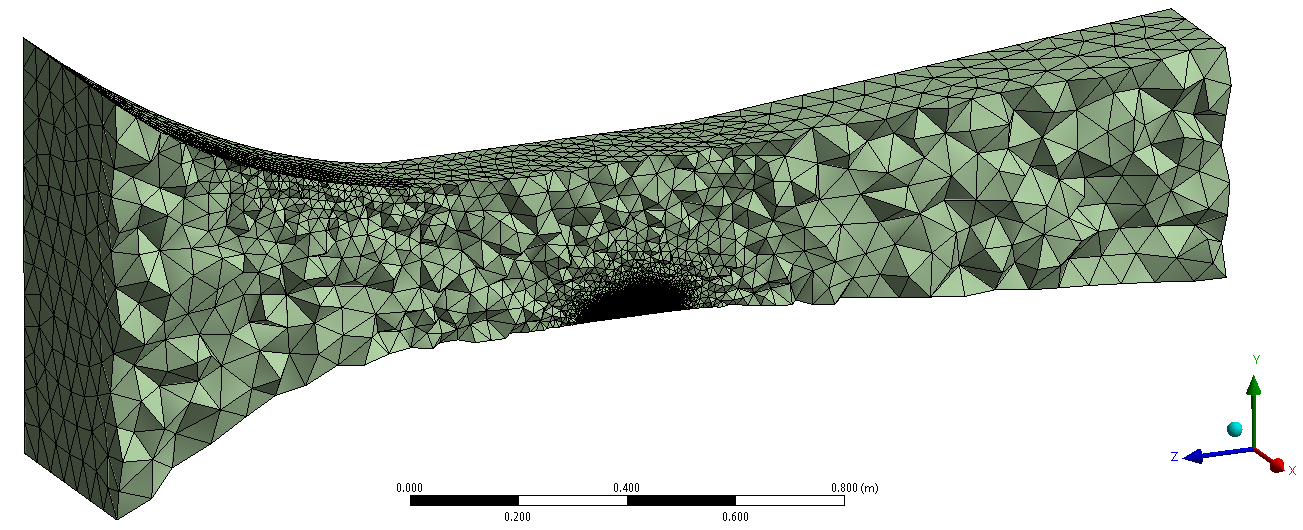


Figure 50: ZY symmetry plane cut showing whole-cell layout of final PI tetrahedral concept car mesh.

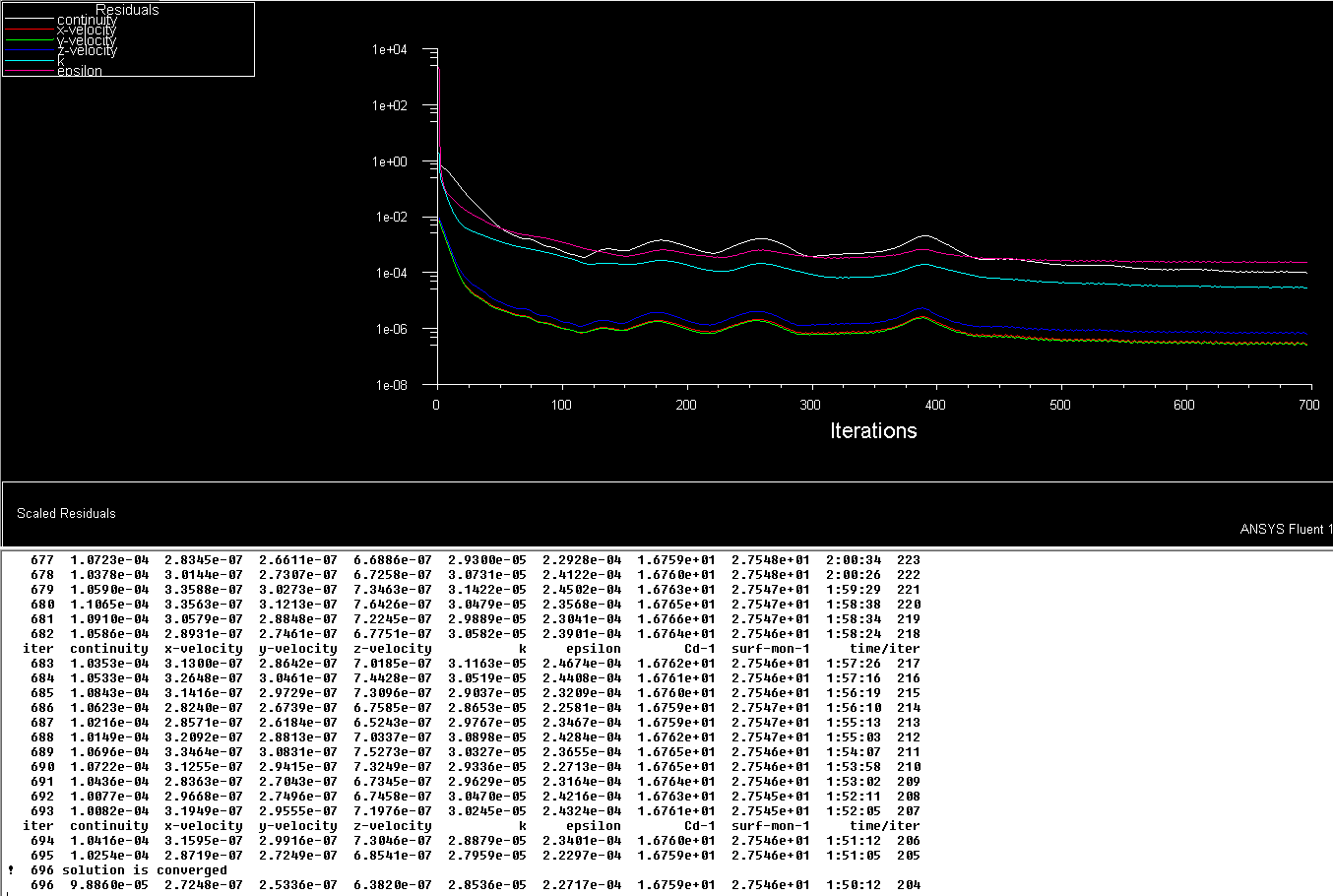


Figure 51: Residual convergence plot and text dialogue, showing number of iterations and various variables including drag coefficient (Cd-1) for final concept car mesh.

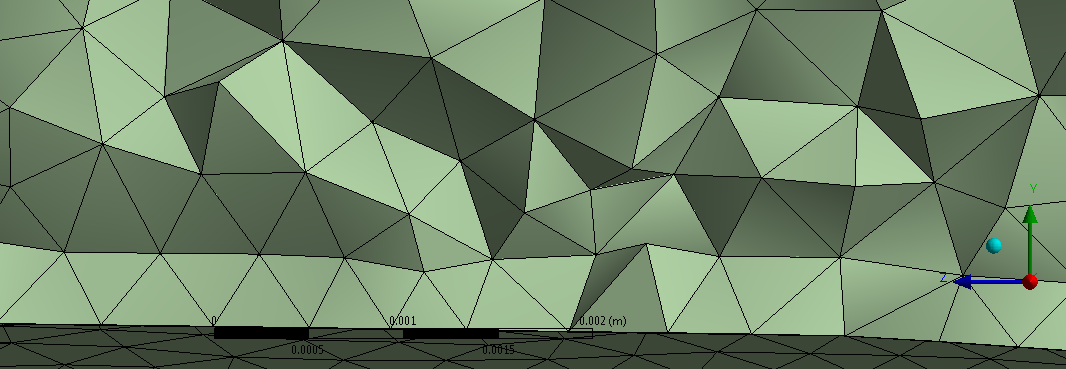


Figure 52: Scale of initial tetrahedral element layers on car surface. Height of cell centroid would be seen to be around 0.0002m, very close to that required to generate the optimum y+ value of 30, required for accurate implementation of the NEWF’s

Table 8: Table showing various quantities recorded during initial PD tetrahedral mesh convergence study on Ahmed body for later comparison.



Table 9: Table showing various quantities recorded during PD hybrid mesh convergence study on Ahmed body for later comparison.

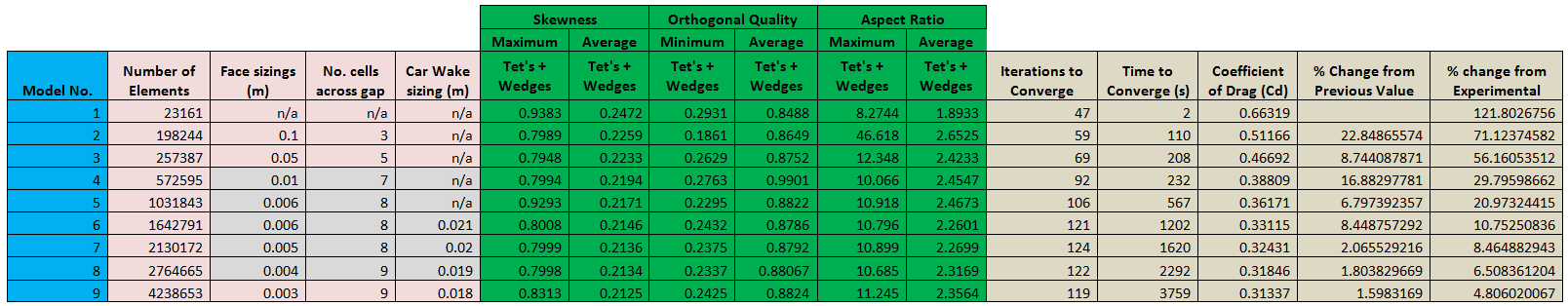


Table 10: Table showing various quantities recorded during PI cut-cell mesh convergence study on Ahmed body for later comparison.

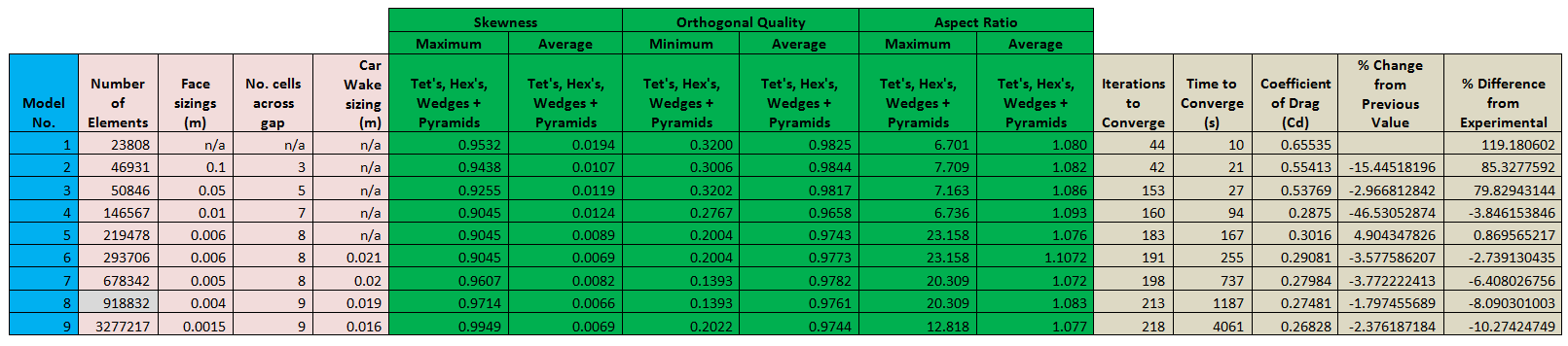


Table 11: Table showing various quantities recorded during PI tetrahedral mesh convergence study on Ahmed body for later comparison.

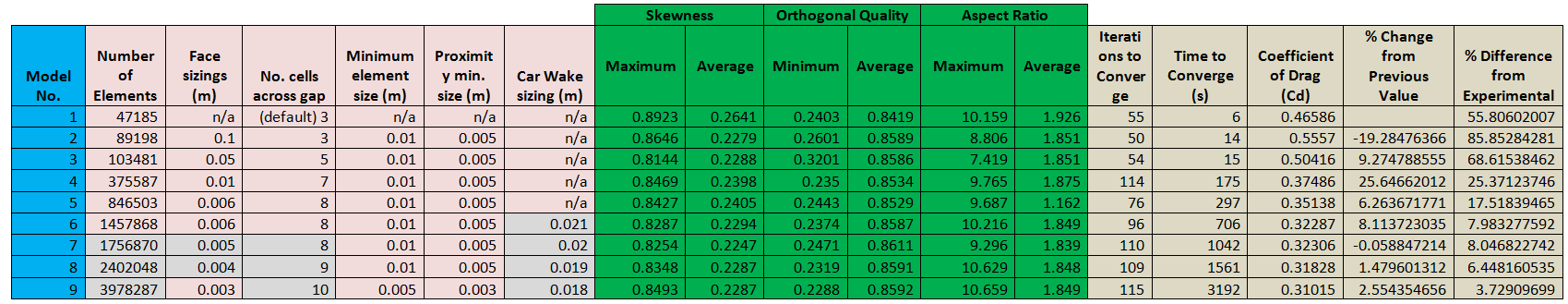


Table 12: Table showing various quantities for final PI tetrahedral concept car mesh.

# Appendix C – Health and safety risk assessment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Likelihood of injury**  ***(L)*** | ***Score*** |  | **Severity of injury**  ***(S)*** | ***Score*** |
| Improbable | *1* |  | Very minor injury; abrasions / contusions | *1* |
| Remote | *2* |  | Minor injuries; cuts / burns | *2* |
| Possible | *3* |  | Major injuries; fractures / cuts / burns / damage to internal organs | *3* |
| Probable | *4* |  | Severe injury; amputation / eye loss / permanent disability | *4* |
| Likely | *5* |  | Death | *5* |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Risk item** | **Effect** | **Cause** | **Likelihood** | **Severity** | **Importance** | **Action to Minimise Risk** |
|  | *Describe the risk briefly* | *What is the effect on any or all of the project deliverables if the cause actually happens?* | *What are the possible cause(s) of this risk?* |  |  | *L\*S* | *What action(s) will you take (and by when) to prevent, reduce the impact of, or transfer the risk of this occurring?* |
| 1 | Tripping over in the office-type environment. | Slowing of progress due to injury. | Garbage, wires etc. | 3 | 2 | 6 | Take due care when moving around the working space. |
| 2 | Inhalation of dust during ALM model preparation. | Slowing of progress due to injury. | Acrylic dust. | 2 | 3 | 6 | Wearing of protective glasses and dust-mask as well as working in well ventilated area. |
| 3 | Risk of eye strain, posture problems, repetitive strain injury. | Slowing of project due to illness/ injury. | Working in an office-type environment. | 4 | 1 | 4 | Adjust chair back so well supported, take regular breaks, light stretching, use screen light diffuser. |
| 4 | Contracting a contagious illness. | Slowing of project due to illness. | Through contact with mouse and keyboard that other people share. | 4 | 1 | 4 | Wipe down surfaces with disinfectant before use, avoid leaving house when possible. |

# Appendix D – Project Management (Gantt Chart)

