Students' Space Association
WARSAW UNIVERSITY OF TECHNOLOGY

Coordinator:	Maciej Michałów	Author(s):	Paweł Gołda
Date:	11/01/2021		Błażej Dobroński
Document number:			Szymon Bogus
Number of pages:	25		Wojciech Kowalski

McDrag code implementation in MATLAB

A program for calculating drag coefficient of projectiles

This is an attempt to write a MatLab code, based on R.L. McCoy's original FORTRAN program and documentation, that calculates drag coefficient for standard shape projectiles.

Contents

Abbreviations, designations

CDSF — skin friction drag coefficient
CDBT — boattail drag coefficient
CDB — base drag coefficient
CDH — head drag coefficient
CDRB — rotating band drag coefficient

Abstract

During the process of designing an engineer moves through many different phases. The first one is a conceptual project, then as a plan progresses they meet stage of testing, at first micro scale, then macro scale. All of these stages require to conduct certain calculations to go further. Some of them demand lots of computing power, therefore there is a need to simplify calculations as much as possible while sustaining good precision of results, especially at the beginning of a project when an engineer needs to know just more or less if they are heading into right direction. Time for very precise simulations will come and at early testing only decent accuracy is desired, while time of calculation is crucial.

It is quite common in aerodynamics that many calculations demand enormous amount of time to be proceeded. Such an aerodynamics estimation might be **calculation of a pressure drag coefficient of a projectile or rocket**. The necessity to calculate it quickly was noticed in 70's or even earlier and as a solution Robert L. McCoy created **MC DRAG-** a computer program, written in FORTRAN, which was capable of returning a pressure drag coefficient for a given projectile within seconds.

Today **STUDENT'S SPACE ASSOCIATION(SKA)** finds it beneficial to utilize the power of MC DRAG. We decided not only to understand how MC DRAG works but also to translate an obsolete version of FORTRAN's code into much more perspective and clear MATLAB language.

In this report we described each of an element that MC DRAG algorithm consists of and supported our points with data we obtained with our MATLAB version of MC DRAG. We challenged our version of the program with the same group of projectile for which original MC DRAG already had results. After comparison it is visible that the differences between estimations of both versions are marginal.

1 Drag coefficient components

Overall we can split drag effects into 2 types:

- Pressure drag – arising from pressure forces acting normal to the surface
- Viscous drag – "skin friction" due to viscosity acting tangential to the surface

The simplest way to approach drag coefficient analysis is to break it down into components. Following the split as originally described in McDrag documentation [1]:

$$C_{D_O} = C_{D_H} + C_{D_{BT}} + C_{D_B} + C_{D_{RB}} + C_{D_{SF}} \tag{1}$$

Where:

- C_{D_O} — total drag coefficient at zero angle of attack
- C_{D_H} — pressure drag coefficient due to projectile head (nose)
- $C_{D_{BT}}$ — pressure drag coefficient due to boattail (or flare)
- C_{D_B} — pressure drag coefficient due to the blunt base
- $C_{D_{RB}}$ — pressure drag coefficient due to a rotating band
- $C_{D_{SF}}$ — skin friction drag coefficient due to the entire SF projectile wetted surface (excluding the base)

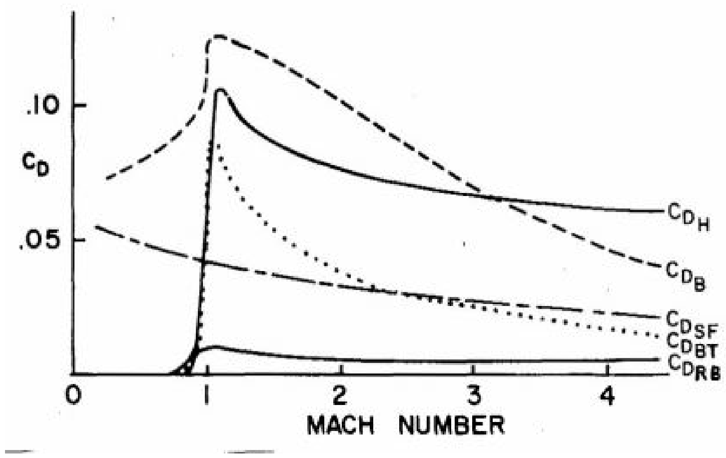


Figure 1: Components of drag coefficients [1]

The behavior of all the above components of drag is strongly dependent on free stream Mach number. The skin friction drag and the base drag depend on Reynolds number as well. Free stream Mach number is Mach number of a body measured in free stream, unaffected by the body itself and it is an idealistic value as opposed to local flow Mach number [1].

2 Projectile dimensions

The dimensions as used in this documentation, names and graphical description, are depicted in Figure 2. **Unless stated otherwise, all the dimensions need to be inputted in *calibers* as the length unit.** To calculate length in *calibers* simply divide the value by bullet diameter (*caliber*). List of

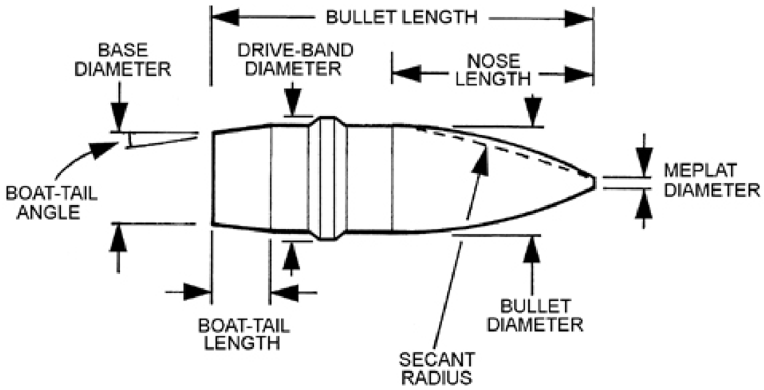


Figure 2: Projectile Dimensions [2]

dimensions required by the model:

- L_T = total length of the projectile [BULLET LENGTH]
- d_{REF} = reference diameter of the projectile in mm (that is milimeters) [BULLET DIAMETER]
- L_N = length of the nosecone [NOSECONE LENGTH]
- L_{cyl}- cylinder length [BULLET LENGTH - NOSE LENGTH]
- d_b- base diameter
- L_{BT} - lenght of the boattail [BOAT-TAIL LENGTH]
- β - boattail angle
- d_M - meplat diameter
- d_{rb} - rotating band diameter [DRIVE-BAND DIAMETER]
- $\frac{R_T}{R}$ = headshape [see subsection 2.1]

2.1 Nosecone shape or headshape

Shape of the nosecone is defined by parameter $\frac{R_T}{R}$, where:

- R = nosecone ogive radius
- R_T = equivalent tangent nosecone radius for the same nosecone length

R_T can be calculated based on the below formula:

$$R_T = L_N^2 + \frac{1}{4} \tag{2}$$

R_T is an ogive radius of a tangent nosecone as presented on figure 3. In other words, the bullet side is tangent to the circle based on radius R_T.

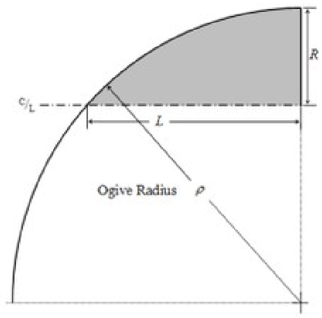


Figure 3: Tangent ogive

Naturally for a tangent nosecone the value of parameter $\frac{R_T}{R} = 1$, since $R_T = R$.

For a conical nosecone the equivalent tangent nosecone radius R_T is still defined by equation 2 but since there is no curvature we can assume the nosecone ogive radius R approaches infinity. Hence $\frac{R_T}{R} = 0$

For a secant nosecone, as presented on figure 4, we need to find the actual radius of a circle that defines the shape of the nosecone R (described on figure 4 as ρ) to calculate $\frac{R_T}{R}$. The most common shape of a nosecone is a secant nosecone with $\frac{R_T}{R} = 0.5$ which is essentially a minimum drag nose shape at supersonic speeds [1].

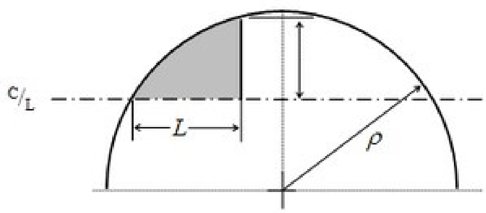


Figure 4: Secant ogive

3 Model limitations

These model as created by McCoy is assumed to be valid over a Mach number range of 0.5 to 5 and a projectile diameter range of 4 to 400mm. Thus, within these ranges, the drag coefficient can be computed directly - that is, without any additional fitting process - for a given set of size and shape parameters [1].

The original "McDrag" program was tested against the actual experimental data for a large range of projectiles of differing shapes. This showed that the typical standard deviation errors to be expected were about 3% in the supersonic region, 11% in the transonic region and about 6% in the subsonic region. The largest errors at transonic speeds occur for boattailed projectiles, and this is believed to be related to the lack of any good similarity parameter for correlating transonic boattail effects. For nose lengths shorter than one calibre, the calculated contribution of drag from the bullet head will probably be too high for transonic and supersonic speeds. Boat-tail lengths longer than 1.5 calibres will result in calculated contributions of drag from the base and boat-tail that are incorrect. Likewise, the base diameter should not be less than 0.65 calibres or greater than 1.5 calibres, as the resulting boat-tail angle (or conical flare) will be too steep to give accurate results. This does not mean that the results will not be useful if these limits are stretched, but warnings will be given to indicate that the accuracies quoted above will probably not be valid [2].

4 Skin Friction Drag Coefficient

This section describes the drag force tangential to the surface caused by air viscosity that we will call **skin friction drag** or $C_{D_{SF}}$ (short for Coefficient - Drag - Skin Friction).

Overall the formula used for $C_{D_{SF}}$ looks as follows [1]:

$$C_{D_{SF}} = \frac{4}{\pi} C_F S_W \tag{3}$$

Where:

- C_F - skin friction coefficient for a smooth flat plane (unitless)
- S_W - projectile wetted surface in $calibers^2$

4.1 Reynolds number

The skin friction coefficient for a smooth plane, C_F, calculation depends primarily on Reynolds number. The Reynolds number expresses the ratio of inertial forces to viscous forces. It is a key similarity parameter used in flow viscosity calculations. Overall Reynolds number is calculated as follows [3]:

$$Re = \frac{\rho V L}{\mu} \tag{4}$$

Where:

- ρ = density of the fluid in $\frac{kg}{m^3}$
- V = flow speed in $\frac{m}{s}$
- L = length of the surface interacting with the fluid in m
- μ = bulk (dynamic) viscosity of the fluid in $\frac{kg}{ms}$

Since dynamic viscosity divided by density results in kinematic viscosity, equation 4 can be simplified to [3]:

$$Re = \frac{VL}{\nu} \tag{5}$$

Where:

- ν = kinematic viscosity in $\frac{m^2}{s}$

Assuming the speed of sound in air of 343$\frac{m}{s}$ and kinematic viscosity of air at sea level and 15°C of approximately 1.4723E-05$\frac{m^2}{s}$ we can get the formula as used by McCoy in the McDrag documentation [1]:

$$Re = 23296.3 M L_T d_{REF} \tag{6}$$

Where:

- M = free stream Mach number (unitless)
- L_T = total length of the projectile
- d_{REF} = reference diameter of the projectile in mm (that is milimeters)

4.2 Boundary layer type

There are two types of boundary layer flow type that need to be considered: laminar flow C_{F_L} and turbulent flow C_{F_T}. The formulas based on McDrag original documentation depend on Reynolds number and Mach number [1]:

$$C_{F_L} = \frac{1.328}{\sqrt{Re}}(1 + 0.12M^2)^{-0.12} \tag{7}$$

$$C_{F_T} = \frac{0.455}{(\log_{10} Re)^{2.58}}(1 + 0.21M^2)^{-0.32} \tag{8}$$

To identify the flow type program expects one of the inputs as listed below:

- L/L = fully laminar flow
- L/T = laminar flow for nosecone, turbulent for afterbody
- T/T = fully turbulent flow

According to McDrag original documentation a fully laminar flow does not apply to any relevant projectiles but the option exists nonetheless. For smooth projectiles under $20mm$ in diameter the transition from laminar to turbulent flow is assumed to happen near the end of the nosecone. The final coefficient is calculated as the weighted average where the weights are derived from the wetted surface formula as described in subsection 4.3. For projectiles over $20mm$ in diameter fully turbulent flow applies [1].

4.3 Wetted surface

In order to calculate the coefficient we need to know the surface area that the air interacts with - the wetted surface. As per McCoy [1] the surface of the projectile base is omitted since the air does not interact much with it at least not in terms of friction. The total surface is a sum of nosecone surface and the cylinder afterbody.

$$S_W = S_{W_{nose}} + S_{W_{cyl}} \tag{9}$$

Where:

$$S_{W_{cyl}} = \pi(L_T - L_N) \tag{10}$$

$$S_{W_{nose}} = \frac{\pi}{2}L_N(1 + \frac{1}{8L_N^2})[1 + (\frac{1}{3} + \frac{1}{50L_N^2})(\frac{R_T}{R})] \tag{11}$$

Where:

- L_N = length of the nosecone
- $\frac{R_T}{R}$ = nosecone shape

For mild boattails allowed in the described model the boattail surface area difference versus cylinder body is negligible. Hence we use the cylinder formula for the entire projectile afterbody.

4.4 Coefficient for a smooth flat plane

For a fully laminar or a fully turbulent flow the coefficient for a smooth flat plane C_F is equal to either C_{F_L} or C_{F_T} as described in subsection 4.2. For a laminar, then turbulent flow L/T a weighted average is calculated:

$$C_F = \frac{C_{F_L} S_{W_{nose}} + C_{F_T} S_{W_{cyl}}}{S_W} \tag{12}$$

This number is then inserted into equation 3 to calculate the final coefficient for the projectile.

4.5 Results

The results from the MatLab program for the three projectiles available in original McDrag documentation are practically identical to the results as calculated by McCoy [1]. The only difference is the rounding since McCoy was rounding the results to third decimal place.

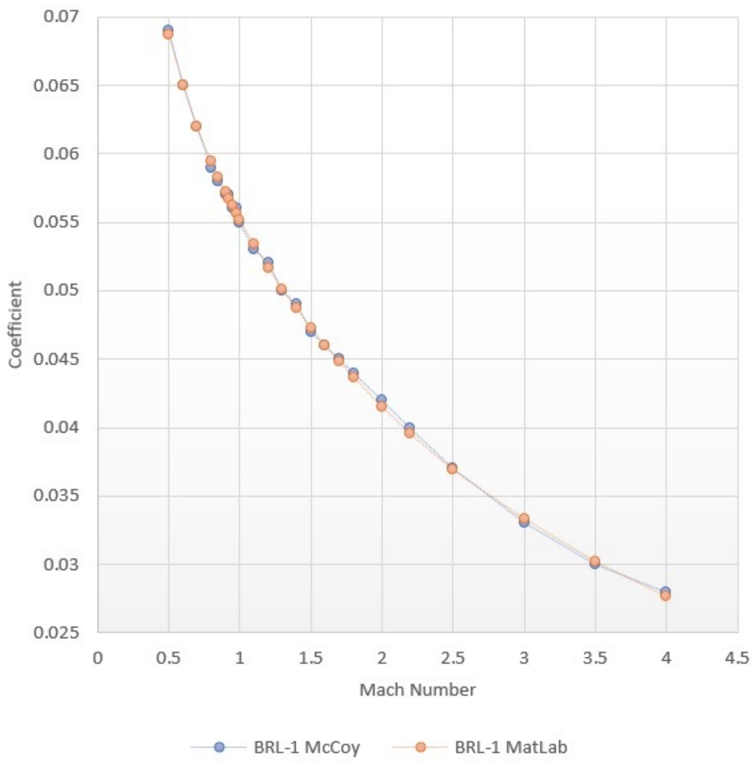


Figure 5: BRL-1 projectile results comparison

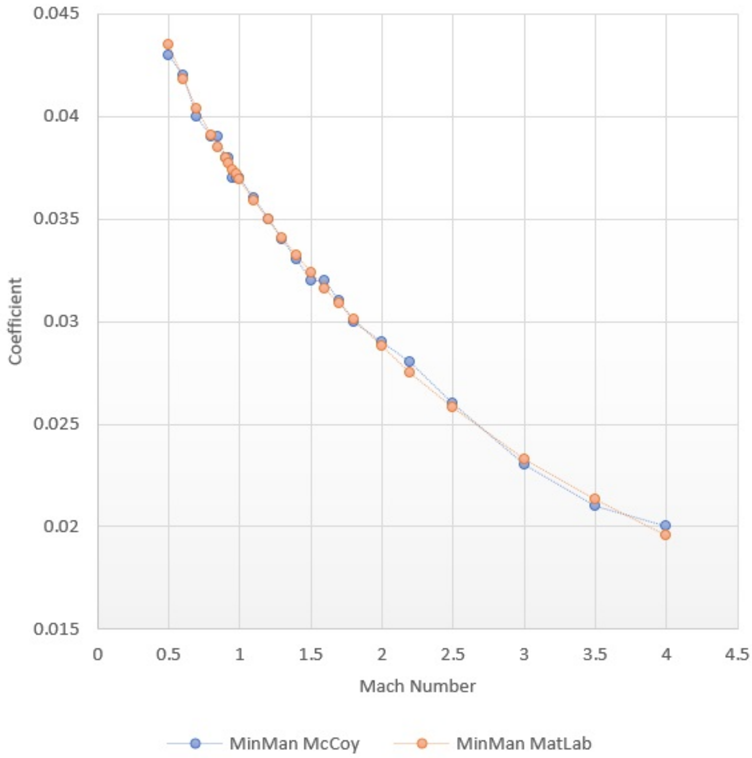


Figure 6: Minuteman projectile results comparison

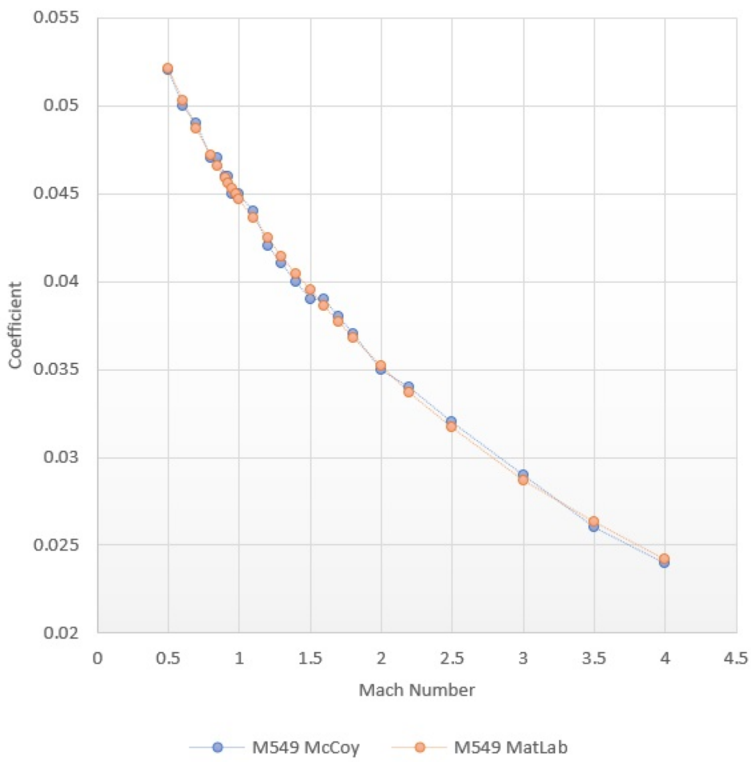


Figure 7: M549 projectile results comparison

5 Base Drag Coefficient

5.1 What it is about.

The role of my function is to calculate the exact drag coefficient deriving from the base of given projectile. The code is fully based on the US Army report **"MC Drag"** from 1981. Length of the base is defined as a difference between total missile length and the length of its nose/head.

5.2 Equations

$$[\frac{p_b}{p_\infty}] = (1 + 0.09 * M^2 * (1 - e^{-L_{cyl}})) * (1 + 0.25 * M^2 * (1 - d_b)) \tag{13}$$

ratio of Base pressure to free stream static pressure

$$C_{D_B} = \frac{\gamma * (1 - [\frac{p_b}{p_\infty}]) * d_b^2}{1 + 0.1875 * M^2 + 0.0531 * M^4} * \frac{1}{M^2} \tag{14}$$

for Mach number lesser than 1

$$C_{D_B} = \frac{\gamma * (1 - [\frac{p_b}{p_\infty}]) * d_b^2}{1 + 0.2477 * M^2 + 0.0345 * M^4} * \frac{1}{M^2} \tag{15}$$

for Mach number greater than 1

$$C_{D_B} = \frac{2 * d_b^2}{\gamma * M^2} * (1 - \frac{p_b}{p_\infty}) \tag{16}$$

equation from the report

where:

- p_b- base pressure
- p_∞- Free stream static pressure
- ratio- ratio of Base pressure to Free stream static pressure
- M- Free stream Mach number
- L_{cyl}- cylinder length (calibers)
- d_b- base diameter (calibers)
- γ- ratio of specific heats

5.3 Explanation

Equation (1) calculates ratio of base pressure to free stream static pressure. It it taken from the report and it matches the code written in *FORTRAN*.
Equation (4) calculates base drag coefficient and it is the exact equation taken from the report. However the results from the equation (4) were far from the truth so I replaced it with equations taken from *FORTRAN* [(2) and (3)].
(2) is used when Mach numbers reaches from 0.5 to 1.0 and (3) calculates coefficient for Mach numbers from 1.0 to 4.0.

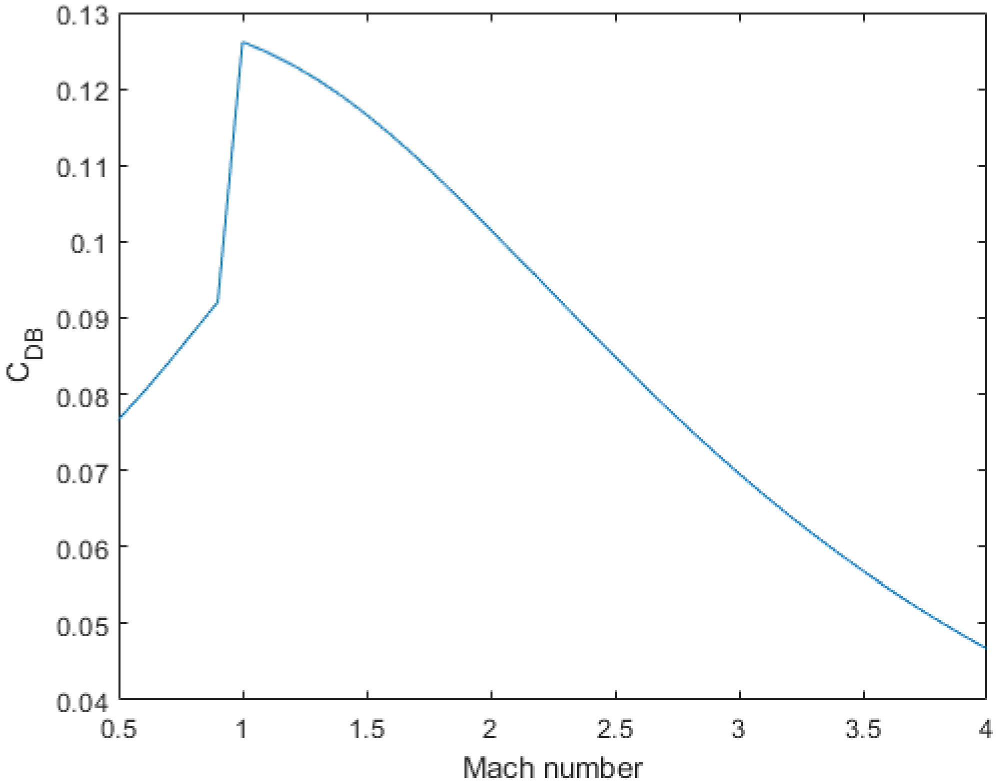


Figure 8: C_{D_B} to Mach number graph

5.4 Results

The results are satisfactory as all the numbers match to the results included in the report, tested with three different projectiles.
The graphs also match to those given in the original report.

5.5 Graphs in comparison

Graphs shown below illustrate the compatibility of the code with the results from the report. The first picture is an accurate graph for base drag coefficient calculated for 155mm M549 Projectile and the second is the information of what the shape of this graph should look like. As it can be observed- the graphs are matching. The same situation occur with other projectiles data included in the report.

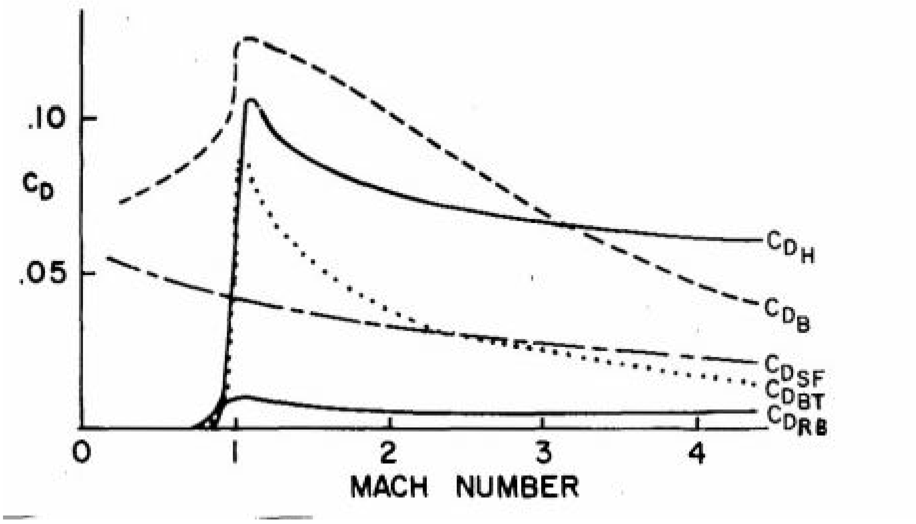


Figure 9: All drag coefficient graphs (including C_{D_B})

6 Boattail Drag Coefficient

6.1 Introduction

This section is about calculating **pressure drag coefficient for boattail** which later will be written as **CDBT**. Boattail is a part of a rocket/projectile which is placed at the very end. On the Figuer.1 it is circled red. The main source of information that was utilized during the creation of CDBT function

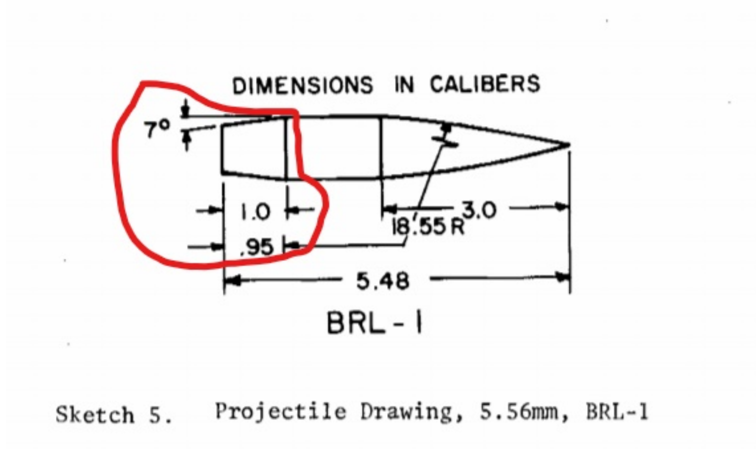


Figure 10: projectile with distinguished boattail

was [1]. I also got tremendously inspired by original **FORTRAN** MC DRAG's source code which can be found either in the documentation or on the Internet.

Whole CDBT section as well as the other functions were created using **MATLAB**.

6.2 Mathematics behind calculating CDBT

As stated int the Abstract the formula for calculating CDBT is described in [1] documentation. It is important to highlight we do distinguish two areas of speed: supersonic and trans-sonic, and by claiming that Mach's numbers from the area $[0.9; 1.0]$ relate to trans-sonic velocities, while Mach's numbers from the area $[1.05; 4.0]$ relate to supersonic we will use two different equitations.

The first formula is being used for supersonic Mach's numbers and small values of the boattail angles β.

$$C_{D_{BT}} = \frac{4Atan\beta}{k}\{(1 - e^{-kL_{BT}}) + 2tan\beta[e^{-kL_{BT}}(L_{BT} + \frac{1}{k}) - \frac{1}{k}]\} \tag{17}$$

where

$$k = \frac{0.85}{\sqrt{M^2 - 1}} \tag{18}$$

$$A_1 = [1 - \frac{3RTR}{5M}]\{\frac{5tau}{6\sqrt{M^2-1}}(\frac{tau}{2})^2 - \frac{0.7435}{M^2}(tau * M)^{1.6}\} \tag{19}$$

$$A = A_1 * e^{-\sqrt{\frac{2l_{CYL}}{\gamma M^2}}} + \frac{2tan\beta}{\sqrt{M^2-1}} - \frac{[(\gamma+1)M^4 - 4(M^2-1)](tan\beta)^2}{2(M^2-1)^2} \tag{20}$$

- L_{BT} - lenght of the boattail
- $beta$ - boattail angle
- k - boattail pressure recovery factor
- l_{CYL} - length of projectile cylinder section(calibers)
- A_1 - headshape correction factor for supersonic boattail drag coefficient

However when implemented into MATLAB this formula proved serious differences in results, the cause of that is not clearly stated. Therefore finding a solution was necessary. It turned out that by studying deeper the documentation and FORTRAN's source code there are significant dissimilarities. Coefficients A_1 and A where different and when FORTRAN's versions of them replaced those taken from the documentation, the results improved a lot. CDBT part was different either, however for the best results, more efficient was the formula from the documentation, rather than from FORTRAN. Parts A_1 and A taken from FORTRAN's source code are like below:

$$A_1 = (1 - \frac{0.6RTR}{M})\frac{5tau}{6\sqrt{M^2-1}} + (\frac{tau}{2})^2 - \frac{0.7435}{M^2}(tau * M)^{1.6} \tag{21}$$

$$A = A_1 e^{\frac{-1.1952}{M}(l_{CYL} - l_{bt})} - \frac{(2.4M^4 - 4(M^2-1)) * \frac{1-d_b}{4l_{BT}^2}}{2(M^2-1)^2} + \frac{2(1-d_b)}{2l_{BT} * \sqrt{M^2-1}} \tag{22}$$

where

- d_b - base diameter

Second formula for CDBT works for trans-sonic Mach's numbers. It is stated in [1], that the literature lacks in similarity parameter applicable to boattails at trans-sonic speeds, thus by means of experimental measures a correlation was found but only if fixed Mach number is used, can it be performed. The formula (7) is what [1] proposes.

$$C_{D_{BT}} = 4(tan\beta)^2(1 + \frac{1}{2}tan\beta)\{1 - e^{-2l_{BT}} + 2tan\beta[e^{-2l_{BT}}(l_{BT} + \frac{1}{2}) - \frac{1}{2}]\} \tag{23}$$

However, it is important to highlight, that this formula does not contain Mach number as a variable, so the documentation proposes to estimate the trans-sonic values of CDBT for fixed Mach's numbers. It caused problems, since for a computation reasons while operating on *for loops* it is much easier when we can operate on fluctuating Mach number. The only solution to solve this problem was to implement trans-sonic formula for CDBT from FORTRAN's source code.
Moreover, FORTRAN's source code provided us with two formulas for trans-sonic CDBT values, one for Mach's numbers $[0.9, 1.0)$ and second for $\{1.1\}$ this approach made it really clear when there is a transition between those two confusing areas, as well as when to begin calculations on supersonic speeds. Formulas (8) and (9) compute trans-sonic CDBT.
$M \subset [0.9, 1.0)$

$$C_{D_{BT}} = 2((2\frac{1-d_b}{2l_{BT}}^2 + \frac{1-d_b}{2l_{BT}}^3)(1 - e^{-2l_{BT}} + (2\frac{1-d_b}{2l_{BT}}((e^{(-2)*l_{BT})(l_{BT}+0.5)} - 0.5)))... \\ \frac{1}{0.564 + 1250\frac{M^2-1}{2.4M^2}^2} \tag{24}$$

$M \subset \{1.1\}$

$$C_{D_{BT}} = 2((2\frac{1-d_b}{2l_{BT}}^2 + \frac{1-d_b}{2l_{BT}}^3)(1 - e^{-2l_{BT}} + 2\frac{1-d_b}{2l_{BT}})((e^{-2l_{BT}}(l_{BT} + 0.5)) - 0.5)))(1.774 - 9.3\frac{M^2-1}{2.4M^2}) \tag{25}$$

6.3 Results Analysis

This section will focus on summarizing what was achieved in comparison to already prepared results that can be found in [1]. The documentation has several precise results for different projectiles. Our aim was to built as good copy of the MC DRAG algorithm as possible and the results below, which MATLAB version of CDBT returned are very similar. A computation error is approximately $\approx 5\%$.

First projectile to analyze is **55mm Minuteman Model**, below on the plot are results obtained by original FORTRAN's MC DRAG and by MATLAB's version of an algorithm.

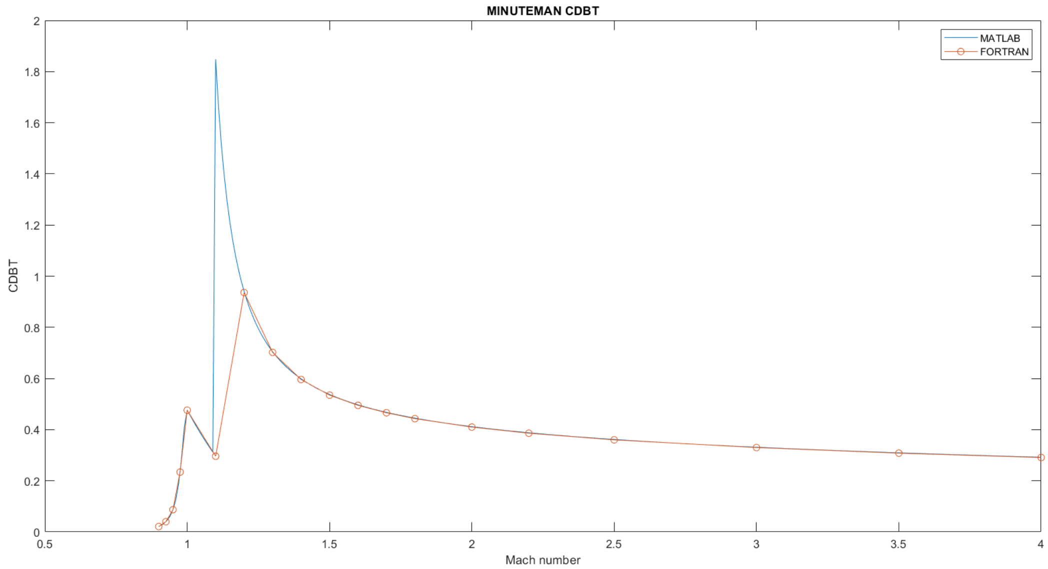


Figure 11: MINUTEMAN FORTRAN results

Second projectile to analyze is **BRL-1**, below there are results obtained by original FORTRAN's MC DRAG and below them there is a plot obtained by MATLAB's version of an algorithm.

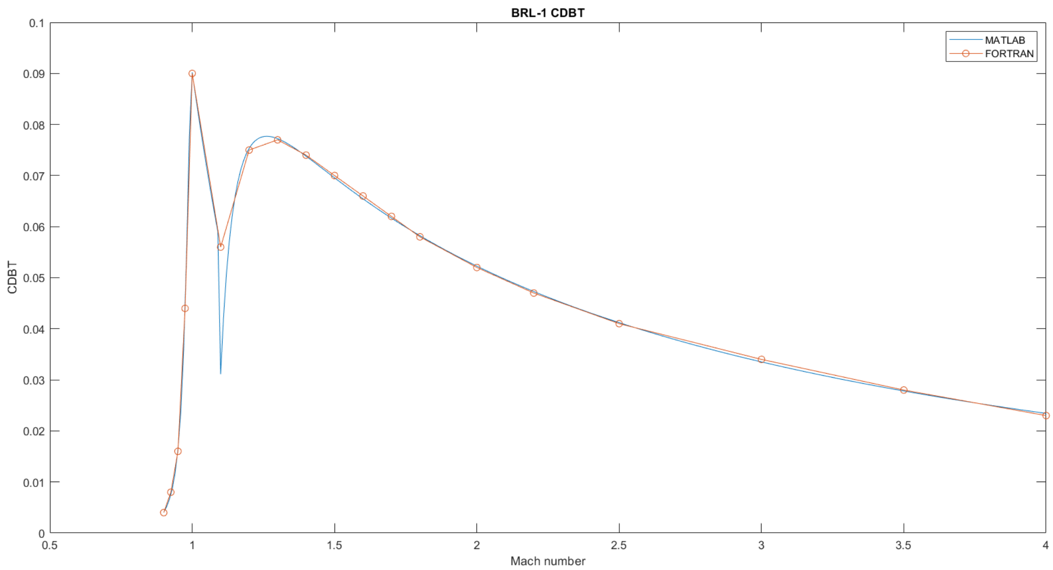


Figure 12: BRL-1 FORTRAN results

7 Head Drag Coefficient

7.1 Introduction

This section is about calculating **pressure drag coefficient for head** which later will be written as **CDH**. Head is a front part of a rocket/projectile. On the Figure 1 it is circled red.

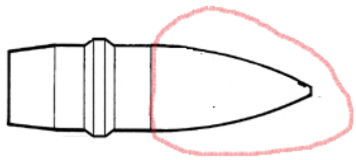


Figure 13: Projectile with marked head

7.2 Mathematics behind calculating CDH

The formula for calculating CDH is described in [1]. The key part of the report, thanks to which we were able to implement CDH in MATLAB, was the fortran source code.

I split CDH int two parts p_1 and p_2. CDH is the sum of these two parts.
The formula of p_1 depends on the value of the Mach number. For Mach number bigger than 1 is as follows:

$$p_1 = (c_1 - c_2 * \tau^2) * \frac{1}{z^2} * (\tau * z)^{c_3 + c_4 * \tau} \tag{26}$$

$$\tau = \frac{1 - d_M}{L_N} \tag{27}$$

Where:

- d_M - meplat diameter

$$\rho = \frac{(M^2 - 1)}{2.4 * M^2} \tag{28}$$

$$c_1 = 0.7156 - 0.5313 * \frac{R_T}{R} + 0.595 * (\frac{R_T}{R})^2 \tag{29}$$

$$c_2 = 0.0796 + 0.0779 * \frac{R_T}{R} \tag{30}$$

$$c_3 = 1.587 + 0.049 * \frac{R_T}{R} \tag{31}$$

$$c_4 = 0.1122 + 0.1658 * \frac{R_T}{R} \tag{32}$$

For Mach number smaller than s:

$$s = 1 + 0.368 * \tau^{1.8} + 1.6 * \tau * \rho \tag{33}$$

$$z = \sqrt{M^2 - 1} \tag{34}$$

Otherwise:

$$z = \sqrt{s^2 - 1} \tag{35}$$

p_1 for Mach nubmer smaller or equal than 1 and smaller than x:

$$x = (1 + 0.552 * \tau^{0.8})^{-0.5} \tag{36}$$

$$p_1 = \frac{0.368 * \tau * d_M^2}{M^2} \tag{37}$$

Otherwise p_1 is zero.
The formula of p_2 too depend of Mach value. For Mach number bigger or equal 1.41:

$$p_2 = 0.85 * \kappa \tag{38}$$

For Mach number bigger than 0.91 and smaller than 1.41:

$$p_2 = (0.254 + 2.88 * \rho) * \kappa \tag{39}$$

For other values of Mach number p_2 is zero.

$$\kappa = \frac{(1.122 * (\nu - 1) * d_M^2)}{M^2} \tag{40}$$

For Mach number value bigger than 1:

$$\nu = (1.2 * M^2)^{3.5} * \frac{6}{(7 * M^2 - 1)^{2.5}}) \tag{41}$$

For other values:

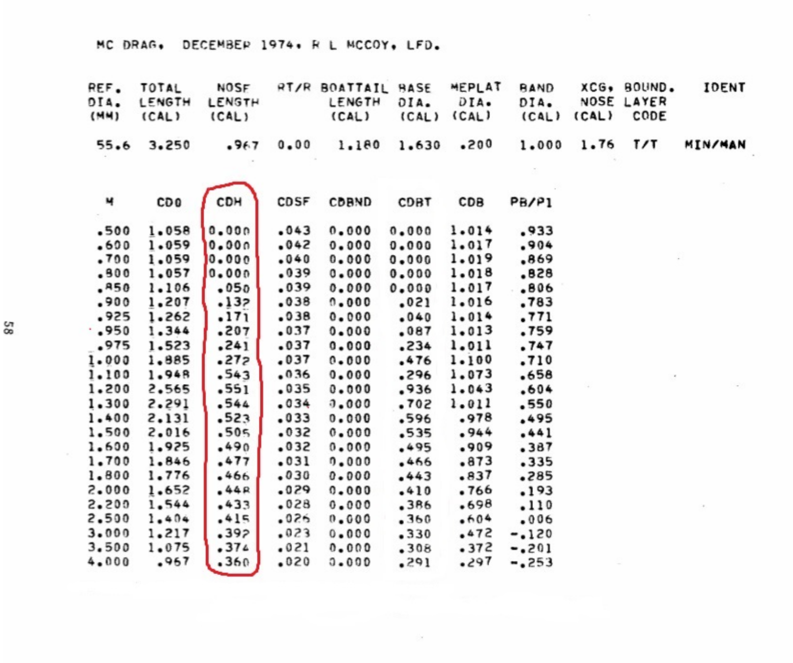
$$\nu = (1 + 0.2 * M^2)^{3.5} \tag{42}$$

7.3 Results Analysis

This section will focus on summarizing what was achieved in comparison to already prepared results that can be found in [1]. The documentation has several precise results for different projectiles. The MATLAB version of CDH returns practically the same results for the mach number between [0.5,0.91] and [1.61,5]. For mach numbers between [0.91,1.6] the computation error may even be 9%, but usually this error is much smaller.

First projectile to analyze is **55mm Minuteman Model**, below there are results obtained by original FORTRAN's MC DRAG (Figure 2), below them there is a plot obtained by FORTRAN and MATLAB's version of an algorithm (Figure 3) and at the end is graph of absolute error (Figure 4).

Second projectile to analyze is **BRL-1**, below there are results obtained by original FORTRAN's MC DRAG (Figure 5), below them there is a plot obtained by FORTRAN and MATLAB's version of an algorithm (Figure 6) and at the end is grapf of absolute error (Figure 7).



MC DRAG, DECEMBER 1974, R L MCCOY, LFD.

REF. DIA. (MM)	TOTAL LENGTH (CAL)	NOSE LENGTH (CAL)	RT/R	BOATTAIL LENGTH (CAL)	BASE DIA. (CAL)	MEPLAT DIA. (CAL)	BAND DIA. (CAL)	XCG, NOSE (CAL)	BOUND. LAYER CODE	IDENT
55.6	3.250	.967	0.00	1.180	1.630	.200	1.000	1.76	T/T	MIN/MAN

M	CD0	CDH	CDSF	CDBND	CDBT	CDB	PB/P1
.500	1.058	0.000	.043	0.000	0.000	1.014	.933
.600	1.059	0.000	.042	0.000	0.000	1.017	.904
.700	1.059	0.000	.040	0.000	0.000	1.019	.869
.800	1.057	0.000	.039	0.000	0.000	1.018	.828
.850	1.106	.050	.039	0.000	0.000	1.017	.806
.900	1.207	.132	.038	0.000	.021	1.016	.783
.925	1.262	.171	.038	0.000	.040	1.014	.771
.950	1.344	.207	.037	0.000	.087	1.013	.759
.975	1.523	.241	.037	0.000	.234	1.011	.747
1.000	1.885	.272	.037	0.000	.476	1.100	.710
1.100	1.948	.543	.036	0.000	.296	1.073	.658
1.200	2.565	.551	.035	0.000	.936	1.043	.604
1.300	2.291	.544	.034	0.000	.702	1.011	.550
1.400	2.131	.523	.033	0.000	.596	.978	.495
1.500	2.016	.505	.032	0.000	.535	.944	.441
1.600	1.925	.490	.032	0.000	.495	.909	.387
1.700	1.846	.477	.031	0.000	.466	.873	.335
1.800	1.776	.466	.030	0.000	.443	.837	.285
2.000	1.652	.448	.029	0.000	.410	.766	.193
2.200	1.544	.433	.028	0.000	.386	.698	.110
2.500	1.404	.415	.026	0.000	.360	.604	.006
3.000	1.217	.392	.023	0.000	.330	.472	-.120
3.500	1.075	.374	.021	0.000	.308	.372	-.201
4.000	.967	.360	.020	0.000	.291	.297	-.253

Figure 14: MINUTEMAN documentation results

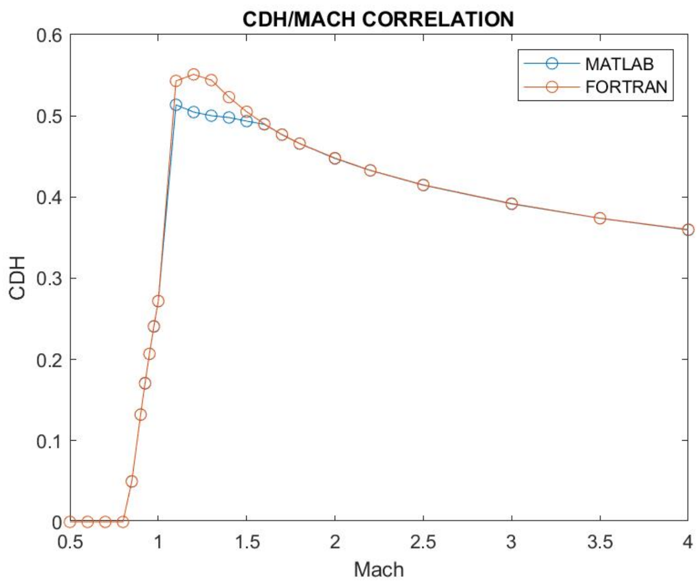


Figure 15: MINUTEMAN MATLAB/Documentation comparison

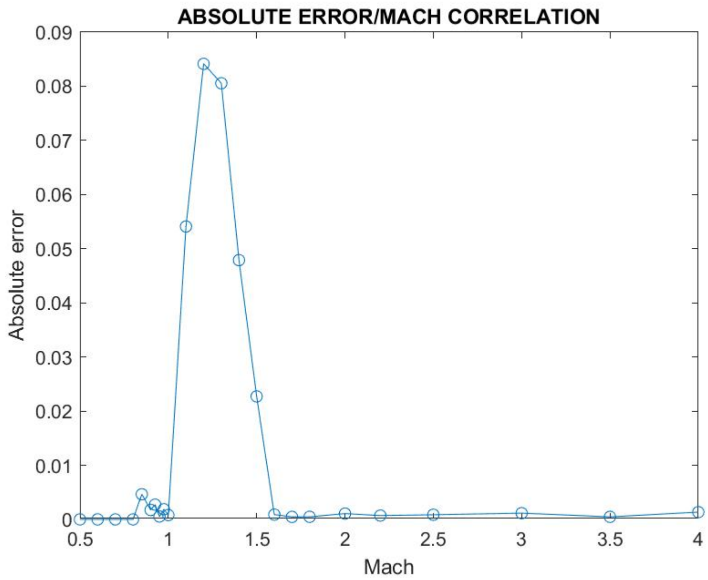
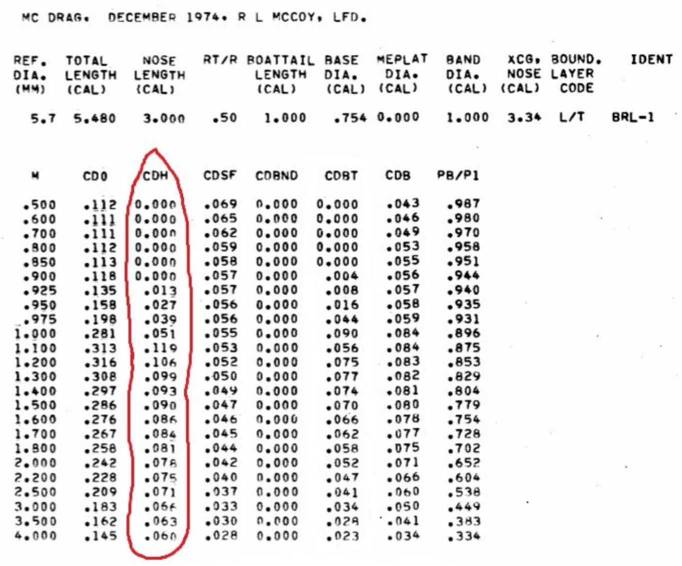


Figure 16: MINUTEMAN MATLAB absolute error



MC DRAG. DECEMBER 1974. R L MCCOY, LFD.

REF. DIA. (MM)	TOTAL LENGTH (CAL)	NOSE LENGTH (CAL)	RT/R	BOATTAIL LENGTH (CAL)	BASE DIA. (CAL)	MEPLAT DIA. (CAL)	BAND DIA. (CAL)	XCG, NOSE (CAL)	BOUND. LAYER CODE	IDENT
5.7	5.480	3.000	.50	1.000	.754	0.000	1.000	3.34	L/T	BRL-1

M	CD0	CDH	CDSF	CDBND	CDBT	CDB	PB/P1
.500	.112	0.000	.069	0.000	0.000	.043	.987
.600	.111	0.000	.065	0.000	0.000	.046	.980
.700	.111	0.000	.062	0.000	0.000	.049	.970
.800	.112	0.000	.059	0.000	0.000	.053	.958
.850	.113	0.000	.058	0.000	0.000	.055	.951
.900	.118	0.000	.057	0.000	.004	.056	.944
.925	.135	.013	.057	0.000	.008	.057	.940
.950	.158	.027	.056	0.000	.016	.058	.935
.975	.198	.039	.056	0.000	.044	.059	.931
1.000	.281	.051	.055	0.000	.090	.084	.896
1.100	.313	.119	.053	0.000	.056	.084	.875
1.200	.316	.106	.052	0.000	.075	.083	.853
1.300	.308	.099	.050	0.000	.077	.082	.829
1.400	.297	.093	.049	0.000	.074	.081	.804
1.500	.286	.090	.047	0.000	.070	.080	.779
1.600	.276	.086	.046	0.000	.066	.078	.754
1.700	.267	.084	.045	0.000	.062	.077	.728
1.800	.258	.081	.044	0.000	.058	.075	.702
2.000	.242	.078	.042	0.000	.052	.071	.652
2.200	.228	.075	.040	0.000	.047	.066	.604
2.500	.209	.071	.037	0.000	.041	.060	.538
3.000	.183	.066	.033	0.000	.034	.050	.449
3.500	.162	.063	.030	0.000	.028	.041	.383
4.000	.145	.060	.028	0.000	.023	.034	.334

Figure 17: BRL-1 documentation results

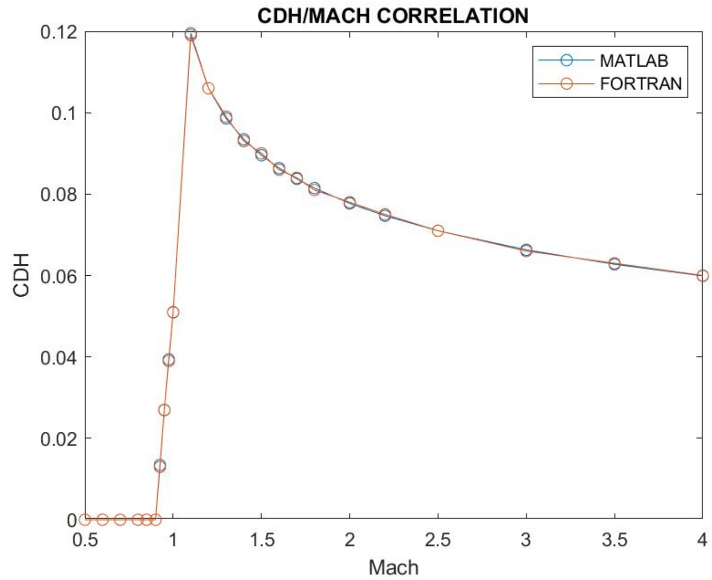


Figure 18: BRL-1 MATLAB/Documentation comparison

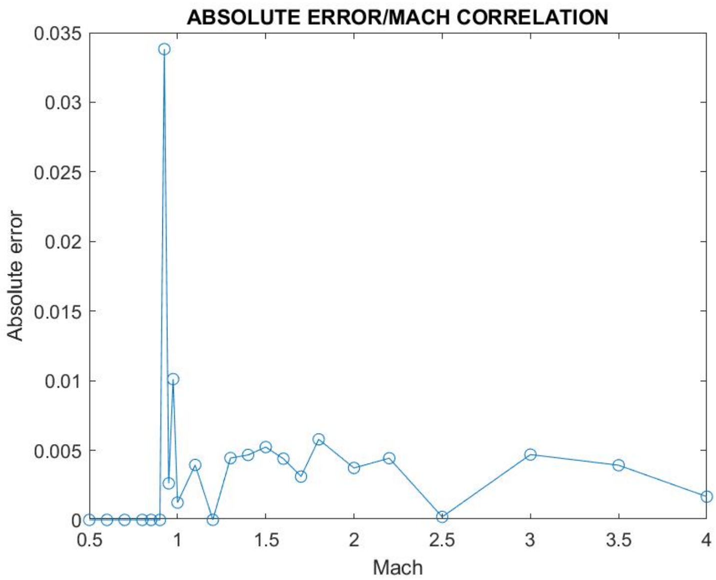


Figure 19: BRL-1 MATLAB absolute error

8 Rotating Band Drag Coefficient

To improve an accuracy and range of small arms and artillery riffle gun barrels were implemented. Rifling causes the projectile to spin rapidly during launch, therefore rotating bands were added to insure that slip between rifling and shell is eliminated. Below, on the scheme we can exactly spot the rotating band.

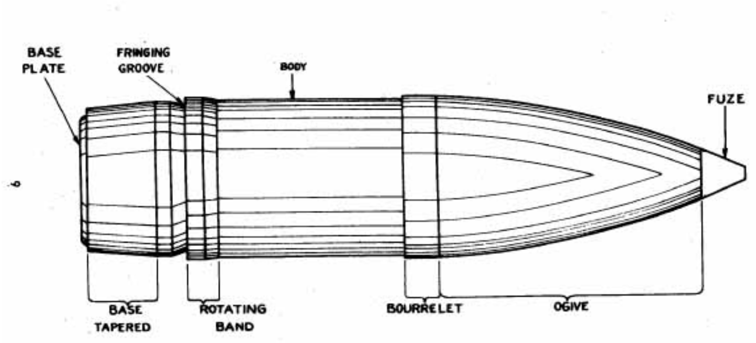


Figure 20: rotating band scheme

8.1 Mathematics behind calculating Rotating Band pressure drag coefficient

According to the [1] documentation to calculate **Rotating Band pressure drag coefficient**(later we will call it **CDRB**), we have to multiply the curve(Fig.2) by $(d_{rb} - 1)$, this assumption Robert L. McCoy makes due to the research of F.G Moore [4]- *"Body Alone Areodynamics of Guided and Unguided Projectiles at Subsonic, Transsonic and Supersonic Mach Numbers"*. According to Moore's paper, his wind tunnel experiment's results seems to prove that it is good approximation for calculating CDBR, however McCoy states that if we obtain more experimental data on the effects of band configuration and location we can get much more precise results.

On the figure below we can see the curve we have referred to at the beginning of this section. Both red and blue curves were extracted from FORTRAN's source code of **MC DRAG** algorithm. This plot was generated in MATLAB but it fits almost perfect to the original which is in *MC DRAG*.

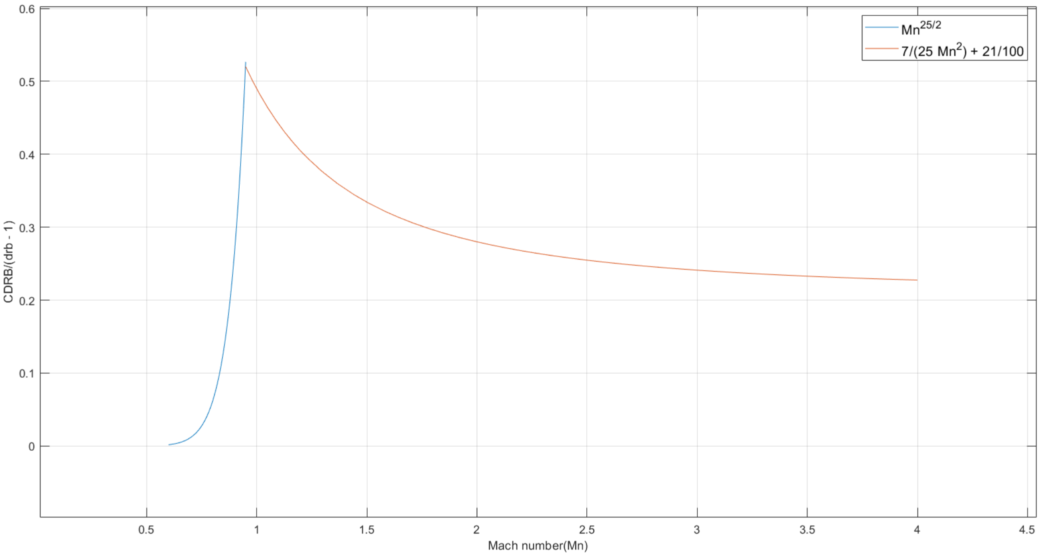


Figure 21: This curve we multiply by $(d_{rb} - 1)$ to get CDRB

Thus, to obtain CDRB we do need to generate two function like on the Figure 2 and properly iterate over **M**(Mach number). The formulas below explain this procedure:

$$Mn^{12.5} * (d_{rb} - 1)\,, for\; M \subset [0.9, 0.95] \tag{43}$$

Where:

- d_{rb} - rotating band diameter

$$(0.21 + \frac{0.28}{M^2})(d_{rb} - 1)\,, for\; M \subset [0.95, 4.0] \tag{44}$$

Here we can see how CDRB plot looks like for an example projectile with $d_{rb} = 1.02[cal]$

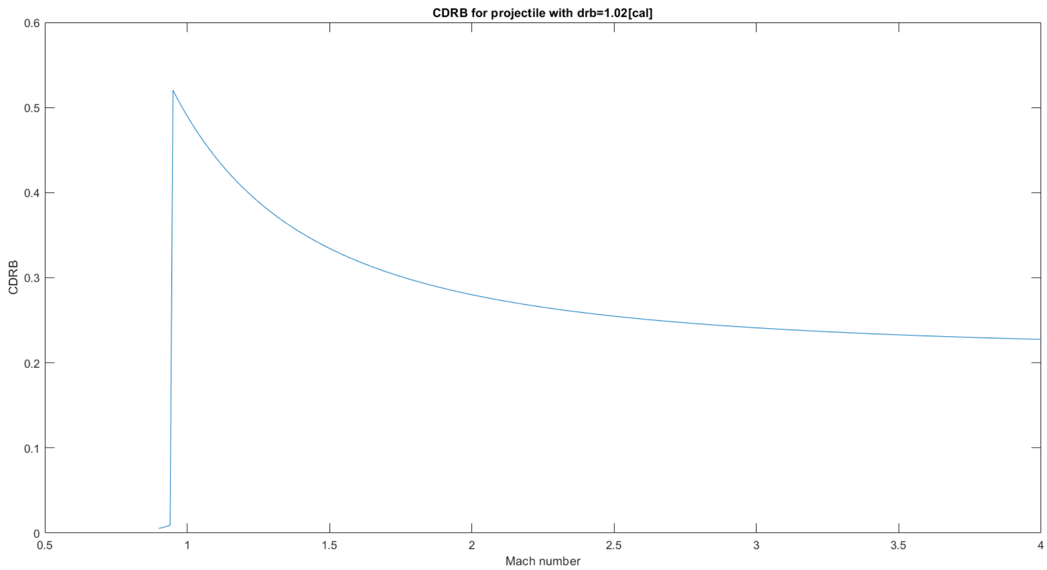


Figure 22: CDRB plot form projectile with $d_{rb} = 1.02[cal]$

8.2 Further conclusions

Method described above works correct however there are few improvements to be conducted on the field of calculating more precise CDRB values. James E. Danberg [5] in 1983 wrote tremendously interesting paper named *"Numerical Modeling of Rotating Band Flow Field and Comparison with Experiment"*. In this document the author explains drawbacks of Moore method as F.G Moore used only one model size of a projectile for his experiment. After reviewing Moore's experiment Danberg mentions two scientists whose works are worth checking. Moreover Danberg points out several conclusions that are quite important to understand what can be done to make CDRB calculations more reliable.

- The added drag of the band is small on the order of 5
- The correlations of the experimental data are not sufficient to accurately define the drag within a factor of two. This is probably because in wind tunnel and ballistic range experiments it is necessary to take the difference between the relatively large drag with and without the band to find the much smaller band drag.
- Pressure distribution measurements in the vicinity of the band might be a more sensitive method of extracting the force acting on the band
- If the correct perturbation of the viscid and inviscid flow field can be obtained in computer models, they may be expected to predict the effects of the rotating band on downstream skin friction, separation and base pressure effects.

There is also a statement which says that applying the full Navier-Stokes solver to the compressible and viscous surrounding flow would give the best results, but it requires major effort.

"Numerical Modeling of Rotating Band Flow Field and Comparison with Experiment" provides with many other interesting facts and formulas that may influence in a key way calculations of CDRB, therefore it is desirable to attempt to read it.

References

[1] R.L. McCoy. *MC DRAG - A Computer Program for Estimating the Drag Coefficients of Projectiles.* Tech. rep. DTIC ADA098110. Army Ballistic Research Lab Aberdeen Proving Ground MD, 1981.

[2] Geoffrey Kolbe. *BULLET DRAG CALCULATOR.* `http://www.geoffrey-kolbe.com/drag.htm` (accessed: 29.12.2020).

[3] Tom Benson. *NASA Glenn Research Center - Reynolds Number.* `https://www.grc.nasa.gov/www/BGH/reynolds.html` (accessed: 30.12.2020).

[4] F.G Moore. *F.G Moore - Body Alone Areodynamics of Guided and Unguided Projectiles at Subsonic, Transsonic and Supersonic Mach Numbers.* Tech. rep. TR-2796. Army Ballistic Research Lab Aberdeen Proving Ground MD, 1972.

[5] James E. Danberg. *James E. Danberg - Numerical Modeling of Rotating Band Flow Field and Comparison with Experiment.* Tech. rep. Army Ballistic Research Lab Aberdeen Proving Ground MD, 1983.