

A report on

**Structural design, simulation
and analysis of a low altitude
sounding rocket**

Submitted in partial fulfilment for the
award of the degree of
Bachelor of Technology in Mechanical
Engineering By

Mr. Niket Shah
(Reg No. M1410053 , Seat No. BTM873)
Mr. Deven Mhadgut
(Reg No. M1410063 , Seat No. BTM821)
Mr. Omkar Joshi
(Reg No. M1410058, Seat No. BTM817)

Supervisor:
Dr. B. N. Bhasme

Assistant Professor,
Mechanical Engineering Department,
Sardar Patel College of Engineering



(Mechanical Engineering Department)
Bharatiya Vidya Bhavan's
SARDAR PATEL COLLEGE OF ENGINEERING
(Government Aided Autonomous Institute)
MUNSHI NAGAR, ANDHERI (WEST), MUMBAI, INDIA.
Year: 2017-18

PROJECT APPROVAL SHEET

This report entitled “**Design, Simulation and Fabrication of a Model Rocket**” by Niket Shah, Deven Mhadgut, and Omkar Joshi has been approved for the degree of Bachelor of Technology in Mechanical Engineering.

Examiners:

Dr. R.S Maurya
Mechanical Engineering Department,
Sardar Patel College of Engineering,
Mumbai.

Dr. S.S Umale
Mechanical Engineering Dept.,
Sardar Patel College of Engineering,
Mumbai.

Prof. Shaikh Haseen
Mechanical Engineering Dept.,
Sardar Patel College of Engineering,
Mumbai.

Head of Department:

Dr. Nilesh Raykar,
Mechanical Engineering Department,
Sardar Patel College of Engineering,
Mumbai.

Principal:

Dr. P.H. Sawant,
Sardar Patel College of Engineering,
Mumbai.

Date and Place: _____

BONAFIDE CERTIFICATE

It is certified that the project “**Structural design, simulation and analysis of a low altitude sounding rocket**” is the bonafide work of Niket Shah, Deven Mhadgut, and Omkar Joshi, who carried out the project work under my supervision for the academic year 2017-18.

Supervisor:

Dr. B.N. Bhasme

Assistant Professor,
Department of Mechanical Engineering,
Sardar Patel College of Engineering,
Andheri (West), Mumbai-400 058

DECLARATION

I declare that this written submission represents my ideas in my own words and in places where other's ideas or words are included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penalty action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Niket Shah
(BTM873)

Deven Mhadgut
(BTM821)

Omkar Joshi
(BTM817)

Date : _____

Abstract

Model rocketry has been gaining prominence in India over the past few years. Students of various engineering disciplines are taking interest in rocket science and technology because of a conducive environment and changing attitude towards the field. However, the technical know-how of rockets, their working, etc., is still obscure for students. This project aims to design, analyze and test a model rocket. By doing this, we intend to expand the existing knowledge base regarding model rocketry and contribute towards popularizing the same in India. The report emphasizes the methodology followed by the team to design and fabricate a sounding rocket. Open Rocket and CATIA softwares were used for basic design. In this report, we have documented design methodology and simulation.

In this project, the drag coefficient of a self-designed model rocket was calculated using CFD Software Ansys FLUENT and wind tunnel analysis and the results were compared. Both the methods were able to provide good information to guide the design development. Crucially, the information and guidance from both methods was consistent; that is either method could have led the design development to a similar final result. Each method has, of course, advantages and limitations, and to some extent these are complementary.

Also, the theory behind passive control and future scope of the work has been discussed. The drag force obtained was used to obtain the trajectory of its flight, (powered by an F-class Estes rocket motor), which was graphically simulated using MATLAB software.

Keywords: model rocketry, drag coefficient, trajectory simulation, wind tunnel analysis, rocket motor.

Contents

List of Figures.....	viii
Chapter 1 Introduction	1
Chapter 2 Literature Review	3
Chapter 3 Methodology	5
Chapter 4 Design.....	6
4.1 Nose Cone	6
4.1.1 Nose cone shape equations	7
4.1.1.1 Conical.....	7
4.1.1.2 Tangent ogive	7
4.1.1.3 Haack series	8
4.1.2 Nose cone drag characteristics	9
4.1.2.2 Wetted Area	9
4.1.2.3 General Shape.....	9
4.1.2.4 Bluffness Ratio.....	10
4.2 Airframe.....	11
4.3 Fins	12
4.3.1 Aspect Ratio.....	14
4.3.2 Airfoil Geometry	14
4.3.3 Fin Flutter.....	16
Chapter 5 Simulation.....	19
5.1 Fluent.....	19
5.1.1 Geometry	19

5.1.1.1 Nose	19
5.1.1.1.1 Fineness ratio	19
5.1.1.1.2 Mach number wave drag.	19
5.1.1.2 Fins.....	20
5.1.2 Mesh.....	21
5.1.2.1 Method.....	21
5.1.2.2 Face Sizing.....	23
5.1.2.3 Mesh Metric.....	25
5.1.3 Setup.....	25
5.1.4 Model	26
5.1.5 Boundary Conditions.....	28
5.1.5.1 Turbulence Intensity	28
5.1.5.2 Turbulence Length Scale and Hydraulic Diameter	28
5.1.5.3 Turbulent Viscosity Ratio.....	29
5.1.6 Solution Methods	29
5.2 Trajectory Prediction	31
5.2.1 Displacement curve.....	31
5.3 Wind tunnel	33
5.3.1 Method.....	33
5.3.2 Wind tunnel Specifications.....	33
 Chapter 6 Manufacturing Process	 37
 Chapter 7 Experimental Setup.....	 39
Chapter 8 Results and Discussion.....	41
Chapter 9 Conclusion.....	45
Chapter 10 Future Scope.....	45
References	46
Appendix	47

Appendix I- Trajectory prediction at different launch angles	47
Appendix II- Load Cell calibration code for arduino.....	55

List of Figures

Figure	Description	Page no
1	Basic components of a model rocket	9
2	A model rocket taking flight	10
3	Nose Cone basic dimensions	14
4	Comparison of drag characteristics of nose cone vs. Mach number	15
5	Percentage change in drag vs Diameter	15
6	Haack Series	16
7	Comparison of drag characteristics of nose cone vs. Mach number	17
8	Percentage change in drag vs Diameter	18
9	Aspect Ratio Illustrated	22
10	Effect of lift force on fin geometry	22
11	Typical Airfoil Cross Section	23
12	Fin Cross Section	24
13	Actual fin airfoil design	24
14	Actual Rocket Design	25
15	NACA airfoil geometrical construction	27
16	Mesh details	29
17	Face sizing of nose cone and body	29
18	Face sizing of fins	30
19	Face sizing of fin base	30
21	Face sizing of body part-1	31
23	Overview of the Density-Based Solution Method	32
24	Full body mesh	34
25	Thrust Curve of Estes F15	37
26	Displacement Curve for Vertical launch	38
27	Trajectory of a rocket launched at 10° launch angle	39
28	Wind Tunnel Schematic Diagram	40
29	Basic Beam Load Cell	41
30	Load Cell Dimensions	41
31	Load Cell Technical Specifications	42
32	Force Balance Mechanism	42
33	HX 711	43
34	Load Cell Circuitry	43
35	IIT Bombay Wind Tunnel	44
36	Selective laser sintering	45
37	Photograph of Fabricated Model rocket	46
38	Load Cell Circuitry	47
39	Photograph of Force Balance Setup	48
40	Pressure Contour	53
41	Velocity Contour	53

Chapter 1 Introduction

A model rocket is a small rocket designed to reach low altitudes (usually to around 100–500 m (330–1,640 ft) for a 30 g (1.1 oz) model and which are recovered by a variety of means.

According to the United States National Association of Rocketry (NAR) Safety Code, model rockets are constructed of paper, wood, plastic and other lightweight materials. The code also provides guidelines for motor use, launch site selection, launch methods, launcher placement, recovery system design and deployment and more. Since the early 1960s, a copy of the Model Rocket Safety Code has been provided with most model rocket kits and motors. Despite its inherent association with extremely flammable substances and objects with a pointed tip traveling at high speeds, model rocketry historically has proven to be a very safe hobby and has been credited as a significant source of inspiration for children who eventually become scientists and engineers. [1]

Model rocketry was developed during the “space race” era as an alternative to the amateur rocket activity — involving metallic airframes and the mixing of dangerous propellants — that was responsible for injuring and even killing numerous young scientific experimenters. Today, model rocket kits and motors can be purchased in almost every hobby shop and toy store. Kits are designed for all ages and all levels of challenge, from simple starter kits to complicated scale models. Rockets and related technology have long been accessible exclusively to the technologically elite. This is primarily due to the simple fact that rockets have extremely high developmental costs associated. Some of these expenses include costly fuels, exotic materials, and many hours of testing and refinement. While this is still mostly the case today, a shift can be seen focusing on newer technology that is more economical and innovative. Critical aspects of the project include keeping the rocket design as simple and as inexpensive as possible while still providing high performance and reliability.

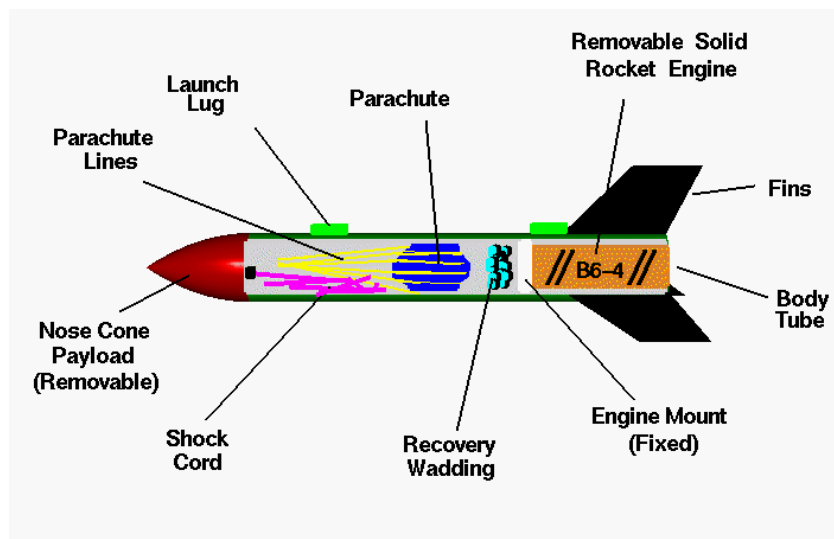


Figure 1- Basic components of a model rocket

Beginning at the far right, the body of the rocket is a green cardboard tube with black fins attached at the rear. The fins can be made of either plastic or balsa wood and are used to provide stability during flight. Model rockets use small, pre-packaged, solid fuel engines. The engine is used only once, and then is replaced with a new engine for the next flight. Engines come in a variety of sizes and can be purchased at hobby stores and at some toy stores. The thrust of the engine is transmitted to the body of the rocket through the engine mount. This part is fixed to the rocket and can be made of heavy cardboard or wood. There is a hole through the engine mount to allow the ejection charge of the engine to pressurize the body tube at the end of the coasting phase and eject the nose cone and the recovery system. Recovery wadding is inserted between the engine mount and the recovery system to prevent the hot gas of the ejection charge from damaging the recovery system. The recovery wadding is sold with the engine. The recovery system consists of a parachute (or a streamer) and some lines to connect the parachute to the nose cone. Parachutes and streamers are made of thin sheets of plastic. The nose cone can be made of balsa wood, or plastic, and may be either solid or hollow. The nose cone is inserted into the body tube before flight. An elastic shock cord is connected to both the body tube and the nose cone and is used to keep all the parts of the rocket together during recovery. The launch lugs are small tubes (straws) which are attached to the body tube. The launch rail is inserted through these tubes to provide stability to the rocket during launch. [9]



Figure 2- Photograph of a model rocket taking flight [18]

Chapter 2 Literature Review

Model rockets have been employed in student projects, but very few papers in aerospace education offer concise summaries of activities at university-course levels. Mark Uitendaal, in his Master's thesis has described the process of designing, simulating and production of the Stratos rocket. This rocket is a project of the Delft Aerospace Rocket Engineering (DARE) team based at TU Delft. The report starts with a project objective, gives an overview of the design process, simulations, and various sub-systems; and concludes with a comparison between the simulation and the results of the flight. The role of the author in the whole Stratos project was project initiator, project leader and responsible for rocket design, motor design and flight simulations. Since this is a thesis, only the work performed by the author is written down in the report. The purpose of this rocket was to break the European altitude record for amateur rockets which was at that time 10.7 km. The Stratos broke this record and soared to 12.551 km above its launch pad at Esrange Space Centre in Kiruna, Sweden. The stage separation mechanisms, propellants used and their properties, parachutes and their types and pyrotechnics used were described well in this report. The various stages such as sustainer, booster and their designs were well documented.[1]

The MIT Rocket Team aimed to develop and test methods of analyzing the causes and effects of fin flutter as it pertains to the flight of high powered rockets. They developed a model sounding rocket which reached an apogee of 1 mile. The scientific payload for the 2011-2012 year was a system for quantitatively measuring flutter on secondary set of fins. This system included a set of high-speed video cameras, and strain-gauges built into test fins. The data and video from the flight was analyzed and compared to computer models developed prior to flight. A secondary payload was flown as part of ongoing educational outreach programs. The secondary payload was a science experiment developed and built by a local high-school team. The payload was selected as part of a mini-design competition. The various modules and subsystem designs, material selection criteria are well explained. Design for fins is explained in detail. [2]

Arash Ramezani et al present the most commonly used mathematical methods for determining the trajectory, calibre and type of a projectile based on the estimation of the ballistic coefficient. The algorithms are implemented on low-end computer systems used in military camps. A comparison between the methods gave information about error propagation and reliability of the system. In addition, simulation-based optimization processes were presented that enabled iterative adjustment of predicted trajectories in real time. Combinations of these methods have been compared to increase the accuracy of simulation[4]. Nikolaos S. Christodoulou presents the algorithm using Runge-Kutta methods of orders 4 and 5 for systems of odes. The running time and maximum errors for the two methods are compared on Rössler system.

M. Khalil et al investigate the trajectory prediction and dispersion for unguided fin stabilized artillery rocket in order to explain the importance of the rocket production accuracy and the benefit of using guided rockets. The total dispersion results mainly from three effects. The first is the dispersion due to rocket production inaccuracy, which includes propellant mass, composition inaccuracy, rocket total mass, axial and lateral moments of inertia and resultant centre of gravity. The second dispersion during boosting phase which includes launcher deflection, missile tip-off from the launcher, thrust and fin misalignments, and atmospheric disturbances such as tail wind, cross wind, and gusts. While the third is the dispersion during

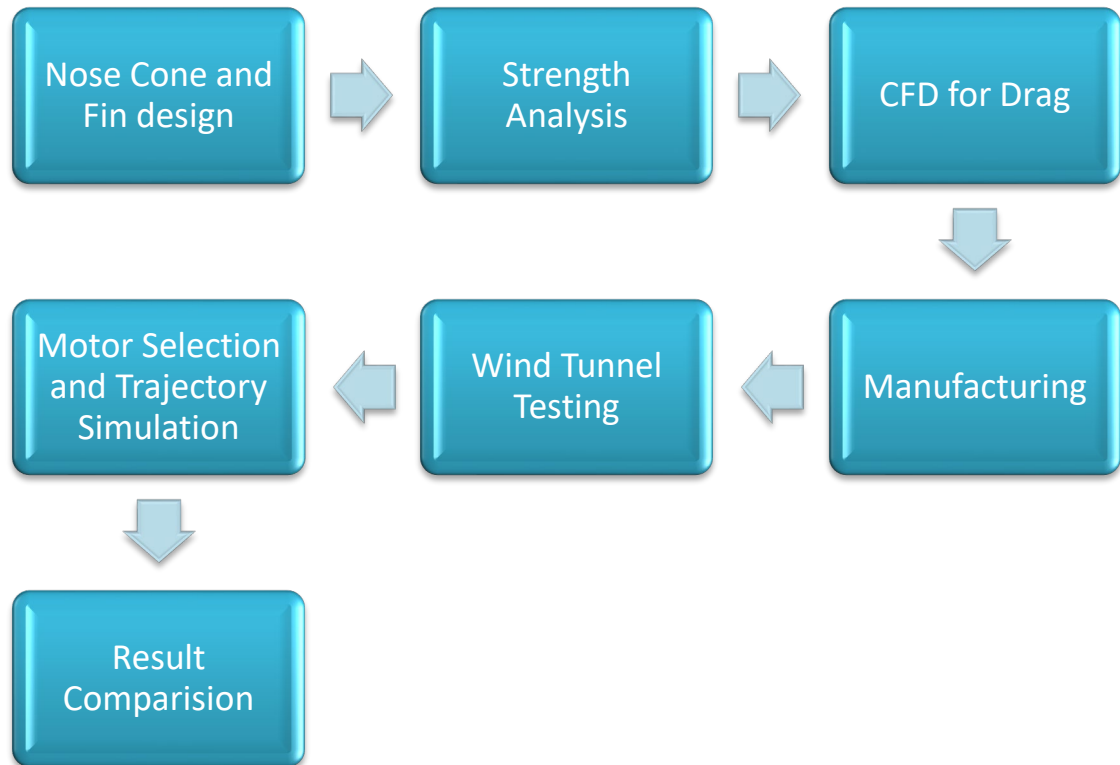
free-flight phase that is due to the fluctuations in wind profile.[5]

NAR Research & Development report describes a series of experiments using a computerized wind tunnel to determine the drag coefficients of typical model rocket designs. The main goal of this report is to derive a practical list of drag coefficients to improve the usefulness of existing altitude prediction software[8]. To verify the accuracy of the data, the predicted altitudes are compared to actual tracked altitudes for a sample of the models tested.

The Spalart Allmaras (One Equation method) turbulence model was adopted for the CFD analysis using Ansys Fluent Workbench. P. Spalart explained the trends in turbulence modelling and presents a new transport equation for the turbulent viscosity which was assembled, using empiricism and arguments of dimensional analysis, Galilean invariance, and selective dependence on the molecular viscosity. It has similarities with the models of Kovasznay, Secundov *et al.* The equation includes a destruction term that depends on the distance to the wall, related to the one in Secundov's model and to one due to Hunt. Unlike early one-equation models the resulting turbulence model is local (i.e. the equation at one point does not depend on the solution at other points), and therefore compatible with –rids of any structure and Navier Stokes solvers in two or three dimensions. It is numerically forgiving, in terms of near-wall resolution and stillness, and yields fairly rapid convergence to steady state. The wall and free stream boundary conditions are trivial. The model yields relatively smooth laminar turbulent at points specified by the user. It is powerful enough to be calibrated on 2-D mixing layers, wake, and flat plate boundary layers, which we consider to be the building blocks for aerodynamic flows. It yields satisfactory predictions of boundary layers in pressure gradients.[12]

Different exist methods for measuring the drag force in wind tunnels have been presented in the Master's Thesis of Li Nan of Gavle University[15]. We have used the Load Cell Method for our wind tunnel drag measurement. Muller explained the theory and application of the load cell for the measurement of aerodynamic forces and the working principle of the strain gauge.

Chapter 3 Methodology



Chapter 4 Design of Model Rocket

4.1 Nose Cone

The nose cone is usually the part that first interacts with the air in flight. The nose cone parts the air as the rocket moves through the atmosphere. Pushing the air out of the rocket's way creates friction. The friction pushes the nose cone downward, transferring the air's force to the airframe. The rocket's passage through the air creates heat of friction; for most model rockets, this heat is trivial, because it lasts only for a few seconds. However, for a slender high-powered rocket that flies several miles high at twice the speed of sound, the heat of friction can melt the paint off the nose cone. While a practical nose cone used in modelling usually includes a shoulder for mounting to a tube, that aspect will be ignored here, as it has no aerodynamic effects, and its mass and inertial contributions are easily handled separately. The nose cone usually has a shoulder, a section that fits inside the airframe to keep it centered on the rocket and a place on the bottom where the recovery system can be attached. This attachment varies in size with the rocket: model rockets have a plastic slot or small screw eye, while mid-sized rockets often have eyebolts, and large rockets have massive U-bolts or welded eyebolts. Once the recovery system has been attached, the nose cone stays joined to the rocket throughout the flight.

Nose cones are made from many different materials as well: small ones are often molded plastic or wood (balsa wood, basswood); high power manufacturers frequently use fiberglass, carbon fiber, or other composite materials. Composite provide greater strength and less weight, but cost more to purchase. Some people make their own nose cones from wood turned in a drill chuck or wood lathe. Frequently, nose cones on large home-built rockets are constructed by stacking and gluing sheets of foam, shaping the stack with a hot wire cutter, then covering the foam with a fiberglass/carbon fiber skin. The variations are endless. Nose cones come in many shapes: rounded, elliptical, parabolic, ogive, and conical, to name a few. We will describe a few important ones with relevant equations.

In all of the following nosecone shape equations, L is the overall length of the nosecone, and R is the radius of the base of the nosecone. y is the radius at any point x , as x varies from 0, at the tip of the nosecone, to L . The equations define the 2-dimensional profile of the nose shape. The full body of revolution of the nosecone is formed by rotating the profile around the centerline (C/L). [14]

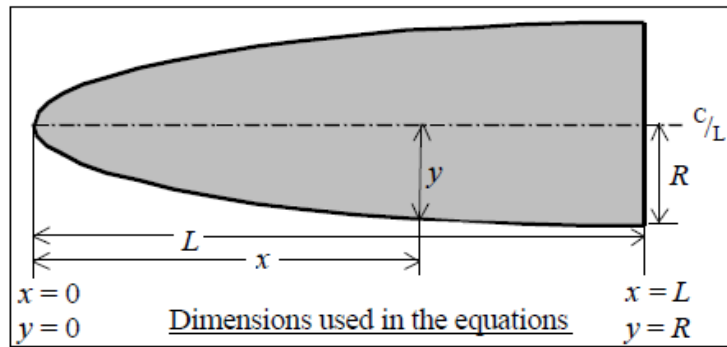


Figure 3- Nose Cone basic dimensions

4.1.1 Nose cone shape equations

4.1.1.1 Conical

A very common nose cone shape is a simple cone. This shape is often chosen for its ease of manufacture and is also often (mis)chosen for its drag characteristics. The sides of a conical profile are straight lines, so the diameter equation is simply,

$$y = \frac{xR}{L}$$

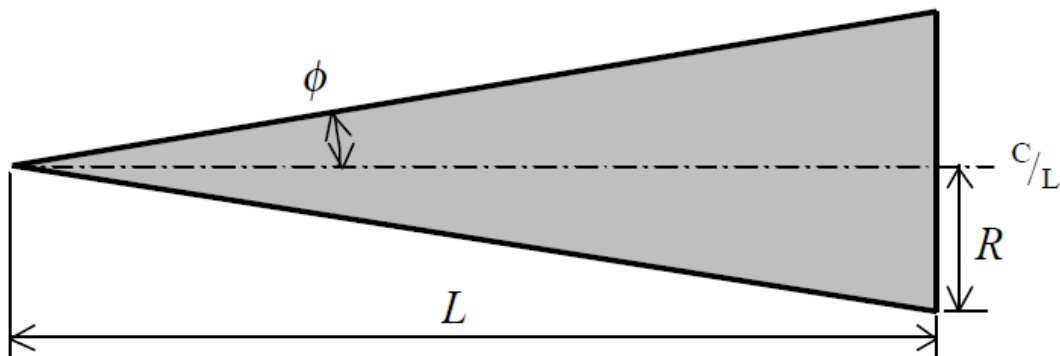


Figure 4- Conical Nose

Cones are also special cases of the Power and Parabolic series.

4.1.1.2 Tangent ogive

Next to a simple cone, the Tangent Ogive shape is the most familiar in hobby rocketry. The profile of this shape is formed by a segment of a circle such that the rocket body is tangent to the curve of the nosecone at its base; and the base is on the radius of the circle. The popularity of this shape is largely due to the ease of constructing its profile, since that profile is just a segment of a circle that can be simply drawn with a compass. The radius of the circle that forms the ogive is called the Ogive Radius, r , and it is related to the length and base radius of the nose cone:

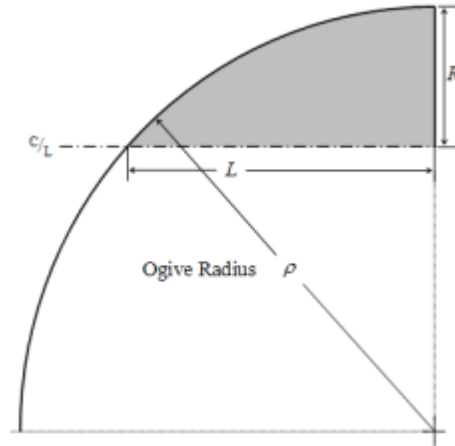


Figure 5- Tangent Ogive Nose

The radius y at any point x , as x varies from 0 to L is:

$$y = \sqrt{\rho^2 - (x - L)^2} + (R - \rho)$$

The nosecone length, L , must be equal to, or less than the Ogive Radius. If they are equal, then the shape is a hemisphere.[18]

4.1.1.3 Haack series

The shape is mathematically derived for the purpose of minimizing drag. While the series is a continuous set of shapes determined by the value of C in the equations below, two values of C have particular significance. When $C=0$, the notation 'LD' signifies minimum drag for the given length and diameter, and when $C=1/3$, 'LV' indicates minimum drag for a given length and volume. Note that the Haack series nose cones are not perfectly tangent to the body at their base, however the discontinuity is usually so slight as to be imperceptible. Likewise, the Haack nose tips do not come to a sharp point, but are slightly rounded.[18]

$$\theta = \cos^{-1}\left(1 - \frac{2x}{L}\right) \quad y = \frac{R \sqrt{\theta - \frac{\sin(2\theta)}{2} + C \sin^3 \theta}}{\sqrt{\pi}}$$

Where: $C = 1/3$ for **LV-HAACK** $C = 0$ for **LD-HAACK**

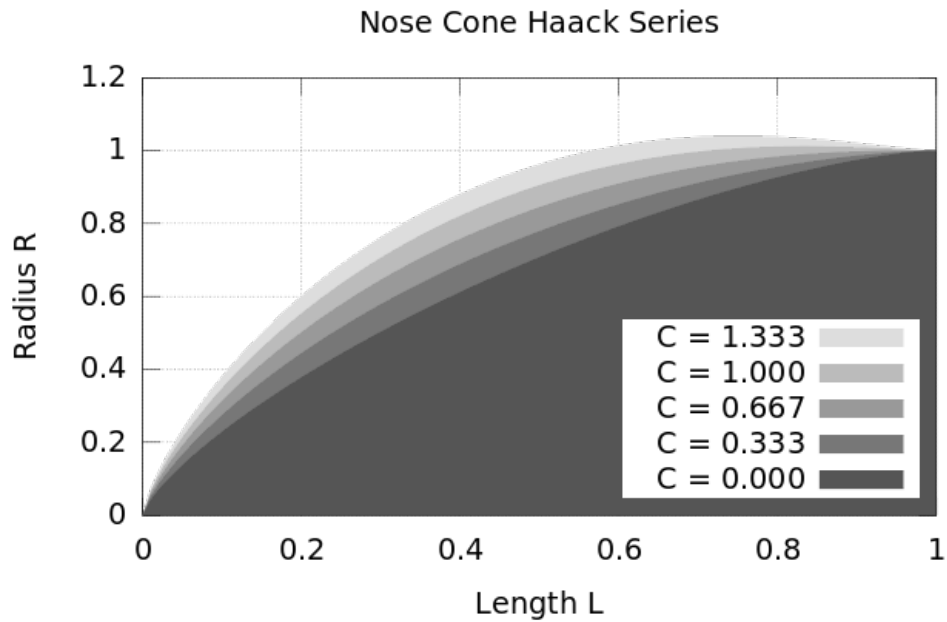


Figure 6- Haack Series

4.1.2 Nose cone drag characteristics

Below Mach .8, the nose pressure drag is essentially zero for all shapes. The major significant factor is friction drag, which is largely dependent upon the wetted area, the surface smoothness of that area, and the presence of any discontinuities in the shape. In strictly subsonic model rockets, a short, blunt, smooth elliptical shape is usually best. In the transonic region and beyond, where the pressure drag increases dramatically, the effect of nose shape on drag becomes highly significant. The factors influencing the pressure drag are the general shape of the nosecone, its fineness ratio, and its bluntness ratio.

4.1.2.2 Wetted Area

The wetted area is the total surface area of the nosecone shape that is exposed to the airflow. This does not include the base area of the nosecone. Friction drag on the rocket will depend upon the total wetted area. Equations for determining wetted area are provided in the appendix, but for a quick comparison, the following table compares the wetted areas for nosecone shapes of a similar 4:1 fineness ratio.

4.1.2.3 General Shape

Many of the references contain empirical data comparing the drag characteristics of various nose shapes in different flight regimes. The chart below, from reference 4, seems to be the most comprehensive and useful compilation of data for the flight regime of greatest interest.

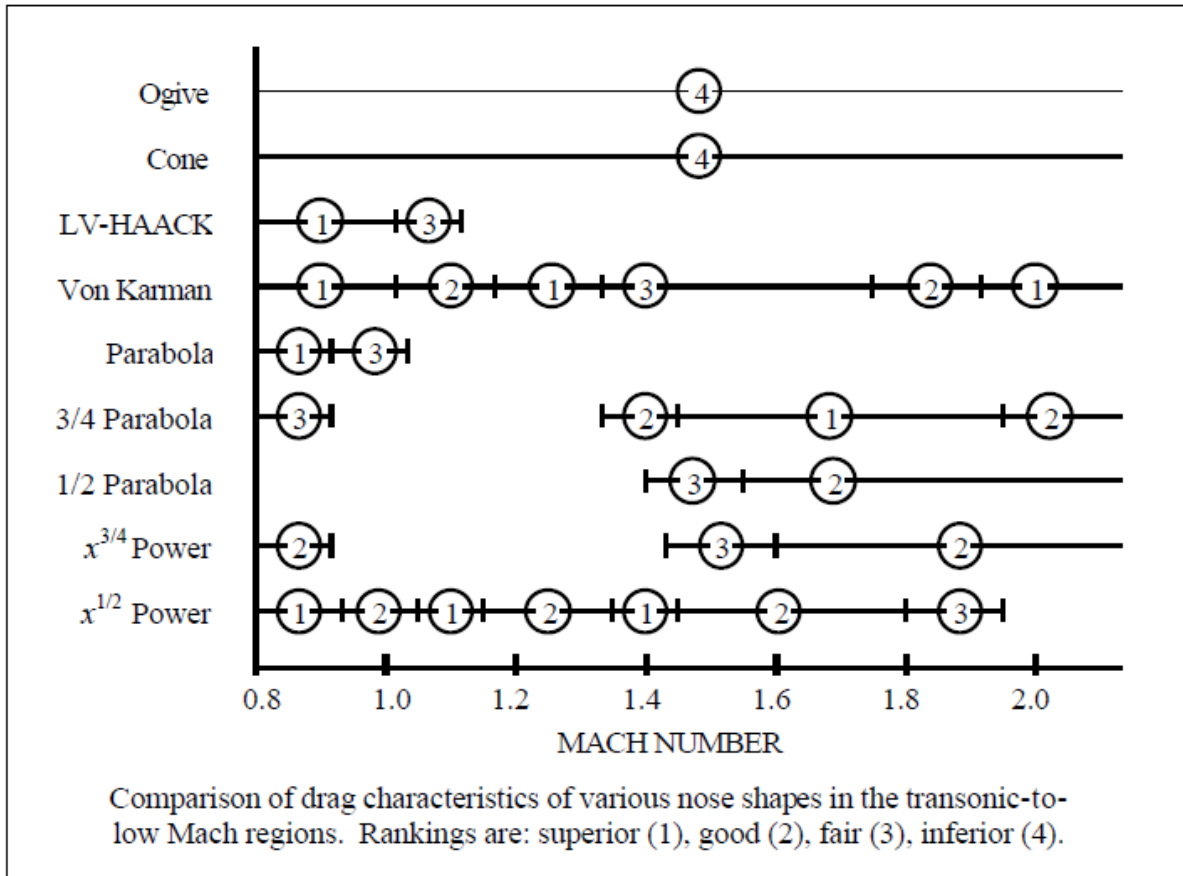


Figure 7- Comparison of drag characteristics of nose cone vs. Mach number

Many high-power and amateur rockets are striving to accomplish the goal of “Mach-busting”. Therefore their greatest concern is flight performance in the transonic region from 0.8 to 1.2 Mach, and nosecone shapes should be chosen with that in mind. Although data is not available for many shapes in the transonic region, the table clearly suggests that either the Von Karman shape, or Power Series shape with $n = \frac{1}{2}$, would be preferable to the popular Conical or Ogive shapes, for this purpose.

4.1.2.4 Bluffness Ratio

While most of the nosecone shapes ideally come to a sharp tip, they are often blunted to some degree as a practical matter for ease of manufacturing, resistance to handling and flight damage, and safety. This blunting is most often specified as a hemispherical ‘tip diameter’ of the nosecone. The term ‘Bluffness Ratio’ is often used to describe a blunted tip, and is equal to the tip diameter divided by the base diameter. Fortunately, there is little or no drag increase for slight blunting of a sharp nose shape. In fact, for constant overall lengths, there is a decrease in drag for bluffness ratios of up to 0.2, with an optimum at about 0.15. A flat truncation of a nose tip is known as a Meplat diameter, and the drag reduction effect of a Meplat truncation is shown in the diagram below.

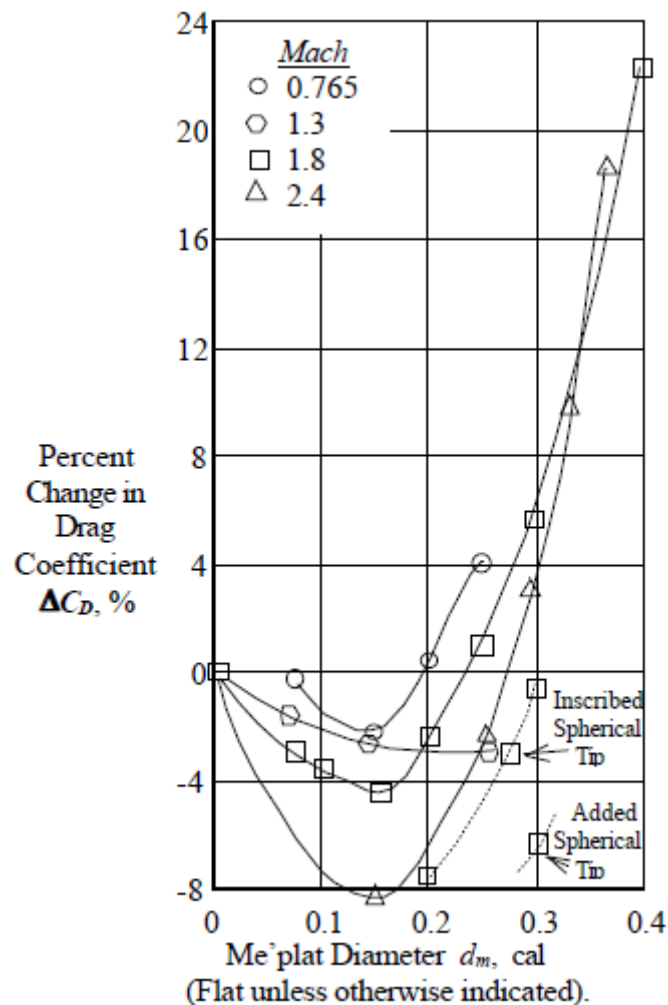


Figure 8- Percentage change in drag vs Diameter

4.2 Airframe

Sometimes called the body tube, the airframe provides the main structure of the rocket, supporting the nose cone and fins. During flight, the airframe is compressed from two directions: the motor pushes up from the bottom during thrust, and the nose cone pushes down as it parts the air. If the rocket veers to one side during flight, the airframe can be hit with air resistance from the side as well. The airframe must be able to withstand these forces, or the rocket will buckle or break. On the inside, the airframe holds the recovery system the parachute, streamer, etc as well as other parts, such as centering rings or motor tubes, some of which can also strengthen the airframe.

Rolled paper tubes are the most common form of airframe. They are relatively light-weight, and they crush during a crash to absorb the impact force (a desirable safety characteristic). In high power rocketry, airframes shift to heavy-walled paper, phenolic tubing (paper soaked in a high-temperature phenolic resin), plastic tubing, fiberglass, carbon fiber and other materials. These advanced materials handle the greater forces imposed by high-thrust motors and greater weights. Some builders reinforce their airframes by covering them with epoxy and fiberglass or other ma-

terials. For scratch-built rockets, paper tubes of all sizes, from paper towel rolls to oatmeal containers to concrete forming tubes, have been employed as airframes. Most airframe tubes can be purchased in lengths of 2-5 feet. When longer airframes are needed, a coupler is placed between two tubes to join them into one long tube. The coupler is usually made of the same material as the airframe. No matter the material, the goal is to create a structure that will not fold under the stresses of flight.

4.3 Fins

Fins provide the rocket's guidance. In flight, air flows over the fins, beginning at the leading edges and ending at the trailing edges. When a rocket is flying in a straight path, it encounters less air resistance (drag) than if it flies at an angle. If the rocket attempts to turn, the fins opposite the direction of turning are moved into the airflow, and the air pushes more on the exposed fin surfaces than on the other fins, until the rocket rights itself, just as a weather vane always points into the wind. Fins are usually the first part of a rocket to fail during powered flight, because they have air flowing around them on every side, and they are made of thin material to reduce their air resistance and weight. Fins can fail because they literally flutter apart, or can simply separate from the rocket because they are not properly attached. In either case, failure of one fin usually dooms the flight, as its guidance system is now unbalanced, and there is less air resistance near the tail once the fin is gone, moving the center of pressure (CP) forward, perhaps even ahead of the center of gravity (CG). At this point, the rocket often does a few quick loops to celebrate the loss, likely tearing the airframe apart in the process.

Because thin fins have less air resistance than thick ones, rigid materials are used to provide stiffness with minimum thickness. For model rockets, balsa wood or basswood are favorites. Competition models may use waferglass, a thin material made from plastics and fiberglass called G-10 (garolite). High power rockets use aircraft-grade birch plywood or thicker sheets of G-10. These materials are strong, but they become very heavy as their size and thickness increases. To reduce the weight of large fins, some builders use lighter materials for the center (core) such as foam or balsa wood, add hardwood strips for the fin edges, then reinforce the core with a skin of thin hardwood or composites, such as fiberglass/epoxy. If built properly, these reinforced fins can perform as well as solid fins, but with a fraction of the weight. [16]

Whatever material is chosen, the fins must be secured to the rocket at their root edges, so they will not separate from the airframe during the most stressful part of the flight (usually at motor burnout), sometimes at speeds beyond 1000 miles per hour. Model rocket fins are usually just glued to the airframe surface, while high power rockets often have fins with tabs that fit through slots cut in the airframe; the tabs are glued to the motor tube. These are known as through-the-wall fins; they gain strength by being glued both to the motor tube and to the airframe. There are many other techniques to strengthen fin attachment that you will find as you progress in the hobby. No matter the technique, the goal is to keep the fins attached to the rocket throughout the flight.

Deciding on what shape and cross-section to use for fins requires some insight into induced drag which was mentioned earlier.

[Ref: Apogee Newsletter #442]

Induced Drag occurs at the tips of the fins (the portion of the fin that is furthest away from the body of the rocket). What happens is that air flows around the corner of the tip edge from the underside to the top side (from the high pressure side to low pressure side). This is not the direction we want the air to flow. We just want the flow of air be parallel to the direction of the rocket in order to generate lift force to stabilize the rocket. But since it flows around the tip, it now has a perpendicular flow direction. In mixes with the parallel flow closer to the rocket body tube, and causes a swirling motion.

Whenever you change the direction of a moving object, a force is needed. The object in this case is the air molecules in the wind. Since the wind went from flowing parallel to the rocket to perpendicular to the rocket, a force must have acted on the rocket. We call this force “drag,” because it slows down the speed of the rocket. And it has the special name of “Induced Drag” since it occurred because a lift force was the root cause of the air flowing from one side of the fin to the other. Without a lift force, there would be no extra drag force, and hence no induced drag.

Elliptical fin has the lowest induced drag because the shape of the fin orients more of the lift force closer to the body tube of the rocket as the fin is longer near the body tube. That means there is less of a lift force created near the tip of the fin because the fin is shorter in that section of the fin. Because there is less lift near the tip of the fin, the difference in pressure (comparing the pressure on the top surface to that on the the bottom surface) is a lot lower near the tip. So less air flows around the tip. Hence, the induced drag force is lower.

A bigger contributor to the drag is not the shape, but the cross section of the fin. The best cross-section/airfoil for a typical low speed rocket like what most people fly has a symmetrical teardrop shape. This means it has a rounded front edge, and reduces to a knife edge at the rear of the cross section. The airfoil needs to have consistent thickness-to-length ratio on both the root edge and the tip edge. This is a very hard airfoil to sand, because the fin tapers in thickness from the root edge against the tube to the tip end away from the rocket. This is called a “radial taper”. If thickness-to-length ratio varies from the root to the tip, the airfoil changes too. This has the effect of moving more of the lift towards the tip of the fin. And this is like changing the fin to an axe-hammer shape, which is worse for induced drag.

Adding a proper airfoil and radial taper to the rocket fins' cross-section can take a little additional effort but it's worth it. Most model rocket kit photos today show the rocket fins with “squared” edges perhaps because it's the easiest and fastest to do. However, this is generally the worse fin design to use.

Simply “rounding” the fins' leading and trailing edges can reduce the “*fin drag*” dramatically (up to 75% less than squared edges). Moreover, adding an efficient airfoil shape to your fins can further reduce fin drag up to 85% less than that of round edge fins (or up to 96% less than fins with squared edges). This means the model rockets will fly faster and much higher.

The additional advantages of efficiently designed fins are

- The fins will produce more “lift force” at smaller angles of attack and so bring the rocket back on-course more quickly.
- The rockets will have superior dynamic response -They will recover from in-flight disturbances (such as wind shear) much more efficiently.

- The fins will produce much less induced drag - “Induced drag” is the additional drag produced when a fin is at an angle of attack (and thus generating a course correcting “lift” force in response to an in-flight disturbance).
- Because of lighter fins (when airfoiled), they have less mass at the tail end of the rocket which increases the stability margin of the model.
- The reduced tail mass also reduces the “longitudinal moment of inertia” of the rocket which (without getting technical) improves the rocket’s dynamics such that it corrects its course faster and more efficiently. [8]

4.3.1 Aspect Ratio

One final factor to consider when choosing (or designing) a fin plan-form shape is the fin Aspect Ratio. This parameter is basically an indication of the fin’s slenderness

The specific definition is

$$\text{Aspect Ratio} = \text{Fin Semi-Span}^2 / \text{Fin Area}$$

All else being equal, a higher aspect ratio fin will be more aerodynamically efficient than a lower aspect ratio fin. Here’s why. Air flowing close to the rocket body tube tends to be turbulent. Air farther away from the body tube tends to have smoother flow. Fins work most efficiently in smooth airflow. Since, generally speaking, higher aspect ratio fins “reach” into air farther away from the rocket’s body tube MORE of the fin ends up in SMOOTH airflow, increasing the fin’s overall efficiency! However there’s a catch. Higher aspect ratio fins tend to be structurally weaker than low aspect ratio fins. High speeds compound the problem! So again we must strike a balance between strength and aerodynamic efficiency. In each example, the light coloured fins have a higher Aspect Ratio (are more slender) than the dark coloured fins.

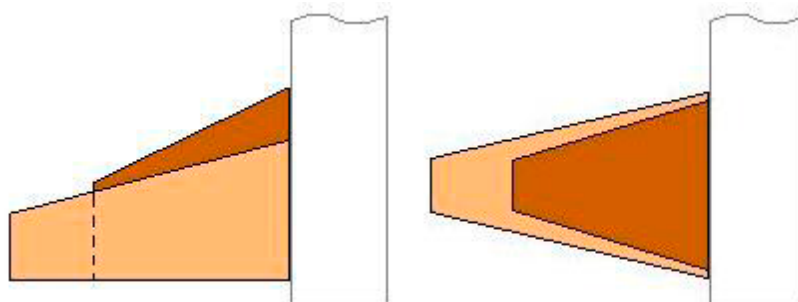


Figure 9- Aspect Ratio Illustrated

4.3.2 Airfoil Geometry

An airfoil is simply a curved or tapered fin cross section that allows air to travel over the

rocket fins much more easily. Below, we will review the basic geometry of model rocket airfoils and also illustrate some optimum airfoil cross sections for the benefit of rocketeers who wish to maximize flight performance of any model rocket. Asymmetrical airfoils produce lift even at zero angles of attack which is undesirable for Model Rockets as shown below:

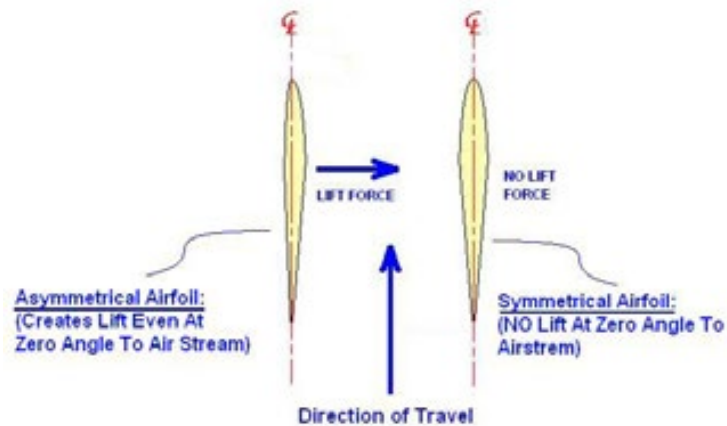


Figure 10- Effect of lift force on fin geometry

Airfoil Geometry: The most basic requirement for model rocket fin airfoils is that their cross sections must be as symmetrical as possible. This is necessary so that no lift forces are generated when the rocket is at zero angle of attack (in other words, if our model rocket is flying straight, we don't want the fins providing any side forces that may cause the rocket to spin and/or deviate from its course). A second basic requirement, when making fin airfoils, is that all of the model rocket's fins should be as identical as possible. Differences in fin geometry from one fin to the next on the same model rocket can create "asymmetrical drag" which may cause the rocket to spin and/or deviate from its proper course. Any spinning or deviation from straight vertical flight robs the rocket of energy (in the form of increased drag and/or non-vertical flight) which would otherwise go towards increasing the rocket's altitude (and speed).

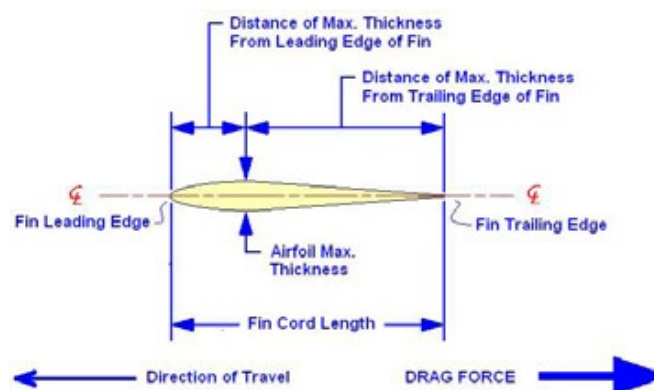


Figure 11- Typical Airfoil Cross Section

4.3.3 Fin Flutter

Flutter is a critical mode of failure arising due to aero-elastic instability. Phenomenon of flutter is concerned with surfaces subjected to aerodynamic drag and lift forces. Flutter is a complex phenomenon where structural modes are simultaneously coupled and excited by aerodynamic loads. In a more formal way, flutter is the condition where an aircraft component exhibits a self-sustained oscillatory behaviour at speeds higher than the critical one. This phenomenon occurs beyond a certain fixed speed of wind known as the critical speed. Flutter is an aero-elastic instability involving 2 degrees of freedom (DoF), one flap-wise and one torsional. The torsional structural mode (usually the first mode) couples with a flap-wise bending mode in a flutter mode through the aerodynamic forces. The aerodynamic forces lead to torsion of the structure. The torsion changes the angle of attack and thus the aerodynamic lift force. Flutter occurs when the change of angle of attack due to torsion changes the lift in an unfavourable phase with the flap wise bending. Flutter causes violent vibrations with rapidly growing amplitude. The flutter mode has a highly negative damping which cannot be compensated by the structural damping. Flutter only happens above a certain relative wind speed on the structure, known as the critical flutter speed.

Large rockets with huge fins can indeed fail due to flutter. But model rockets being small in size with small wing-spans, flutter can rarely occur in these systems. Nevertheless, its important to prove that the phenomena need not be considered in our case, which is why, the team carried out calculation of critical flutter velocity.

Critical flutter velocity expression, as developed by Dennis. J Martin, National Advisory Committee for Aeronautics(NACA)[11] is given below:

$$V_f = a \sqrt{\frac{G}{\frac{1.337AR^3P(\lambda + 1)}{2(AR + 2)\left(\frac{t}{c}\right)^3}}$$

$$S = \frac{1}{2}(c_r + c_t)b$$

$$AR = \frac{b^2}{s}$$

$$T(^{\circ}\text{F}) = 59 - 0.00356h$$

$$\lambda = \frac{c_t}{c_r}$$

$$P(\text{lbs/ft}^2) = 2116 \times \left(\frac{T+459.7}{518.6}\right)^{5.256}$$

$$a = \sqrt{1.4 \times 1716.59 \times (T(^{\circ}\text{F}) + 460)}$$

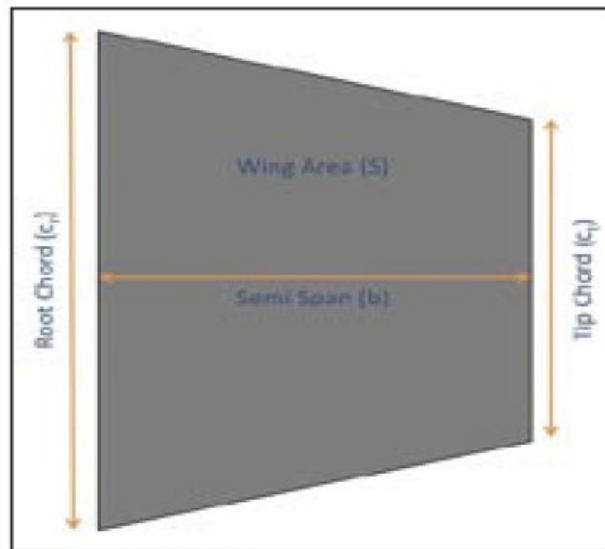


Figure 12- Fin Cross Section

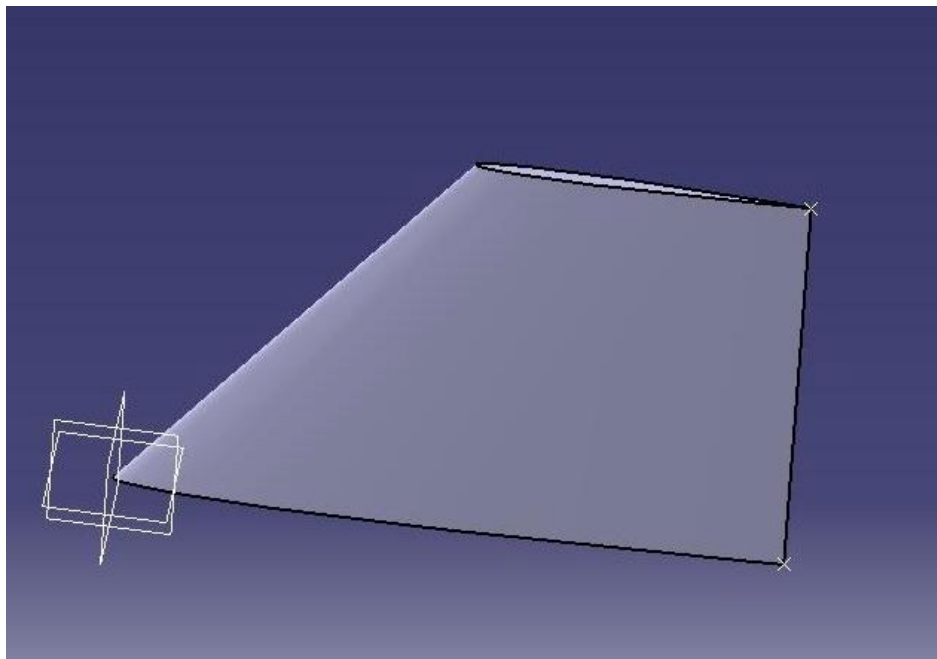


Figure 13- CAD model of fin

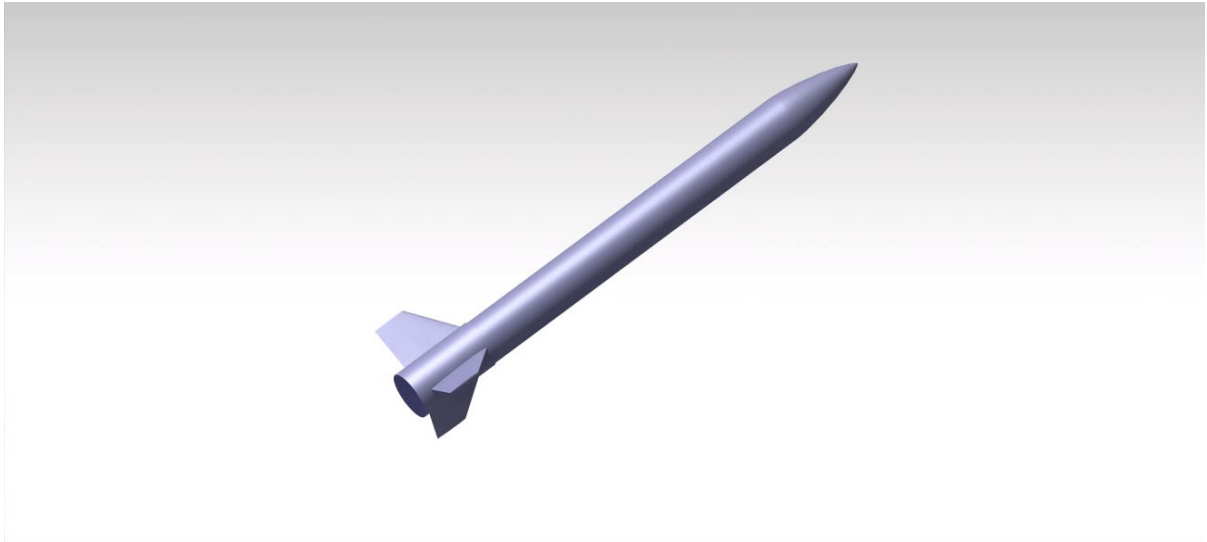


Figure 14- Solid CAD Model

Chapter 5 Simulation

5.1 Fluent

ANSYS Fluent software contains the broad physical modeling capabilities needed to model flow, turbulence, heat transfer, and reactions for industrial applications—ranging from air flow over an aircraft wing to combustion in a furnace, from blood flow to semiconductor manufacturing, and from clean room design to wastewater treatment plants. Fluent covers a broad reach, including special models with capabilities to model in-cylinder combustion, aero-acoustics, turbomachinery and multiphase systems.[8]

5.1.1 Geometry of Model Rocket

The geometry file was imported from CATIA to Ansys Workbench for the flow simulation of the rocket body.

5.1.1.1 Nose

The fineness ratio and Mach number wave drag were studied for the Von Karman nose cone.

5.1.1.1.1 Fineness ratio

It is the ratio of the length of a body to its maximum width. The main considerations are

- For any given fineness ratio, the hypersonic optimum shape has the least drag loss. The difference in drag loss between conical and hypersonic optimum shapes decreases however for increasing fineness ratio, and at 7:1 it has nearly vanished.
- For any given shape, altitude increases with increasing fineness ratio. However, the benefits of increasing fineness ratio beyond 5:1 are small.
- For fineness ratios up to 6:1, the hypersonic optimum shape yields the highest apogee - but for increasing fineness ratios, the conical shape gets increasingly closer in performance and at 7:1 fineness ratio, the conical shape yields the highest apogee, although only by a very small margin. The wave drag contribution decreases at higher fineness ratios, so skin friction becomes more important, and it becomes less important what shape has the least high.

5.1.1.1.2 Mach number wave drag.

Not surprisingly, the elliptical shape has poorer performance than the other shapes, but except from that, and perhaps the parabolic shape, the difference in apogee between the other shapes is so small for the higher fineness ratios, that other criteria may be taken into account when selecting the shape. A 5:1 fineness ratio may be chosen over 7:1 for practical reasons. The hypersonic optimum shape may be chosen

for performance, but the penalty for choosing a conical shape is negligible, and it would have the advantage of simplicity. Also there are the thermal considerations. In general, the aerodynamic heating increases with the equivalent nose vertex angle, so a conical nose would be expected to have higher temperature at the base than the hypersonic optimum but lower temperature at the tip. However, the temperature rise also depends on the local heat capacity, and the tip of a hypersonic optimum nose can have a larger heat capacity so it may still have the lowest skin temperature overall. A blunted cone could be a reasonable way of approximating the hypersonic optimum shape while keeping the simplicity of a cone.

5.1.1.2 Fins

The early NACA airfoil series, the 4-digit, 5-digit, and modified 4-/5-digit, were generated using analytical equations that describe the camber (curvature) of the mean-line (geometric centerline) of the airfoil section as well as the section's thickness distribution along the length of the airfoil. Later families, including the 6-Series, are more complicated shapes derived using theoretical rather than geometrical methods. Before the National Advisory Committee for Aeronautics (NACA) developed these series, airfoil design was rather arbitrary with nothing to guide the designer except past experience with known shapes and experimentation with modifications to those shapes.

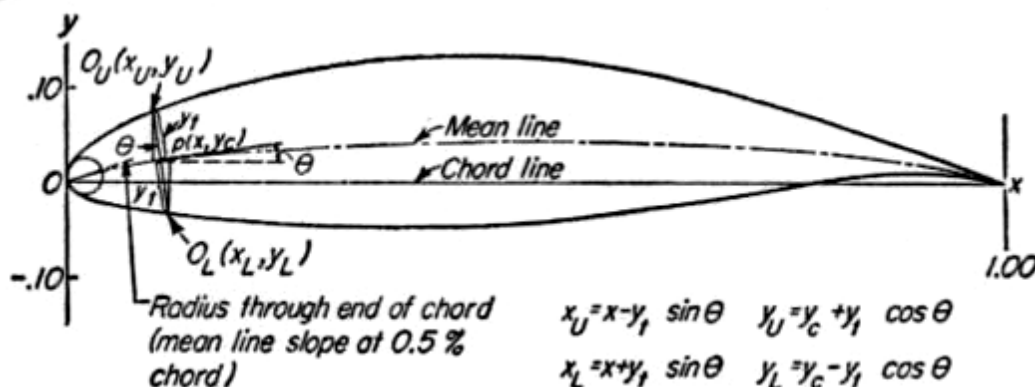


Figure 15- NACA airfoil geometrical construction

NACA Four-Digit Series:

The first family of airfoils designed using this approach became known as the NACA Four-Digit Series. The first digit specifies the maximum camber (m) in percentage of the chord (airfoil length), the second indicates the position of the maximum camber (p) in tenths of chord, and the last two numbers provide the maximum thickness (t) of the airfoil in percentage of chord. For example, the NACA 2415 airfoil has a maximum thickness of 15% with a camber of 2% located 40% back from the airfoil leading edge

Family	Advantages	Disadvantages	Applications
4-Digit	1. Good stall characteristics 2. Small center of pressure movement across large speed range 3. Roughness has little effect	1. Low maximum lift coefficient 2. Relatively high drag 3. High pitching moment	1. General aviation 2. Horizontal tails Symmetrical: 3. Supersonic jets 4. Helicopter blades 5. Shrouds 6. Missile/rocket fins

5.1.2 Mesh Generation

Face Sizing option was used to generate a tetrahedral mesh for different parts of the model rocket.

5.1.2.1 Meshing Method

The **Method** control determines whether an assembly meshing algorithm will be used and filters user interface components appropriately. The following options are available:

- **None** - This is the default. Assembly meshing will not be used and controls are not exposed.
- **Cut Cell** - Selects the Cut Cell strategy for assembly meshing. Exposes assembly meshing controls and hides controls that are not applicable to assembly meshing. The Cut Cell option is supported only in the Meshing application.
- **Tetrahedrons** - Selects the Tetrahedrons strategy for assembly meshing. Exposes assembly meshing controls and hides controls that are not applicable to assembly meshing. The Tetrahedrons option is available and supported in the Meshing application. Tetrahedrons are also available in the Mechanical application; however, meshes generated using assembly meshing are not supported for Mechanical solvers. If you try to use a Mechanical solver to solve an analysis of an assembly mesh, the solution is blocked and an error message is issued. To proceed using a Mechanical solver, you must first deactivate assembly meshing (set Method to None) and then regenerate the mesh.

Display	
Display Style	Body Color
Defaults	
Physics Preference	CFD
Solver Preference	Fluent
<input type="checkbox"/> Relevance	0
Export Format	Standard
Sizing	
Size Function	Curvature
Relevance Center	Fine
<input type="checkbox"/> Curvature Normal Angle	20.0 °
<input type="checkbox"/> Min Size	0.70 mm
<input type="checkbox"/> Max Tet Size	44.80 mm
<input type="checkbox"/> Growth Rate	1.10
Minimum Edge Length	1.0 mm
Quality	
Check Mesh Quality	Yes, Errors and Warnings
Smoothing	High
Mesh Metric	Element Quality
<input type="checkbox"/> Min	0.21398
<input type="checkbox"/> Max	0.99984
<input type="checkbox"/> Average	0.84362
<input type="checkbox"/> Standard Deviation	9.5736e-002
Inflation	
Use Automatic Inflation	Program Controlled
Inflation Option	First Layer Thickness
<input type="checkbox"/> First Layer Height	4. mm
<input type="checkbox"/> Maximum Layers	4
<input type="checkbox"/> Growth Rate	1
View Advanced Options	Yes
Collision Avoidance	None
<input type="checkbox"/> Maximum Height over Base	1
Growth Rate Type	Geometric
<input type="checkbox"/> Maximum Angle	180.0 °
<input type="checkbox"/> Fillet Ratio	1
Use Post Smoothing	Yes
<input type="checkbox"/> Smoothing Iterations	5
Assembly Meshing	
Method	Tetrahedrons
Feature Capture	Program Controlled
Tessellation Refinement	Program Controlled
Intersection Feature Creation	Program Controlled
Advanced	
Number of CPUs for Parallel Part Meshing	Program Controlled
Statistics	
<input type="checkbox"/> Nodes	205525
<input type="checkbox"/> Elements	1145048

Figure 16- Mesh details

5.1.2.2 Face Sizing

The nosecone and body have an element size of 5mm

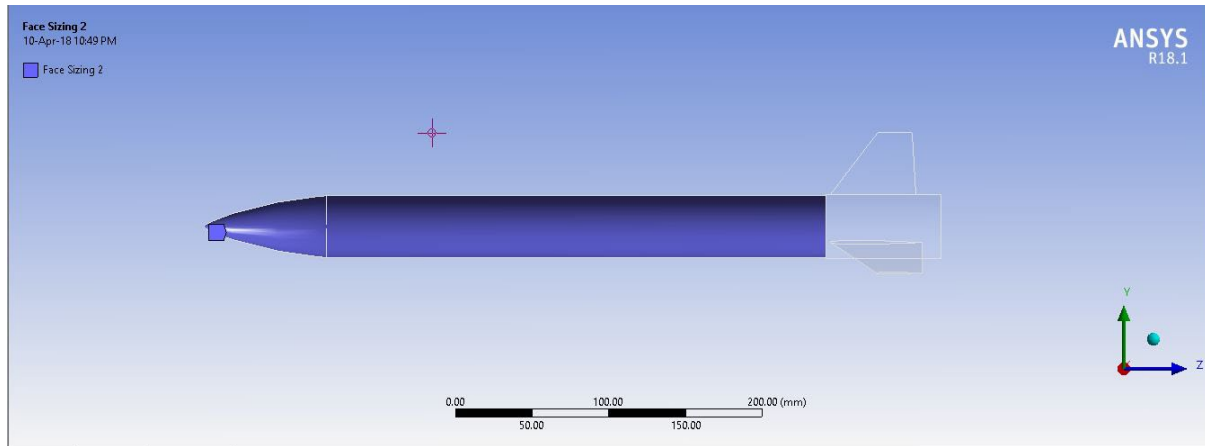


Figure 17- Face sizing of nose cone and body

The fins have an element size of 5mm

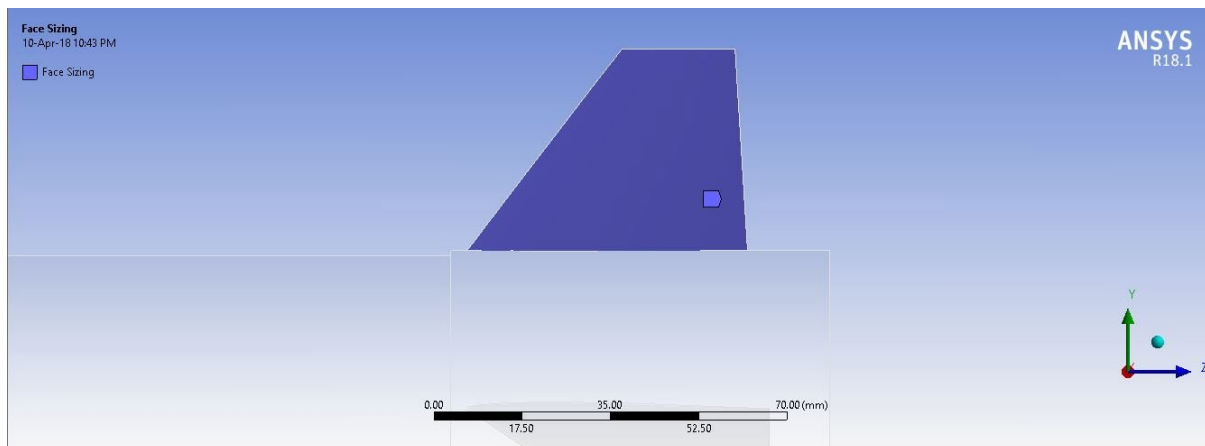


Figure 18- Face sizing of fins

The fin base has an element size of 1 mm

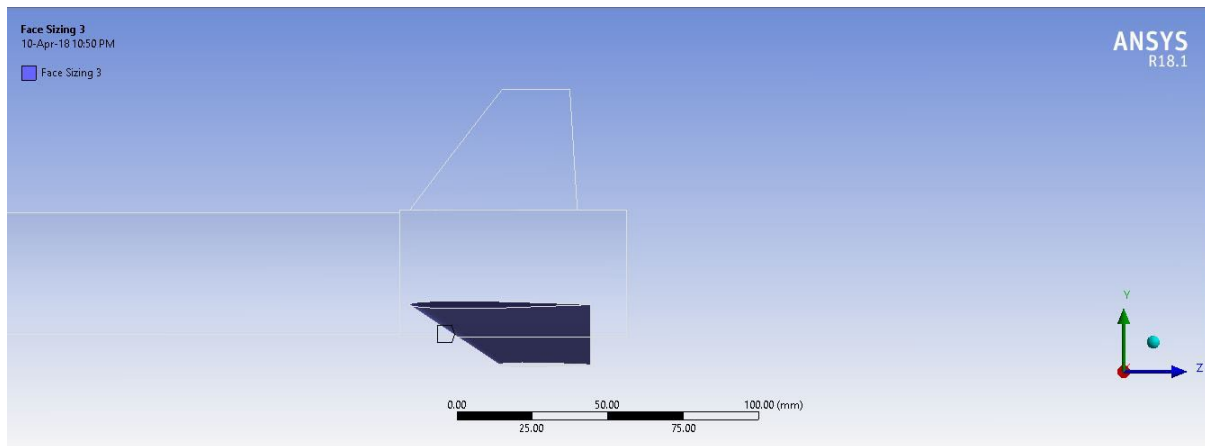


Figure 19- Face sizing of fin base

The element size for this part of the body is 3mm

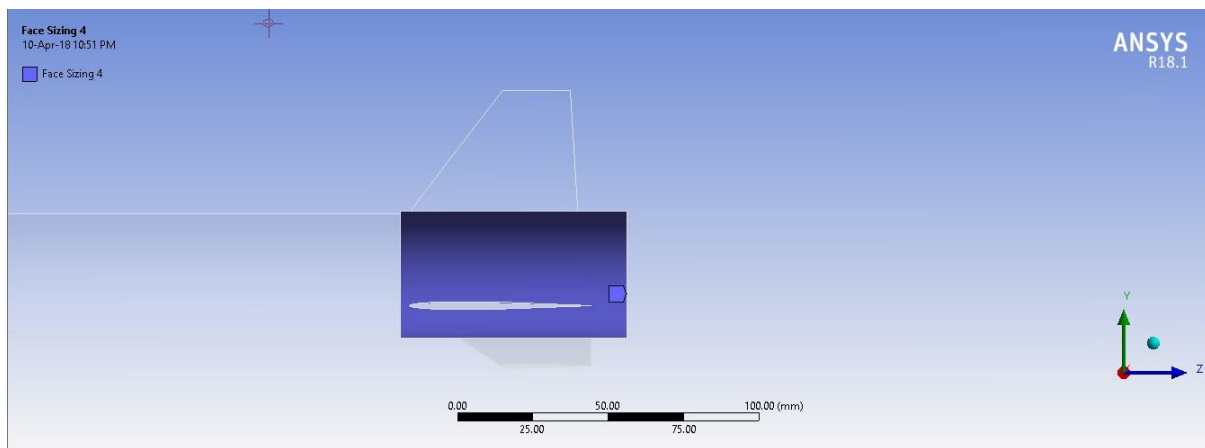


Figure 20- Face sizing of fin mount

The element size considered here is 0.5mm

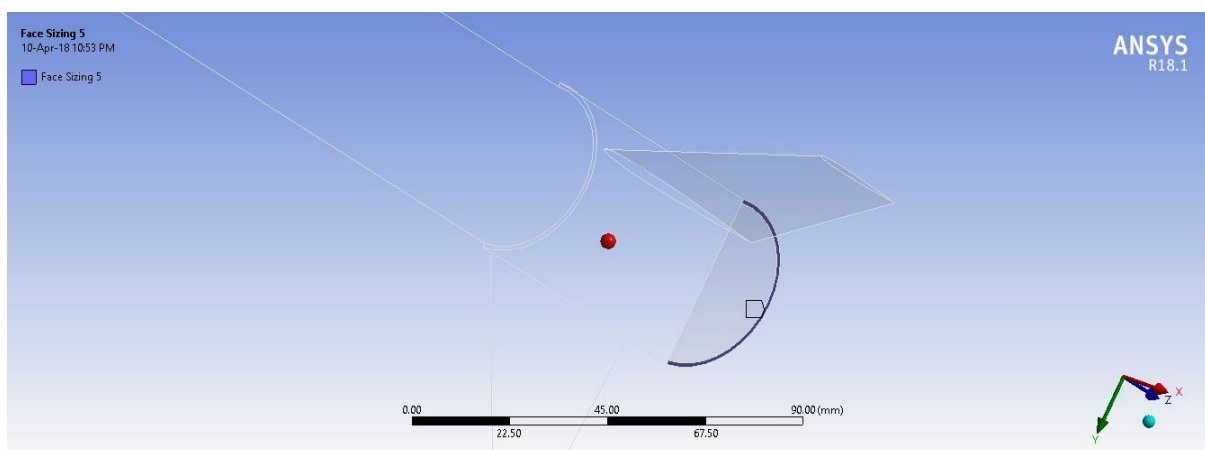


Figure 21- Face sizing of body part-1

The element size considered here is 0.7mm

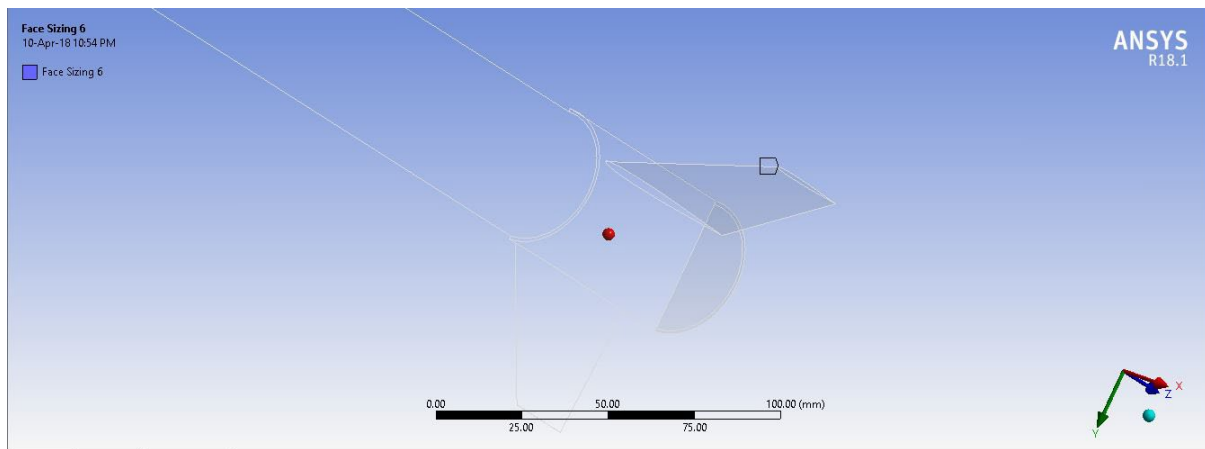


Figure 22- Face sizing of body part-2

5.1.2.3 Mesh Metric

The Element Quality option provides a composite quality metric that ranges between 0 and 1. This metric is based on the ratio of the volume to the sum of the square of the edge lengths for 2D quad/tri elements, or the square root of the cube of the sum of the square of the edge lengths for 3D elements. A value of 1 indicates a perfect cube or square while a value of 0 indicates that the element has a zero or negative volume.

This can also be expressed as follows:

- For two-dimensional quad/tri elements:

$$Quality = C \left(\frac{area}{\sum (EdgeLength)^2} \right)$$

- For three-dimensional brick elements:

$$Quality = C \left[\frac{volume}{\sqrt{[\sum (Edge length)^2]^3}} \right]$$

5.1.3 Setup

The density-based solver solves the governing equations of continuity, momentum, and (where appropriate) energy and species transport simultaneously (i.e., coupled together). Governing equations for additional scalars will be solved afterward and sequentially, several iterations of the solution loop must be performed before a converged solution is obtained. Each iteration consists of the steps illustrated in Figure and outlined below:

1. Update the fluid properties based on the current solution. (If the calculation has just begun, the fluid properties will be updated based on the initialized solution.)
2. Solve the continuity, momentum, and (where appropriate) energy and species

equations simultaneously.

3. Where appropriate, solve equations for scalars such as turbulence and radiation using the previously updated values of the other variables.

4. When interphase coupling is to be included, update the source terms in the appropriate continuous phase equations with a discrete phase trajectory calculation.

5. Check for convergence of the equation set.

These steps are continued until the convergence criteria are met.

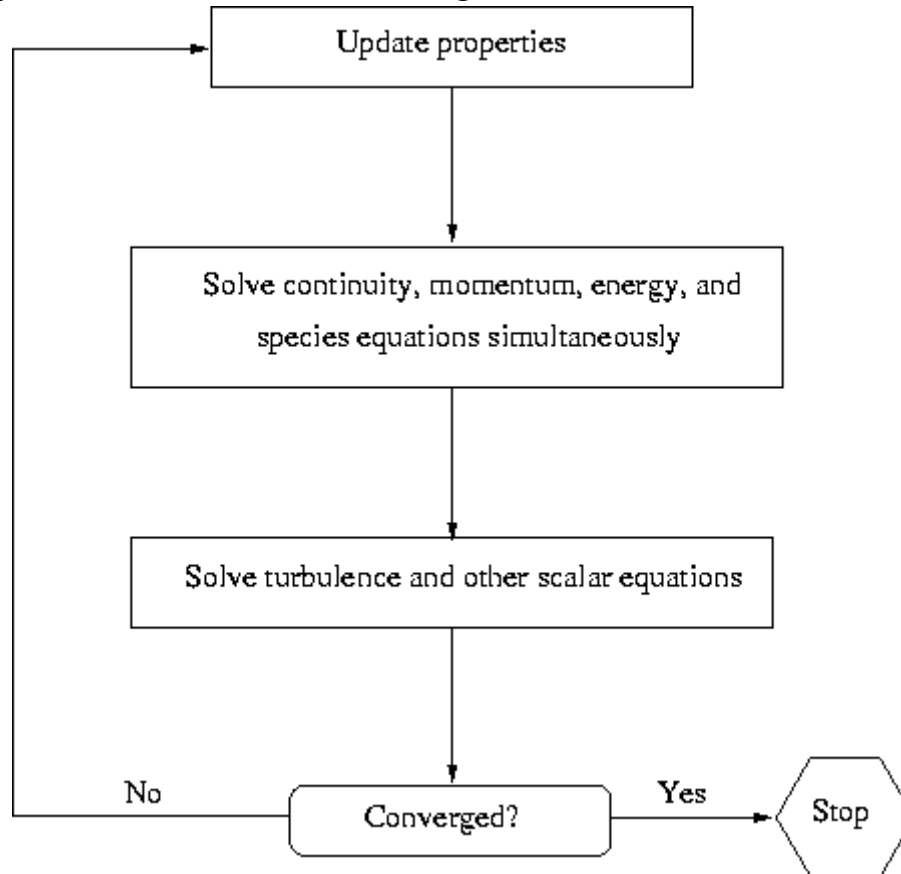


Figure 23- Overview of the Density-Based Solution Method

In the density-based solution method you can solve the coupled system of equations (continuity, momentum, energy and species equations if available) using, either the coupled-explicit formulation or the coupled-implicit formulation. The main distinction between the density-based explicit and implicit formulations is described next.

In the density-based solution methods the discrete, non-linear governing equations are linearized to produce a system of equations for the dependent variables in every computational cell. The resultant linear system is then solved to yield an updated flow-field solution.

5.1.4 Model

The Spalart-Allmaras model is a relatively simple one-equation model that solves a modeled transport equation for the kinematic eddy (turbulent) viscosity. This embodies a relatively new class of one-equation models in which it is not necessary to calculate a length scale related to the local shear layer thickness. The Spalart-

Allmaras model was designed specifically for aerospace applications involving wall-bounded flows and has been shown to give good results for boundary layers subjected to adverse pressure gradients. It is also gaining popularity in the turbomachinery applications. In its original form, the Spalart-Allmaras model is effectively a low-Reynolds-number model, requiring the viscosity-affected region of the boundary layer to be properly resolved. In ANSYS FLUENT, however, the Spalart-Allmaras model has been implemented to use wall functions when the mesh resolution is not sufficiently fine. the near-wall gradients of the transported variable in the model are much smaller than the gradients of the transported variables in the k - ϵ or k - ω models. This might make the model less sensitive to numerical errors when non-layered meshes are used near walls.

Original model

The turbulent eddy viscosity is given by

$$\nu_t = \tilde{\nu} f_{v1}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad \chi := \frac{\tilde{\nu}}{\nu}$$

Advantages

1. Computational efficiency: The standard k - $\hat{\mu}$ model is a classical model developed by turbulence researchers in the early 1970's, whereas the SA model is a recent model developed in the early 1990's with the objective of numerical efficiency and robustness. The SA model can perform much faster than the k - $\hat{\mu}$ model for the same or better level of accuracy.
2. Accuracy as Low-Re Model: Inherently, the SA model is effective as a low-Reynolds number model and provides a superior accuracy than the standard k - $\hat{\mu}$ model for wall-bounded and adverse pressure gradients flows in boundary layers. The k - $\hat{\mu}$ model does not perform well in boundary layers and requires additional terms to be added to the governing equations to produce boundary layer profiles. The standard k - $\hat{\mu}$ model involves a two equation coupled differential system, which can lead to stiff algebraic system for non-diffusive & accurate flow solver like AcuSolve. Some numerically dissipative solvers can easily handle such stiff differential equations. On the contrary, the SA model possesses a well-behaved one equation differential system.

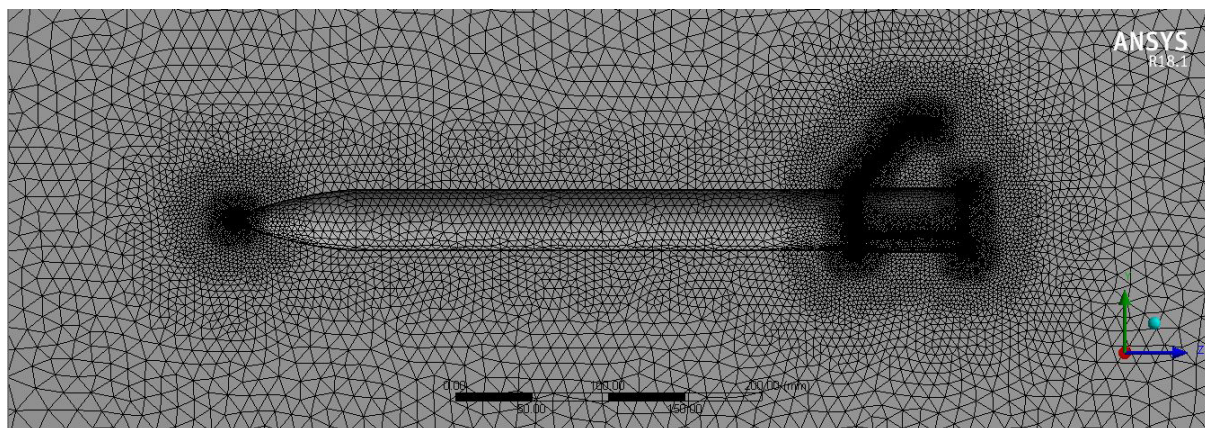


Figure 24- Full body mesh

5.1.5 Boundary conditions

Various boundary condition parameters have been specified below.

5.1.5.1 Turbulence Intensity

The turbulence intensity is defined as the ratio of the root-mean-square of the velocity fluctuations to the mean flow velocity. A turbulence intensity of 1% or less is generally considered low and turbulence intensities greater than 10% are considered high. Ideally, you will have a good estimate of the turbulence intensity at the inlet boundary from external, measured data. For example, if you are simulating a wind-tunnel experiment, the turbulence intensity in the free stream is usually available from the tunnel characteristics. In modern low-turbulence wind tunnels, the free-stream turbulence intensity may be as low as 0.05%.

For internal flows, the turbulence intensity at the inlets is totally dependent on the upstream history of the flow. If the flow upstream is under-developed and undisturbed, you can use a low turbulence intensity. If the flow is fully developed, the turbulence intensity may be as high as a few percent. The turbulence intensity at the core of a fully-developed duct flow can be estimated from the following formula derived from an empirical correlation for pipe flows:

$$I \equiv \frac{u'}{u_{\text{avg}}} = 0.16(\text{Re}_{D_H})^{-1/8}$$

At a Reynolds number of 50,000, for example, the turbulence intensity will be 4%, according to this formula.

5.1.5.2 Turbulence Length Scale and Hydraulic Diameter

The turbulence length scale, ℓ , is a physical quantity related to the size of the large eddies that contain the energy in turbulent flows.

In fully-developed duct flows, ℓ is restricted by the size of the duct, since the turbulent eddies cannot be larger than the duct. An approximate relationship between ℓ and the physical size of the duct is

$$\ell = 0.07L$$

where L is the relevant dimension of the duct. The factor of 0.07 is based on the maximum value of the mixing length in fully-developed turbulent pipe flow, where L is the diameter of the pipe. In a channel of non-circular cross-section, you can base on the hydraulic diameter. If the turbulence derives its characteristic length from an obstacle in the flow, such as a perforated plate, it is more appropriate to base the turbulence length scale on the characteristic length of the obstacle rather than on the duct size. The equation which relates a physical dimension to the turbulence

length scale is not necessarily applicable to all situations. For most cases, however, it is a suitable approximation.

Guidelines for choosing the characteristic length L or the turbulence length scale ℓ for selected flow types are listed below:

- For fully-developed internal flows, choose the Intensity and Hydraulic Diameter specification method and specify the hydraulic diameter $L = D_H$ in the Hydraulic Diameter field.
- For flows downstream of turning vanes, perforated plates, etc., choose the Intensity and Hydraulic Diameter method and specify the characteristic length of the flow opening for L in the Hydraulic Diameter field.
- For wall-bounded flows in which the inlets involve a turbulent boundary layer, choose the Intensity and Length Scale method and use the boundary-layer thickness, δ_{99} , to compute the turbulence length scale, ℓ , from $\ell = 0.4\delta_{99}$. Enter this value for ℓ in the Turbulence Length Scale field.

5.1.5.3 Turbulent Viscosity Ratio

The turbulent viscosity ratio, μ_t/μ , is directly proportional to the turbulent Reynolds number ($Re_t \equiv k^2/(\epsilon\nu)$). Re_t is large (on the order of 100 to 1000) in high-Reynolds-number boundary layers, shear layers, and fully-developed duct flows.

However, at the free-stream boundaries of most external flows, μ_t/μ is fairly small.

Typically, the turbulence parameters are set so that $1 < \mu_t/\mu < 10$.

To specify quantities in terms of the turbulent viscosity ratio, you can choose Turbulent Viscosity Ratio (for the Spalart-Allmaras model) or Intensity and Viscosity Ratio (for the k - ϵ models, the k - ω models, or the RSM).

5.1.6 SOLUTION METHODS

The manner in which the governing equations are linearized may take an "implicit" or "explicit" form with respect to the dependent variable (or set of variables) of interest. By implicit or explicit we mean the following:

- Implicit: For a given variable, the unknown value in each cell is computed using a relation that includes both existing and unknown values from neighboring cells. Therefore each unknown will appear in more than one equation in the system, and these equations must be solved simultaneously to give the unknown quantities.
- Explicit: For a given variable, the unknown value in each cell is computed using a relation that includes only existing values. Therefore each unknown will appear in only one equation in the system and the equations for the unknown value in each cell can be solved one at a time to give the unknown quantities.

In the density-based solution method you have a choice of using either an implicit or explicit linearization of the governing equations. This choice applies only to the coupled set of governing equations. Transport equations for additional scalars are solved segregated from the coupled set (such as turbulence, radiation, etc. If you choose the implicit option of the density-based solver, each equation in the coupled set of governing equations is linearized implicitly with respect to all dependent variables in the set. This will result in a system of linear equations with N equations for each cell in the domain, where N is the number of coupled equations in the set. Because there are N equations per cell, this is sometimes called a "block" system of equations.

A point implicit linear equation solver (Incomplete Lower Upper (ILU) factorization scheme or a symmetric block Gauss-Seidel) is used in conjunction with an algebraic multigrid (AMG) method to solve the resultant block system of equations for all N dependent variables in each cell. For example, linearization of the coupled continuity, x-, y-, z-momentum, and energy equation set will produce a system of equations in which p , u , v , w , and T are the unknowns. Simultaneous solution of this equation system (using the block AMG solver) yields at once updated pressure, u -, v -, w -velocity, and temperature fields.

In summary, the coupled implicit approach solves for all variables (p , u , v , w , T) in all cells at the same time.

5.2 Trajectory Prediction

$$a_y = \frac{T}{m} - g - \frac{\rho V^2 C_D A}{2m}$$

This equation was used to predict trajectory.[7] This is a non-linear differential equation because of the V^2 term. Also, the changing mass of the rocket gives it a non-linear character. It was decided that the equation should be solved using 4th order Runge-Kutta method[10]. Following thrust curve was obtained from OpenRocket for F-class Estes motor:

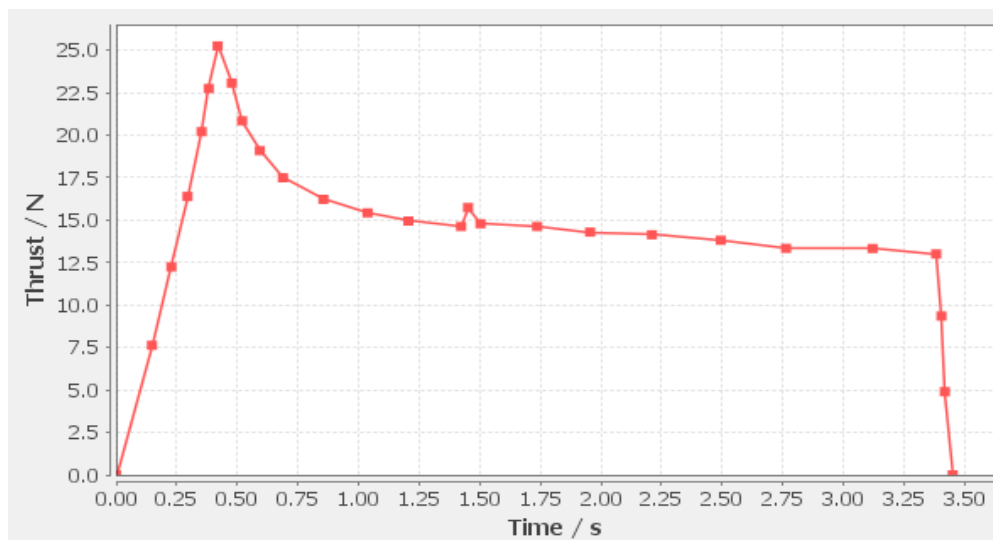


Figure 25- Thrust Curve of Estes F15

The thrust curve was divided into 2 regions, the linear region from 0-0.4s and the exponential curve fit from 0.4-3.4s. The mass ejected from the rocket was assumed proportional to the thrust force(F). Therefore,

$$F = (dm/dt) \times C$$

Total mass of propellant consumed was 60g. Hence, $\int_0^m dm = \int_0^{0.4} (F/C) dt + \int_{0.4}^{3.4} (F/C) dt = 0.06$

Hence, an effective velocity value(C) was calculated from the values of total mass consumed and thrust curves, the F/C equation were integrated to get the mass consumed at any time t.

The drag equation was used to calculate drag force with drag coefficient as 0.68(For smooth ellipsoidal 2:1 head). The mass of the rocket body, area of fins was all used from the preliminary design. The resulting differential equation was solved using 4th order Runge Kutta method, by writing a program on MATLAB (see Appendix). These programs gave values of velocity achieved by the rocket at different points of time. A curve was fit into these values of velocities by least square regression method (see Appendix for program) and the obtained equation was integrated to get displacement at any time t. The various displacement curves were plotted for their corresponding intervals to obtain the entire trajectory. Below is a plot of displacement v/s time for a vertical launch.

5.2.1 Displacement curve

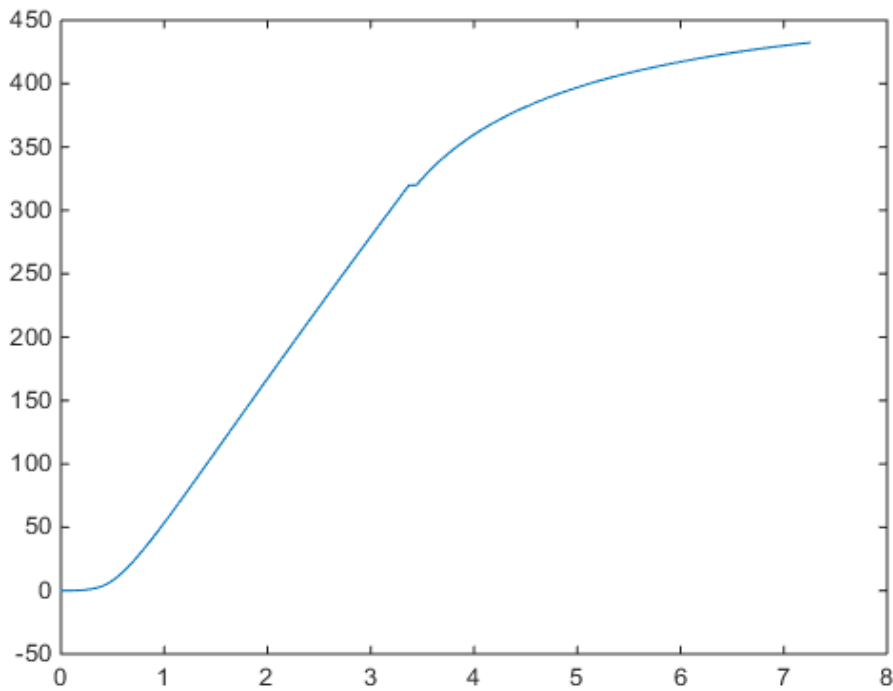


Figure 26- Displacement Curve for Vertical launch

Now the next case analysed was rocket launched at an angle 'a' with the vertical. An iterative loop program was written which calculated both horizontal and vertical velocities at various time instances and the angle of heading 'a' was also iteratively varied with obtained values of velocity. For case 1, the launch angle was taken as 10° . Cases with 5° and 15° launch angles were also analysed.

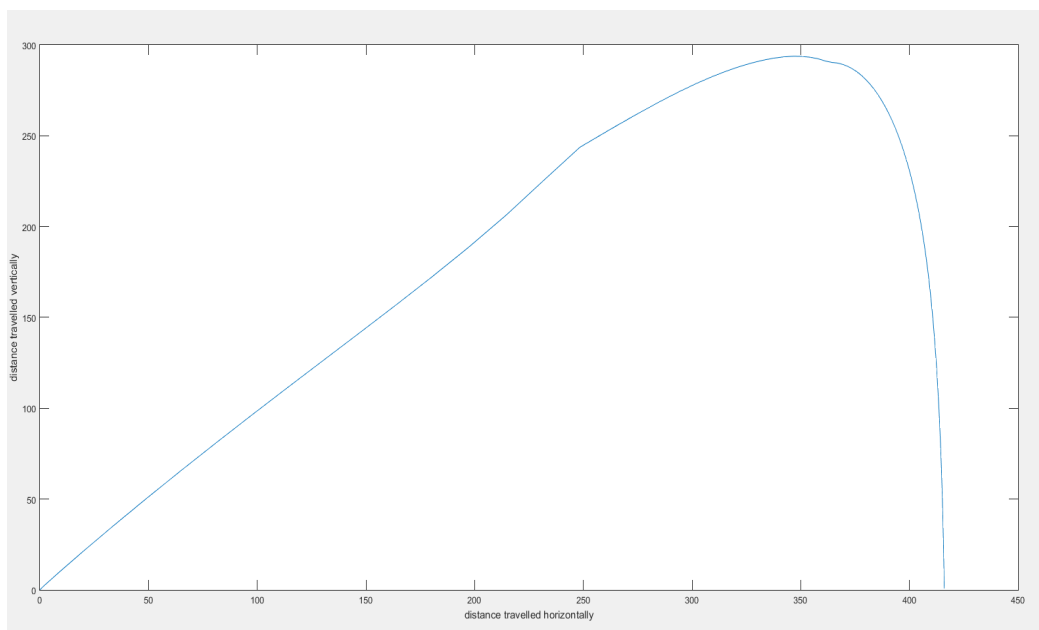


Figure 27- Trajectory of a rocket launched at 10° launch angle

Conclusion from trajectory analysis:

For inclined launch, the angle of launch came out to be an important parameter as expected. For vertical launch, altitude achieved was about 450 m, while for launch inclined at 10° , the height achieved was around 300 m. Also, the necessity for launch lugs came out clear, as no launch lug would mean the rocket would start tilting due to forces of gravity from the moment it launches which would be very fast and hence achieve very low altitudes and pose a danger.[13]

5.3 Wind tunnel

Aerodynamicists use wind tunnels to test models of proposed aircraft and engine components. During a test, the model is placed in the test section of the tunnel and air is made to flow past the model. Various types of instrumentation are used to determine the forces on the model. In some wind tunnel tests, the aerodynamic forces and moments on the model are measured directly. The model is mounted in the tunnel on a special machine called a force balance. The output from the balance is a signal that is related to the forces and moments on the model. Balances can be used to measure both the lift and drag forces. The balance must be calibrated against a known value of the force before, and sometimes during, the test.

5.3.1 Method

Many experiments for measuring drag force in wind tunnels have been carried out to grasp the various aerodynamic effects in real environment. Such as, Maruyama used a floating element in the water bath which is not connected to the floor of the wind tunnel. Mochizuki installed a floating element to a mechanical device. While Gillies measured drag force on individual obstacles on a load cell. Cheng conducted the experiment using an oil bath with a raft. Mats Sandberg and Hans Wigö measured drag force in the wind tunnels with a standard load cell. [15]

We have used the load cell method for the drag measurement due to the following advantages:

- The load cell measured the force in only one direction which the air flow direction in the wind tunnel. So where the force attacked the blocks can be ignored, because the load cell was not sensitive to any torque.
- Calibration was easy to be conducted.

5.3.2 Wind tunnel Specifications

An all steel suction type wind tunnel is housed in the Department of Aerospace Engineering at the Indian Institute of Technology-Bombay (IIT-B) and has many unique features. In the recent times, beside academic research, because of increasing awareness about significance of wind loads on different kinds of structures including high rise buildings, a variety of tests have been carried out in this wind tunnel. A schematic diagram of the wind tunnel is given in the figure below.

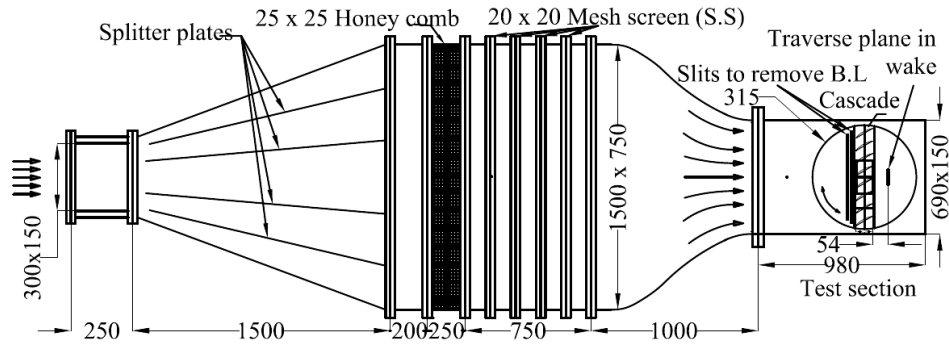


Figure 28-Wind Tunnel Schematic Diagram

Test Conditions

- Flow Velocity = 40 m/s
- Mach Number = 0.1
- Blade Chord Re number = 1.3×10^5
- Test Section Mass Flow = 2.4 kg/s
- Angles of Incidence = -17° to $+13^\circ$
- Atmospheric Temperature = 300 K
- Compressor and turbine blades
- Moving end wall

Load Cell Selection

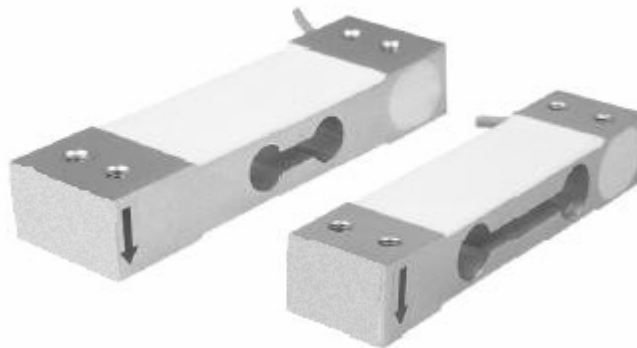


Figure 29- Basic Beam Load Cell

A load cell is a transducer that is used to create an electrical signal whose magnitude is directly proportional to the force being measured. Strain gauge load cells are the most common in industry. Strain gauge load cells work on the principle that the strain gauge (a planar resistor) deforms when the material of the load cells deforms appropriately. Deformation of the strain gauge changes its electrical resistance, by an amount that is proportional to the strain. The change in resistance of the strain gauge provides an electrical value change that is calibrated to the load placed on the load cell.

We have used a double shear type cantilever beam load cell for our application. The least count and sensitivity requirements were solely based on our CFD simulation of the drag force in ANSYS-Fluent.

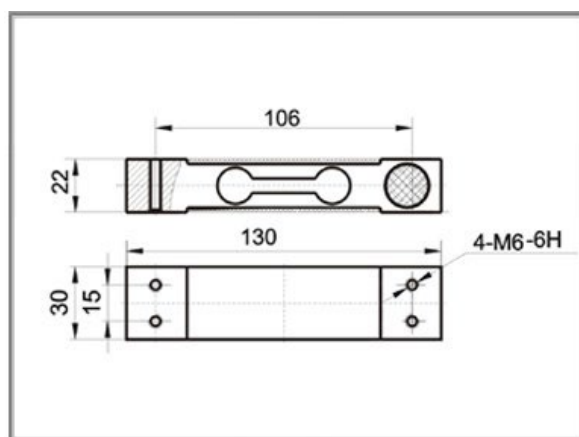


Figure 30- Load Cell Dimensions

Technical Data

Sensitivity	2.0 \pm 10% mV /V
Non-Linearity	\pm 0.02% FS
Hysteresis	\pm 0.02% FS
Repeatability	\pm 0.02% FS
Creep (20 min)	\pm 0.02% FS
Zero Balance	\pm 1%FS
Temp. effect span	\pm 0.015% FS/10°C
Temp effect zero	\pm 0.02% FS/10°C
Temperature compensated range	-10 \sim +50°C
Input resistance	410 \pm 10 Ω
Output resistance	350 \pm 3 Ω
Insulation resistance	> 5000 M Ω (@ 50V DC)
Safe Over Load	150% FS
Ultimate load	200% FS
Recommended excitation	10V (DC/AC)
Maximum excitation	12V (DC/AC)
Construction	Aluminium Alloy
Environmental protection	IP 65
Max. platform size	3 - 20 kg 300 X 300 mm 30 - 200 kg 400 X 400 mm

Figure 31- Load Cell Technical Specifications

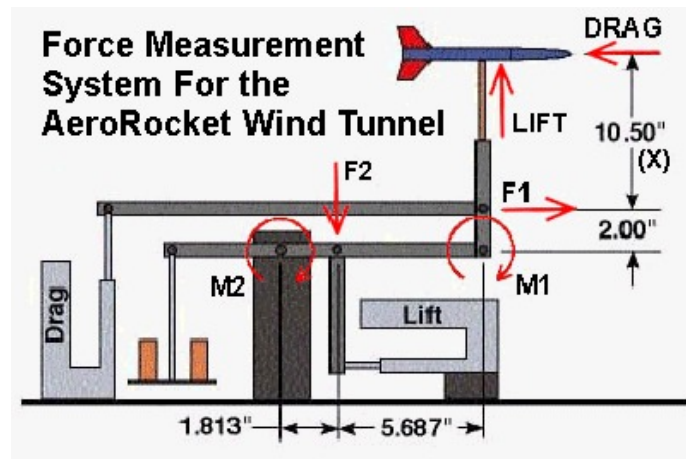


Figure 32- Force Balance Mechanism Schematic Diagram

HX711

We are interfacing a 20Kg load cell to the Arduino using the HX711 load cell amplifier module. HX711 is a precision 24-bit analog to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor. The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA) which greatly amplifies the low output of the load cell. The trend in weigh scales towards higher accuracy and lower cost has produced an increased demand for high-performance analog signal processing at low cost.

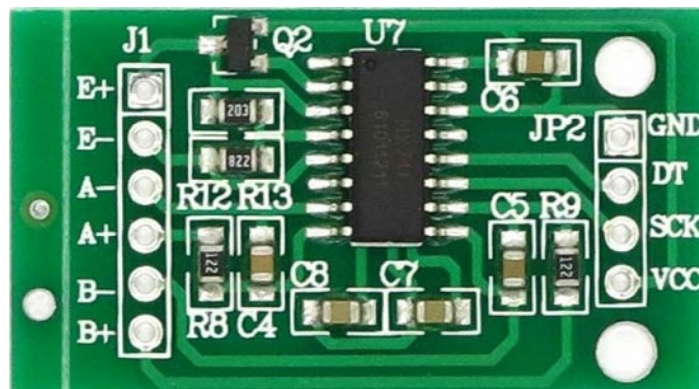


Figure 33- HX 711

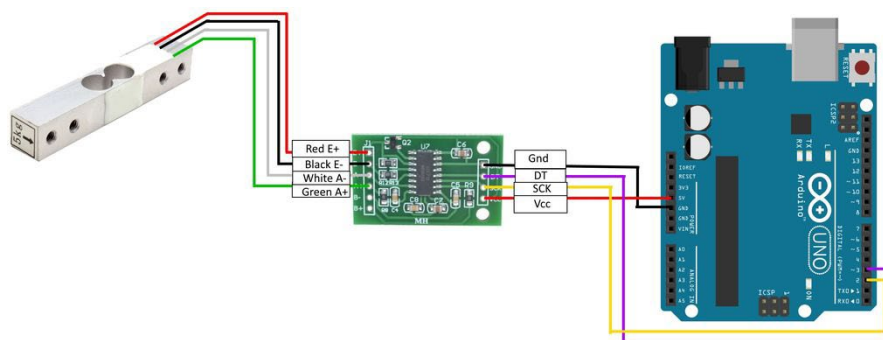


Figure 34- Load Cell Circuitry

Chapter 6 Manufacturing Process

Selective laser sintering (SLS) is an additive manufacturing technique that uses a laser as the power source to sinter powdered material (typically nylon/polyamide), aiming the laser automatically at points in space defined by a 3D model, binding the material together to create a solid structure.

An additive manufacturing layer technology, SLS involves the use of a high power laser (for example, a carbon dioxide laser) to fuse small particles of plastic, metal, ceramic, or glass powders into a mass that has a desired three-dimensional shape. The laser selectively fuses powdered material by scanning cross-sections generated from a 3-D digital description of the part (for example from a CAD file or scan data) on the surface of a powder bed. After each cross-section is scanned, the powder bed is lowered by one layer thickness, a new layer of material is applied on top, and the process is repeated until the part is completed. Because finished part density depends on peak laser power, rather than laser duration, a SLS machine typically uses a pulsed laser. The SLS machine preheats the bulk powder material in the powder bed somewhat below its melting point, to make it easier for the laser to raise the temperature of the selected regions the rest of the way to the melting point. [18]

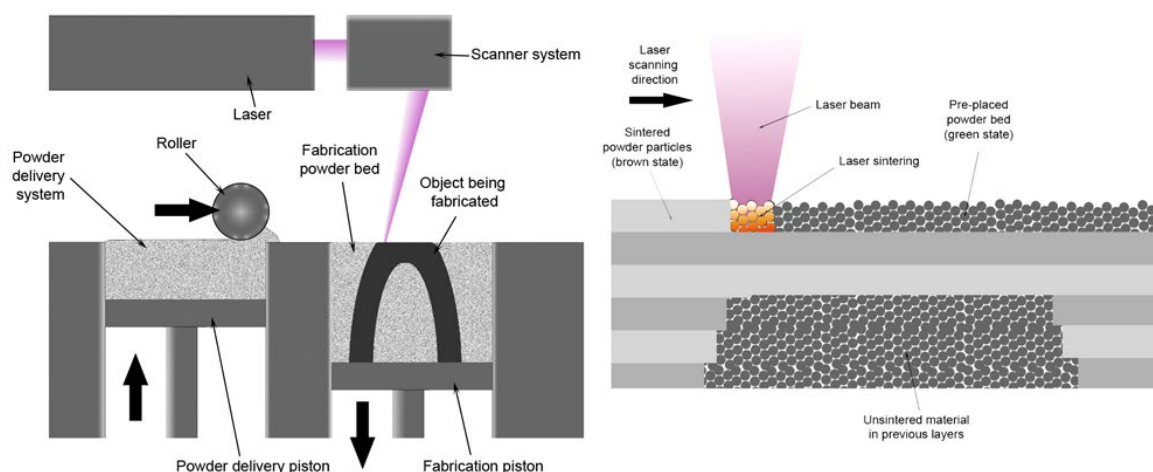


Figure 36- Selective Laser Sintering

Advantages:

- A distinct advantage of the SLS process is that because it is fully self-supporting, it allows for parts to be built within other parts in a process called nesting – with highly complex geometry that simply could not be constructed any other way
- Parts possess high strength and stiffness
- Good chemical resistance
- Complex parts with interior components, channels, can be built without trapping the material inside and altering the surface from support removal.

Reasons for selecting SLS:

- Relatively high accuracy as compared to injection moulding
- As the order quantity was only 1 piece, it was very difficult to find any manufacturer where quality and cost wouldn't be compromised
- Though surface finish isn't as great as some other processes, it can be rectified by sanding or painting the product.

Chapter 7 Experimental Setup

The project aim was to design, analyze and simulate low-altitude sounding rocket. trajectory analysis forms an important part of it. Accurate trajectory simulation requires accurate value of drag coefficient, for which CFD simulation was undertaken. However, to verify results from CFD simulation, wind-tunnel testing was proposed. A model was fabricated. The Von-Karman O-give nose and the fins were fabricated using SLS (Selective Laser Sintering) and a piece of PVC pipe formed the rocket body.



Figure 37- Fabricated Model rocket

A force balance mechanism for testing drag force was fabricated and used to carry out the wind-tunnel experiment at IIT-Bombay. A load cell with an amplifier and an Arduino micro-controller was used to measure the force.

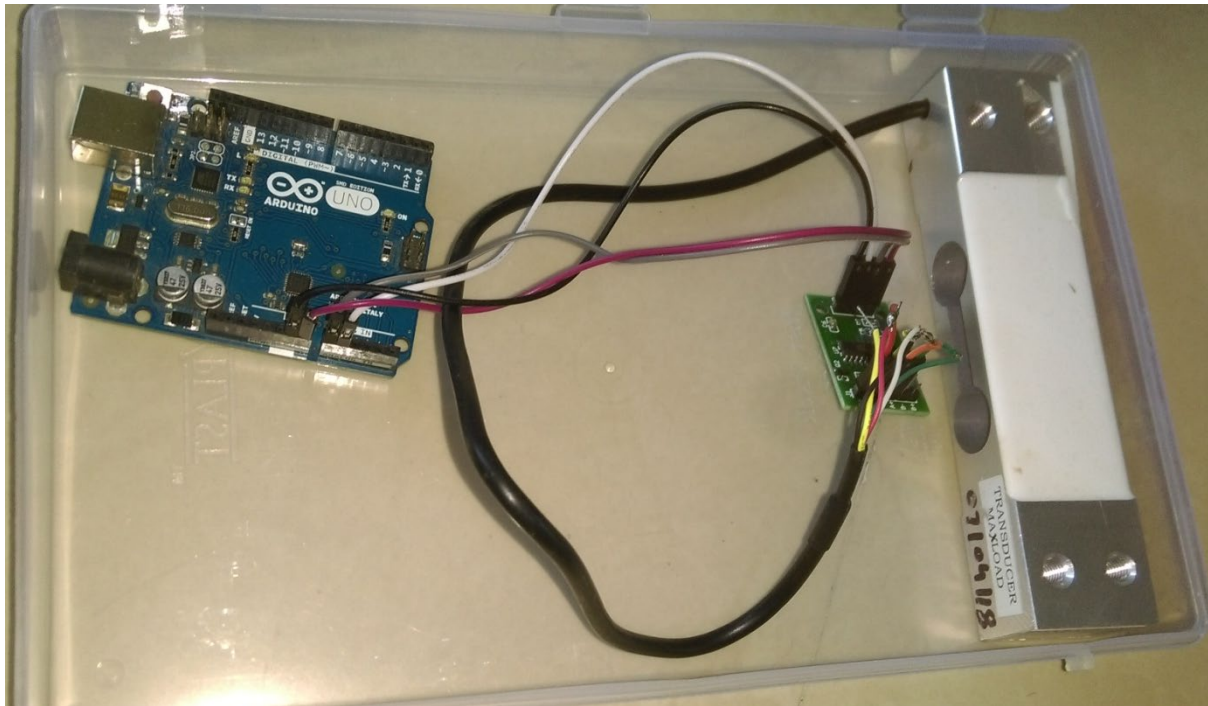


Figure 38- Load Cell Circuitry



Figure 39- Photograph of the test setup

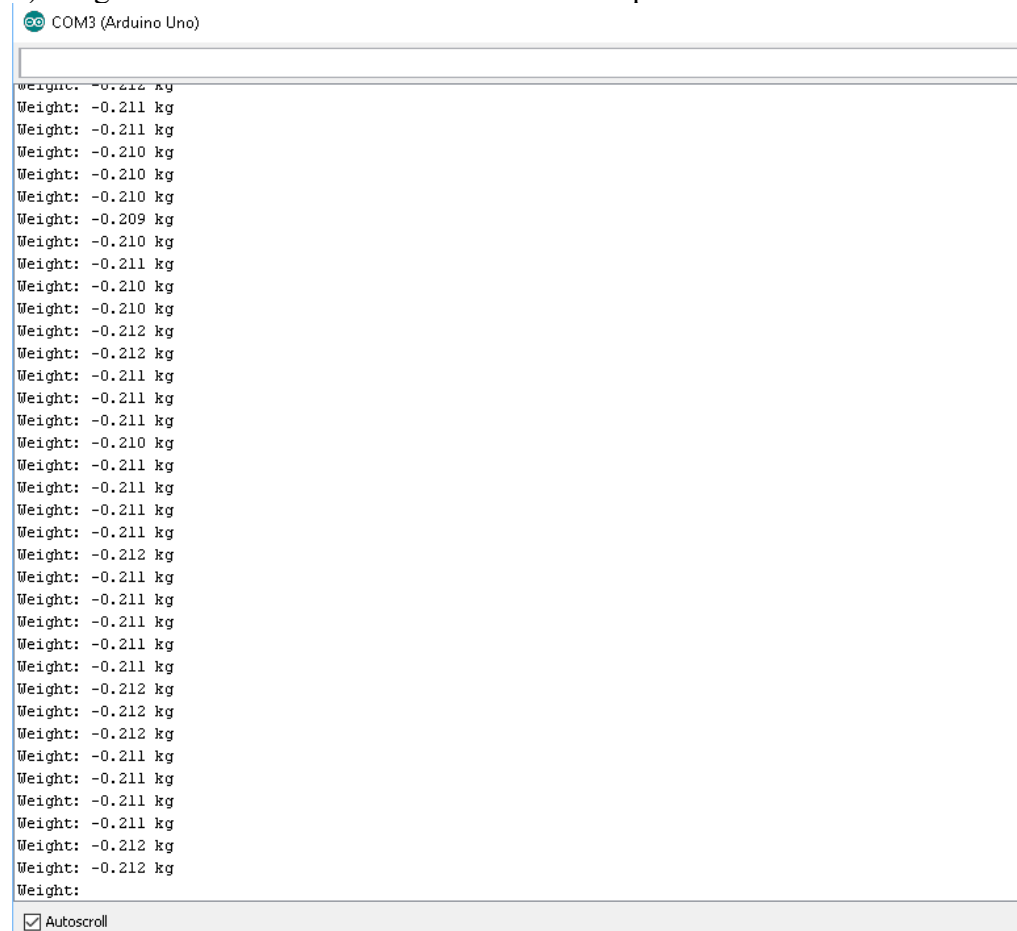
Chapter 8 Results and Discussion

Wind Tunnel: Low speed Open Jet wind tunnel at IIT, Bombay.

Wind Speed: 15 m/s

Given below are the screenshots of the Arduino Uno Serial Monitor for the results:

1) Drag force for the model and mechanism setup



The screenshot shows the Arduino Uno Serial Monitor interface. At the top, it says "COM3 (Arduino Uno)". Below this, a list of weight measurements is displayed, each preceded by "Weight:". The values range from -0.212 kg to -0.209 kg, with most values being -0.211 kg or -0.210 kg. At the bottom of the monitor, there is a checkbox labeled "Autoscroll" which is checked.

```
Weight: -0.212 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.210 kg
Weight: -0.210 kg
Weight: -0.210 kg
Weight: -0.209 kg
Weight: -0.210 kg
Weight: -0.211 kg
Weight: -0.210 kg
Weight: -0.210 kg
Weight: -0.212 kg
Weight: -0.212 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.210 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.212 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.212 kg
Weight: -0.212 kg
Weight: -0.212 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.211 kg
Weight: -0.212 kg
Weight: -0.212 kg
Weight:
Autoscroll
```

Average Initial Value : 0.211kgf = **2.06991 N**

COM3 (Arduino Uno)

```
ight: -1.337 kg
ight: -1.354 kg
ight: -1.378 kg
ight: -1.346 kg
ight: -1.303 kg
ight: -1.351 kg
ight: -1.394 kg
ight: -1.382 kg
ight: -1.346 kg
ight: -1.339 kg
ight: -1.368 kg
ight: -1.364 kg
ight: -1.356 kg
ight: -1.353 kg
ight: -1.352 kg
ight: -1.366 kg
ight: -1.387 kg
ight: -1.379 kg
ight: -1.339 kg
ight: -1.343 kg
ight: -1.380 kg
ight: -1.358 kg
ight: -1.342 kg
ight: -1.351 kg
ight: -1.341 kg
ight: -1.353 kg
ight: -1.368 kg
ight: -1.354 kg
ight: -1.333 kg
ight: -1.354 kg
ight: -1.378 kg
ight: -1.353 kg
ight: -1.346 kg
ight: -1.362 kg
ight: -1.404 kg
ight: -1.392 kg
ight:
```

Autoscroll

Average Final Value : 1.359 kg = **13.3317 N**

Drag Force experienced = Final Value - Initial Value = 13.3317 - 2.06991 = **11.2618 N**

2) Drag force for the mechanism :

COM3 (Arduino Uno)

```
Weight: 0.003 kg
Weight: 0.002 kg
Weight: 0.003 kg
Weight: 0.004 kg
Weight: 0.004 kg
Weight: 0.004 kg
Weight: 0.003 kg
Weight: 0.003 kg
Weight: 0.003 kg
Weight: 0.002 kg
Weight: 0.002 kg
Weight: 0.002 kg
Weight: 0.001 kg
Weight: 0.001 kg
Weight: 0.002 kg
Weight: 0.001 kg
Weight: 0.002 kg
Weight: 0.001 kg
Weight: 0.002 kg
Weight: 0.002 kg
Weight: 0.002 kg
Weight: 0.001 kg
Weight: 0.001 kg
Weight: 0.001 kg
Weight: 0.001 kg
Weight: 0.002 kg
Weight: 0.002 kg
Weight: 0.001 kg
Weight: 0.002 kg
Weight: 0.002 kg
Weight: 0.002 kg
Weight: 0.001 kg
Weight: 0.001 kg
Weight: 0.001 kg
Weight: 0.001 kg
Weight: 0.002 kg
Weight: 0.002 kg
Weight: 0.001 kg
Weight:
```

☒ Autoscroll

Average Initial Value : 0.0025 kg = **0.0245 N**

```
COM3 (Arduino Uno)
Weight: -0.019 kg
Weight: -0.804 kg
Weight: -0.780 kg
Weight: -0.794 kg
Weight: -0.793 kg
Weight: -0.793 kg
Weight: -0.789 kg
Weight: -0.790 kg
Weight: -0.783 kg
Weight: -0.764 kg
Weight: -0.775 kg
Weight: -0.760 kg
Weight: -0.776 kg
Weight: -0.797 kg
Weight: -0.806 kg
Weight: -0.799 kg
Weight: -0.779 kg
Weight: -0.795 kg
Weight: -0.775 kg
Weight: -0.808 kg
Weight: -0.805 kg
Weight: -0.807 kg
Weight: -0.793 kg
Weight: -0.790 kg
Weight: -0.805 kg
Weight: -0.781 kg
Weight: -0.794 kg
Weight: -0.806 kg
Weight: -0.797 kg
Weight: -0.791 kg
Weight: -0.790 kg
Weight: -0.796 kg
Weight: -0.795 kg
Weight: -0.794 kg
Weight: -0.780 kg
Weight: -0.788 kg
Weight: -0.796 kg
```

☒ Autoscroll

Average Final Value : 0.7907 kg = **7.7541 N**

Drag Force experienced by mechanism = Final Value – Initial Value = 7.7541 – 0.0245 = **7.7296N**

Therefore Drag force experienced by the model alone = Drag force (Model+Mechanism setup) – Drag force (Mechanism) = 11.2618 – 7.7296 = **3.5322 N**

Interpolating the value for a velocity of 100 m/s,
 Drag Force (100m/s) = $3.5322 * (100^2)/(15^2)$ = **156.987 N**
 So the corresponding Drag Coefficient = **20.4**
 Compensating for the load cell moment arm length,

Drag Coefficient= 20.4/4 = 5.1

The coefficient of drag obtained from the density based solver in Ansys FLUENT simulation software for a velocity of 100 m/s was **1.4**

The difference in the values obtained from the two methods might be attributed to one of the following reasons:

- 1) Limitations in the manufacturing process and the level of surface finish associated with it.
- 2) Poor quality of mesh obtained around critical areas of curvature in the CFD software.
- 3) Residual forces measured by the load cell due to initial tension and compression forces in the force balance setup.

The pressure and velocity contours were obtained using the CFD post processor and are shown below:

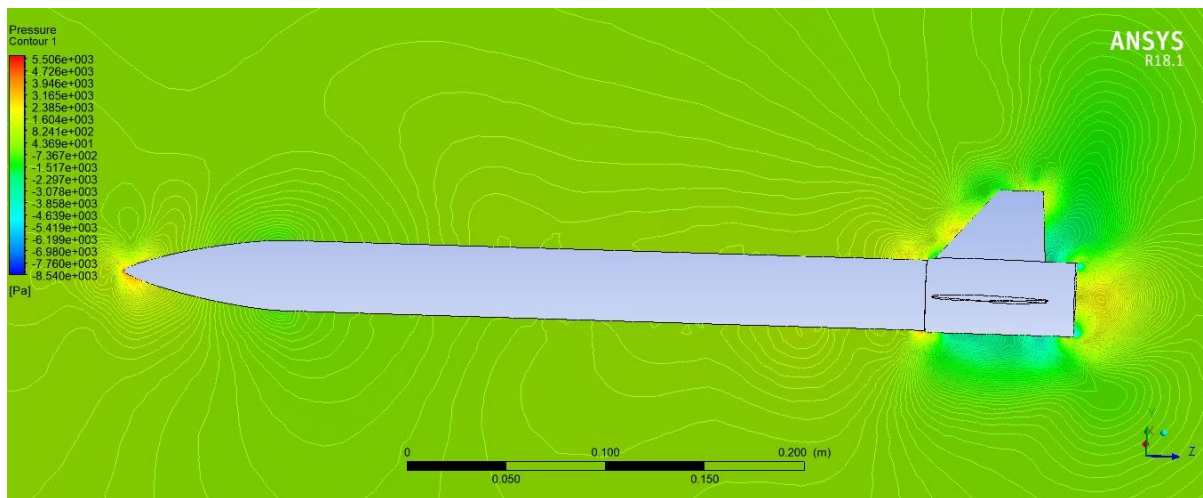


Figure 40- Pressure Contour

As shown in the above figure, maximum pressure is obtained at the tip of the nose (Shown in orange). The minimum pressure is obtained in the wake region of the wing aerofoil.

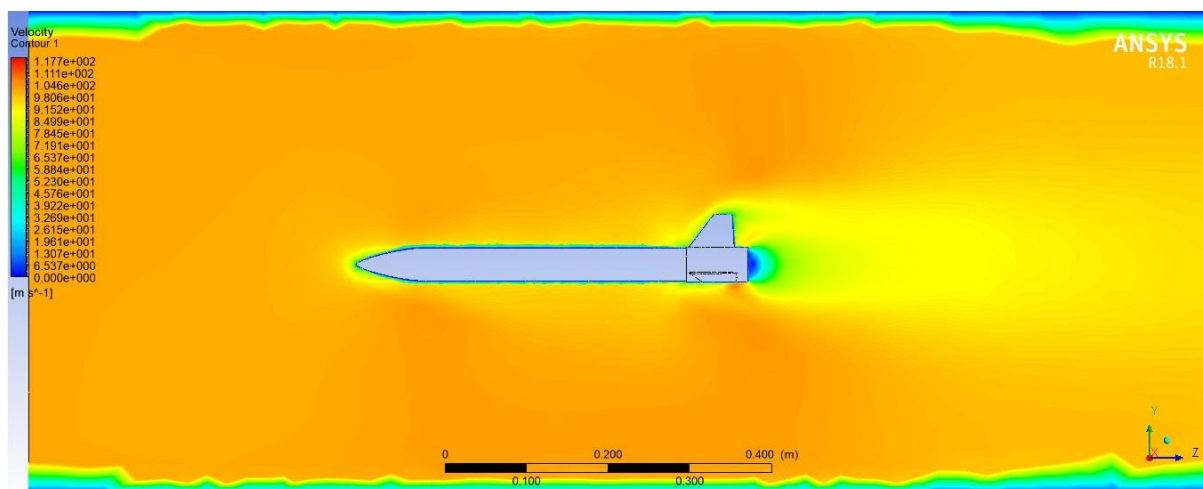


Figure 41- Velocity Contour

Zero velocity (Indicated by blue) is obtained in the region of air column in contact with the rocket body as well as near the walls of the bounding region. Velocity goes on increasing across the boundary layer and maximum velocity obtained is equal to the free stream velocity (Indicated in orange).

Chapter 9 Conclusion

The drag coefficient was calculated using two different methods and the results were compared. The reasons for the difference in values were also discussed. The open jet wind tunnel was used for the experimental testing and a force balance was designed to measure the drag force experienced by the rocket. Based on the results and the thrust force of the rocket motor selected, the trajectory of the flight of the model rocket was mathematically simulated on software.

Both CFD and wind tunnel testing are, after all, only tools whose optimality of usage is always as good as the experience and the relevant design database of the designer.

The laudable growth over the past three decades in the capability of CFD has convincingly brought home its distinctive role as an adjuvant to the design and development efforts in aerospace. It is not to be expected, however, that these traditional methods will be put to disuse because of the mature developments in computational fluid dynamics; it is indeed important to realize that these traditional methods will continue to have their roles to play but now with a synergy from CFD, to arrive at an optimum design sooner and cheaper than in the earlier times.

Chapter 10 Future Scope

- Through this project we have highlighted the basic design methodology and the simulation procedure which any designer of a sounding rocket has to go through.
- This knowledge base shall directly encourage further research projects relating to data collection and other experimentation using sounding rockets.
- Further improvements in the computational external aerodynamics analysis can be incorporated by further mesh refinement.

References:

- [1] Mark Uitendaal, *The Stratos Rocket*, Delft Aerospace Rocket Engineering (DARE), TU Delft, 2009.
- [2] MIT Rocket Team, *Critical Design Review*, January 2012
- [3] Maryel Gonzalez et al., *Intercollegiate Rocket Engineering Competition*, Florida International University, 2015
- [4] Arash Ramezani et al., *Comparison of Methods for simulation-based early prediction of rocket, artillery and mortar trajectories*, Institute of Automation Technology, Helmut Schmidt University
- [5] M. Khalil et al., *Trajectory Prediction for a typical fin stabilized artillery rocket*, 13th International Conference on Aerospace Sciences and Aviation Technology, 2009
- [6] John S. DeMar, *Model Rocket Drag Analysis using a Computerized Wind Tunnel*, National Association of Rocketry Research & Development Report, NAR 52094
- [7] Travis S. Taylor, *Introduction to Rocket Science and Engineering*, CRC Press
- [8] G. Harry Stine, *Handbook Of Model Rocketry*, 7th edition, National association of Rocketry
- [9] Ann Grimm, *Estes Model Rocketry Guide*, 2009
- [10] Nikolaos S. Christodoulou, *An Algorithm using Runge-Kutta methods of orders 4 and 5 for Systems of Odes*
- [11] Dennis J. Martin, *Summary of Flutter Experiences as a guide to the preliminary design of the lifting surfaces of missiles*, NACA Technical Note 4197, 1958
- [12] P. Spalart, S. R. Allmaras, *A One-Equation Turbulence Model for Aerodynamic Flows*, 1992
- [13] N.J. Lawson et al, *Jetstream 31 national flying laboratory: Lift and drag measurement and modelling*, 2017
- [14] Brian Cooper, *Analysis and Design of an Affordable High Altitude Rocket System*, 2011
- [15] Li Nan, *The methods of drag force measurement in wind tunnels*, 2013
- [16] “Why do spinning rockets fly higher”, Apogee Rockets
- [17] www.nasa.gov
- [18] en.wikipedia.org

Appendix

Appendix I- Trajectory prediction at different launch angles

Vertical launch(0°)

Case i: Vertical launch, velocity equation for initial 0-0.4s:

```
function eqn1
h=0.01;
x=0;
w=0;
fprintf('Step 0: x=%12.8f, w=%12.8f\n',x,w);
for i=1:40
    k1=h*f(x,w);
    k2=h*f(x+h/2,w+k1/2);
    k3=h*f(x+h/2,w+k2/2);
    k4=h*f(x+h,w+k3);
    w=w+(k1+2*k2+2*k3+k4)/6;
    x=x+h;
    fprintf('Step %d: x=%f, w=%18.15f\n',i,x,w);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function v=f(x,y)
v=(59.57*x-0.00105976*(y^2)-(1.48131-0.09266*(x^2)))/((0.151-0.00945*x^2));
```

Case ii: Vertical launch, velocity equation for 0.4-3.4s

```
function eqn1
h=0.01;
x=0.4;
w=27.8366;
fprintf('Step 0: x=%12.8f, w=%12.8f\n',x,w);
for i=1:300
    k1=h*f(x,w);
    k2=h*f(x+h/2,w+k1/2);
    k3=h*f(x+h/2,w+k2/2);
    k4=h*f(x+h,w+k3);
    w=w+(k1+2*k2+2*k3+k4)/6;
    x=x+h;
    fprintf('Step %d: x=%f, w=%18.15f\n',i,x,w);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function v=f(x,y)
v=(116.16*exp(-5.997*x)+16.48*exp(-0.07028*x)-0.00105976*(y^2)-(1.42245-(19.37*exp(-5.997*x)+234.49*exp(-0.07028*x))/321.428))/(0.145-(19.37*exp(-5.997*x)+234.49*exp(-0.07028*x))/3150);
```


Case iii: Vertical launch, velocity equation for 3.4s upto top.

```
function eqn1
h=0.05;
x=3.4;
w=108.531418776688890;
fprintf('Step 0: x=%12.8f, w=%12.8f\n',x,w);
for i=1:77

    k1=h*f(x,w);
    k2=h*f(x+h/2,w+k1/2);
    k3=h*f(x+h/2,w+k2/2);
    k4=h*f(x+h,w+k3);
    w=w+(k1+2*k2+2*k3+k4)/6;
    x=x+h;
    fprintf('Step %d: x=%f, w=%18.15f\n',i,x,w);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function v=f(x,y)
v=(-0.00105976*(y^2)-(0.091*9.8))/(0.091);
```

Inclined launch, at 10 degrees:

Case i) From 0-0.4s:

```
function eqn1
a=10;
h=0.01;
x=0.1;
w=0;
z=0;
fprintf('Step 0: x=%12.8f, w=%12.8f, z=%12.8f\n',x,w,z);
for i=1:30
    k1=h*f(x,w,a);
    k2=h*f(x+h/2,w+k1/2,a);
    k3=h*f(x+h/2,w+k2/2,a);
    k4=h*f(x+h,w+k3,a);
    w=w+(k1+2*k2+2*k3+k4)/6;
    k5=h*g(x,z,a);
    k6=h*g(x+h/2,z+k1/2,a);
    k7=h*g(x+h/2,z+k2/2,a);
    k8=h*g(x+h,z+k3,a);
    z=z+(k5+2*k6+2*k7+k8)/6;
    a=abs(atan(w/z));
    x=x+h;
    fprintf('Step %d: x=%f, w=%18.15f, z=%18.15f, a=%18.15f\n',i,x,w,z,a);
```

```

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function v=f(x,y,a)
v=((59.57*x-0.00105976*(y^2)/(sind(a)*sind(a)))*sind(a))/((0.151-0.00945*x^2));
function m=g(x,y,a)
m=((59.57*x-0.00105976*(y^2)/(cosd(a)*cosd(a)))*cosd(a)-(1.48131-
0.09266*(x^2)))/((0.151-0.00945*x^2));

```

Case ii) From 0.4-3.4s:

```

function eqn1
a=23.271441946390670;
h=0.01;
x=0.4;
w=12.683325254305018;
z=29.490804602001951;

fprintf('Step 0: x=%12.8f, w=%12.8f, z=%12.8f\n',x,w,z);
for i=1:300
    k1=h*f(x,w,a);
    k2=h*f(x+h/2,w+k1/2,a);
    k3=h*f(x+h/2,w+k2/2,a);
    k4=h*f(x+h,w+k3,a);
    w=w+(k1+2*k2+2*k3+k4)/6;
    k5=h*g(x,z,a);
    k6=h*g(x+h/2,z+k1/2,a);
    k7=h*g(x+h/2,z+k2/2,a);
    k8=h*g(x+h,z+k3,a);
    z=z+(k5+2*k6+2*k7+k8)/6;
    a=atand(w/z);
    x=x+h;
    fprintf('Step %d: x=%f, w=%18.15f, z=%18.15f, a=%18.15f\n',i,x,w,z,a)
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

function v=f(x,y,a)
v=(116.16*exp(-5.997*x)+16.48*exp(-0.07028*x)-
0.00105976*(y^2)/(sind(a)*sind(a)))*sind(a)/(0.145-(19.37*exp(-5.997*x)+234.49*exp(-
0.07028*x))/3150);
function m=g(x,y,a)
m=((116.16*exp(-5.997*x)+16.48*exp(-0.07028*x)-
0.00105976*(y^2)/(cosd(a)*cosd(a)))*cosd(a)-(1.42245-(19.37*exp(-5.997*x)+234.49*exp(-
0.07028*x))/321.428))/(0.145-(19.37*exp(-5.997*x)+234.49*exp(-0.07028*x))/3150);

```

Case iii: From 3.4-6.78 s:

```

function eqn1
h=0.02;
x=3.4;
w=55.055155441275836;
z=94.026674286954346;

```

```

a=30.350124600451394;
fprintf('Step 0: x=%12.8f, w=%12.8f, z=%12.8f\n',x,w,z);
for i=1:169
    k1=h*f(x,w,a);
    k2=h*f(x+h/2,w+k1/2,a);
    k3=h*f(x+h/2,w+k2/2,a);
    k4=h*f(x+h,w+k3,a);
    w=w+(k1+2*k2+2*k3+k4)/6;
    k5=h*g(x,z,a);
    k6=h*g(x+h/2,z+k1/2,a);
    k7=h*g(x+h/2,z+k2/2,a);
    k8=h*g(x+h,z+k3,a);
    z=z+(k5+2*k6+2*k7+k8)/6;
    a=abs(atan(w/z));
    x=x+h;
    fprintf('Step %d: x=%f, w=%18.15f, z=%18.15f, a=%18.15f\n',i,x,w,z,a)
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function v=f(~,y,a)
v=(-0.00105976*(y^2)/sind(a))/(0.091);
function m=g(~,y,a)
m=(-0.00105976*(y^2)/cosd(a)-(0.091*9.8))/(0.091);

```

Case iv) From 6.78-19.5 s:

```

function eqn1
h=0.02;
x=6.78;
w=13.095857169568916;
z=0.095447479362257;
a=89.582414436022731;
fprintf('Step 0: x=%12.8f, w=%12.8f, z=%12.8f\n',x,w,z);
for i=1:636
    k1=h*f(x,w,a);
    k2=h*f(x+h/2,w+k1/2,a);
    k3=h*f(x+h/2,w+k2/2,a);
    k4=h*f(x+h,w+k3,a);
    w=w+(k1+2*k2+2*k3+k4)/6;
    k5=h*g(x,z,a);
    k6=h*g(x+h/2,z+k1/2,a);
    k7=h*g(x+h/2,z+k2/2,a);
    k8=h*g(x+h,z+k3,a);
    z=z+(k5+2*k6+2*k7+k8)/6;
    a=abs(atan(w/z));
    x=x+h;
    fprintf('Step %d: x=%f, w=%18.15f, z=%18.15f, a=%18.15f\n',i,x,w,z,a)
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function v=f(~,y,a)
v=(-0.00105976*(y^2)/sind(a))/(0.091);

```

```
function m=g(~,y,a)
m=(0.00105976*(y^2)/cosd(a)-(0.091*9.8))/(0.091);
```

Sample program for curve-fitting:

```
A=[6.78 6.8 6.82 6.84 6.86 6.88 6.9];
B=[0.09544748 -0.100366057 -0.295924497 -0.490902349 -0.685295873 -0.879108075 -
1.072342331];
C=[637, sum(A), sum(A.^2),sum(A.^3),sum(A.^4),sum(A.^5);sum(A),
sum(A.^2),sum(A.^3),sum(A.^4),sum(A.^5),sum(A.^6);sum(A.^2),sum(A.^3),sum(A.^4),su
m(A.^5),sum(A.^6),sum(A.^7);sum(A.^3),sum(A.^4),sum(A.^5),sum(A.^6),sum(A.^7),sum(
A.^8);sum(A.^4),sum(A.^5),sum(A.^6),sum(A.^7),sum(A.^8),sum(A.^9);sum(A.^5),sum(A.
.^6),sum(A.^7),sum(A.^8),sum(A.^9),sum(A.^10)];
D=[sum(B),sum(A.*B),sum((A.^2).*B),sum((A.^3).*B),sum((A.^4).*B),sum((A.^5).*B)];
E=linsolve(C,D')
```

Program used for integrating fitted equations

```
%1 hori%
p11 = 146.1 ;
p12 = -11.75 ;
p13 = 0.118 ;

syms x
f1(x) = p11*x^2 + p12*x + p13
j(x)=int(f1(x));
ch11=feval(j,0.06);
j1(x)=j(x)-ch11;
ch12=feval(j1,0.4);

%1 vertical%
p21 = 132.3;
p22 = 0.1748;
p23 = -0.5318;

f2(x) = p21*x^2 + p22*x + p23
j2(x)= int(f2(x),x);
cv11=feval(j2,0.06);
j3(x)=j2(x)-cv11;
cv12=feval(j3,0.4);

%2 hori%
a = 82.98 ;
b = 0.02196 ;
c = -293.3 ;
d = -3.764 ;
g1(x) = a*exp(b*x) + c*exp(d*x)
j4(x)=int(g1(x),x);
ch21=feval(j4,0.4);
j5(x)=j4-ch21+ch12;
```

```

ch22=feval(j5,3.4);
%2 verti%

p41 = -0.567 ;
p42 = 10.08 ;
p43 = -77.27 ;
p44 = 334.7 ;
p45 = -898.4 ;
p46 = 1540 ;
p47 = -1661 ;
p48 = 1037 ;
p49 = -207 ;
g2(x) = p41*x^8 + p42*x^7 + p43*x^6 + p44*x^5 + p45*x^4 + p46*x^3 + p47*x^2 +
p48*x + p49
j6(x)=int(g2(x),x) ;
cv21=feval(j6,0.4);
j7(x)=j6-cv21 + cv12 ;
cv22=feval(j7,3.4);
%3 hori%
syms x
p51 = 0.03454 ;
p52 = -1.4256 ;
p53 = 24.2829 ;
p54 = -218.7817 ;
p55 = 1102.1 ;
p56 = -2957.3 ;
p57 = 3350.1 ;
h1(x) = p51*x^6 + p52*x^5 + p53*x^4 + p54*x^3 + p55*x^2 + p56*x + p57
j8(x)=int(h1(x),x);
ch31=feval(j8,3.4);
j9(x)=j8(x)-ch31+ch22;
ch32=feval(j9,6.78);

%3 verti%

p61 = 0.00220943362565710 ;
p62 = -0.219288040768267 ;
p63 = 5.67941116446358 ;
p64 = -67.2646849851963 ;
p65 = 414.276878586597 ;
p66 = -1306.75708247465 ;
p67 = 1697.36700966451 ;
h2(x) = p61*x^6 + p62*x^5 + p63*x^4 + p64*x^3 + p65*x^2 + p66*x + p67

j10(x)=int(h2(x),x);
cv31=feval(j10,3.4);
j11(x)=j10(x)-cv31+cv22;
cv32=feval(j11,6.78);
%eqn 4 horizontal
i1(x)=-29.9702559131054+22.3916448878920*x-

```

Final equation plotting program for trajectory

53

```

(1183876702481435*x.^2)/70368744177664 + (5314697456518041*x)/35184372088832 -
8122977591605164763564069894237827/60528378991859466240000000000000000).*(6.78<=
x&x<=19.1)];
plot(f(x),g(x))
xlabel('distance travelled horizontally');
ylabel('distance travelled vertically');

```

Appendix II- Load Cell Calibration Code for Arduino

```
/*  
  
* https://circuits4you.com  
  
* 2016 November 25  
  
* Load Cell HX711 Module Interface with Arduino to measure weight in Kgs  
  
Arduino  
  
pin  
  
2 -> HX711 CLK  
  
3 -> DOUT  
  
5V -> VCC  
  
GND -> GND
```

Most any pin on the Arduino Uno will be compatible with DOUT/CLK.

The HX711 board can be powered from 2.7V to 5V so the Arduino 5V power should be fine.

```
*/
```

```
#include "HX711.h" //You must have this library in your arduino library folder
```

```
#define DOUT 3
```

```
#define CLK 2
```

```
HX711 scale(DOUT, CLK);
```

```
//Change this calibration factor as per your load cell once it is found you may need  
to vary it in thousands
```



```

float calibration_factor = -96650; //-106600 worked for my 40Kg max scale setup

//=====

//          SETUP

//=====

void setup() {

  Serial.begin(9600);

  Serial.println("HX711 Calibration");

  Serial.println("Remove all weight from scale");

  Serial.println("After readings begin, place known weight on scale");

  Serial.println("Press a,s,d,f to increase calibration factor by 10,100,1000,10000 respectively");

  Serial.println("Press z,x,c,v to decrease calibration factor by 10,100,1000,10000 respectively");

  Serial.println("Press t for tare");

  scale.set_scale();

  scale.tare(); //Reset the scale to 0

  long zero_factor = scale.read_average(); //Get a baseline reading

  Serial.print("Zero factor: "); //This can be used to remove the need to tare the scale. Useful in permanent scale projects.

  Serial.println(zero_factor);

}

//=====

```

```

//                                LOOP

//=====

void loop() {

    scale.set_scale(calibration_factor); //Adjust to this calibration factor


    Serial.print("Reading: ");

    Serial.print(scale.get_units(), 3);

    Serial.print(" kg"); //Change this to kg and re-adjust the calibration factor if you
follow SI units like a sane person

    Serial.print(" calibration_factor: ");

    Serial.print(calibration_factor);

    Serial.println();


    if(Serial.available())
    {
        char temp = Serial.read();

        if(temp == '+' || temp == 'a')
            calibration_factor += 10;

        else if(temp == '-' || temp == 'z')
            calibration_factor -= 10;

        else if(temp == 's')
            calibration_factor += 100;

        else if(temp == 'x')
            calibration_factor -= 100;
    }
}

```

```

else if(temp == 'd')

    calibration_factor += 1000;

else if(temp == 'c')

    calibration_factor -= 1000;

else if(temp == 'f')

    calibration_factor += 10000;

else if(temp == 'v')

    calibration_factor -= 10000;

else if(temp == 't')

    scale.tare(); //Reset the scale to zero

}

}

//=====
=====

```

Weight Measurement Code

```

/*
 * https://circuits4you.com
 * 2016 November 25
 * Load Cell HX711 Module Interface with Arduino to measure weight in Kgs
 Arduino
 pin
 2 -> HX711 CLK
 3 -> DOUT
 5V -> VCC
 GND -> GND

```

Most any pin on the Arduino Uno will be compatible with DOUT/CLK.
The HX711 board can be powered from 2.7V to 5V so the Arduino 5V power should be fine.

```
*/
```

```
#include "HX711.h" //You must have this library in your arduino library folder
```

```
#define DOUT 3
```

```
#define CLK 2
```

```
HX711 scale(DOUT, CLK);
```

```
//Change this calibration factor as per your load cell once it is found you may need  
to vary it in thousands
```

```
float calibration_factor = -96650; //-106600 worked for my 40Kg max scale setup
```

```
//=====
```

```
=====
```

```
//          SETUP
```

```
//=====
```

```
=====
```

```
void setup() {  
  Serial.begin(9600);  
  Serial.println("Press T to tare");  
  scale.set_scale(-96650); //Calibration Factor obtained from first sketch  
  scale.tare();           //Reset the scale to 0  
}
```

```
//=====
```

```
=====
```

```
//          LOOP
```

```
//=====
```

```
=====
```

```
void loop() {  
  Serial.print("Weight: ");  
  Serial.print(scale.get_units(), 3); //Up to 3 decimal points  
  Serial.println(" kg"); //Change this to kg and re-adjust the calibration factor if you  
follow lbs
```

```
  if(Serial.available())  
  {  
    char temp = Serial.read();  
    if(temp == 't' || temp == 'T')  
      scale.tare(); //Reset the scale to zero  
  }  
}
```

```
//=====
```

```
=====
```