

POLITECNICO DI TORINO

Bachelor's Degree in Aerospace Engineering



**Politecnico
di Torino**



Bachelor's Degree Thesis

Efficient Computation of Center of Pressure for Supersonic Finned Vehicles via Reduced Order Modeling

Supervisors

Prof. Domenic D'AMBROSIO

Ing. Federica PORTIS

Candidate

Andrea PANTANO

July 2024

Summary

The determination of the center of pressure is fundamental in rocket engineering, because it is not just necessary to get accurate flight simulation, but it is also needed to insure the stability of the flight throughout every phase of the flight. Using *Computational Fluid Dynamics* gives accurate results, but it requires a high computational cost and can be inefficient during the preliminary design phase. The case study for this thesis is the high-powered sounding rocket ***Vittorio Emanuele II (VES)***, designed and built by Polito Rocket Team. The final goal is to develop a *Matlab* script to be implemented in the Team's simulators, and to be used in a preliminary phase of the fins design for future sounding rocket. Various simulators, such as OpenRocket and the *python* library *rocketpy* are able to conduct center of pressure and stability margin calculations, but they are not considered reliable at supersonic speed.

Table of Contents

List of Tables	VI
List of Figures	VII
Symbols	IX
1 Polito Rocket Team and VES	1
1.1 Team description	1
1.2 Project Vittorio Emanuele II	3
1.3 Mission profile and CONOPS (Concept of Operations)	4
1.4 Simulations	5
2 Center of pressure analysis	8
2.1 Traditional analysis in rocketry: the Barrowman method	9
2.2 The FIN algorithm	10
2.2.1 Busemann’s Second Order Airfoil Theory	11
2.2.2 Geometric calculations	18
2.3 Overall center of pressure computation	19
2.3.1 Lift on nose	20
2.3.2 Nosecone approximation and expansion correction	21
2.3.3 Lift on wing in presence of body	23
2.3.4 Lift on body in presence of wing	23
2.3.5 Overall center of pressure	26
3 Results and conclusions	27
3.1 Computation results	27
3.2 CFD results and comparison	31
3.3 Conclusions, possible future improvements	33
A Supersonic cp calculator	35
B Fin 2	40

C	Nosecone approximation function	44
D	Technical drawings	47
	Bibliography	50

List of Tables

1.1	Motor Specifications	4
2.1	VES Fin Geometry Parameters	14
3.1	Freestream parameters	27
3.2	Results comparison	32

List of Figures

1.1	The OpenRocket interface	5
2.1	Fin profiles that can be analyzed using FIN [3]	10
2.2	Cross section of a fin. The blue lines are the two layers of reinforcement nylon fibers	12
2.3	Fin geometry parameters [3]	13
2.4	Fin section dimensions and parameters [3]	13
2.5	Fin profile angles [3]	14
2.6	Fin Tip Mach cone schematics [3]	16
2.7	3D and equivalent planar model introduced Van Dyke and Lagerstrom [1]	24
3.1	Results for $\alpha = 0.5^\circ$	28
3.2	Results for $\alpha = 1^\circ$	28
3.3	Results for $\alpha = 2^\circ$	29
3.4	Results for $\alpha = 5^\circ$	29
3.5	Results for $\alpha = 7^\circ$	30
3.6	Results for $M = 1.59$ with variable angle of attack	31
3.7	CFD result for the Mach number	32
3.8	CFD result for the flow pressure	33

Symbols

A Reference area

C_{LN} Nose lift coefficient

C_{LBW} Body lift coefficient due to wing interference

C_{LWB} Wing lift coefficient due to body interference

$C_{L\alpha W}$ Wing-alone lift coefficient gradient

c_r Fins root chord

c_t Fins tip chord

d Rocket internal diameter

d_i Component of l_i parallel to the freestream

K_N Wing-nose interference coefficient

K_{BW} Wing on body interference coefficient

K_{WB} Body on wing interference coefficient

l_{BW} Body center of pressure position due to wing interference

l_N Nose center of pressure position

l_s Nose length up to shoulder

l_{Lr} Length of root airfoil leading edge region chord

l_{Tr} Length of root airfoil trailing edge region chord

l_w position of the leading edge intersection with root chord, relative to reference point (nose tip)

L_W Wing-alone lift

$m = \cotan(\Gamma_L)$

M Freestream Mach number

M_{BW} Moment acting on body due to wing interference

n_i Component of n_i parallel to the freestream

q_∞ Freestream dynamic pressure

r Rocket body radius

r_n Rocket radius at nose shoulder

R Dry air gas constant

s Wing-alone span

$s_t = r_n + s$, total span

T_∞ Freestream air temperature

V_S Nose volume up to shoulder

X_{CP} Overall center of pressure position

x_{CPW} Wing local center of pressure

X_L Distance between reference axis and intersection of root and leading edge

$x_{relCPBW}$ Local position of body center of pressure due to wing interference

α Angle of attack

$\beta = \sqrt{M^2 - 1}$

γ Ratio of specific heats

Γ_L Fins leading edge sweepback angle

Γ_1 Fins first region sweepback angle

Γ_2 Fins second region sweepback angle

Γ_T Fins trailing edge sweepback angle

$\lambda = \frac{c_t}{c_r}$, fins taper ratio

$\mu = \arcsin\left(\frac{1}{M}\right)$

ρ Air density

ζ_L Leading edge semi-opening angle

ζ_T Trailing edge semi-opening angle

Chapter 1

Polito Rocket Team and VES

1.1 Team description

Polito Rocket Team (PRT) is the rocket engineering team of Politecnico di Torino, Italy, officially recognized by our university on June 7th, 2022. Fueled by a resolute commitment to innovation and a forward-looking perspective, our mission is to advance Italy's aerospace industry. In an aerospace market characterized by constant evolution, our group recognizes the pivotal role that academia, young students, and researchers play in bringing fresh ideas, energy, and dedication. We strive to merge the theoretical foundations provided by Politecnico di Torino with hands-on experience within a well-structured team environment, mirroring industry practices. The primary objective of PRT is to equip our members with the knowledge and skills essential for success in the job market, positioning them advantageously and providing a robust support network. We emphasize the significance of outreach and networking, aiming to inspire enthusiasm within the general public, especially among the youth who represent the future Space Generation. Polito Rocket Team is open to any student of our university and was founded as an international team. The team's founders chose to adopt English as the official language and the team managers and leads strongly believe in the importance of the contribute of each individual with its culture and personal background. The Team tries to do so without forgetting the traditions, strong values and cultural roots of Italy and Turin, the city that represent home for our Team. PRT's inaugural project, named Cavour, represented a significant ambition for a first project in rocket engineering as a reusable and reconfigurable sounding rocket, powered by a commercial off-the-shelf (COTS) solid motor. Cavour demonstrated remarkable versatility, capable of achieving variable apogees ranging from approximately 1.4

km to 5 km, enabling adaptation to diverse mission profiles without necessitating alterations to the core design or critical components. The integration of additive manufacturing, topological optimization, and generative design techniques rendered Cavour exceptionally lightweight while maintaining structural integrity to fulfill its mission objectives. Despite its lightness, Cavour had the capability to carry payloads weighing up to 4 kg to the desired apogee, constituting approximately one-third of the rocket's total wet mass. On April 29th, 2023, in Germany, Cavour achieved a milestone by reaching an apogee of approximately 1350m, marking a successful mission with no major issues encountered during flight. This accomplishment bolstered the confidence of the Polito Rocket Team for future launches. Shortly after the successful launch, on May 2nd, our team hosted the inaugural Italian Rocket Engineering Students Summit at Politecnico di Torino. We hold a firm belief in the significance of fostering collaboration and dialogue among fellow rocketry enthusiasts from across Italy. This inaugural event marked the inception of what we envision as a longstanding tradition within the Italian rocketry community. Building upon this momentum, the second edition took place in Padova on May 7th, 2024. Moreover, this event served as a platform for our community to engage with the general public and bolster credibility within the academic realm. By facilitating discussions and knowledge-sharing sessions, we aim to strengthen the bonds among rocketry enthusiasts and propel the field forward collectively. In June 2023, the team made history by becoming the first Italian team to participate in the prestigious Spaceport America Cup, the foremost rocket engineering competition globally. The performance of the Cavour project exceeded expectations, with the rocket reaching an impressive apogee of approximately 3,150 km and being successfully recovered. Among more than 110 competing teams, the team secured the 20th position overall and the 13th position in the 10k ft COTS category. The innovative design and manufacturing approach utilizing additive technology garnered admiration from the judges, who honored the project with the esteemed "Dr. Gill Moore Award for Innovation," recognizing the team's substantial contributions to the sport and industry. Furthering its success, Cavour was relaunched in October 2023 at the European Rocketry Challenge, where it clinched another accolade, the Anacom Award for the best spectral telemetry signature, and achieved an impressive 8th position overall among the 22 participating universities. This remarkable feat of conducting three launches within a year underscores the exceptional progress of the rocketry team in its inaugural year of full-force operation, earning recognition and appreciation within the industry. In October, 2023, PRT started its second rocketry project, Vittorio Emanuele II (VES). VES is the Team's next competition rocket and will be a significant progress with respect to Cavour, because of enhanced avionics and an even more efficient structure, manufactured thanks to all the know-how in additive manufacturing acquired through the Cavour project. VES will be capable of reaching a 10km apogee and will be our first supersonic rocket,

with a top speed of 1.8 Mach. It is also an important aerodynamics and flight dynamics research opportunity for the team, because of the presence of an airbrakes system. Since January, 2023, another important project is run in parallel with the rocketry programs: project Efesto. It will be the first biliquid rocket engine designed, tested and built by the Team. It is a long-term project because of its immense complexity, and it represents an important research opportunity for our members. The design phase is nearing the end, and once build, Efesto will be the first ever biliquid rocket engine built by students in Italy, and one of the most innovative in the world using cryogenic methane and liquid oxygen as propellants. It will make extensive use of cutting edge additive manufacturing technologies and advanced aerospace grade materials. The Efesto project is the core of this paper, and the details will be explained in the scientific part. We plan on integrating the Efesto engine on a future competition rocket, probably in 2025.

1.2 Project Vittorio Emanuele II

Vittorio Emanuele II (VES) is the second competition rocket built by Polito Rocket Team. It represent an important step forward with respect to its predecessor, Cavour. Every subsystem has been enhanced. The airframe have been built out of carbon fiber, with a glass fiber section corresponding to the recovery bay to ensure RF transparency. The structural flanges are made of advanced carbon-based polymers (e.g. PEEK, PEKK, *Onyx*). The finset has been designed as a monolithic component, reinforced with nylon fibers, laid down following an optimized pattern to ensure flutter resistance. The recovery system features a redundant double-deployment mechanism, which includes a COTS (Commercial Off The Shelf) *Raptor CO₂* system and a SRAD (Students Research and Developed) system for redundancy. The avionics bay houses both COTS and SRAD flight computers. Inertial Measurement Units (IMUs) have been added to support the barometric altimeters.

VES will be the Team's first supersonic rocket, and its nosecone has been designed with a $\frac{1}{2}$ power parabolic shape to reduce wave drag. VES will also mount an active aerodynamic surface, the airbrakes. The airbrakes will be controlled by their independent avionics, and actuated by a servo-motor. During its first flight (Delft, October 2024) the airbrakes efficiency will be measured using various pressure sensors mounted inside one of the airbrakes surfaces. The rocket will also mount a pitot tube to precisely measure its velocity.

Note that the airbrakes will only be opened during subsonic flight. This thesis is focusing on the supersonic portion of the flight, therefore the airbrakes will be ignored and considered closed.

1.3 Mission profile and CONOPS (Concept of Operations)

VES is designed to participate in the most important European and Global competitions. The design phase has been conducted following the *Spaceport America Cup* and *European Rocketry Challenge* regulations. The goal is to carry a payload of 1 to 4 kg at a variable apogee, the highest being 9 km at Euroc. The aerodynamics analysis conducted in this work particularly focuses on the supersonic portion of the flight of the Euroc launch. The rocket will be propelled by a Cesaroni Technologies COTS solid motor. In this configuration, the rocket will be launched from a ramp

Manufacturer	Cesaroni Technology
Designation	21062O3400-P
Common Name	O3400
Motor Type	reload
Diameter	98 mm
Length	1,239 mm
Total Weight	16,842 g
Prop. Weight	10,930 g
Avg. Thrust	3,416.7 N
Initial Thrust	4,319.2 N
Max. Thrust	4,750.3 N
Total Impulse	21,062.2 Ns
Burn Time	6.2 s
Isp	197 s
Motor Case	Pro98-6GXL
Propellant	Imax

Table 1.1: Motor Specifications

that is 12m long and inclined at an angle of 84° . The vehicle should reach a maximum Mach number of $M = 1.62$ at the burnout time of 6.2s, at an altitude of 2162.4m above ground level (AGL). After motor burnout, the rocket will continue its flight unpropelled, following a ballistic trajectory up to its apogee altitude. Although the rocket is designed to reach 10000 m (AGL), the chosen motor will likely enable it to achieve an altitude of around 8200 m. This indicates that the motor is not optimal, but the choice was constrained by market availability.

The mission profile also includes the opening of airbrakes near the apogee altitude to improve the precision of reaching the apogee. However, this aspect will not be considered in the current analysis since the airbrakes will be deployed at a lower speed.

Once the vehicle reaches apogee, its primary flight computer will use altimeter data to actuate the first recovery mechanism, which separates the body into two sections and deploys the drogue parachute. This smaller parachute will decelerate the rocket to a terminal velocity of approximately 35 m/s. At 400 m above ground level, the avionics will trigger the deployment of the main parachute, and a third section will separate. All sections will remain connected via Kevlar lines. The vehicle is expected to touch down at approximately 6.5 m/s.

1.4 Simulations

The flights are simulated using various software tools and techniques. An initial round of preliminary simulations is generally conducted using the *OpenRocket* software. Originally developed for amateur rocketry, *OpenRocket* has undergone continuous upgrades over the years and is now widely used by many student teams. This software allows users to build a detailed model of the rocket by adding each component individually.

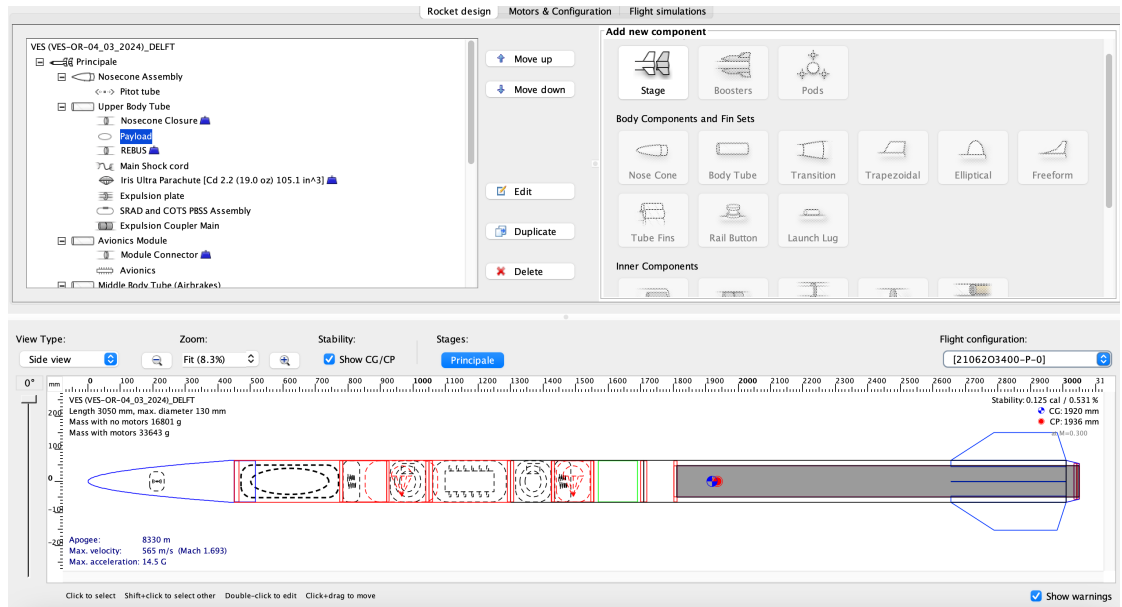


Figure 1.1: The OpenRocket interface

In Figure 1.1, the *OpenRocket* user interface is shown. The "add new component" menu allows users to incorporate pre-modeled components into the rocket. All parameters of each component, including position, density, dimensions, mass, and center of gravity, can be specified. The 2D model of the rocket is displayed and updated in real-time as components are added. Using the specified mass values,

the software calculates the center of gravity.

OpenRocket also conducts an aerodynamic analysis on the model using reduced-order methods, which, to the best of our knowledge, are based on the Barrowman equations, a widely recognized method in the USA. This analysis has proven to be quite reliable at subsonic speeds, with numerous teams validating the software's simulations through test flights.

In the "Motors and Configurations" tab, users can add a motor from a comprehensive list of commercially available off-the-shelf (COTS) motors. The software retrieves the thrust curve of the selected motor and conducts six degrees of freedom (6-DOF) simulations, providing users with all necessary plots and data.

Particular attention is always necessary when evaluating the static margin values, which are crucial for determining the rocket's stability. The static (or stability) margin is the distance between the center of gravity and the center of pressure, typically expressed in calibers (cal), where 1 cal equals one rocket diameter. The static margin is considered positive if the center of pressure is below the center of gravity (with the nose pointing up).

If the stability margin is less than 1 cal, the rocket is deemed unstable: a perturbation in its flight that induces a non-zero angle of attack would generate a lift force insufficient to return the angle of attack to zero. If the static margin is negative, the situation is even worse, as the lift force would increase the angle of attack. A static margin between 1 cal and 3.5 cal indicates a stable rocket, with the lift force effectively counteracting perturbations. However, if the static margin increases beyond this range, the rocket is considered "overstable" and may be unsafe to fly, as it could be overly sensitive to wind, potentially aligning its flight direction with the wind direction under moderately intense wind conditions. These values are not hard constraints, and the precise determination of the boundaries does not come from a rigorous theoretical study. Instead, they are general guidelines derived from years of experience in rocketry. Many of the major competitions often adopt these "common sense" rules to enhance safety.

Amateur rocketry is very popular in the USA, where the availability of wide and safe launch ranges has fostered a "validate-by-flying" philosophy. This approach, proven to be quite effective, has been widely adopted by the main competitions.

When flying in transonic and supersonic regimes, the Barrowman method loses its reliability (See 2.1). *OpenRocket* displays a warning in such cases, indicating that errors may occur.

Another popular tool for conducting simulations is the *Python* library *rocketpy*. This library is used to perform six degrees of freedom (6-DOF) simulations and, although it lacks a graphical interface, it is a more powerful instrument than *OpenRocket*. *Rocketpy* does not calculate drag coefficient values; instead, it requires the drag curve as an input. The center of pressure is computed by the software and has the same limitations as *OpenRocket*. The simulation results are generally very

precise: during the 2023 Spaceport America Cup flight, the apogee was simulated with a 0.005% error.

The main advantage of *rocketpy* is its versatility. It includes built-in classes for conducting Monte Carlo analyses, importing weather databases and forecasts, and leveraging all other open-source *Python* libraries for specific simulations. This flexibility allows for presenting data in an appealing manner and easily exporting results. Wind can be precisely modeled because the library allows users to input a wind profile curve, considering wind direction changes at each specific elevation.

Many teams also use in-house software, often *MATLAB* scripts, for simulations and aerodynamic analyses. These typically follow the *Missile DATCOM* model. Although the use of *Missile DATCOM* is restricted to the US Air Force, the original documentation is available, and many software tools have adopted and adapted its methods to achieve similar results. These methods are usually not open source, and it may take many years for a team to develop a precise one. These software tools are generally considered very reliable at supersonic speeds only for slender rockets. The goal of this work is to contribute to the creation of a reliable software tool that performs similarly, tailored specifically for the VES geometry.

Chapter 2

Center of pressure analysis

The goal of this analysis is to implement a method to calculate the rocket's center of pressure using the fins and nosecone geometry, as well as the flight conditions (angle of attack, velocity, and fluid parameters) as inputs.

The center of pressure of a body moving through a fluid is the point at which the total moment of the aerodynamic forces acting on the body is zero. The location of the center of pressure depends on the Mach number and the angle of attack. Its position shifts with the angle of attack in subsonic flight and undergoes a sudden shift when the body reaches the supersonic flight regime. Predicting the center of pressure position is crucial for ensuring flight stability.

If we consider a slender rocket, its nosecone and body tube are rotational bodies, while the fin set has usually two symmetry planes. Let's assume a zero cant angle for the fins, meaning that each fin angle of attack is the same. Sometimes, as in the NACA-TR-1307 report [1], the cant angle is also called *wing inclination angle*. These conditions imply that when the rocket flies with an angle of attack equal to zero, it generates no lift, and thus, there are no aerodynamic forces generating a moment on the rocket. Therefore, the concept of the center of pressure becomes less significant. However, a zero-lift center of pressure may be defined as the limit of the center of pressure position as the angle of attack approaches zero.

The analysis must account for the presence of shock waves. When a body moves at supersonic speed, it travels faster than the speed of sound, meaning that flow perturbations cannot propagate upstream. This results in sudden changes in pressure and density, forming shock waves. A shock wave is a non-isentropic process that significantly alters the flow properties, including pressure, temperature, and density, and results in energy loss. Flow separation may also occur due to the adverse pressure gradient introduced by the shock wave. Additionally, expansion fans, which are also associated with supersonic flow, involve a decrease in pressure and density as the flow expands around a convex corner. Both shock waves and expansion fans significantly influence the aerodynamic characteristics of a body.

2.1 Traditional analysis in rocketry: the Barrowman method

The most commonly used approach for semi-empirical analysis in rocketry is the Barrowman method [2]. This method, presented in the Master Thesis of Dr. James Barrowman, involves dividing the rocket into different sections (nose, body, and tail), analyzing each component independently, and then combining the results. The Barrowman method is based on four main hypotheses:

1. A near-zero angle of attack, $\alpha \approx 0$
2. Steady and irrotational flow
3. The rocket is assumed to be a rigid body, with all strains on aerodynamic surfaces considered negligible
4. The nosecone tip is considered a sharp point

The third hypothesis implies that no aeroelastic phenomena are causing significant strains. Fin flutter is a major cause of many failed flights. The VES fins have been developed to minimize flutter and include a safety factor of 4. The fourth hypothesis can be more or less satisfied depending on the specific nosecone geometries.

The first hypothesis is generally valid during flight when the rocket's velocity is much greater than the external flow velocity (e.g., wind velocity). However, it may not hold during the initial moments of the flight, when the wind velocity can be significant relative to the relatively low velocity of the rocket. For instance, considering a hypothetical wind velocity of 4 m/s, a rail exit velocity of 30 m/s would result in an 8° angle of attack, which is not negligible. Using this method, the computed drag coefficient (C_D) will likely be inaccurate during the first second of flight.

Regarding the second hypothesis, it can be considered acceptable in most part of the trajectory.

Due to the semi-empirical nature of the Barrowman method, the underlying hypotheses are not strictly determinant for obtaining valid results. Most of the coefficients are calculated based on experimental parameters, and numerous corrective empirical factors are introduced. This method has proven effective multiple times in rocket and missile engineering, though Polito Rocket Team typically trusts it only up to Mach 0.9. Most importantly, the corrections used by Barrowman do not yield valid results for more complex fin geometries, such as the VES diamond profile fins.

The Barrowman method can be used to calculate all the force coefficients and the center of pressure position. However, the wave drag and supersonic center of

pressure position are not analyzed in the Master Thesis [2]. However, a method for calculating the supersonic center of pressure is mentioned and briefly described in the Appendix and it had been previously published by Barrowman himself as a NASA Technical Memorandum [3].

2.2 The FIN algorithm

The FIN computer program [3] is based on the Busemann's Second Order Airfoil Theory[4], which will be described in the next subsection. The numerical solutions of Busemann's equations allow the program to calculate the center of pressure of each airfoil strip. The overall coefficients are then calculated by spanwise summing the contributions of these airfoil strips. FIN can also account for the effect of the fin-tip Mach cone. Barrowman claims that this method produces less than a 10% error compared to wind tunnel tests (see [3], summary). This method can only be applied to vehicles with four identical fins, as is the case with the VES rocket. The applicable airfoil shapes for the FIN method are shown in Figure 2.1.

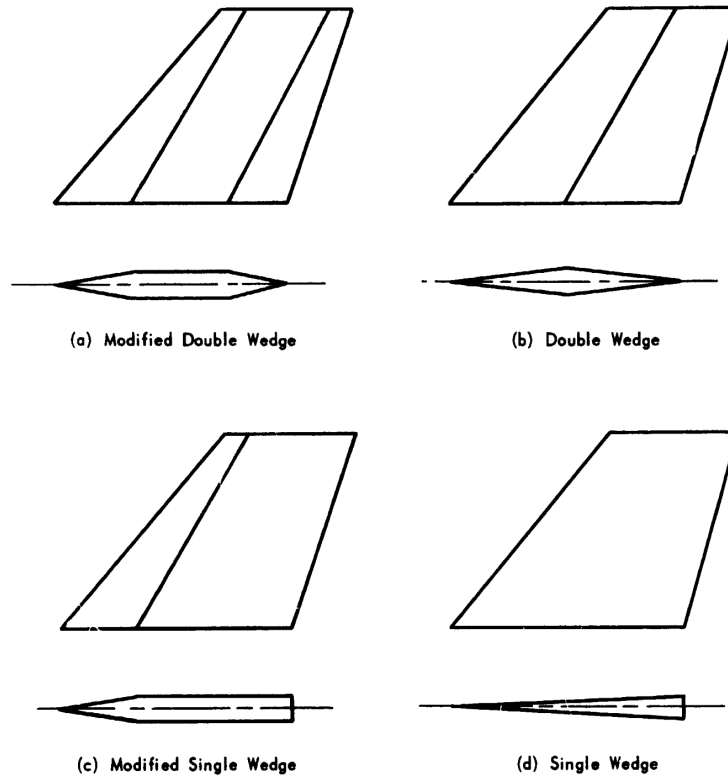


Figure 2.1: Fin profiles that can be analyzed using FIN [3]

The VES fins have a variable profile, because the leading and trailing edges meet at a spanwise distance of $y_{lim} = 0.07348m$. If the spanwise coordinate is lower than this, the profile will be a "modified double wedge" (see 2.1), farther it becomes a "double wedge" (see appendix D). The original *FIN* code has been modified to account for that.

2.2.1 Busemann's Second Order Airfoil Theory

Busemann introduced this theory during a lecture at the *Fifth Volta Conference* in Rome in 1935. Unfortunately, the original paper authored by Busemann is extremely difficult to find. These difficulties in obtaining the original report have caused problems over the years, also because, as stated by Hilton ([5], page 226), in the original conference the third coefficient introduced by Busemann was misquoted. This has caused errors in the past; for example, Barrowman reports the formula correctly in the *FIN* report [3], but in the code in the appendix, the third coefficient is incorrect.

Busemann's theory can be applied if the following assumptions are respected (as reported by Barrowman):

1. All parts of the airfoil are in supersonic flow and make small angles with the flow. This implies a small angle of attack.
2. The leading and trailing edges are sharp.
3. The shock waves are all attached.
4. Each region of flow over the surface acts independently of the others.

The first geometrical limitation is satisfied by the VES fins. Regarding the second assumption, although the original fin design features sharp edges, they have been modified to slightly blunted edges in the advanced stages of CAD design. This modification is necessary for manufacturing, as the fin set will be 3D printed and reinforced with two layers of nylon fibers. The design must therefore adhere to the minimal curvature radius allowed by the printer for nylon fiber layering. Although a curvature is present, it is sub-millimetric, and the leading edges will be sanded to increase their sharpness.

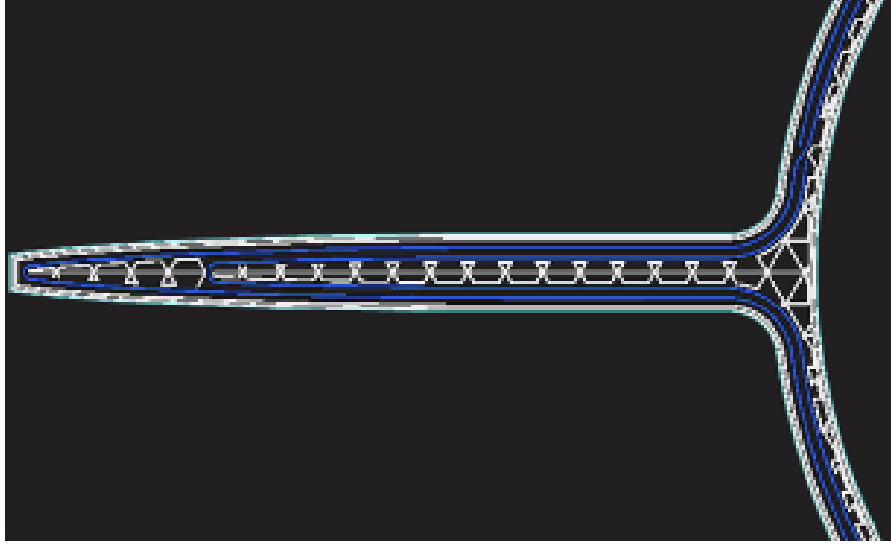


Figure 2.2: Cross section of a fin. The blue lines are the two layers of reinforcement nylon fibers

The third and fourth assumptions are more restrictive, and it is challenging to determine if they are fully met without a CFD analysis. Hilton explains how the third in particular is strictly linked to the first one.

Hilton adds an important consideration regarding what *small angle* means, discussing the limit wedge angle for which the theory remains valid. On page 239 of reference [5], a significant chart is presented. It indicates that Busemann's second-order theory is valid for a maximum wedge angle of approximately 10° ; however, this limit decreases when the Mach number is relatively low. The wedge angle for VES fins is about 2.56° , and for this angle, the chart suggests not to expect precise results at a Mach number of around $M = 1.25$. This consideration will be crucial when analyzing the final results.

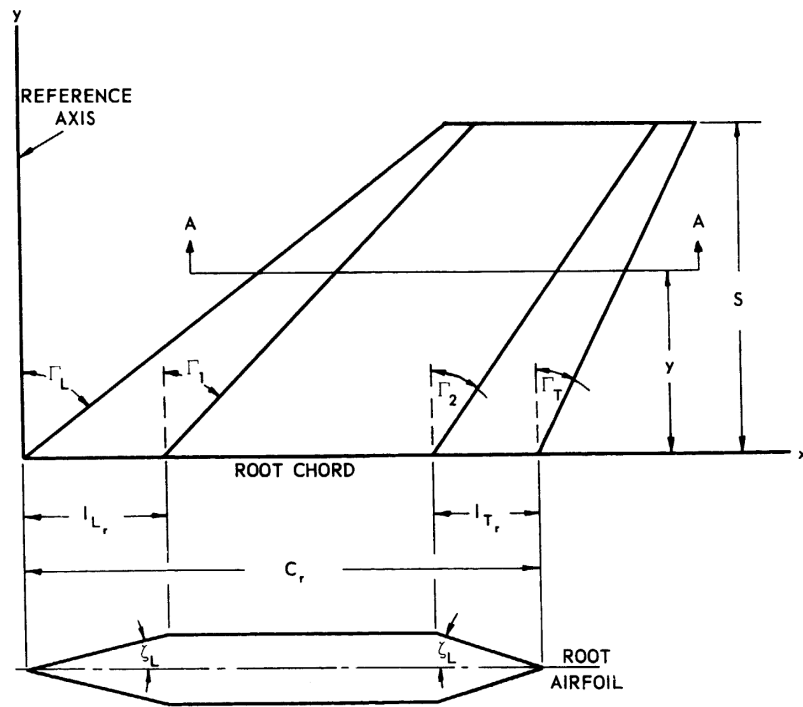


Figure 2.3: Fin geometry parameters [3]

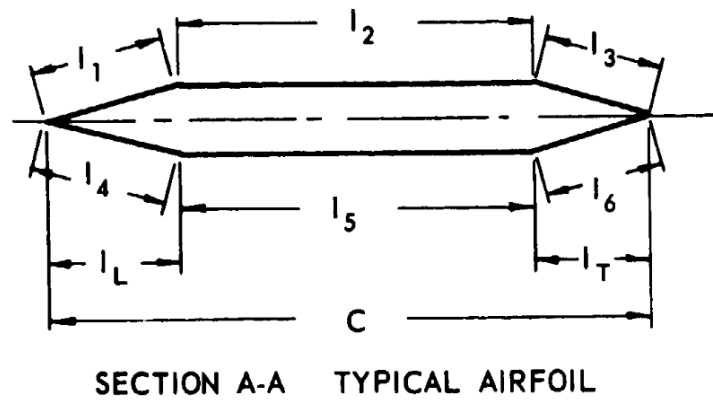


Figure 2.4: Fin section dimensions and parameters [3]

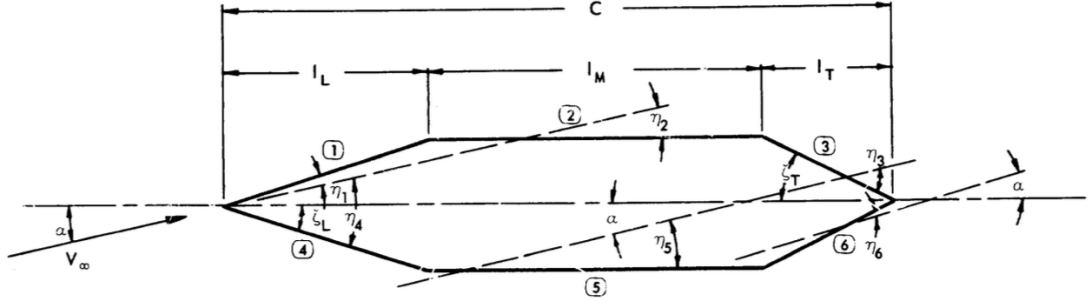


Figure 2.5: Fin profile angles [3]

Table 2.1: VES Fin Geometry Parameters

Parameter	Value
Γ_L	$\frac{\pi}{3}$ (radians)
Γ_1	$\frac{\pi}{3}$ (radians)
Γ_2	$-\frac{9}{4}\pi$ (radians)
Γ_T	$-\frac{9}{4}\pi$ (radians)
ζ_L	2.56°
ζ_T	2.56°
s	0.114 (m)
c_r	0.3377 (m)
l_{Lr}	0.09587 (m)
l_{Tr}	0.09073 (m)
X_L	0

Under the mentioned hypothesis, Busemann expresses the local pressure coefficient as a third order expansion:

$$C_{P_i} = \frac{P_i - P_\infty}{q_\infty} = K_1 \eta_i + K_2 \eta_i^2 + K_3 \eta_i^3 - K^* \eta_L^3 \quad (2.1)$$

The Busemann's coefficients are expressed as:

$$\begin{aligned} \mathbf{K}_1 &= \frac{2}{\beta} \\ \mathbf{K}_2 &= \frac{1}{2} \left\{ (M^2 - 2)^2 + \gamma M^4 \right\} (M^2 - 1)^{-2} \\ \mathbf{K}_3 &= \frac{(\gamma + 1)M^8 + (2\gamma^2 - 7\gamma - 5)M^6 + 10(\gamma + 1)M^4 - 12M^2 + 8}{6\beta^7} \end{aligned} \quad (2.2)$$

Where $\beta = \sqrt{M^2 - 1}$. According to Barrowman[3], if the flow has been preceded by a single shock wave anywhere upstream, the factor K^* is:

$$\mathbf{K}^* = \frac{(\gamma + 1)M^4 [(5 - 3\gamma)M^4 + 4(\gamma - 3)M^2 + 8]}{48} \quad (2.3)$$

And otherwise, it must be set to zero. According to reference [5], considering $\gamma = 1.4$ for the air, the coefficient K_3 through a single shock becomes:

$$b_3 = \frac{0.36M^8 - 1.493M^6 + 3.6M^4 - 2M^2 + 1.33}{\beta^7} \quad (2.4)$$

Barrowman uses this term in the code without theoretically introducing it. Hilton adds an important observation: at $M = 1.24$ and $M = 2.54$, $K_3 = K^*$, making a fourth-order term potentially relevant because the overall third-order term is zero. The expression for b_3 does not add anything to the theory, but is reported here for clarity since in the *FORTRAN* code in the appendix of [3] the third order coefficient is defined using this formula. In figure 10.24 of ref. [5], Hilton also provides an interpretation of each of Busemann's coefficients. The first-order term, $K_1\eta$, produces lift and drag on a profile but no half-chord moment. The second-order term, $K_2\eta^2$, produces no lift or drag since the angle is squared, causing both the upper and lower surfaces to experience a pressure variation of the same sign when considering only this term. Unlike the first-order term, however, it generates a non-zero half-chord moment. The third-order term accounts for the effects of shock waves, which are virtual surfaces within the flowfield through which the gas experiences sudden changes, such as a sudden increase in pressure.

The critical step of the FIN algorithm is the following. Each airfoil strip is divided into two portions, at the intersection of the fin strip and the Mach cone, to consider the fin tip mach cone. For the region within the Mach cone, a 1/2 correction factor is applied. The center of pressure is taken at the midpoint of that portion (eq. 2.6). The local lift, drag, center of pressure and lifting moment for the i-th surface of a strip are calculated as:

$$F_{l_i} = C_{p_i} d_i \left(1 - \frac{r_i}{2} \right) \quad (2.5)$$

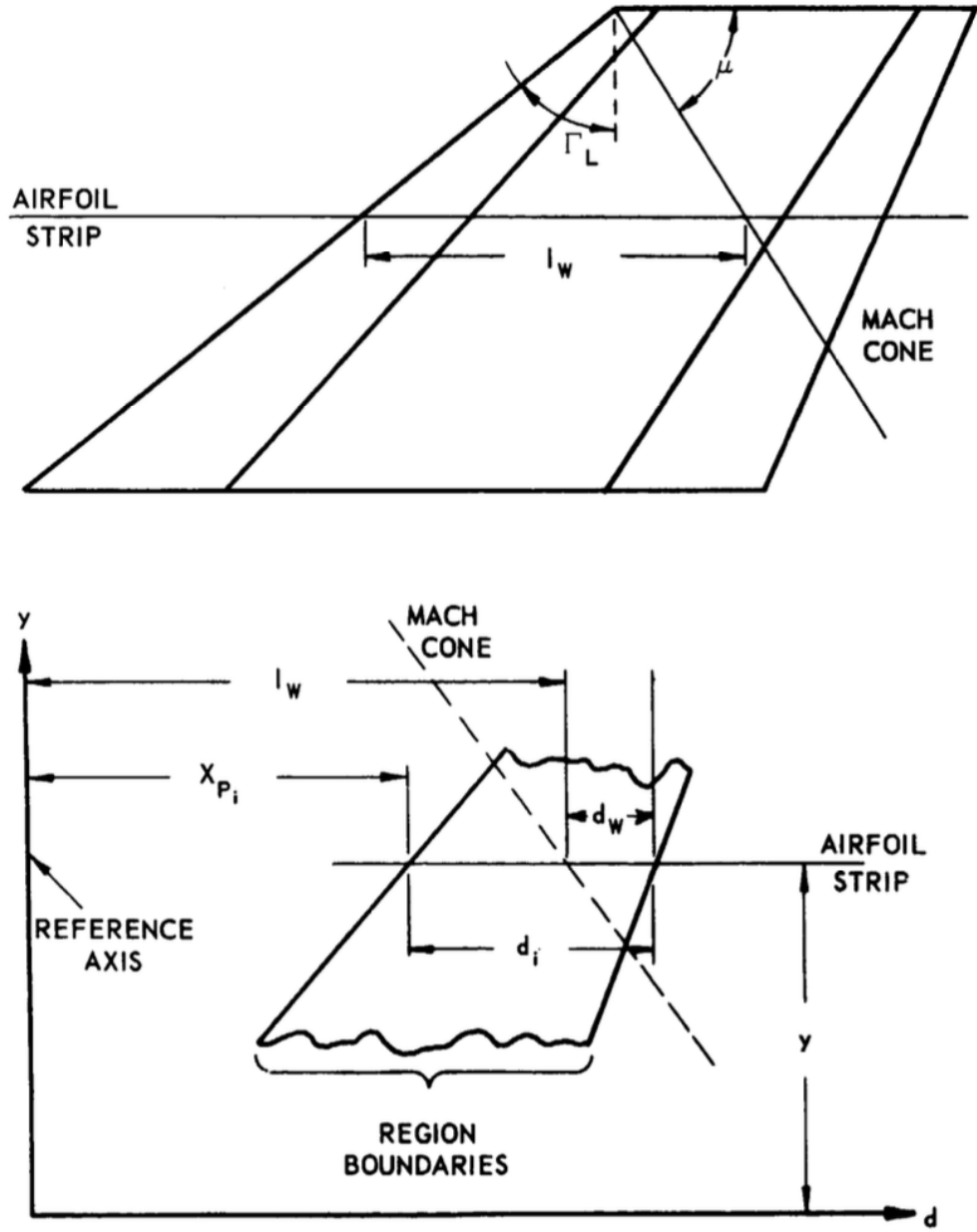


Figure 2.6: Fin Tip Mach cone schematics [3]

$$F_{d_i} = C_{p_i} n_i \left(1 - \frac{r_i}{2}\right) \quad (2.6)$$

In this context, d_i and n_i are, respectively, the parallel and normal to the freestream

components of l_i .

$$\bar{X}_i = \frac{1/2d_i \left[1 - r_i + \frac{r_i}{2} + \frac{x_{pi}}{1_i} (2 - r_i) \right]}{1 - \frac{r_i}{2}} \quad (2.7)$$

Note that this coordinate must be set to 0 for the "degenerate" sides of the strip, so $\bar{X}_2 = \bar{X}_5 = 0$, otherwise the code will produce *NaN* exceptions. The physical interpretation for this override is the fact that those sides do not exist on the "double wedge" portion of the fin.

$$M_{p1} = F_{l_i} \bar{x}_i \quad (2.8)$$

These provide the results for a single portion of a single airfoil strip. To get the overall result for the j -th airfoil strip, the results of the six modified double wedge fin sides are added together:

$$F_{1j} = \sum_{i=1}^6 F_{l_i} \quad (2.9)$$

$$F_{d_j} = \sum_{i=1}^6 F_{d_i} \quad (2.10)$$

$$M_{P_j} = \sum_{i=1}^6 M_{p_i} \quad (2.11)$$

Defining y as the general spanwise position of the strip, it is also possible to calculate the rolling moment around the root chord.

$$M_{r_j} = y F_{l_j} \quad (2.12)$$

Adding everything up in a spanwise direction, the total characteristics are obtained.

$$F_l = \sum_{j=1}^n F_{l_j} \quad (2.13)$$

$$F_d = \sum_{j=1}^n F_{d_j} \quad (2.14)$$

$$M_p = \sum_{j=1}^n M_{p_j} \quad (2.15)$$

$$M_r = \sum_{j=1}^n M_{r_j} \quad (2.16)$$

Considering that these equations are already divided by the ambient pressure q_∞ and strip width Δy , and keeping in mind that drag, lift, and pitching moment are

caused by the forces on two fins only (as the forces on the other two cancel each other out), while the rolling moment is generated by all four fins, the associated coefficients can be computed:

$$C_L = \frac{2F_1 \cdot \Delta y}{A} \quad (2.17)$$

$$C_{DW} = \frac{2F_d \cdot \Delta y}{A} \quad (2.18)$$

$$C_m = \frac{2M_p \cdot \Delta y}{AL} \quad (2.19)$$

$$C_1 = \frac{4M_r \cdot \Delta y}{AL} \quad (2.20)$$

The appropriate forces, moments, and coefficients can be used to ultimately calculate the position of the center of pressure.

$$\bar{X} = \frac{M_p}{F_1 \cos a} \quad (2.21)$$

$$\bar{Y} = \frac{M_r}{F_1} \quad (2.22)$$

2.2.2 Geometric calculations

To calculate d_i and n_i in this specific application, some simple geometrical considerations are necessary. First, the distance X_L between the reference axis and the specific airfoil being analyzed at the spanwise position y , and the chord of that airfoil, need to be calculated:

$$X_L = y \times \tan \Gamma_L \times \cos a \quad (2.23)$$

$$C = y (\tan \Gamma_T - \tan \Gamma_L) + C_r \quad (2.24)$$

Now, with reference to Figure 2.3, we calculate:

$$\begin{cases} l_L = y (\tan \Gamma_1 - \tan \Gamma_L) + l_{Lr} & \text{if } y < y_{lim} \\ l_L = \frac{(s-y)(\tan \Gamma_L - \tan \Gamma_T)}{2} & \text{if } y > y_{lim} \end{cases} \quad (2.25)$$

$$\begin{cases} l_T = y (\tan \Gamma_1 - \tan \Gamma_L) + l_{Lr} & \text{if } y < y_{lim} \\ l_T = l_L & \text{if } y > y_{lim} \end{cases} \quad (2.26)$$

$$\begin{aligned}
 l_1 = l_4 &= \frac{l_L}{\cos \zeta_L} \\
 l_3 = l_6 &= \frac{l_T}{\cos \zeta_T} \\
 \begin{cases} l_2 = l_5 = C - l_L - l_T & \text{if } y < y_{lim} \\ l_2 = l_5 = 0 & \text{if } y > y_{lim} \end{cases}
 \end{aligned} \tag{2.27}$$

Then, with reference to Figure 2.5:

$$\begin{aligned}
 \eta_1 &= \zeta_L - \alpha \\
 \eta_2 &= -\alpha \\
 \eta_3 &= -\zeta_T - \alpha \\
 \eta_4 &= \zeta_L + \alpha \\
 \eta_5 &= \alpha \\
 \eta_6 &= -\zeta_T + \alpha
 \end{aligned} \tag{2.28}$$

The normal and parallel components of l_i can now be determined.

$$\begin{aligned}
 d_i &= l_i \cos \eta_i (i = 1 \rightarrow 6) \\
 n_i &= l_i \sin \eta_i (i = 1 \rightarrow 6)
 \end{aligned} \tag{2.29}$$

2.3 Overall center of pressure computation

Using the FIN program allows for obtaining the center of pressure of the tail alone. However, to consider the overall center of pressure, another procedure is necessary. For implementation in simulators and usefulness in preliminary analysis, the method should not be excessively computationally expensive. The most suitable procedure available is described in NACA-TR-1307 [1]. This method calculates the overall center of pressure similarly to the concept in the Barrowman Master Thesis [2]. It computes the center of pressure of each component along with their lift coefficients, and then applies interference factors to correct the result. Similar to the FIN program, this method is applicable only if the angle of attack of the vehicle is small and the wing cant angle is small [1].

The gradient of the tail lift coefficient and the center of pressure are crucial in this calculation. The interference factors are significantly influenced by these parameters, as demonstrated by the equations in the following paragraphs. Therefore, obtaining these values with low error is essential for achieving reliable results. The method provides solutions for rectangular, triangular, or trapezoidal wings, but not for the particular modified double wedge geometry of the VES fins. This is why FIN was used to determine these values rather than solely relying on the NACA-TR-1307 [1] report.

The NACA-TR-1307 [1] report can be used for subsonic, transonic, and supersonic flows. In this context, focus is placed on the supersonic portion of the flight. The report is also applicable if the rocket mounts both tail wings and another set of wings, which introduces additional interference coefficients (wing-body, wing-tail due to wing vortices). However, these coefficients are ignored since VES does not mount wings other than the tail wings.

2.3.1 Lift on nose

The rocket nosecone is a body of revolution modeled using a $\frac{1}{2}$ power function. Setting the origin of the longitudinal axis at the nose tip, the radius at any point before the nose shoulder (where the cylindrical body tube begins) is expressed as:

$$y = r_n \left(\frac{x}{l_s} \right)^{\frac{1}{2}} \quad (2.30)$$

Therefore, its volume is:

$$V = \int_0^l \iint_A dz dy dx = \int_0^l \left[\iint_{\left[0, x^{\frac{1}{2}}\right] \times [0, 2\pi]} r dr d\theta \right] = \int_0^l (2\pi) \frac{x}{2} dx = 2\pi \frac{l^2}{4} = \frac{\pi l^2}{2} \quad (2.31)$$

The volume is needed to calculate the tail-nose interference factor:

$$K_N = \frac{2\pi r_N^2}{S_W (C_{L_\alpha})_W} \quad (2.32)$$

where $(C_{L_\alpha})_W$ is the lift coefficient gradient of the wings alone and S_W is the wing surface. Note that, since we are calculating the tail wings coefficients using a determined reference area A_{ref} , it is important to set $A_{ref} = S_W$. The nosecone lift coefficient is given by:

$$C_{L_N} = K_N (C_{L_\alpha})_W \alpha \quad (2.33)$$

Finally, the position of the nose center of pressure is obtained as:

$$\bar{l}_N = l_S \left(1 - \frac{V_S}{\pi r_N^2 l_S} \right) \quad (2.34)$$

This method, derived from slender-body theory, is expected to be sufficiently accurate for the VES nosecone. For higher accuracy requirements, alternative theories such as viscous cross-flow theory [6] are available. Since the nosecone follows a parabolic equation, it forms an angular point where it joins the body tube. The original report does not suggest any method to account for this. A new method is proposed in the next paragraph.

2.3.2 Nosecone approximation and expansion correction

The idea comes from a method suggested in the Barrowman report [2], page 21. The nosecone is approximated by a set of tangent conical frustums. A *Matlab* script has been created to accomplish this (see appendix, C), and a good approximation has been achieved by selecting appropriate points on the surface of the nosecone, called *tangent nodes*. These nodes are obtained by intersecting the tangent lines at these points, which define the segments of each frustum. The script selects the tangent nodes using a constant interval in the gradient, possible because the gradient of the nosecone shape function is injective (each y value is associated with one x value only, meaning the curve does not reach a maximum and then decrease again). This method effectively approximates the curve around the gradient maximum.

The tip and the trailing edge are forced to be nodes.

After identifying the tangent nodes on the nosecone curve, two possible methods to obtain the final approximated curve were considered:

1. Joining the tangent nodes directly. This approach avoids numerical cancellation problems, works with any number of desired segments, and with any curve.
2. Drawing tangents at two consecutive tangent nodes and using the intersections between these tangents as the final nodes.

The latter method is more precise but can cause numerical cancellation problems with some curves, such as the $1/2$ power curve. It works better with the Von Karman nose curve. In general, the first method has been considered preferable by Polito Rocket Team, because the loss in precision is not significant, and the number of segments can always be increased to better approximate the nosecone curve.

The flow characteristics over the first section, which is the only one that is effectively a cone and not just a conical section, can be determined by numerically solving the Taylor-Maccoll equations. These are ordinary differential equations, and their solutions are tabulated [7], page 137. The code developed by the Polito Rocket Team utilizes an existing open-source *Matlab* script for this purpose. For the flow over the subsequent sections, the shock-expansion method can be employed [7].

$$\Delta\theta = \sqrt{\frac{\gamma+1}{\gamma-1}} \tan^{-1} \sqrt{\frac{\gamma-1}{\gamma+1}} (M_n^2 - 1) - \tan^{-1} \sqrt{\frac{\gamma-1}{\gamma+1}} (M_i^2 - 1) - \left[\tan^{-1} \sqrt{M_n^2 - 1} - \tan^{-1} \sqrt{M_i^2 - 1} \right] \quad (2.35)$$

Equation 2.35 is the Prandtl-Meyer expansion formula. Note that the $\Delta\theta$ term represents the angle of deflection encountered when moving from one section to

the next. Its value can be obtained from the nosecone approximation, as it is known. M_i is the Mach number of the flow over the preceding section. For the first expansion, this value is known from the Taylor-Maccoll solution. γ is the ratio of specific heats and is a known quantity. This means that, for the first expansion, M_n , the Mach number over the following section, is the only unknown. The equation is transcendental, but it can be solved numerically, and numerous open-source scripts and *Matlab* functions are available for this purpose. Once M_n for the first expansion is obtained, it can be used as M_i for the next expansion, and the procedure continues until the Mach number over the entire nose is determined. The final goal is to correct equations 2.2 by incorporating the results of these calculations. One possible approach is to use the ratio $\frac{M_i}{M_n}$ from the last section, where the actual angular point is present (while all other sections are results of the approximation), and multiply it by the freestream Mach number. This "corrected" value can then be used in equations 2.2 instead of the freestream Mach number.

Another possibility could be to use the Taylor-Maccoll equation, which can describe the flow field behind the shock, to obtain the Mach number on the nosecone surface immediately behind the shock wave and then use it to calculate the Busemann coefficients. In this case, a different K_3 coefficient would be needed:

$$K_3 = \frac{0.4M^8 - 1.813M^6 + 4M^4 - 2M^2 + 1.33}{(M^2 - 1)^{3.5}} \quad (2.36)$$

This coefficient is indicated by Hilton [5] for expansive, shock-free flow. This would involve calculating the flow conditions after the first shock wave and then proceeding with the calculation while ignoring its presence. However, in the practical case of the VES rocket, this approach would not work accurately due to the geometry and flight conditions. This can be shown with some estimations by first examining the half-angle of the cone as approximated by the script presented above, which is approximately 30° when the nose-cone is divided into 20 sections. Inputting this angle into the Taylor-Maccoll solver script with the correct flight conditions and $M = 1.6$, which is approximately the maximum value expected for the VES flight, the result for the Mach number behind the shock is $M_0 = 1.10$. Referring to figure 10.28 of Hilton's reference and considering that the VES fins' maximum surface inclination is 2.56° , it becomes clear that Busemann's theory does not apply for such a small Mach number. This method presents one major limitation: it assumes zero angle of attack, which is not the case in all the rest of the calculations. A CFD analysis may confirm if the result is better than just ignoring the expansion at the angular point.

2.3.3 Lift on wing in presence of body

The analysis falls in the case $\alpha \neq 0$ and $\delta = 0$. The coefficient factor varies depending on the fin geometry and aspect ratio of the fins.

Some possible alternatives for calculation include slender-body theory, which is generally satisfactory for slender wing-body combinations (Eq. 2.37. See also [1], page 4, *Lift on wing in presence of body*, and page 5, figure 2); or upwash theory (Eq. 2.38), which assumes the wing operates in the upwash field of the body alone (neglecting the presence of the nose).

$$K_{W(B)} = \frac{2}{\pi} \frac{\left\{ \left(1 + \frac{r^4}{s_t^4}\right) \left[\frac{1}{2} \tan^{-1} \frac{1}{2} \left(\frac{s_t}{r} - \frac{r}{s_t} \right) + \frac{\pi}{4} \right] - \frac{r^2}{s_t^2} \left[\left(\frac{s_t}{r} - \frac{r}{s_t} \right) + 2 \tan^{-1} \frac{r}{s_t} \right] \right\}}{\left(1 - \frac{r}{s_t}\right)^2} \quad (2.37)$$

$$K_{W(B)} = \frac{\frac{1}{2}(1 + \lambda) - \frac{\lambda r}{s_t} - \frac{r^2(1-\lambda)}{s_t^2 - r^2} \ln \left(\frac{s_t}{r} \right)}{\frac{1}{2} \left(\frac{s_t - r}{s_t + r} \right) (1 + \lambda)} \quad (2.38)$$

A comparison of these two methods is presented in figure 2, page 5 of [1]. The difference between the two results is relatively small. After comparing the results, the linear theory, using eq. 2.37, has been selected for use in the code. Linear theory yields a final center of pressure that is slightly closer to the nose tip, which is a conservative choice in terms of rocket stability (as explained in 1.4, the rocket is more stable if the center of pressure is further from the nose tip).

This results in the overall center of pressure of the wing position:

$$\bar{l}_{W(B)} = l_W + \left(\frac{\bar{x}}{c_r} \right)_{W(B)} (c_r)_W \quad (2.39)$$

2.3.4 Lift on body in presence of wing

The wing-on-body interference coefficient can be obtained through two different methods. Slender-body theory provides a result theoretically applicable across any Mach range, under the usual assumptions of small angle of attack and sharp edges ([1], page 6, *Lift on body due to wing*). The coefficient is calculated using Equation 2.40:

$$K_{B(W)} = \frac{\left(1 - \frac{r^2}{s_t^2}\right)^2 - \frac{2}{\pi} \left\{ \left(1 + \frac{r^4}{s_t^4}\right) \left[\frac{1}{2} \tan^{-1} \frac{1}{2} \left(\frac{s_t}{r} - \frac{r}{s_t} \right) + \frac{\pi}{4} \right] - \frac{r^2}{s_t^2} \left[\left(\frac{s_t}{r} - \frac{r}{s_t} \right) + 2 \tan^{-1} \frac{r}{s_t} \right] \right\}}{\left(1 - \frac{r}{s_t}\right)^2} \quad (2.40)$$

The second method was developed by Lagerstrom and Van Dyke [8]. Based on slender-body theory, a non-expanding section of a body in uniform flow generates

no lift. Therefore, the lift on the body primarily results from the wing mounted on it. Lagerstrom and Van Dyke proposed that each point on the wing acts as a source of disturbances that propagate downstream in all directions. They assumed that the body only displaces these disturbances downstream without reducing their intensity. The influence of these disturbances is restricted to the body region within the Mach helix defined by the wings. To simplify calculations, a planar model can be employed, considering the planar section of the body instead of its full 3D representation (see Figure 2.7).

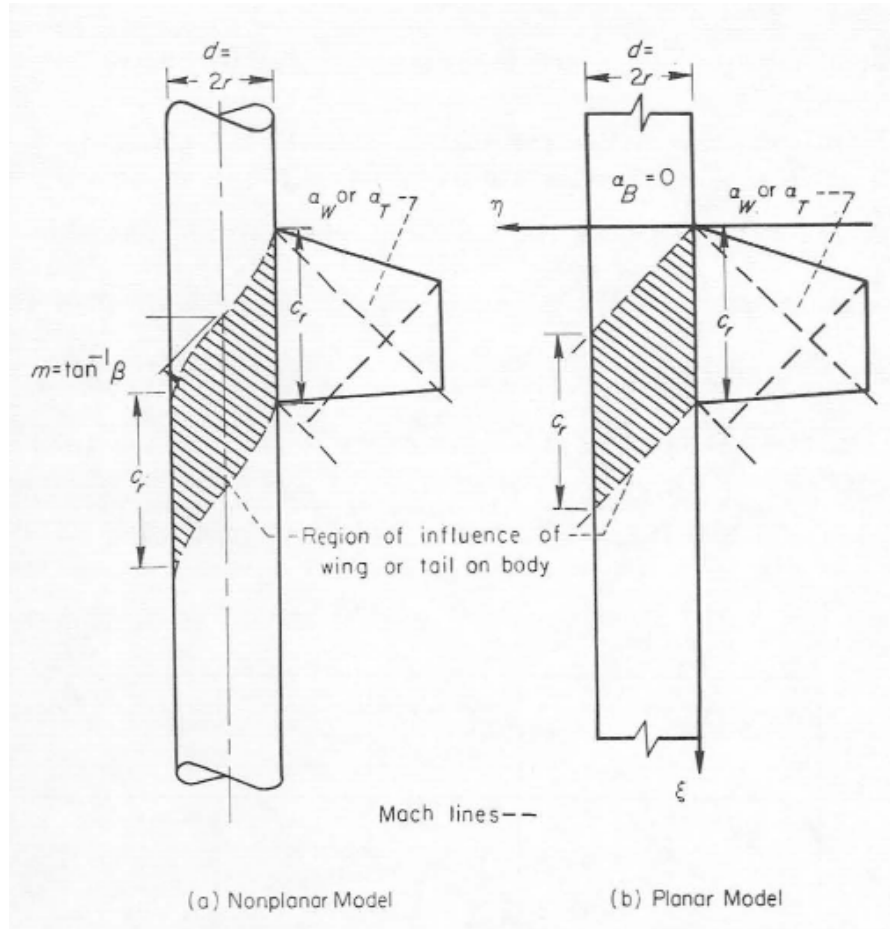


Figure 2.7: 3D and equivalent planar model introduced Van Dyke and Lagerstrom [1]

Under this assumptions, the result for the interference factor if the leading edge

is supersonic is expressed as:

$$\begin{aligned}
 K_{B(W)} = & \frac{8\beta m}{\pi\sqrt{\beta^2 m^2 - 1}(1 + \lambda) \left(\beta \frac{d}{c_r}\right) \left(\frac{s_t}{r} - 1\right) (\beta C_{L_\alpha W})} \times \\
 & \left\{ \left(\frac{\beta m}{1 + \beta m} \right) \left[\frac{(\beta m + 1) \frac{\beta d}{c_r} + \beta m}{\beta m} \right]^2 \cos^{-1} \left[\frac{1 + (1 + \beta m) \beta \frac{d}{c_r}}{\beta m + (\beta m + 1) \frac{\beta d}{c_r}} \right] + \right. \\
 & \frac{\sqrt{\beta^2 m^2 - 1}}{(\beta m + 1)} \left[\sqrt{1 + 2 \frac{\beta d}{c_r}} - 1 \right] - \\
 & \left. \frac{\sqrt{\beta^2 m^2 - 1}}{\beta m} \left(\frac{\beta d}{c_r} \right)^2 \cosh^{-1} \left(1 + \frac{c_r}{\beta d} \right) - \frac{\beta m}{1 + \beta m} \cos^{-1} \left(\frac{1}{\beta m} \right) \right\}
 \end{aligned} \tag{2.41}$$

Due to geometrical constraints, eq. 2.41 can be applied only if:

$$\beta A(1 + \lambda) \left(\frac{1}{\beta m} + 1 \right) \geq 4 \tag{2.42}$$

The *Matlab* code will use eq. 2.41 if the condition expressed by 2.42 is respected, otherwise the only remaining option is slender-body theory, so eq. 2.40. The coefficient is used to calculate, with $\alpha \neq 0$:

$$C_{L_{B(W)}} = K_{B(W)} (C_{L_\alpha})_W \alpha \tag{2.43}$$

Sticking to the planar model proposed by Lagerstrom and Van Dyke, it is also possible to calculate the moment of lift carried onto the body by wing perturbations:

$$\begin{aligned}
 M_{B(W)} = & \frac{4q_\infty \alpha_W m}{3\pi\beta} c_r^3 \left\{ \sqrt{1 + \frac{2\beta d}{c_r}} \left[\frac{2m\beta + 5}{3(m\beta + 1)^2} + \frac{\beta d/c_r}{3(m\beta + 1)} - \frac{(\beta d/c_r)^2}{\beta m} \right] + \frac{1}{\sqrt{m^2\beta^2 - 1}} \left[\left(1 + \frac{\beta d}{c_r} \right)^3 - \frac{(\beta d/c_r)^3}{m^2\beta^2} - \right. \right. \\
 & \left. \left. \frac{1}{(1 + m\beta)^2} \right] \cos^{-1} \left[\frac{1 + \frac{\beta d}{c_r}(m\beta + 1)}{m\beta + \frac{\beta d}{c_r}(m\beta + 1)} \right] + \left(\frac{\beta d}{c_r} \right)^3 \frac{1}{m^2\beta^2} \cosh^{-1} \left(1 + \frac{c_r}{\beta d} \right) - \left[\frac{2m\beta + 5}{3(m\beta + 1)^2} \right] - \frac{1 - \left(\frac{1}{m\beta + 1} \right)^2}{\sqrt{m^2\beta^2 - 1}} \cos^{-1} \frac{1}{m\beta} \right\}
 \end{aligned} \tag{2.44}$$

Finally, the relative center of pressure location can be found using:

$$\left(\frac{\bar{x}}{c_r} \right)_{B(W)} = \frac{M_{B(W)}}{L_{B(W)} c_r} = \frac{M_{B(W)}}{K_{B(W)} L_W c_r} \tag{2.45}$$

And its position relative to the standard reference frame, so relative to the nose tip:

$$\bar{l}_{B(W)} = l_W + (c_r)_W \left(\frac{\bar{x}}{c_r} \right)_{B(W)} \tag{2.46}$$

2.3.5 Overall center of pressure

The overall center of pressure is obtained calculating the weighted average, with the weighs being the lift coefficients.

$$\bar{l}_C = \frac{\bar{l}_N (C_L)_N + \bar{l}_{W(B)} (C_L)_W + \bar{l}_{B(W)} (C_L)_{B(W)}}{(C_L)_N + (C_L)_W + (C_L)_{B(W)}} \quad (2.47)$$

Some considerations must be made regarding the lift and moment components carried onto the body due to the presence of the wing. Various methods for calculating $K_{B(W)}$ are introduced in reference [1], including equations 2.40 and 2.41, but also equation 30 of the original report:

$$\begin{aligned} K_{B(W)} \left[\beta (C_{L_\alpha})_W \right] (\lambda + 1) \left(\frac{s}{r} - 1 \right) = \\ \frac{8}{\pi \sqrt{\beta^2 m^2 - 1}} \left(\frac{\beta d}{c_r} \right) \left[\left(1 + \frac{m c_\tau}{d} \right)^2 \cos^{-1} \left(\frac{m \beta + \frac{c_\tau}{\beta d}}{1 + \frac{m c_\tau}{d}} \right) - \right. \\ \left. m^2 \beta^2 \left(\frac{c_\tau}{\beta d} \right)^2 \cos^{-1} \left(\frac{1}{m \beta} \right) + m \beta \left(\frac{c_\tau}{\beta d} \right)^2 \sqrt{m^2 \beta^2 - 1} \sin^{-1} \frac{\beta d}{c_\tau} - \right. \\ \left. \sqrt{m^2 \beta^2 - 1} \cosh^{-1} \frac{c_\tau}{\beta d} \right] \end{aligned} \quad (2.48)$$

This equation is only valid if there is no afterbody. Later in [1], on page 17, when discussing interference factors between the wings and the fuselage, the authors explain how different theories are more or less accurate depending on the geometry of the fins, their aspect ratio, and the slenderness ratio of the rocket. Adopting the wrong theory, according to the report, may result in a center of pressure that is too far aft or forward compared to the correct one, determined experimentally. Unfortunately, no experimental data are available for VES, and the CFD result will be taken as the correct one.

The report specifies that, in some conditions, wing-alone theory gives better results than other methods, and ignoring the interactions that carry some lift onto the fuselage due to the wing's presence generally yields better results. Therefore, results will be presented also using the formula:

$$\bar{l}_C = \frac{\bar{l}_N (C_L)_N + \bar{l}_{W(B)} (C_L)_W}{(C_L)_N + (C_L)_W} \quad (2.49)$$

which eliminates the fuselage contribution.

Chapter 3

Results and conclusions

3.1 Computation results

All the simulations have been conducted using the parameters of Table 3.1. The air density value was obtained using *Rocketpy* simulations, utilizing the *ERA5* database for atmospheric conditions. *ERA5* is a comprehensive database that compiles significant weather and atmospheric data for any location on the planet. For this simulation, data averaged over the past fifteen years were employed. The selected air density and temperature correspond to the altitude of maximum Mach number, $h = 1770m$.

Figures 3.1, 3.2, 3.3, 3.4, and 3.5 depict the results of simulations across various

Table 3.1: Freestream parameters

Parameter	Value
ρ_{inf}	1.03 kg/m ³
T_{inf}	282.6 K

angles of attack, comparing both the complete slender-body theory approach and the wing-alone approach, as discussed in the section 2.3.5. A notable observation emerges: when employing the wing-alone theory, the center of pressure is positioned farther forward compared to its calculation using the complete theory under identical conditions. This suggests that the lift exerted on the body can be considered as applied at a point farther from the nose tip than indicated by the wing-alone center of pressure. Alternatively, the center of pressure of the body alone, influenced by the wings, is not located near the geometrical center of the body, but closer to the tail.

A plausible explanation is that the body itself generates negligible lift, with the wings primarily responsible for its overall lift. Consequently, the body tube

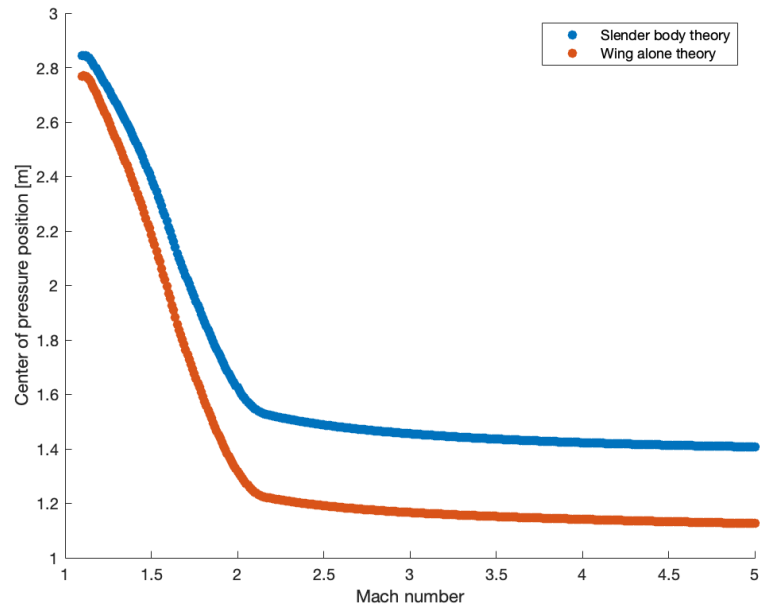


Figure 3.1: Results for $\alpha = 0.5^\circ$

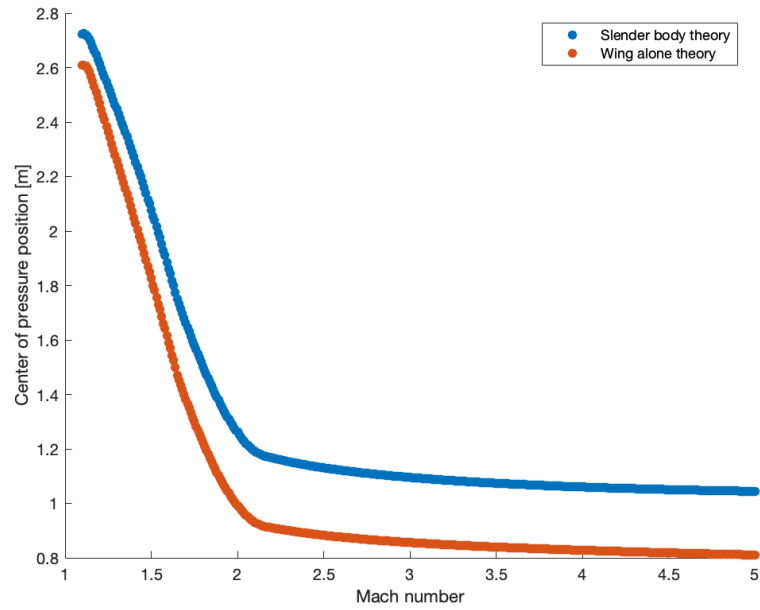


Figure 3.2: Results for $\alpha = 1^\circ$

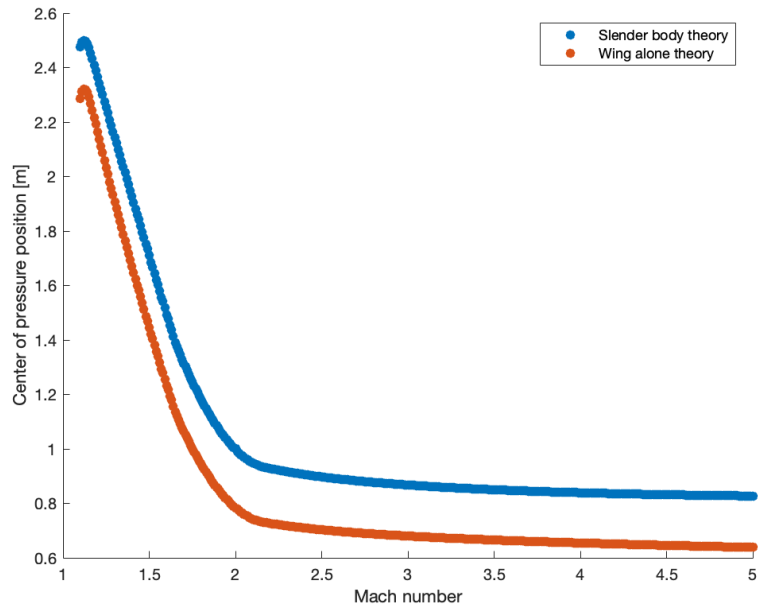


Figure 3.3: Results for $\alpha = 2^\circ$

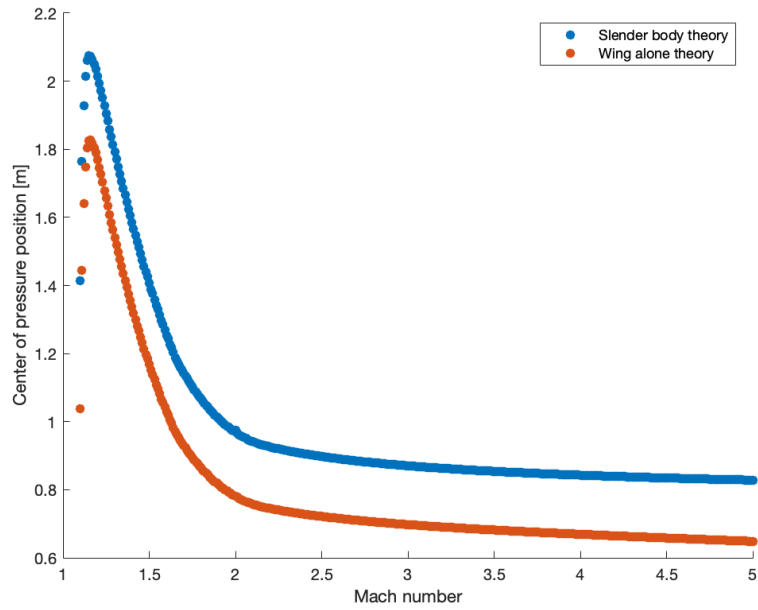


Figure 3.4: Results for $\alpha = 5^\circ$

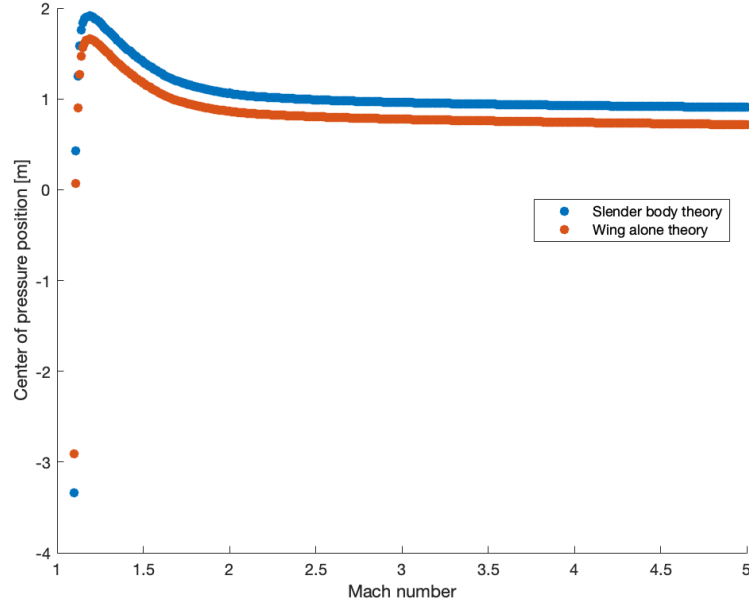


Figure 3.5: Results for $\alpha = 7^\circ$

produces more lift nearer to the tail due to the influence of the wings.

For both wing-alone and slender body theory, these plots also confirm what has been discussed in chapter 2 regarding the limits of Busemann's second order theory. Note that, in this context, "slender body theory" approach means calculating the body-wing interference factor using slender body theory, so equation 2.40; "wing-alone" approach means ignoring that coefficient and using equation 2.49 for the overall center of pressure position. While increasing the angle of attack even slightly, results at a Mach number close to $M = 1$ seem to become unstable. With $\alpha = 7^\circ$ some negative center of pressure positions are shown, and this clearly make no sense, confirming the fact that Busemann's theory can not be applied in these conditions.

Figure 3.6 shows the output of the script with a fixed Mach number, equal to the maximum Mach number of the flight, and a variable angle of attack. As α approaches 0° , the gradient of the curve increases in magnitude. Consequently, the original definition of the center of pressure at zero angle of attack, assumed to be the limit of this curve as α approaches zero, becomes difficult to apply. According to Hilton [5], the pressure distribution on the fins calculated using Busemann's coefficients should not be significantly affected by small changes in angle of attack. The high variability in the center of pressure position is likely due to the remainder of the algorithm, specifically the NACA-TR-1307 [1] method. This is reasonable,

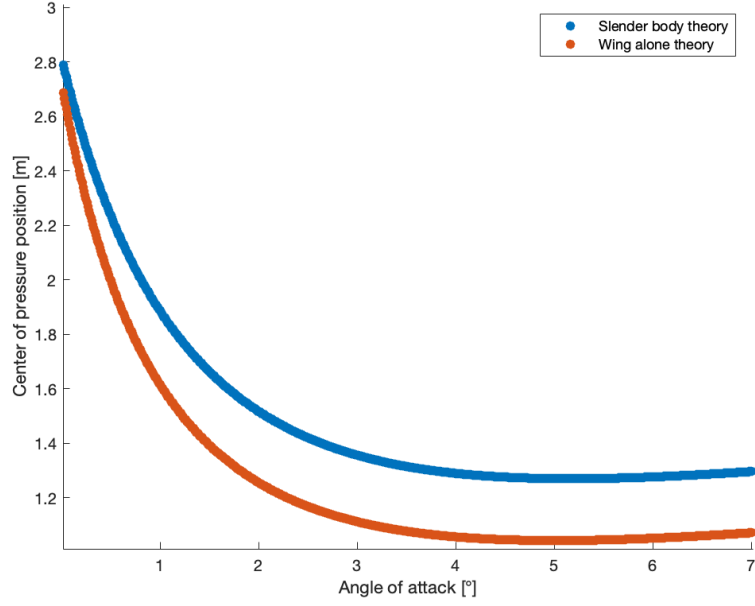


Figure 3.6: Results for $M = 1.59$ with variable angle of attack

considering that the entire method was developed for $\alpha \neq 0^\circ$. Therefore, simulations with variable Mach number have not been conducted for $\alpha < 0.5^\circ$, the approximate value at which the curve becomes overly sensitive to angle of attack perturbations.

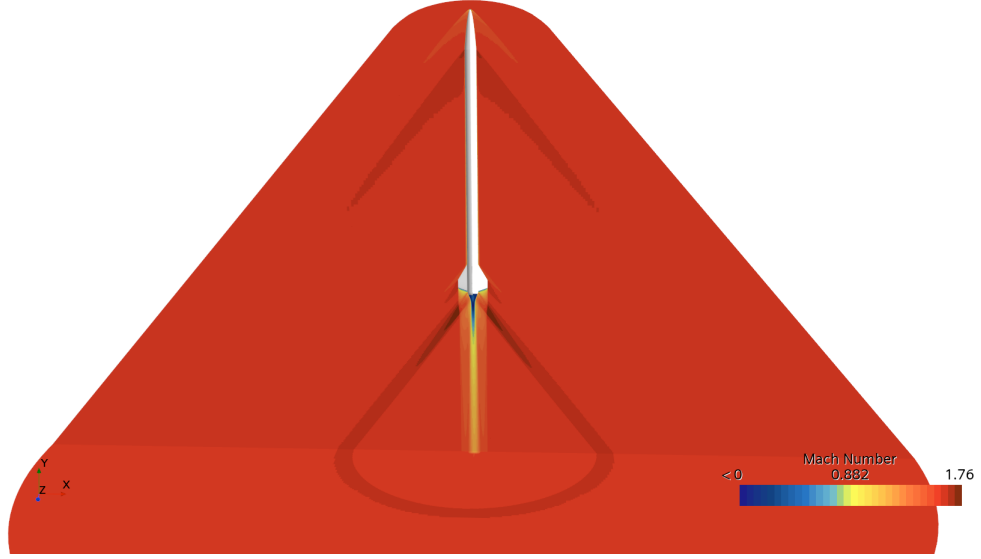
3.2 CFD results and comparison

A CFD analysis was carried out by the thesis supervisor as a reference for the present analysis. The freestream conditions are similar to those reported in Table 3.1, namely: $T = 276.6K$, $p = 81795Pa$, $\rho = 1.03kg/m^3$, $Mach = 1.589$. The Mach number is the highest in our preliminary trajectory simulation, and it was reached at an altitude of $1770m$. The thermodynamic conditions are those pertaining to that altitude according to the International Standard Atmosphere. Since the analysis in this work, as stated in the previous section, has been conducted for a non-zero angle of attack, a direct comparison of the results is not possible. However, considering the limitations of the method, it is possible to infer a rough estimation of the error that has been committed.

The CFD analysis has shown a center of pressure position $X_{CP} = 1.92m$. Results are reported in table 3.2. Results show that, for this specific application on the VES rocket, the wing-alone approach works better than the slender-body approach. The error for both methods is in line with the 10% margin declared by

Table 3.2: Results comparison

Method	$\alpha[^\circ]$	$X_{CP}[m]$	Error [%]	Error [cal]
CFD analysis	0	1.92	0	0
Slender body theory	0.5	2.24	16.7	2.39
wing-alone theory	0.5	2.00	4.2	0.60

**Figure 3.7:** CFD result for the Mach number

Barrowman in the *FIN* report. The error using slender-body theory is probably acceptable in the preliminary design phase of a rocket. However, this discrepancy in results implies that at least one CFD simulation will always be needed to select the appropriate approach. It is interesting to express the errors in calibers (remembering $1 \text{ cal} = 2 \times r_N$, or one rocket diameter), since the caliber is the typical unit of measure to discuss static margin and stability during competitions. Considering the discussion in section 1.4, both methods have errors that may significantly impact the static margin. Therefore, neither approach is appropriate for final, safety-related simulations. More concerningly, both approaches tend to place the center of pressure too far aft compared to the real position. This would result in a static margin that is higher than the actual one, rendering the results non-conservative.

The CFD simulations have also demonstrated why reduced-order methods can never be completely accurate. A shock wave is present at the nose tip, as expected, and an expansion fan, which was supposed to be generated at the nose-body intersection,

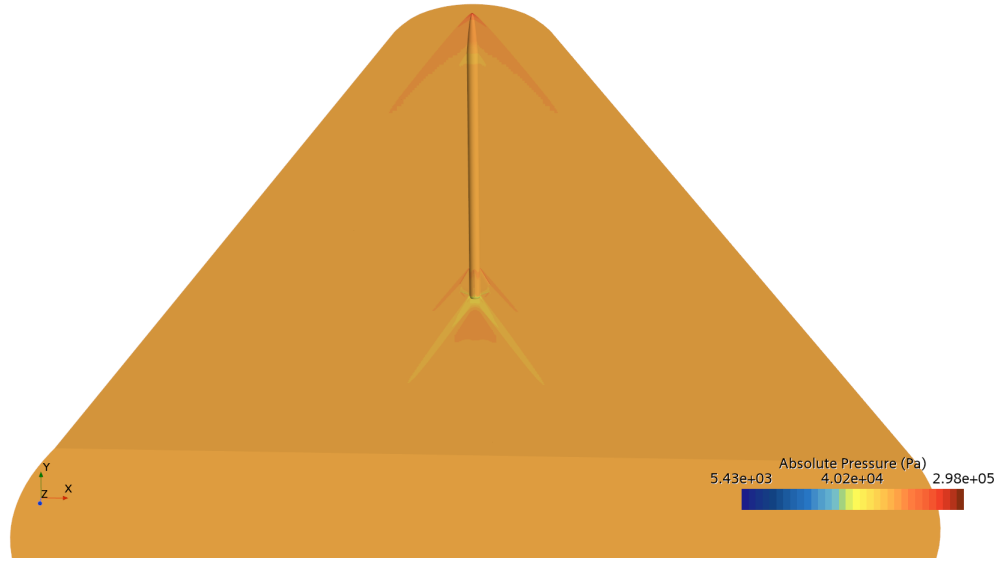


Figure 3.8: CFD result for the flow pressure

is indeed present. However, the flow around the finset and behind the boattail is extremely complex, as shown in 3.7 and 3.8 and will probably never be captured by a generic, reduced-order method. A shock wave is generated at the fuselage/fin intersection, probably due to the fact that the fin leading edge is slightly rounded and to a shock wave/boundary layer interaction. An expansion wave is generated at the fin tip. Behind the boattail, both expansion fans and reattachment shocks occur.

3.3 Conclusions, possible future improvements

The method has demonstrated various limitations, particularly at low Mach numbers and very low angles of attack. Improvements are necessary to implement it in the flight simulator for precise results. Currently, all safety-related simulations cannot rely solely on this method. Computational Fluid Dynamics (CFD) remains the preferred approach for calculating the center of pressure in these applications. Despite its limitations, the program remains useful, particularly for preliminary design, as it significantly reduces computation time. A CFD analysis for a single flight condition requires hours of computation and an adequate computer. Additionally, changes in fin geometry necessitate modifications in the CAD design. Fins optimization, therefore, can be time-consuming, which is not compatible with the timeline of a typical university rocketry project. The method presented in this

thesis, despite its limitations, offers the significant advantage of running in just a fraction of a second on an average computer.

This algorithm could be implemented in a future optimization program, which is in the plans of the Polito Rocket Team. Due to its low computational cost, it may be used to calculate the target function in a standard optimization algorithm or in a physics-based neural network. One possible technique could be a simple *random forest* algorithm, starting from the initial design point and then propagating randomly in the space of variables (such as surface inclination, span, sweepback angles) until a geometry that meets the stability requirements is found.

Appendix A

Supersonic cp calculator

```
1 close all
2 clear all
3 Machs = 1.1:0.01:5;
4
5 l_c = zeros(1, length(Machs));
6 l_c_wa = zeros(1, length(Machs));
7 counter = 1;
8 for i = Machs
9
10 r_N = 0.134./2; %rocket radius
11
12 %Freestream parameters
13
14 M = 1*i;
15 beta = sqrt(M^2-1);
16 rho = 1.03;
17 gamma = 1.4; %ratio of specific heats
18 alfa = 0.5*(pi/180); %angle of attack
19 mu = asin(1./M);
20 T_inf = 282.6;
21 R = 287;
22 q_inf = 0.5.*rho.*(M.^2).*gamma.*R*T_inf.*0.000145; %in psi
23
24 %Fin geometry (DIAMOND SHAPED, no cant angle)
25 GAMMA_L = pi/3; %leading edge sweepback angle
26 GAMMA_1 = pi/3; %first region boundary sweepback angle
27 GAMMA_2 = -70*(pi/180); %second region boundary sweepback angle
28 GAMMA_T = -70*(pi/180); %trailing edge sweepback angle
29 m = 1./tan(GAMMA_L); %%%%
30
31 zeta_L = 2.56*(pi/180); %semi-opening angle of the leading edge
```

```

32 zeta_T = 2.56*(pi/180); %semi-opening angle of the trailing edge
33
34 s = 0.114; %wing-alone semispan
35 c_r = 0.3377; % rootchord
36 c_t = c_r+s*tan(GAMMA_T)-s*tan(GAMMA_L); %tipchord
37 l_L_r = 0.09587; %length of root airfoil leading edge region chord
38 l_T_r = 0.09073; %length of root airfoil trailing edge region chord
39 X_L = 0; %distance between reference axis and intersection of root
    and leading edge
40 A=(c_r+c_t).*(s./2);
41 lambda = c_t./c_r; %%%
42 %running fin to gather fin parameters
43 run("FIN3.m")
44
45
46 %NOSE CENTER OF PRESSURE
47
48 %Method 1 (NACA 1307)
49
50 r_N = 0.134/2; %nose radius at shoulder
51 l_s = 0.45; %shoulder length
52 % C_L_alfa_W = 3.4; %to be obtained using FIN
53 % L_W = 1.3351; %to be determined using FIN
54 % x_cp_w = 0.0238; %to be determined using FIN
55 % run("FIN.m")
56 %S_W = (c_r+c_t).*(s./2); %wing-alone surface. Not clear which, this
    is the planform area on one side. Shouldn't we use the same
    reference area? |-/
57 S_W = A;
58 K_N = (2.*pi.*(r_N.^2))./(S_W.*C_L_alfa_W);
59 V_S = (pi/2).*(r_N.^2).*l_s; %volume of the nose up to the shoulder
60 l_N = l_s.*(1-(V_S./((pi.*(r_N.^2).*l_s))));
61 l_w = 3.03; %position of the leading edge intersection with root
    chord, relative to reference point (nose tip)
62 C_L_N = K_N.*C_L_alfa_W .* alfa;
63 s_t = s + r_N;
64
65 K_W_B = (2/pi) * (((1 + r_N.^4 ./ s_t.^4) .* (0.5 * atan(0.5 * (s_t
    ./ r_N - r_N ./ s_t)) + pi/4) - (r_N.^2 ./ s_t.^2) .* ((s_t ./ r_N
    - r_N ./ s_t) + 2 * atan(r_N ./ s_t))) ./ (1 - r_N ./ s_t).^2);
66 %K_W_B = ((0.5 * (1 + lambda) - lambda * r_N ./ s - (r_N.^2 * (1 -
    lambda)) ./ (s.^2 - r_N.^2) .* log(s ./ r_N)) ./ (0.5 * ((s - r_N)
    ./ (s + r_N)) * (1 + lambda)));
67
68
69 %BODY CENTER, OF PRESSURE DUE TO TAIL INFLUENCE (Supersonic only)
70 % lines not needed if always supersonic
71 % if beta.*A.*(1+lambda)*(1./(m.*beta)+1)<4

```



```

72 %      K_B_W = ((1 - (r_N.^2)/(s_t.^2)).^2 - (2./pi) .* ((1 + (r_N
      .^4)/(s_t.^4)) .* (0.5 .* atan(0.5 .* (s_t./r_N - r_N./s_t)) + pi
      /4) - (r_N.^2 ./ s_t.^2) .* ((s_t./r_N - r_N./s_t) + 2 * atan(r_N
      ./s_t)))) ./ ((1 - r_N./s_t).^2);
73 %
74 %      %slender theory, eqz 21 NACA 1307
75 % else
76 %
77 %      K_B_W = ((8*beta.*m)/(pi.*(1+lambda).*((2.*beta.*r_N)./c_r)
      .*((s_t./r_N)-1).*(beta.*C_L_alfa_W).*sqrt(beta.^2.*m.^2-1))).*(((
      beta.*m)/(1+beta.*m)).*(((beta.*m+1).*(2.*r_N.*beta./c_r)+beta.*m)
      )./(beta.*m)).^2.*acos(((beta.*m+1).*(2.*r_N.*beta./c_r)+1)./(
      beta.*m+1).*(2.*r_N.*beta./c_r)+beta.*m)))+(sqrt(beta.^2.*m.^2-1)
      )./(beta.*m)).*(sqrt(1+4.*r_N.*beta./c_r)-1)-(sqrt(beta.^2.*m.^2-1)
      )./(beta.*m)).*(beta.*2.*r_N./c_r).^2.*acosh(1+c_r./(beta.*2.*r_N))
      -((beta.*m)/(1+beta.*m)).*acos(1./(beta.*m)));
78 %      %independent method, eqz 24 NACA 1307
79 % end
80 %
81
82 %slender theory, eqz 21 NACA 1307
83 K_B_W = ((1 - (r_N.^2)/(s_t.^2)).^2 - (2./pi) .* ((1 + (r_N.^4)/(
      s_t.^4)) .* (0.5 .* atan(0.5 .* (s_t./r_N - r_N./s_t)) + pi/4) - (
      r_N.^2 ./ s_t.^2) .* ((s_t./r_N - r_N./s_t) + 2 * atan(r_N./s_t)))
      ) ./ ((1 - r_N./s_t).^2);
84
85
86 %K_B_W = ((8*beta.*m)/(pi.*(1+lambda).*((2.*beta.*r_N)./c_r).*((s_t
      ./r_N)-1).*(beta.*C_L_alfa_W).*sqrt(beta.^2.*m.^2-1))).*(((beta.*m)
      )./(1+beta.*m)).*(((beta.*m+1).*(2.*r_N.*beta./c_r)+beta.*m)./(
      beta.*m)).^2.*acos(((beta.*m+1).*(2.*r_N.*beta./c_r)+1)./(
      beta.*m+1).*(2.*r_N.*beta./c_r)+beta.*m)))+(sqrt(beta.^2.*m.^2-1)./(beta.*
      m)).*(sqrt(1+4.*r_N.*beta./c_r)-1)-(sqrt(beta.^2.*m.^2-1)./(beta.*
      m)).*(beta.*2.*r_N./c_r).^2.*acosh(1+c_r./(beta.*2.*r_N))-((beta.*
      m)/(1+beta.*m)).*acos(1./(beta.*m)));
87 %      %independent method, eqz 24 NACA 1307
88
89
90 %% Equation components eqz 30 NACA 1307
91 %% Compute sqrt term
92 % sqrt_term = sqrt(beta^2 * m^2 - 1);
93 %
94 %% Compute acos and asin terms
95 % cos_inv1 = acos((m * beta + c_t / (beta * (2 * r_N))) / (1 + m *
      c_t / (2 * r_N)));
96 % cos_inv2 = acos(1 / (m * beta));
97 % sin_inv = asin(beta * (2 * r_N) / c_t);
98 % cosh_inv = acosh(c_t / (beta * (2 * r_N)));
99 %

```

```

100 % % Compute the main term
101 % term1 = (1 + m * c_t / (2 * r_N))^2 * cos_inv1;
102 % term2 = m^2 * beta^2 * (c_t / (beta * (2 * r_N)))^2 * cos_inv2;
103 % term3 = m * beta * (c_t / (beta * (2 * r_N)))^2 * sqrt_term *
    sin_inv;
104 % term4 = sqrt_term * cosh_inv;
105 %
106 % % Final K_B(W) calculation
107 % K_B_W = ((8 / (pi * sqrt_term)) * (beta * (2 * r_N) / c_r) * (term1
    - term2 + term3 - term4)) ./ (beta * C_L_alfa_W * (lambda + 1) * (s
    / r_N - 1));
108
109
110
111 M_B_W = ((4 * q_inf * alfa * m) ./ (3 * pi * beta)) * (c_r).^3 * (sqrt(1 + (2 *
    beta * 2 * r_N)) * (((2 * m * beta + 5) ./ (3 * (m * beta + 1).^2)) + ((2 * r_N *
    beta ./ c_r) ./ (3 * (m * beta + 1))) - ((2 * r_N * beta ./ c_r).^2 ./ (beta * m)))
    + (1 ./ sqrt(m.^2 * beta.^2 - 1)) * ((1 + 2 * r_N * beta ./ c_r).^3 - (2 * r_N *
    beta ./ c_r).^3 ./ (m.^2 * beta.^2) - 1 ./ (1 + m * beta).^2) * acos((1 + (2 * r_N
    * beta ./ c_r) * (m * beta + 1)) ./ (m * beta + (2 * r_N * beta ./ c_r) * (m * beta
    + 1))) + (beta * 2 * r_N ./ c_r).^3 * (1 ./ (m.^2 * beta.^2)) * acosh(1 + c_r
    ./ (2 * r_N * beta)) - ((2 * m * beta + 5) ./ (3 * (m * beta + 1).^2)) - ((1 - (1 ./ (m
    * beta + 1)).^2) ./ sqrt(m.^2 * beta.^2 - 1)) * acos(1 ./ (m * beta))) );
112
113 %eqx 67 NACA 1307
114
115 d = 2 * r_N;
116
117 x_rel_c_B_W = M_B_W ./ (K_B_W * L_W * c_r);
118 l_B_W = l_w + c_r * x_rel_c_B_W;
119 C_L_W_B = C_L_alfa_W * alfa * K_W_B;
120 C_L_B_W = K_B_W * C_L_alfa_W * alfa;
121
122 %center of pressure on tail due to body considered to be the same as
    wing
123 %only
124
125 X_CP = (l_N * C_L_N + (x_cp_w + l_w) * C_L_W_B + l_B_W * C_L_B_W) ./ (C_L_N +
    C_L_W_B + C_L_B_W)
126 X_CP_wa = (l_N * C_L_N + (x_cp_w + l_w) * C_L_W_B) ./ (C_L_N + C_L_W_B)
127 l_c(1, counter) = X_CP;
128 l_c_wa(1, counter) = X_CP_wa;
129 counter = counter + 1;
130
131
132 end
133 hold on
134 scatter(Machs, l_c, 'filled', 'DisplayName', 'Slender body theory')
135 scatter(Machs, l_c_wa, 'filled', 'DisplayName', 'Wing alone theory')

```

```
136 xlabel('Mach number')
137 ylabel('Center of pressure position [m]')
138 legend('show')
139 legend('Location','best')
```

Appendix B

Fin 2

```
1
2 % THIS SCRIPT IS USED TO CALCULATE THE CENTER OF PRESSURE OF THE TAIL
3 % DURING SUPERSONIC FLIGHT
4
5 close all
6 disp("running FIN")
7
8
9 %Busemann's coefficients calculated as in the original report
10 msq = M.^2;
11 dem = -2.*msq+4/3;
12 den = 1./(beta.^7);
13 k_1 = 2./beta;
14 k_2 = 0.5 * ((M.^2 - 2).^2 + gamma * M.^4) .* (M.^2 - 1).^-2;
15 k_3 = ((gamma + 1) .* M.^8 + (2 * gamma.^2 - 7 * gamma - 5) .* M.^6 +
16       10 * (gamma + 1) .* M.^4 - 12 .* M.^2 + 8) ./ (6 .* beta.^7);
17 B_3 = (0.36 .* M.^8 - 1.493 .* M.^6 + 3.6 .* M.^4 - 2 .* M.^2 + 1.33)
18       ./ beta.^7;
19 %third order busemann coefficient: not zero if there is a single
20 %compressive shock before the fin
21
22 cmu=tan(GAMMA_L)+beta;
23 mu = s.*cmu;
24 %local surface inclination angles
25 eta_1 = zeta_L-alfa;
26 eta_2 = -alfa;
27 eta_3 = -zeta_T-alfa;
28 eta_4 = zeta_L + alfa;
29 eta_5 = alfa;
```

```

28 eta_6 = -zeta_T + alfa;
29 eta_i = [eta_1, eta_2, eta_3, eta_4, eta_5, eta_6];
30 cp_i = k_1.*eta_i + k_2.*(eta_i.^2) + k_3.*(eta_i.^3);
31 cp_i(1,4) = cp_i(1,4)-B_3.*eta_4.^3;
32 cp_i(1,5) = cp_i(1,5)-B_3.*eta_4.^3;
33 cp_i(1,6) = cp_i(1,6)-B_3.*eta_4.^3;
34 if eta_1 > 0
35     cp_i(1,1)=cp_i(1,1)-B_3.*eta_1.^3;
36     cp_i(1,2)=cp_i(1,2)-B_3.*eta_1.^3;
37     cp_i(1,3)=cp_i(1,3)-B_3.*eta_1.^3;
38 end
39
40
41 %number of strips
42 n_strips = 300;
43 %strips width
44 delta_y = s./n_strips;
45
46 %local lift, drag and pitching moment vector initialization
47 F_l_j = ones(1,n_strips+1);
48 F_d_j = ones(1,n_strips+1);
49 M_p_j = ones(1,n_strips+1);
50 M_r_j = ones(1,n_strips+1);
51
52 for j = 1:1:(n_strips+1)
53     y = (j-1)*delta_y;
54     if y < 0.07348
55         %local surface region lengths
56         l_L = y.*(tan(GAMMA_1)-tan(GAMMA_L))+l_L_r;
57         l_T = y.*(tan(GAMMA_T)-tan(GAMMA_2))+l_T_r;
58         C = c_r + y.*(tan(GAMMA_T)-tan(GAMMA_L));
59         l_M = C - l_L - l_T;
60         l_1 = l_L./(cos(zeta_L));
61         l_4 = l_1;
62         l_3 = l_T./(cos(zeta_T));
63         l_6 = l_3;
64         l_2 = C - l_T - l_L;
65         l_5 = l_2;
66         l_i = [l_1, l_2, l_3, l_4, l_5, l_6];
67         d_i = l_i.*(cos(eta_i));
68         n_i = l_i.*(sin(eta_i));
69         %reference distances (reference axis x=0 —> X_L = 0)
70         x_p_1 = X_L;
71         x_p_2 = d_i(1,1)+X_L;
72         x_p_3 = d_i(1,1)+d_i(1,2)+X_L;
73         x_p_4 = X_L;
74         x_p_5 = d_i(1,4)+X_L;
75         x_p_6 = d_i(1,5)+X_L+d_i(1,4);
76         x_p_i = [x_p_1,x_p_2,x_p_3,x_p_4,x_p_5,x_p_6];

```

```

77 %intersection between surface and Mach cone
78 l_w = (s-y).*(GAMMA_L+beta);
79 r_i = [0,0,0,0,0,0];
80 if (C > l_w) && (l_w >= C - l_T)
81     r_i = [0,0,(C-l_w)./l_T,0,0,(C-l_w)./l_T];
82 elseif (C-l_T > l_w) && (l_w >= l_L)
83     r_i = [0,(C-l_T-l_w)./l_M,1,0,(C-l_T-l_w)./l_M,1];
84 elseif (C - l_T - l_M > l_w) && (l_w >= 0)
85     r_i = [(l_L-l_w)./l_L,1,1,(l_L-l_w)./l_L,1,1];
86 end
87 %local coefficients on the strip
88 F_l_i = cp_i.*d_i.*(1-r_i./2);
89 F_d_i = cp_i.*n_i.*(1-r_i./2);
90 x_i = 0.5.*((d_i.*(1-r_i+r_i./2+(x_p_i./l_i).*(2-r_i)))./(1-
r_i./2));
91 M_p_i = F_l_i.*x_i;
92 else
93 %local surface region lengths
94 l_L = ((s-y).*(tan(pi/3)+tan(pi/9))+c_t)./2;
95 l_T = l_L;
96 C = l_L+l_T;
97 l_M = 0;
98 l_1 = l_L./(cos(zeta_L));
99 l_4 = l_1;
100 l_3 = l_T./(cos(zeta_T));
101 l_6 = l_3;
102 l_2 = 0;
103 l_5 = 0;
104 l_i = [l_1,l_2,l_3,l_4,l_5,l_6];
105 d_i = l_i.*(cos(eta_i));
106 n_i = l_i.*(sin(eta_i));
107 %reference distances (reference axis x=0 —> X_L = 0)
108 x_p_1 = X_L;
109 x_p_2 = d_i(1,1)+X_L;
110 x_p_3 = d_i(1,1)+d_i(1,2)+X_L;
111 x_p_4 = X_L;
112 x_p_5 = d_i(1,4)+X_L;
113 x_p_6 = d_i(1,5)+X_L+d_i(1,4);
114 x_p_i = [x_p_1,x_p_2,x_p_3,x_p_4,x_p_5,x_p_6];
115 %intersection between surface and Mach cone
116 l_w = (s-y).*(GAMMA_L+beta);
117 r_i = [0,0,0,0,0,0];
118 if (C > l_w) && (l_w >= C - l_T)
119     r_i = [0,0,(C-l_w)./l_T,0,0,(C-l_w)./l_T];
120 elseif (C - l_T > l_w) && (l_w >= 0)
121     r_i = [(l_L-l_w)./l_L,1,1,(l_L-l_w)./l_L,1,1];
122 end
123 %local coefficients on the strip
124 F_l_i = cp_i.*d_i.*(1-r_i./2);

```

```

125     F_d_i = cp_i.*n_i.*(1-r_i./2);
126     x_i = 0.5.*((d_i.*(1-r_i+r_i./2+(x_p_i./l_i).*(2-r_i)))./(1-
r_i./2));
127     x_1 = 0.5.*((d_i(1,1).*(1-r_i(1,1)+r_i(1,1)./2+(x_p_i(1,1)./
l_i(1,1)).*(2-r_i(1,1)))./(1-r_i(1,1)./2));
128     x_2 = 0;
129     x_3 = 0.5.*((d_i(1,3).*(1-r_i(1,3)+r_i(1,3)./2+(x_p_i(1,3)./
l_i(1,3)).*(2-r_i(1,3)))./(1-r_i(1,3)./2));
130     x_4 = 0.5.*((d_i(1,4).*(1-r_i(1,4)+r_i(1,4)./2+(x_p_i(1,4)./
l_i(1,4)).*(2-r_i(1,4)))./(1-r_i(1,4)./2));
131     x_5 = 0;
132     x_6 = 0.5.*((d_i(1,6).*(1-r_i(1,6)+r_i(1,6)./2+(x_p_i(1,6)./
l_i(1,6)).*(2-r_i(1,6)))./(1-r_i(1,6)./2));
133     x_i = [x_1,x_2,x_3,x_4,x_5,x_6];
134     M_p_i = F_l_i.*x_i;
135     end
136
137
138
139     %adding up...
140     F_l_j(1,j) = sum(F_l_i);
141     F_d_j(1,j) = sum(F_d_i);
142     M_p_j(1,j) = sum(M_p_i);
143     M_r_j(1,j) = y.* F_l_j(1,j);
144     end
145
146     %total characteristics
147     F_l = sum(F_l_j);
148     F_d = sum(F_d_j);
149     M_p = sum(M_p_j);
150     M_r = sum(M_r_j);
151
152     X_CP = M_p./(F_l.*cos(alfa))
153     Y_CP = M_r./F_l
154
155     C_L = (2.*F_l.*delta_y)./A
156     C_L_alfa_W = C_L ./ alfa
157     L_W = F_l
158     x_cp_w = X_CP
159
160
161
162
163
164     disp("FIN run completed")

```

Appendix C

Nosecone approximation function

```
1
2      %Nose cone function and tangent approximation
3  clc
4  close all
5  clear all
6
7  L = 0.52; %nose length
8  R = 0.05 ; %nose base radius
9  C= 0; %haack series parameter, 0 for Von Karman
10 n = 1/2; %power series parameter
11 x = linspace(0,L,1000);
12
13 %nosecone shape function
14 %von karman
15 %y = @(x) (R/sqrt(pi))*sqrt(acos(1-(2.*x)/L) - ...
16 (sin(2.*acos(1-(2.*x)/L)))/2+C*(sin(acos(1-(2.*x)/L)))^3);
17 %power series
18 y = @(x) R*(x./L).^n;
19 ys = y(x);
20
21 %slope of the tangent lines
22 %von karman
23 %dy_by_dx = @(x) -(R.*((2.*cos(2.*acos(1 - (2.*x)./L)))) ./ ...
24 (L.*(1 - ((2.*x)./L - 1).^2).^^(1/2)) - 2./(L.*...
25 (1 - ((2.*x)./L - 1).^2).^^(1/2)) + (6.*C.*(1 - ...
26 ((2.*x)./L - 1).^2).^^(1/2).*((2.*x)./L - 1))./L))./...
27 (2.*sqrt(pi)).*(acos(1 - (2.*x)./L) - sin(2.*...
28 acos(1 - (2.*x)./L))/2 + C.*(1 - ((2.*x)./L - 1).^2).^^(3/2)).^(1/2));
```

```

29 %power series
30 dy_by_dx = @(x) n*(R/L^n).*x.^(n-1);
31 y_f = dy_by_dx(x);
32
33 plot(x,ys,"k","LineWidth",1)
34 hold on
35 plot(x,-ys,"k","LineWidth",1)
36 axis([0 L -0.1 0.1])
37
38 S = 15; % Desidered Segments
39 N = S + 1; % Number of Nodes
40 delta = (max(y_f)-min(y_f))/(N); %gradient delta
41 x_t = zeros(N,1); %x position of the tangent nodes
42 y_t = zeros(N,1); %y position of the tangent nodes
43 x_t(1) = 0; %the tip is set to be a tangent node
44 y_t(1) = 0;
45 k = 2;%number of tangent nodes already found (tip+base)
46
47 for i = 1:length(y_f)
48
49     if y_f(i) <= max(y_f) - delta && k == 2
50         x_t(k) = x(i);
51         k = k + 1;
52
53     elseif (y_f(i) <= (dy_by_dx(x_t(k - 1)) - delta)) && (k > 2)
54         x_t(k) = x(i);
55         k = k + 1;
56     end
57 end
58
59 x_t(N) = L;%the base is set to be a tangent node
60 y_t(N) = R;
61 y_t = y(x_t);
62 %to solve numerical cancellation problem that might leave some zeros
   in the
63 %x_t array:
64 k = 2;
65 while k<N
66     if x_t(k) == 0 %if a 0 is found in the array (in a position
   different ...
67     %from 1, because k starts at 2)
68         ns = N-k;%number of nodes to correct
69         d = (x_t(N)-x_t(k-1))/(ns+1);%the 0 nodes are substituted by
   ...
70         %equispaced nodes of length d
71         for j = 1:ns
72             x_t(k+j-1)=x_t(k-1)+d*j;
73             y_t(k+j-1) = y(x_t(k+j-1));
74         end

```

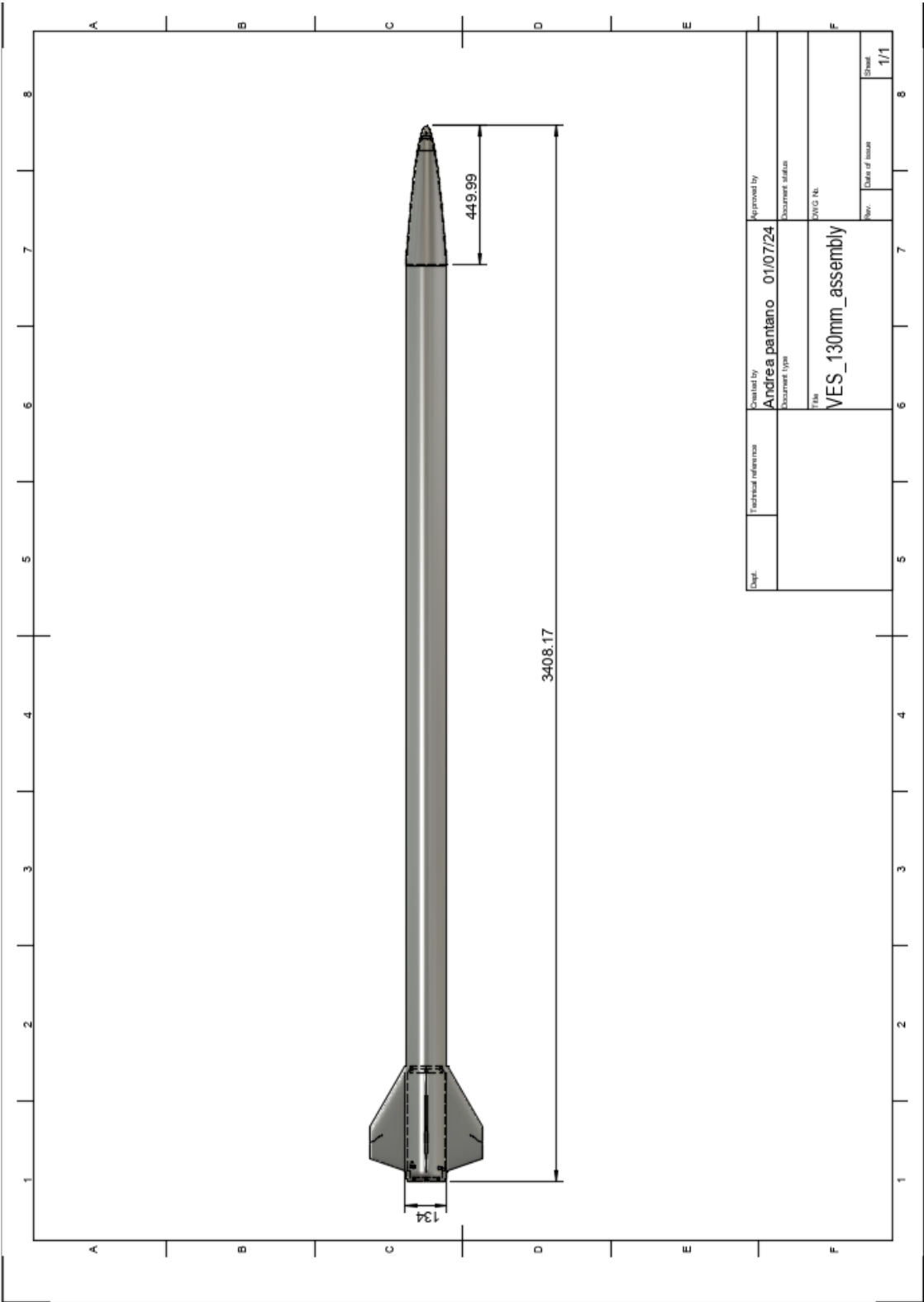
```

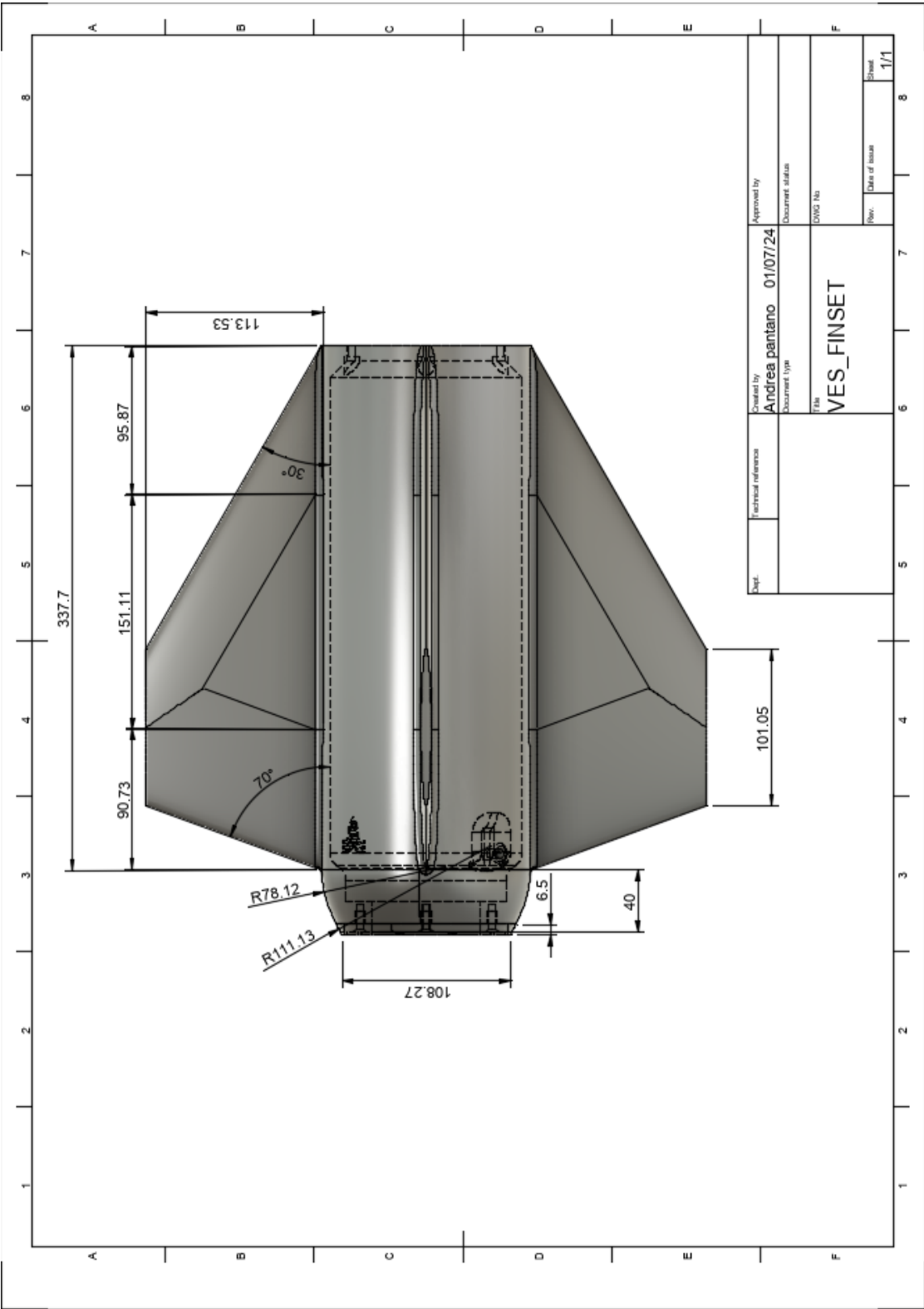
75         k = N+1;%to break the loop
76     end
77     k = k+1;
78 end
79 %Method 1: join the tangent nodes. Tangent nodes coincident to the
80 %final nodes. Red segments
81 plot(x_t,y_t,"bo", "LineWidth",2)
82 plot(x_t,-y_t, "bo", "LineWidth",2)
83 for i = 1:S
84     x1 = x_t(i);
85     x2 = x_t(i + 1);
86     plot([x1;x2], [y(x1);y(x2)], 'r', 'linewidth',2)
87     plot([x1;x2], [-y(x1);-y(x2)], 'r', 'linewidth',2)
88 end
89
90 %Method 2: find the final nodes by intersecting the tangents to the
91 %already found tangent nodes. Green segments.
92 x_n = zeros(N+1,1); %x position of the final nodes
93 y_n = zeros(N+1,1); %y position of the final nodes
94 for j = 1:(length(x_t)-1)
95     x1 = x_t(j)
96     x2 = x_t(j+1)
97     y1 = y_t(j)
98     y2 = y_t(j+1)
99     m1 = n*(R/L^n)*x1^(n-1)%slopes of the tangents in the tangent
nodes
100     m2 = dy_by_dx(x2)
101     x = (y2-y1+m1*x1-m2*x2)/(m1-m2);
102     y = m1*(x-x1)+y1;
103     x_n(j+1)=x;
104     y_n(j+1)=y;
105
106 end
107 x_n(length(x_t)+1) = L;
108 y_n(length(y_t)+1) = R;
109
110 for k=1:length(x_n)-1
111     plot([x_n(k);x_n(k+1)], [y_n(k);y_n(k+1)], 'g', 'linewidth',2)
112     plot([x_n(k);x_n(k+1)], [-y_n(k);-y_n(k+1)], 'g', 'linewidth',2)
113 end

```

Appendix D

Technical drawings





Bibliography

- [1] Pitts, W.C., Nielsen, J.N., and Kaatari, G.E. *Lift and Center of Pressure of Wing-Body-Tail Combinations at Subsonic, Transonic and Supersonic Speeds*. Tech. rep. NACA-TR-1307, 1959.
- [2] Barrowman, J.S. *The Practical Calculation of the Aerodynamic Characteristics of Slender Finned Vehicles*. Master Thesis, The Catholic University of America, Washington D.C., 1967.
- [3] Barrowman, J.S. *FIN - a computer program for calculating the aerodynamic characteristics of fins at supersonic speeds*. NASA-TM-X-55523, NASA Goddard Space Flight Center, 1966.
- [4] Busemann, A. *Aerodynamic Lift at Supersonic Speeds*. Lecture Given at the 5th Volta Conference in Rome, 1937.
- [5] Hilton, W.F. *High speed aerodynamics*. Green: [1st ed.] ed. Longmans, 1951.
- [6] Allen, H.J. and Perkins, E.W. *A Study of Effects of Viscosity on Flow Over Slender Inclined Bodies of Revolution*. Tech. rep. NACA-TR-1048, 1951.
- [7] Anderson, J.D. *Hypersonic and High-temperature Gas Dynamics*. AIAA education series. American Institute of Aeronautics and Astronautics, 2006. ISBN: 9781563477805.
- [8] Lagerstrom, P.A. and Van Dyke, M.D. *General Considerations About Planar and Non-Planar Lifting Systems*. Tech. rep. SM-13432, Douglas Aircraft Co. Inc., Santa Monica, 1949.