Comparison between trajectory models for firing table application

MAM Aldoegre

orcid.org/0000-0002-7020-344X

Dissertation submitted in partial fulfilment of the requirements for the degree *Master of Science in Mechanical Engineering* at the North-West University

Supervisor: Prof WL den Heijer

Graduation ceremony July 2019

Student number: 27359948

ACKNOWLEDGEMENTS

Above all else I would like to thank my Mother Norah, and my siblings for their unconditional love and support.

This research would not have been possible without the support of my supervisors Prof. Den Heijer and Mr. Du Plessis. Their efforts and advice were critical in much of the development of this research.

I would also like to thank my colleagues in the Military Industries Corporation (MIC), Raney Almehmadi, Naif Alotaibi, Saad Alqarni, Abdullah Altufayl, Shehab Alzahrani, Abdulaziz Bin Sultan and Marwan Alosaimi for their help and encouragement.

ABSTRACT

TITLE: Comparison between trajectory models for firing table application
AUTHOR: Motassm Abdullah Aldoegre
SUPERVISOR: Prof. Willem L den Heijer
CO-SUPERVISOR: Mr. Louis du Plessis

KEYWORDS: Trajectory Simulation Model, Point Mass, Modified Point Mass, Five Degree of Freedom.

Firing tables are mainly based on trajectory models such as the Point Mass model, the Modified Point Mass model or the Five Degree of Freedom model.

This research investigates and provides a comparison between these models for different projectiles, such as 81 mm mortar as a fin-stabilized projectile, and 105 mm artillery and 155 mm artillery as spin-stabilized projectiles.

Firstly, research was conducted to establish a comparison to identify the optimum model to generate the applicable firing table based on accuracy and time consumed during simulation time.

Secondly, the research showed he process of generation firing table by using MATLAB and discussed the effect of integration step on accuracy.

ABBREVIATIONS

AR:	Augmented Reality
BRL:	Ballistic Research Laboratory
CFD:	Computational Fluid Dynamics
DOF:	Degrees Of Freedom
ENIAC:	Electronic Numerical Integrator and Computer
MPMM:	Modified Point Mass Model
NABK:	NATO Armament Ballistic Kernel
NATO:	North Atlantic Treaty Organization
PRODAS:	Projectile Design/Analysis System
PMM:	Point Mass Model
STANAG:	NATO Standardization Agreement
WBS:	Work Breakdown Structure

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

Chapter 1 INTRODUCTION

1.1 *Introduction*

Generating a firing table is one of the final stages in projectile development which is a book of tables that contain all the data needed to describe a projectile trajectory between two points, and firing table is the primary input to fire control. As soon as the production item is available, it and its firing table are released to troops for training (Dickinson, 1967). Firing table is mainly based on trajectory models such as the Point Mass Model (PMM), the Modified Point Mass Model (MPMM) or the Five Degrees of Freedom (5-DOF).

PMM is the simplest amongst these models and depends only on gravity, drag and thrust forces, while 5-DOF is the most accurate. The MPMM is the most commonly used by NATO (North Atlantic Treaty Organization) as the international standard for armament trajectory simulations (STANAG 4355). The physical data requirements for the PMM and the MPMM are also less than that required by the 5-DOF. To provide some indication of the complexity in each model, please refer to Appendix 1.

To define the optimum model for a certain projectile such as spin-stabilized or fin-stabilized projectiles, one needs to find the optimum solution based on the following two factors:

- Accuracy with which model represents the real trajectory.
- Simulation time.

These factors play an important role where modern devices such as tablets are considered because of the limitation on processor speed and memory size to generate firing table and to predict fire control data.

Research shows that the MPMM has failed for high elevation trajectories, and Table 1 shows the elevations at which these critical conditions are reached for some shells (Pope, 1985)

Table 1: Critical gun elevations from (Pope, 1985)

Caliber (mm)	Muzzle velocity (m/s)	Elevation (deg)
105	464.8	66.6
105	708.0	65.2
155	826.2	63.5
155	643.0	64.7

STANAG 4355 also declares that “elevations exceeding some limits can lead to erratic flight behavior reflected by increased dispersion” (STANAG 4355, 2009).

1.2 *Background*

Standard firing table (STANAG 4119, 2007) contain 10 tables. as shown in Table 2. Note that Table F contains the fundamental information on trajectory sensitivities. Initially, the calculation of these sensitivities was done by hand. This is a tedious task, requiring several thousand trajectory calculations. Then, the tasks began to be performed by electronic computers since the birth of ENIAC (Electronic Numerical Integrator and Computer). A tabular firing table typically consists of a collection of smaller tables as shown in Table 2. Additional examples are provided in Appendix 3. (Nangsue, 2010)

Table 2: List of tables for artillery firing table

Table code	Table name
A	Line numbers of meteorological message
B	Complementary range and line numbers of the meteorological message
C	Components of a one-knot wind
D	Air temperature and density corrections
E	Effects on muzzle velocity due to propellant temperature
F	Basic data and correction factors
G	Supplementary data
H	Corrections to range in meters to compensate for the rotation of the earth
I	Corrections to azimuth in mils to compensate for the rotation of the earth
J	Fuse correction factors

1.3 *Problem statement*

Regarding the importance of the simulation time in firing table prediction, “The model must be simple in terms of computation (time-consuming process). Time-consuming calculations cannot be accepted in live-firing scenarios on the battlefield, where time is the critical issue.” (Baranovski, 2013b). Selecting the appropriate model will allow for high accuracy, efficient calculation and minimum characteristic data required. Therefore, the problem to be solved is to identify an optimum model that provides a more efficient firing table prediction for a specific projectile.

1.4 *Research aim and objectives*

1.4.1 *Research aim*

To study trajectory models in firing table application and identify the most appropriate model in relation to the projectile and firing conditions. This leads to implementing it on modern devices and integrating it with available Geophysical (Google Earth) and Meteorological (weather) data.

1.4.2 *Research objectives*

The primary objective of this research is to:

- To study the accuracy of existing models;
- To evaluate these models with respect to the simulation time;
- To determine the best model for any projectile to generate the firing table; and
- To identify the limitations of each model.

1.5 *Research methodology*

1. Build a trajectory model simulator for each model (PMM, MPMM and 5-DOF): "A trajectory model is a mathematical model that describes the motion of the projectile starting from the howitzers muzzle to its termination" (Nangsue, 2010), and the simulator is based on:

I. Equations of Motion relative to an Earth-fixed reference frame

II. Data required for each model (Physical, Aerodynamic and Fitting Factors) as listed in Appendix 1.

2. Evaluate various projectiles into the simulators. The following three projectiles as representative case studies, are included within the scope of work of this research:

I. 81 mm (fin-stabilized mortar).

II. 105 mm (spin-stabilized artillery).

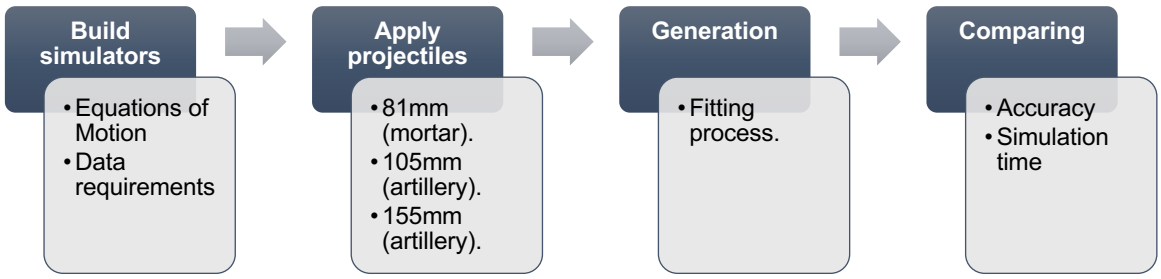
III. 155 mm (spin-stabilized artillery).

3. Derive model data for these projectiles through a fitting process of available test data. "STANAG 4355 (Edition 2,1997) recommends a polynomial fit of

degree 2 (quadratic). However, a polynomial fit of degree 3 (cubic) is recommended in the updated 3rd edition (2009 issue)"(Baranovski, 2013a). Use these models to generate firing table according to the guidelines in STANAG 4355 and STANAG 4119.

4. Compare the computed results obtained from the different models in terms of trajectory accuracy and simulation time.

1.6 *Work Breakdown Structure (WBS)*



1.7 *Limitations of the study*

Different types of projectiles that are considered for this study are limited to three. However, the results of these 3 types of projectiles can be generalised for similar munitions (e.g. small-munition). This study also does not consider rocket artillery.

1.8 *Summary*

Firing tables are mainly based on Trajectory models such as the Point Mass model (PMM), the Modified Point Mass model (MPMM) or the Five Degree of Freedom (5-DOF). This research is going to serve as a comparison between these models for different projectiles. The process involves four main steps; build trajectory model simulators, apply various projectiles, generate firing table and comparing them with artillery test results. Therefore, one can achieve an optimum model that provides a more efficient firing table for a specific projectile based on the trajectory accuracy and computation time.

Chapter 2 Literature review

2.1 *Introduction*

This chapter focuses on the literature that was studied to identify suitable trajectory models. Since Isaac Newton devised a universal theory of mechanics, so called “Newtonian Mechanics”, including the three laws of motion, it is now the standard way of modelling projectile motion. The literature review focuses on the trajectory from a numerical simulation point of view, keeping in mind the process to generate a firing table and the integration method, followed by the previous effort in trajectory models comparing field.

2.2 *Trajectory models*

The motion of a projectile is very complex in launch process and flight, because of the complex structure of self-propelled projectiles and the severe environments, such as, high temperature, high pressure, high speed. The errors of models depend on how much the projectile equations of motion are simplified. Increasing model complexity means decreasing model error (Khalil, 2013).

Models for symmetric projectiles (to be either a body of revolution whose spin axis coincides with a principal axis of inertia, or a fin stabilized projectiles such as mortars) traces its origin to the formulation by Fowler, Gallop, Lock and Richmond in 1920. In 1950, Clippinger and Gerber programmed the ENIAC computer at BRL to solve supersonic flow fields past projectiles by the method of characteristics, and computational aerodynamics at last began to catch up with experimental results. (McCoy, 2012)

Exterior ballistics scientists have made great developments during last 50 years, more than what has been done in the past, due to the fast developments occurring in the modern engineering sciences, especially in the computing process. During the 1960s, the ballistic research laboratory developed 6-degree-of-freedom (6-DOF) rigid body equations of motion (Lieske and McCoy 1964), which is similar to the 5-degree-of-freedom (5-DOF) for symmetrical projectiles. However, due to complexity and a long

processing time, Lieske and Reiter (1966) developed a Point Mass Model (PMM) and modified point mass model (MPMM) assuming that the yawing motion of the projectile are neglected or small everywhere along the trajectory (Khalil, 2014). The following sections will discuss these models briefly.

2.2.1 *PMM*

The first known analytical solution of the differential equations describing a point-mass trajectory was obtained in 1711, by Johann Bernoulli (1667-1748) of Switzerland. Bernoulli's solution assumed constant air density and constant drag coefficient, thus it was valid only for low velocity, flat-fire trajectories. About forty years later another Swiss mathematician, Leonhard Euler (1707-1783), developed the mean-value, short-arc method for solving systems of ordinary differential equations, and used his method to solve elementary point-mass trajectories (McCoy, 2012).

Until the mid-sixties firing table have been established successfully with the point mass model. Field artillery computers, applying this model, have been operational until the beginning of the eighties. The results obtained were mostly satisfactory for rather limited ranges on the assumption that range fitting and drift reduction from ballistic range firings were accurately performed. Correct, prediction of flight time using residual corrections deduced from the ballistic range firing was rather problematic such that fire for effect with air burst was hardly possible without the use of proximity fuses. (Celens, 1993)

2.2.2 *MPMM*

The modified point mass trajectory model was originally proposed by Lieske and Reiter (1966). There have been many contributions towards a sound theoretical base for this model see for example Celens (1993) and Bradley (1990). It was also critically evaluated by for example Baranoski (2013). But stood the test of time and became the preferred standard for NATO armament trajectory simulations as described in STANAG 4355.

On the other hand, research done by Pope (1985), indicated limitations of the modified point mass trajectory model to predict trajectories of very high gun elevations as shown in Table 1.

According to (Pope, 1985), “problems do arise in the development of firing tables. In general, firing tables are produced for elevations up to 70°. Therefore the range and accuracy trials on which the tables are based often contain up to 20% of data points from firings to be at elevations between 65° and 70°, sometimes even more. Because of the rapidly deteriorating performance of the modified point mass trajectory model at these elevations, the calibration process has to cope with significant errors from this portion of the data, which will produce corresponding distortions of the fitting process at low to moderate elevations“.

2.2.3 *5-DOF*

5-degree-of-freedom (5-DOF) is a simplified model of 6-degree-of-freedom (6-DOF), which suggested to be applied in firing table application rather than the 6-DOF, due to (Celens, 1993) :

I. The longitudinal symmetry (allowing small angles approximation and linearization).
II. Practical independency of the pitch and yaw motion (allowing both motions to be studied separately without interaction).

2.3 *Process to generate a firing table*

Research done by Dr. Phaderm Nangsue (2010), describes the process that had been performed in generating firing tables for the howitzer with base-bleed extended range projectiles. The process involves four main steps as shown in Figure 1, which are:

I. Determining projectile coefficients.
II. Selecting the desired trajectory model.
III. Use the model to predict firing table data.
IV. Present firing table results into the user interface.

In determining projectile coefficients, the model that Nangsue selected is the modified point-mass model because of its high accuracy without significant computational

requirements. Moreover, this is the main disadvantage in this study that it is only focused on one model.

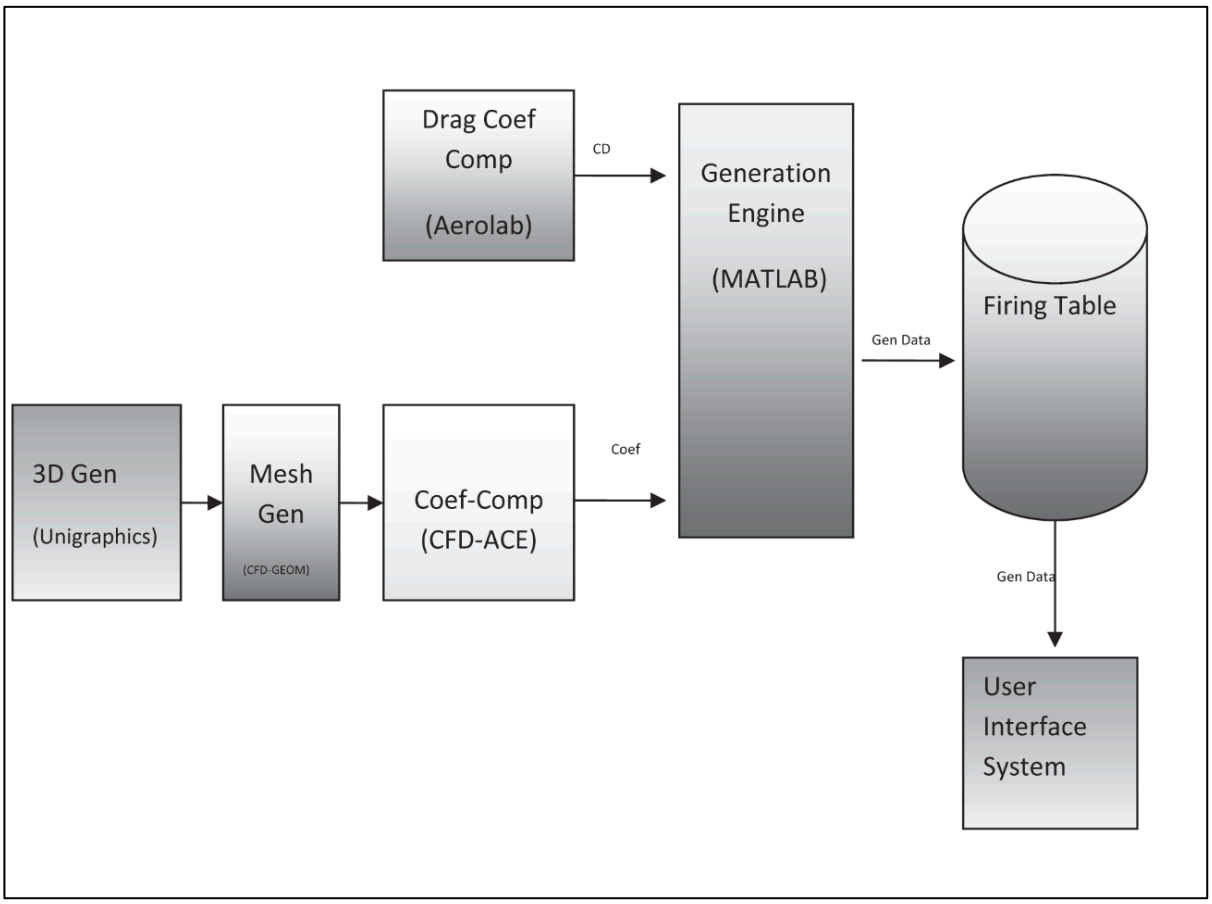


Figure 1: Firing Table Generation Process from (Nangsue, 2010)

Finally, the generated firing table for the howitzer with a base-bleed project was able to achieve 80 m accuracy for a 9.500m firing range, which is less than 1% of firing range. (Nangsue, 2010).

2.4 *Integration schemes*

New research on numerical solutions of classical equations of motion done by Dr. Anders W. Sandvik (2018). Discussed some of the numerical schemes that are suitable for integrating classical equations of motions which is the basis of trajectory models, the author studied three common schemes:

- Euler's forward method: $x_{n+1} = x_n + \Delta_t v_n$
- Leapfrog method: $x_{n+1} = x_n + \Delta_t v_{n+\frac{1}{2}}$
- The Runge-Kutta method.

Where x_{n+1} is the position at n, v_n is the velocity and Δ_t is the time step. The research shows differences between schemes in the simplicity and the error build-up (Sandvik, 2018).

2.4.1 *Effect of the integration step on the accuracy of the results of computation of artillery projectiles*

Leszek Baranowski (2013b), worked with a spin-stabilised projectile. Using a simulation of three different mathematical trajectory models. The mathematical models have been used to simulate the flight of 155 mm artillery projectiles, and to conduct general research on the influence of the applied models and the time-step, on the accuracy and time of computation of projectile trajectory.

Unfortunately, the study did not include fin-stabilised projectiles. One of the main outcomes of this research was, first, to obtain computation accuracy of 0.1% for drift and 0.02% for the remaining parameters (Range, time of flight, maximum height). Secondly, The model based on the body axis system should use an integration step enabling approximately 100 computations per projectile revolution (Baranowski, 2013).

2.5 *Previous effort in comparing between trajectory models*

Most of the researches in this field was studying and analysing the trajectory models in general, No research focused on firing table generation. In the following sections some of the main studies in this field are addressed.

2.5.1 *Review the Mathematical Models Used to Describe the Flight Dynamics*

Celens (1993), overviewed different trajectory calculation models used in NATO countries. First, the well-known numerical models (6-DOF, PMM, MPMM) are treated from the point of view of their use, access of parameters and possible restrictions. Secondly, the analytical approaches are treated as they give good physical insight; especially the different formulations of stability criteria are critically analysed.

2.5.2 *Modified Projectile Linear Theory for Rapid Trajectory Prediction Leonard*

Leonard Hainz (2005), worked with some smart weapons, estimation of the impact point of the shell at each computation cycle of the control law is an integral part of the control strategy. In these situations, the impact point predictor is part of the embedded computing system onboard the projectile. Using PMM show to exhibit poor impact point prediction for long-range shots with high gun elevations characteristic of smart indirect fire munitions. Leonard results provided for a short-range trajectory of a fin-stabilized projectile and a long-range trajectory for a spin-stabilized round.

2.5.3 *Feasibility analysis of a model for the need of firing table*

Leszek Baranowski, (2013) presented computational results for a spin-stabilised projectile using the modified point mass model for the flight trajectory, to determine an appropriate mathematical model (either MPMM or 6-DOF) for the automated fire control systems of ground artillery. The research concluded with two important recommendations:

Approximation of the fitting factors by the cubic polynomial gives better results (reduces error) than approximation by the quadratic polynomial.

In most cases of live firing, the range and deflection errors do not exceed a few meters, whereas the highest errors in the range are associated with the steep firing trajectories under strong longitudinal wind conditions (Baranovski, 2013b).

2.6 *Conclusion*

From previous studies it can be concluded that :

- Modified Point Mass has a breakdown point and limitation in high gun elevation. Despite this, the MPMM is the most commonly used by NATO.
- The process selected to generate the firing table will affect the accuracy of it.
- The integration method is Euler's forward, as the simplest numerical scheme for equation of motion. Higher order integration schemes were not considered in this research regarding to simplicity.
- The trajectory model and integration step have a significant influence on the accuracy and the time required to simulate a trajectory.

Chapter 3 Models of Trajectory

3.1 *Introduction*

This chapter focus on the forces and moments acting on axially symmetrical projectiles and the implementation thereof in various trajectory simulation models. This is followed by an evaluation of the accuracy of different trajectory simulation models.

The chapter concludes with an evaluation of the integration time-step on accuracy. A trajectory model is a mathematical model that describes the motion of the projectile starting from the gun to its termination point. Several trajectory models have been developed. Some models can be solved analytically, while others require a numerical method to solve it (Nangsue, 2010).

Important characteristics of a trajectory model is its accuracy, simplicity and computational requirements. The research will briefly describe three trajectory models in this chapter, starting from the least complex (i.e. PMM) to the most complex model considered here (i.e. 5-DOF).

3.2 *Aerodynamics forces and moments*

The dynamics of rigid bodies has evolved as a specialized branch in the modern science of exterior ballistics, poignant under the influence of gravitational and aerodynamic forces. A comprehensive history of exterior ballistics would fill several volumes, and only a few pertinent highlights are included in this chapter. Robert L. McCoy in his book “Modern Exterior Ballistics” serves as a good body of work in explaining these forces.

3.2.1 *Gravity:*

There are many models available to describe the earth’s gravity field as shown in Figure 2. Starting from the most elementary:

I. Flat Earth Model: Constant gravity field
II. Spherical Earth Model: an inverse square gravity model
III. Ellipsoidal Earth Model

IV. Complex models that utilised both longitudinal and latitudinal harmonics to present the gravity field

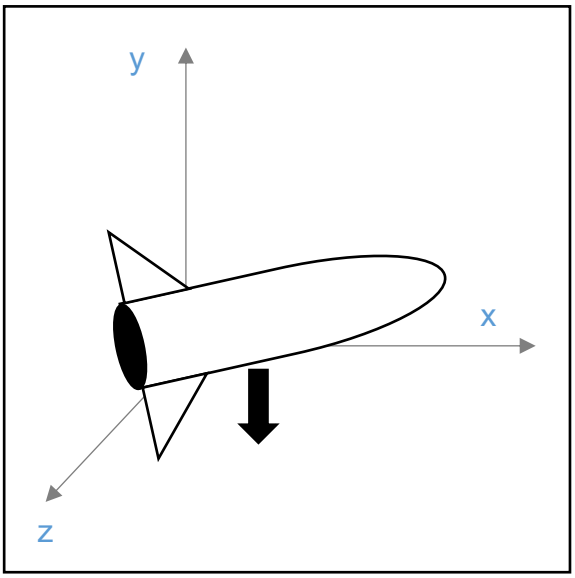


Figure 2: Gravity Force where black bold arrow shows the effect direction

STANAG 4355 (that has become an international standard for artillery) uses the following approximation to describe the gravity field for ellipsoidal earth:

$$\vec{g} = -g_0 \left(\frac{R^2}{r^3}\right)\vec{r} \tag{3.1}$$

Where

$$g_0 = 9.80665[1 - 0.0026\cos(2lat)] \quad \mathrm{m/s} \tag{3.2}$$

$$\vec{r} = \vec{X} - \vec{R} \tag{3.3}$$

$$\vec{R} = \begin{bmatrix} 0 \\ -6.356766 \times 10^6 \\ 0 \end{bmatrix} \tag{3.4}$$

R is the radius of the Earth, locally approximating the geoid, $\vec{X}$ is the position vector of the projectile in ground-fixed coordinate system, lat is the latitude of the projectile on Earth.

3.2.2 *Earth rotation (Coriolis effect):*

The Coriolis force is an apparent force with long range trajectoris. Unlike all apparent forces, it is an imaginary force which has no real existence. Consequently physicists prefer to use the words “Coriolis effect” rather than force, in order to distinguish the effect from Earth rotation. The Coriolis effect, as predicted by Newton, is not due to a force, but due to the fact that motion is described relative to a non-inertial (rotating) Earth fixed reference frame as shown in Figure 3.

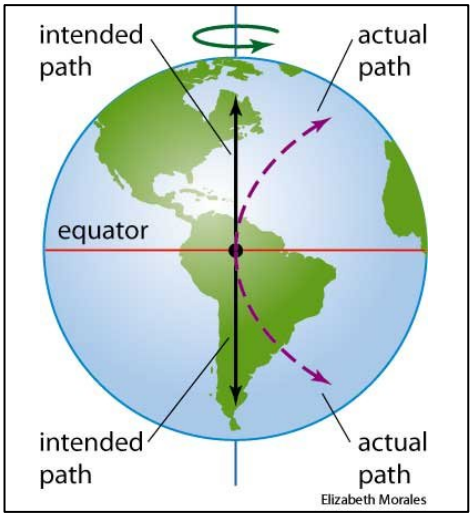


Figure 3: The effect of the earth rotation on a trajectory

STANAG 4355 described it by the following equations:

$$\vec{\Lambda} = -2(\vec{\omega} \times \vec{u}) \tag{3.5}$$

where

$$\vec{\omega} = \begin{bmatrix} \Omega \cos(lat) \cos(AZ) \\ \Omega \sin(lat) \\ -\Omega \cos(lat) \sin(AZ) \end{bmatrix} \tag{3.6}$$

$$\Omega = 7.292115 \times 10^{-5} \text{ rad/s} \tag{3.7}$$

AZ is the launching Azimuth angle from north clockwise, $\vec{\omega}$ is the angular velocity of the coordinate system due to the angular speed of the earth, Ω is the angular speed of the earth, $\vec{\Lambda}$ is the acceleration due to Coriolis effect Coefficient and $\vec{u}$ is the velocity of the projectile with respect to ground-fixed axes.

3.2.3 *Drag force:*

Aerodynamic drag is the resistance from an object moving through a fluid as shown in Figure 4. In external ballistics, the different forces and moments caused by airflow play a major role in determining where the projectile will end up. Modelling the drag can be a great challenge as it is very complex, or often near impossible, to analytically model this aerodynamic coefficient due to the complex behaviour of the flow of air around the projectile based on its shape.

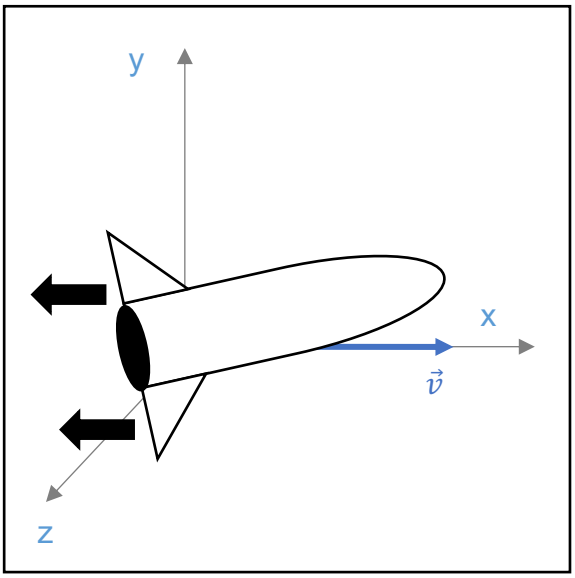


Figure 4: Drag force where bold arrow shows the effect direction

From literature, we know that it is possible to represent acceleration due to the drag force felt by a body by the equation below.

$$\frac{\overrightarrow{DF}}{m} = -\frac{S\rho}{2m}\left(C_{D_o}\right)v\vec{v} \tag{3.8}$$

Where S is the cross-section area of the body, for a projectile flying straight this is generally the area of a circle with a diameter as the calibre of the projectile, m is the projectile mass, ρ is the density of air, considerations should be made that the density is not constant but varies with atmospheric conditions and altitude. also v is the projectile velocity (vector and scalar) with respect to air and C_{D_o} is the aerodynamic drag coefficient, that varies with the flow around the object and is usually presented as a function of the Mach number.

3.2.4 *Lift force:*

The lift force acts on the centre of pressure for a projectile, and it is pointed perpendicular to the airflow as shown in Figure 5. The magnitude of lift is affected by several factors mainly the projectile design.

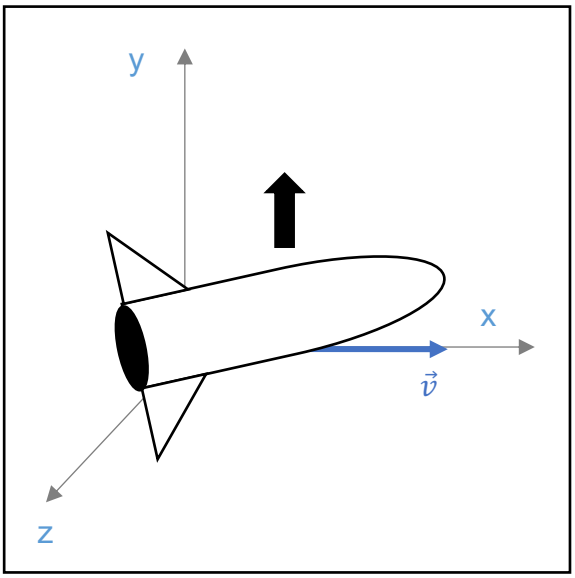


Figure 5: Left force where bold arrow shows the effect direction

STANAG 4355 described it for MPMM and 5-DOF by the following equations:

$$\overrightarrow{LF} = \frac{S\rho}{2} C_{L_\alpha}(v^2\vec{x} - (\vec{v}\cdot\vec{x})\vec{v}) \qquad (3.9)$$

Where $\vec{x}$ is the projectile orientation vector and C_{L_α} is the aerodynamic Lift coefficient, that varies with the flow around the object, and it is usually presented as a function of Mach number.

3.2.5 *Magnus Force:*

The Magnus force is commonly associated with a spin-stabilised projectile when airflow drags faster around one side, creating pressure difference affecting the projectile to move in the direction of the lower-pressure side as shown in Figure 6.

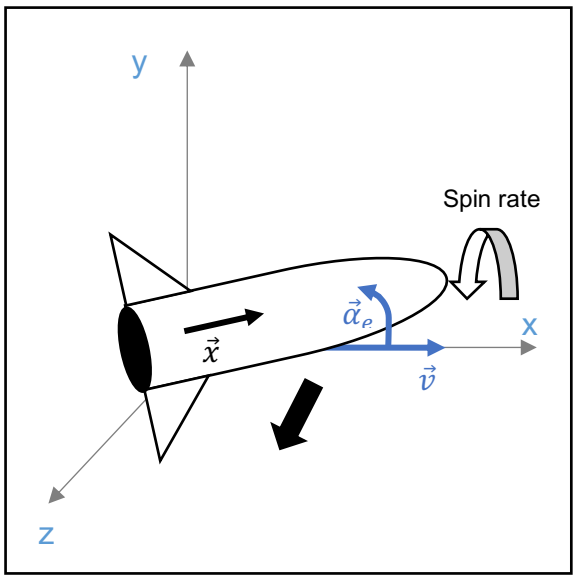


Figure 6: Magnus force where bold arrow shows the effect direction, and spin rate is along the longitudinal axis.

STANAG 4355 descripted the acceleration due to magnus for MPMM as the following equations:

$$\frac{\overrightarrow{MF}}{m} = -\frac{\pi\rho d^3}{8m} p C_{mag-f} (\vec{\alpha}_e \times \vec{v}) \tag{3.10}$$

Or for 5-DOF

$$\overrightarrow{MF} = -\frac{\pi\rho d^3}{8I_x} C_{mag-f} (\vec{H} \cdot \vec{x})(\vec{x} \times \vec{v}) \tag{3.11}$$

Where d is the projectile calibre, H is the total angular momentum of the body, I_x is the Axial moment of inertia Axial and C_{mag-f} is the aerodynamic Magnus force coefficient, which is usually a small negative quantity, p is the axial spin rate and $\vec{\alpha}_e$ is the yaw of repose. The yaw of repose is a convenient way to describe the angle of yaw along the trajectory, illustrated by this equation:

$$\vec{\alpha}_e = -8\frac{I_x p(\vec{v} \times \dot{\vec{u}})}{\pi\rho d^3 (C_{M_\alpha}) v^4} \tag{3.12}$$

Where C_{M_α} is overturning moment coefficient.

3.2.6 *Pitch damping force:*

The pitch damping force acts in the plane of transverse angular velocity, which is not necessarily the same as the yaw plane as shown in Figure 7.

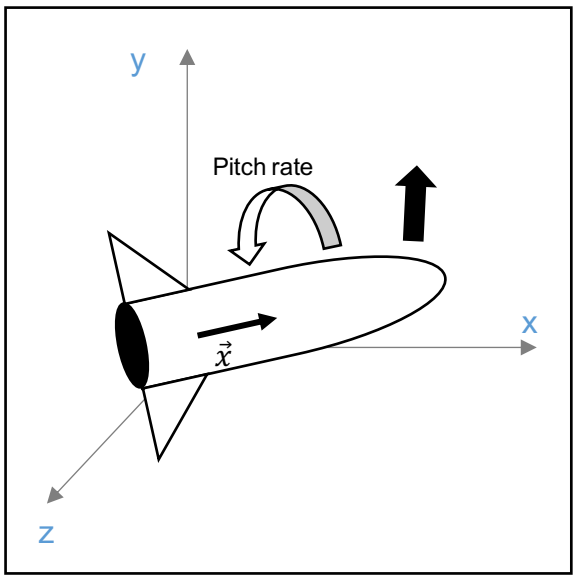


Figure 7: Pitch damping force where bold arrow shows the effect direction, and pitch rate is along the lateral axis.

The pitch damping force contains two parts; one part proportional to transverse angular velocity, and a second part proportional to the rate of change of total angle of attack.

STANAG 4355 describe for 5-DOF by the following equations:

$$\overrightarrow{PDF} = \left(\frac{\pi \rho d^3 \left(C_{N_q} + C_{N_{\dot{\alpha}}} \right)}{8 I_y} \right) v\left(\vec{H} \times \vec{x} \right) \qquad (3.13)$$

Where I_y is the transversal moment of inertia axial and the pitch damping force coefficient found from the sum of C_{N_q} and $C_{N_{\dot{\alpha}}}$.The pitch damping force acting on spin-stabilised projectiles is generally much smaller than the normal force, and few direct measurements of it have ever been made in spark photography ranges. The pitch damping force, like the Magnus force, must be retained for logical completeness, but it is usually neglected in practice.

3.2.7 *Overturning moment:*

The overturning moment is the aerodynamic moment associated where with the normal force as illustrated in Figure 8.

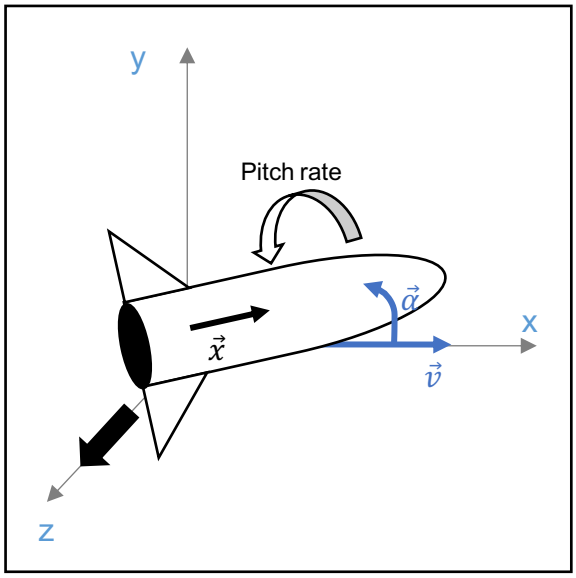


Figure 8: Overturning moment where bold arrow shows the effect direction, and pitch rate is along the lateral axis.

A positive overturning moment acts to increase the yaw angle. STANAG 4355 described it for 5-DOF by the following equations:

$$\overrightarrow{OM} = \frac{\rho d^3 \pi}{8}\left(C_{M_\alpha} + C_{M_{\alpha^3}}\alpha^2\right) v(\vec{v} \times \vec{x}) \qquad (3.14)$$

Some authors refer to the overturning moment as the "pitching moment" or "static moment". The overturning moment varies with the sine of the total yaw angle.

3.2.8 *Pitch damping moment:*

STANAG 4355 describe for 5-DOF by the following equations:

$$\overrightarrow{PDM} = \frac{\rho d^4 \pi}{8 I_y} \left(C_{M_q} + C_{M_{\dot{\alpha}}} \right) v \left[\vec{H} - \left(\vec{H} \cdot \vec{x} \right) \vec{x} \right] \qquad (3.15)$$

Pitch damping moment is always opposite to the pitch rate (therefore it is usually negative) as illustrated in Figure 9. It damps the pitch rate as its name indicate. In general, a positive pitch damping moment acts to increase the pitch rate and is therefore destabilising

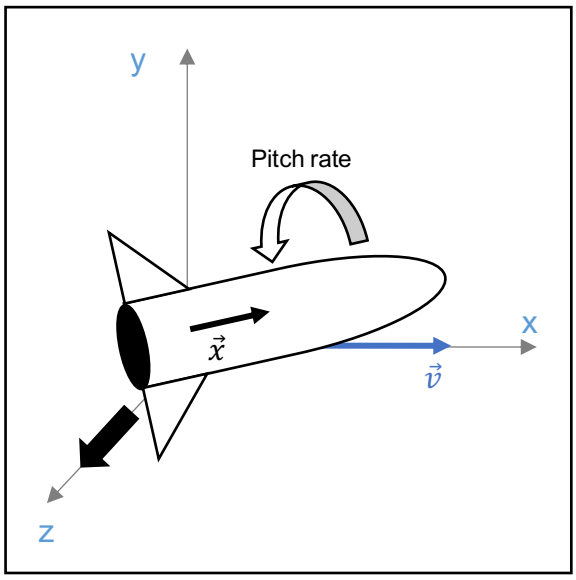


Figure 9: Pitch damping moment where bold arrow shows the effect direction, and pitch rate is along the lateral axis.

3.2.9 *Magnus moment:*

STANAG 4355 describe for 5-DOF by the following equations:

$$\overrightarrow{MM} = \frac{\rho d^4 \pi}{8 I_y} C_{mag-m}\left(\vec{H} \cdot \vec{x}\right)[(\vec{v} \cdot \vec{x})\vec{x} - \vec{v}] \qquad (3.16)$$

Although the Magnus force coefficient is usually a small value, which means the magnus force is also small enough to be often neglected, but the Magnus moment must always be considered, due to a large value of Magnus moment coefficient, either positive or negative, can have a disastrous effect on the dynamic stability as illustrated in Figure 10.

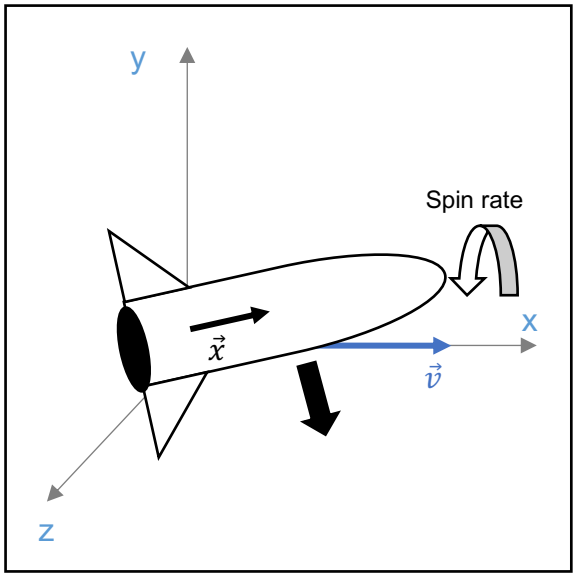


Figure 10: Magnus moment where the black bold arrow shows the effect direction, and spin rate is along the longitudinal axis.

3.2.10 *Spin damping moment:*

The spin damping moment opposes the spin of the projectile; it always reduces the axial spin, as shown in Figure 11.

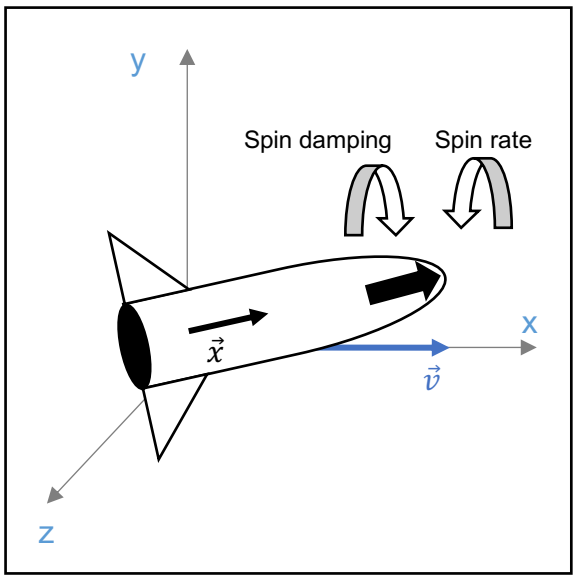


Figure 11: Spin damping moment where the black bold arrow shows the effect direction, and spin rate is along the longitudinal axis.

The vector spin damping moment is defined as in this equation:

$$\overrightarrow{SDM} = \frac{\rho d^3 \pi}{8 I_x} C_{spin} v \left(\vec{H} \cdot \vec{x}\right) \vec{x} \qquad (3.17)$$

The spin damping moment coefficient is always negative. Therefore, both the axial spin and the forward velocity decrease along the trajectory, for typical spin-stabilised projectiles. Due to the fact that the spin reduces at a slower rate (due to spin damping) then the reduction in velocity (due to drag), the gyroscopic stability usually increases along the trajectory.

3.2.11 *Forces and moments related to each trajectory model:*

The following chart (Figure 12) shows briefly the forces and moment related to each model where the five degrees of freedom contained all the previous forces and moments.

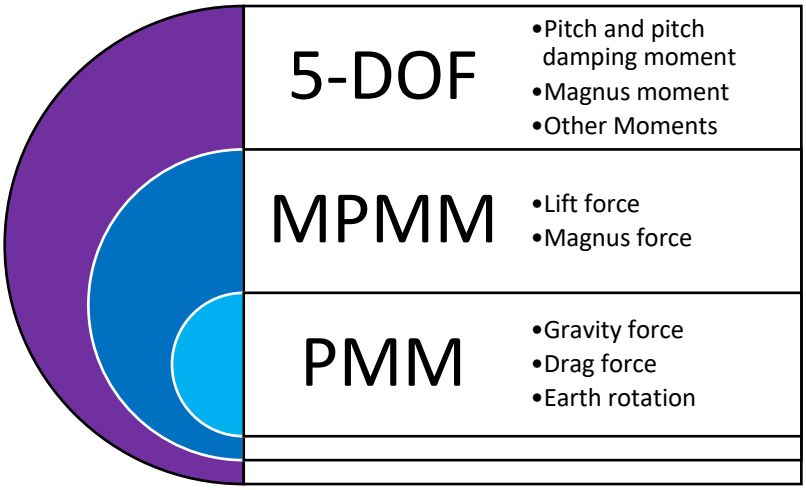


Figure 12: Relationship between models and forces

3.3 *Point mass model:*

3.3.1 *Introduction:*

This model is also known as three-degree-of-freedom. Basically the point mass model is not able to provide the drift motion due to drag and lift caused by yawing of spinning projectiles. Moreover, this model cannot describe the stability condition of a projectile in flight.

3.3.2 *Mathematical description:*

Point mass model assumes a projectile as a single point of mass. This model accounts only for the drag caused by the airflow, whilst it neglects the other aerodynamic forces that act on projectile. This model is fairly accurate and has been used by manufacturers to generate firing tables for some period of time.

3.3.2.1 Equation of motion:

Newton's law of motion for the centre of mass is;

$$\vec{F} = m\vec{a} = \overrightarrow{DF} + \vec{\Lambda} + m\vec{g} \qquad (3.18)$$

Where acceleration due to drag force is;

$$\frac{\overrightarrow{DF}}{m} = -\frac{\pi\rho d^2}{8m}\left(C_{D_o}\right)v\vec{v} \tag{3.19}$$

$$\vec{v} = \vec{u} - \overrightarrow{W} \tag{3.20}$$

And wind ($\overrightarrow{W}$) vector is;

$$\overrightarrow{W} = \begin{bmatrix} W_h \\ 0 \\ W_c \end{bmatrix} \tag{3.21}$$

Where W_h is the head wind and W_c is the cross wind.

Acceleration due to gravity is from Equation 3.1;

$$\vec{g} = -g_0\left(\frac{R^2}{r^3}\right)\vec{r} \tag{3.1}$$

3.3.3 *Simulation:*

3.3.3.1 Flow diagram

The following chart indicates the algorithm of trajectory generation by PMM

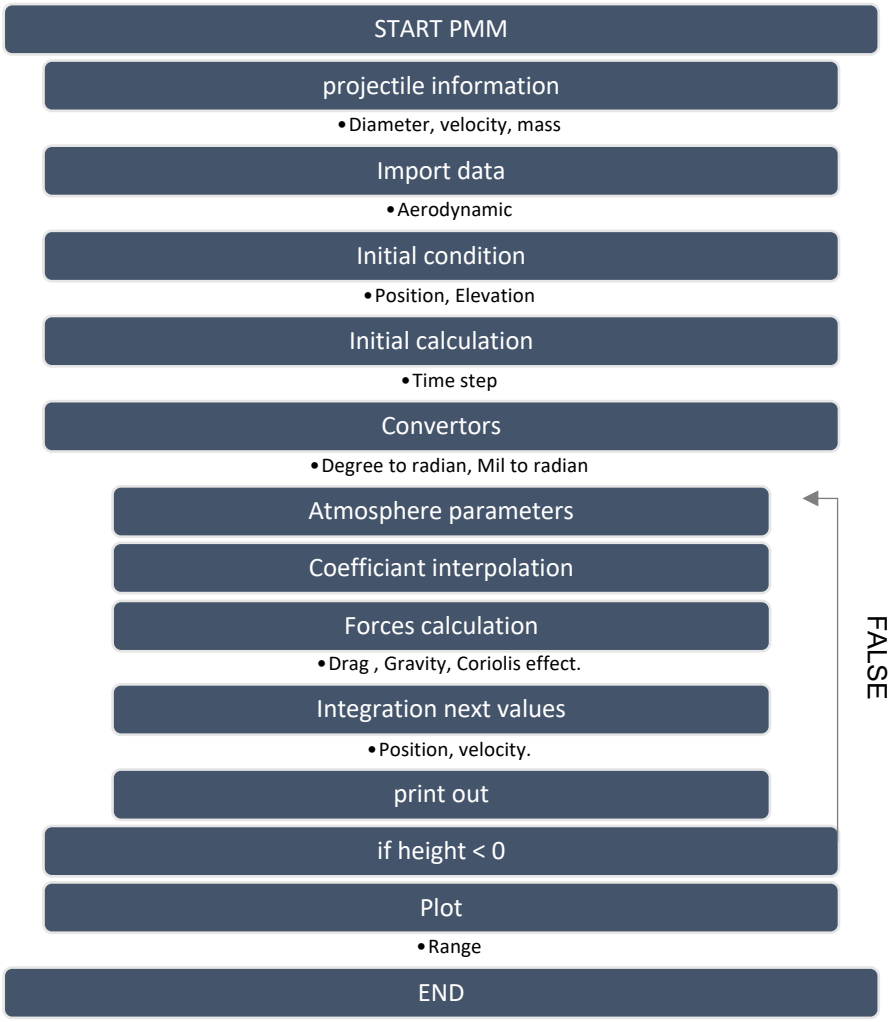


Figure 13: PMM flow diagram

3.4 *Modified point mass model:*

3.4.1 *Introduction:*

This model is also known as four-degree-of-freedom. It has been widely used in the calculation of the trajectories. One of the platforms that uses this model is NATO NABK (NATO Armament Ballistic Kernel), which is a software component that aims to standardise and reduce the effort of developing ballistic fire control related applications among member countries.

Developing the modified point mass model was aimed at to eliminating drift motion problems associated with the point mass model, which does not include yaw of repose in a fair manner.

3.4.2 *Mathematical description:*

Similar to the point-mass model, this model also assumes that all of the projectile mass is located in a single point. However, the modified point-mass includes effects of the aerodynamic forces or moments and also the effect of the earth's rotation while the projectile is travelling in the air.

This results in a very accurate model and is the standard model being used to generate firing tables. The following set of equations represents the model.

3.4.2.1 Equation of motion:

Newton's law of motion of the centre of mass of the projectile is;

$$\vec{F} = m\vec{a} = \overrightarrow{DF} + \overrightarrow{LF} + \overrightarrow{MF} + m\vec{g} + m\vec{\Lambda} \tag{3.22}$$

Where acceleration due to drag force is;

$$\frac{\overrightarrow{DF}}{m} = -\frac{\pi\rho d^2}{8m}\left(C_{D_o} + C_{D_{\alpha^2}} * {\alpha_e}^2\right) v\vec{v} \tag{3.23}$$

Acceleration due to lift force is;

$$\frac{\overrightarrow{LF}}{m} = \frac{\pi\rho d^2}{8m}\left(C_{L_\alpha} + C_{L_{\alpha^3}} * {\alpha_e}^2\right) v^2\vec{\alpha}_e \tag{3.24}$$

Acceleration due to Magnus force is;

$$\frac{\overrightarrow{MF}}{m} = -\frac{\pi\rho d^3}{8m} P C_{mag-f}(\vec{\alpha}_e \times \vec{v}) \tag{3.25}$$

Acceleration due to gravity from Equation 3.1;

$$\vec{g} = -g_0\left(\frac{R^2}{r^3}\right)\vec{r} \tag{3.1}$$

Acceleration due to the Coriolis effect gravity from Equation 3.5;

$$\vec{\Lambda} = -2(\vec{\omega} \times \vec{u}) \tag{3.5}$$

The magnitude of spin acceleration is given by;

$$P = \frac{2\pi u_o}{t_c d} + \int_0^t \dot{P} dt \tag{3.26}$$

Where the first term represents initial spin rate, achieved through grooves in the barrel, t_c is the twist of groove at muzzle;

$$\dot{P} = \frac{\pi \rho d^4 P v C_{spin}}{8 I_x} \tag{3.27}$$

The yaw of repose is given by;

$$\vec{\alpha}_e = -\frac{8 I_x P (v \times u)}{\pi \rho d^3 (C_{M_\alpha} + C_{M_{\alpha^3}} {\alpha_e}^2) v^4} \tag{3.28}$$

3.4.3 *Simulation:*

3.4.3.1 Flow diagram

The following chart (Figure 14) shows the algorithm of trajectory generation by MPMM

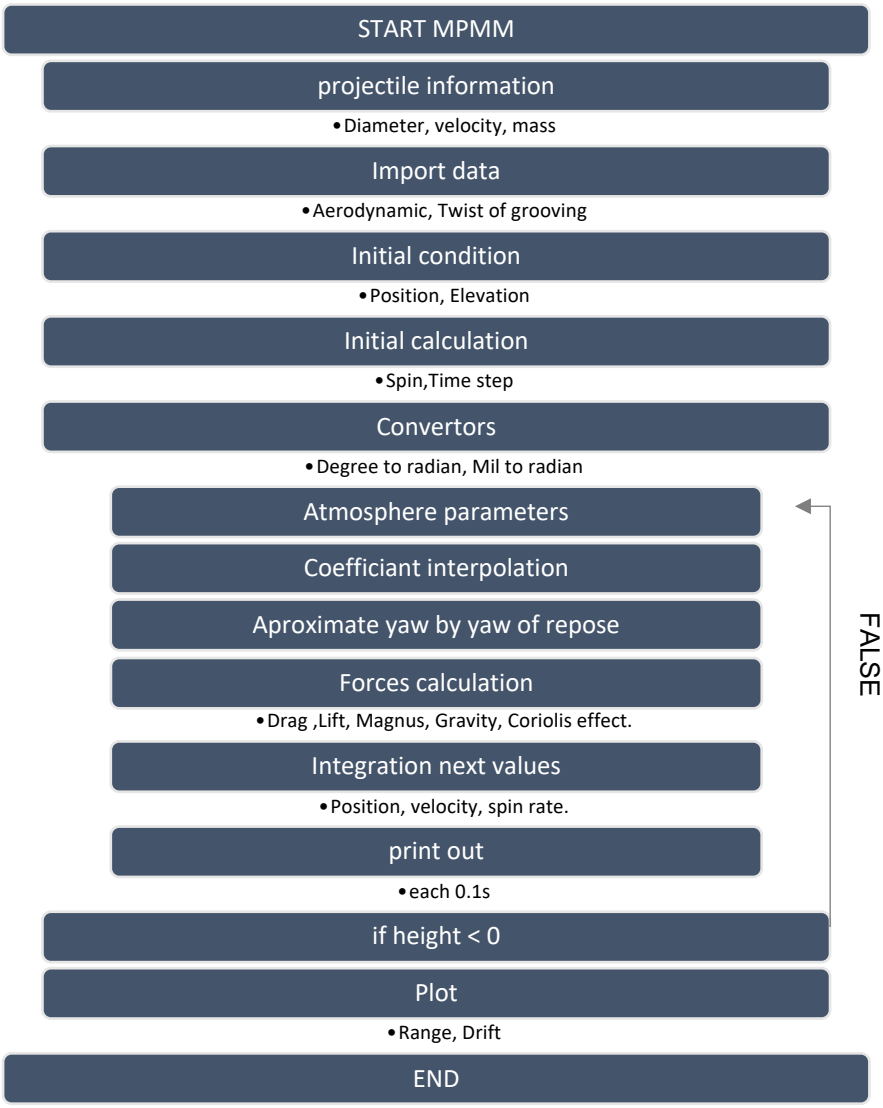


Figure 14: MPMM flow diagram

3.5 *Five degree of freedom:*

3.5.1 *Introduction:*

This model is the most complex and most accurate of all trajectory models (Nangsue, 2010). STANAG 4355 suggested using the Modified Point Mass Trajectory Model for spin-stabilised projectiles and a Five Degrees of Freedom Model for exterior ballistic trajectory simulation of fin-stabilised rockets.

3.5.2 *Mathematical description:*

3.5.2.1 Equation of motion

The motion of the centre-of-mass of the body is a summation of the accelerations acting on the body and is given by;

$$m\vec{a} = \overrightarrow{DF} + \overrightarrow{LF} + \overrightarrow{MF} + \overrightarrow{PDF} + m\vec{g} + m\vec{\Lambda} \tag{3.29}$$

Where acceleration due to drag force is;

$$\frac{\overrightarrow{DF}}{m} = -\frac{\pi\rho d^2}{8m}\left(C_{D_o} + C_{D_{\alpha^2}} * \alpha^2\right) v\vec{v} \tag{3.30}$$

Acceleration due to lift force is;

$$\frac{\overrightarrow{LF}}{m} = \frac{\pi\rho d^2}{8m}\left(C_{L_\alpha} + C_{L_{\alpha^3}} * {\alpha_e}^2\right)(v^2\vec{x} - (\vec{v}\cdot\vec{x})\vec{v}) \tag{3.31}$$

Acceleration due to Magnus force is;

$$\frac{\overrightarrow{MF}}{m} = -\frac{\pi\rho d^3}{m8I_x} C_{mag-f}\left(\vec{H}\cdot\vec{x}\right)(\vec{x}\times\vec{v}) \tag{3.32}$$

Acceleration due to pitch damping force is:

$$\frac{\overrightarrow{PDF}}{m} = \left(\frac{\pi\rho d^3\left(C_{N_q} + C_{N_{\dot{\alpha}}}\right)}{m8I_y}\right) v(H\times x) \tag{3.33}$$

Acceleration due to gravity is from Equation 3.1;

$$\vec{g} = -g_0\left(\frac{R^2}{r^3}\right)\vec{r} \tag{3.1}$$

Acceleration due to the Coriolis effect is;

$$\vec{\Lambda} = -2(\vec{\omega}\times\vec{u}) \tag{3.5}$$

The angular momentum of the body is the summation of the moments acting on the body and is given by:

$$\vec{M} = \overrightarrow{OM} + \overrightarrow{PDM} + \overrightarrow{MM} + \overrightarrow{SDM} \tag{3.34}$$

Where:

Angular moment due to Overturning Moment:

$$\overrightarrow{OM} = \frac{\rho d^3\pi}{8}\left(C_{M_\alpha} + C_{M_{\alpha^3}}\alpha^2\right) v(\vec{v}\times\vec{x}) \tag{3.35}$$

Angular moment due to Pitch Damping Moment:

$$\overrightarrow{PDM} = \frac{\rho d^4\pi}{8I_y}\left(C_{M_q} + C_{M_{\dot{\alpha}}}\right) v\left[\vec{H} - \left(\vec{H}\cdot\vec{x}\right)\vec{x}\right] \tag{3.36}$$

Angular moment due to Magnus Moment:

$$\overrightarrow{MM} = \frac{\rho d^4 \pi}{8 I_y} C_{mag-m}\left(\vec{H} \cdot \vec{x}\right)[(\vec{v} \cdot \vec{x})\vec{x} - \vec{v}] \tag{3.37}$$

Angular moment due to Spin Damping Moment:

$$\overrightarrow{SDM} = \frac{\rho d^3 \pi}{8 I_x} C_{spin} v\left(\vec{H} \cdot \vec{x}\right)\vec{x} \tag{3.38}$$

3.5.3 *Simulation:*

3.5.3.1 Flow diagram

The following chart shows the algorithm of trajectory generation by 5-DOF

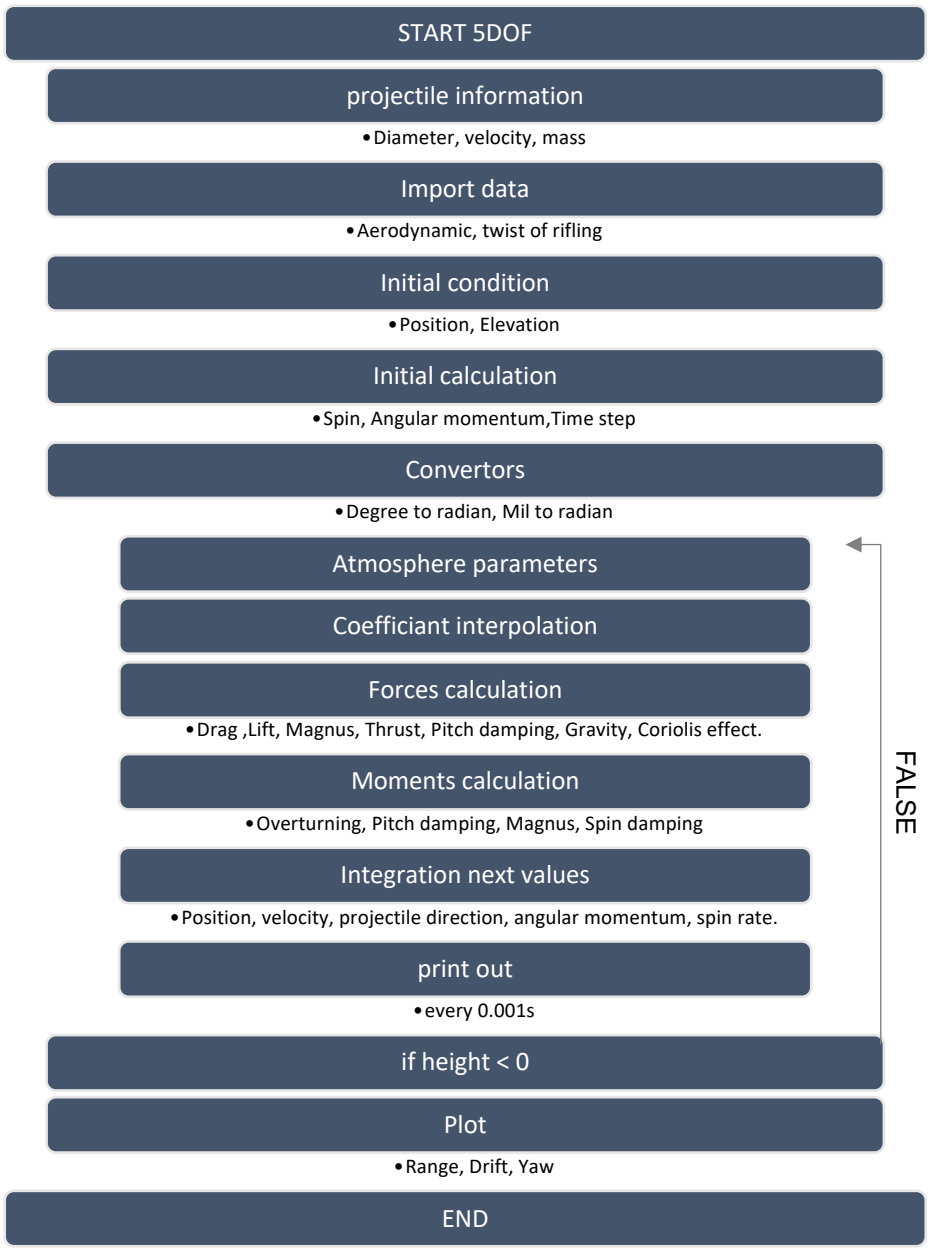


Figure 15: five degree of freedom flow diagram

3.6 *Verification*

Before the verification of the various models are presented, it is necessary to introduce the concept of “degree of similarity”.

3.6.1 *Degree of similarity:*

It is a way to measure similarity between the external ballistic performances of the two configurations as suggested in STANAG 4106, which is basically divide acceptance into three levels:

I. Match: if none of the differences in range and drift between reference (R) and test (T) samples are greater than one probable error in range and drift, then ballistic match is achieved.

II. If none of the differences in range and drift between (R) and (T) samples are greater than 0.95% of range in range and 0,3% of range in drift, then ballistic similitude (1%) can be achieved.

III. If none of the differences in range and drift between (R) and (T) samples, without corrections, are greater than 4.75% of range in range and 1,5% of range in drift, then ballistic similitude (5%) can be achieved.

Guided by this similarity measurement between two simulations working with the same Aerodynamic data, level 2 will be used as the limit for acceptance.

3.6.2 *Results*

To verify the correctness of the programs before embarking on using it for case studies, the verification will go through two stages:

I. Analytically: Comparing results for a case where an analytic solution is available.

II. Numerically: Comparing results with available commercial simulation code PRODAS (Projectile Design/Analysis System), details shown in Appendix 2.

This will be done for three projectiles:

- Fin-stabilised mortar (81 mm), details provided in Appendix 2.
- Spin-stabilised low muzzle velocity (105 mm), details provided in Appendix 2.
- Spin-stabilised high muzzle velocity (155 mm), details provided in Appendix 2.

I. Analytically:

First, compare the predicted results using all models (while in vacuum they will give same results) with analytic results for a vacuum trajectory and constant gravity as shown in Table 3 for the range and Table 4 for the TOF (Time of Flight). The following formula could represent the analytical result , where V is the initial velocity, θ is the elevation and G is the gravity constant (Regan, 1984):

$$\text{Range} = \frac{V^2 sin2\theta}{G} \quad (3.39)$$

$$\text{Time of flight} = \frac{2Vsin\theta}{G} \quad (3.40)$$

Table 3: Range-comparison (m) for 45º elevation

	81mm	**105mm**	**155mm**
Constant gravity (PMM, MPMM, 5-DOF)	7992.85	25245.69	32898.06
Analytical	7994.6	25248.76	32898.49
Error %	0.02%	0.01%	0.001%

Table 4: TOF-comparison (s) for 45º elevation

	81mm	**105mm**	**155mm**
Constant gravity (PMM, MPMM, 5-DOF)	40.370	71.749	81.910
Analytical	40.378	71.759	81.911
Error %	0.02%	0.01%	0.001%

The error in Tables 3 and 4 show good degree of similarity. This confirms the correct implementation of integration scheme.

Secondly, Compare the predicted results with the analytic results for a vacuum trajectory with constant gravity and rotating earth as shown in Table 5, The analytical result could be represented by the following formula where La is the initial latitude (Regan, 1984):

$$\text{Drift}=\mathrm{V}\cos(\theta)\sin(\mathrm{Az})\,\mathrm{TOF}-\omega\cos(\mathrm{La})\,\mathrm{TOF}^2\left(\mathrm{V}\sin(\theta)-\frac{\mathrm{Gt}}{3}\right)+\mathrm{V}\omega\mathrm{TOF}^2\sin(\mathrm{La})\cos(\theta)\cos(\mathrm{Az}) \tag{3.41}$$

Table 5: Drift-comparison (m) 45^o elevation

	81mm	**105mm**	**155mm**
Constant gravity	-7.850	-44.04	-65.50
Analytical	-7.844	-44.03	-65.48
Error %	0.07%	0.02%	0.03%

The error in Table 5 is less than 1% which is accepted degree of similarity. This verify the correct implementation of Earth rotation (Coriolis effects).

II. Numerically:

Used a set of aerodynamic coefficients of the three projectiles and compared the predicted results with the results obtained with PRODAS, note that elevation (slightly above 45^o) was used to coincide with an evaluation in PRODAS's firing table that is sorted by the ranges. The results are shown in Table 6 for range, Table 7 for TOF, Table 8 for drift Table 9 for the angle of impact and Table 10 for trajectory maximum height

Table 6: Range-comparison (m) for ~45º elevation

Projectile	**81mm**			**105mm**			**155mm**		
Model	PMM	MPMM	5DOF	PMM	MPMM	5DOF	PMM	MPMM	5DOF
Results	4829.4	4830	4843.1	11483	11561	11568	14552	14604	14596
PRODAS	4800			11500			14500		
Error%	0.6%	0.6%	0.9%	0.1%	0.5%	0.6%	0.35%	0.7%	0.66%

Table 7: TOF-comparison (sec) for ~45º elevation

Projectile	**81mm**			**105mm**			**155mm**		
Model	PMM	MPMM	5DOF	PMM	MPMM	5DOF	PMM	MPMM	5DOF
Results	34.68	34.67	34.83	47.18	47.66	47.71	56.7	57.08	57.05
PRODAS	34.8			47.24			56.747		
Error%	0.3%	0.3%	0.1%	0.1%	0.9%	0.9%	0.08%	0.6%	0.5%

Table 8: Drift-comparison (m) for ~45º elevation

Projectile	**81mm**			**105mm**			**155mm**		
Model	PMM	MPMM	5DOF	PMM	MPMM	5DOF	PMM	MPMM	5DOF
Results	**	**	**	*	27.37	26.12	*	34.72	28.78
PRODAS	**			20.948			31.48		
Error%	**	**	**	*	0.06%	0.05%	*	0.02%	0.02%

(**) No drift calculation on mortar regardless earth rotation.

(*) There is no drift calculation within PMM.

Table 9: Impact Angle-comparison (degree) for ~45^o elevation

Projectile	**81mm**			**105mm**			**155mm**		
Model	PMM	MPMM	5DOF	PMM	MPMM	5DOF	PMM	MPMM	5DOF
Results	-56.6	-56.6	-56.6	-48.5	-48.5	-48.6	-53.3	-53.4	-53.4
PRODAS	-56.6			-48.48			-53.22		
Error%	0.04%	0.04%	0.04%	0.06%	0.12%	0.2%	0.1%	0.3%	0.4%

Table 10: max height-comparison (m) for ~45^o elevation

Projectile	**81mm**			**105mm**			**155mm**		
Model	PMM	MPMM	5DOF	PMM	MPMM	5DOF	PMM	MPMM	5DOF
Results	1478	1478	1484	2805	2824	2826	4088	4106	4109
PRODAS	1480.5			2809.8			4093.3		
Error%	0.2%	0.2%	0.2%	0.17%	0.5%	0.6%	0.13%	0.3%	0.4%

The small difference between the simulation results from PMM, MPMM, 5-DOF with both of analytic solution and the results from PRODAS (as shown in Tables 6-10) provide confidence to proceed with case studies.

3.7 *Effect of Integration step on accuracy:*

The present research focusses on the comparison of trajectory models, therefore, the same (relatively simple) integration method was used for all three trajectory models which is Euler's forward method. Optimising the integration method, using Runge-Kutta or other higher order schemes will definitely affect the simulation time, but that was not considered in this study and may be considered in future research.

The focus in this section aims to determine how the integration step (time step) affect the accuracy of the trajectory path and investigate the possibility to introduce dynamic time step adjustment. Secondly, research focus is placed on identifying a relationship between the error and the integrated variable (acceleration, spin, projectile direction and moment). This gives an indication of the simulation stability to be used in defining the threshold of the dynamic time step. The main advantage of the dynamic time step is to reduce the total simulation time. This mainly gives improvement for the 5-DOF because it is the most time-consuming model.

A similar study was done by Baranowski in 2013, to look at the influence of similar models and integration step on the simulation accuracy. That research concludes with the following:

- To obtain computation accuracy of 0.1% for drift and 0.02% for the remaining parameters, the model based on the body axis system (similar to five degree of freedom) should use an integration step enabling approximately 100 computations per projectile revolution (Baranowski, 2013).

The question is if spin integration is the only factor to consider in selecting the timestep, or are there other factors that should also be taken into account when selecting the optimum timestep.

3.7.1 *Fin-stabilised (81 mm) mortar projectile:*

For a mortar as shown in the Table 11, there is no relation with the integrated variables due to the way of stability is not affected by these variables, therefore it is recommended to work with static time step $3x10^{-3}$.

Table 11: Time step vs range accuracy for 81mm mortar

Time-step (Sec x10^{-3})	**Range (m)**	**Error (%)**	**Max integration during flight for**		
			Velocity (m/s^2)	Direction (rad/s)	Moment (kg m^2 rad/s^2)
1	4842.9	0.006%	19.32	0.0715	0.021738
2	4842.6	0.012%	19.32	0.0715	0.022372
3	4842.3	0.019%	19.32	0.0715	0.023029
5	4841.7	0.031%	19.32	0.0716	0.024387
8	4840.9	0.047%	19.32	0.0716	0.026824

3.7.2 *Spin-stabilised (105 mm):*

In this case as shown in Table 12, there is a direct relationship between the integration timestep and the error increasing. The simulation is stable only when the direction integration is less than 1 rad/s with margin up to probably 4 rad/s.

Table 12 :Time step vs drift accuracy for 105mm

Time-step (Sec x10^{-4})	**Drift (m)**	**Error (%)**	**Max integration during flight for**			
			Velocity (m/s^2)	Spin (rad/s^2)	Direction (rad/s)	Moment (kg m^2 rad/s^2)
1	274.05	0.00%	40.73	25.33	0.50	7.27
1.1	274.06	0.00%	40.73	25.33	0.55	7.30
1.11	274.06	0.00%	40.73	25.33	0.71	7.31
1.16	274.21	0.06%	40.73	25.33	2.72	7.39
1.17	274.32	0.10%	40.73	25.33	3.58	7.54
1.18	274.50	0.16%	40.73	25.33	4.69	7.68
1.19	274.82	0.28%	40.73	25.33	6.10	7.77
1.20	275.33	0.47%	40.73	25.33	7.87	8.10
1.30	322.39	17.6%	40.73	25.33	32.14	45.48

3.7.3 *Spin-stabilised (155 mm):*

In this case, as shown in Table 13, there is a clear relationship between the direction integration and the error. The simulation is stable only when the direction integration is less than 1 rad/s with margin up to probably 5 rad/s.

Table 13: Time step vs drift accuracy for 155mm

Time-step (Sec x10^{-4})	**Drift (m)**	**Error (%)**	**Max integration during flight for**			
			Velocity (m/s^2)	Spin (rad/s^2)	Direction (rad/s)	Moment (kg m^2 rad/s^2)
1.1	457.16	0.00%	36.09	14.67	0.05	5.50
1.2	457.16	0.00%	36.09	14.67	0.20	5.50
1.3	457.11	0.01%	36.09	14.67	1.60	10.33
1.35	456.74	0.09%	36.09	14.67	4.74	26.53
1.36	456.52	0.14%	36.09	14.67	5.86	32.34
1.37	456.20	0.21%	36.09	14.67	7.23	39.31
1.4	454.00	0.69%	36.09	14.67	12.94	66.98
10	12.30	97.31%	379.79	14.67	110.17	1062.92

Finally, this section shows that for spin-stabilized projectiles time step value can be related to the value of direction integration (rate of change of unit vector along the longitudinal axis of the body) in five degree of freedom as mention in (Baranowski, 2013), Therefore, instead of using static time steps, it is better to work with dynamic time steps that changes every iteration based on direction integration. The rule is that the time step should be as large as possible while direction integration is less than one.

Chapter 4 Results and discussion

4.1 *Introduction:*

This chapter focusses on the final result of firing tables for the three projectiles, specifically, Table F which is the heart of the firing table and contain all the basic information that rely on the projectile model and corrections to range for nonstandard conditions. The American artillery manual (FM 6-40) describe it as “lists information needed to determine firing data to attack a target and for solving concurrent and subsequent met”, Other tables are attached in Appendix 3. It is assumed that the five degree of freedom model is the most accurate model because it models forces and effects accurately. Thus the comparison of the other models will be against the 5-DOF as a reference. The chapter will conclude with recommendations regarding the optimal trajectory model for different projectile types.

4.2 *Implementation:*

The whole process that is used in generating the firing table is shown in Figure 19, The code written by the author as shown in Appendix 4 by using MATLAB.

Figure 16: Firing Table Generation Process

4.2.1 *Inputs:*

The primary data required to start any model is the aerodynamics coefficients for the projectile which can be generated using CFD (computational fluid dynamics) software, or empirically as shown in Figure 20, Also, the geometric characteristics such as moments of inertia, finally, the initial condition to start with.

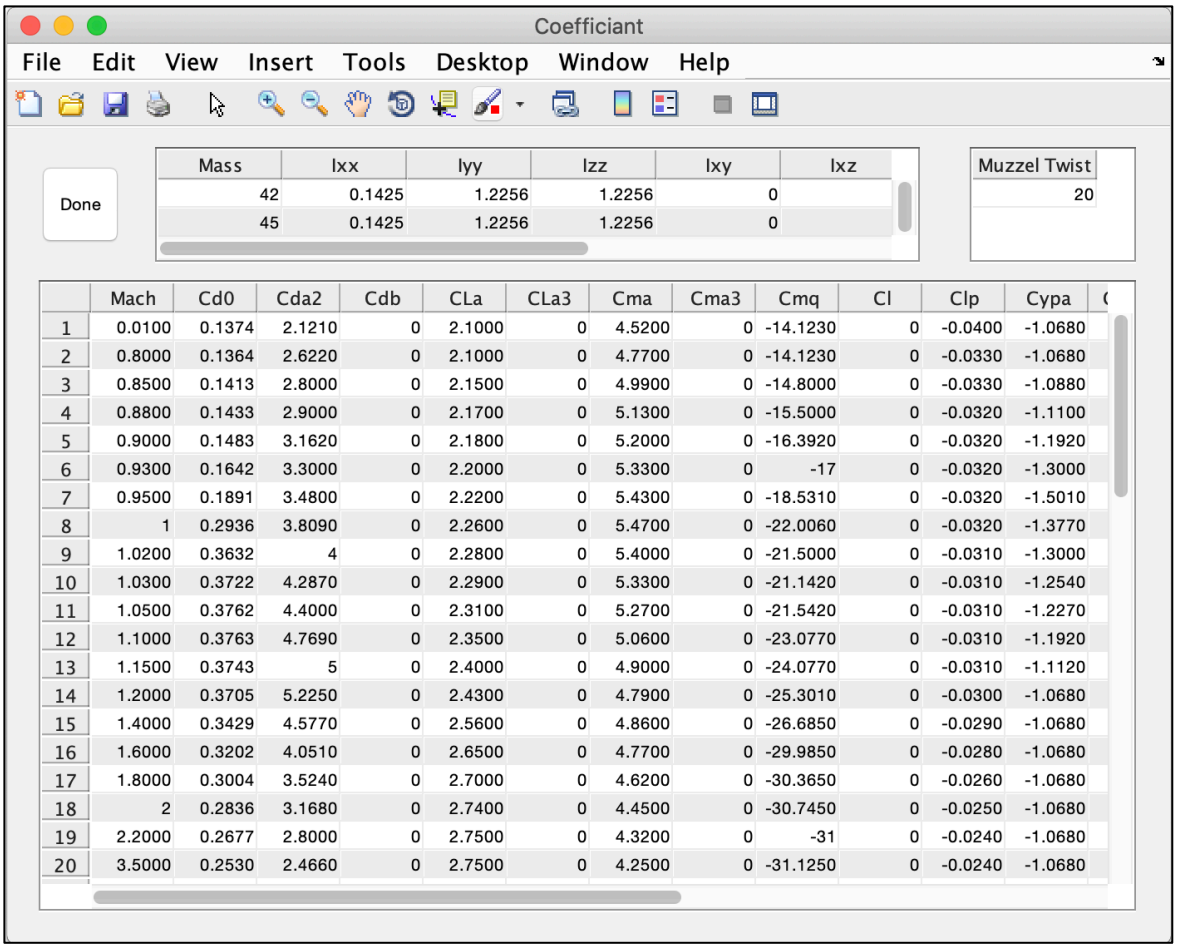


Figure 17: The User Interface for defining projectile properties

4.2.2 *Data generation:*

This stage starts with the selection of trajectory model such as PMM, MPMM or 5-DOF as evident in the following figure (Figure 18) to generate all the data of the firing table. This stage is the most time-consuming one.

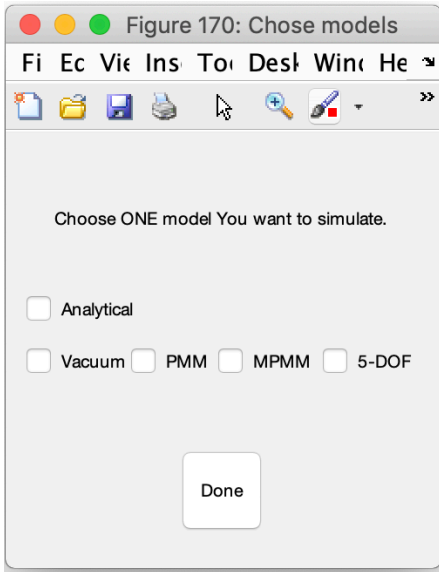


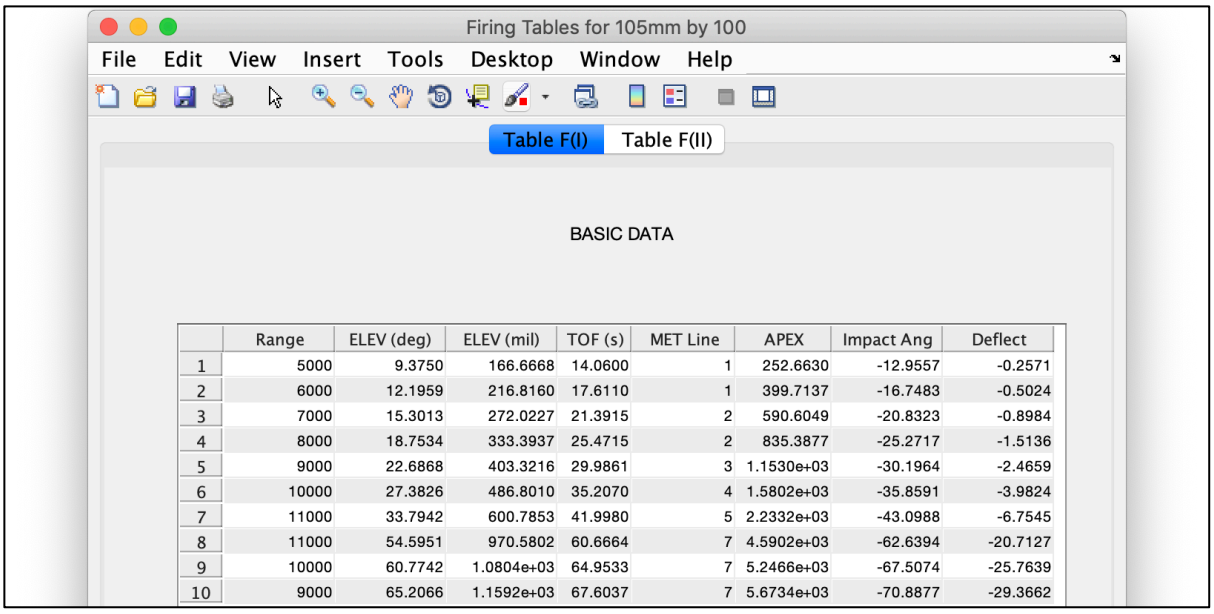
Figure 18: Model selection

4.2.3 *Fitting process:*

As suggested in (Baranovski, 2013b) that “approximation of the fitting factors by the cubic polynomial generally gives better results than approximation by the quadratic polynomial”. However, In this study it is important to utilize one fitting process for all the models to be relevant.

4.2.4 *Printing:*

The firing table, either in printed or digital format, forms the user interface to mission planning and laying of the weapon in preparation for firing. This data is presented as a function of range as illustrated in Figure 19. The data shown cover from the minimum elevation (around 8 degree) to the maximum (around 70 degree) with reasonable intervals.



	Range	ELEV (deg)	ELEV (mil)	TOF (s)	MET Line	APEX	Impact Ang	Deflect
1	5000	9.3750	166.6668	14.0600	1	252.6630	-12.9557	-0.2571
2	6000	12.1959	216.8160	17.6110	1	399.7137	-16.7483	-0.5024
3	7000	15.3013	272.0227	21.3915	2	590.6049	-20.8323	-0.8984
4	8000	18.7534	333.3937	25.4715	2	835.3877	-25.2717	-1.5136
5	9000	22.6868	403.3216	29.9861	3	1.1530e+03	-30.1964	-2.4659
6	10000	27.3826	486.8010	35.2070	4	1.5802e+03	-35.8591	-3.9824
7	11000	33.7942	600.7853	41.9980	5	2.2332e+03	-43.0988	-6.7545
8	11000	54.5951	970.5802	60.6664	7	4.5902e+03	-62.6394	-20.7127
9	10000	60.7742	1.0804e+03	64.9533	7	5.2466e+03	-67.5074	-25.7639
10	9000	65.2066	1.1592e+03	67.6037	7	5.6734e+03	-70.8877	-29.3662

Figure 19: Example of printed table F

4.3 *Results:*

The most important data used in mission planning is provided in the so-called Table F. This table consist of two tables. Part 1 provides the basic data, and part 2 provides the correction of range for variations in flight conditions.

4.3.1 *Table F (I):*

Table F(I) to give all the basic data of the trajectory such as range, elevation, TOF, etc. For each table the first section present data as predicted by using the 5-DOF model then MPMM and PMM. At the bottom of the second and the third sections is the maximum error of data presented by these models, a maximum error is shown, Where*:*

$$Error = \frac{X_{model} - X_{5DOF}}{X_{5DOF}} \qquad (4.1)$$

X represent any of trajectory parameters. For conditions where the error is large but acceptable, a specific cell is highlighted in yellow. If the error is unacceptable high it is highlighted in red.

4.3.1.1 Mortar bomb (81 mm) :

The mortar bomb case is a typical example of a fin-stabilised projectile in ballistic flight, details shown in Appendix 2.

Table 14: Table F(I) for 81 mm mortar

5-DOF						
Range (m)	ELEVATION (DEG)	TOF (S)	MET LINE	APEX (m)	IMPACT ANGEL (DEG)	DEFLECT (m)
2000	8.66	8.26	0	83.38	-10.21	-0.06
2500	11.46	10.76	0	141.2	-14.00	-0.13
3000	14.65	13.5	1	222.5	-18.50	-0.24
3500	18.39	16.6	1	336.4	-23.89	-0.43
4000	23.02	20.3	2	501.1	-30.57	-0.76
4500	29.58	25.1	2	768.7	-39.55	-1.39
4800	47.07	35.9	4	1581.9	-58.46	-3.98
4300	57.55	41.1	5	2056.8	-66.95	-5.98
3800	63.47	43.3	5	2295	-71.23	-7.16
3300	68.16	44.9	5	2460.8	-74.50	-8.06
MPMM						
2000	8.72	8.26	0	83.89	-10.28	-0.06
2500	11.50	10.73	0	141.44	-14.05	-0.13
3000	14.71	13.48	1	223.17	-18.57	-0.24
3500	18.51	16.60	1	338.90	-24.05	-0.44
4000	23.15	20.24	2	503.36	-30.72	-0.77
4500	29.80	25.11	2	775.19	-39.83	-1.41
4800	46.70	35.62	4	1559.30	-58.13	-3.91
4300	57.55	40.92	5	2052.10	-66.97	-5.99
3800	63.52	43.29	5	2292.60	-71.28	-7.18
3300	68.00	45.00	5	2458.00	-75.00	-8.00
Max Error	0.8%	0.34	0%	1.4%	0.7%	0%
PMM						
2000	8.70	8.24	0	83.42	-10.24	-0.06
2500	11.51	10.73	0	141.35	-14.06	-0.13
3000	14.73	13.48	1	223.18	-18.57	-0.24
3500	18.50	16.58	1	337.90	-24.01	-0.44
4000	23.17	20.25	2	503.71	-30.75	-0.77
4500	29.83	25.12	2	775.63	-39.85	-1.42
4800	46.65	35.59	4	1555.7	-58.09	-3.90
4300	57.54	40.90	5	2049.8	-66.95	-5.99
3800	63.51	43.27	5	2290.6	-71.27	-7.18
3300	68.00	45.00	5	2456.0	-75.00	-8.00
Max Error	0.9%	0.38	0%	1.7%	0.0%	0%

Table 14 shows the following:

- Low muzzle velocity = 280 m/s, Azimuth = 0 deg
- Minimum range is 2000m and maximum range is 4800m with 500m interval.
- The mortar is a fin-stabilised projectile so the deflect (drift) is only effected by earth rotation.

MPMM section shows the following:

- The maximum error appear between 4500m and the maximum range (4800m), however, it is less than 1% except in apex.
- Max error in apex is 1.4%, but acceptable since range and deflect are almost accurate.

PMM section shows the following:

- The maximum error appear between 4500m and the maximum range (4800m), however, it is less than 1% except in apex.
- Max error in apex is 1.7%, but acceptable since range and deflect are almost accurate.

4.3.1.2 Artillery (105 mm) :

The 105 mm artillery case is a typical example of a spin-stabilised projectile used for relatively short ranges, details shown in Appendix 2.

Table 15: Table F(I) for 105 mm artillery

5-DOF						
Range (m)	ELEVATION (DEG)	TOF (S)	MET LINE	APEX (m)	IMPACT ANGEL (DEG)	DEFLECT (m)
5000	9.39	14.17	1	255.04	-12.98	22.83
6000	12.18	17.70	1	401.28	-16.75	34.29
7000	15.27	21.49	2	592.44	-20.83	48.95
8000	18.74	25.62	2	839.84	-25.30	67.52
9000	22.63	30.11	3	1155.50	-30.19	91.26
10000	27.33	35.36	4	1584.30	-35.86	123.06
11000	33.72	42.18	5	2238.50	-43.12	171.81
11000	54.44	61.00	7	4597.40	-62.74	365.87
10000	60.47	65.30	7	5243.30	-67.61	432.38
9000	64.78	68.01	7	5664.50	-71.08	485.08
MPMM						
5000	9.31	14.06	1	250.95	-12.86	24.25
6000	12.09	17.60	1	396.04	-16.62	36.86
7000	15.17	21.38	2	585.67	-20.67	53.31
8000	18.59	25.45	2	828.08	-25.07	74.49
9000	22.49	29.97	3	1142.90	-29.98	102.03
10000	27.12	35.17	4	1564.80	-35.59	139.35
11000	33.43	41.83	5	2196.70	-42.65	196.67
11000	54.81	61.43	7	4641.10	-63.03	445.26
10000	60.73	65.66	7	5271.50	-67.81	529.53
9000	64.96	68.34	7	5682.10	-71.23	599.56
Max Error	0.9%	0.43	0%	1.9%	1%	1.3%
PMM						
5000	9.38	14.06	1	252.66	-12.96	-0.26
6000	12.19	17.61	1	399.62	-16.75	-0.50
7000	15.30	21.39	2	590.60	-20.83	-0.90
8000	18.75	25.47	2	835.38	-25.27	-1.51
9000	22.69	29.99	3	1153.00	-30.20	-2.47
10000	27.38	35.21	4	1580.20	-35.86	-3.98
11000	33.79	42.00	5	2233.20	-43.10	-6.75
11000	54.60	60.67	7	4590.30	-62.64	-20.71
10000	60.77	64.95	7	5246.70	-67.51	-25.77
9000	65.21	67.60	7	5673.40	-70.89	-29.37
Max Error	0.7%	0.41	0%	0.9%	0.3%	106%

Table 15 shows the following:

- Low muzzle velocity = 497 m/s, Azimuth = 0 deg
- Minimum range is 5000m and maximum range is 11000m with 1000m interval.

MPMM section shows the following:

- The maximum error appear with the maximum range (11000m), however, it is less than 1% except for apex.
- Max error in apex is 1.9, but acceptable since range and deflect are almost accurate.
- This study also noticed increasing error in deflecting with high elevation as seen by (Pope, 1985).

PMM section shows the following:

- The PMM cannot predict projectile drift due to yaw and can only predict drift due to the earth rotation effect. This mean PMM is not a reliable model for spin-stabilized projectiles.

4.3.1.3 Artillery (155 mm) :

The 155 mm artillery case is a well-known spin-stabilised projectile used for relatively long ranges. With muzzle velocity reach 900 m/s and more with base-bleed, details shown in Appendix 2.

Table 16: Table F(I) for 155 mm artillery

5-DOF						
Range (m)	ELEVATION (DEG)	TOF (S)	MET LINE	APEX (m)	IMPACT ANGEL (DEG)	DEFLECT (m)
14000	11.37	28.51	3	1006.00	-22.74	76.15
16000	15.00	35.37	4	1584.50	-30.02	113.81
18000	19.37	42.97	5	2389.00	-37.29	164.13
20000	24.47	51.41	7	3453.00	-44.33	230.82
22000	30.66	61.39	7	4899.20	-51.49	323.36
24000	39.60	75.57	7	7224.60	-59.91	479.40
24000	55.99	99.87	7	11887.00	-71.08	869.39
22000	62.54	107.75	7	13692.00	-74.43	1064.10
20000	66.49	111.91	7	14697.00	-76.61	1166.70
18000	69.47	114.91	7	15390	-78.57	1205.10
MPMM						
14000	11.32	28.40	3	997.55	-22.58	91.71
16000	14.95	35.27	4	1573.10	-29.83	140.63
18000	19.26	42.81	5	2364.50	-37.01	207.14
20000	24.31	51.20	7	3415.10	-44.02	296.25
22000	30.45	61.13	7	4843.60	-51.15	420.16
24000	39.15	74.76	7	7059.90	-59.31	622.48
24000	56.50	100.77	7	12030.00	-71.31	1169.70
22000	62.78	108.34	7	13754.00	-74.53	1409.60
20000	66.74	112.55	7.00	14760.00	-76.69	1558.90
18000	70.44	116.38	7.00	15678.00	-78.74	1696.40
Max Error	1.4%	0.01	0%	2.3%	1.0%	2.7%
PMM						
14000	11.42	28.40	3	1006.90	-22.82	1.54
16000	15.10	35.30	4	1592.70	-30.20	2.83
18000	19.48	42.86	5	2397.50	-37.45	4.98
20000	24.64	51.32	7	3470.80	-44.55	8.53
22000	30.94	61.38	7	4939.90	-51.78	14.72
24000	40.06	75.64	7	7298.50	-60.24	28.28
24000	55.96	98.99	7	11810.00	-70.98	69.04
22000	62.78	107.01	7	13677.00	-74.32	92.52
20000	67.00	111.26	7	14740.00	-76.44	107.42
18000	70.96	115.14	7	15723.00	-78.46	121.38
Max Error	2.2%	0.01	0%	2.2%	0.6%	98.4%

Table 16 shows the following:

- High muzzle velocity = 900 m/s, Azimuth = 0 deg
- Minimum range is 12000m and maximum range is 24000m with 2000m interval

MPMM section shows the following:

- The maximum error appear with the maximum elevation 70.44^{o} (18000m), where the error is not acceptable (1.4%).
- Max error in apex is 2.3%, but acceptable while range and deflect are accurate.
- For this high speed spin-stabilized projectile this study noticed a large error in deflection at high elevation as seen by (Pope, 1985).

PMM section shows the following:

- The PMM cannot predict projectile drift due to yaw and can only predict drift due to the earth rotation effect. This mean PMM is not a reliable model for spin-stabilized projectiles, but acceptable for fin-stabilized projectiles as seen is section 4.3.1.1.

4.3.2 *Table F (II):*

Table F(II) to gives the range correction data for variations in non-standard flight conditions. For each table the first section give the data as predicted by the 5-DOF trajectory model then MPMM and PMM. At the bottom of the second and the third sections is the maximum error of data presented by these models, a maximum error is shown for each column, Where:

$$Error = \frac{X_{model} - X_{5DOF}}{Range_{5DOF}} \qquad (4.2)$$

The yellow highlighted cell if the error high but acceptable while red highlighted when it is not acceptable.

4.3.2.1 Mortar bomb (81 mm) :

Initial condition and more details shown in Appendix 2.

Table 17: Table F(II) for 81 mm mortar

5-DOF									
Range (m)	VELOCITY (±1M/S)		WIND (±1 KNOT)			TEMPERATURE (±1%)		DENSITY (±1%)	
	Inc	Dec	Tail	head	cross	Inc	Dec	Inc	Dec
2000	13.8	-11.3	-12.9	15.4	0.2	2.9	-0.5	-0.4	3
2500	16.3	-12	-17.4	21.6	1.1	6.2	-1.9	-1.9	6.3
3000	17.8	-13.4	-29.6	30.1	1.2	9	-4.6	-4.5	9.1
3500	17.3	-16.8	-39.2	43.3	2.8	10.1	-6.1	-6	10.2
4000	19.9	-17.1	-52.5	55.2	0.2	11.4	-12	-11.9	11.6
4500	19.7	-19.7	-67.1	70.2	-1.4	13.9	-13.9	-13.7	14.1
4800	20.6	-23.1	-86.6	86.3	-2.4	17.4	-19.9	-19.7	17.6
4300	20	-17	-78.5	81.5	0.7	16.1	-16.8	-16.6	16.3
3800	18.2	-15.4	-65.6	71	-0.5	17.3	-14.5	-14.3	17.4
MPMM									
2000	12.4	-7.5	-31.3	35.3	2.5	4.8	0.1	0.2	4.8
2500	15.3	-10.5	-33.7	33.6	0.4	3.9	-3.1	-3.1	4
3000	15	-16.1	-35.1	33	-0.5	8.1	-5.6	-5.5	8.2
3500	18.7	-17	-32.4	36.5	2.6	9.3	-7.6	-7.6	9.4
4000	18.2	-18.3	-37	36	-1.5	10.3	-10.4	-10.4	10.4
4500	20.5	-18.9	-35.4	36	0.8	14.7	-13.1	-13	14.8
4800	21	-20.7	-38.9	38.2	-0.8	18.6	-18.3	-18.2	18.7
4300	19.1	-17.9	-38.3	38.5	0.6	17.6	-16.5	-16.4	17.7
3800	17.1	-16.3	-39.4	38	-0.2	14.6	-15.1	-15	14.7
Max Error	0.1%	0.2%	1.0%	1.0%	0.1%	0.1%	0%	0%	0.1%
PMM									
2000	12	-10.1	1.3	2.9	2.1	4.4	-2.4	-2.4	4.4
2500	12.7	-13.1	-1.3	1	-0.1	5.4	-3.7	-3.7	5.4
3000	15.9	-15.1	-1.1	2	0.5	7.3	-6.4	-6.4	7.3
3500	17.7	-16.2	-2.1	3.6	1.6	10.1	-8.6	-8.5	10.1
4000	18.2	-19.7	-2.4	2.5	0.1	11.9	-11.9	-11.9	12
4500	20.8	-20	-3.4	4.3	1.1	15	-14.2	-14.1	15.1
4800	21.5	-21.2	-5.3	5.8	0.7	18.1	-17.8	-17.7	18.3
4300	18.8	-18.2	-5.5	6.3	0.4	16.6	-16	-15.9	17.5
3800	16.9	-16.5	-6	6.5	0.3	15.1	-14.6	-14.5	15.2
Max Error	0.1%	0.1%	1.7%	1.7%	0.1%	0.1%	0.1%	0.1%	0.1%

Table 17 shows the following:

- Low muzzle velocity = 280 m/s, Azimuth = 0 deg
- Minimum range is 2000m and maximum range is 4800m with 500m interval.

MPMM section shows the following:

- The maximum error in tail and head wind appear in the maximum range (4800m), however, it is around 1%.

PMM section shows the following:

- The maximum error in tail and headwind appear in the upper trajectories and reach up to 1.7%.

4.3.2.2 Arterially (105 mm) :

Initial condition and more details shown in Appendix 2.

Table 18: Table F(II) for 105 mm artillery

5-DOF									
Range	VELOCITY (±1M/S)		WIND (±1 KNOT)			TEMPERATURE (±1%)		DENSITY (±1%)	
	Inc	Dec	Tail	head	cross	Inc	Dec	Inc	Dec
5000	15.3	-18.4	-27.2	20.1	9.6	12.9	-17.3	-17.1	13
6000	17	-9.8	-28.6	17.1	-1.2	2.8	-9.1	-8.9	3
7000	21.6	-21.8	-30.7	45	-9.9	20.5	-21.3	-21.1	20.7
8000	21.2	-4.5	-51.7	58.3	11.2	23.6	-6.6	-6.4	23.9
9000	16.9	-14.9	-61.5	68.6	5	21.6	-20.5	-20.2	21.9
10000	32.7	-6.5	-76.3	80.2	-3.5	40.6	-35.7	-35.4	41
11000	33.7	-16	-91.8	118	13.4	43.5	-26.2	-25.8	44
11000	21.5	-9	-110	106	-1.6	46	-58.6	-58.1	46.5
10000	15.1	-19.7	-91	108.8	5.6	57.3	-36.9	-36.5	57.8
MPMM									
5000	11.8	-12	-3.1	2.9	-0.1	14.2	-14.5	-14.4	14.3
6000	14.9	-14.7	-1.5	7.3	2.9	18.8	-18.7	-18.6	18.9
7000	16	-13.7	-1.8	9.6	3.8	21.7	-19.4	-19.3	21.8
8000	17.8	-12	-6.7	7.3	0.2	25.7	-20.1	-19.9	25.8
9000	17.4	-12.6	-8.3	13.1	4.7	32.9	-23.4	-23.2	33.1
10000	19.3	-15.5	-7.6	11.5	1.8	33.1	-29.5	-29.3	33.4
11000	18.3	-16.5	-14.1	11.8	0.7	40.1	-38.5	-38.2	40.4
11000	19	-17.2	-12.7	14.4	0.1	46	-44.5	-44.2	46.3
10000	17.8	-16.6	-14.9	16.1	0.7	42	-41	-40.7	42.3
Max Error	0.2%	0.1%	0.9%	1.0%	0.2%	0.3%	0.2%	0.2%	0.3%
PMM									
5000	12.8	-11	-2.2	3.9	0.9	15.2	-13.5	-13.5	15.3
6000	15.1	-11.7	-1.3	4.7	3.1	19	-15.7	-15.6	19.1
7000	15.2	-14.5	-5.4	6	0.3	20.8	-20.3	-20.2	20.9
8000	15.2	-14.6	-6.7	7.3	0.3	25.7	-22.7	-22.5	25.8
9000	16.7	-15.7	-9	10	1.7	29.8	-26.5	-26.3	30
10000	16.8	-15.8	-10.2	11.2	-0.6	32.9	-32.1	-31.9	33.1
11000	18.5	-16.3	-11.8	14.1	1.1	40.3	-38.3	-38.1	40.6
11000	20	-16.6	-12.7	16	1.7	47.1	-43.9	-43.6	47.4
10000	16.8	-15.9	-13.9	14.9	0.5	42.2	-40.3	-40	42.5
Max Error	0.2%	0.1%	0.9%	0.9%	0.2%	0.3%	0.2%	0.2%	0.3%

Table 18 shows the following:

- Low muzzle velocity = 497 m/s, Azimuth = 0 deg
- Minimum range is 5000m and maximum range is 11000m with 1000m interval.

MPMM section shows:

- Only head wind reached a max error of 1% at maximum range, when 5-DOF results compared to MPMM results.

PMM section shows:

- No Significant Error are noticed when 5-DOF results compared to PMM results.

4.3.2.3 Arterially (155 mm) :

Initial condition and details shown in Appendix 2.

Table 19: Table F(II) for 155 mm artillery

5-DOF									
Range (m)	VELOCITY (±1M/S)		WIND (±1 KNOT)			TEMPERATURE (±1%)		DENSITY (±1%)	
	Inc	Dec	Tail	head	cross	Inc	Dec	Inc	Dec
12000	4.5	0.7	-82.4	87.4	-3.8	56.2	-51.4	-50.7	56.9
14000	34.9	-9.3	-112.3	106.1	-8.7	65.7	-71.3	-70.4	66.5
16000	3	-46.7	-138.6	122.1	-13.3	71.3	-115.2	-114.3	72.2
18000	18.1	-8	-149.9	153.7	-2.2	92.3	-112.8	-111.7	93.3
20000	16.3	-16.9	-177.8	196.1	-8.8	127.2	-128.1	-126.9	128.3
22000	28.1	-13.3	-188.4	214.1	-2.8	144.3	-129.7	-128.5	145.5
24000	54	0.3	-194.9	240.2	13.6	144.8	-125.9	-124.6	146.1
24000	35.3	-14.9	-209.4	257.1	1.7	197	-175.7	-173.9	198.9
22000	34.7	-57.6	-204.7	185.7	-33.9	171.4	-192.6	-187.2	154.9
MPMM									
12000	23	-11.2	3.6	8.2	5.9	65.9	-47.9	-47.4	66.4
14000	22	-18.8	0.5	2.8	4.6	78.6	-75.4	-74.9	79.1
16000	20.7	-20.5	-3.7	3.9	-2.7	88.1	-88	-87.3	88.7
18000	20.5	-21.3	-7.2	6.4	2.1	103.1	-104	-103.3	103.8
20000	26. 4	-21	-5.1	10.4	2.5	118.1	-112.8	-112.1	118.9
22000	29.9	-22.1	-10.2	17.9	1.6	133.7	-126.1	-125.3	134.6
24000	29.7	-26.3	-19.1	19.1	1.3	144.9	-146.2	-145.2	145.9
24000	37.8	-33.4	-19.8	26.3	2.2	184.3	-177	-175.6	185.7
22000	40.3	-31	-21	30.2	2	183.3	-174	-172.5	184.8
Max Error	0.2%	0.2%	0.9%	1.0%	0.2%	0.1%	0.2%	0.2%	0.1%
PMM									
12000	24.9	-9.4	5.5	10.1	7.8	64.6	-49.4	-48.9	65
14000	21.7	-13.1	0.2	8.5	4.3	78.4	-72.8	-72.2	78.9
16000	19.4	-19	-5	5.4	1.5	89.6	-89.3	-88.7	90.3
18000	24.7	-19.6	-5.5	10.7	1.3	104.9	-102.3	-101.6	105.6
20000	22.9	-22	-10.9	11.8	-0.7	117	-116.2	-115.4	117.8
22000	27.5	-24.3	-12.8	15.9	1.5	131.2	-128.2	-127.4	132.1
24000	30.4	-26.8	-18.9	20.8	1	146.2	-144.3	-143.3	147.2
24000	37.5	-31.7	-21.5	26.1	2.3	180.3	-175	-173.6	182.9
22000	39.5	-33.3	-20.8	27.9	3.5	183.3	-176.8	-174.4	184.8
Max Error	0.2%	0.2%	0.8%	1.0%	0.2%	0.1%	0.2%	0.2%	0.1%

Table 19 shows the following:

- High muzzle velocity = 900 m/s, Azimuth = 0 deg
- Minimum range is 12000m and maximum range is 24000m with 2000m interval.

MPMM section shows:

- Only head wind reached a max error of 1% at maximum range, when 5-DOF results compared to MPMM results.

PMM section shows:

- Only head wind reached a max error of 1% at maximum range, when 5-DOF results compared to PMM results.

4.4 *Conclusion:*

This chapter shows all the data generated by the three models for the three projectiles to establish firing table. In term of accuracy Table 20 shows accuracy acceptance of PMM and MPMM compared to 5-DOF results.

Table 20: Models accuracy

Projectile	Table	PMM	MPMM
81 mm spin-stabilized	F (I)	Accepted	Accepted
	F (II)	Not accepted	Accepted
105 mm spin-stabilized	F (I)	Not accepted	Accepted
	F (II)	Accepted	Accepted
155 mm spin-stabilized	F (I)	Not accepted	Not accepted
	F (II)	Accepted	Accepted

On the other hand, Table 21 gives a range of the time consumed of each model.

Table 21: Simulation time comparison

Time-consuming	**PMM**	**MPMM**	**5-DOF**
	Less than 1 min	Around 3 min	Around 30 min

Therefore, PMM only generate parts of the firing tables and it cannot deliver a complete one, while the MPMM generate a full firing table except with the high velocity spin-stabilised projectiles. Lastly, the 5-DOF is delivering acceptable results at all times, but with significant simulation time consumption.

Chapter 5 Summary and Conclusions

The study objectives stated in Chapter 1 can be summarised as :

To study the accuracy of existing models;

To evaluate these models with respect to the simulation time;

To determine the best model for any projectile to generate the firing table; and

To identify the limitations of each model.

These objectives have been achieved. PMM can only generate parts of the firing tables, while the MPMM can generate a full firing table except with less accuracy for the high velocity spin-stabilised projectiles. Lastly, the 5-DOF is delivering acceptable results at all times, but with significant simulation time consumption.

On the other hand, The conclusion regarding the advantages and disadvantages of each model, with comments on the best solution for various scenarios are:

5.1 *PMM:*

Advantages: It is a very simple and fast simulation model with good level of acceptance, especially for fin-stabilised projectiles. Requiring minimum data to present a projectile trajectory.

Disadvantages: poor result with spin-stabilised projectiles.

Use: Powerful in Rapid Trajectory Prediction or fire control correction.

5.2 *MPMM:*

Advantages: It is a fast simulation model (5-6 trajectory/min) with high level of acceptance, can work with both the fin-stabilised and the spin-stabilised projectiles.

Disadvantages: build up error with spin-stabilised projectiles at high elevation (higher than 65^{o}).

Use: The best to generate firing table or simple trajectories analysis.

5.3 *5-DOF:*

Advantages: It is a very accurate simulation model with best level of acceptance.

Disadvantages: complicated and slow simulation model.

Usage: the most accurate for trajectories analysis.

5.4 *Future work:*

For future work in this field the study can be expanded into such areas as:

Find the optimum way to integrate the three models in one frame work by using Fuzzy logic or other methods.

Study Appling trajectory models in mobile devices.

Optimising the integration method used for trajectory simulation.

Optimising the method of iteration in the dynamic time step that work with five degree of freedom model.

Bibliography

Baranovski, L. (2013a) ‘Feasibility Analysis of the Modified Point Mass Trajectory Model for the Need of Ground Artillery Fire Control Systems’, *Journal of Theoretical and Applied Mechanics*, 51(3), pp. 511–522.

Baranowski, L. (2013b) ‘Effect of the mathematical model and integration step on the accuracy of the results of computation of artillery projectile flight parameters’, *Bulletin of the polish academy of sciences technical sciences.*, 61(2), pp. 475-484

Barbosa, L. (2005) ‘A Critical Evaluation of Three Models of External Ballistics’, in The Brazilian Congress of Mechanical Engineering.

Chudinov, P. S., Eltyshev, V. A. and Barykin, Y. A. (2013) ‘Simple analytical description of projectile motion in a medium with quadratic drag force’, Latin-American Journal of Physics Education, 7(3), pp. 345–349.

Chusilp, P., Charubhun, W. and Ridluan, A. (2011) ‘Developing Firing Table Software for Artillery Projectiles using Iterative Search and 6-DOF Trajectory Model’, in *The Second TSME International Conference on Mechanical Engineering*.

Dickinson, E. R. (1967) ‘The production of firing tables for cannon artillery’, *U.S. Army materiel command ballistic research laboratories*.

Dykes, J. *et al.* (2011) ‘Periodic projectile linear theory for aerodynamically asymmetric projectiles’, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 228(11), pp. 2094–2107.

Fann, C. M. (2006) ‘Development of an artillery accuracy model’. Nanyang Technological University.

Field Manual 6-40 (FM 6-40). (1999). ‘Field artillery manual cannon gunnery’. Washington, D.C., USA: Headquarters Department of the Army.

Hainz, L. C. and Costello, M. (2005) 'Modified Projectile Linear Theory for Rapid Trajectory Prediction', *Journal of Guidance, Control, and Dynamics*, 28(5), pp. 1006–1014.

Khalil, M., Rui, X., Zha, Q., Yu, H., & Hendy, H. (2013). Projectile Impact Point Prediction Based on Self-Propelled Artillery Dynamics and Doppler Radar Measurements.

Lieske, R. F. and DANBERG, J. E. (1992) 'Modified point mass trajectory simulation for Base-Burn projectiles'.

McCoy, R. (2012) Modern Exterior Ballistics : The Launch and Flight Dynamics of Symmetric Projectiles.

Nangsue, P. (2010) 'Generation of Artillery Firing Tables for The L119 Howitzer with Base-Bleed Projectiles', *Journal of the Faculty Senate*, 8, pp. 127–135.

Oregon, R. S. (2011) A Cots approach to tactical fire support using smartphone. University of California.

Pope, R. L. (1985) 'Breakdown of the modified point mass model for high elevation trajectories', *Defence Research Centre Salisbury*.

Regan, Frank J. (1984) Re-Entry Vehicle Dynamics, AIAA (American Institute of Aeronautics and Astronautics) Education Series.

Roetzel, W., Czarnetzki, W. and Maier, T. (2007) 'Fast calculation of direct fire trajectories taking the earth's rotation into account', in *WIT Transactions on Modelling and Simulation*, pp. 21–30.

Sandvik, A.W. (2018). Numerical Solutions of Classical Equations of Motion.

Skande, M. (2014) Numerical Solution to a Nonlinear Numerical Solution to a Nonlinear External Ballistics Model for a Direct Fire Control System. KTH Industrial Engineering and Management.

STANAG (Standardization Agreement by NATO). 1995. Procedures to determine the degree of ballistic performance similarity of NATO indirect fire ammunition and the applicable corrections to aiming data. (STANAG 4106).

STANAG (Standardization Agreement by NATO). 2007. Adoption of a standard cannon artillery firing table format. (STANAG 4119).

STANAG (Standardization Agreement by NATO). 2008. The NATO error budget model. (STANAG 4635).

STANAG (Standardization Agreement by NATO). 2009. The modified point mass and five degrees of of freedom trajectory models. (STANAG 4355).

APPENDIX 1: Models data requirements

List of physical data requirements (STANAG 4355, 2009).

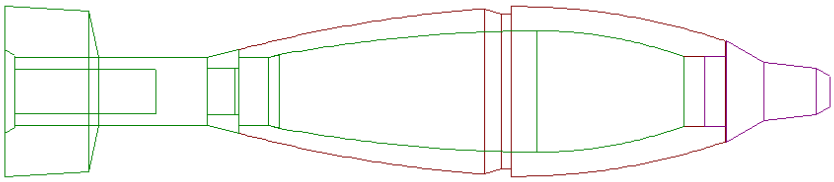
Physical Data Requirements	Five degree of freedom	Modified point mass model	Point mass model
Initial speed of projectile with respect to ground (muzzle velocity)	Yes	Yes	Yes
Twist of rifling at muzzle	Yes	Yes	
Initial mass of fuzzed projectile	Yes	Yes	Yes
Reference mass of fuzzed projectile	Yes	Yes	Yes
Reference diameter	Yes	Yes	Yes
Initial axial moment of inertia	Yes	Yes	
Initial transverse moment of inertia	Yes	Yes	
Axial spin rate of rocket at apogee	Yes		
Axial moment of inertia at apogee	Yes		
Transverse moment of inertia at apogee	Yes		
Specific impulse of rocket fuel Mass flow rate of the motor fuel	Yes		
Aerodynamic Coefficients as function of Mach Number	Yes	Yes	Yes
Fin cant angle	Yes		
Distance from body center of mass to motor nozzle exit	Yes		
Distance from body center of mass to motor nozzle throat	Yes		
Radius of motors mass flow exit	Yes		

APPENDIX 2: Projectiles description

1. Introduction:

   The PRODAS simulation program is supplied with example of various real projectiles. Three of these projectiles were selected for case studies and evaluation of the trajectory models:

   1.1. Fin-stabilised (81mm)



Projectile Data

Projectile Diameter	80.950 mm
Weight	4.340 kg
Axial Inertia	0.00336 kg-m^2
Trans Inertia	0.02467 kg-m^2
CG (from nose)	164.66 mm

Initial Conditions

Muzzle Velocity	279.7 m/sec

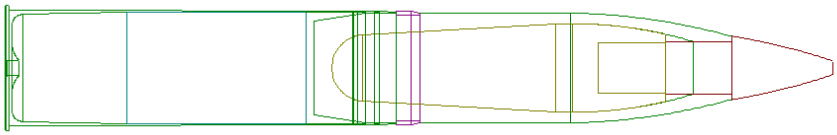
Met Data at the Gun Altitude of 0. m

Temperature	15.0 C	Pressure	1013.2 millibars
Range Wind	0.0 m/sec	Cross Wind	0.0 m/sec

Basic Aerodynamic Coefficients

MACH	CD0	CDA2	CLA	CLA3	CMA	CMA3	CMQ	CL	CLP
0.010	0.157	3.125	1.75	-1.3	-1.8	-2.5	-55.7	0.003	-0.25
0.500	0.178	3.125	1.75	-1.3	-1.8	-2.5	-56	0.003	-0.25
0.600	0.168	3.125	1.75	-1.4	-1.8	-2.5	-56	0.003	-0.25
0.800	0.173	3.125	1.75	-1.7	-2.1	-3	-57	0.003	-0.25
0.880	0.244	4.25	1.8	-2	-2.2	-3	-57.5	0.003	-0.25
0.900	0.264	5	2	-2.3	-2.4	-3.5	-58	0.003	-0.25
0.950	0.507	5	2	-2.6	-2.4	-3.5	-58	0.003	-0.25
1.000	0.535	5.25	2.05	-2.8	-2.5	-3.8	-58.5	0.003	-0.25
1.050	0.621	5.75	2.1	-2.8	-2.6	-4	-59	0.003	-0.25
1.100	0.663	6.5	2.15	-2.8	-2.7	-4.2	-59.5	0.003	-0.25
1.200	0.642	6.75	2.15	-2.8	-2.6	-4.4	-60	0.003	-0.25
1.350	0.6	6.5	2.2	-2.8	-2.5	-4.5	-60.5	0.003	-0.25
1.500	0.567	6	2.25	-2.8	-2.4	-4.5	-61	0.003	-0.25

1.2. Spin-stabilised low muzzle velocity (105mm)



Projectile Data

Projectile Diameter 104.850 mm

Weight 14.970 kg

Axial Inertia 0.0233 kg-m^2

Trans Inertia 0.226 kg-m^2

CG (from nose) 311.40 mm

Initial Conditions

Muzzle Velocity	497.6 m/sec	Initial Spin	1638.9 rad/sec

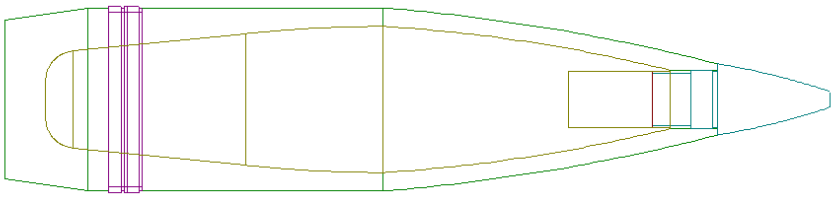
Met Data at the Gun Altitude of 0. m

Temperature	15.0 C	Pressure	1013.2 millibars
Range Wind	0.0 m/sec	Cross Wind	0.0 m/sec

Basic Aerodynamic Coefficients

MACH	CD0	CDA2	CDB	CLA	CMA	CMQ	CL	CLP	CYPA	CMPA
0.010	0.130	3.900	0.000	1.450	3.450	-7.300	0.000	-0.028	-0.800	-1.000
0.400	0.130	3.900	0.000	1.450	3.549	-7.300	0.000	-0.028	-0.800	-1.000
0.600	0.130	3.900	0.000	1.450	3.600	-7.300	0.000	-0.028	-0.800	-1.000
0.700	0.120	3.900	0.000	1.450	3.700	-7.300	0.000	-0.028	-0.800	-0.910
0.750	0.117	3.900	0.000	1.500	3.750	-7.300	0.000	-0.028	-0.800	-0.860
0.800	0.115	3.900	0.000	1.530	3.850	-7.300	0.000	-0.028	-0.800	-0.810
0.850	0.115	4.000	0.000	1.560	3.970	-8.000	0.000	-0.028	-0.840	-0.760
0.875	0.118	4.100	0.000	1.590	4.100	-8.400	0.000	-0.028	-0.870	-0.480
0.900	0.125	4.200	0.000	1.630	4.300	-8.700	0.000	-0.028	-0.890	-0.240
0.925	0.145	4.400	0.000	1.650	4.600	-9.100	0.000	-0.028	-1.010	0.140
0.950	0.185	4.600	0.000	1.650	4.700	-9.500	0.000	-0.027	-1.130	0.290
0.975	0.260	4.800	0.000	1.650	4.400	-10.200	0.000	-0.027	-1.080	0.330
1.000	0.337	5.000	0.000	1.700	4.100	-12.000	0.000	-0.027	-1.030	0.350
1.025	0.385	5.200	0.000	1.800	3.980	-13.000	0.000	-0.026	-0.990	0.440
1.050	0.415	5.300	0.000	1.900	3.930	-14.000	0.000	-0.026	-0.940	0.440
1.100	0.415	5.400	0.000	2.000	3.850	-15.000	0.000	-0.026	-0.890	0.440
1.200	0.405	5.400	0.000	2.150	3.780	-17.000	0.000	-0.025	-0.800	0.450
1.350	0.390	5.300	0.000	2.400	3.670	-17.000	0.000	-0.025	-0.800	0.450
1.500	0.375	5.200	0.000	2.500	3.600	-17.000	0.000	-0.025	-0.800	0.460
1.750	0.357	5.050	0.000	2.550	3.525	-17.000	0.000	-0.025	-0.800	0.460
2.000	0.340	4.900	0.000	2.600	3.450	-17.000	0.000	-0.025	-0.800	0.460
2.250	0.323	4.750	0.000	2.700	3.325	-17.000	0.000	-0.025	-0.800	0.460
2.500	0.305	4.600	0.000	2.750	3.200	-17.000	0.000	-0.025	-0.800	0.460

1.3. Spin-stabilised high velocity (155mm)



Projectile Data

Projectile Diameter	154.700 mm		
Weight	43.096 kg		
Axial Inertia	0.1425 kg-m^2		
Trans Inertia	1.226 kg-m^2		
CG (from nose)	458.37 mm		

Initial Conditions

Muzzle Velocity	568.0 m/sec	Initial Spin	1151.7 rad/sec

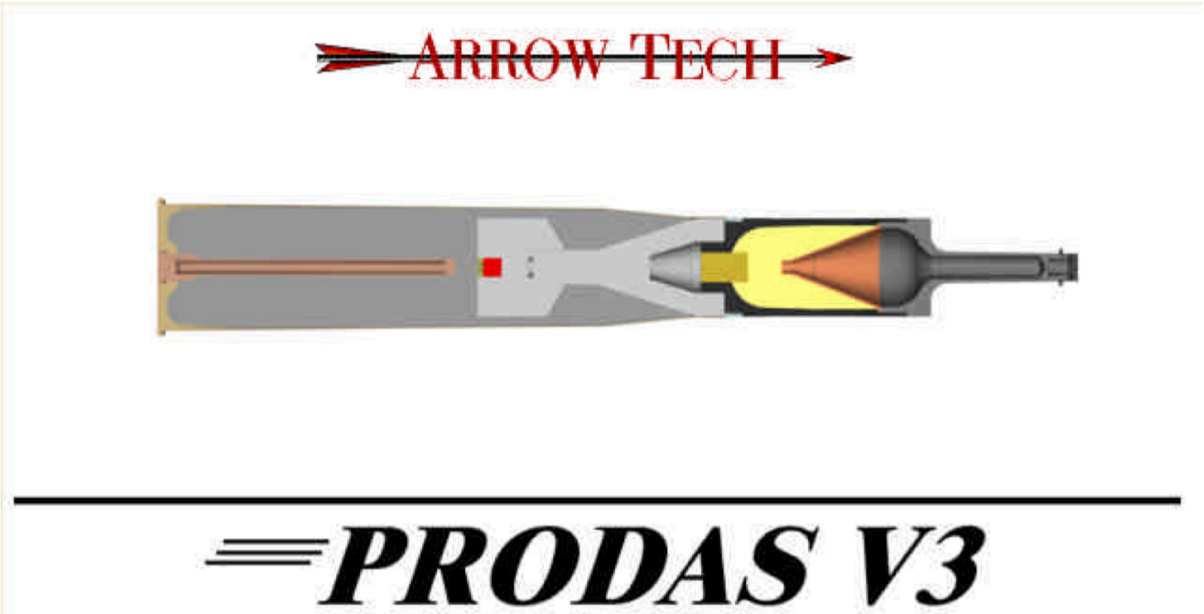
Met Data at the Gun Altitude of 0. m

Temperature	15.0 C	Pressure	1013.2 millibars
Range Wind	0.0 m/sec	Cross Wind	0.0 m/sec

Basic Aerodynamic Coefficients

MACH	CD0	CDA2	CDB	CLA	CMA	CMQ	CL	CLP	CYPA	CMPA
0.010	0.1414	1.78	0	1.889	3.342	-8.6	0.000	-0.02768	-0.71	-1.16
0.400	0.1416	1.78	0	1.89	3.339	-8.7	0.000	-0.02794	-0.71	-1.16
0.600	0.1428	1.79	0	1.901	3.369	-8.5	0.000	-0.02782	-0.71	-1.16
0.700	0.1436	1.96	0	1.909	3.391	-8.4	0.000	-0.02769	-0.72	-1.26
0.750	0.1449	2.07	0	1.916	3.43	-8.5	0.000	-0.0275	-0.72	-1.28
0.800	0.1485	2.18	0	1.933	3.497	-8.4	0.000	-0.02747	-0.74	-1.23
0.850	0.1496	2.36	0	1.941	3.6	-8.4	0.000	-0.02751	-0.76	-1
0.875	0.1596	2.48	0	1.977	3.673	-8.8	0.000	-0.02735	-0.77	-0.84
0.900	0.1754	2.57	0	2.033	3.741	-9.3	0.000	-0.02706	-0.79	-0.8
0.925	0.195	2.74	0	2.112	3.819	-10	0.000	-0.02652	-0.89	-0.71
0.950	0.2288	2.9	0	2.194	3.826	-10.8	0.000	-0.02618	-1.01	-0.55
0.975	0.2733	3.08	0	2.269	3.756	-11.1	0.000	-0.0262	-0.9	-0.32
1.000	0.3237	3.27	0	2.328	3.622	-11.4	0.000	-0.02638	-0.83	-0.16
1.025	0.3573	3.46	0	2.385	3.523	-11.3	0.000	-0.02596	-0.8	-0.01
1.050	0.3795	3.69	0	2.425	3.457	-11.3	0.000	-0.02566	-0.76	0.05
1.100	0.3835	4.19	0	2.445	3.416	-11.6	0.000	-0.0253	-0.71	0.13
1.200	0.3812	4.69	0	2.476	3.408	-12.6	0.000	-0.02504	-0.65	0.15
1.350	0.3626	4.21	0	2.533	3.377	-12.9	0.000	-0.02448	-0.58	0.23
1.500	0.3462	3.72	0	2.621	3.27	-13	0.000	-0.02362	-0.56	0.31
1.750	0.3215	3.25	0	2.724	3.141	-13	0.000	-0.02274	-0.54	0.31
2.000	0.2972	2.74	0	2.812	3.059	-12.4	0.000	-0.02204	-0.51	0.33
2.250	0.2764	2.45	0	2.866	2.975	-12	0.000	-0.0212	-0.5	0.32

2. PRODAS program



The PRODAS Program was developed to satisfy a need for rapid performance evaluation of ammunition characteristics. Projectile design and performance evaluation, in general, requires detailed analysis and testing involving interior, exterior, and terminal ballistics. The development of an effective design/analysis tool for use by the design engineer in the development and evaluation of projectiles has been a multi-year project which began at General Electric in 1972 and has continued at Arrow Tech Associates, Inc. since 1991. The developed tool is called PRODAS which is an acronym for the Projectile Design/Analysis System. The primary objective of PRODAS development has been to provide an effective analytical tool that allows for rapid and complete design of projectiles and rockets. In general, the system makes use of the display and interaction capabilities of interactive graphics to provide the engineer with a user-friendly working environment. The basic approach has been to develop PRODAS in an open-ended fashion, such that as its capabilities are extended, it will at all times exist as a functional design analysis tool. The projectile modeling phase and interactive graphics medium provides the design capability. The analysis capability is provided by the methodology and techniques contained in the individual analysis segments.

PRODAS has been developed using proven methodologies and techniques such that predicted performance estimates are based in part on prior experimental testing. The approach has been to link these diversified analyses together by means of a common database such that the required results of one analysis feed directly to the subsequent

analysis. For example, the stability analysis results in the estimation of the aerodynamic force and moment coefficients. These will be passed directly to the trajectory analysis for input towards evaluating the motion patterns that may result during the actual firings. The common database provides inherent continuity. Utilizing the interactive graphics medium provides effective presentation of the results for rapid interpretation by the engineer and subsequent iteration with modified input conditions. The database is maintained such that, as experimental data become available, the analysis may be easily redone using the actual parameters instead of estimated parameters.

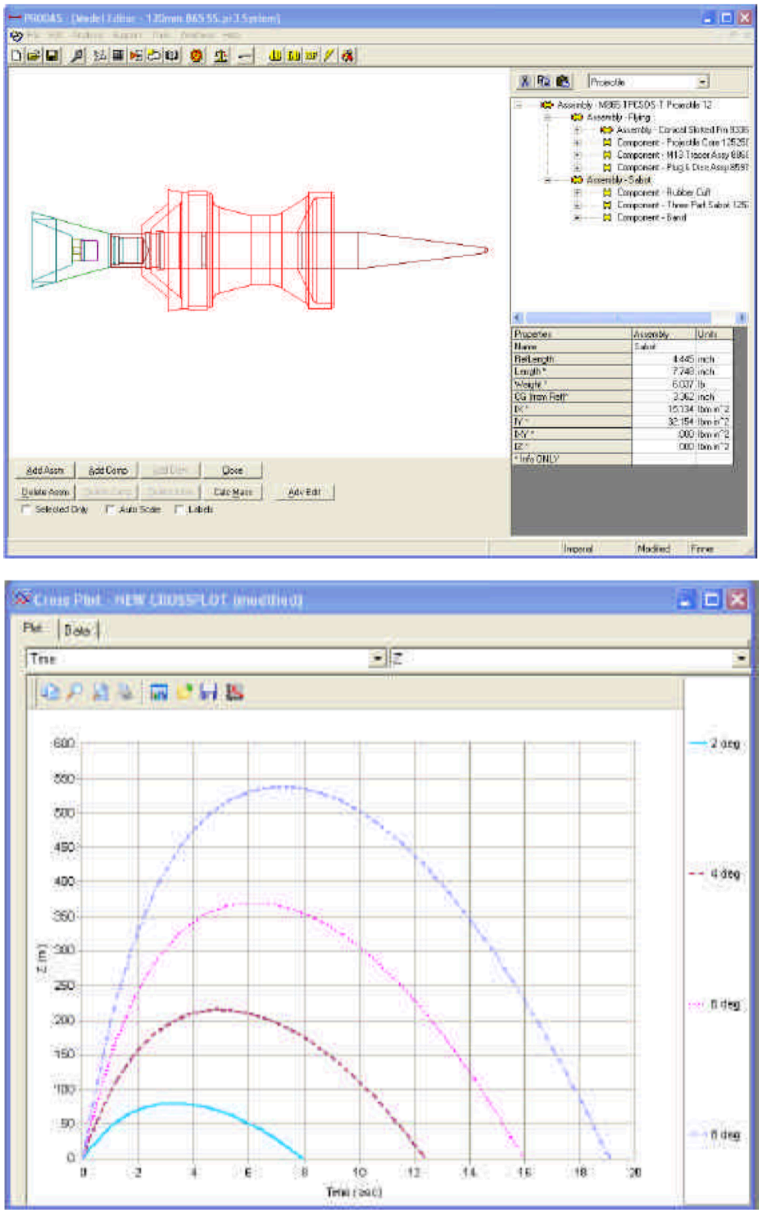


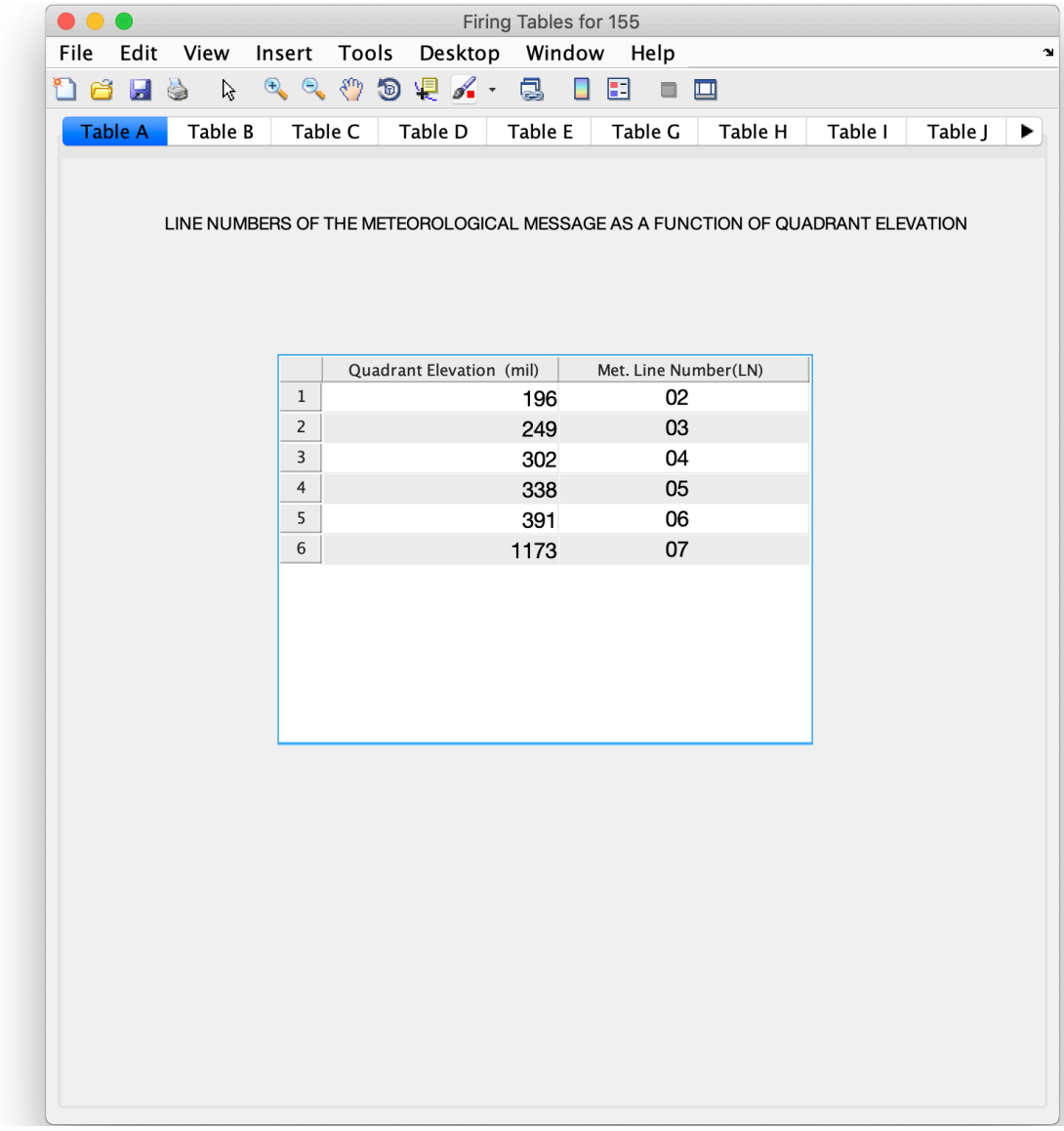
Figure 20: Examples for PRODAS UI

APPENDIX 3: Fringe table Examples

In this appendix some tables from firing table of 155 mm, However, some of the these tables are not generated by any trajectory model, more details shown in Appendix 4:

Table A:

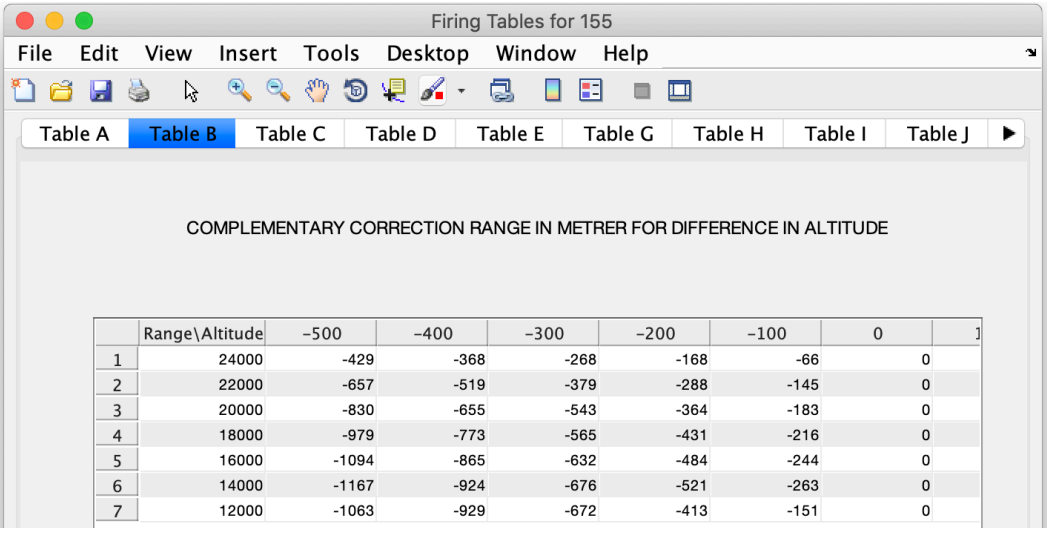
A Met Line Number as a function of Quadrant Elevation



	Quadrant Elevation (mil)	Met. Line Number(LN)
1	196	02
2	249	03
3	302	04
4	338	05
5	391	06
6	1173	07

Table B:

Correction to range (or elevation) for difference in altitude of target and gun

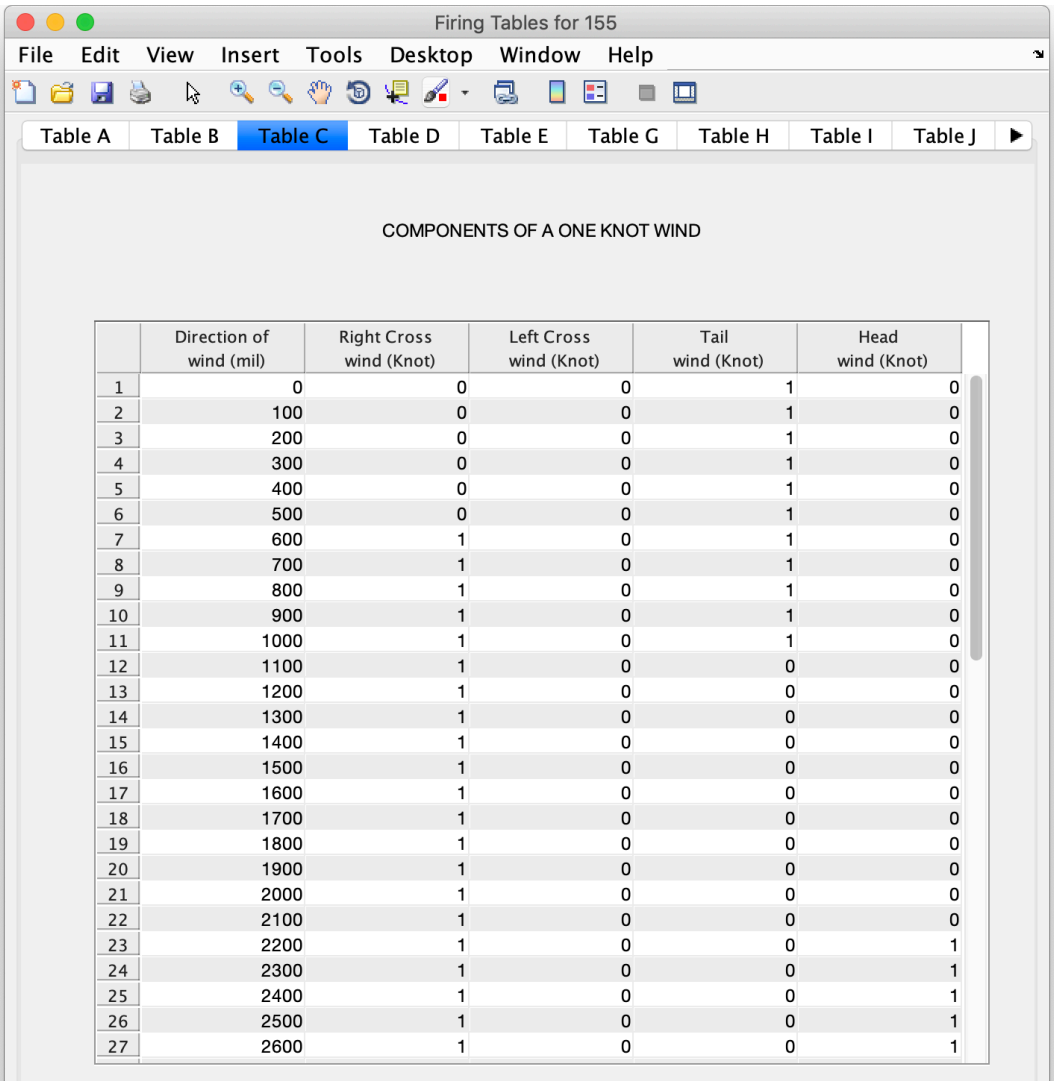


COMPLEMENTARY CORRECTION RANGE IN METRER FOR DIFFERENCE IN ALTITUDE

	Range\Altitude	-500	-400	-300	-200	-100	0
1	24000	-429	-368	-268	-168	-66	0
2	22000	-657	-519	-379	-288	-145	0
3	20000	-830	-655	-543	-364	-183	0
4	18000	-979	-773	-565	-431	-216	0
5	16000	-1094	-865	-632	-484	-244	0
6	14000	-1167	-924	-676	-521	-263	0
7	12000	-1063	-929	-672	-413	-151	0

Table C:

Four direction of wind components as one-knot wind (note 1600 mil = 90 degree).

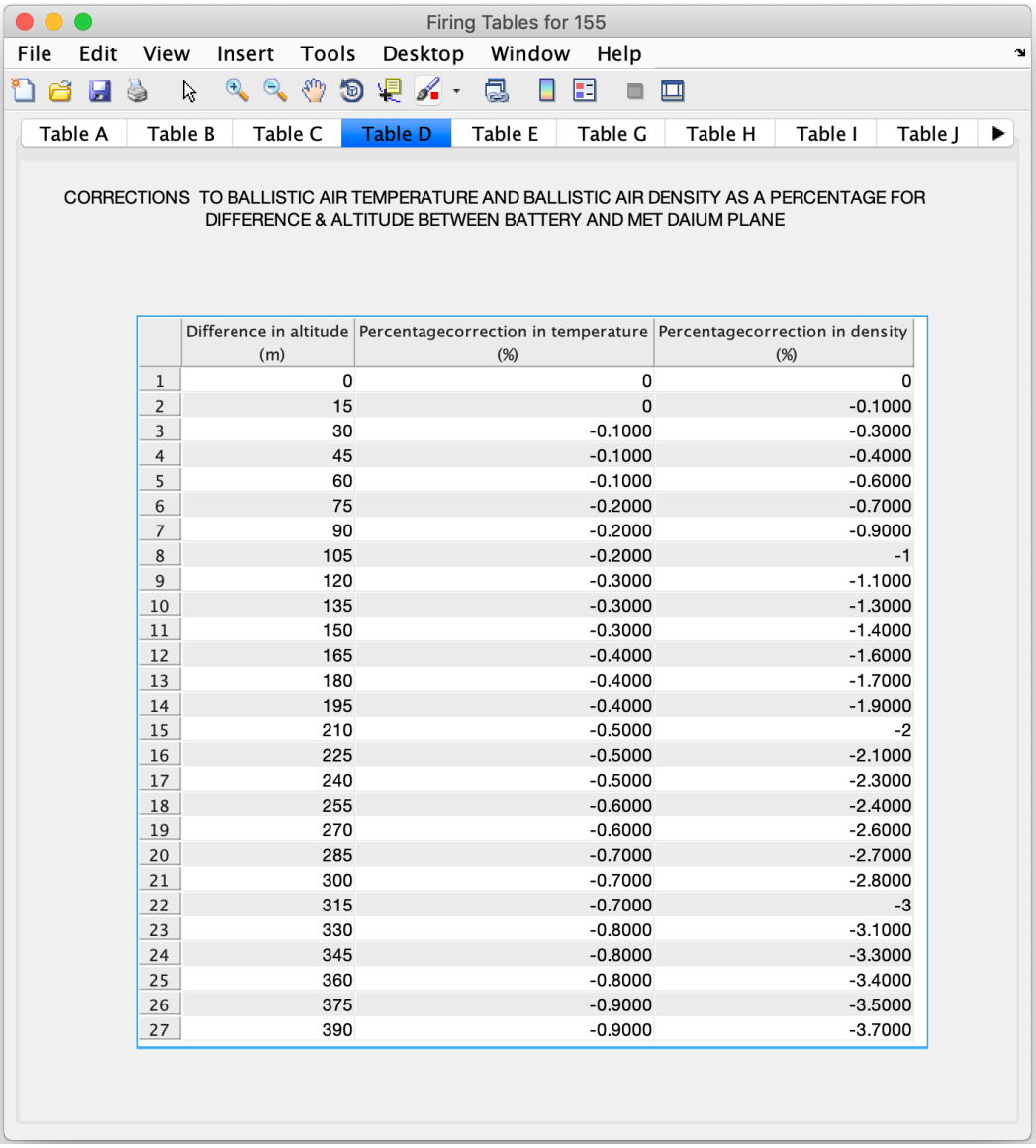


COMPONENTS OF A ONE KNOT WIND

	Direction of wind (mil)	Right Cross wind (Knot)	Left Cross wind (Knot)	Tail wind (Knot)	Head wind (Knot)
1	0	0	0	1	0
2	100	0	0	1	0
3	200	0	0	1	0
4	300	0	0	1	0
5	400	0	0	1	0
6	500	0	0	1	0
7	600	1	0	1	0
8	700	1	0	1	0
9	800	1	0	1	0
10	900	1	0	1	0
11	1000	1	0	1	0
12	1100	1	0	0	0
13	1200	1	0	0	0
14	1300	1	0	0	0
15	1400	1	0	0	0
16	1500	1	0	0	0
17	1600	1	0	0	0
18	1700	1	0	0	0
19	1800	1	0	0	0
20	1900	1	0	0	0
21	2000	1	0	0	0
22	2100	1	0	0	0
23	2200	1	0	0	1
24	2300	1	0	0	1
25	2400	1	0	0	1
26	2500	1	0	0	1
27	2600	1	0	0	1

Table D:

Corrections to ballistic temperature and ballistic density to compensate for the difference in altitude.



CORRECTIONS TO BALLISTIC AIR TEMPERATURE AND BALLISTIC AIR DENSITY AS A PERCENTAGE FOR DIFFERENCE & ALTITUDE BETWEEN BATTERY AND MET DAIUM PLANE

	Difference in altitude (m)	Percentagecorrection in temperature (%)	Percentagecorrection in density (%)
1	0	0	0
2	15	0	-0.1000
3	30	-0.1000	-0.3000
4	45	-0.1000	-0.4000
5	60	-0.1000	-0.6000
6	75	-0.2000	-0.7000
7	90	-0.2000	-0.9000
8	105	-0.2000	-1
9	120	-0.3000	-1.1000
10	135	-0.3000	-1.3000
11	150	-0.3000	-1.4000
12	165	-0.4000	-1.6000
13	180	-0.4000	-1.7000
14	195	-0.4000	-1.9000
15	210	-0.5000	-2
16	225	-0.5000	-2.1000
17	240	-0.5000	-2.3000
18	255	-0.6000	-2.4000
19	270	-0.6000	-2.6000
20	285	-0.7000	-2.7000
21	300	-0.7000	-2.8000
22	315	-0.7000	-3
23	330	-0.8000	-3.1000
24	345	-0.8000	-3.3000
25	360	-0.8000	-3.4000
26	375	-0.9000	-3.5000
27	390	-0.9000	-3.7000

Table E:

Effects on muzzle velocity due to propellant temperature, which cannot be generated by trajectory model.

Table F:

Basic Data and Corrections for non-standard conditions. As shown in chapter 4.

Table G,H,J:

Required experimental data.

APPENDIX 4: MATLAB code

Inputs

```
clear
input = figure('Position',[600 400 250 250],'Name','Chose models');
txt = uicontrol('Parent',input,...
                'Style','text',...
                'Position',[20 170 210 40],...
                'String','Choose needed tables to generate.');

chA = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[20 130 80 40],...
                'String','A');
chB = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[60 130 80 40],...
                'String','B');
chC = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[100 130 80 40],...
                'String','C');
chD = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[140 130 80 40],...
                'String','D');
chFi = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[180 130 80 40],...
                'String','F(I)');
chFii = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[20 90 80 40],...
                'String','F(II)');
chG = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[60 90 80 40],...
                'String','G');
chH = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[100 90 80 40],...
                'String','H');
chI = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[140 90 80 40],...
                'String','I');
chJ = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[180 90 80 40],...
                'String','J');

uicontrol('Position',[100 20 50 50],'String','Done',...
              'Callback','uiresume(gcbf)');
uiwait(gcf);
chA=get(chA,'value');
chB=get(chB,'value');
chC=get(chC,'value');
chD=get(chD,'value');
chFi=get(chFi,'value');
```

```
chFii=get(chFii,'value');
chG=get(chG,'value');
chH=get(chH,'value');
chI=get(chI,'value');
chJ=get(chJ,'value');
input.Visible='off';
%
input = figure('Position',[600 400 250 250],'Name','Chose models');
txt = uicontrol('Parent',input,...
                'Style','text',...
                'Position',[20 170 210 40],...
                'String','Choose ONE model You want to simulate.');

ch1 = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[10 100 80 40],...
                'String','Vacuum');
ch2 = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[70 100 80 40],...
                'String','PMM');
ch3 = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[120 100 80 40],...
                'String','MPMM');
ch4 = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[180 100 80 40],...
                'String','5-DOF');
ch5 = uicontrol('Parent',input,...
                'Style','checkbox',...
                'Position',[10 130 80 40],...
                'String','Analytical');
uicontrol('Position',[100 20 50 50],'String','Done',...
              'Callback','uiresume(gcbf)');
uiwait(gcf);
ch1=get(ch1,'value');
ch2=get(ch2,'value');
ch3=get(ch3,'value');
ch4=get(ch4,'value');
ch5=get(ch5,'value');
input.Visible='off';
%
in = inputdlg({'Muzzel Velocity','Diameter','Mass'},...
    'Input',[1 30],{'95.6','81','4.34'});

V=str2double(in{1});
dia=str2double(in{2})/1000;
mass=str2double(in{3});
```

Data check

```
[aero,iner,tc]=aero_data((dia*1000));
fi=figure('NumberTitle','off','Name','Coefficiant','Position',[300 200 700
490]);
uicontrol('Position',[20 420 50 50],'String','Done',...
              'Callback','uiresume(gcbf)');
inretia=uitable(fi,'data',num2cell(iner),'Position',[90 410 460 70]);
inretia.ColumnName={ 'Mass' ,  'Ixx'  ,     'Iyy'  ,     'Izz'  ,
'Ixy'   ,'Ixz'  , 'Iyz' , 'Ycg'  , 'Zcg'  , 'Xcg'};
inretia.RowName={};
```

```
inretia.ColumnEditable=true;

twist=uitable(fi,'data',num2cell(tc),'Position',[580 410 100 70]);
twist.ColumnName={'Muzzel Twist'};
twist.RowName={};
twist.ColumnEditable=true;

coeff=uitable(fi,'data',num2cell(aero),'Position',[20 20 660 380]);
coeff.ColumnName={'Mach'  ,  'Cd0'  ,  'Cda2'  , 'Cdb' ,    'CLa'  , 'CLa3'
, 'Cma'  ,  'Cma3' , 'Cmq'  ,    'Cl'  ,    'Clp'  ,  'Cypa'  , 'Cypa3'
,'Cmpa'  , 'Cmpa3' , 'Cn0'  ,  'Bet1'  , 'Cm0'  ,  'Bet2'  , 'Cxd'  ,
'Xc'};
coeff.ColumnWidth={50};
coeff.ColumnEditable=true;
uiwait(gcf);
mnm=(get(coeff,'data'));
[wid,len]=size(aero);
for row=1:wid
    for col=1:len
        aero(row,col)=mnm{row,col};
    end
end
fi.Visible='off';
```

Trajectory generator

```
ele_start=8;
ele_inter=1;
ele_end=66;
if ch1==1
    j=1;
    for elev = ele_start:ele_inter:45

[POS_V{j},i_V(j),T_V{j},TOF_V{j},MH_V_L(j),ang_L(j)]=Vacuum(V,elev,0,0,0,0,
0,0);
    Ran=POS_V{j};
    Tim=TOF_V{j};
    Range_L(j)=Ran(i_V(j),1);
    Drift_L(j)=Ran(i_V(j),3);
    TOF_L(j)=Tim(1);
    j=j+1;
    end
Fit_range_L = fit( rot90(Range_L),(rot90(ele_start:ele_inter:45)),
'linearinterp' );
Fit_drift_L = fit( rot90(Range_L),rot90(Drift_L),   'linearinterp' );
Fit_TOF_L = fit( rot90(Range_L),(rot90(TOF_L)),   'linearinterp' );
Fit_MH_L = fit( rot90(Range_L),(rot90(MH_V_L)),   'linearinterp' );
Fit_ang_L = fit( rot90(Range_L),(rot90(ang_L)),   'linearinterp' );

    j=1;
    for elev = 45:ele_inter:ele_end

[POS_V{j},i_V(j),T_V{j},TOF_V{j},MH_V_U(j),ang_U(j)]=Vacuum(V,elev,0,0,0,0,
0,0);
    Ran=POS_V{j};
    Tim=TOF_V{j};
    Range_U(j)=Ran(i_V(j),1);
    Drift_U(j)=Ran(i_V(j),3);
    TOF_U(j)=Tim(1);
    j=j+1;
    end
```

```
Fit_range_U = fit( rot90(Range_U),(rot90(45:ele_inter:ele_end)),
'linearinterp' );
Fit_drift_U = fit( rot90(Range_U),rot90(Drift_U),   'linearinterp' );
Fit_TOF_U = fit( rot90(Range_U),(rot90(TOF_U)),   'linearinterp' );
Fit_MH_U = fit( rot90(Range_U),(rot90(MH_V_U)),   'linearinterp' );
Fit_ang_U = fit( rot90(Range_U),(rot90(ang_U)),   'linearinterp' );

i=1;
if chA==1
    for rang = round(min(Range_U),-2):1000:round(max(Range_U),-2)
    elev=Fit_range_U(rang);
    j=1;
    for Y = -500:100:500

[POS_V_1{j},i_V_1(j),T_V_1{j},TOF_V_1{j},MH_V_1(j)]=Vacuum(V,elev,Y,0,0,0,0
,0);
    Ran=POS_V_1{j};
    Range_Alt(j)=Ran(i_V_1(j),1);
    j=j+1;
    end
    Range_B(i,:)=round(Range_Alt-Range_Alt(6));
    i=i+1;
    end
end
end

if ch2==1
    j=1;
    for elev = ele_start:ele_inter:45

[POS_V{j},i_V(j),T_V{j},TOF_V{j},ang_L(j),MH_V_L(j)]=PMM(V,elev,dia,mass,ae
ro,0,0,0,0,0,0);
    Ran=POS_V{j};
    Tim=TOF_V{j};
    Range_L(j)=Ran(i_V(j)-1,1);
    Drift_L(j)=Ran(i_V(j)-1,3);
    TOF_L(j)=Tim(1);
    j=j+1;
    end
Fit_range_L = fit( rot90(Range_L),(rot90(ele_start:ele_inter:45)),
'linearinterp' );
Fit_drift_L = fit( rot90(Range_L),rot90(Drift_L),   'linearinterp' );
Fit_TOF_L = fit( rot90(Range_L),(rot90(TOF_L)),   'linearinterp' );
Fit_MH_L = fit( rot90(Range_L),(rot90(MH_V_L)),   'linearinterp' );
Fit_ang_L = fit( rot90(Range_L),(rot90(ang_L)),   'linearinterp' );

    j=1;
    for elev = 45:ele_inter:ele_end

[POS_V{j},i_V(j),T_V{j},TOF_V{j},ang_U(j),MH_V_U(j)]=PMM(V,elev,dia,mass,ae
ro,0,0,0,0,0,0);
    Ran=POS_V{j};
    Tim=TOF_V{j};
    Range_U(j)=Ran(i_V(j)-1,1);
    Drift_U(j)=Ran(i_V(j)-1,3);
    TOF_U(j)=Tim(1);
    j=j+1;
    end
Fit_range_U = fit( rot90(Range_U),(rot90(45:ele_inter:ele_end)),
'linearinterp' );
Fit_drift_U = fit( rot90(Range_U),rot90(Drift_U),   'linearinterp' );
Fit_TOF_U = fit( rot90(Range_U),(rot90(TOF_U)),   'linearinterp' );
```

```
Fit_MH_U = fit( rot90(Range_U),(rot90(MH_V_U)),   'linearinterp' );
Fit_ang_U = fit( rot90(Range_U),(rot90(ang_U)),   'linearinterp' );
if chA==1
i=1;
    for rang = round(min(Range_U),-2):1000:round(max(Range_U),-2)
    elev=Fit_range_U(rang);
    j=1;
        for Y = -500:100:500

[POS_V_1{j},i_V_1(j),T_V_1{j},TOF_V_1{j},ang_L_1(j),MH_V_L_1(j)]=PMM(V,elev
,dia,mass,aero,Y,0,0,0,0,0);
        Ran=POS_V_1{j};
        Range_Alt(j)=Ran(i_V_1(j)-1,1);
        j=j+1;
        end
    Range_B(i,:)=round(Range_Alt-Range_Alt(6));
    i=i+1;
    end
end

end

if ch3==1
    j=1;
    for elev = ele_start:ele_inter:45

[POS_V{j},i_V(j),~,T_V{j},TOF_V{j},~,ang_L(j),MH_V_L(j)]=MPMM(V,elev,dia,ma
ss,aero,iner,tc,0,0,0,0,0,0);
    Ran=POS_V{j};
    Tim=TOF_V{j};
    Range_L(j)=Ran(i_V(j)-1,1);
    Drift_L(j)=Ran(i_V(j)-1,3);
    TOF_L(j)=Tim(1);
    j=j+1;
    end
Fit_range_L = fit( rot90(Range_L),(rot90(ele_start:ele_inter:45)),
'linearinterp' );
Fit_drift_L = fit( rot90(Range_L),rot90(Drift_L),   'linearinterp' );
Fit_TOF_L = fit( rot90(Range_L),(rot90(TOF_L)),   'linearinterp' );
Fit_MH_L = fit( rot90(Range_L),(rot90(MH_V_L)),   'linearinterp' );
Fit_ang_L = fit( rot90(Range_L),(rot90(ang_L)),   'linearinterp' );

    j=1;
    for elev = 45:ele_inter:ele_end

[POS_V{j},i_V(j),~,T_V{j},TOF_V{j},~,ang_U(j),MH_V_U(j)]=MPMM(V,elev,dia,ma
ss,aero,iner,tc,0,0,0,0,0,0);
    Ran=POS_V{j};
    Tim=TOF_V{j};
    Range_U(j)=Ran(i_V(j)-1,1);
    Drift_U(j)=Ran(i_V(j)-1,3);
    TOF_U(j)=Tim(1);
    j=j+1;
    end
Fit_range_U = fit( rot90(Range_U),(rot90(45:ele_inter:ele_end)),
'linearinterp' );
Fit_drift_U = fit( rot90(Range_U),rot90(Drift_U),   'linearinterp' );
Fit_TOF_U = fit( rot90(Range_U),(rot90(TOF_U)),   'linearinterp' );
Fit_MH_U = fit( rot90(Range_U),(rot90(MH_V_U)),   'linearinterp' );
Fit_ang_U = fit( rot90(Range_U),(rot90(ang_U)),   'linearinterp' );
if chB==1
i=1;
```

```
    for rang = round(min(Range_U),-2):1000:round(max(Range_U),-2)
    elev=Fit_range_U(rang);
    j=1;
        for Y = -500:100:500

[POS_V_1{j},i_V_1(j),~,T_V_1{j},TOF_V_1{j},~,ang_U_1(j),MH_V_U_1(j)]=MPMM(V
,elev,dia,mass,aero,iner,tc,Y,0,0,0,0,0);
        Ran=POS_V_1{j};
        Range_Alt(j)=Ran(i_V_1(j)-1,1);
        j=j+1;
        end
    Range_B(i,:)=round(Range_Alt-Range_Alt(6));
    i=i+1;
    end
end

end

if ch4==1
    j=1;
    for elev = ele_start:ele_inter:45

[POS_V{j},i_V(j),~,T_V{j},TOF_V{j},~,ang_L(j),MH_V_L(j)]=DOF5(V,elev,dia,ma
ss,aero,iner,tc,0,0,0,0,0,0);
    Ran=POS_V{j};
    Tim=TOF_V{j};
    Range_L(j)=Ran(i_V(j)-1,1);
    Drift_L(j)=Ran(i_V(j)-1,3);
    TOF_L(j)=Tim(1);
    j=j+1;
    end
Fit_range_L = fit( rot90(Range_L),(rot90(ele_start:ele_inter:45)),
'linearinterp' );
Fit_drift_L = fit( rot90(Range_L),rot90(Drift_L),   'linearinterp' );
Fit_TOF_L = fit( rot90(Range_L),(rot90(TOF_L)),   'linearinterp' );
Fit_MH_L = fit( rot90(Range_L),(rot90(MH_V_L)),   'linearinterp' );
Fit_ang_L = fit( rot90(Range_L),(rot90(ang_L)),   'linearinterp' );

    j=1;
    for elev = 45:ele_inter:ele_end

[POS_V{j},i_V(j),~,T_V{j},TOF_V{j},~,ang_U(j),MH_V_U(j)]=DOF5(V,elev,dia,ma
ss,aero,iner,tc,0,0,0,0,0,0);
    Ran=POS_V{j};
    Tim=TOF_V{j};
    Range_U(j)=Ran(i_V(j)-1,1);
    Drift_U(j)=Ran(i_V(j)-1,3);
    TOF_U(j)=Tim(1);
    j=j+1;
    end
Fit_range_U = fit( rot90(Range_U),(rot90(45:ele_inter:ele_end)),
'linearinterp' );
Fit_drift_U = fit( rot90(Range_U),rot90(Drift_U),   'linearinterp' );
Fit_TOF_U = fit( rot90(Range_U),(rot90(TOF_U)),   'linearinterp' );
Fit_MH_U = fit( rot90(Range_U),(rot90(MH_V_U)),   'linearinterp' );
Fit_ang_U = fit( rot90(Range_U),(rot90(ang_U)),   'linearinterp' );

end
```

Tables generator

```
d=figure('Name',['Firing Tables for ' num2str(dia*1000) 'mm projectile'
],'NumberTitle','off'...
    ,'Position',[500 500 700 700]);
m=uitabgroup(d);

in = inputdlg({'Range interval'},...
    'Input',[1 30],{'500'});

    Print_start=round(min(Range_U),-3);
    Print_Int=str2double(in{1});
    Print_max=round(max(Range_L),-3);

%%%
if chFi == 1
   j=1;
for rang=round(Print_start):Print_Int:round(max(Range_L),-2)
 elev_deg=Fit_range_L(rang);
elev_mil=Fit_range_L(rang)*800/45;
TOF=Fit_TOF_L(rang);
Max_H=Fit_MH_L(rang);
    if Max_H <= 200
        Met_line_=00;
    elseif  Max_H <= 500
        Met_line_=01;
    elseif  Max_H <= 1000
        Met_line_=02;
    elseif  Max_H <= 1500
        Met_line_=3;
    elseif  Max_H <= 2000
        Met_line_=04;
    elseif  Max_H <= 2500
        Met_line_=05;
    elseif  Max_H <= 3000
        Met_line_=06;
    else
        Met_line_=07;
    end
ang_Im=Fit_ang_L(rang);
drift=Fit_drift_L(rang);
 Table_Fi(j,:)=([rang,elev_deg,elev_mil,TOF,Met_line_,Max_H,ang_Im,drift]);
 j=j+1;
end
for rang=round((Print_max)):-Print_Int:round(min(Range_U),-2)
 elev_deg=Fit_range_U(rang);
 elev_mil=Fit_range_U(rang)*800/45;
 TOF=Fit_TOF_U(rang);
 Max_H=Fit_MH_U(rang);
 if Max_H <= 200
        Met_line_=00;
    elseif  Max_H <= 500
        Met_line_=01;
    elseif  Max_H <= 1000
        Met_line_=02;
    elseif  Max_H <= 1500
        Met_line_=3;
    elseif  Max_H <= 2000
        Met_line_=04;
    elseif  Max_H <= 2500
        Met_line_=05;
```

```
    elseif  Max_H <= 3000
        Met_line_=06;
    else
        Met_line_=07;
 end
    ang_Im=Fit_ang_U(rang);
drift=Fit_drift_U(rang);

 Table_Fi(j,:)=([rang,elev_deg,elev_mil,TOF,Met_line_,Max_H,ang_Im,drift]);
 j=j+1;
end

 tab6 = uitab(m,'Title','Table F(I)');
  S=uitable(tab6,'data',Table_Fi,'Position',[50 50 600 500]...
      ,'ColumnWidth',{'auto' 'auto' 'auto' 50 70 65 'auto' 'auto'});
  S.ColumnName={'Range','ELEV (deg)','ELEV (mil)','TOF (s)','MET
Line','APEX','Impact Ang','Deflect'};
  S.FontSize=10;
  Label = uicontrol('Parent',tab6,...
      'Style', 'text',...
      'Position', [0 600 700 20],...
      'String', 'BASIC DATA',...
      'FontSize',12);

end
%%%
if chFii == 1
    j=1;

    if ch2==1
for rang=round(Print_start):Print_Int:round(max(Range_L),-2)
 elev=Fit_range_L(rang);
 [POS_V,i_V,~,~,~,~]=PMM(V+1,elev,dia,mass,aero,0,0,0,0,0,0);
 r_V_inc=POS_V(i_V-1,1)-rang;
  [POS_V,i_V,~,~,~,~]=PMM(V-1,elev,dia,mass,aero,0,0,0,0,0,0);
 r_V_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0.51444,0,0,0,0);
 r_Wt=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,-0.51444,0,0,0,0);
 r_Wh=POS_V(i_V-1,1)-rang;
  [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0,0.514444,0,0,0);
 r_Wc=POS_V(i_V-1,1)-rang;
[POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0,0,1,0,0);
 r_temp_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0,0,-1,0,0);
 r_temp_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0,0,0,1,0);
 r_dencity_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0,0,0,-1,0);
 r_dencity_dec=POS_V(i_V-1,1)-rang;
   r_mass_inc=0;
   r_mass_dec=0;


Table_Fii(j,:)=round([rang,r_V_inc,r_V_dec,r_Wt,r_Wh,r_Wc,r_temp_inc,r_temp
_dec...
      ,r_dencity_inc,r_dencity_dec,r_mass_inc,r_mass_dec],1);
 j=j+1;
end
for rang=round(Print_max):-Print_Int:round(min(Range_U),-2)
    elev=Fit_range_U(rang);
 [POS_V,i_V,~,~,~,~]=PMM(V+1,elev,dia,mass,aero,0,0,0,0,0,0);
```

```
 r_V_inc=POS_V(i_V-1,1)-rang;
  [POS_V,i_V,~,~,~,~]=PMM(V-1,elev,dia,mass,aero,0,0,0,0,0,0);
 r_V_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0.51444,0,0,0,0);
 r_Wt=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,-0.51444,0,0,0,0);
 r_Wh=POS_V(i_V-1,1)-rang;
  [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0,0.514444,0,0,0);
 r_Wc=POS_V(i_V-1,1)-rang;
[POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0,0,1,0,0);
 r_temp_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0,0,-1,0,0);
 r_temp_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0,0,0,1,0);
 r_dencity_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=PMM(V,elev,dia,mass,aero,0,0,0,0,-1,0);
 r_dencity_dec=POS_V(i_V-1,1)-rang;
   r_mass_inc=0;
   r_mass_dec=0;


Table_Fii(j,:)=round([rang,r_V_inc,r_V_dec,r_Wt,r_Wh,r_Wc,r_temp_inc,r_temp
_dec...
      ,r_dencity_inc,r_dencity_dec,r_mass_inc,r_mass_dec],1);

  j=j+1;
end

    end

    if ch3==1
for rang=round(Print_start):Print_Int:round(max(Range_L),-2)
 elev=Fit_range_L(rang);

 [POS_V,i_V,~,~,~,~]=MPMM(V+1,elev,dia,mass,aero,iner,tc,0,0,0,0,0,0);
 r_V_inc=POS_V(i_V-1,1)-rang;
  [POS_V,i_V,~,~,~,~]=MPMM(V-1,elev,dia,mass,aero,iner,tc,0,0,0,0,0,0);
 r_V_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0.51444,0,0,0,0);
 r_Wt=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,-0.51444,0,0,0,0);
 r_Wh=POS_V(i_V-1,1)-rang;

[POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0,0.514444,0,0,0);
 r_Wc=POS_V(i_V-1,1)-rang;
[POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0,0,1,0,0);
 r_temp_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0,0,-1,0,0);
 r_temp_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0,0,0,1,0);
 r_dencity_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0,0,0,-1,0);
 r_dencity_dec=POS_V(i_V-1,1)-rang;
   r_mass_inc=0;
   r_mass_dec=0;


Table_Fii(j,:)=round([rang,r_V_inc,r_V_dec,r_Wt,r_Wh,r_Wc,r_temp_inc,r_temp
_dec...
      ,r_dencity_inc,r_dencity_dec,r_mass_inc,r_mass_dec],1);
 j=j+1;
end
```

```
for rang=round(Print_max):-Print_Int:round(min(Range_U),-2)
    elev=Fit_range_U(rang);
 [POS_V,i_V,~,~,~,~]=MPMM(V+1,elev,dia,mass,aero,iner,tc,0,0,0,0,0,0);
 r_V_inc=POS_V(i_V-1,1)-rang;
  [POS_V,i_V,~,~,~,~]=MPMM(V-1,elev,dia,mass,aero,iner,tc,0,0,0,0,0,0);
 r_V_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0.51444,0,0,0,0);
 r_Wt=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,-0.51444,0,0,0,0);
 r_Wh=POS_V(i_V-1,1)-rang;

[POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0,0.514444,0,0,0);
 r_Wc=POS_V(i_V-1,1)-rang;
[POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0,0,1,0,0);
 r_temp_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0,0,-1,0,0);
 r_temp_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0,0,0,1,0);
 r_dencity_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=MPMM(V,elev,dia,mass,aero,iner,tc,0,0,0,0,-1,0);
 r_dencity_dec=POS_V(i_V-1,1)-rang;
   r_mass_inc=0;
   r_mass_dec=0;


Table_Fii(j,:)=round([rang,r_V_inc,r_V_dec,r_Wt,r_Wh,r_Wc,r_temp_inc,r_temp
_dec...
      ,r_dencity_inc,r_dencity_dec,r_mass_inc,r_mass_dec],1);

  j=j+1;
end

    end

    if ch4==1
for rang=round(Print_start):Print_Int:round(max(Range_L),-2)
 elev=Fit_range_L(rang);

 [POS_V,i_V,~,~,~,~]=DOF5(V+1,elev,dia,mass,aero,iner,tc,0,0,0,0,0,0);
 r_V_inc=POS_V(i_V-1,1)-rang;
  [POS_V,i_V,~,~,~,~]=DOF5(V-1,elev,dia,mass,aero,iner,tc,0,0,0,0,0,0);
 r_V_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0.51444,0,0,0,0);
 r_Wt=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,-0.51444,0,0,0,0);
 r_Wh=POS_V(i_V-1,1)-rang;

[POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0,0.514444,0,0,0);
 r_Wc=POS_V(i_V-1,1)-rang;
[POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0,0,1,0,0);
 r_temp_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0,0,-1,0,0);
 r_temp_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0,0,0,1,0);
 r_dencity_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0,0,0,-1,0);
 r_dencity_dec=POS_V(i_V-1,1)-rang;
   r_mass_inc=0;
   r_mass_dec=0;
```

```
Table_Fii(j,:)=round([rang,r_V_inc,r_V_dec,r_Wt,r_Wh,r_Wc,r_temp_inc,r_temp
_dec...
      ,r_dencity_inc,r_dencity_dec,r_mass_inc,r_mass_dec],1);
 j=j+1;
end
for rang=round(Print_max):-Print_Int:round(min(Range_U),-2)
    elev=Fit_range_U(rang);
 [POS_V,i_V,~,~,~,~]=DOF5(V+1,elev,dia,mass,aero,iner,tc,0,0,0,0,0,0);
 r_V_inc=POS_V(i_V-1,1)-rang;
  [POS_V,i_V,~,~,~,~]=DOF5(V-1,elev,dia,mass,aero,iner,tc,0,0,0,0,0,0);
 r_V_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0.51444,0,0,0,0);
 r_Wt=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,-0.51444,0,0,0,0);
 r_Wh=POS_V(i_V-1,1)-rang;

[POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0,0.514444,0,0,0);
 r_Wc=POS_V(i_V-1,1)-rang;
[POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0,0,1,0,0);
 r_temp_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0,0,-1,0,0);
 r_temp_dec=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0,0,0,1,0);
 r_dencity_inc=POS_V(i_V-1,1)-rang;
 [POS_V,i_V,~,~,~,~]=DOF5(V,elev,dia,mass,aero,iner,tc,0,0,0,0,-1,0);
 r_dencity_dec=POS_V(i_V-1,1)-rang;
   r_mass_inc=0;
   r_mass_dec=0;


Table_Fii(j,:)=round([rang,r_V_inc,r_V_dec,r_Wt,r_Wh,r_Wc,r_temp_inc,r_temp
_dec...
      ,r_dencity_inc,r_dencity_dec,r_mass_inc,r_mass_dec],1);

  j=j+1;
end


    end

    tab7 = uitab(m,'Title','Table F(II)');
  S=uitable(tab7,'data',Table_Fii,'Position',[50 50 600 500]...
      ,'ColumnWidth',{'auto' 'auto' 'auto' 'auto' 'auto' 'auto' 'auto'
'auto' 'auto' 'auto'...
      'auto' 'auto'});
  S.ColumnName={'Range','Vel +1','Vel -1','Tail wind','Head wind','Cross
Wind',...
      'Temp +1','Temp -1','denc +1','denc -1','mass +1','mass -1'};
  S.FontSize=10;
  Label = uicontrol('Parent',tab7,...
      'Style', 'text',...
      'Position', [0 600 700 20],...
      'String', 'CORRECTIONS TO BEARING',...
      'FontSize',12);
end
%%%
if chA==1
  tab1 = uitab(m,'Title','Table A');

  Label = uicontrol('Parent',tab1,...
      'Style', 'text',...
      'Position', [0 600 700 20],...
```

```
      'String', 'LINE NUMBERS OF THE METEOROLOGICAL MESSAGE AS A FUNCTION
OF QUADRANT ELEVATION !! EXTRA WINDOW',...
      'FontSize',12);
end
%%%
if chB == 1

  tab2 = uitab(m,'Title','Table B');

  Label = uicontrol('Parent',tab2,...
      'Style', 'text',...
      'Position', [0 600 700 20],...
      'String', 'COMPLEMENTARY CORRECTION RANGE IN METRER FOR DIFFERENCE IN
ALTITUDE !! EXTRA WINDOW',...
      'FontSize',12);
end
%%%
if chC == 1

  tab3 = uitab(m,'Title','Table C');

  Label = uicontrol('Parent',tab3,...
      'Style', 'text',...
      'Position', [0 600 700 20],...
      'String', '?COMPONENTS OF A ONE KNOT WIND !! EXTRA WINDOW',...
      'FontSize',12);
end
%%%
if chD == 1
  tab4 = uitab(m,'Title','Table D');

  Label = uicontrol('Parent',tab4,...
      'Style', 'text',...
      'Position', [0 600 650 40],...
      'String', {'CORRECTIONS  TO BALLISTIC AIR TEMPERATURE AND BALLISTIC
AIR DENSITY AS A PERCENTAGE FOR DIFFERENCE & ALTITUDE BETWEEN BATTERY AND
MET DAIUM PLANE !! EXTRA WINDOW'},...
      'FontSize',12);
end
```

PMM

```
function
[POS,i,TIME,TOF,IMP_ang,Max_height]=PMM(V,elev,D,mass,aero,Y,Wt,Wc,dtemp,dd
ensity,dmass)
%simulation Time
TIME=clock();

% projectile_info
mass = mass + dmass;

% intial condition
lat=0;          %latitude
AZ=0;           %Azmuth

v=zeros(2,3);   %Velocity victor
v(1,:)=V*([cosd(elev) sind(elev) 0]);

% intial calculation
R=[0 -6356766 0];       %Earth raduis
```

```
go=9.80665*(1-.0026*cosd(2*lat)); %intial gravity
earth_rot=7.29e-5;      %Earth rotation speed
w=earth_rot*[cosd(lat)*cosd(AZ) sind(lat) -1*cosd(lat)*sind(AZ)]; %Earth
rotation velocity
%
pos=[0 Y 0;0 0 0];      %Position
POS=zeros(15000,3);     %Position recorder
ANG=zeros(15000,3);     %Angle recorder
wind=[Wt Wc];

% Pointers and Times
n=1;
i=1;
TOF=[0 0]; %Time of Flight
t=.0001;   %Time step
seg_p=.1/t;%Print pointer

% convertor
elev=elev*(pi/180);

% Integration loop
while pos(1,2) >= 0 || pos(1,1) < 100 %min range 100

    %atmosphere parameters
    [~,ro,Va,~]=atmostd(pos(1,2),dtemp,ddensity);

    %forces constant
    K=.5*(pi*((D/2)^2))*ro;

    %Interpolation
    vr=[v(1,1)+wind(1) v(1,2) v(1,3)+wind(2)];
    mach=norm(vr)/Va;
    cdo=interpo(aero(:,1),aero(:,2),mach);

    %force claculation
    DF=(-1*K*(cdo)*V*vr)/mass;          %Drag force = af/mass

    r=pos(1,:)-R;
    G=-1*go*(norm(R)^2 / norm(r)^3)*r; %Gravity

    A=-2*cross(w,v(1,:));               %Coriolis effect

    a=DF+G+A;                           %Acceleration

    %next step inegration
    v(2,:)=v(1,:)+(a*t);
    pos(2,:)=pos(1,:)+(v(2,:)*t);
    V=norm(v(2,:));

    %
    TOF(2)=TOF(1)+t;

    %old step reset
    TOF(1)=TOF(2);
    v(1,:)=v(2,:);
    pos(1,:)=pos(2,:);
    n=n+1;

    %Recording
 if rem(n,seg_p)== 0 || n==2
```

```
    POS(i,:)=pos(1,:);
    ANG(i,1)=atan(v(1,2)/v(1,1));
    i=i+1;
 end

    %warning overtime
     if TOF(1) > 200
         disp('SOMETHIG WRONG');
     break
     end

end

POS(i-1,1)=interpo([POS(i-2,2) pos(1,2)] , [POS(i-2,1) pos(1,1)], 0);
POS(i-1,3)=interpo([POS(i-2,2) pos(1,2)] , [POS(i-2,3) -pos(1,3)], 0);
POS(i-1,2)=0;

% Results
Max_height=max(POS(:,2));
ang=rad2deg(ANG(i-1,1));
% POS(i-1,1:3)
IMP_ang=rad2deg(atan(v(2,2)/v(2,1)));

TIME=clock()-TIME;
end
```

5 degree of freedom

```
function
[POS,i,AE,TIME,TOF,spin_re,IMP_ang,Max_height]=DOF5(V,eleva,D,mass,aero,ine
r,tc,Y,Wt,Wc,dtemp,ddensity,dmass)
TIME=clock();

Sar=pi*((D/2)^2);
mass = mass + dmass;

%Import_aerodynamic data from AERODYN6
Ix=interpo(iner(:,1),iner(:,2),mass-.7);
Iy=interpo(iner(:,1),iner(:,3),mass-.7);

%intial condition
alpha=[0 0 0];
lat=0;
AZ=0/800*45;

pos=[0 Y 0];
elev=[eleva AZ];
v=V*([cosd(elev(1,1))*cosd(AZ) sind(elev(1,1))*cosd(AZ) sind(AZ)]);
x=[cosd(elev(1,1))*cosd(AZ) sind(elev(1,1))*cosd(AZ) sind(AZ)];

%intial calculation
spino=(2*pi*V)/(tc*D);
spin=spino;
H=Ix*spin*x + Iy*crossi(x,[0 0 0]);
%
R=[0 -6356766 0];
go=9.80665*(1-0.0026*cosd(2*lat));
```

```
earth_rot=7.29e-5;
w=earth_rot*[cosd(lat)*cosd(AZ) sind(lat) -1*cosd(lat)*sind(AZ)];
%
POS=zeros(2000,3);
AE=zeros(2000,1);

%
n=1;
i=1;
TOF(1)=0;

%
wind=[Wt Wc];
DX=[0 0];
m=[0 0];
DSPIN=[0 0];
acc=[0 0];
%convertor
elev=elev*(pi/180);

%print out
   t=0.1/(spin/2/pi);

if 1/t==0
    seg_p=1/0.001;
else
    seg_p=round(1/t);
end
while pos(1,2) >= 0 || pos(1,1) < 600
 t=max(.000001,.000001/DX(1));
 t=min(t,.0002);

%           atmosphere parameters
            [~,ro,Va,~]=atmostd(pos(1,2),dtemp,ddensity);
%           forces constant
            K=.5*Sar*ro;
%           Interpolation
            vr(1,:)=[v(1,1)+wind(1) v(1,2) v(1,3)+wind(2)];
            mach=norm(vr(1,:))/Va;
            cdo=interpo(aero(:,1),aero(:,2),mach);
            cda2=interpo(aero(:,1),aero(:,3),mach);
            cla=interpo(aero(:,1),aero(:,5),mach);
            cla3=interpo(aero(:,1),aero(:,6),mach);
            cma=interpo(aero(:,1),aero(:,7),mach);
            cma3=interpo(aero(:,1),aero(:,8),mach);
            cmq=interpo(aero(:,1),aero(:,9),mach);
            cl=interpo(aero(:,1),aero(:,10),mach);
            clp=interpo(aero(:,1),aero(:,11),mach);
            cypa=interpo(aero(:,1),aero(:,12),mach);
            cypa3=interpo(aero(:,1),aero(:,13),mach);
            cmpa=interpo(aero(:,1),aero(:,14),mach);
            cmpa3=interpo(aero(:,1),aero(:,15),mach);

%   force and moment claculation
    DF=(-1*K* (cdo+(cda2*((norm(alpha)^2)))) * V * vr(1,:)) / mass;
    LF=K*(cla+(cla3*(norm(alpha)^2)))*(((V^2)*x)-
(doti(vr(1,:),x)*vr(1,:)))/mass;

    MF=-
1*(K*D*(cypa+(cypa3*((norm(alpha)^2)))))*doti(H,x)*crossi(x,vr(1,:))/(Ix*ma
ss);
```

```
    PDF=K*(D/Iy)*(cmq)*V*crossi(H,x)/mass;
    TF=0;...((dmf*Isp)+((Pr- P )*Ae))*x/mass;

    r=pos(1,:)-R;
    G=-1*go*(norm(R)^2 / norm(r)^3)*r;
    A=-2*crossi(w,v(1,:));

    a=DF+LF+MF+PDF+TF+G+A;
    acc(1)=norm(a);
    if acc(1)>acc(2)
        acc(2)=acc(1);
    end
    fin_cant=0;
    %overturning
    OM=K*D*(cma+(cma3*(norm((alpha))^2)))*V*crossi(vr(1,:),x);
    %pitch damping
    PDM=K*((D^2)/Iy)*(cmq)*V*(H-(doti(H,x)*x));
    %magnus
    MM=-
K*((D^2)/Ix)*(cmpa+(cmpa3*(norm(alpha)^2)))*doti(H,x)*((doti(vr(1,:),x)*x)-
vr(1,:));
    %spin damping
    SDM=K*((D^2)/Ix)*clp*V*doti(H,x)*x;
    %fin cant
    FCM=K*D*cl*fin_cant*(V^2)*x;
    %jet damping
    AJDM=[0 0 0];...(dm*(rne^2)/2*Ix)*dot(H,pos(n,:))*pos(n,:);
    TJDM=[0 0 0];...(dm*re*rt/Iy)*(H-(dot(H,pos(n,:))*pos(n,:)));

    M=OM+PDM+MM+SDM+FCM+AJDM+TJDM;
    m(1)=norm(M);
    if m(1)>m(2)
        m(2)=m(1);
    end
%   next step
    dx=crossi(H,x)/Iy;
    DX(1)=norm(dx);
     if DX(1)>DX(2)
         DX(2)=DX(1);
     end
    H=H+(M*t);
    %
    x=x+(dx*t);
    x=x/norm(x);

    v(2,:)=v(1,:)+(a*t);
    pos(2,:)=pos(1,:)+(v(2,:)*t);
    V=norm(v(2,:));
    %
    dspin=(pi*ro*(D^4)*spin*V*clp)/(16*Ix);
     DSPIN(1)=norm(dspin);
     if DSPIN(1)>DSPIN(2)
         DSPIN(2)=DSPIN(1);
     end
    spin=spin+(dspin*t);
    %
    alpha=acos(doti(v(1,:),x)/norm(v(1,:)));
    TOF(2)=TOF(1)+t;

%   simulation timing
    TOF(1)=TOF(2);
    v(1,:)=v(2,:);
```

```
    pos(1,:)=pos(2,:);
    n=n+1;

     if rem(n,seg_p)== 0||n==2
                 i=i+1;
          POS(i,1)=pos(1,1);

          POS(i,2)=pos(1,2);
          POS(i,3)=-pos(1,3);
          AE(i,:)=rad2deg(norm(alpha));

     end
          if AE(i-1,:)>40

          end

     if TOF(1) > 400

         disp('SOMETHIG WRONG');
     break
     end
end
POS(i,1)=interpo([POS(i-2,2) pos(1,2)] , [POS(i-2,1) pos(1,1)], 0);
POS(i,3)=interpo([POS(i-2,2) pos(1,2)] , [POS(i-2,3) -pos(1,3)], 0);
POS(i,2)=0;
Max_height=max(POS(:,2));

IMP_ang=rad2deg(atan(v(2,2)/v(2,1)));
spin_re=0;

end
```

MPMM

```
function
[POS,i,AE,TIME,TOF,spin_re,IMP_ang,Max_height]=MPMM(V,eleva,D,mass,aero,ine
r,tc,Y,Wt,Wc,dtemp,ddensity,dmass)
TIME=clock();
%projectile_info
mass = mass + dmass;

%Import_aerodynamic data from AERODYN6
Ix=interpo(iner(:,1),iner(:,2),mass-.7);

%intial condition
ae=[0 0 0];
lat=0;
AZ=0;

pos=[0 Y 0];
elev=[eleva AZ];
v=V*([cosd(elev(1,1))*cosd(AZ) sind(elev(1,1))*cosd(AZ) sind(AZ)]);

%intial calculation
spino=(2*pi*V)/(tc*D);
spin=spino;
%
R=[0 -6356766 0];
go=9.80665*(1-.0026*cosd(2*lat));
```

```
earth_rot=7.29e-5;
w=earth_rot*[cosd(lat)*cosd(AZ) sind(lat) -1*cosd(lat)*sind(AZ)];
%
POS=zeros(15000,3);
AE=zeros(15000,1);
spin_re=zeros(15000,1);
%
n=1;
i=1;
TOF(1)=0;

%
wind=[Wt Wc];

%convertor
elev=elev*(pi/180);

%print out
  t=.0001;
seg_p=.1/t;

while pos(1,2) >= 0 || pos(1,1) < 100
    %atmosphere parameters
    [~,ro,Va,~]=atmostd(pos(1,2),dtemp,ddensity);
    %forces constant
    K=.5*pi*((D/2)^2)*ro;
    vr(1,:)=[v(1,1)+wind(1) v(1,2) v(1,3)+wind(2)];

    %Interpolation
    mach=norm(vr(1,:))/Va;
    cdo=interpo(aero(:,1),aero(:,2),mach);
    cda2=interpo(aero(:,1),aero(:,3),mach);
    cla=interpo(aero(:,1),aero(:,5),mach);
    cla3=interpo(aero(:,1),aero(:,6),mach);
    cma=interpo(aero(:,1),aero(:,7),mach);
    cma3=interpo(aero(:,1),aero(:,8),mach);
    clp=interpo(aero(:,1),aero(:,11),mach);
    cypa=interpo(aero(:,1),aero(:,12),mach);
    cypa3=interpo(aero(:,1),aero(:,13),mach);

    %force claculation
    MF=-1*(K*D*(cypa+(cypa3*(norm(ae)^2)))*spin)*cross(ae,vr(1,:))/mass;
    DF=(-1*K*(cdo+(cda2*(norm(ae)^2)))*V*vr(1,:))/mass;
    LF=K*(cla+(cla3*((norm(ae)^2))))*(V^2)*ae/mass;

    r=pos(1,:)-R;
    G=-1*go*(norm(R)^2 / norm(r)^3)*r;

    A=-2*cross(w,v(1,:));

    a=DF+LF+MF+G+A;

    %next step inegration
    v(2,:)=v(1,:)+(a*t);
    pos(2,:)=pos(1,:)+(v(2,:)*t);
    V=norm(v(2,:));
    %
    dspin=(pi*ro*(D^4)*spin*V*clp)/(16*Ix);
    spin=spin+(dspin*t);
    %Yaw of repose
```

```
    ae=(-
8*Ix*spin*cross(vr(1,:),a))/(pi*ro*(D^3)*(cma+(cma3*((norm(ae)^2))))*(V^4))
;
    %
    TOF(2)=TOF(1)+t;

    %next step reset
    TOF(1)=TOF(2);
    v(1,:)=v(2,:);
    pos(1,:)=pos(2,:);
    n=n+1;


 if rem(n,seg_p)== 0 || n==2
    POS(i,:)=pos(1,:);
    AE(i,:)=rad2deg(norm(ae));
    spin_re(i)=V;
    i=i+1;
 end

    %warning overtime
     if TOF(1) > 400
         disp('SOMETHIG WRONG');

     break
     end

end
POS(i,1)=interpo([POS(i-1,2) pos(1,2)] , [POS(i-1,1) pos(1,1)], 0);
POS(i,3)=interpo([POS(i-1,2) pos(1,2)] , [POS(i-1,3) -pos(1,3)], 0);
spin_re(i)=V;
POS(i,2)=0;
Max_height=max(POS(:,2));


IMP_ang=rad2deg(atan(v(2,2)/v(2,1)));
TIME=clock()-TIME;
end
```

Other Functions

```
    %Interpolation function
function [ y ] = interpo( X,Y,x )

n=1;
if X(n)>=x
    y=Y(n);
else
    while X(n)< x

n=n+1;
if n==40
   disp('error')
   break
end
    end
y=((Y(n)-Y(n-1))*(x-X(n-1)) / (X(n)-X(n-1)))+Y(n-1);
end
```

```
end

     %Atmosphere function
function [p,ro,Speed_Sound,Temp] = atmostd(h,dt,dd)

HIGHT=[-1000,-
500,0,250,500,750,1000,1250,1500,1750,2000,2250,2500,2750,3000,3250,3500,37
50,4000,4250,4500,4750,5000,5500,6000,6500,7000,7500,8000,9000,10000,11000,
12000,13000,14000,15000,16000,17000,18000,19000,20000,21000,22000,23000,240
00,25000,26000,27000,28000,29000,30000,31000,32000,33000,34000,35000,36000,
37000,38000,39000,40000,42000,44000,46000,48000,50000,55000,60000,70000,800
00,90000,100000,100000000];
PRESSURE=[113930,107480,101330,98357,95461,92634,89876,87185,84560,81999,79
501,77066,74692,72377,70121,67923,65780,63693,61660,59680,57753,55875,54048
,50539,47218,44075,41105,38300,35652,30801,26500,22700,19399,16580,14170,12
112,10353,8849.70000000000,7565.20000000000,6467.50000000000,5529.300000000
00,4728.90000000000,4047.50000000000,3466.90000000000,2971.70000000000,2549
.20000000000,2188.40000000000,1880,1616.20000000000,1390.40000000000,1197,1
031.30000000000,889.060000000000,767.310000000000,663.410000000000,574.5900
00000000,498.520000000000,433.250000000000,377.140000000000,328.82000000000
0,287.140000000000,219.970000000000,169.500000000000,131.340000000000,102.3
00000000000,79.7790000000000,42.7520000000000,22.4610000000000,5.5205000000
0000,1.03660000000000,0.164380000000000,0.0300750000000000,0.03007500000000
00];
AIR_VELOCITY=[344.110000000000,342.210000000000,340.290000000000,339.300000
000000,338.370000000000,337.400000000000,336.440000000000,335.500000000000,
334.490000000000,333.500000000000,332.530000000000,331.500000000000,330.560
000000000000,329.600000000000,328.580000000000,327.600000000000,326.5900000000
00,325.600000000000,324.590000000000,323.600000000000,322.570000000000,321.
600000000000,320.550000000000,318.500000000000,316.450000000000,314.4000000
00000,312.310000000000,310.200000000000,308.110000000000,303.850000000000,2
99.530000000000,295.140000000000,295.070000000000,295.070000000000,295.0700
00000000,295.070000000000,295.070000000000,295.070000000000,295.07000000000
0,295.070000000000,295.070000000000,295.070000000000,296.380000000000,297.0
50000000000,297.720000000000,298.390000000000,299.060000000000,299.72000000
0000,300.390000000000,301.050000000000,301.710000000000,302.370000000000,30
3.030000000000,304.670000000000,306.490000000000,308.300000000000,310.10000
0000000,311.890000000000,313.670000000000,315.430000000000,317.190000000000
,320.670000000000,324.120000000000,327.520000000000,329.800000000000,329.80
0000000000,326.700000000000,320.610000000000,297.140000000000,269.440000000
000,269.440000000000,269.440000000000,269.440000000000];
TEMP=[294.646201704771,291.399413915527,288.159073591186,286.510638968152,2
84.887792006577,283.270133092158,281.634902554590,280.028682491454,278.3995
98887547,276.768795266818,275.135107738514,273.525886793848,271.90374361680
7,270.277439662233,268.655129170021,267.032713032338,265.406876011573,263.7
84092023665,262.158780199907,260.535647240661,258.917567552036,257.29270185
4785,255.669252159539,252.424472578219,249.184842408186,245.936152714431,24
2.693566902156,239.456045365222,236.211980617831,229.732525582168,223.24934
9740061,216.771095712304,216.639986791731,216.648021086015,216.636735419804
,216.655012426494,216.650941047752,216.647655618843,216.639999460261,216.63
7404287635,216.645607889204,217.574777913510,218.569803001479,219.564233391
019,220.551862151634,221.545636980447,222.539815541173,223.537412675092,224
.526282912201,225.515432210145,226.501304289879,227.498337729008,228.487209
970579,230.969827704419,233.738501156729,236.506750743832,239.277549040834,
242.045845702232,244.812571493966,247.580952821202,250.340530209278,255.873
892347546,261.398470696066,266.926199087268,270.656925042927,270.6392490513
84,265.593535093532,255.771153765931,219.698465434136,180.646168941406,180.
642551644286,21063.4836715160,21063.4836715160];
n=1;

    if HIGHT(n)>=h
```

```
        p=PRESSURE(n);
        Temp=TEMP(n);
        ro=(p/287.058/Temp);
        Speed_Sound=AIR_VELOCITY(n);
    else

    while HIGHT(n)< h

    n=n+1;
         if n==80
         disp('error')
         break
         end
    end

    p=((PRESSURE(n)-PRESSURE(n-1))*(h-HIGHT(n-1)) / (HIGHT(n)-HIGHT(n-
1)))+PRESSURE(n-1);
    Temp= (((TEMP(n)-TEMP(n-1))*(h-HIGHT(n-1)) / (HIGHT(n)-HIGHT(n-
1)))+TEMP(n-1));
    Temp=Temp+(Temp*dt/100);
    Speed_Sound=((AIR_VELOCITY(n)-AIR_VELOCITY(n-1))*(h-HIGHT(n-1)) /
(HIGHT(n)-HIGHT(n-1)))+AIR_VELOCITY(n-1);
    ro=(p/287.058/Temp );
    ro=ro+(ro*dd/100);

    end
end


    %Aerodynamics coefficients function

function [ aero,INER,tc ] = aero_data( d )

aero=zeros(13,21);

if d==81

aero(:,1)=[0.01 .5 .6 .8 .88 .9 .95 1 1.05 1.1 1.2 1.35 1.5];
aero(:,2)=[.157 .178 .168 .173 .244 .264 .507 .535 .621 .663 .642 .6 .567];
aero(:,3)=[3.125 3.125 3.125 3.125 4.25 5 5 5.25 5.75 6.5 6.75 6.5 6];
aero(:,4)=zeros(1,13);
aero(:,5)=[1.75 1.75 1.75 1.75 1.8 2 2 2.05 2.1 2.15 2.15 2.2 2.25];
aero(:,6)=[-1.3 -1.3 -1.4 -1.7 -2 -2.3 -2.6 -2.8 -2.8 -2.8 -2.8 -2.8 -2.8
];
aero(:,7)=[-1.8 -1.8 -1.8 -2.1 -2.2 -2.4 -2.4 -2.5 -2.6 -2.7 -2.6 -2.5 -
2.4];
aero(:,8)=[-2.5 -2.5 -2.5 -3 -3 -3.5 -3.5 -3.8 -4 -4.2 -4.4 -4.5 -4.5];
aero(:,9)=[-55.7 -56 -56 -57 -57.5 -58 -58 -58.5 -59 -59.5 -60 -60.5 -61];
aero(:,10)=[0.003 0.003 0.003 0.003 0.003 0.003 0.003 0.003 0.003 0.003
0.003 0.003 0.003];
aero(:,11)=[-.25 -.25 -.25 -.25 -.25 -.25 -.25 -.25 -.25 -.25 -.25 -.25 -
.25];
INER=[1 .003382 .02725;2000 .003382 .02725];
tc=1/0;
clearvars filename startRow formatSpec fileID dataArray ans;
return
end
aero=zeros(13,21);
aero(:,1)=[0.01 .5 .6 .8 .88 .9 .95 1 1.05 1.1 1.2 1.35 1.5];
INER=zeros(2,10);
tc=0;
```

```
clearvars filename startRow formatSpec fileID dataArray ans;
end
```

Published with MATLAB® R2017a