POLITECNICO DI TORINO

Bachelor's Degree in Aerospace Engineering

Bachelor's Degree Thesis

Numerical Computation of the Wave Drag Coefficient for a Slender Wing-Body Combination

Supervisors

Prof. Domenic D'AMBROSIO

Ing. Federica PORTIS

Candidate

Alessandro RABINO

July 2024

Abstract

The purpose of this document is to outline the work carried out in order to calculate numerically the aerodynamic wave drag coefficient for a certain wing-body configuration with known geometry. The object chosen for this calculation is the currently under development *Vittorio Emanuele II*, also known as *VES*, a solid propellant supersonic rocket designed by the *PoliTo Rocket Team* student team in the *Politecnico di Torino*.

The ultimate purpose of this work is to provide a simple tool that allows to easily reconfigure the rocket's geometry and to iterate efficiently the calculations to aid the early stage design and prototyping of future rockets within the same team, similarly to semi-empirical formulas based softwares like DATCOM, as the use of CFD in this design stage is impractical. The supersonic area rule method was therefore chosen to be implemented with a script on MATLAB, with the final result being numerically computed.

Table of Contents

List of Tables

List of Figures

Notation

c	Chord
C_D	Wave drag coefficient
d	Diameter
D	Wave drag
$D\{S\}$	Wave drag associated with the generic 'area' distribution S
$\mathscr{D}$	Domain of integration
l	Total axial length
M	Free-stream Mach number
n	Power law series exponent
q	Kinetic pressure $\frac{1}{2}\rho_0 U_0^2$
R	Local radius of the body's cross-section
s	Span
$S_B(x)$	Body tube cross-sectional area
$S(x, \theta, M)$	Elemental area distribution
$S_W(x, \theta, M)$	Exposed fin elemental area distribution
t	Maximum thickness
$T(x, y)$	Thickness function for a wing lying on the $z = 0$ plane
U_0	Free-stream velocity
β	$\sqrt{M^2 - 1}$
Γ	Sweep angle
θ	Cylindrical polar coordinate
ρ_0	Free-stream fluid density
$_{1},\ _{2}$	Subscripts denoting integration variables, e.g., x_1, x_2
$'$	Denotes partial differentiation with respect to x, e.g., $S'(x, \theta, M) = \frac{\partial S(x,\theta,M)}{\partial x}$
$_{B},_{BT},_{F},_{L},_{NS},_{NW},_{r},_{T},_{W}$	Subscripts denoting body (or fuselage), boat tail, fin, leading edge, nosecone, net wing, root, trailing edge, exposed wing

Chapter 1

Introduction

1.1 Definition of the coefficient of drag and of its components

The *coefficient of drag* is a dimensionless quantity associated with the resistance, or drag, of an object when it moves in a fluid. The drag is a force that acts opposite to the relative motion of the object in the fluid. It is defined as follows:

$$C_{D_{tot}} = \frac{2D_{tot}}{\rho U^2 S} \tag{1.1}$$

where D_{tot} is the total resistant force acting on the object when it moves in a fluid, ρ is the density of the medium in which the object is traveling, U the stream speed of the fluid relative to the object, and S is a reference area chosen for the particular configuration of the problem.[1]
The coefficient of drag can be subdivided into different components like so:

$$C_{D_{tot}} = C_{D_f} + C_{D_p} \tag{1.2}$$

where C_{D_f} is that part of the coefficient of drag due to surface friction and C_{D_p} is the part of the coefficient of drag caused by the pressure field created around the object when it travels in a fluid. These quantities are usually referred to as *friction drag* and *pressure drag* coefficient respectively. They are defined in the same way as in equation 1.1, but with the total drag force substituted with the friction or pressure drag accordingly. [1][2]

Depending on the conditions of the problem, either one between the friction or pressure drag can constitute a larger percentage of the total drag than the other. In fluid dynamics the *Reynolds number Re* is a dimensionless quantity used to predict the general properties and behaviour of all flows in a condition of dynamic similarity. It is defined as follows:

$$Re = \frac{Ul}{\nu} \tag{1.3}$$

where U is the flow velocity, l is a reference length chosen for the configuration, and ν is the kinematic viscosity of the fluid. Re can be interpreted as the ratio between inertial and viscous forces acting in the system.[1]
In general, when the Reynolds number is low the friction drag is predominant in the total drag of an object, while when the Reynolds number is high the pressure drag is the most important term.[1]

1.2 Definition of the wave drag coefficient

The *Mach number* M is a dimensionless quantity which is defined as follows:

$$M = \frac{U}{c_s} \tag{1.4}$$

where U is the flow speed and c_s is the speed of sound in the fluid. When $M = 1$ the flow is *sonic*, meaning that its speed is equal to the speed of sound of the particular medium in which the flow is present.[2]

When an object reaches a high enough velocity so that the flow around it is accelerated locally to $M > 1$, therefore generating an expansion of the flow, during the recompression of the flow a shock wave is formed and some of the total pressure, sum of the staic and dynamic pressure, of the flow will be lost. This shock wave together with the loss of total pressure changes the pressure field around the object, increasing the pressure drag acting on it. This additional drag caused by the shock wave is called *wave drag*. A *wave drag coefficient* C_{D_w} can therefore be defined in the same way as in equation 1.1. The C_{D_w} is itself part of C_{D_p} as the wave drag is added to the total pressure drag acting on the object.
The object itself doesn't need to travel at $M = 1$ or higher with respect to the fluid in order to form shock waves, if its shape can accelerate the flow around it to $M > 1$. However, if the object travels at $M > 1$ with respect to the fluid, then a shock wave will always form at the contact point between the fluid and the object. When the object is shaped like a cone and it is sufficiently sharp the shock wave that is formed is called *Mach cone*.[2]

The calculation of the wave drag for a particular object traveling at speeds greater than $M = 1$ will be the subject of this work. In the following chapters, the wave drag coefficient will be simply identified as C_D and the force associated with it, i.e. the wave drag, will be known as simply D.

Chapter 2

Wing-body combination: *VES* rocket

2.1 Overview of the object

The *Vittorio Emanuele II* is the second solid propellant rocket designed by the *PoliTo Rocket Team*. Its primary objective is the participation in the *European Rocketry Challenge*, an annual event where European students' teams from different universities compete with their own developed rockets. The goal of the competition is to reach as precisely as possible a certain target apogee and to land the rocket safely intact after its flight. Each rocket belongs to a certain class, with each class having a different target apogee to reach.

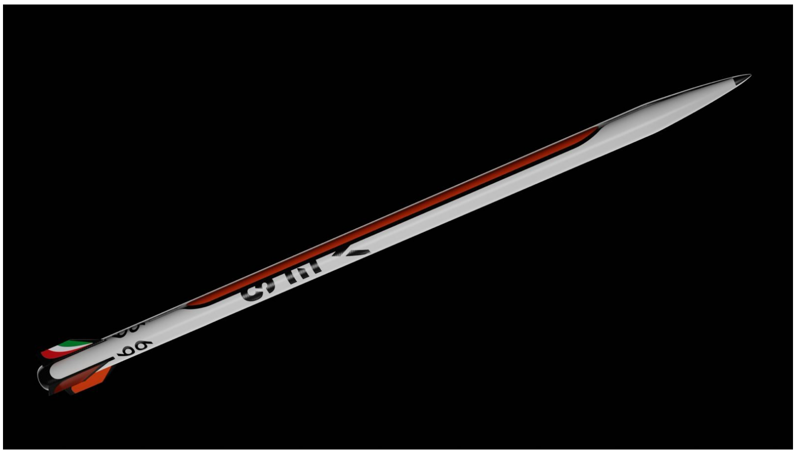

Figure 2.1: *Vittorio Emanuele II.*

VES is designed to compete in the 9 km class, with a target apogee of 9000m. To reach its designed altitude, it will be capable of reaching speeds of up to Mach 1.8. It will be equipped with an airbrakes system, an active control surface whose purpose is to decelerate during the ascent phase in order to more accurately reach its target apogee. During the course of its double stage recovery, *VES* will separate into three parts to deploy first the drogue parachute and then the main parachute.

The rocket itself will be composed of carbon fiber body tubes and of carbon reinforced polymer flanges and finset. The tip of the rocket will be made out of aluminum. The flanges and the finset will be 3D printed.

2.2 Geometrical characteristics of the outer body

Below is an image of VES in the OpenRocket software with the internal components of the rocket. The center of mass and the center of pressure of the entire rocket are visible in the picture.

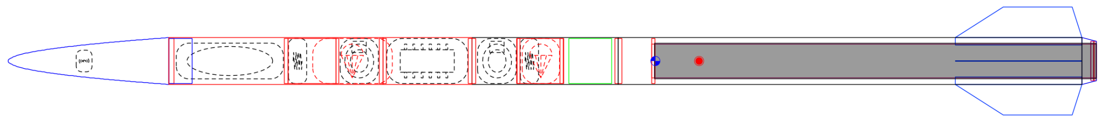

Figure 2.2: VES as it appears in OpenRocket.

Below is a table listing the relevant geometrical characteristics of the outer fuselage of the rocket.

Parameter	Value/Type
Total length l	3.41 m
Diameter of the body tube d	13.4 cm
Length of the body tube l_{BT}	2.92 m
Nosecone length l_{NS}	45 cm
Nosecone type	Power law series
Power law series exponent n	$\frac{1}{2}$ (Parabolic)
Boat tail length l_{BT}	3.5 cm
Boat tail exit diameter d_{EXIT}	11.5 cm
Boat tail type	Conical

Table 2.1: Geometrical characteristics of the fuselage of VES.

Below are an image and a table detailing the characteristics of the finset and the geometry of its fins.

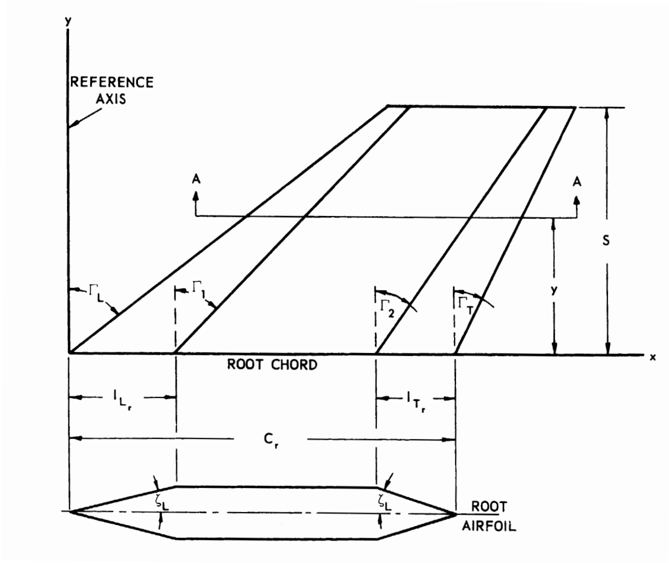

Figure 2.3: Diagram of the fin's geometry parameters.

Parameter	Value/Type
Number of fins	4
Fins' axial position x_F	2.67 m
Airfoil type	Hexagonal
c_r	34.0 cm
s	11.3 cm
l_{L_r}	8.88 cm
l_{T_r}	8.88 cm
Maximum thickness t	8 mm
Γ_L	$\frac{\pi}{6}$ (rad)
Γ_1	$\frac{\pi}{6}$ (rad)
Γ_2	-0.161 (rad)
Γ_T	-0.161 (rad)

Table 2.2: Geometrical characteristics of the finset of VES.

Below is an image of the CAD model of a fin for VES. Clearly visible is the connection between the two slopes of the leading and trailing edges of the fin at its tip.

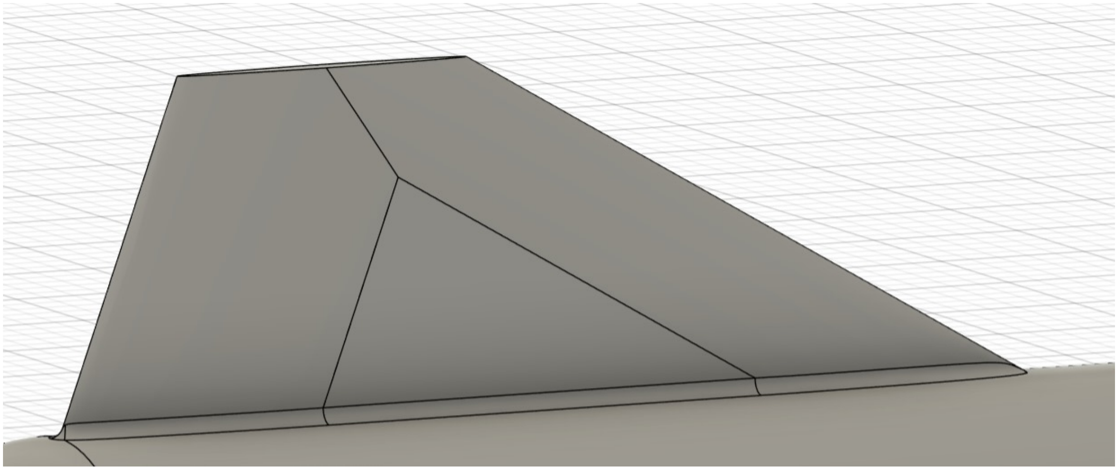

Figure 2.4: Detail of a single fin as it appears in the CAD software.

2.3 CONOPS of *VES*

Below is a table detailing the *CONOPS*, or *CONcept of OPerationS*, of VES. The various flight stages are presented, and a brief description of each one is provided.

Stage	Description	Duration
Powered Flight	The boost phase starts when the motor is ignited. During the boost phase, the rocket climbs until the burnout happens, when the motors run out of fuel and shut down. It is the only powered flight stage. During this stage, the rocket reaches the supersonic regime, with speeds of up to Mach 1.6/1.7.	est. 5/6 s

Stage	Description	Duration
Coast Phase	During this phase the rocket continues to climb due to inertia, following the shut down of the motor. It is slowed down by air resistance and gravity until it reaches the apogee, the highest altitude reached during the flight. The rocket is decelerated from supersonic speeds down to subsonic ones.	est. 20 s
Initial Descent	As soon as the rocket reaches the apogee and stops, the nosecone is ejected and the drogue parachute is deployed by the altimeter. The rocket then starts to fall, its descent slowed down by the drogue parachute.	est. 100 s
Final Descent	When the rocket reaches a certain altitude, the main parachute is deployed. The main parachute further slows down the rocket to a velocity which is safe to land.	est. 100 s

Table 2.3: CONOPS table

Chapter 3

Theoretical background of the considered reduced order method

3.1 Brief outline of available methods

Various methods used to calculate the wave drag of a generic wing-body configuration were given by a number of authors in the last century. In particular, three methods are usually adopted to calculate the wave drag, namely the *supersonic area rule*, or simply *area rule*, the *moment of area rule* and the *transfer rule*, all derived from linearised theory. They are used in non-lifting conditions, i.e. when the wing-body combination has two planes of symmetry.[3][4][5][6]

The first method achieves its result by calculating the average of a series of double integrals. The second method calculates the wave drag through a Fourier series. The third calculates the wave drag by adding three double integrals together. All of the methods take into account the axial distribution of cross-sectional area of the configuration. They also all assume that the effect of the interference velocity potential on the wave drag is negligible, meaning that the perturbation velocity potential of a given wing-body combination is equal to the sum of the single perturbation velocity potentials of the isolated exposed wing and of the isolated body. Also, it was demonstrated that the results given by those three methods are equivalent between each other.[3]

The *supersonic area rule* was chosen to be implemented in MATLAB in order to carry out the calculations needed for this analysis. This method was chosen as it is one of the most straightforward methods and because its practical application is well documented and easy to follow. A detailed description of the *supersonic area rule* is presented in the next section. The other methods, while deserving a mention, will not be discussed further in this document.

3.2 Supersonic area rule

3.2.1 Assumptions of the method

Because of the fundamental assumption that the effect of the interference velocity potential on the wave drag is zero, the application of the method is restricted to wing-body combinations with slender fuselages and thin wings.[7]

The former condition is satisfied when at a given Mach number M the quantity $\frac{R\sqrt{M^2-1}}{l}$ is small everywhere along the body axis, where R is the radius of the fuselage cross-section at a given point of the body axis and l is the total length of the body.[3]

In addition to the previous ones, another condition is set regarding the smoothness of the outer fuselage of the body. The local fuselage radius of curvature, in any meridian section, must always be large compared to l, meaning that discrete changes in the surface slope with respect to the body axis are not contemplated.[7]

3.2.2 Description of the method

As previously mentioned, the *supersonic area rule* calculates the wave drag of wing-body configuration by calculating the average of a series of double integrals on the axial cross-sectional area distribution of the equivalent bodies of revolutions of the configuration $S(x, \theta, M)$. The double integral calculates the wave drag of a single equivalent body of revolution.[8]

For a given free-stream Mach number M each value of θ specifies one member of the series of the equivalent bodies of revolution for the axial position x. $S(x, \theta, M)$, which will be simply referred to as *elemental area distribution* going forward, is obtained as the frontal projection of the cross-sectional area distribution intercepted on the given configuration by a set of parallel oblique planes tangent to the Mach cones. θ is the polar coordinate of the tangent planes with respect to the Mach cones when the object is fixed in space. It can also be interpreted as the roll angle of the object with respect to the horizontal plane when the oblique planes are fixed.[7][8]

Figures 3.1 and 3.2 show these concepts in relation to an example non lifting wing-body combination.

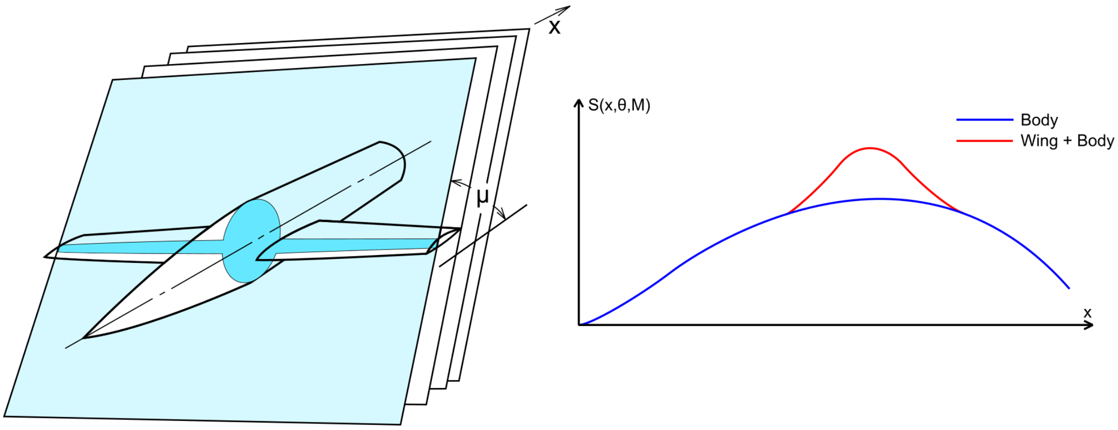

Figure 3.1: Example of an elemental area distribution for a set Mach number. In this example, $\theta = 0$. $\mu = \arcsin(\frac{1}{M})$ is the Mach angle. In cyan the projection of the cross section of the object onto the lighter cyan plane is highlighted. The blue graph represents the elemental area distribution of the body alone, the red one represents the added contribution of the wings.

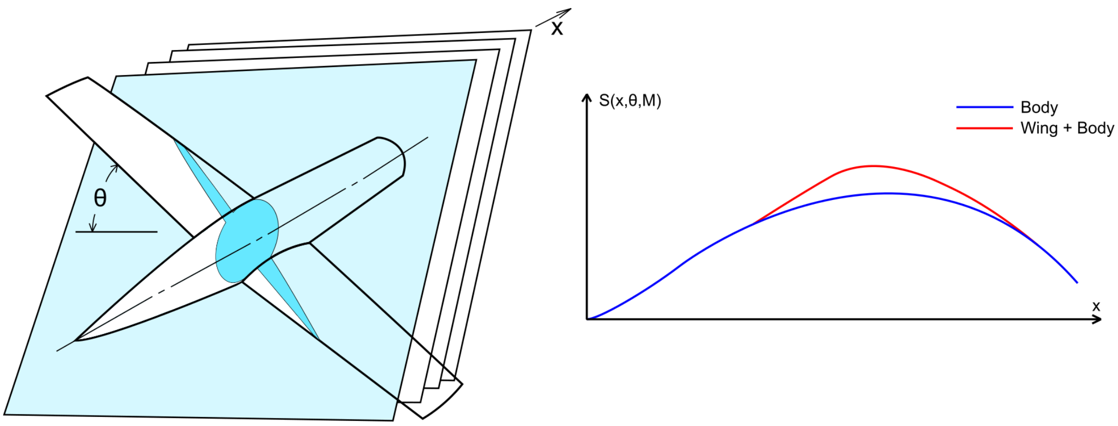

Figure 3.2: Another example of an elemental area distribution for a set Mach number. In this example, the object and Mach number are the same as the previous image but $\theta \neq 0$. The elemental area distribution of the body doesn't change because it's axisymmetric, but the contribution of the wings changes due to the different relative positions of the wings and the incident plane.

In formulas, the wave drag of a the whole wing-body configuration is given as follows:

$$D = \frac{1}{2\pi} \int_0^{2\pi} D\{S(x,\theta,M)\}\, d\theta \tag{3.1}$$

where $D\{S(x,\theta,M)\}$ is:

$$D\{S(x,\theta,M)\} = -\frac{q}{2\pi} \iint S''(x_1,\theta,M) S''(x_2,\theta,M) \log|x_1 - x_2|\, dx_1\, dx_2 \tag{3.2}$$

where $S''(x,\theta,M)$ denotes $\frac{\partial^2 S(x,\theta,M)}{\partial x^2}$, q is the kinetic pressure $q = \frac{1}{2}\rho_0 U_0^2$, x_1 and x_2 are the names of the variables of integration given to differentiate between them, although the integral itself is done two times on the same coordinate x. The double integral containing $S''(x,\theta,M)$ is evaluated for all values of x where $S''(x,\theta,M)$ is defined. In practice, the double integral is done on a square shaped domain in x_1 and x_2 which bounds all of the values of (x_1, x_2) where the product $S''(x_1,\theta,M)S''(x_2,\theta,M) \neq 0$. Note that the logarithm presents a singularity when $x_1 = x_2$, so those points are excluded from the domain of integration. It is implicitly assumed that $S'(X,\theta,M)$ is continuous in the domain where the elemental area distribution is defined. Note that the quantity $D\{S(x,\theta,M)\}$ is not a function of x. The notation used for this quantity follows the notation used by the sources.[3][7]

3.2.3 Definition of the elemental area distribution

Using equation 3.1 and 3.2 the wave drag can be calculated knowing the function $S(x,\theta,M)$. To determine $S(x,\theta,M)$ an approximation valid for thin wings and slender bodies combinations is used. It is assumed that the elemental area distribution can be written as:

$$S(x,\theta,M) = S_B(x) + S_{NW}(x,\theta,M) \tag{3.3}$$

where $S_B(x)$ is the cross-sectional area distribution of the body and $S_{NW}(x,\theta,M)$ is the contribution given by the net wing. The net wing is formed by placing together the two diametrically opposite exposed wing panels.[3]

For a thin wing lying on the plane $z = 0$ an elemental wing area distribution $S_W(x,\theta,M)$ is given by the following expression:

$$S_W(x,\theta,M) = \int_{T\neq 0} T(x + \beta y \cos(\theta), y)\, dy \tag{3.4}$$

where $T(x,y)$ denotes the wing thickness at the point (x,y) and $\beta = \sqrt{M^2 - 1}$. The integral is evaluated for each (x,y) point on the planform of the wings where $T(x,y)$ is defined, as in figures 3.3 and 3.4. In the case of two orthogonally placed pair of wings around the body, the thickness function for the net wing which has the z axis located in the span-wise direction will be $T(x, z)$ and the integral 3.4 will be done on z. As previously said, to get $S_{NW}(x, \theta, M)$ the two exposed wing panels should be placed together side by side and $T(x, y)$ or $T(x, z)$ modified accordingly.[3]

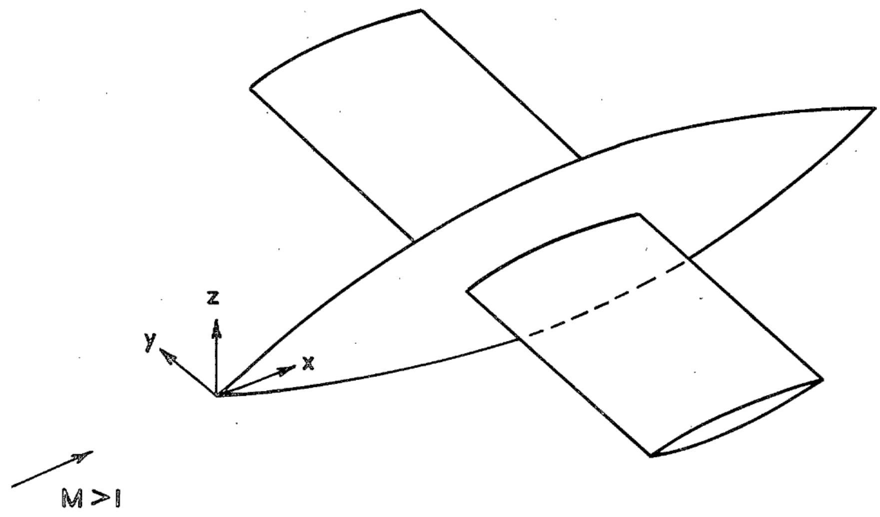

Figure 3.3: System of coordinates for a generic wing-body combination.

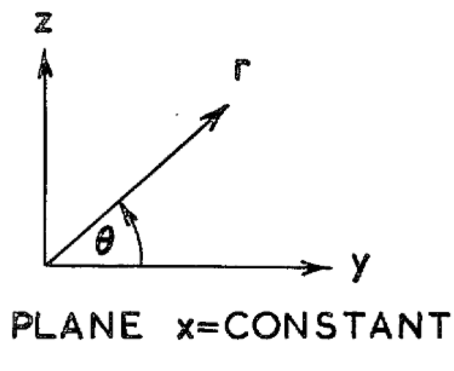

Figure 3.4: System of coordinates for a generic plane perpendicular to the body axis x.

Since the interference velocity potential is assumed to be negligible and the method itself is derived from linearised-theory, if a wing-body combination has two net wings present to get the total expression of $S(x, \theta, M)$ the elemental area distributions of each single net wing needs to be added together.

It follows that, to get a complete description of the wing-body combination in order to calculate the wave drag, the functions $S_B(x)$ and $T(x, y)$ need to be known.

Chapter 4

Definition of geometrical functions

4.1 Cross-sectional area distribution of the body

To get the cross-sectional area distribution $S_B(x)$ of VES the nosecone is first considered.

Since the nosecone is of the power law series, to get the radius of the nose at a certain axial point the following expression is used:

$$R(x) = R_B \left(\frac{x}{l_{NS}}\right)^n \tag{4.1}$$

where R_B is the radius of the body tube, l_{NS} is the length of the nosecone and n is the chosen exponent for the power series, for VES it is $\frac{1}{2}$, which means that the nosecone is shaped like a paraboloid.

Therefore we get:

$$S_{NS}(x) = \pi R_B^2 \frac{x}{l_{NS}} \tag{4.2}$$

therefore the cross-sectional area distribution of the nosecone is linear with respect to the axial position x. This means that for the nosecone:

$$S''(x, \theta, M) = S''_{NS}(x) = \frac{d^2}{dx^2}\left(\pi R_B^2 \frac{x}{l_{NS}}\right) = 0 \tag{4.3}$$

so the contribution of the nosecone for the integral in equation 3.2 is zero. This result will be exploited in the MATLAB script implementation.

For the body tube, we have a constant cross-sectional area distribution $S_B(x) = \pi R_B^2$, meaning that:

$$S''(x, \theta, M) = S''_B(x) = \frac{d^2}{dx^2}(\pi R_B^2) = 0 \tag{4.4}$$

so the body tube as well gives no contribution to equation 3.2.

Finally for the boat tail we have a linear decrease in the diameter:

$$d(x) = d_B + \frac{d_{EXIT} - d_B}{l_{BT}}(x - x_{BT}) \tag{4.5}$$

where x_{BT} is the axial position of the change in slope of the fuselage, where the boat tail begins. This means that for the boat tail:

$$S_{BT}(x) = \frac{\pi}{4}\left(d_B + \frac{d_{EXIT} - d_B}{l_{BT}}(x - x_{BT})\right)^2 \tag{4.6}$$

It follows that:

$$S''(x, \theta, M) = S''_{BT}(x) = \frac{\pi}{2}\left(\frac{d_{EXIT} - d_B}{l_{BT}}\right)^2 \tag{4.7}$$

therefore the boat tail is the only part of the body that has a non zero contribution to the equation 3.2.

4.2 Thickness function of the fins

The thickness function $T(x, y)$ for a single exposed fin lying on the $z = 0$ plane is first defined as a starting point to get the $T(x, y)$ function for the whole net fin.

Since for both the leading edge and the trailing edge the edges between the planar region of the fin and the sloped sections are parallel to the corresponding leading or trailing edge, the semi-thickness of the blunted edge section can be expressed as the function of a plane that contains the two parallel edges.
The slopes of the leading and trailing edges beyond the point where the planar region ends can be expressed by the same plane equations before the conjunction. Since the airfoil is symmetrical with respect to the $z = 0$ plane, it follows that the whole thickness can be found by simply doubling the semi-thickness and can therefore be expressed as a function of three different planes.

To find the planes that describe the function $T(x, y)$ in those sections of the fin, the expression of the straight lines lying in the plane $z = 0$ that bound the sections need to be found first. Assuming that the root chord lies in the x axis, for the leading edge straight line up to the point where the planar region ends the following expression is used:

$$y = \frac{s}{x_L}(x - x_F) \tag{4.8}$$

where x_F is the axial position of the leading edge at the root of the fin, and x_L is the axial distance of the leading edge at the tip of the fin with respect to x_F, which is found in the following way:

$$x_L = s\tan(\Gamma_L) \tag{4.9}$$

For the corresponding dividing edge between the sloped and plain sections of the fins up until the point where the planar region terminates the following expression is used:

$$y = \frac{s}{x_L}(x - x_F - l_{L_r}) \tag{4.10}$$

therefore, by defining the thickness of the fin to be equal to the maximum thickness along the edge defined by equation 4.10 the following equation for the plane containing the two edges is found:

$$t(x - x_F) - \frac{t x_L}{s} y - l_{L_r} z = 0 \tag{4.11}$$

where z is the local total thickness at the position (x, y) of the plan of the fin. Therefore, for the section of the fin constrained by the 4.8 and 4.10 straight lines $T(x, y)$ is defined as:

$$T(x, y) = \frac{t}{l_{L_r}}\left(x - x_F - \frac{x_L}{s} y\right) \tag{4.12}$$

The equations of the edges that constrain the sloped section of the fin for the trailing edge before the planar region ends are defined in the same way as before. The straight line defining the trailing edge is defined as follows:

$$y = \frac{s}{x_T}(x - x_F - c_r) \tag{4.13}$$

where x_T is the distance of the tip of the trailing edge with respect to the trailing end of the root chord, which is found as:

$$x_T = s\tan(\Gamma_T) \tag{4.14}$$

For the other corresponding edge the following expression is used:

$$y = \frac{s}{x_T}(x - x_F - c_r + l_{T_r}) \tag{4.15}$$

and, in the same way as before the following equation of a plane is found:

$$tx - \frac{tx_T}{s}y + l_{T_r}z - t(c_r + x_F) = 0 \tag{4.16}$$

from which the following thickness function for the fin section bounded by the straight lines 4.13 and 4.15 is found:

$$T(x,y) = \frac{t}{l_{T_r}}\left(\frac{x_T}{s}y - x + x_F + c_r\right) \tag{4.17}$$

The planar section of the fin, bounded by the straight lines 4.10 and 4.15, has a constant thickness:

$$T(x,y) = t \tag{4.18}$$

The span-wise position of the point where the planar region ends is found by intersecting the lines 4.10 and 4.15, and is as follows:

$$y_c = s\frac{(l_{L_r} + l_{T_r} - c_r)}{x_T - x_L} \tag{4.19}$$

The line dividing the two slopes of the leading and trailing edges after the end of the planar section is found by first finding the line which is the intersection of the planes 4.11 and 4.16 and then finding its orthogonal projection in plane $z = 0$. The line found is the following:

$$y = \frac{s}{x_L + \frac{l_{L_r}}{l_{T_r}}}\left(x\left(1 + \frac{l_{L_r}}{l_{T_r}}\right) - x_F - \frac{l_{L_r}}{l_{T_r}}(c_r + x_F)\right) \tag{4.20}$$

Using the equations 4.12, 4.17 and 4.18 the thickness function $T(x,y)$ is defined for a single exposed fin.

To get the thickness function for a net fin created by aligning two exposed fins, assuming that one of the fins lies in the positive y half-plane and the other in the negative y half-plane, the function $T(x,y)$ is extended for the negative y points by inverting the sign of y for the previous equations.

Since the two net wings of the finset are identical, in the final calculation the total $S_{NW}(x,\theta,M)$ will be obtained by doubling the result for a single net fin.

4.3 Continuity of the first order derivative of the elemental area distribution

As previously mentioned, an essential assumption for the supersonic area rule is that the first order derivative $S'(x, \theta, M)$ of the elemental area distribution should be continuous for every x where it is defined. This condition isn't respected in at least one region in VES.

From equation 4.6 the cross-sectional area distribution of the boat tail was obtained. Taking the first order derivative gives this result:

$$S'_{BT}(x) = \frac{\pi}{2}\frac{d_{EXIT} - d_B}{l_{BT}}\left(d_B + \frac{d_{EXIT} - d_B}{l_{BT}}(x - x_{BT})\right) \tag{4.21}$$

which gives a non zero value for $x = x_{BT}$, or the point where the cone shaped boat tail begins. Since for the body tube $S'(x) = 0$, this means that the aforementioned condition isn't satisfied for $x = x_{BT}$. The condition isn't satisfied either for the point $x = l_{NS}$, the point where the nosecone ends and the body tube begins, as in this point $S'(x) = S'_{NS}(x) = \frac{\pi R_B^2}{l_{NS}}$. The nosecone in this point isn't *smoothly* connected with the cylindrical part of the body. However, given that the condition isn't satisfied for only two points of the fuselage, it is known that the method is still applicable if the quantity of points where the condition doesn't apply is finite and low.[3]

Chapter 5

MATLAB implementation

5.1 Initial remarks

As previously mentioned, the second order derivative of the cross-sectional area distribution of the body tube and of the nosecone is zero. The only section in the fuselage where this isn't the case is the boat tail, where the second order derivative is found to be constant:

$$S''(x, \theta, M) = S''_{BT}(x) = \frac{\pi}{2}\left(\frac{d_{EXIT} - d_B}{l_{BT}}\right) \tag{5.1}$$

This means that the only sections of the outer body of the rocket that give a non zero contribution to equation 3.2 are the finset and the boat tail of the rocket.
In fact, if we allow the domain of integration for equation 3.2 to be defined as such:

$$\mathscr{D} = \{x_1, x_2 \in \mathbb{R} : 0 \leq x_1 \leq l, 0 \leq x_2 \leq l, x_1 \neq x_2\} \tag{5.2}$$

where l is the total axial length of the body, there will inevitably be points where either x_1 or x_2 or both will be lower than x_F, the position of the fins, and so at least one between $S''(x_1, \theta, M)$ and $S''(x_2, \theta, M)$ will be zero in integral 3.2.
It is better therefore to shift the origin of the system of coordinates from the tip of the nosecone, as implicitly done up to this point, to the point x_F where the fins are located, and define the domain of integration like so:

$$\mathscr{D} = \{x_1, x_2 \in \mathbb{R} : 0 \leq x_1 \leq c_r + l_{BT}, 0 \leq x_2 \leq c_r + l_{BT}, x_1 \neq x_2\} \tag{5.3}$$

with the upper boundary being $c_r + l_{BT}$ (figure 2.3 and table 2.2 for reference) for both x_1 and x_2 as the boat tail begins immediately after the end of the fins.
However, this domain is too small. If we allow $\mathscr{D}$ to be defined as in 5.3, there will be some points missing for the calculation of the integral 3.4. Because $T(x, y)$

is defined for $x > 0$ if we shift the origin of the system of coordinates as done previously, the function $T(x + \beta y \cos(\theta), y)$ can accept negative values of x if $\cos(\theta) > 0$. Also, since $T(x, y)$ is defined for $x < c_r$, the function $T(x+\beta y \cos(\theta), y)$ can accept values of x higher than c_r if $\cos(\theta) < 0$. Since the boundary values of $\cos(\theta)$ are -1 and 1 and $y_{max} = s$, the domain of integration can be defined as:

$$\mathscr{D} = \{x_1, x_2 \in \mathbb{R} : -\beta s \leq x_1 \leq \varphi, -\beta s \leq x_2 \leq \varphi, x_1 \neq x_2\} \tag{5.4}$$

so that no points are missed during the integration of equation 3.4. φ is defined to be $\varphi = max(c_r + \beta s, c_r + l_{BT})$.
By defining the domain of integration in this way, the nosecone and the section of the body tube without the fins are effectively discarded, and only the finset and boat tail of the rocket are considered for the calculation. Taking into account this last fact, everything has been defined for the calculation to take place, and the MATLAB code can be implemented.

5.2 Code implementation

First, the thickness function for a net fin and the cross-sectional area distribution of the section of the fuselage mounting the fins and the boat tail were defined in MATLAB. The definition of these functions are the starting point of the MATLAB code implementation. The definition of these functions is described in the previous chapter.

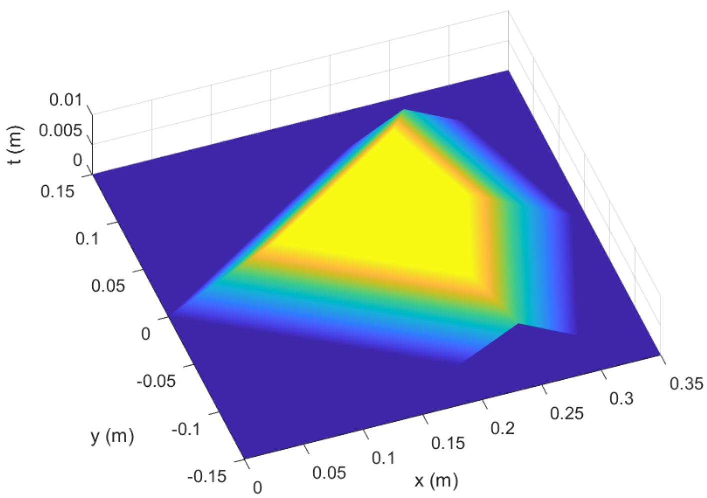

Figure 5.1: Thickness function for a single net fin as it appears in MATLAB. The alignment of the two single exposed fins is visible.

5.2.1 Calculation of the elemental area distribution for the fin set

To get the elemental area expression for the net fin, a MATLAB function handle is defined which calls the MATLAB `integral` function on $T(x, y)$. The calculation that is done is the following:

$$S_{NW}(x, \theta, M) = \int_{-s}^{s} T(x + \beta(M) y \cos(\theta), y)\, dy \tag{5.5}$$

The integral is performed numerically by `integral`. The result is then doubled to account for the other net fin present in the finset of VES.

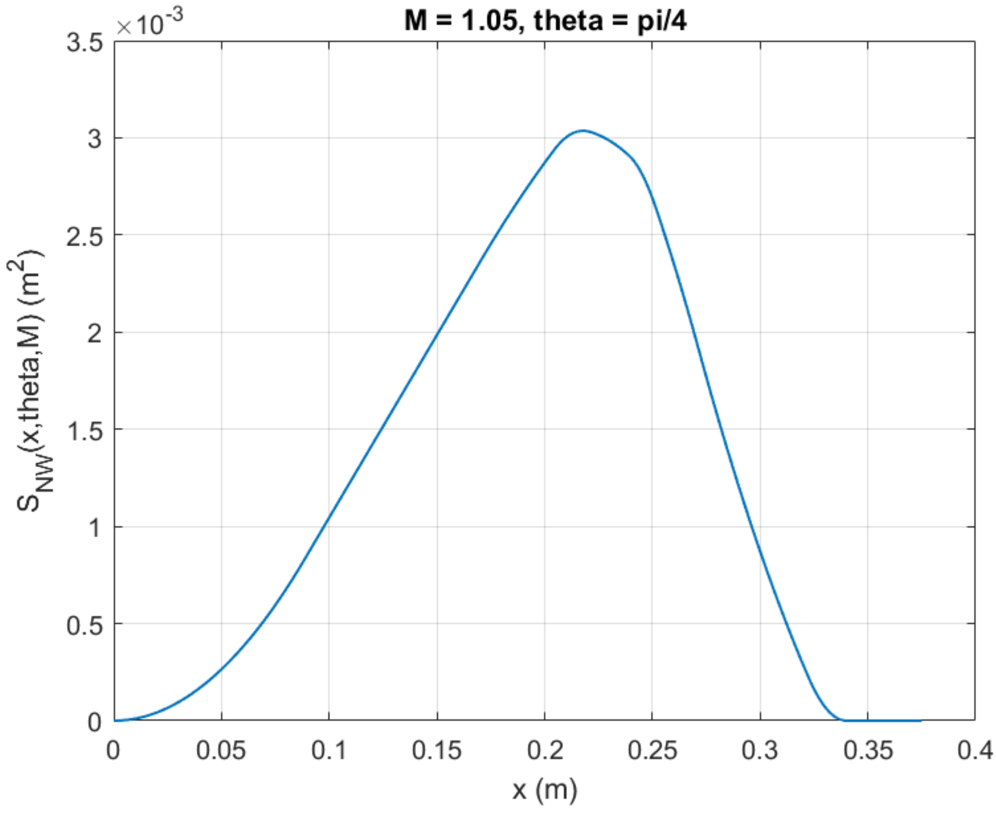

Figure 5.2: Example calculation of the elemental area distribution for the finset. In this example, $\theta = \pi/4$ and $M = 1.05$.

5.2.2 Calculation of the total elemental area distribution and of its second order derivative

To calculate the function $S''(x, \theta, M)$ which is needed for the 3.2 integral, a function called `d2_elemental_area` was defined. It takes in input x, θ and β and first performs the calculation described in the previous subsection. It then calculates the second order derivative of $S_{NW}(x, \theta, M)$ by calling the MATLAB function `gradient` two times consecutively. Finally, it adds the constant contribution given by the second order derivative of the cross-sectional area distribution of the boat tail to get $S''(x, \theta, M)$.

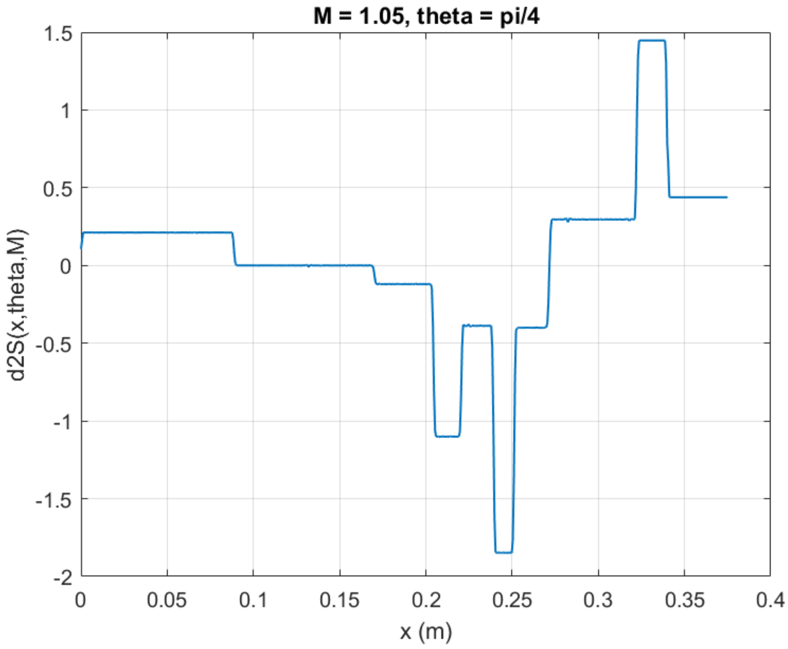

Figure 5.3: Example calculation of $S''(x, \theta, M)$. In this example, $\theta = \pi/4$ and $M = 1.05$. The discontinuity of the function is visible. Very small local spikes caused by numerical cancellation in the sections where the function should be constant are visible.

Due to the delicacy of the second order derivation because of numerical cancellation, the amount of points where the function is evaluated to calculate the second order derivative need to be set so that a sufficient amount of points is provided to accurately calculate the next integrals. However, the amount of points considered shouldn't be too high due to aforementioned problem, that can yield a bad approximation of the function. For the calculation, it was decided to employ 300 points as a compromise.

5.2.3 Calculation of the double integral

To calculate the double integral in equation 3.2 the function `operation` was first defined. It takes in input x_1, x_2, $theta$, $beta$ and n_{int}.

The value n_{int} defines the amount of points to evaluate the second order derivative calculated by `d2_elemental_area`. Once the second order derivative is evaluated in the specified number of equidistant points, a linear interpolation is done to be able to numerically perform the following integrals in more points than those specified by n_{int}.

The function `operation` then defines a function handle which performs the operation to obtain the surface to be integrated by the double integral. Below is a picture of the obtained surface in an example setting of θ and M.

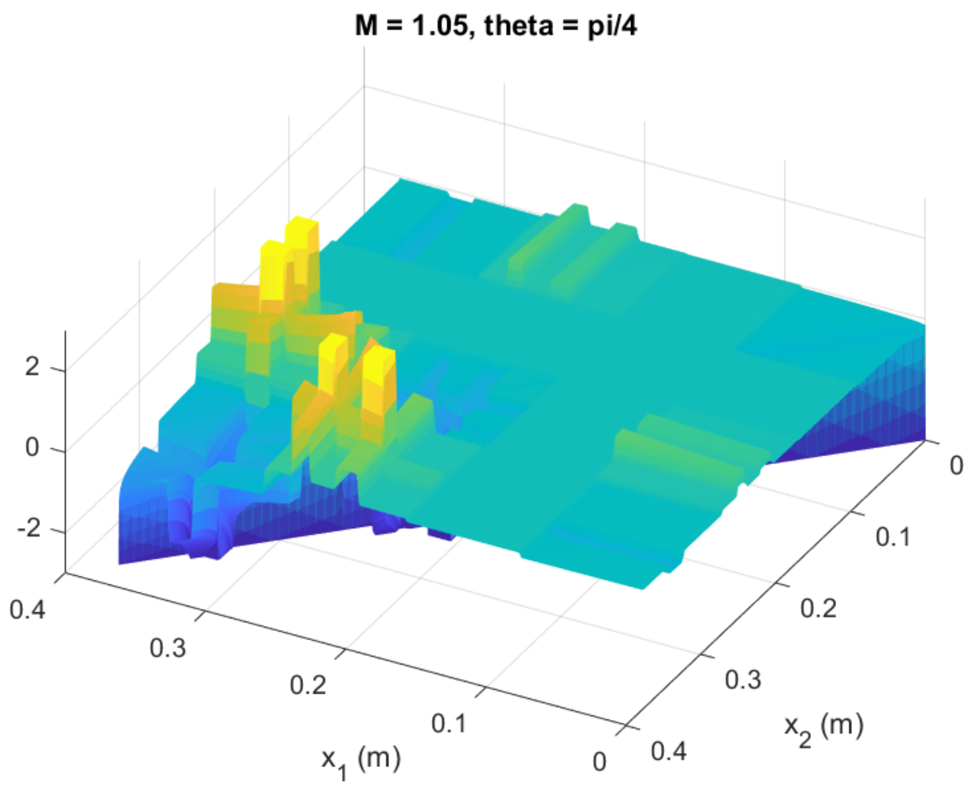

Figure 5.4: Example surface to be integrated. In this example, $\theta = \pi/4$ and $M = 1.05$.

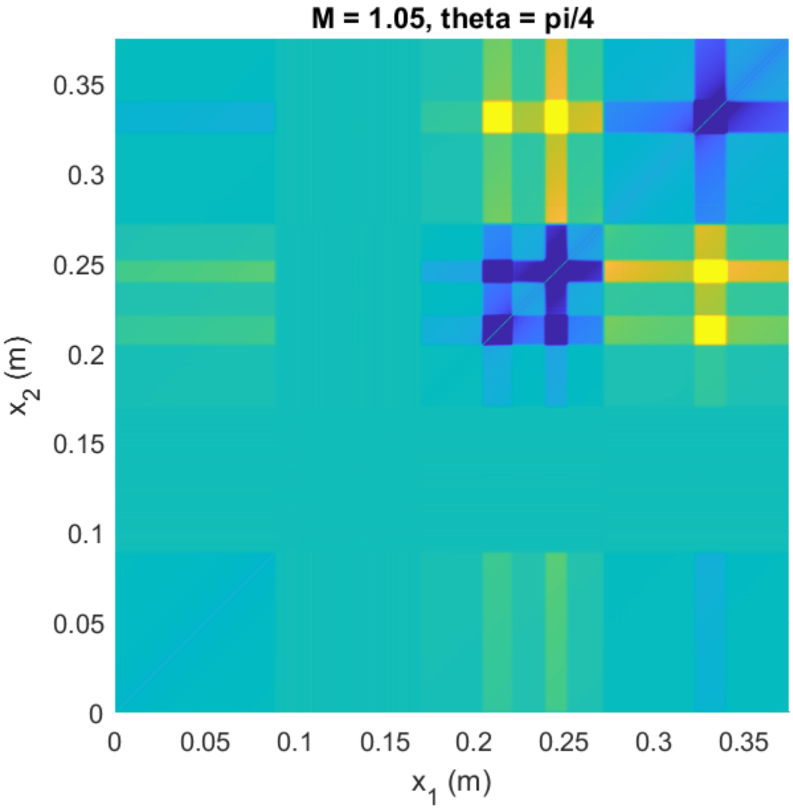

Figure 5.5: Height map of the previous example surface. The symmetries of the configuration are visible. The singularity present when $x_1 = x_2$ due to the expression inside the logarithm in equation 3.2 is also faintly visible.

The function then performs a final check on the values of the output matrix and substitutes each NaN and Inf value caused by the singularity for $x_1 = x_2$ with the value 0. This is done as an approximation to be able to calculate the double integral simply with a squared domain.

The double integral:

$$D\{S(x,\theta,M)\} = -\frac{q}{2\pi} \iint S''(x_1,\theta,M)S''(x_2,\theta,M) \log|x_1 - x_2| \, dx_1 \, dx_2 \qquad (5.6)$$

is then calculated by the function `final_integ` using the MATLAB function `trapz`, which numerically integrates the function `operation` with the trapezoidal method.

5.2.4 Final integration

The previously mentioned function `final_integ`, which takes in input θ and β, calculates the double integral for each different value of θ and for a set value of M. The results of `final_integ` are then averaged as per equation 3.1 to get the wave drag D for a set Mach number.

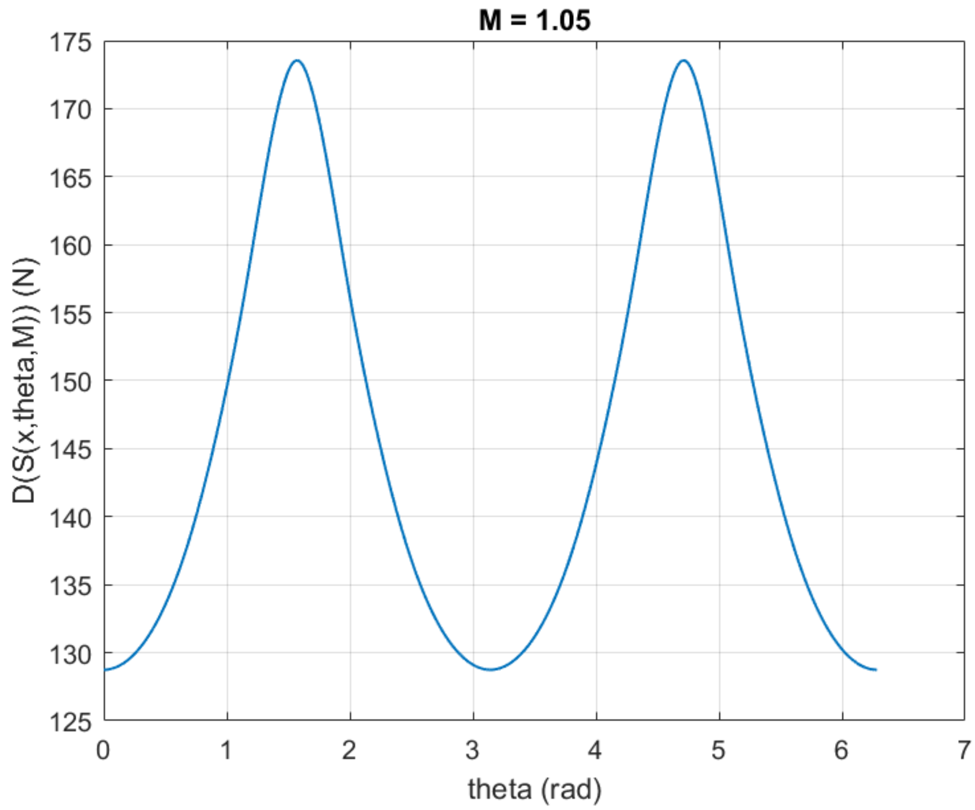

Figure 5.6: Example calculation of $D\{S(x,\theta,M)\}$. In this example, $M = 1.05$.

Finally, the wave drag coefficient C_D is calculated with the cross-section of the body tube used as the reference area. The process is iterated for each Mach number the calculations need to be done. The MATLAB script then displays the resultant C_D to Mach number curve. The complete MATLAB script is included in Appendix A of this document.

Chapter 6

Final results and conclusion

6.1 Wave drag curve

The MATLAB script produced the following C_D to Mach number curve for VES for Mach numbers between 1 and 2.

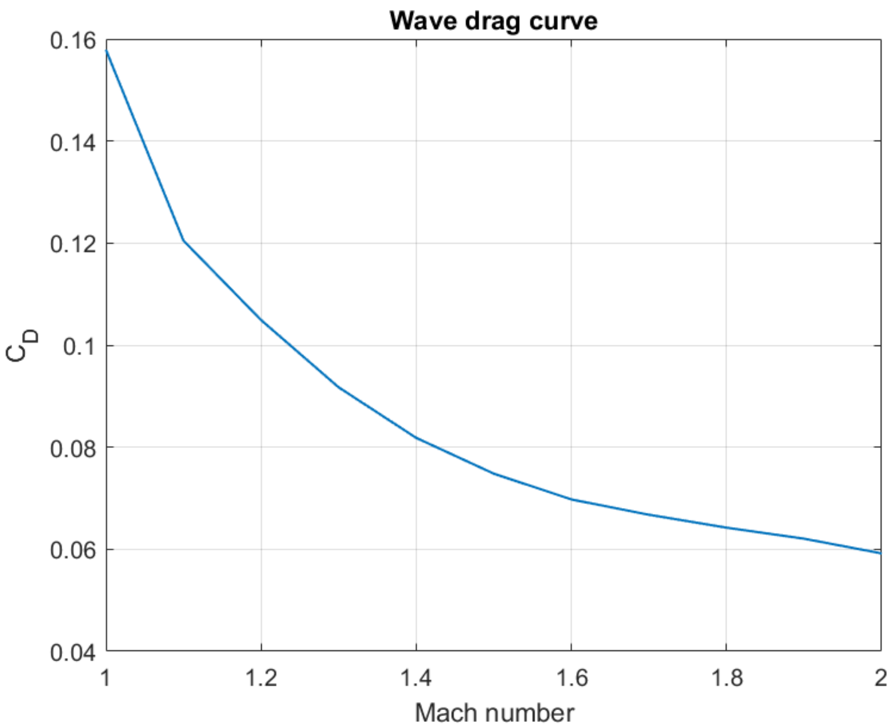

Figure 6.1: Wave drag coefficient curve for VES.

It can be seen that at $M = 1$ the C_D is equal to around 0.16. It then goes down to around 0.06 for $M > 1.8$.

To get a better idea of the influence of the wave drag in the total resistance acting on VES, the total drag coefficient $C_{D_{tot}}$ to Mach number curve was calculated using a semi-empirical software similar to DATCOM that is used by the *PoliTo Rocket Team* to estimate the total drag coefficient. The curve obtained is the following:

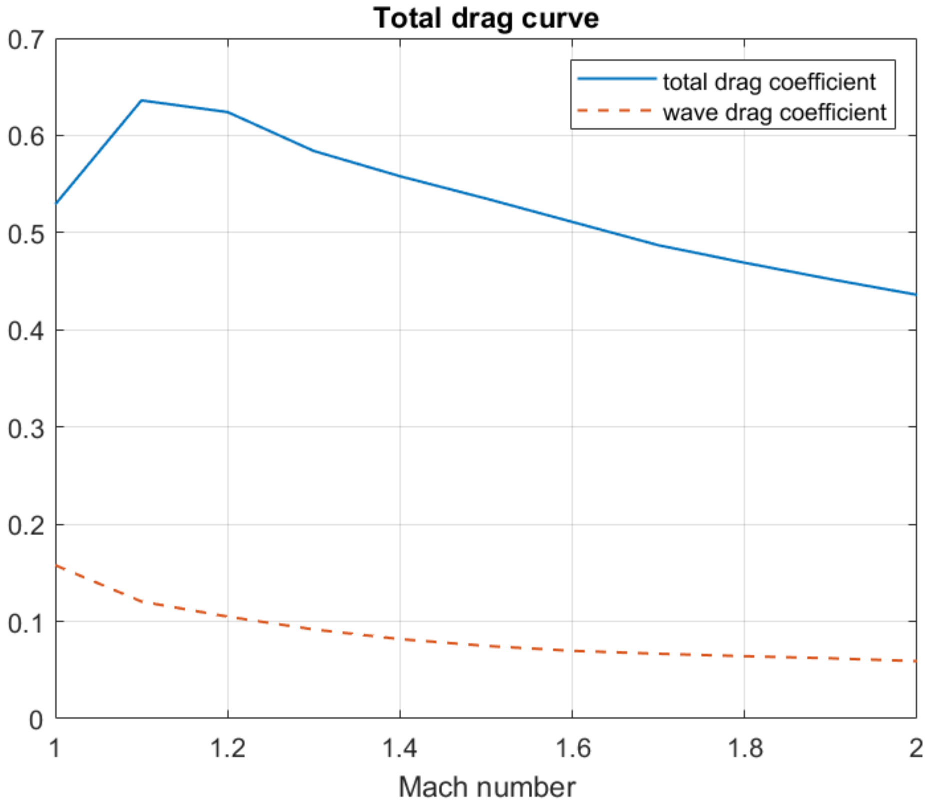

Figure 6.2: Total drag coefficient curve for VES. The dashed line represents the wave drag coefficient only.

The total drag curve reaches its maximum value of around 0.64 at $M = 1.1$, while the wave drag curve has its maximum for $M = 1$. After $M = 1.1$, the total drag starts to decrease as well, reaching the minimum value of around 0.44 for $M = 2$.

Another diagram was then made, this time by dividing the wave drag coefficient by the total wave drag coefficient to get the fraction of the total resistance consisting of the wave resistance. The result is as follows:

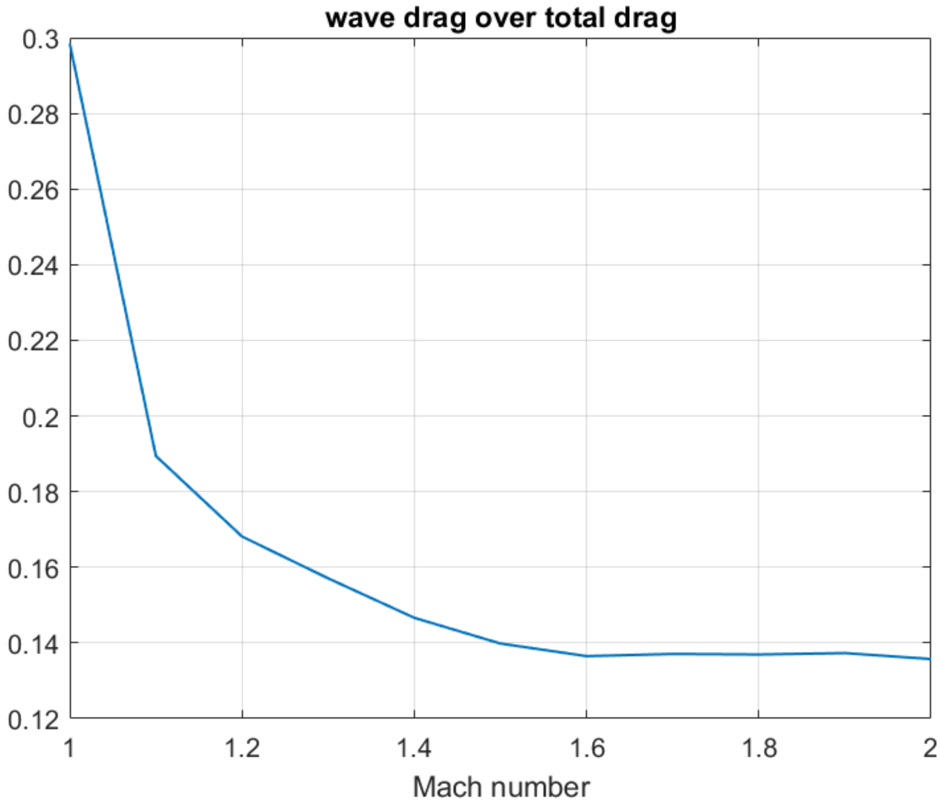

Figure 6.3: Graph of the wave drag coefficient over the total drag coefficient.

At $M = 1$, 30% of the total drag is made up by the wave drag. This percentage drops sharply after $M = 1.1$, stabilizing to around 13.5% for Mach numbers higher than 1.6, less than half of the percentage at $M = 1$.

6.2 Comparison with a CFD analysis

A CFD analysis is beyond the scope of this thesis, but it was conducted by the thesis supervisor to provide reference data for comparison with the present reduced-order model results. The simulations were performed using the *STAR-CCM+* software, assuming the standard density of air 1.225 $\frac{kg}{m^3}$, a fixed Mach number $M = 1.59$ and an angle of attack of zero. The analysis was conducted on only a quarter of the rocket looking from the tip of the nosecone. Since the object has two planes of symmetry, to obtain the total coefficient of drag the value needs to be multiplied by 4. The software returns only the total coefficient of drag $C_{D_{tot}}$, so a direct comparison isn't possible. Next are two images from the CFD analysis, the first one showing the velocity field in Mach numbers and the second one showing the pressure field around the quarter part of the rocket.

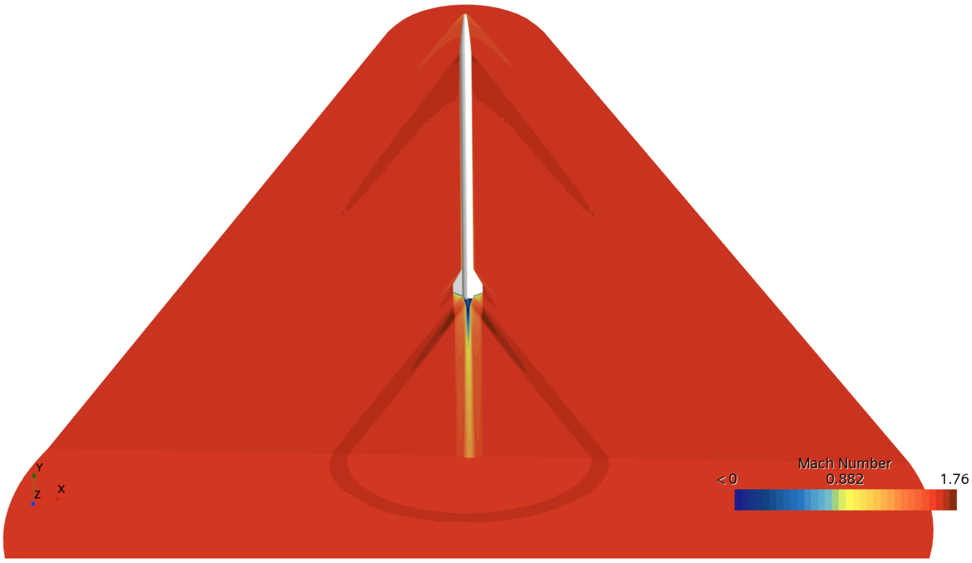

Figure 6.4: CFD calculated velocity field around VES. Values in Mach numbers.

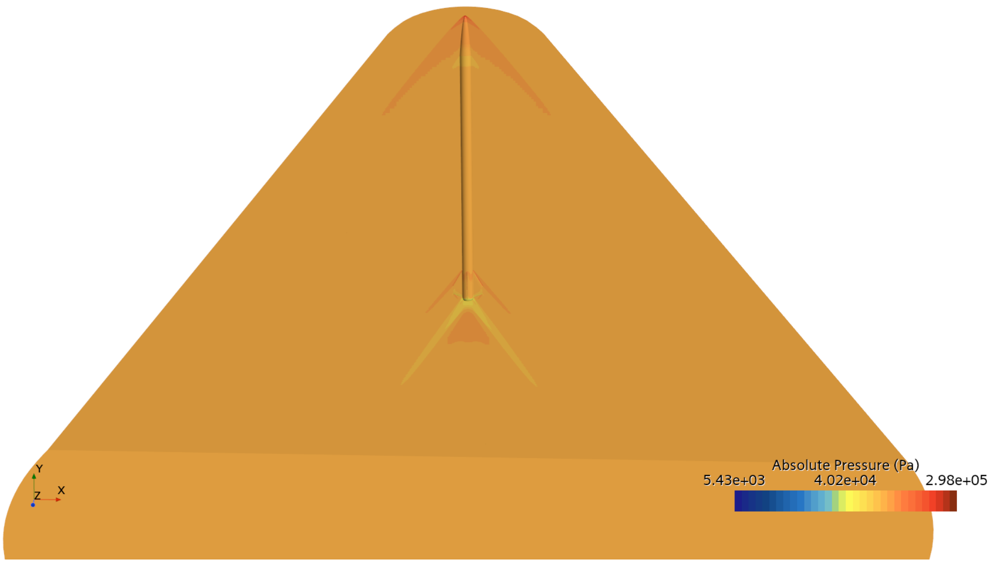

Figure 6.5: CFD calculated pressure field around VES. Values in Pa.

The CFD analysis calculated a total coefficient of drag of 0.115 for the quarter part of the rocket looking from the tip of the nosecone. The total coefficient of drag is therefore 0.460. This value is noticeably different from the value given by the semi-empirical software at $M = 1.6$, which is 0.511. This error between the CFD analysis and the semi-empirical software is expected, and it is assumed that the value given by CFD analysis is the more truthful one.

The CFD analysis also returned the percentage of the total drag acting on the fins alone divided by the total drag acting on the entire rocket. This percentage was found to be around 17%. This gives a total coefficient of drag acting on the fins of 0.0785. In comparison, by discarding the contribution of the boat tail in the MATLAB script a value of 0.0564 for the coefficient of wave drag of the fins alone was obtained. Below is a table tallying the contributions to the $C_{D_{tot}}$ of each single component from the CFD analysis, already multiplied by 4 to get the total for the whole rocket.

Part	Value	Percentage
Cylindrical body tube	0.114	24.7%
Fins	0.0785	17.1%
Nosecone	0.656	143%
Boat tail	-0.388	-84.3%
Total	0.460	100%

Table 6.1: Tally of the $C_{D_{tot}}$ from CFD. The values are for the whole rocket.

6.2.1 Comparison of the results and further comments

It is difficult to ascertain the accuracy of the method implemented on MATLAB as the CFD analysis doesn't return the coefficient of wave drag only. However, an indirect comparison can be made by comparing the total coefficient of drag acting on the fins alone given by the CFD analysis and the wave drag coefficient of the fins alone given by the MATLAB script. As mentioned previously, the respective values are 0.0785 and 0.0564.

The value obtained by MATLAB is lower than the CFD value, which is to be expected from a hypothetically correct value as the CFD coefficient is the total drag. The MATLAB value also constitutes a big fraction of the total, around 72%. This is plausible, as the drag of the wings, or in this case the fins, is small in conditions where the effects of shock waves are absent.[1] In fact, by subtracting from the total coefficient of drag obtained from the CFD analysis the wave drag obtained in MATLAB and by multiplying it by the fraction of the planform area of a single fin over the cross-section of the rocket to change the reference area, the value 0.0390 is obtained, which is the total coefficient of drag at an angle of attack of zero for the fins assuming no shock waves, C_{D_0}. Dividing it by two gives the value of 0.0195 for a couple of fins only, the typical configuration of planes. This value is plausible, as it is within the general range of values for this coefficient.[1] This doesn't prove the accuracy of the MATLAB value, but it otherwise indicates that it is coherent with the theory. Since the CFD analysis was conducted for $M = 1.59$ only, nothing can be said about the validity of the other data calculated using MATLAB.

The CFD analysis however shows the limits of the assumptions used to implement the MATLAB script. The first questionable one is that the assumption that the interference velocity potential is negligible in the calculations for the wave drag holds for VES, the key assumption for the method used. This is most likely not true, as the CFD analysis clearly showed that the flow around the finset and behind the boat tail of VES is very complex and, although not proven directly, it is improbable that all the interference effects on the flow around the rocket are negligible, especially around its bottom end.

Another obvious limit is the assumption that the elemental area distribution for the body for $M > 1$ remains equal to its distribution for $M = 1$. This was assumed to simplify the calculations so that for the body only the cross-sectional area needed to be calculated. Otherwise, the projected area on the oblique plane tangent to the Mach cone of the cross section needed to be calculated, which is more difficult. A more accurate value could be obtained by not neglecting the change in the elemental area distribution for the body for $M > 1$, although it would still be impossible to directly assess the accuracy of this other way of calculating the wave drag.

The CFD analysis also showed the limits of the semi-empirical DATCOM-like software used by the team to preliminarily assess the coefficient of drag in the early stage design of the rocket. However, while the values between the two analyses are different, the approximation of the semi-empirical software proves sufficient for the early stage design. The usefulness of the software is still present considering the difference in time spent during the setup of the conditions, as the CFD analysis needs the CAD of the object while the semi-empirical software only needs the values of certain geometrical parameters, and the computation of the results.
Next is a recap table listing the various values and percentages obtained while comparing all of the software used.

Parameter	Value	Percentage over CFD	Percentage over SES
$C_{D_{tot}}$ from CFD	0.460	100%	90.0%
$C_{D_{tot}}$ from SES	0.511	111%	100%
C_D from MATLAB	0.0703	15.3%	13.7%
C_D from MATLAB, fins only	0.0564	12.3%	11.0%
$C_{D_{tot}}$ from CFD, fins only	0.0785	17.1%	15.4%

Table 6.2: Recap of values and percentages of various coefficients calculated. *SES* stands for the semi-empirical software.

6.3 Conclusion

The method employed can't be directly and definitely assessed to be accurate from the results of the CFD analysis, due to the fact that this last analysis doesn't calculate the coefficient of wave drag only. The CFD analysis was also conducted for only one Mach number, $M = 1.59$, so nothing can be said about the accuracy and validity of the other results. Also, some assumptions of the method aren't valid in the conditions of the problem, while others need to be revised to get more accurate data in principle.

However, given the fact that standard CFD analysis doesn't calculate the wave drag alone acting on the rocket, and given that the results of the MATLAB numerical computation are coherent with the theory and, although not proven directly, seem to have an accuracy similar to the semi-empirical software used currently by the team with respect to the results from the CFD analysis, this method proves useful to the needs of the team in the early stage design of rockets. The calculation can also be crucially repeated every time the geometry of the rocket is modified through the software's parameters thanks to the far lower setup and computation times of this method than those needed for a CFD analysis. The method is considered to satisfy the needed criteria for a useful tool as defined in the abstract in this work with an accuracy that is deemed sufficient.

Appendix A

MATLAB script

```
clear
clc
close all

mach = linspace(1,2,11);
drag_w = zeros(1,length(mach));
c_d_w = zeros(1,length(mach));
air_density = 1.225;
theta = linspace(0,2*pi,300);

for i = 1:length(mach)
    beta = sqrt(mach(i).^2 - 1);
    y = final_integ(theta,beta);
    z = - y / (2 * pi);
    drag_w(i) = trapz(theta,z) / (2 * pi);
    c_d_w(i) = drag_w(i) / 0.014103;
end

figure
plot(mach,c_d_w,'LineWidth',1)
grid on
xlabel('Mach number')
ylabel('C_D')
title('Wave drag curve')


%% setup of the variables needed for various functions
function [root_chord,tip_chord,span,thickness,sweep_angle,x_le,...
    x_te_t,x_te_r,c_1,c_2,nosecone_length,boattail_length,...
    d_exit,x_fins,diameter,bodytube_length,cong] = setup_variables
    root_chord = 0.340; % in m
```

```
    tip_chord = 0.101; % in m
    span = 0.113; % in m
    thickness = 0.008; % in m
    nosecone_length = 0.45; % in m
    boattail_length = 0.035; % in m
    d_exit = 0.1155; % in m
    x_fins = 3.03; % in m
    diameter = 0.134; % in m
    bodytube_length = 2.92; % in m
    sweep_angle = 60; % in deg
    root_leading_edge_ratio = 0.261;
    root_trailing_edge_ratio = 0.261;
    sweep_angle = deg2rad(sweep_angle);
    x_le = span * tan(sweep_angle);
    c_1 = root_chord * root_leading_edge_ratio;
    c_2 = root_chord * root_trailing_edge_ratio;
    x_te_t = x_le + tip_chord - c_2;
    x_te_r = root_chord - c_2;
    cong = span*(x_te_r-c_1)/(x_le-x_te_t+x_te_r);
end


%% thickness function definition
function t = thickness_wing(x,y)
    [root_chord,~,span,thickness,~,x_le,x_te_t,x_te_r,...
        c_1,c_2] = setup_variables;
    t = zeros(1,length(y)); %thickness
    cong = span*(x_te_r-c_1)/(x_le-x_te_t+x_te_r);

    for i = 1:length(y)
        if y(i) >= 0
            if y(i) < cong
                if y(i) * x_le / span <= x(i) && y(i) * x_le / ...
                        span + c_1 >= x(i)
                    t(i) = (x(i) - y(i) * x_le / span) * ...
                        thickness / c_1;
                elseif y(i) * x_le / span + c_1 <= x(i) && y(i) * ...
                        (x_te_t - x_te_r) / span + x_te_r >= x(i)
                    t(i) = thickness;
                elseif y(i) * (x_te_t - x_te_r) / span + x_te_r ...
                        <= x(i) && y(i) * (x_te_t - x_te_r) / ...
                        span + root_chord >= x(i)
                    t(i) = (- x(i) + y(i) * (x_te_t - x_te_r) / ...
                        span + root_chord) * thickness / c_2;
                end
            elseif cong <= y(i) && y(i) <= span
                if y(i) * x_le / span <= x(i) && y(i) * ...
                        ((x_te_t - x_te_r) / c_2 + x_le / c_1) /...
                        (span * (1 / c_1 + 1 / c_2)) + root_chord ...
```

```matlab
                        / ((c_2 / c_1) + 1) >= x(i)
                    t(i) = (x(i) - (y(i) * x_le / span)) * ...
                        thickness / c_1;
                elseif y(i) * ((x_te_t - x_te_r) / c_2 + x_le / ...
                        c_1) / (span * (1 / c_1 + 1 / c_2)) + ...
                        root_chord / ((c_2 / c_1) + 1) <= x(i) && ...
                        y(i) * (x_te_t - x_te_r) / span + ...
                        root_chord >= x(i)
                    t(i) = (- x(i) + y(i) * (x_te_t - x_te_r) / ...
                        span + root_chord) * thickness / c_2;
                end
            else
                t(i) = 0;
            end
        else
            if y(i) > -cong
                if y(i) * (-x_le) / span <= x(i) && y(i) * (-x_le)...
                        / span + c_1 >= x(i)
                    t(i) = (x(i) - (-y(i) * x_le / span)) * ...
                        thickness / c_1;
                elseif y(i) * (-x_le) / span + c_1 <= x(i) && ...
                        -y(i) * (x_te_t - x_te_r) / span + x_te_r ...
                        >= x(i)
                    t(i) = thickness;
                elseif -y(i) * (x_te_t - x_te_r) / span + x_te_r ...
                        <= x(i) && -y(i) * (x_te_t - x_te_r) / ...
                        span + root_chord >= x(i)
                    t(i) = (- x(i) - y(i) * (x_te_t - x_te_r) / ...
                        span + root_chord) * thickness / c_2;
                end
            elseif -span <= y(i) && y(i) <= -cong
                if y(i) * (-x_le) / span <= x(i) && -y(i) * ...
                        ((x_te_t - x_te_r) / c_2 + x_le / c_1) / ...
                        (span * (1 / c_1 + 1 / c_2)) + root_chord ...
                        / ((c_2 / c_1) + 1) >= x(i)
                    t(i) = (x(i) - (-y(i) * x_le / span)) * ...
                        thickness / c_1;
                elseif -y(i) * ((x_te_t - x_te_r) / c_2 + x_le / ...
                        c_1) / (span * (1 / c_1 + 1 / c_2)) + ...
                        root_chord / ((c_2 / c_1) + 1) <= x(i) && ...
                        -y(i) * (x_te_t - x_te_r) / span + ...
                        root_chord >= x(i)
                    t(i) = (- x(i) - y(i) * (x_te_t - x_te_r) / ...
                        span + root_chord) * thickness / c_2;
                end
            else
                t(i) = 0;
            end
        end
```

```
    end
end


function d2f = d2_elemental_area(x,theta,beta)
    [root_chord,~,span,~,~,~,~,~,~,~,~,boattail_length,d_exit,~,...
    diameter] = setup_variables;
    elemental_area = @(x,theta_p,beta) 2*integral(@(y) ...
        thickness_wing(x + beta * y * cos(theta_p), y), -span, span);
    f = zeros(1,length(x));
    for i = 1:length(x)
        f(i) = elemental_area(x(i),theta,beta);
    end
    df = gradient(f,x);
    d2f = gradient(df,x);
    for i = 1:length(x)
        if x(i) > root_chord && x(i) <= root_chord + boattail_length
            d2f(i) = d2f(i) + 0.5 * pi * ...
                ((d_exit - diameter) / boattail_length) ^ 2;
        end
    end
end


function res = operation(x_1,x_2,theta,beta,n_int)
    [root_chord,~,~,~,~,~,~,~,~,~,~,boattail_length] = ...
        setup_variables;
    x_interp = linspace(0,root_chord + boattail_length,n_int);
    y_interp = d2_elemental_area(x_interp,theta,beta);
    d2f_int = @(x) interp1(x_interp,y_interp,x);
    integrand = @(x_1,x_2) d2f_int(x_1).*d2f_int(x_2).*...
        log(abs(x_1 - x_2));
    res = integrand(x_1,x_2);
    for i = 1:length(res(:,1))
        for j = 1:length(res(1,:))
            if isnan(res(i,j)) || isinf(res(i,j))
                res(i,j) = 0;
            end
        end
    end
end


function res = final_integ(theta,beta)
    [root_chord,~,span,~,~,~,~,~,~,~,~,boattail_length] ...
        = setup_variables;
    res = zeros(1,length(theta));
    x_integ = linspace(-beta*span,max(beta*span + ...
        root_chord,root_chord + boattail_length),1000);
```

```
    [X_1,X_2] = meshgrid(x_integ, x_integ);
    for i = 1:length(theta)
        F = operation(X_1,X_2,theta(i),beta,300);
        res(i) = trapz(x_integ,trapz(x_integ,F,2));
    end
end
```

Bibliography

[1] R. Arina. *Fondamenti di Aerodinamica*. Torino, Italy: Levrotto & Bella, 2015. ISBN: 978-8-882-18187-1 (cit. on pp. 1, 2, 30, 31).

[2] J. D. Anderson. *Fundamentals of Aerodynamics*. New York, U.S.A.: McGraw-Hill Education, 2017. ISBN: 978-1-259-12991-9 (cit. on pp. 1, 2).

[3] L. M. Sheppard. *Methods for Determining the Wave Drag of Non-Lifting Wing-Body Combinations*. R.A.E. Report Aero 2590. A.R.C. 19,320. Aeronautical Research Council, Apr. 1957 (cit. on pp. 8, 9, 11, 12, 18).

[4] R. T. Jones. *Theory of wing-body drag at supersonic speeds*. N.A.C.A. Technical Report 1284. National Advisory Committee for Aeronautics, Jan. 1956 (cit. on p. 8).

[5] B. S. Baldwin and R. R. Dickley. *Application of wing-body theory to drag reduction at low supersonic speeds*. N.A.C.A. Research Memo A54J19. National Advisory Committee for Aeronautics, Jan. 1955 (cit. on p. 8).

[6] G. N. Ward. *The drag of source distributions in linearized supersonic flow*. Report 88. A.R.C. 17,478. College of Aeronautics, Feb. 1955 (cit. on p. 8).

[7] L. M. Sheppard. *The Wave Drag of Non Lifting Combinations of Thin Wings and 'Non-Slender' Bodies*. R.A.E. Report Aero 2496. A.R.C. 19,319. Aeronautical Research Council, Mar. 1957 (cit. on pp. 9, 11).

[8] L. L. Levy Jr. *Supersonic and Moment-of-Area Rules Combined for Rapid Zero-Lift Wave-Drag Calculations*. NASA Memorandum 4-19-59A. National Aeronautics and Space Administration, June 1959 (cit. on p. 9).