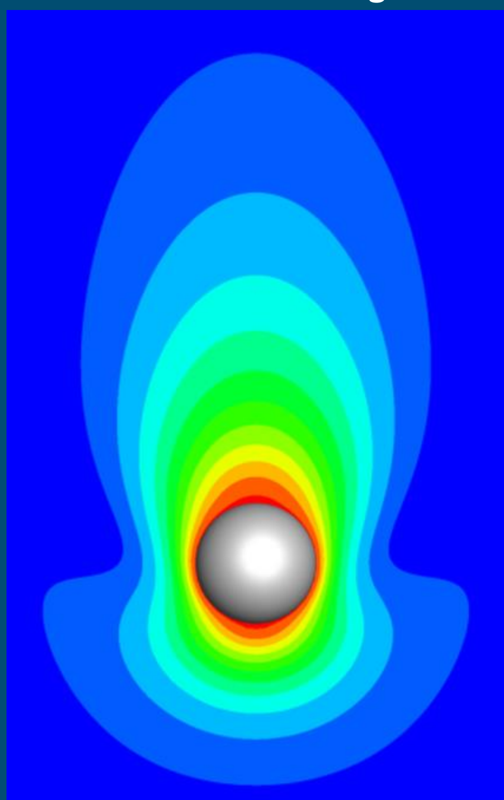
TUDUBLIN

*CFD Research Group Guide Series*

# FY and ME CFD Project Guide

**Authors:** **Fergal Boyle**
**Camila D' Bastiani**
**Gerald Gallagher**



**Rev. 3**

RESEARCH GROUP

OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH

TU DUBLIN

TECHNOLOGICAL UNIVERSITY DUBLIN

# Computational Fluid Dynamics in Engineering

What is Computational Fluid Dynamics? Computational Fluid Dynamics, or simply CFD, is the process whereby numerical methods are employed to simulate real-life fluid flows. CFD is now regarded as the “third” technique for the solution of fluid flow problems, complementing, but not replacing, the well-established approaches of theory and experiment. It is a relatively new branch of fluid mechanics and finds its niche in predicting fluid flows that are difficult or impossible to analyse using theory and are complex, time consuming, or expensive to measure experimentally.

CFD has evolved in parallel with the digital computer. The requirements of the aerospace industry provided the main impetus behind early research in CFD, and during the 1960s and 1970s computational methods first began to have a significant impact on aerodynamic analysis and design. Since then, with the rapid advances in computer hardware together with the parallel improvements in numerical methods, CFD has matured to the point where it is now recognised as a key tool in engineering design and is employed across all disciplines where the flow of fluid is important. Using state-of-the-art CFD methods on modern computers, a wide range of fluid flow problems encountered in engineering can be modelled, from the high-speed air flow over an aircraft to the movement of water and effluent in an estuary subject to tidal effects.

In understanding how CFD works, it must be borne in mind that the governing equations of fluid mechanics are fundamental to CFD. In the CFD literature, the complete form of the equations governing fluid motion are referred to as the Navier-Stokes equations. These are named after the French civil engineer Claude-Louis-Marie-Henri Navier and the Irish mathematical physicist George Gabriel Stokes who independently formulated a subset of these equations in the first half of the nineteenth century. The Navier-Stokes equations are the mathematical representations of three fundamental laws of nature as applied to fluid flow: (1) conservation of mass, (2) Isaac Newton’s second law of motion (force is equal to the time rate of change of momentum) and (3) conservation of energy. The most common strategies employed in CFD involve applying these equations, which may be in partial differential equation or integral equation form, at user-defined points in the domain in which the flow is being calculated. This application is achieved using the finite difference, finite volume, or finite element discretisation method, and leads to a set of discrete algebraic equations that are solved using an appropriate numerical technique. The outcome is a set of results for each of the specified points that usually includes static pressure, density, velocity components, temperature etc. As the number of discrete equations is usually quite large, typically ranging from hundreds of thousands to several million, the whole process is programmed for execution by a computer. This is why the ability to model fluid flows with higher levels of accuracy or with more complex flow physics is closely linked to advances in computer hardware design.

What are the steps in performing a CFD analysis? At present, using the finite volume method, the standard approach is to employ the following three-step process: pre-processing, flow solution, and post-processing. In the pre-processing phase software called a "mesh generator" or "grid generator", typically with CAD capabilities, is employed to build a model of the flow domain, referred to as the computational domain, in which the flow will be predicted. For example, in order to model the flow around a car, an accurate model of the car geometry is built (or imported from a CAD package) and then the extents of the computational domain are established. Subsequently, the computational domain is divided into a contiguous set of smaller non-overlapping sub-domains, called cells, to form a mesh. These cells may vary in shape and size throughout the domain, with smaller size cells usually being employed in regions where it is known that the nature of the flow changes rapidly. Also, the mesh topology depends on the form of the equations solved in the flow solution phase. The accuracy of the flow solution depends significantly on a sufficient number of cells being employed to resolve the flow detail and also on the quality of the mesh generated. When the pre-processing step is complete the mesh is transferred to the flow solver.

In the flow solution phase the CFD solver solves a particular form of the governing equations of fluid mechanics for each of the cells generated in the previous step, subject to appropriate boundary conditions imposed on the boundaries of the computational domain. These boundary conditions inform the flow solver as to the properties of the flow at these locations and are specified by the user from prior knowledge of the flow. The ability to solve the Navier-Stokes equations exactly for common engineering flow problems would require computational resources, in terms of both processing power and memory, beyond the capability of any current or projected computer. To-date, simplifications to the governing equations have been introduced that change the nature of these equations to make the computational simulation of fluid flow both feasible and cost-effective, e.g. the omission of viscosity. As the capability of computer hardware and CFD solvers have improved, the number of simplifications have been progressively reduced with the result that state-of-the-art CFD solvers are now capable of accurately resolving very complex flow physics. CFD flow solvers typically employ an iterative process to solve the discretised equations and convergence criteria are required to indicate when the iteration process should be terminated. Once the flow solution step is complete results are output for each cell in the computational domain.

Having successfully completed the solution phase it is then necessary to post-process the results obtained. This step is as important as the previous two. The flow solution phase may generate millions of numbers and it is the successful analysis of these that will justify the cost and effort expended in performing the CFD analysis. Post-processing involves putting the generated data into easy-to-use forms, e.g., graphical or tabular, and then analysing and interpreting these data in order to draw useful conclusions.

When getting started with CFD the main requirement is a good knowledge of fluid mechanics, and in particular the Navier-Stokes equations. This is essential before embarking on the application of CFD to any engineering problem. After this a good starting point is to look at the web site at http://www.cfd-online.com. This site, best described as a CFD portal, contains a number of useful resources including recommended books on CFD for all levels, a list of commercial-off-the-shelf software tools, online discussion fora, etc. It should contain sufficient information when starting out. On the subject of commercially available software tools it is possible to purchase products such as Fluent or CFX that contain the requisite applications for performing a complete CFD analysis, typically including a variety of mesh generators and flow solvers, and a tool for post-processing. However, it is also worth noting that it is possible to purchase stand-alone mesh generators such as Gridgen or ICEM CFD, and post-processors such as EnSight, Tecplot, or FIELDVIEW. Some members of the CFD community tend to favour using separate applications for each stage of a CFD analysis because of their extra functionality and user friendliness.

In summary, CFD has evolved from being a tool of limited use to one that is essential in engineering design that involves fluid flow. This evolution was facilitated by parallel improvements in numerical algorithms and in computer hardware. Current CFD capabilities enable complex flow features to be modelled in an efficient and cost-effective manner. The relentless improvements in computer hardware coupled with the on-going intense research into more sophisticated numerical methods means that CFD will permeate the engineering design process even further in the future.

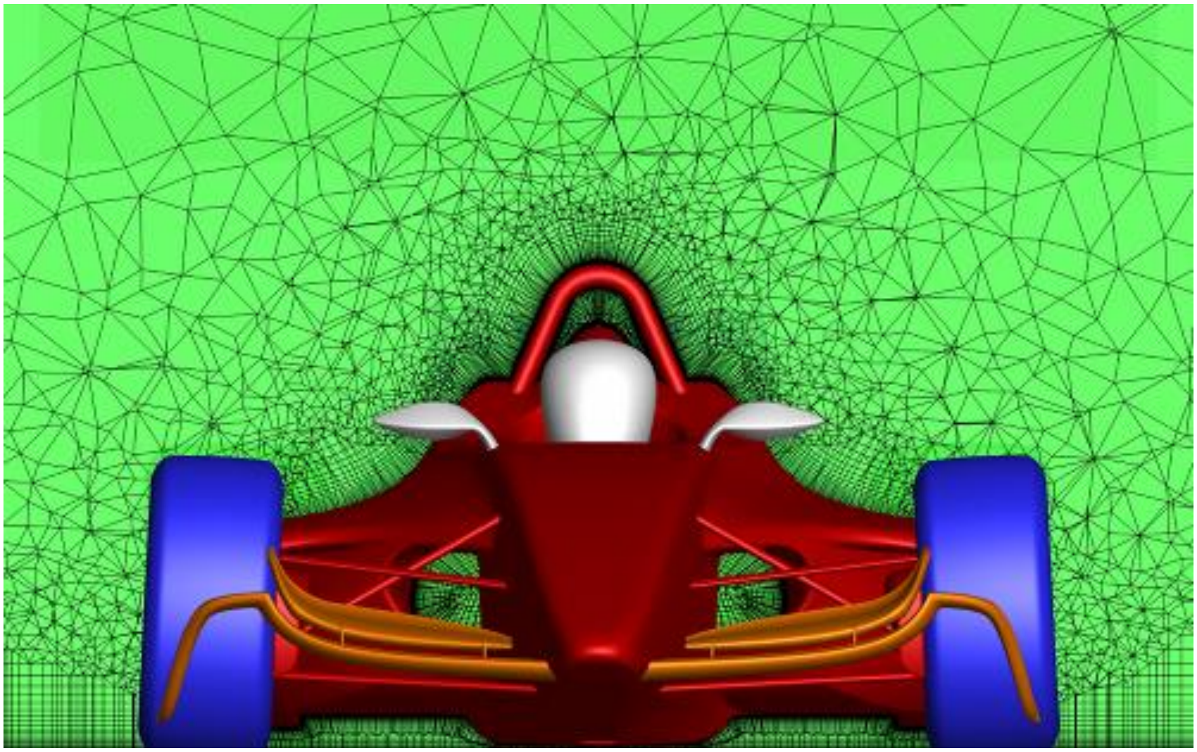


**Figure 1: Pointwise Billion-Cell Hybrid Mesh**

## Checklist When Performing a CFD Simulation

## (in no particular order)

| | | |
|---|---|---|
| 1 | **What software are you using?**<br><br>If using ANSYS CFX or Fluent you can use a “student” version which you can download yourself from the internet or use a “teaching” version through the TU Dublin ‘’Azure Virtual Desktop’’ (AVD) system which is now open to all Engineering & Built Environment students and staff. There seem to be some issues using the student version, and also it is limited to 500,000 cells. The teaching version has twice this limit, i.e. 1,000,000 cells and so provides more flexibility when performing complex CFD analyses. The AVD system (previously known as Windows Virtual Desktop or WVD, but now renamed by Microsoft) is a cloud-based multi-session Virtual Desktop Infrastructure (VDI) system which provides TU Dublin students and staff with access to a range of software applications on the cloud. The AVD has been recently updated to include the latest 2022 software licences to match the software available on networked PCs in campus computer rooms. You will find guidance on how to access the system on the TU Dublin Azure Virtual Desktop webpage. Standalone versions of CFX and Fluent have been set up on Azure. There is a 5-minute delay the first time someone logs on to Azure. This should not happen for subsequent logons. There is a short delay on starting workbench (around 10 s), until the side menu populates itself. After this, performance is normal.<br><br>If using OpenFOAM then you should contact Gerald Gallagher (gerald.gallagher@tudublin.ie) for an easy-to-use solution for installation. | Tick Yes/No |
| 2 | **Is the flow compressible or incompressible?**<br><br>Usually a flow is treated as incompressible if the density does not vary throughout the flow field, i.e. density variations are negligible or zero. This typically is the case if the Mach number (dimensional parameter named after Ernest Mach and defined as V/a, where V is velocity and a is the speed of sound) through the flow field is less than 0.3. If the flow is incompressible then, typically, you should use a pressure-based solver within the package you are employing, such as SIMPLE, SIMPLEC, SIMPLER, PISO etc. These are all just variations on the same thing, i.e. pressure correction. If the flow is compressible | |

| | | |
|---|---|---|
| | then typically you use a density-based solver, as the density will vary throughout the flow field, and compressible-flow features such as shock waves may appear. | |
| 3 | **Is the flow laminar or turbulent?**<br><br>This will be determined by the Reynolds number, a dimensionless parameter that relates inertial to viscous forces, of the flow. A laminar flow is far more easily predicted than a turbulent flow. There are three main techniques/approaches for predicting turbulent flows: RANS, LES and DNS.<br><br>a) **RANS**: This stands for *Reynolds Averaged Navier Stokes* (modelling). This is the standard and typically-employed model for projects, and still the most commonly employed model in industry. The governing equations of fluid mechanics, i.e. the Navier-Stokes equations are time averaged leading to new terms in the governing equations, called Reynolds stresses, i.e. the transients effectively are time-averaged /filtered out. To “close” the set of equations, a turbulence model must be employed. There are various turbulence models you can employ such as k-ω, k-ε, SST, etc. Each turbulence model has its own advantages and disadvantages, and situations for which it is ideally suited. You must be careful which turbulence model you select!<br><br>b) **LES:** This stands for *Large Eddy Simulation*. With this technique, the large eddies in the flow are predicted and the smaller eddies are filtered out. These predictions require extensive computational resources as the simulations are transient, and generate a huge amount of data. Not recommended for projects, and anyway the results you get using a simpler RANS model might be just as good.<br><br>c) **DNS:** This stands for *Direct Numerical Simulation*. This is the ultimate in terms of CFD modelling but is way beyond current and foreseeable computational resources. With DNS nothing is filtered out. Eddies of all time and spatial scales are accurately predicted. Maybe in 100 years! Hopefully you will be finished before then. | |
| 4 | **Is the flow steady or transient?**<br><br>A flow is steady if it doesn’t vary with time. Steady-flow analyses require substantially less computational resources than transient analyses. | |

| | | |
|---|---|---|
| 5 | **For steady-flow analyses, are your simulations solution converged?**<br><br>If doing a steady-flow analysis how do you know your solution is solution converged? When performing a steady-flow simulation you will see the solver iterating. The solver will typically stop iterating for one of three reasons:<br><br>a) The software crashes, i.e. it can't continue as something is wrong and the solution has diverged.<br><br>b) A prescribed maximum number of iterations has been reached, i.e. 200, 400 etc.<br><br>c) The solution tolerance had been achieved, e.g. the RMS of the u velocity is less than 1e-08.<br><br>Using a solution tolerance is probably the best approach to take, as there is no guarantee that your solution is converged when using a "maximum number of iterations". In the past students have used a maximum number of iterations criterion, and it is quite obvious from the convergence plots that the solution is not converged. It is very important that students monitor solution convergence and study convergence plots (usually automatically) produced by the software, and try and understand what is going on. What level of tolerance is standard? What does this tolerance actually mean? It's a tolerance on what? | |
| 6 | **Did you do a mesh convergence study?**<br><br>The idea behind performing a mesh convergence study is to identify the minimum mesh density (no. of cells/elements) to get "sufficiently accurate" results. Minimising the mesh density saves on computational resources, both memory requirements and processing time. But what is meant by "sufficiently accurate"? This is not straightforward. If you have theoretical data or experimental data with which to compare your predictions, then "sufficiently accurate" might mean that your predictions, i.e. drag coefficient, might differ by less than 5%. You must identify a <u>metric</u> for comparison, e.g. drag coefficient, lift coefficient, max centreline velocity, etc., and you must also identify a <u>tolerance</u>, i.e. the percentage difference between values that you deem acceptable. If you don't have theoretical data or experimental data with which to compare you predictions, then you could compare results between a pair of meshes, for example a coarse mesh and a finer one. | |

| | | |
|---|---|---|
| 7 | **If the flow is transient did you do a timestep convergence study?**<br><br>The idea behind performing a timestep convergence study is to identify the maximum timestep that can be used when performing a transient analysis to get sufficiently accurate results. Maximising the timestep saves on computational resources, both memory requirements and processing time. Choosing a very small timestep may result in long run times, while choosing too large of a timestep may lead to inaccurate results. The temptation is to use a large timestep and therefore cut down on the runtime. However, the results may be poor. Most commercial software packages employ implicit time stepping for transient flow simulations, and thus the user can choose whatever timestep he/she likes. The timestep chosen should be such as to accurately captures the flow transients while at the same time minimising computational resources. | |
| 8 | **Did you undertake any validation test case(s)?**<br><br>It is very important to perform some sort of validation test case(s) to demonstrate that you/the package are capable of accurately predicting some benchmark flow problem or some flow problem for which theoretical or measured data exist. Successfully completing a validation test cases gives you confidence in your knowledge and skills and gives a solid foundation for moving forward. | |
| 9 | **Are your boundary conditions correct?**<br><br>The correct placement of boundaries and application of boundary conditions is one of the most critical steps when performing a CFD simulation. When performing a CFD simulation of external flow over an airfoil for example, if applying freestream conditions at the farfield boundary, then it is recommended to place this boundary 50 airfoil chord lengths away from the airfoil so that the effects of the airfoil are negligible at this boundary, and application of freestream boundary conditions is acceptable and physically correct. Another example. When performing a CFD simulation of external flow over a truck or car, once again if applying free stream conditions at the farfield boundary then this boundary must be sufficiently far away so that application of these boundary conditions is physically correct. Students tend to place these boundaries too close to the car or truck and don't appreciate the consequences. | |

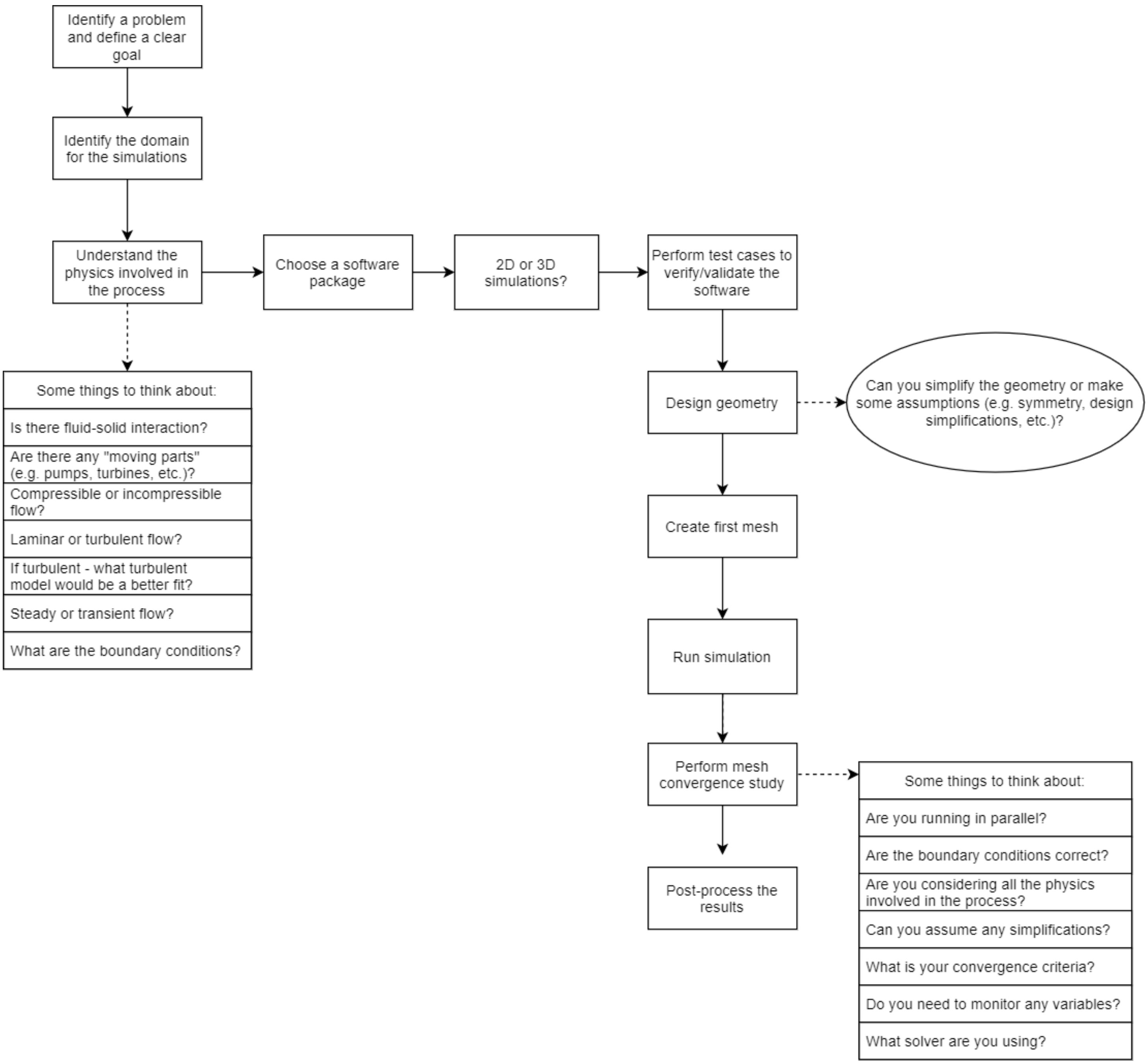



**Figure 2: CFD Project Workflow Flowchart**

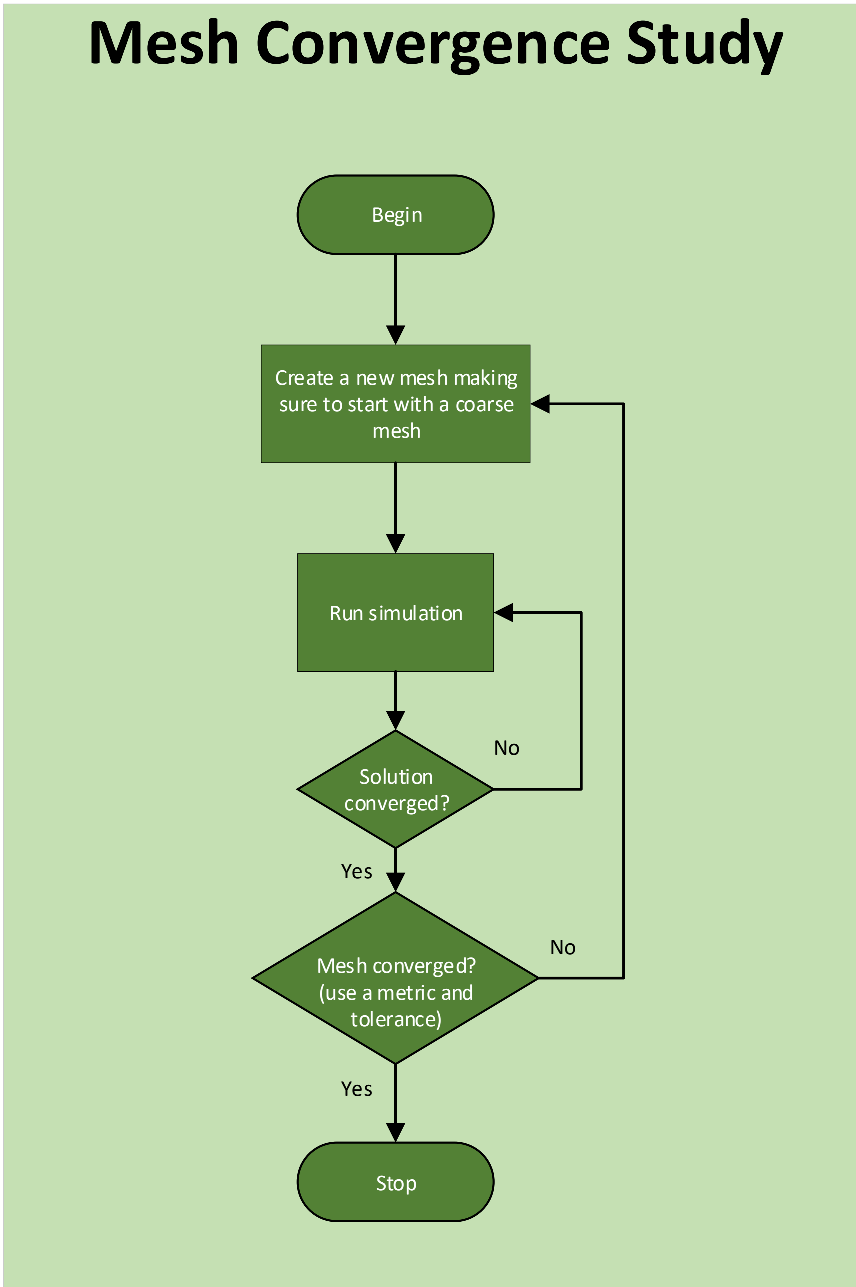
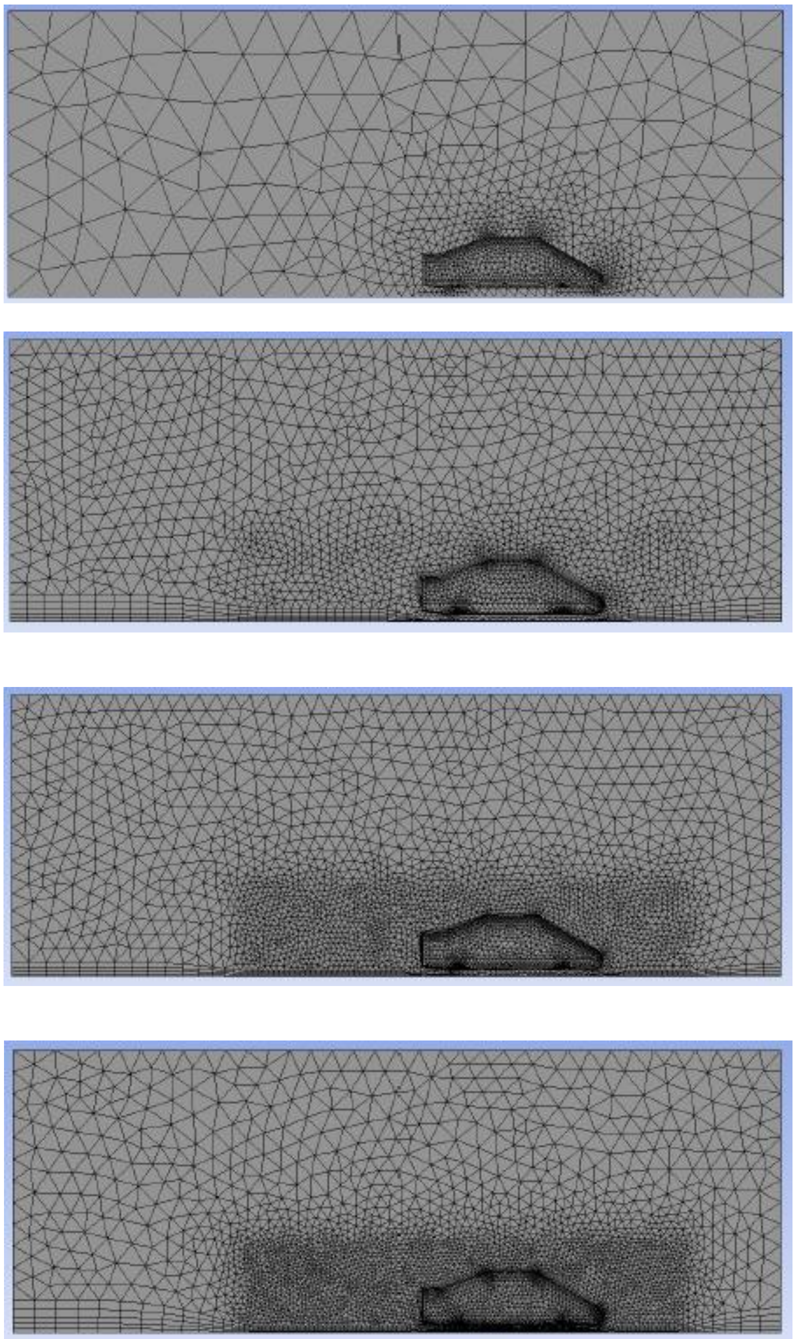



**Figure 3: Mesh Convergence Study Flowchart**



*Mesh 1*

*Mesh 2*

*Mesh 3*

*Mesh 4*

**Figure 4: CFD Mesh Convergence Study of Flow Over a Car Performed by a Former Undergrad Student**

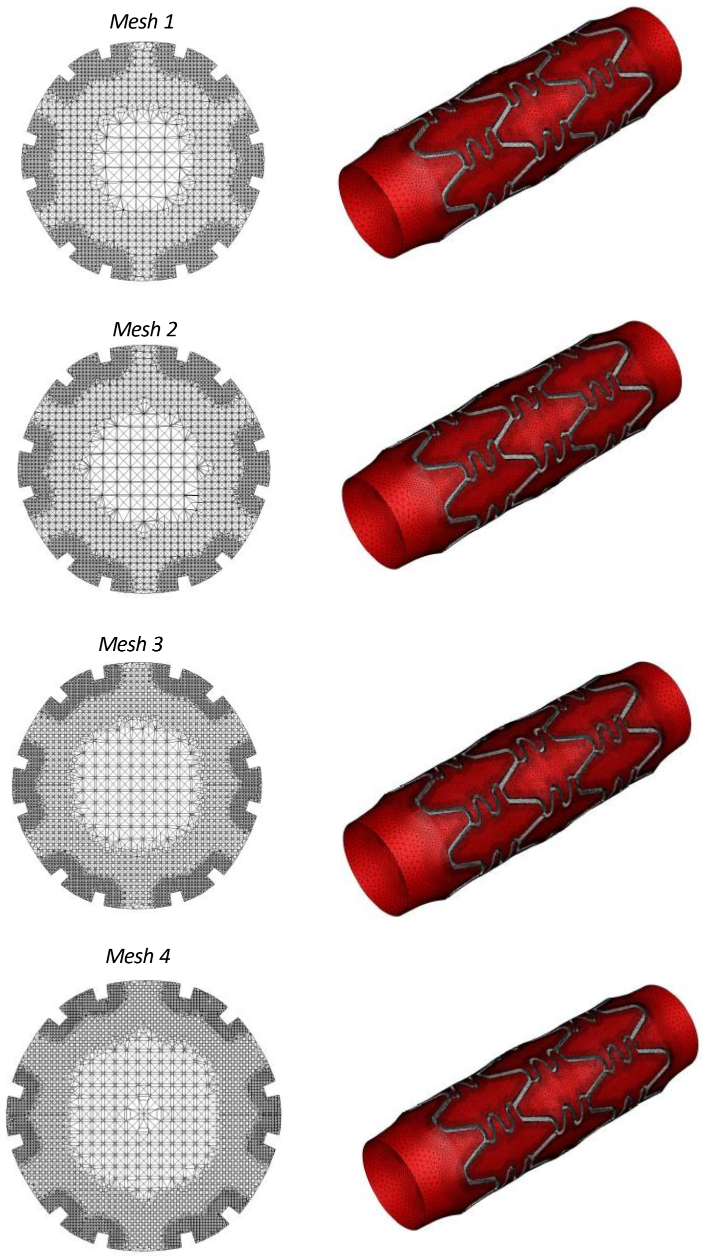



**Figure 5: CFD Mesh Convergence Study of Flow Through a Coronary Stent Performed by a Former PhD Student**