PoliTo Rocket Team

POLITECNICO DI TORINO

Technical Report

Cd Model - Barrowman Method

Code:			VES-MSA-004-V1
Subteam:	VITTORIO EMANUELE II	**Division:**	MISSION ANALYSIS
Author(s):	Andrea PANTANO	Mission Analyst	05-11-2023
Verified by:	Mattia SABA	MSA Lead Engineer	13-12-2023
Approved by:	Edoardo VIGLIETTI	VES Chief Engineer	06-01-2024

Disclosure

PU	Public	
DT	Disclosable within the Team	
DS	Disclosable within the Subteam	
DD	Disclosable within the Division	•
RL	Reserved to the Leads	

Contents

List of figures

1 Introduction

The purpose of this document is to describe the Barrowman method for the practical calculation of the drag coefficient of the rocket, highlighting the positive and negative sides of the application of this method.

The Barrowman method was introduced by him in his Master's thesis. [1] All the formulas that can be used for the computation will be reported, specifying the meaning of every variable and making some considerations on all the terms, both the one that we know or that we will have to investigate more in the future.

The Barrowman method is a semi-empirical method. All the experimental data used to provide correction factors for the supersonic and transonic flow partially make up for a set of inadequate hypothesis. Despite being theoretically faulty, the Barrowman method has been proven really effective.

The C_D is always expressed as function of A_r, the reference area. The numerical value is strongly influenced by the choice of this area, so it's important to set a reference area and always use the same one when calculating the different coefficients and so the drag force.
Our **Matlab** scripts use the cross section of the body tube as the reference area.

2 Contents

2.1 Hypothesis of the Method

The Barrowman method is based on four hypothesis:

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

The third and fourth hypothesis are probably acceptable, even if the nosecone of transonic and supersonic rockets is usually not so pointy. Further considerations are to be made when choosing the shape of the nosecone.

The first hypothesis is generally valid during the flight, when the rocket velocity is far bigger than the external flow velocity (e.g. wind velocity). However, it is not respected during the first moments of the flight during which the wind velocity can be relevant with respect to the relatively low velocity of the rocket.
Considering an hypothetical wind velocity of 4 m/s, a rail exit velocity of 30 m/s would result in a 8° angle of attack, which is not negligible. Using this method, the computed C_D will probably be inaccurate for the first second of flight.

The second hypothesis is clearly faulty, because of the presence of wind shear, and of the turbulent nature of the atmosphere. Futhermore the flow around the rocket body may be not irrotational depending on the geometry of the rocket itself.

Please note that the hypothesis of this method, because of it being semi-empirical, are not really determinant for a valid result. You will see that most of the coefficients are calculated based on experimental parameters, and a lot of corrective empirical factors are introduced. The application of this method has proved to be effective multiple times in rocket and missile engineering.

2.2 Strategy Adopted by the Barrowman Method

The drag force acting on a body in a flow in motion has two components. The first one is **friction drag**, that is caused by the roughness of the surface, and in general by energy dispersion due to the interaction between the surface and the flow particles. Heat exchange is also a factor in some of the equations used to compute friction drag.

The second component is **pressure drag**, which is caused by an uneven pressure distribution on the body, caused by its shape.

The Barrowman method consists in treating separately this two types of drag, and dividing the rocket into different components: the nosecone, the body and the fins. After calculating the pressure and friction drag coefficients for each of these parts, we can combine them exploiting linearity, getting the final value of C_D.

This method is really easy and effective to apply to subsonic, incompressible flow. It's relatively easy to correct it for subsonic, compressible flow. The harder part is to apply corrections for supersonic flow.

In this case, we need to consider an additional form of drag known as **wave drag**. In fact when travelling at supersonic velocities through a flow, shock waves are generated and therefore the flow field around the rocket changes accordingly. There are some approximated equations that should be easy to implement for the supersonic friction drag, but computing the supersonic pressure drag is not as simple.

2.3 Friction Drag

2.3.1 Friction Coefficient

The friction drag coefficient is defined as:

$$C_f = \frac{D_f}{\frac{1}{2}\rho {V_\infty}^2 A_W} \tag{1}$$

Where D_f is the friction drag force, ρ is the fluid density, V_∞ is the freestream flow velocity and A_W is the wetted area of the body. The value of this coefficient can be obtained using some empirical formulas. The correct formula must be used based on a critical Reynolds number, defined as:

$$Re_{cr} = 51\left(\frac{R_s}{L_r}\right)^{-1.039} \tag{2}$$

Where Re is the Reynolds number, R_s is the roughness of the surface and L_r is the reference length. Keep in mind that $Re = \frac{\rho U L_r}{\mu}$.
For subsonic incompressible flow:

$$\begin{cases} Re < 5 \times 10^5 \\ C_f = \frac{1.328}{\sqrt{Re}} \end{cases} \tag{3}$$

When the Reynolds number increases, the flow around the body ceases to be laminar. Up to the critic Reynolds number, we need to take in account the transition to turbulent flow. The following relation proved to be effective:

$$\begin{cases} 5 \times 10^5 < Re < Re_{cr} \\ C_f = \frac{1}{(3.46 \log Re - 5.6)^2} - \frac{1700}{Re} \end{cases} \tag{4}$$

If the Reynolds continues increasing and passes the critical value, this last formula shall be used:

$$\begin{cases} Re > Re_{cr} \\ C_f = 0.032 \left(\frac{R_s}{L_r}\right)^2 \end{cases} \tag{5}$$

These relations are valid for subsonic, incompressible flow. We need to introduce some corrections in order to use them for compressible flows.

2.3.2 Corrected Friction Coefficients

After computing the "baseline" C_f, which could already be considered good enough for low mach numbers (up to $M = 0.3$), we can introduce some corrective coefficients to improve the results for subsonic compressible flow, and for supersonic flow.
For **subsonic, compressible** flow:

- If we are in presence of roughness, $C_{f_C} = C_f(1 - 0.12M^2)$
- For "smooth" surfaces, $C_{f_C} = C_f(1 - 0.09M^2)$

Some "practical" considerations on the corrections above need to be made. It is not easy to establish when we can consider a surface smooth or rough. We can see that the two formulas differ by a small factor so if it's not clear which one to use, it is probably better to use the first one and be conservative. At first sight, it seems these formulas could give negative C_D values, but we need to keep in mind that they are valid for **subsonic** flow, so $M < 1 \longrightarrow C_D > 0$, as we want.

For **supersonic flow**:

- If the supersonic flow is **laminar**:

$$C_{f_C} = \frac{C_f}{(1 + 0.045M^2)^{0.25}} \tag{6}$$

Note that supersonic, laminar flow is a rare condition to face in our applications, in particular in cases of rough paintings of the rocket. For this reason, if the rocket goes supersonic, the air flow is expected to be turbulent.
The possibility of a supersonic laminar flow exists, we will need instructions from the FEA Division to choose which corrective factor should be used.

- If we are in conditions of **”smooth” turbulence**:

$$C_{f_C} = \frac{C_f}{\left(1+\kappa M^2\right)^{0.58}} \tag{7}$$

  In these conditions, heat exchange becomes a relevant factor. The coefficient κ introduces a correction for that phenomenon. If the walls are not actively cooled, $\kappa = 0.15$. If the walls are kept at ambient temperature, $\kappa = 0.0512$. We are interested in the first of these two values, since the skin of a rocket is not usually cooled. Developing a system to cool the walls and therefore reduce the drag could be an idea, but is probably not worth the effort and time.

- For **turbulent flow, in case of a rough surface**:

$$C_{f_C} = \frac{C_f}{1+0.18M^2} \tag{8}$$

Please note that the value calculated ”with roughness” should always be bigger than the ”smooth” one. Just to be conservative and add a layer of control to simulators, it could be a good idea to just use the bigger between these two values in case of a supersonic flow. The code would also turn out to be more simple.

2.3.3 Friction C_D Component Calculation from C_{f_C}

Let’s call C_{D_f} the component of the drag coefficient due to friction. For a general surface with a wetted area A_W:

$$C_{D_f} = C_{f_C}\left(\frac{A_W}{A_r}\right) \tag{9}$$

We can see here the importance of the reference area, as anticipated in the introductions of this report. We can apply it for the bodytube and the fins set.

- Application to the tail, or finset:

$$C_{D_{f_t}} = 2NC_{f_C}\left(\frac{A_T}{A_r}\right) \tag{10}$$

  Where N is the number of fins and A_T is the planform area of a single fin. Each fin counts for two planform surfaces, an upper and a lower one, that’s why the coefficient 2 is in the equation.

- Application to the body:

$$C_{D f_b} = \left[1 + \frac{0.5}{f_b}\right] C_{f_C} \left(\frac{A_{W b}}{A_r}\right) \tag{11}$$

Where f_b is the *fineness ratio* of the body, and it's equal to his length divided by the maximum width. This term is sometimes called *slender ratio* in rocketry engineering reports.
We can include the nosecone in the body and use the total wetted area to compute a single coefficient, or treat the nosecone separately using the same equation. This equation is linear, so it won't make any difference.

2.4 Pressure Drag

2.4.1 Fins Pressure Drag

Regarding the pressure drag acting on the fins, we need to highlight three components. The first is generated by the leading edge, the second one is caused by the thickness of the fin itself, and last one is due to the thick trailing edge.

2.4.2 Fins Leading Edge Generated Drag

Base drag is the force opposed to motion caused by a low pressure in the aft part of a rocket. Let's define ΔC_D as the difference between total drag and base drag. Empirical data proves that we can calculate this value using one of the following formulas:

$$\begin{cases} M < 0.9 \\ \Delta C_D = \left(1 - M^2\right)^{-0.417} - 1 \end{cases} \tag{12}$$

$$\begin{cases} 0.9 < M < 1 \\ 1 - 1.5(M - 0.9) \end{cases} \tag{13}$$

$$\begin{cases} M > 1 \\ \Delta C_D = 1.214 - \frac{0.502}{M^2} + \frac{0.1095}{M^4} + \frac{0.0231}{M^6} \end{cases} \tag{14}$$

This expression will later be needed.

Let's define A_{LE}, the cross section of the leading edge. We also need the length of the exposed leading edge, that can also be obtained using the fin span, S, and the leading edge sweepback angle, Γ_L.

$$\frac{S}{\cos \Gamma_L} \tag{15}$$

Barrowman proves that the pressure drag coefficient due to the leading edge is:

$$C_{DLT} = 2N\left(\frac{Sr_L}{A_r}\right)\cos^2\Gamma_L\left(\Delta C_D\right) \tag{16}$$

Where r_L is the radius of the rounded leading edge. In case of a flat leading edge, r_L is half the thickness.

2.4.3 Trailing Edge Generated Base Drag

For **subsonic, incompressible flow** the trailing edge base drag coefficient can be calculated as:

$$C_{DBT} = \frac{0.135}{\sqrt[3]{C_{f_B}}} \tag{17}$$

Where C_{f_B} is a function of the friction drag coefficient:

$$C_{f_B} = 2C_f\left(\frac{c_r}{h_r}\right) \tag{18}$$

With c_r is fin root chord and h_r is the fin trailing edge thickness at its root.
We can correct it for **subsonic, compressible flow**:

$$C_{DBT} = \frac{0.135 \times N\left(\frac{A_{Bf}}{A_r}\right)}{\sqrt[3]{C_{f_B}}\sqrt{K - M^2\cos^2\Gamma_c}} \tag{19}$$

Where:

$$K = \cos^2\Gamma_c + \frac{\left[0.223 + 4.02 \times C^*_{f_C}\left(\frac{t_r}{h_r}\right)^2\right]^2}{\left[\left(\frac{c_r}{h_r}\right)C^*_{f_C}\right]^{\frac{2}{3}}} \tag{20}$$

For **supersonic flow**:

$$C_{DBT} = \frac{N \times \left(1 - 0.52 \times M^{-1.19}\right)\left(\frac{A_{Bf}}{A_r}\right)}{\left[1 + 18C_{f_C}\left(\frac{t_r}{h_r}\right)^2\right]M^2} \tag{21}$$

2.4.4 Thickness Pressure Drag Coefficient

For **subsonic, compressible flow** on a three dimensional fin, Barrowman proposes a corrected formula, obtained using the corrected friction coefficient and introducing the Prandtl corrective factor. The equation is then manipulated to eliminate a singularity at $M = 1$. The mathematical passages are not relevant for the implementation on a simulator, you can find them in the original Barrowman report.

$$C_{DTT} = 4NC_{f_C}\left(\frac{A_T}{A_r}\right)\left[\left(\frac{t_r}{c_r}\right)\cos\Gamma_c + \frac{30\left(\frac{t_r}{c_r}\right)^4\cos^2\Gamma_c}{\left(\sqrt[2]{K - M^2\cos^2\Gamma_c}\right)^3}\right] \tag{22}$$

Where t_r is the maximum fin thickness, Γ_c is the sweepback angle of the fins and K is a corrective factor:

$$K = \cos^2\Gamma_c + \left[\frac{\frac{C^*_{DTT}}{C^*_{f_C}}\left(\frac{A_r}{A_T}\right) - 4\left(\frac{t_r}{c_r}\right)\cos\Gamma_c}{120\left(\frac{t_r}{c_r}\right)^4\cos^2\Gamma_c}\right]^{\frac{2}{3}} \tag{23}$$

Where the "*" indicates the coefficients calculated at $M = 1$. In particular, an interpolation for C^*_{DTT} is the following:

$$C^*_{DTT} = 1.15\left(\frac{t_r}{c_r}\right)^{\frac{5}{3}}\left[1.61 + \xi - \sqrt{(\xi - 1.43)^2 + 0.578}\right] \tag{24}$$

$$\xi = (A.R.)\frac{t_r}{c_r}^{\frac{1}{3}} \tag{25}$$

With $A.R.$ aspect ratio of the exposed fin, defined as the square of the span divided the planform area.

2.5 Fins Wave Drag Coefficient

Barrowman, in the appendix of his thesis, proposes a numerical method to compute the wave drag coefficient for the fins. As previously said, this is needed to take into account the interactions between the supersonic shock wave and the fins. First of all, we need to introduce the coefficient of pressure according to the Busemann theory:

$$C_{Pi} = K_1\eta_i + K_2\eta_i^2 + K_3\eta_i^3 - k^*\eta_L^3 \tag{26}$$

Where:

$$K_1 = \frac{2}{\beta} \tag{27}$$

$$K_2 = \frac{(\gamma+1)M^4 - 4M^2}{4\beta^4} \tag{28}$$

$$K_3 = \frac{(\gamma+1)M^8 + (2\gamma^2 - 7\gamma - 5)M^6 + 10(\gamma-1)M^4 - 12M^2 + 8}{6\beta^7} \tag{29}$$

η_i is the inclination of a generic surface region relatively to the flow, this will be further clear as we proceed with this method; η_L is the inclination of the leading edge surface and β is defined as follows :

$$\beta = \sqrt{(M^2 - 1)} \tag{30}$$

The value of k^*, if the surface has been preceded by a single compressive shock wave (as it should be in this case, a single compression shock wave at the nosecone tip - there's of course another shock wave at the root chord leading edge of each fin but those doesn't count in this coefficient evaluation), is the following:

$$K^* = \frac{(\gamma+1)M^4\left[(5-3\gamma)M^4+4(\gamma-3)M^2+8\right]}{48} \tag{31}$$

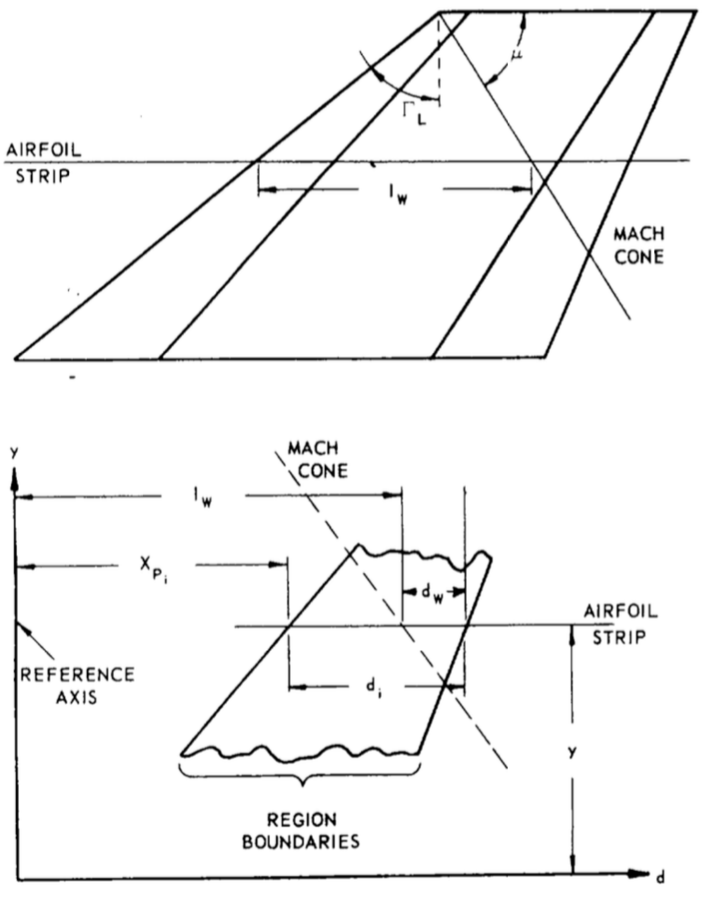

Figure 1: The Geometry of a Fin with the Mach Cone at the Tip

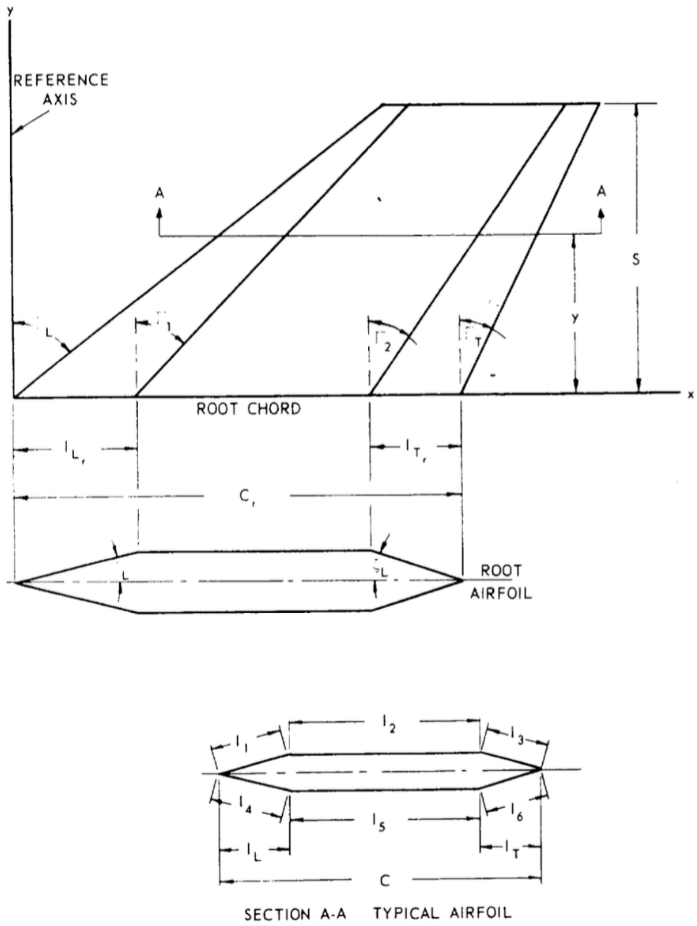

Figure 2: A Typical Fin Hexagonal Airfoil, Useful for a Practical Example

If the region upstream of the fins has only contained expansion fans, $K^* = 0$. The geometry showed in figures 1 and 2 will be used as an example.

You can see that the profile can be divided into 6 surfaces, defined by the segments $l_1, l_2, ..., l_6$. To conduct the analysis, this surfaces will be divided in span-wise strips of width Δy. The wave drag of each strip (on a single surface) is calculated as:

$$F_{di} = C_{p_i} n_i \left(1 - \frac{r_i}{2}\right) \tag{32}$$

Where n_i is the component of the profile segment l_i which is normal to the flow. Just as an example:

$$n_1 = l_1 \sin \eta_1 \tag{33}$$

Where η_1 is the inclination of the surface. To understand what r_i is, we just need to take a closer look at figure 1: it is the the ratio $\frac{d_w}{d_i}$ between the portion of the strip covered by the mach cone and the entire length of the strip. It is needed to do this calculation for each of the 6 (or whatever the number of the surfaces is) surfaces. The wave drag force of the j-th strip will be:

$$F_{dj} = \sum_{i=1}^{6} F_{di} \tag{34}$$

If we have k strips (we want our code to divide our fin span in many strips, k should be a sufficiently big number), the total force on the fin is:

$$F_d = \sum_{j=1}^{k} F_{dj} \tag{35}$$

To compute the drag wave coefficient:

$$C_{DW} = \frac{NF_d \cdot \Delta y}{A_r} \tag{36}$$

Where N is number of fins, A_r is the reference area and Δy is the width of a strip, obtained as the total span divided by the number of strips k.
Note that this method is a quite simple one to implement and to be used. The fundamental assumption is that the center of pressure of each small strip is at 50% of its chord length, because it is travelling at supersonic speeds. This is a result of thin airfoils theory.

The Mach cone originating at the tip of the leading edge is probably a result of the theory that Barrowman uses for this computation. Why it is located as described above, is still unclear.

2.5.1 Trailing Edge Generated Base Drag

For **subsonic, incompressible flow** the trailing edge base drag coefficient can be calculated as:

$$C_{DBT} = \frac{0.135}{\sqrt[3]{C_{fB}}} \tag{37}$$

Where C_{fB} is a function of the friction drag coefficient:

$$C_{fB} = 2C_f \left(\frac{c_r}{h_r}\right) \tag{38}$$

With c_r is fin root chord and h_r is the fin trailing edge thickness at its root.
We can correct it for **subsonic, compressible flow**:

$$C_{DBT} = \frac{0.135 \times N \left(\frac{A_{Bf}}{A_r}\right)}{\sqrt[3]{C_{fB}}\sqrt{K - M^2 \cos^2 \Gamma_c}} \tag{39}$$

Where:

$$K = \cos^2 \Gamma_c + \frac{\left[0.223 + 4.02 \times C_{f_C}^* \left(\frac{t_r}{h_r}\right)^2\right]^2}{\left[\left(\frac{c_r}{h_r}\right) C_{f_C}^*\right]^{\frac{2}{3}}} \tag{40}$$

For **supersonic flow**:

$$C_{DBT} = \frac{N \times \left(1 - 0.52 \times M^{-1.19}\right)\left(\frac{A_{Bf}}{A_r}\right)}{\left[1 + 18 C_{f_C}\left(\frac{t_r}{h_r}\right)^2\right] M^2} \tag{41}$$

Overall drag of the tail for subsonic flow:

$$(C_D)_T = C_{Df_T} + C_{DLT} + C_{DBT} + C_{DTT} \tag{42}$$

making sure to use the corrected values for incompressible or compressible flow.

Overall drag of the tail for supersonic flow:

$$(C_D)_T = C_{Df_T} + C_{DLT} + C_{DBT} + C_{DWT} \tag{43}$$

The last term is the wave drag coefficient. Wave drag is caused by the interactions between the supersonic shock wave and the rocket, computed in appendix A of the original Barrowman report. In this appendix, Barrowman introduces a numerical method to solve the problem and the solution is highly dependent on the geometry. Despite that, it shouldn't be too hard to apply the method once the fins geometry is defined.

2.6 Body pressure drag

For **subsonic, incompressible flow**, the pressure drag coefficient is:

$$C_{DP} = \frac{6 \times A_{WB} C_{f_C}}{f_B^3 A_r \left(K - M^2\right)^{0.6}} \tag{44}$$

If the nose cone is a ogive:

$$K_{ogive} = 1 + \left[\frac{6.82 \times A_{WB} C_{f_C}^* \left(f_N + 1\right)^{2.22}}{f_B^3 A_r}\right]^{\frac{5}{3}} \tag{45}$$

Where f_N and f_B are the fineness ratios of the nose and the body.

The problems begin when we reach supersonic regime due to the fact that Barrowman suggests to compute the new coefficient using the second-order shock-wave expansion method.
The nosecone is approximated with various segments each one of them tangent to the actual profile. The pressure function is then evaluated and integrated around all the outer surface of the nosecone.

This part is the most complex calculation needed to apply this method.

Here is a brief description of how the method works, even if some details still need some studies. We need to first compute the pressure distribution, to get the final coefficient by integration:

$$C_{DP} = \frac{4}{\gamma M^2} \int_0^L \int_0^\eta P \sin\delta \, d\delta \, dx \tag{46}$$

This equation is not clearly readable from the Barrowman report, so another approach has been taken, calculating the drag coefficient using its definition. Since we will be using conic sections to approximate the nosecone, let's keep in mind that the surface with half-angle $theta$ (which is constant) is:

$$\Omega = \int_{x_A}^{x_B} 2\pi x \sin\theta, dx \tag{47}$$

The drag force generated by the pressure is therefore:

$$F_x = \int_{x_A}^{x_B} 2\pi P(x) \sin^2\theta x, dx \tag{48}$$

And the drag coefficient:

$$C_{DP} = \frac{F_x}{\frac{1}{2}\rho V_\infty{}^2 A_{ref}} \tag{49}$$

Now the shock wave expansion method should come in handy. First, we need to approximate the nosecone of the rocket as multiple tangent cones with each conical section called *frustrum* from now on. Therefore a *Matlab* script has been created to do this and a good approximation has been made by choosing appropriate points on the surface of the nosecone, called *tangent nodes*. The proper nodes have been obtained by intersecating the tangent lines in this tangent nodes, obtaining some segments that define each frustrum. The script has been reported in the appendix section of this report and a figure of that approximation is presented below.

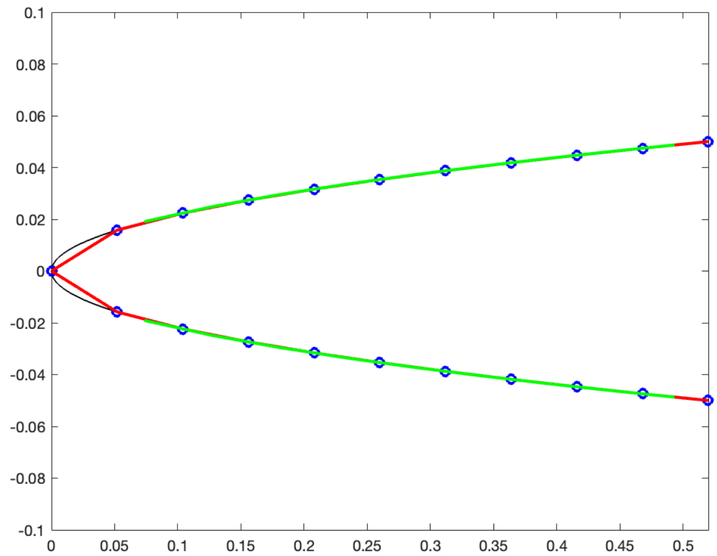

Figure 3: Result of the Nosecone Approximation

This scripts selects the tangent nodes by using constant interval in the gradient, this is possible because the gradient of the nosecone shape function is injective (each y value is associated with one x value only, meaning the curve can not reach a maximum and then come back down for example, with this method we would badly approximate the curve around the gradient maximum).
The tip and the trailing edge are forced to be nodes.

After finding the tangent nodes on the nosecone curve, we tried two possible methods to obtain the final approximated curve.

The first one is to just join the tangent nodes. This avoids any numerical cancellation problem, works with any number of desired segments and with any curve.
The second method is to conduct the tangent in two consequent tangent nodes and to use the intersections between those tangents as our final nodes.

This last method is more precise, but gives numerical cancellation problems with some curves, such as the 1/2 power series. It works better with the Von Karman nose curve.

In general, it's probably better to use the first method, because the loss in precision is not really relevant and we can always increase the number of segments to better approximate the nosecone curve.

Let's call each frustrum with a letter (A,B,C...) and let's indicate, for each frustrum: $()_2$ all the values referring to the leading surface of the current frustrum, $()_3$ all the variables referred to the trailing surface of the current frustrum, $()_1$ all the variables referred to the trailing edge of a previous frustrum. This means that the values in the point 3 of a frustrum is the same value indicated as 1 for the following frustrum. Barrowman then suggests an approximate solution for the pressure distribution, valid on each frustrum:

$$P = P_C - (P_C - P_2)e^{-\eta} \tag{50}$$

Where P_C is the pressure on a cone travelling at the same speed of the rocket, with the same length as the nosecone, but with a constant slope (as it's been deducted by the report since this passage is not very clear on the report). This equivalent pressure could be calculated for each frustrum using the Taylor-Maccoll approximation. A *Matlab* package to do this is available.

Let's now define η as follows :

$$\eta = \left(\frac{\delta p}{\delta s}\right)_2 \frac{x - x_2}{(P_C - P_2)\cos\delta_2} \tag{51}$$

Where s is the surface coordinate and δ is the semi-aperture of the cone. In particular:

$$\left(\frac{\delta p}{\delta s}\right)_2 = \frac{B_2}{r}\left(\frac{\Omega_1}{\Omega_2}\sin\delta_1 - \sin\delta_2\right) + \frac{B_2}{B_1}\frac{\Omega_1}{\Omega_2}\left(\frac{\delta p}{\delta s}\right)_1 \tag{52}$$

The coefficients B and Ω are:

$$B = \frac{\gamma M^2}{2\,(M^2 - 1)} \tag{53}$$

Not that this value is particularly critical when M is close to one.

$$\Omega = \frac{1}{M}\left[\frac{1 + \left(\frac{\gamma-1}{2}\right)M^2}{\frac{\gamma+1}{2}}\right]^{\frac{\gamma+1}{2(\gamma-1)}} \tag{54}$$

On the other side, Ω becomes more and more relevant when M increases. This variables, seeing equation 52, should change from one frustrum to the other.
The problem is that Barrowman does not suggest how to calculate the change in Mach number at each section so the team decided to use the expansion waves formulas.
Given the ν function:

$$\nu(M) = \left[\frac{\sqrt{\gamma+1}}{\gamma-1}\arctan\left(\sqrt{\frac{\gamma-1}{\gamma+1}}\sqrt{M^2-1}\right) - \arctan\left(\sqrt{M^2-1}\right)\right] \tag{55}$$

Then we apply the expansion relation:

$$\nu(M_2) = \theta_2 + \nu(M_1) \tag{56}$$

Where θ_2 is the deflection angle, and can be calculated as the difference between the half-cone angles of the two frustrums. From the ν function the Mach number behind the shockwave can be obtained using the Newton method to solve non linear functions. This method is better defined as a ”root-finding algorithm”.

Let’s continue with this method:

$$\left(\frac{\delta p}{\delta s}\right)_3 = \frac{P_C - P_3}{P_C - P_2}\left(\frac{\delta p}{\delta s}\right)_2 \tag{57}$$

That means we can proceed in the following way:

- For the first frustrum (A), we can immediately get the pressure, because it is in fact a cone, so $P_C = P_2$ and equation 50 just becomes $P = P_C$, constant on the cone surface, as we expect from the Taylor-Maccol theory. We can use a pre-written *Matlab* script for the Taylor-Maccoll equations. It applies for different subroutines to provide the pressure given the freestream Mach number, the half-cone angle and the ratio of specific heats.

- On the second frustrum (B), P_2 is just the constant P of the first frustrum, assuming continuity. As for equation 52, we can note that since pressure on the first frustrum is constant, the second term of the sum just goes to zero. We have the expression that we can substitute in equation 51 and then in 50. Note that we get the P_2 and mach values from the expansion formulas. To find P_3, we can just substitute the coordinates of the trailing surface in 50;

- We can proceed for the third frustrum (C) in the same way, and so on. From now on, the second term of 52 will not be zero, but we can use equation 57 to find all the needed terms. Note that terms $()_1$ of a frustrum are the same as $()_3$ of the previous one.

We have now the pressure piecewise function to be use for the integral in 49.
Results given by this method are not satisfying if the Mach number is close to 1, but they become better if we increase in Mach’s values.

One of the problems is that the ν function give a complex number, and not a real number. This problem has been solved using only the real part of the solution, considering the immaginary part was usually really small. This was due to the fact that the Taylor-Maccoll solver was returning a subsonic value for the flow velocity after the shock which is not expected. Further investigations on the matter are needed. The script used is now presented :

```
for k=1:length(x_t)-1
theta = atan((y_t(k+1)-y_t(k))./(x_t(k+1)-x_t(k)))
if k ==1
    [M0,P0] = taylor_maccoll_solver(M_inf, theta, g, P_inf,...
    T_inf, rho_inf);%P is consta
    Fx = @(x) P0*2*pi*sin(theta)*sin(theta)*x;
    %Cdwl = (4/(g*M_inf^2))*int(P0*sin(theta)*2*pi,x,0,x_t(k+1)) given
    %by Barrowman
    Cdwl = (integral(Fx,0,x_t(k+1)))/(0.5*rho_inf*V_inf^2*A_ref)
    %local Cd, a part of the final integral
    M1 = M0; %mach number on the first frustrum is constant
    dpds1 = 0;
    P2 = P0;
else
    %to find the equivalent pressure on a cone with the same half-angle
    [MC,PC] = taylor_maccoll_solver(M_inf, theta,g, P_inf, T_inf, rho_inf)
    B1 = B(M1);
    nu1 = real(NU(M1));
    O1 = Omega(M1);
    f = @(M) (atan((M^2 - 1)^(1/2)*((g - 1)/(g + 1))^(1/2))*...
    (g + 1)^(1/2))/(g - 1)^(1/2) - atan((M^2 - 1)^(1/2)) ...
    - (thetapr-theta) - nu1;
    df = @(M) (M*((g - 1)/(g + 1))^(1/2)*(g + 1)^(1/2))/...
    ((M^2 - 1)^(1/2)*(((M^2 - 1)*(g - 1))/(g + 1) + 1)*...
    (g - 1)^(1/2)) - 1/(M*(M^2 - 1)^(1/2));
    %getting the mack behind the shockwave using the newton method
    [M2,num_it,M2_vec] = NewtonMethod(M_inf,f,df,-10,0.01,10);
    M2 = real(M2);
    B2 = B(M2);
    nu2 = (thetapr-theta) + nu1;
    O2 = Omega(M2);
    dpds2 = (B2/y_t(k))*((O1/O2)*sin(thetapr)-sin(theta))...
    +(B2/B1)*(O1/O2)*dpds1;
    %eta = dpds2 * (x-x2)/((PC-P2)*cos(theta))
    %pressure distribution on the frustrum
    P = @(x) (PC-(PC-P2)*exp(-dpds2 * (x-x_t(k))/));
    Fx = @(x) 2*pi.*sin(theta).*sin(theta).*x.*(PC-(PC-P2).*...
    exp(-dpds2 .* (x-x_t(k))./((PC-P2).*cos(theta))));
    %drag force on the frustrum
    Cdwl = (integral(Fx,x_t(k),x_t(k+1)))/...
    (0.5*rho_inf*V_inf^2*A_ref)%local cd component
    %updating the variables
```

```
        M1 = M2;
        dpds1 = dpds2;
        P2 = P(x_t(k+1));
    end
    thetapr = theta;
    Cdp = Cdp + Cdwl
end
```

it's interesting to compare the results of this method with the calculations done by *Missile Datcom*.

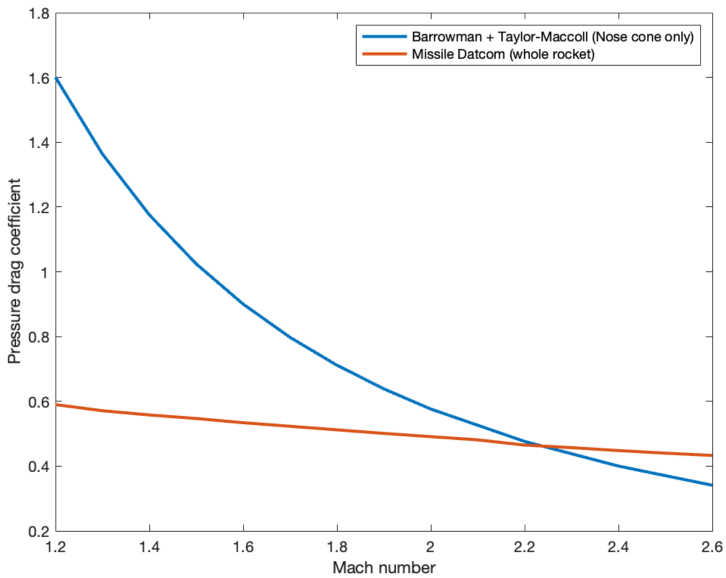

Figure 4: A Comparison between Missile Datcom and the Barrowman Method

We can notice that the two results are far off when the Mach number is close to one. Note that this is only one component of the overall drag, so the blue curve should be definitely lower than the red one. We can see that this is true above Mach 2.3, but this means we can not use this method for our VES simulations, since we won't reach such velocities. If we observe that the red curve is quite constant, we could suppose that a good C_{DP} value could be the asymptotic value of the blue curve. This hypothesis needs to be proven via CFD simulations or experimentally.

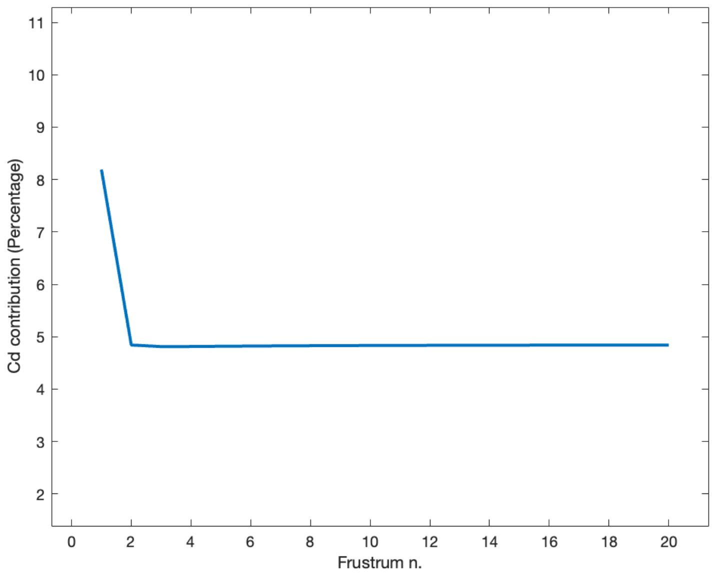

Figure 5: Frustrums Influence on the overall Cdp with $M = 2$ and 20 Frustrums Approximation

From picture 5 we can see that the first frustrum, where the shock wave originates, is the most relevant for the final pressure drag coefficient value, but the other frustrums still play an important role. All the other frustrums have a really similar influence on the final C_d value, and that was to be expected since we approximated the curve using constant slope changes and the ν function takes into account the angle variation.

2.6.1 Shock Wave - Expansion Method

We noticed some major issues in the output of the *Matlab* script above. The most serious one was related to the pressure distribution: with the Mach number increasing, pressure should decrease but we got constant, or slightly increasing pressures. The cause of this problem is still not known, and it is probably related to an incorrect application of Barrowman formulas.
This is really plausible given the fact that the only available Barrowman report is hardly readable in some passages.

Our approach is theoretically simple nonetheless should reveal accurate and useful to the MSA division. However a FEA analysis on the matter can be also important to check the obtained results.

The idea is to keep dividing the nosecone in frustrums, but we significantly increased the number of conical sections. This approximation can be done with same

script we used before.
The Taylor-Maccoll equations can be used to calculate the mach value and the pressure on the first conical section. After that, we use shock wave expansion formulas to calculate the new Mach number and pressure values on each frustrum (considering pressure constant on that frustrum). Once again the Newton Method has been used to get the mach's value from the ν equation presented above. In fact the mach number can not be calculated analitically and the equation can only be solved numerically.

We use the pressure equation for the expansion wave instead of all the Barrowman equations:

$$\frac{p_2}{p_1} = \left(\frac{1 + \frac{\gamma-1}{2} M_1^2}{1 + \frac{\gamma-1}{2} M_2^2} \right)^{\frac{\gamma}{\gamma+1}} \tag{58}$$

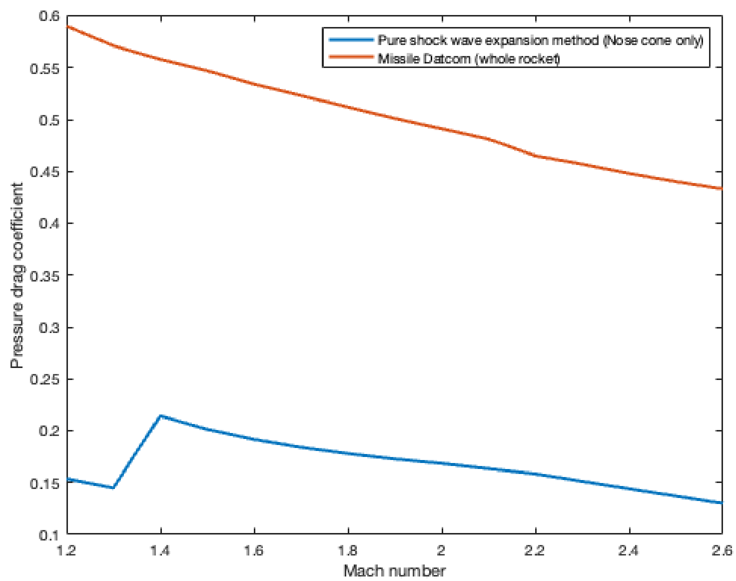

Figure 6: Comparison between our Method and the Missile Datcom output

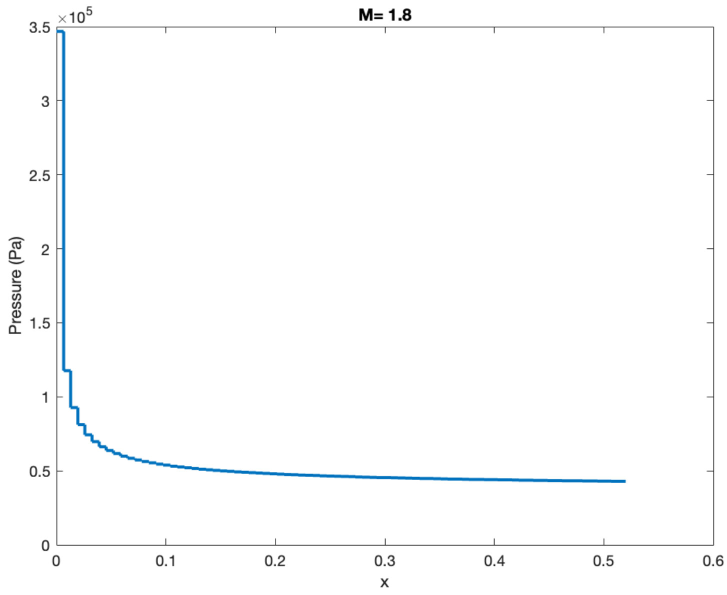

Figure 7: Pressure Distribution along 80 Frustrums at Mach 1.8

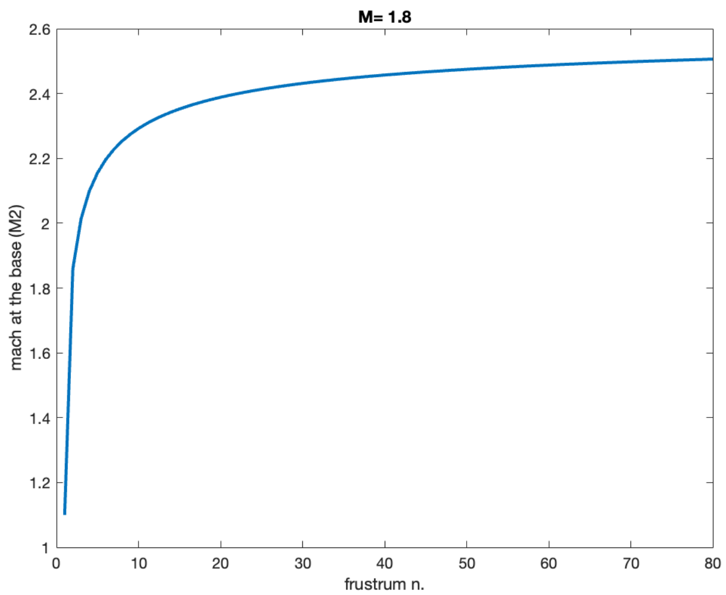

Figure 8: Local Mach Number along each Frustrum at Freestream Mach 1.8

Mach and pressure are now step functions, but increasing the number of frustrums they get closer and closer to a continuous function. Note that in both this script and the previous one, we forced the mach numbers to be always higher than

1.1 after the shock. This was needed because otherwise the expansion method could not be applied. As said before we do not expect a subsonic value behind the shock wave at the tip of the nosecone, being this an oblique shock wave. However for mach numbers slightly above 1 we get subsonic values and for this reason we set for this conditions, a mach number of 1.1.

This is a factor that contributes in making the results for Mach values lower than 1.4 less reliable. Note that for mach values above 1.4 the Taylor-Maccoll solver works as intended.

IMPORTANT: This simple method has the potential to be applied for other useful tasks, and to be used as a baseline for further studies, such as:

- Iterating at a fixed Mach number varying the power series parameter, to optimize the shape of nosecone in terms of drag coefficients.

- Using *Datcom* and/or a CFD simulation, to get the real pressure wave drag of the nosecone, and than introducing a corrective function for our method and the Barrowman one eventually.

- We could try to find a close expression for the nosecone supersonic pressure drag. One idea could be to get the ”corrected” function, as specified in the point above, to interpolate it with a polinomial and to study the relations between the coefficients of the polinomial and the parameter of the power series of the nosecone. If we find a correlation, this could be very useful for the Team and quite a big leap forward in these kind of methods. Nobody has ever found a closed semi-empirical function for this.

2.7 Base drag

For **incompressible flow**, the base drag coefficient of the body is:

$$C_{DB} = \frac{0.029}{\sqrt{C_{f_B}}} \tag{59}$$

With:

$$C_{f_B} = C_f \frac{A_W}{A_B} \tag{60}$$

Where A_B is the base area. As in the previous cases, we can apply the Prandtl correction for subsonic compressible flow, getting:

$$C_{DB} = \frac{0.029\left(\frac{A_B}{A_r}\right)}{\sqrt{C_{fC}\left(\frac{A_W}{A_B}\right)(K - M^2)}} \tag{61}$$

With:

$$K = 1 + \frac{1}{\left[6.38 + 39.7\left(\frac{h_r}{c_r}\right)\right] C_{fC}^*\left(\frac{A_W}{A_B}\right)} \tag{62}$$

To extend this formulas to supersonic flow, we start by defining a critical Mach number:

$$M_{cr} = \frac{0.892}{\sqrt{C_{fB}^*}} \tag{63}$$

Base on the Mach range we are in, we apply one of the following formulas:

$$\begin{cases} 1 \leq M \leq M_{cr} \\ C_{DB} = C_{DB}^*\left[0.88 + 0.12 \times e^{-3.58(M-1)}\right] \end{cases} \tag{64}$$

$$\begin{cases} M > M_{cr} \\ C_{DB} = \frac{0.7\left(\frac{A_B}{A_r}\right)}{M^2} \end{cases} \tag{65}$$

To get the final body pressure drag drag coefficient:

$$(C_{DT})_B = C_{DP} + C_{DB} \tag{66}$$

2.8 Workflow: incompressible, subsonic flow

1. Calculate the Mach number;
2. Calculate the Reynolds;
3. Calculate the critical Reynolds number using 2;
4. Calculate the friction coefficient C_f using either 3, 4, 5;
5. Calculate the body and tail friction coefficients, C_{Df_t} and C_{Df_b} using 10 and 11;
6. Calculate the friction drag coefficient by adding C_{Df_t} and C_{Df_b};
7. Calculate ΔC_D using either 12;

8. Calculate the pressure drag coefficient due to the leading edge, C_{DLT}, using 16;
9. Calculate the trailing edge induced base drag coefficient, C_{DBT}, using 37 and 38;
10. Calculate the thickness drag coefficient C_{DTT} using 22;
11. Calculate the final tail pressure drag by adding the results of points 8, 9, 10 together;
12. Calculate the body pressure drag coefficient C_{DP} using 44 and $K = 1$;
13. Calculate the base drag coefficient using 59;
14. Calculate the final body pressure drag coefficient adding the results of 12 and 13;
15. Calculate the final C_D adding the results of points 6, 11 and 14.

2.9 Workflow: compressible, subsonic flow

1. Calculate the Mach number;
2. Calculate the Reynolds;
3. Calculate the critical Reynolds number using 2;
4. Calculate the friction coefficient C_f using either 3, 4, 5;
5. Calculate the corrected friction coefficient C_{f_C};
6. Calculate the body and tail friction coefficients, C_{Df_t} and C_{Df_b} using 10 and 11;
7. Calculate the friction drag coefficient by adding C_{Df_t} and C_{Df_b};
8. Calculate ΔC_D using 12 or 13;
9. Calculate the pressure drag coefficient due to the leading edge, C_{DLT}, using 16;
10. Calculate the trailing edge induced base drag coefficient, C_{DBT}, using 39 and 40;
11. Calculate the thickness drag coefficient C_{DTT} using 22 and 42;
12. Calculate the final tail pressure drag by adding the results of points 9, 10, 11 together;

13. Calculate the body pressure drag coefficient C_{DP} using 44 and 45;
14. Calculate the base drag coefficient using 61 and 62;
15. Calculate the final body pressure drag coefficient adding the results of 13 and 14;
16. Calculate the final C_D adding the results of points 7, 12 and 15.

2.10 Workflow: supersonic

1. Calculate the Mach number;
2. Calculate the Reynolds;
3. Calculate the critical Reynolds number using 2;
4. Calculate the friction coefficient C_f using either 3, 4, 5;
5. Calculate the corrected friction coefficient C_{f_C} using 6, 7 or 8;
6. Calculate the body and tail friction coefficients, C_{Df_t} and C_{Df_b} using 10 and 11;
7. Calculate the friction drag coefficient by adding C_{Df_t} and C_{Df_b};
8. Calculate ΔC_D using 14;
9. Calculate the pressure drag coefficient due to the leading edge, C_{DLT}, using 16;
10. Calculate the trailing edge induced base drag coefficient, C_{DBT}, using 41;
11. Calculate the wave drag coefficient, using the appendix A of the Barrowman report;
12. Calculate the final tail pressure drag by adding the results of points 9, 10, 11 together;
13. Calculate the body pressure drag coefficient C_{DP} using the second-order wave-expansion method;
14. Calculate the base drag coefficient using 63 and 64 and 66;
15. Calculate the final body pressure drag coefficient adding the results of 13 and 14;
16. Calculate the final C_D adding the results of points 7, 12 and 15.

3 Conclusions

The Barrowman method even if its hypothesis are usually not satisfied, has proven to be effective so there's no valid reason to avoid its usage for a rough estimation of the drag coefficient. It's considered a low-fidelity method and when possible, CFD is to be preferred.

The problem of this method is the difficulty to implement on *Matlab* the algorithms needed for the wave drag calculation and the second-order wave-expansion method but we think that valid results are presented in this document.

4 Appendix

```
    %Nose cone function and tangent approximation
clc
close all
clear all

L = 0.52; %nose length
R = 0.05 ; %nose base radius
C= 0; %haack series parameter, 0 for Von Karman
n = 1/2; %power series parameter
x = linspace(0,L,1000);

%nosecone shape function
%von karman
%y = @(x) (R/sqrt(pi))*sqrt(acos(1-(2.*x)/L)-...
(sin(2.*acos(1-(2.*x)/L)))/2+C*(sin(acos(1-(2.*x)/L))).^3);
%power series
y = @(x) R*(x./L).^n;
ys = y(x);

%slope of the tangent lines
%von karman
%dy_by_dx = @(x) -(R.*((2.*cos(2.*acos(1 - (2.*x)./L)))./...
(L.*(1 - ((2.*x)./L - 1).^2.).^(1/2)) - 2./(L.*...
(1 - ((2.*x)./L - 1).^2).^(1/2)) + (6.*C.*(1 - ...
((2.*x)./L - 1).^2).^(1/2).*((2.*x)./L - 1))./L))./...
(2.*sqrt(pi).*(acos(1 - (2.*x)./L) - sin(2.*...
acos(1 - (2.*x)./L))/2 + C.*(1 - ((2.*x)./L - 1).^2).^(3/2)).^(1/2));
%power series
dy_by_dx = @(x) n*(R/L^n).*x.^(n-1);
y_f = dy_by_dx(x);

plot(x,ys,"k","LineWidth",1)
hold on
plot(x,-ys,"k","LineWidth",1)
axis([0 L -0.1 0.1])

S = 15; % Desidered Segments
N = S + 1; % Number of Nodes
delta = (max(y_f)-min(y_f))/(N); %gradient delta
x_t = zeros(N,1); %x position of the tangent nodes
```

```
y_t = zeros(N,1); %y position of the tangent nodes
x_t(1) = 0; %the tip is set to be a tangent node
y_t(1) = 0;
k = 2;%number of tangent nodes alreay found (tip+base)

for i = 1:length(y_f)

    if  y_f(i) <= max(y_f) - delta && k == 2
        x_t(k) = x(i);
        k = k + 1;

    elseif (y_f(i) <= (dy_by_dx(x_t(k - 1)) - delta)) && (k > 2)
        x_t(k) = x(i);
        k = k + 1;
    end
end

x_t(N) = L;%the base is set to be a tangent node
y_t(N) = R;
y_t = y(x_t);
%to solve numerical cancellation problem that might leave some zeros in tne
%x_t array:
k = 2;
while k<N
    if x_t(k) == 0 %if a 0 is found in the array (in a position different ...
    %from 1, because k starts at 2)
        ns = N-k;%number of nodes to correct
        d = (x_t(N)-x_t(k-1))/(ns+1);%the 0 nodes are substituted by ...
        %equispaced nodes of length d
        for j = 1:ns
            x_t(k+j-1)=x_t(k-1)+d*j;
            y_t(k+j-1) = y(x_t(k+j-1));
        end
        k = N+1;%to break the loop
    end
    k = k+1;
end
%Method 1: join the tangent nodes. Tangent nodes coincident to the
%final nodes. Red segments
plot(x_t,y_t,"bo", "LineWidth",2)
plot(x_t,-y_t, "bo", "LineWidth",2)
for i = 1:S
```

```
    x1 = x_t(i);
    x2 = x_t(i + 1);
    plot([x1;x2], [y(x1);y(x2)],'r','linewidth',2)
    plot([x1;x2], [-y(x1);-y(x2)],'r','linewidth',2)
end

%Method 2: find the final nodes by intersecting the tangents to the
%already found tangent nodes. Green segments.
x_n = zeros(N+1,1); %x position of the final nodes
y_n = zeros(N+1,1); %y position of the final nodes
for j = 1:(length(x_t)-1)
    x1 = x_t(j)
    x2 = x_t(j+1)
    y1 = y_t(j)
    y2 = y_t(j+1)
    m1 = n*(R/L^n)*x1^(n-1)%slopes of the tangents in the tangent nodes
    m2 = dy_by_dx(x2)
    x = (y2-y1+m1*x1-m2*x2)/(m1-m2);
    y = m1*(x-x1)+y1;
    x_n(j+1)=x;
    y_n(j+1)=y;

end
x_n(length(x_t)+1) = L;
y_n(length(y_t)+1) = R;

for k=1:length(x_n)-1
    plot([x_n(k);x_n(k+1)], [y_n(k);y_n(k+1)],'g','linewidth',2)
    plot([x_n(k);x_n(k+1)], [-y_n(k);-y_n(k+1)],'g','linewidth',2)
end
```

References

[1] James S. Barrowman. The practical calculation of the aerodynamic characteristics of slender finned vehicles, 1967.