

2012

Simplified Interior Ballistics Analysis of a Tube Launched UAV Prototype

Edward R. Jackson
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Jackson, Edward R., "Simplified Interior Ballistics Analysis of a Tube Launched UAV Prototype" (2012). *Graduate Theses, Dissertations, and Problem Reports*. 4871.
<https://researchrepository.wvu.edu/etd/4871>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Simplified Interior Ballistics Analysis of a Tube Launched UAV Prototype

Edward R. Jackson

Thesis submitted to the Benjamin M. Statler
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science in Mechanical Engineering

Victor H. Mucino, Ph.D., Chair
Patrick Browning, Ph.D
Wade Huebsch, Ph.D

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia
2012

ABSTRACT

Simplified Interior Ballistics Analysis of a Tube Launched UAV Prototype

Edward R. Jackson

Structural analysis is a critical aspect in the successful design of tube launched projectiles, such as mortar rounds. Ongoing research conducted at West Virginia University has focused on a tube-launched, folding-wing UAV design inspired by mortars. This has driven the necessity of a structural analysis of the prototype design to provide vital feedback to designers to ensure that the UAV is likely to survive the act of launching. Due to the extreme accelerations during the launching phase, a typical mortar round experiences dramatic impulse loads for an extremely brief duration of time. Such loads are the result of the propellant combustion process. Thermodynamic-based interior ballistic computations have been formulated and were used to solve the dynamic equations of motion that govern the system. Modern ballistic programs solve these equations by modeling the combustion of the propellant. However, mathematical procedures for such analyses require complex models to attain accurate results. Consequently, the objective of this research is to create a ballistics program that can evaluate interior ballistics by using archived pressure-time data without having to simulate the propellant combustion in order to minimize the computational effort required. A program routine created for this purpose reduces the complexity of calculations to be performed, while maintaining a reasonable degree of accuracy for the motion dynamics results (temporal displacement, velocity, acceleration of the projectile) and thermodynamic results (combustion gas pressure and volume). Additionally, the program routine was used to produce a mathematical model describing the pressure as a function of time. Advanced simulations could then be conducted via explicit-dynamic finite element solvers such as ANSYS LS-DYNA using the ballistics code outputs as loading conditions to simulate the transient response and stress wave propagation of the prototype and individual payload components. Such simulations remove uncertainties related to the transient loads needed to assess the structural integrity of the projectile and its components. Results obtained from the simulations were compared for verification purposes to review the accuracy of the solutions. The program provided researchers with an effective design tool that may be used in the optimization of a successful structural design. The results obtained from the simulations will be examined in the context of this thesis.

ACKNOWLEDGMENTS

First of all, I would like to express my deepest appreciation to my advisor Dr. Victor Mucino. The guidance and support he has given to me throughout my educational career has strongly affected my intellectual growth and development as a student, and has prepared me for my future as an Engineer.

I would also like to thank my graduate committee members Dr. Pat Browning and Dr. Wade Huebsch. Dr. Browning's expert guidance has significantly enhanced my skills and appreciation of the design process. Furthermore, his teachings will always remind me of the reason I wanted to be an Engineer. The support and guidance I received from Dr. Huebsch has educated me of the consequences of the choices I make, a lesson that will forever impact the decisions I make as an Engineer and as a person.

Additionally, I would like to thank the project team members whom I have worked with over the past months for their insight, assistance, and patience.

I also am grateful to the MAE administrative staff, all of whom were extremely kind and helpful during my time at WVU.

To my parents, I cannot thank you enough for everything you have done for me in my lifetime. Without your wisdom, unconditional care, and support, I would not be the man I am today. I love you both.

To my beautiful wife Marisa, you are all I long for, all I worship, and adore. Your endless support and encouragement enabled me to continue and successfully complete my graduate education. A wife like you is scarcer than a blue moon. I love you with all my heart and soul.

Finally, I would like to dedicate this work to my son, Luca. I only hope that I can teach you as much in my lifetime as you have taught me over the last 4 months.



TABLE OF CONTENTS

Chapter	Page
ABSTRACT.....	ii
ACKNOWLEDGMENTS.....	iii
LIST OF TABLES.....	vi
LIST OF FIGURES.....	vii
LIST OF SYMBOLS AND SUBSCRIPTS	ix
CHAPTER 1 : INTRODUCTION	1
1.1 – System Description	1
1.2 – Motivation.....	3
1.3 – Objectives.....	4
1.4 – Importance of Research.....	6
1.5 – General Approach	6
1.6 – Scope of Research	8
CHAPTER 2 : BACKGROUND AND LITERATURE REVIEW	9
2.1 – Background	9
2.2 – Mathematical Modeling of Interior Ballistics	9
2.2.1 – XNOVAKTC Code and Extensions	11
2.2.2 – IBHVG2 Code and Extensions.....	12
2.2.3 – 3D-MIB Code and Extensions.....	13
CHAPTER 3 : METHODOLOGY	15
3.1 – Simplified Interior Ballistics Code	15
3.1.1 – Data Retrieval.....	15

3.1.2 – Data Curve Fitting	16
3.1.3 – Propellant Burn Rate.....	18
3.1.4 – Equations of Motion	20
3.1.5 – Thermodynamics.....	23
3.1.6 – LS-DYNA Loading Conditions.....	25
CHAPTER 4 : RESULTS and DISCUSSION	27
4.1 – Data Curve Fitting	27
4.2 – Propellant Burn Rate.....	29
4.3 – Equations of Motion	32
4.4 – Thermodynamics.....	37
CHAPTER 5 : CONCLUSION	42
REFERENCES	43
APPENDICES	48
Appendix A – MATLAB Code (Imperial Units).....	49
Appendix B – MATLAB Code (SI Units).....	53
Appendix C – Flowcharts of Computational Procedures	57

LIST OF TABLES

Table	Page
Table 1.1 – Specifications and Performance Data M721 Round and M224 Cannon”	2
Table 2.1 – Summary of Unknown Variables and Corresponding Equations	11
Table 3.1 – Excel Spreadsheet Example.....	16

LIST OF FIGURES

Figure 1.1 – M721 Mortar Round (M721 60mm Illuminating Mortar Cartridge, 1998)	1
Figure 1.2 – M224 Mortar System (Cooke G. W., 2004).....	2
Figure 1.3 – Projectile Prior to Combustion at Time = 0 (Asfaw, 2008)	4
Figure 1.4 – Projectile Motion from Exhaust Gas Expansion at Time = t (Asfaw, 2008)	4
Figure 1.5 – Projectile at Cannon Exit at Time = t_{exit} (Asfaw, 2008).....	5
Figure 1.6 – Flowchart Outlining General Approach of Interior Ballistics Analysis	7
Figure 2.1 – Illustration of Initial Lag in Pressure Observed in Comparison to IBHVG2 Solution	13
Figure 3.1 - Arbitrary Plot of Pressure-Time Data	17
Figure 3.2 – ANSYS Function Editor Screenshot	26
Figure 4.1 –Curve Fit and Archived Pressure Data Comparison (Non-Dimensional)	27
Figure 4.2 – Deformation of Imperial Curve Fit Plot Due to Rounding (Non-Dimensional).....	28
Figure 4.3 – Deformation of SI Curve Fit Plot Due to Rounding (Non-Dimensional)	28
Figure 4.4 – Predicted Propellant Mass (Non-Dimensional)	30
Figure 4.5 – Hypothetical Mass Fraction (Heiser, 1991).....	30
Figure 4.6 – Mass Fraction Results (Non-Dimensional).....	31
Figure 4.7 –Projectile Displacement Results (Non-Dimensional).....	33
Figure 4.8 – Projectile Velocity Results (Non-Dimensional)	34
Figure 4.9 – Computed Projectile Dynamics (Non-Dimensional)	35
Figure 4.10 – Computed Projectile Dynamics of a 120-mm Mortar Round (Acharya, 2009).....	35
Figure 4.11 – Projectile Acceleration Results (Non-Dimensional).....	36
Figure 4.12 –Volume behind Projectile Results (Non-Dimensional)	37
Figure 4.13 –Combustion Gas Temperature Results (Non-Dimensional).....	38

Figure 4.14 – Hypothetical Combustion Gas Temperature (Shelton, Bergles, & Saha, 1973)	39
Figure 4.15 –Local Speed of Sound Results (Non-Dimensional).....	40
Figure C1 – Flowchart of Propellant Burnout Time Computation	57
Figure C2 – Flowchart of Propellant Mass/Geometry Computations	58
Figure C3 – Flowchart of Kinematic Computations	59
Figure C4 – Flowchart of Thermodynamic Computations	60
Figure C5 – Flowchart of Helmholtz Resonance Computation	60

LIST OF SYMBOLS AND SUBSCRIPTS

SYMBOLS

a – acceleration
b – Co-volume
c – local speed of sound
 c_v – specific heat constant volume
d – diameter
 f_H – Helmholtz frequency
g – gravitational acceleration
m – mass
n – burn rate exponent
r – radius
t – time
v – velocity
x – displacement
A – cross sectional area
BR – burn rate
F – force
L – length
M – molar mass
N – mass of propellant burnt
P – pressure
R – gas constant
T – gas temperature
 T_f – adiabatic flame temperature
V – volume
 β – burn rate coefficient
 γ – specific heat ratio
 ρ – density

SUBSCRIPTS

air – air
amb – ambient
base – projectile base
BO – burn out
BRN – burning
cb – cannon barrel
cf – curve fit
ch – propelling charge
cham – chamber
cof – coefficients
cumsum – cumulative summation
delta – change in
eqv – equivalent
exp – experimental
exit – exit condition
f – final condition
gas – gas
i – initial condition
IMP – imperial units
muz – muzzle
ob – obturating ring
prch – projectile & charge
prj – projectile
SI – SI units
vec – vector

CHAPTER 1: INTRODUCTION

The purpose of this research was to analyze the interior ballistics of a tube launched UAV (TL-UAV) prototype inspired by traditional mortar rounds. The approach taken presented in this thesis involves the creation of a simplified interior ballistics simulation in MATLAB that replaces solid propellant combustion modeling with archived pressure-time data. Output results from the simulation could then be used as loading conditions in a more computationally intense finite element analysis (FEA) using ANSYS to simulate the transient response and stress wave propagation acting upon prototype components to predict the response and avoid failure. Result verification was conducted by comparing the simulated muzzle velocity obtained from each MATLAB model with performance data from public domain literature.

1.1 – System Description

Design of the TL-UAV was inspired by M721 60mm mortar rounds. These mortar rounds are utilized in the M224 60mm mortar system by the U.S. military. The M224 mortar system is a “muzzle-loaded, smooth-bore, high-angle-of-fire weapon” consisting of a M225 cannon, M170 bipod, M7 baseplate, and M64-series sight unit that was introduced to the U.S. military in 1978 (Cooke G. W., 2004). Images of the mortar and launcher assembly are provided figures 1.1 and 1.2:

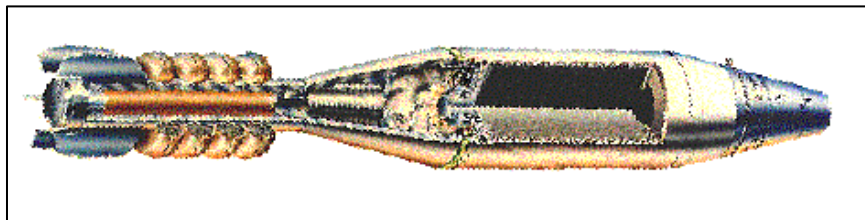


Figure 1.1 – M721 Mortar Round (M721 60mm Illuminating Mortar Cartridge, 1998)

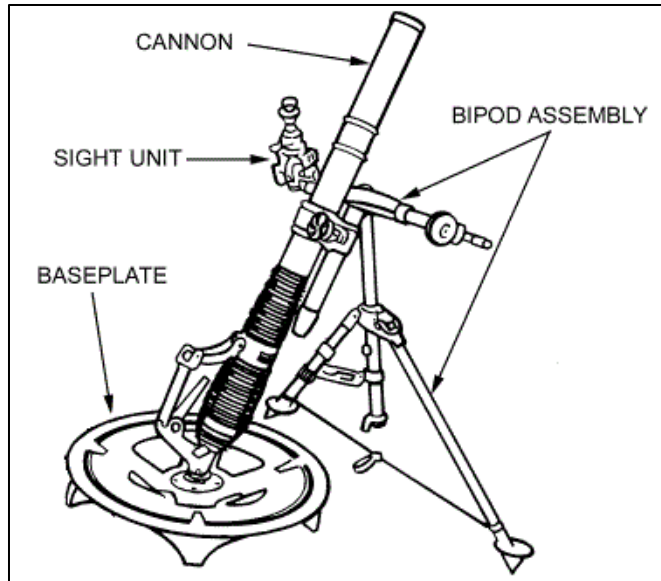


Figure 1.2 – M224 Mortar System (Cooke G. W., 2004)

Specifications and performance data for the M721 mortar round and M224 mortar system retrieved from public domain sources are listed in table 1.1:

Table 1.1 – Specifications and Performance Data M721 Round and M224 Cannon^{1,2,3,4}

	SI Units	Imperial Units	Source
Cartridge Weight	1.71 kg	3.76 lbm	Literature ¹
Cartridge Diameter	60 mm	2.362 in	Approximated
Cartridge Length	421 mm	16.58 in	Literature ¹
Cannon Bore Diameter	60 mm	2.362 in	Literature ²
Cannon Length	1024 mm	40.32 in	Literature ³
Muzzle Velocity	124 m/s	407 ft/s	Literature ⁴

Superficially, the prototype TL-UAV design closely resembles the appearance of the M721 mortar round. Internally, however, the TL-UAV conceals an array of electronic components and mechanisms necessary to actuate internally-stowed wings to complete the transformation of the prototype from its initial mortar round orientation to its final UAV orientation. Due to ITAR restrictions, the actual geometry of the prototype cannot be displayed in the contents of this thesis. Therefore, the focus of this thesis will be on the procedure to be used for analysis of the

¹ (Cooke G. W., 2004) – 60mm Mortar Ammunition and Fuzes

² (Cooke G. W., 2004) – M224 60mm Light Mortar

³ (U.S. Army, 2009)

⁴ (M721 60mm Illuminating Mortar Cartridge, 1998)

TL-UAV using an arbitrarily designed 60mm mortar resembling the M721 round. Results discussed in the context of this thesis have been non-dimensionalized due to export control regulations.

1.2 – Motivation

Structural analysis is a vital aspect of the design process for tube launched projectiles. During the act of launch, extreme impulse forces applied to the projectiles occur in a matter of milliseconds. Such forces can ultimately lead to structural failure and undesired damage of the projectile, particularly payload or projectile components. Determination of the forces acting upon the projectile required an interior ballistics analysis of the system to obtain physically representative loading conditions. This has driven the need for an interior ballistics model of the prototype that accurately represented the anticipated behavior of an actual launch. Results attained from the analysis will provide significant feedback to designers to ensure that the TL-UAV projectile will survive the launch, or where modifications of the prototype may be required to prevent failure from occurring. With this in mind, certain considerations had to be made with respect to the analysis, namely minimizing simulation time of the analysis without dramatically compromising simulation accuracy.

Simulation of the interior ballistics process was an important aspect of the structural analysis. Solutions attained from such simulations provided data which could be used in determining loading conditions for a FE model of the TL-UAV. Literature survey performed pertaining to interior ballistics revealed the complex programming necessary to effectively simulate the interior ballistic cycle. Taking this into consideration, it was considered that a more efficient method could directly evaluate archived pressure-time data. The basis of this conceptual approach was that the propellant combustion process was erratic and unpredictable on a microscopic level, yet somewhat uniform on a macroscopic level. Therefore, archived data could be implemented in place of the numerous and complex computations required to approximate the temporal pressure distribution of the combustion gas.

1.3 – Objectives

Interior ballistics evaluates in-bore projectile motion caused by pressure changes resulting from energy addition due to solid propellant combustion. Prior to propellant combustion, at time $t_1=0$, the projectile is not affected by pressure changes, as depicted in figure 1.3. After the initiation of propellant combustion, at time $t_2=t$, exhaust gases expand causing an increase in pressure. The resulting pressure acts upon the projectile causing motion, as illustrated in figure 1.4. The combustion gas continues to expand, forcing the projectile to move toward the exit of the cannon barrel. Once the projectile has reached the cannon exit, at time $t_3=t_{exit}$, the combustion gases escape the system to seek equilibrium, as shown in figure 1.5.

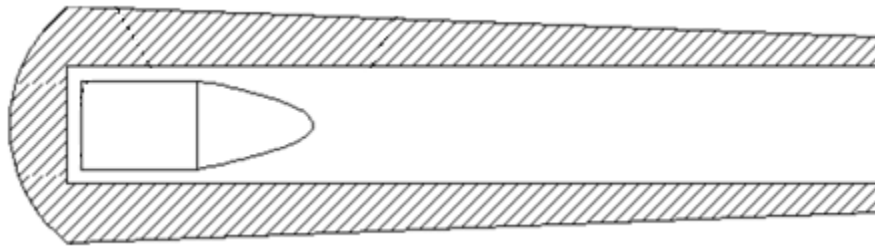


Figure 1.3 – Projectile Prior to Combustion at Time = 0 (Asfaw, 2008)

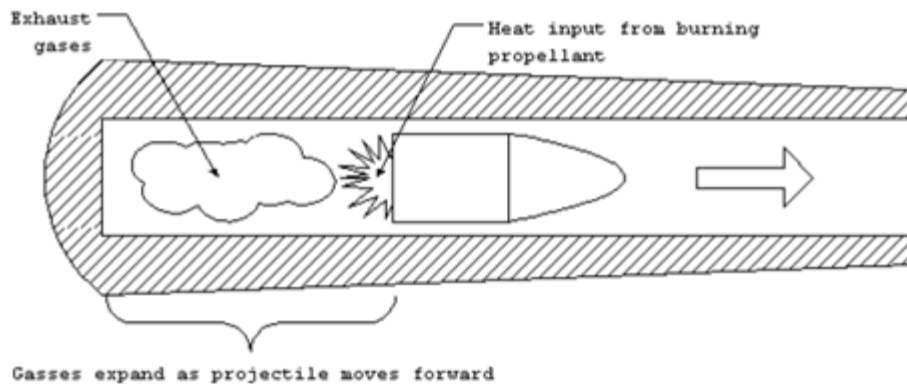


Figure 1.4 – Projectile Motion from Exhaust Gas Expansion at Time = t (Asfaw, 2008)

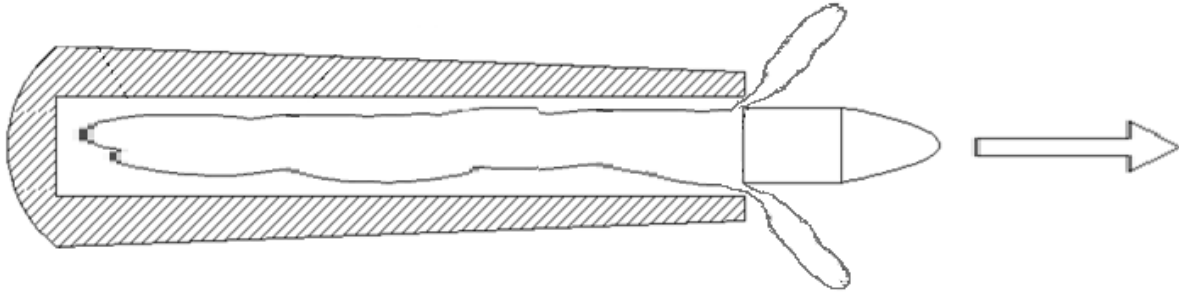


Figure 1.5 – Projectile at Cannon Exit at Time = t_{exit} (Asfaw, 2008)

Simulations of interior ballistics require numerous complex calculations to attain accurate results. Significant reduction in simulation time can be attained by implementation of archived pressure-time data in place of the tedious and lengthy computations required to model solid propellant combustion. Further simplification can be achieved by conducting the simulation one-dimensionally. Considering these points, the desired objectives for the simplified interior ballistics code include:

- Solving the equations of motion to determine the displacement, velocity, and acceleration of the projectile with respect to time within the cannon barrel and at the muzzle
- Partial evaluation of the thermodynamics and thermochemistry to accurately predict the volume behind the projectile and to approximate the combustion gas temperature
- Approximating the propellant burn rate to approximate propellant geometry during combustion
- Evaluating the local speed of sound and Helmholtz resonance at the muzzle
- Allow parameters to be modified via user input to extend the range of scenarios in which the code can be used to analyze launch dynamics

To better analyze the structural integrity of the prototype, an explicit-dynamic, finite-element simulation was desired to simulate the transient response and stress wave propagation of the mortar and individual UAV components. Such a simulation would utilize

ballistic code outputs as loading conditions. Solutions attained from the finite element simulation could then be compared to interior ballistics simulation results and public domain performance data for verification purposes.

1.4 – Importance of Research

The importance of this research resides on the applicability and usefulness of this type of analysis on the design and manufacturing of mortars in a variety of ways. It is desired that the results of this study will lead to an overall increase in the efficiency with respect to the design, fabrication, and economics of successful mortar prototypes. Furthermore, more extensive structural analyses can be performed on the proposed prototype design by means of an explicit dynamic solver such as LS-DYNA. The obtained results from the simplified ballistic code can be used as loading data for such explicit dynamic analyses, which can then be used as a pass/fail criterion for design considerations.

1.5 – General Approach

The base model of the ballistics code was used to analyze a 60mm mortar round, which consisted of the ignition cartridge and one solid propellant charge increment attached to the tail boom. Simulation of the interior ballistics of the system was performed via MATLAB. The following outline summarizes the general approach of the MATLAB simulation:

- Input user defined variables and archived pressure-time data
- Perform polynomial curve fit to archived pressure-time data
- Simulate solid propellant geometry during combustion from burn rate equations
- Solve the equations of motion of the projectile
- Evaluate the thermodynamics of the system

Figure 1.6 illustrates the general approach of the TL-UAV structural analysis by means of a flowchart. Individual processes of the flowchart are explained more thoroughly in Chapter 3, and are illustrated by means of flowcharts located in the appendix.

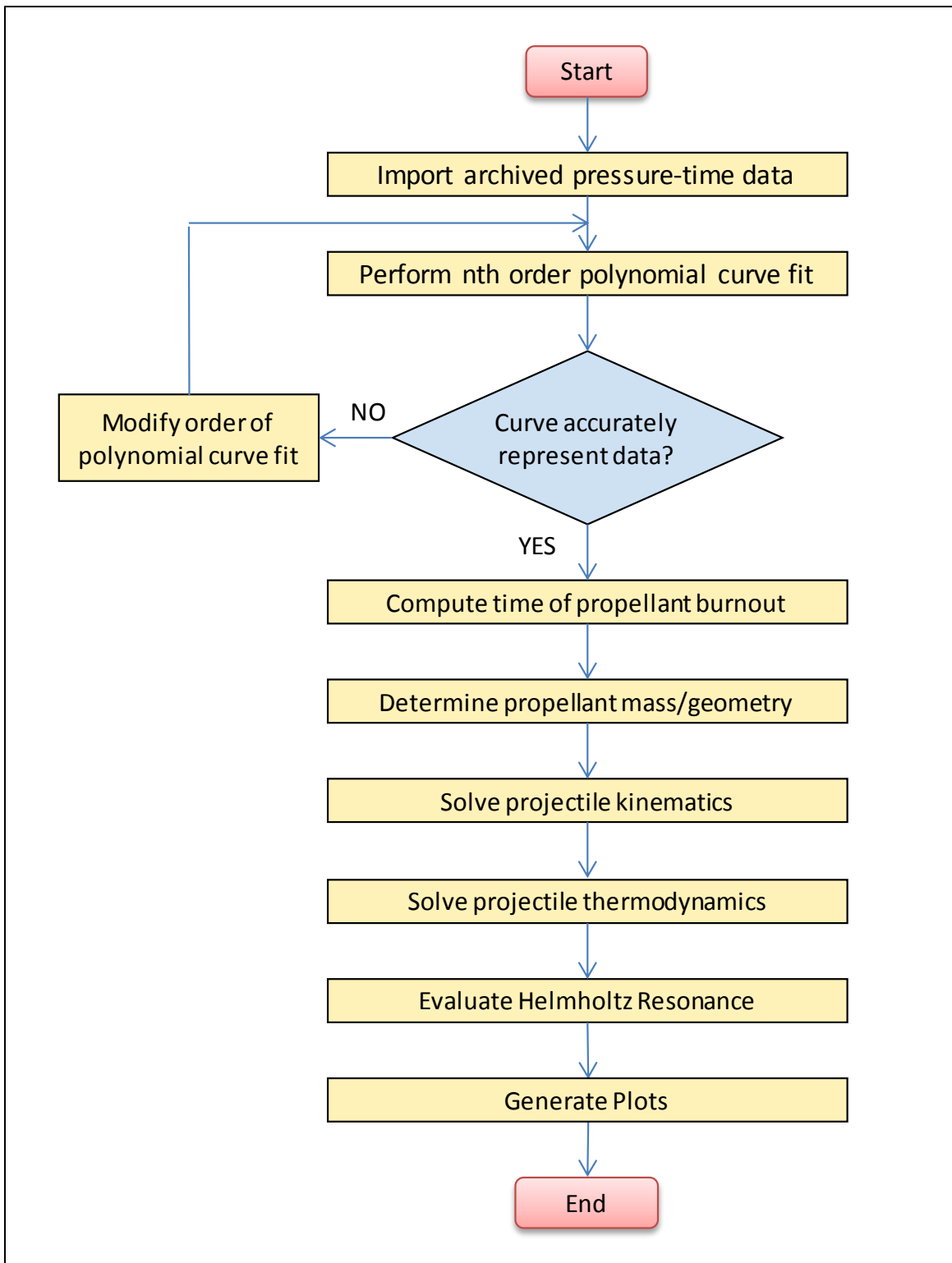


Figure 1.6 – Flowchart Outlining General Approach of Interior Ballistics Analysis⁵

5 - Character “n” used in flowchart denotes integer (n = 1, 2, 3, etc.)

1.6– Scope of Research

Interior ballistics simulation is of particular importance in the production of loading conditions necessary for an explicit-dynamic FE simulation. Typical interior ballistics simulations obtain pressure-time data by modeling the propellant combustion process. In this thesis, an effective approach to simulate interior ballistics without modeling the propellant combustion processes is presented. The simplified ballistics model proved to be an effective tool in regards to producing the desired loading conditions for FE simulations.

CHAPTER 2: BACKGROUND AND LITERATURE REVIEW

2.1 – Background

Internal structural integrity of projectiles is a critical aspect in a reliable design, but prediction of launch forces can be difficult to determine before experimentation. Ongoing design research conducted at West Virginia University has focused on tube launched folding-wing UAV designs based on mortar ballistics. Due to extreme accelerations during the launching phase, a typical mortar round experiences dramatic impulse loads for an extremely brief duration of time. Modern ballistic models solve for the projectile dynamics by modeling the actual combustion of the propellant. Mathematical formulations for such analysis require complex modeling to obtain accurate results. Therefore, it was desired to create a ballistic model capable of accurately evaluating the interior ballistics of a projectile by using experimentally recorded pressure-time data without simulating propellant combustion. Conceptually, a model created in this manner reduces the complexity of calculations to be performed, while maintaining a reasonable degree of accuracy for the motion dynamics results (displacement, velocity and acceleration of the projectile as a function of time).

2.2 – Mathematical Modeling of Interior Ballistics

The science of interior ballistics has been defined to evaluate: the ignition and burning processes of the propellant, the pressure changes in the barrel, the onset of projectile motion in the barrel, the friction between the projectile and the barrel, the in-bore projectile dynamics, and the barrel dynamics (wave propagation, thermal expansion, etc.) during launch. The physical foundation of interior ballistics can be built upon gas laws that describe how the chemical energy of a propellant is exchanged into the kinetic energy of a projectile (Carlucci & Jacobson, 2007). For this reason, computer models that have been developed to simulate the interior ballistics of a projectile initialize by evaluating the gas laws associated with propellant combustion.

Historical accounts reveal that solutions to interior ballistic problems have been attempted since the late 1700's when Lagrange tried to solve for the pressure distribution, gas density, and

gas velocity with respect to time after the combustion process initiated. The research performed since are split into two broad categories: semi-empirical solutions and exact theories. The purpose of semi-empirical solutions was to obtain a solution that accurately predicted the peak chamber pressure and muzzle velocity of experimental data. Alternatively, exact theories aim to accurately simulate the interior ballistic process from formulating a mathematical model of the multiphase fluid flow (Shelton, Bergles, & Saha, 1973).

Flow modeling has historically proven to be extremely challenging in applied mathematics and computational fluid dynamics (CFD). Two of the more significant approaches used to solve such problems are the Lagrangian approach and the Eulerian approach. The Lagrangian approach considered the fluid phase to be a continuum and “the particulate second phase” to be comprised of single particles. Trajectories of the particles in the fluid phase were predicted from forces interacting with the particles. The Eulerian approach considered the solid and gas phases as continuum and made use of conservation equations to obtain solutions (Acharya, 2009). Advancements in the field of interior ballistics have been made possible through development of equations describing mathematical models of fluid motion. Furthermore, implementation of high performance computers has enabled the creation of highly detailed mathematical computer codes capable of simulating the interior ballistic process (Nyberg, 2009). Such computer codes vary in detail and complexity.

In 1973, a technical report created at the Georgia Institute presented findings from a study that focused on heat transfer and gun barrel erosion. In the report, an interior ballistic model was outlined that simulated solid propellant combustion, gas dynamics, and projectile dynamics. According to the report, an interior ballistic analysis involved the creation of a propellant combustion model via the conservation equations, “the boundary layer momentum integral equation”, and the thermodynamic equation of state for the combustion gas. Initially, five basic unknowns were evaluated in the propellant combustion model; the volume fraction of the solid propellant, as well as the density, one dimensional velocity, pressure, and temperature of the combustion gas. A summary of the necessary equation corresponding to each unknown is provided in table 2.1 (Shelton, Bergles, & Saha, 1973):

Table 2.1 – Summary of Unknown Variables and Corresponding Equations

Unknown	Equation
Solid Volume Fraction	Solid Continuity
Gas Density	Gas Continuity
One-Dimensional Gas Velocity	Conservation of Momentum
Gas Pressure	Conservation Of Energy
Gas Temperature	Thermal Equation of State

Performing such an analysis required the conservation equations to be written in finite difference form, and implementation of numerical techniques to solve the coupled non-linear partial differential equations, since an analytical solution did not exist (Shelton, Bergles, & Saha, 1973). Encoding of the numerical techniques so that they could be solved computationally demanded complex modeling to obtain realistic results.

Once solutions had been obtained from the propellant combustion model, the dynamics of the projectile could be evaluated. This phase of the interior ballistic analysis entailed use of the equation of motion to determine the displacement, velocity, and acceleration of the projectile as a function of the computed combustion gas pressure. Unfortunately, the technique described was not applicable to the boundary points of the system, which in turn further complicated the analysis (Shelton, Bergles, & Saha, 1973).

2.2.1 – XNOVAKTC Code and Extensions

The XNOVAKTC interior ballistic computer code was introduced in 1990. The code was based on the Lagrangian approach to model the interior ballistic cycle based on numerical solutions for one-dimensional, multi-phased flow governing equations (Gough, 1990). Additional versions of the code enabled modeling of two and three-dimensional compressible flows, which ultimately led to the next generation interior ballistic code NGEN3 (Acharya, 2009). NGEN3 enabled flame-spreading and combustion modeling of direct and indirect gun propulsion systems using a coupled Eulerian/Lagrangian approach, and effectively predicted the pressure load on the projectile as a result of propellant combustion (Ray, Newill, Nusca, & Horst, 2004).

One NGEN3 source reviewed discussed the connection between the interior ballistic code and structural analyses. According to the source, the mechanical response experienced by the projectile as a result of the pressure load was predicted via finite-element solid mechanics codes, one of which being DYNA3D (Ray, Newill, Nusca, & Horst, 2004). Review of this particular source reinforced the idea behind using pressure data retrieved from interior ballistic analyses as loading conditions in advanced explicit-dynamic finite-element simulations.

2.2.2 – IBHVG2 Code and Extensions

Another popular interior ballistic model is the IBHVG2 (Interior Ballistics High Velocity Guns version 2) code, which is a lumped parameter interior ballistic code developed by the Army Research Laboratory (ARL) as an updated version of a classic ballistic code (Schmidt, Nusca, & Horst, 2009). Lumped parameters are essentially discrete entities that are capable of exchanging energy with other entities (Voldman, 2007). The original use of the code was to evaluate “the interior ballistic cycle of the standard 5-Inch Propelling Charge Mk 67” (Erline & Fischer, 1996). The code operated by simulating the combustion of the ignition and propelling charges while computing several variables with respect to time including; average gas pressure and temperature, projectile acceleration and velocity, and the mass fraction of unburned propellant (Anderson & Fickie, 1987). However, the IBHVG2 code initially proved to be ineffective with respect to mortar simulations as a result of the complex nature of the combustion process associated with launching (Schmidt, Nusca, & Horst, 2009).

Mortar combustion modeling requires evaluation of two chambers that contain propellant, which differs from single chamber combustion modeling associated with conventional gun configurations. Advancement made to the IBHVG2 code enabled two chamber modeling through a permeable canister mode to better represent mortar configurations (Schmidt, Nusca, & Horst, 2009). The permeable canister mode takes into consideration that ignition charges for mortars are contained in closed canisters located inside of the tail boom. Propellant combustion is initiated by burning the ignition charge in the closed canister. As a result, the pressure in the small volume canister increases rapidly. At high pressures, the combustion products in the canister escape into the larger chamber through burst vents, resulting in the

ignition of the external propellant charge, rapid expansion of combustion product gases, and acceleration of the projectile (Anderson, 2006).

One controversial aspect of the IBHVG2 code is that it simulates the propellant combustion uniformly. Experiments have shown that the flame propagation is “neither complete nor instantaneous immediately after ignition” (Anderson & Fickie, 1987). As a result, the actual pressure is observed to initially lag behind the simulation results, as demonstrated in figure 2.1 (Anderson & Fickie, 1987):

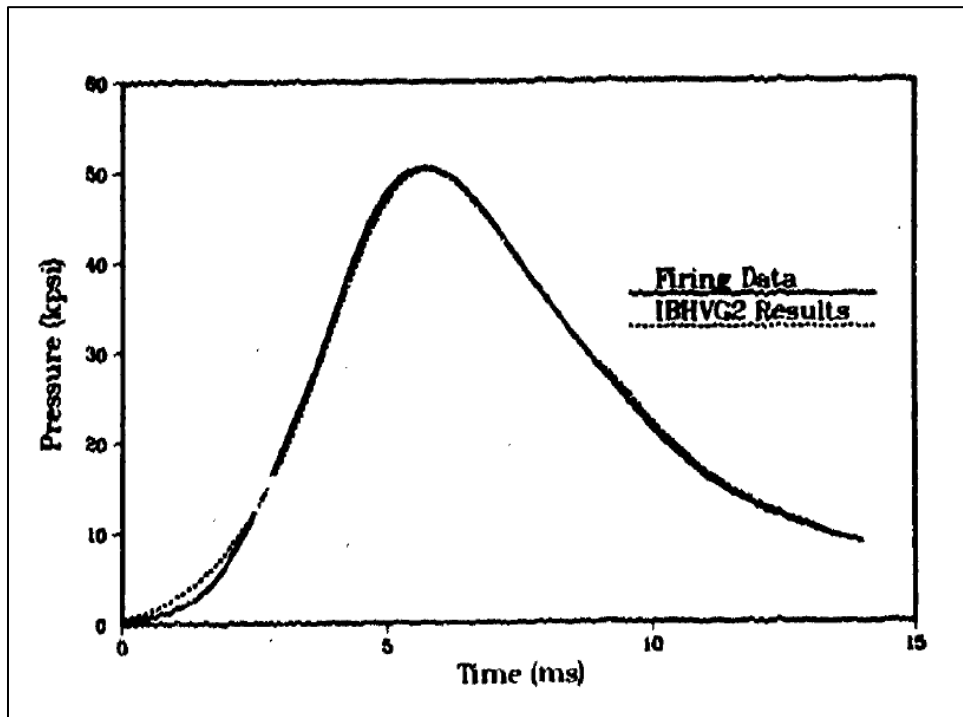


Figure 2.1 – Illustration of Initial Lag in Pressure Observed in Comparison to IBHVG2 Solution

2.2.3 – 3D-MIB Code and Extensions

The most realistic interior ballistic model reviewed was created in a joint effort between The Pennsylvania State University and the Army Research and Development Engineering Center. This model, named 3D-MIB, was created to realistically simulate the three-dimensional interior ballistics of a 120-mm mortar system. Interior ballistic process simulations in this model were performed in a particular order. First, an empirical flash tube sub-model was used to determine the instantaneous energy and mass flux of the two-phase combustion products from the flash tube. Next, a mobile granular bed combustion sub-model was executed using the two-

phase conservation equations, the equation of state for gas inside of the tail boom, propellant burn rate and surface temperature equations, and a relationship pertaining to the intragranular stress. Next, another sub model evaluated the flame propagation and two-phase properties inside the tail boom section, which included gas pressure, temperature, density, and velocity, as well as propellant burn rate and surface temperature. A fourth sub-model was then executed that examined the 3-D combustion of the propellant in the free volume between the projectile and launcher, and the pressure distribution behind the projectile. After considering these sub-models, a final sub-model was used to predict the dynamics of the projectile. The solutions attained from the sub models were then integrated into a single model, which was then compared to experimental data for validation purposes. Years of research were put into the construction of the 3D-MIB model, which ultimately could only be used to simulate the interior ballistics of a 120-mm mortar system. However, work has been done to extend the application of the model to other mortar systems (Acharya, 2009).

Extensive review of literature failed to provide a simplified interior ballistics model capable of accurately simulating projectile dynamics from archived pressure-time data. Instead, a variety of models were discovered that simulated the combustion process of solid propellants to obtain pressure-time data. Further review revealed that a majority of existing interior ballistic models were not capable of evaluating the complex combustion process that occurs during the act of launching a mortar. Observations made from an efficiency point of view motivated the desire to create a simplified model that could effectively be used for the structural analysis and design considerations of a tube-launched UAV prototype being developed at West Virginia University. The next chapter discusses the procedure used in the creation of the simplified interior ballistics model.

CHAPTER 3: METHODOLOGY

3.1 – Simplified Interior Ballistics Code

Creation of a ballistic code capable of solving motion dynamics and thermodynamics of the prototype was an important aspect of the structural analysis. As previously mentioned, a desired objective of the code was to maintain simplicity without compromising solution effectiveness. To simplify the interior ballistics code, some assumptions had to be made. The first assumption was that the burning process of the propellant was to be neglected by implementing archived pressure-time data. The second assumption was that the dynamic pressure at the nose of the round and frictional effects acting on the projectile were to be neglected. Incorporating an effective projectile mass enabled friction and resistive pressure effects to be accounted for by an addition to the mass of the projectile. Literature survey revealed the effective mass to be approximately equal to the projectile mass multiplied by 1.02 (U.S. Army Materiel Command, 1964). Finally, it was assumed that accurate results could be attained by creating the model one-dimensionally. The steps taken in the creation of such a code are included in the following subsections.

3.1.1– Data Retrieval

Microsoft Excel was used to store the archived pressure-time data for the projectile. The data supplied occupied over a thousand cells of the spreadsheet per variable. For this reason, the MATLAB command “*xlsread*” was implemented in the ballistic code to automatically retrieve the data from the spreadsheet file rather than manually inserting the data directly into the code. The original archived data supplied for the projectile consisted of a column containing the time in milliseconds, and the recorded gage pressure in pounds per square inch (psi). To ensure that data was imported properly to MATLAB, unit conversions of the data was performed in the Excel spreadsheet. The modified spreadsheet included two columns containing time steps in seconds and milliseconds, and four columns containing pressure data; gage pressure and absolute pressure in psi, and gage pressure and absolute pressure in Pascals (Pa). A sample of the created spreadsheet is included in table 3.1:

Table 3.1 – Excel Spreadsheet Example⁶

<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<u>time</u> <u>(ms)</u>	<u>time</u> <u>(sec)</u>	<u>Gage Pressure</u> <u>(psi)</u>	<u>Absolute Pressure</u> <u>(psi)</u>	<u>Gage Pressure</u> <u>(Pa)</u>	<u>Absolute Pressure</u> <u>(Pa)</u>
0	0	0	14.7	0	101325

To ensure proper function of the ballistic code, it is encouraged that spreadsheets created for importation purposes follow the format of table 3.1. It is also necessary to note that the spreadsheet file must be saved in the same folder as the ballistic code (Read Microsoft Excel Spreadsheet File).

Input of the *xlsread* command required certain identification properties of the destination Excel file. The properties of interest were the file name, the sheet name, and the column identification letter associated with variable of interest. For example, the time data required for the code was saved under the Excel file named “*P-t data.xlsx*” in a sheet named “*Data*”. The column identification letter associated with the designated time data, as seen in table 3.1, was “*B*”. Use of the *xlsread* command in the MATLAB code can be observed in the appendices (Read Microsoft Excel Spreadsheet File).

3.1.2– Data Curve Fitting

A negative quality observed in the experimentally recorded data was a distinct oscillating behavior, as represented in figure 3.1. The behavior of the plot can be attributed to interactions between reflecting and refracting shock and rarefaction wave fronts inside of the cannon barrel (Williams, Brandt, Kaste, & Colburn, 2006). After solid propellant charges are ignited, the combustion gases expand rapidly, causing regions of extreme pressure gradients which lead to the formation of shock waves. Within milliseconds, the waves move and bounce off of solid boundaries within the control volume resulting in reflected shock and rarefaction waves. As these waves reflect and refract throughout the system, dozens of thermodynamically discreet zones are created. The pressure sensor used for recording data is affected, which results in the many peaks and valleys as seen in figure 3.1 (Carlucci & Jacobson, 2007). For the purposes of the ballistic code, an equation of pressure as a function of time was required to perform the

⁶ The first row shaded gray represents the column identification letter from Microsoft Excel.

interior ballistic calculations. The equation was attained by performing a curve fit to the experimental data. Several curve fitting options were available via MATLAB; however, to maintain the simplicity of the ballistics code, a polynomial curve fit was selected. Evaluation of the data through the MATLAB curve fitting tool, *cftool*, revealed that the best curve fit and most desirable R-square value was obtained from a 9th order polynomial curve fit. Note that lower order polynomial curve fits may be more appropriate for different scenarios. Archived data can be evaluated by use of the *cftool* command to summon the MATLAB Curve Fitting Toolbox (Open Curve Fitting Tool).

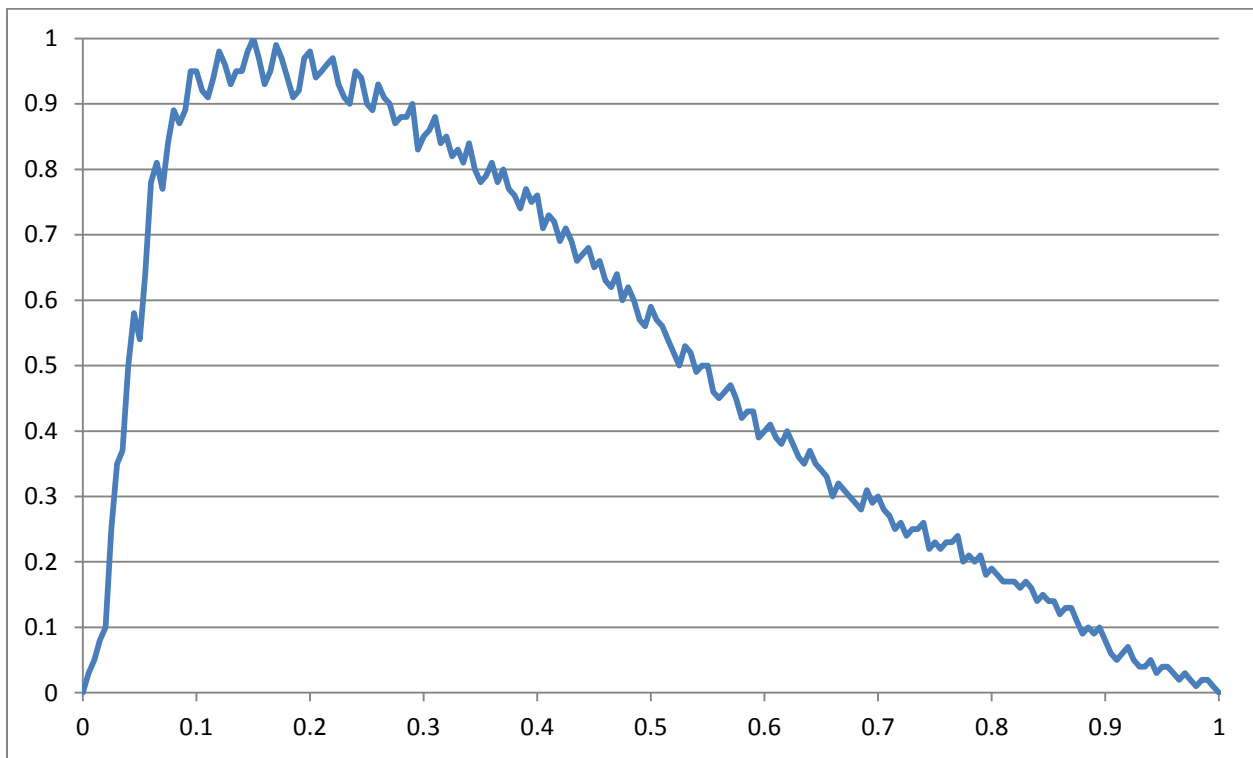


Figure 3.1 - Arbitrary Plot of Pressure-Time Data

Performing an nth order polynomial curve fit in MATLAB was accomplished via the *polyfit* command (Polynomial Curve Fitting). The 9th order curve fit used in the MATLAB code resulted in an equation of the form:

$$P_{cf} = p_1t^9 + p_2t^8 + p_3t^7 + p_4t^6 + p_5t^5 + p_6t^4 + p_7t^3 + p_8t^2 + p_9t + p_{10} \quad \mathbf{3.1}$$

The symbol P_{cf} in equation 3.1 denoted the curve fit pressure at time t , while the symbols p_j ($j=1, 2, 3$, etc.) represented the coefficients obtained from the curve fit. The importance of this

equation was to enable integration and differentiation to be performed. However, the pressure equation could not be integrated by MATLAB in the form output from the polynomial curve fit since it was not a function of symbolic time.

Resolving the issue pertaining to integration was combated by use of the MATLAB Symbolic Toolbox (Symbolic Math Toolbox). Using this toolbox, a symbolic time variable, t , was created via the *sym* MATLAB command (Creating Symbolic Objects). The symbolic time variable was then substituted into equation 3.1 resulting in an equation of pressure as a function of symbolic time. At this point, integration could be performed to the symbolic pressure polynomial via the *int* MATLAB command (Symbolic Integration). Once integration had been performed to the polynomial, the symbolic pressure expression could be reverted back to an expression of numerical time via the *sym2poly* MATLAB command (Symbolic-to-Numeric Polynomial Conversion). The *sym2poly* command was used to retrieve the coefficients associated with the symbolically integrated function, which were then used in the numerical calculations of the equations of motion. However, the next procedure undertaken was to evaluate the burn rate of the propellant using the approximated pressure expression.

3.1.3– Propellant Burn Rate

Prior to solving the burn rate computations, a few assumptions had to be made. First, the dimensions of the actual propelling charge to be used on the projectile could not be obtained due to export control regulations. Therefore, the charge was assumed to be a torus which encompassed the area between the tail-boom of the projectile and the inner wall of the cannon. Furthermore, the ignition cartridge of the propellant was neglected from the model. The initial volume of the charge was approximated to be the volume of a torus with the same geometry. Next, the initial mass was calculated by dividing the initial volume by the density of the solid propellant. The density of the charge was retrieved from a public domain source. With these assumptions in mind, the burn rate equations were then evaluated.

A variety of burn rate equations can be used to describe how a solid propellant oxidizes. The most commonly used equation for the purpose of computer codes was discovered via research (Carlucci & Jacobson, 2007):

$$\frac{dr_{ch}}{dt} = -\beta P_{cf}^n \quad 3.2$$

The left hand side of equation 3.2 represents the burn rate of the propellant, which can be defined as the “time rate of change of the web” (Carlucci & Jacobson, 2007). Note that the generic unit for the propellant burn rate was defined as length/time. The right hand side of the equation contains burn rate constants, β and n , which represented the burn rate coefficient and burn rate exponent respectively, and the curve fit pressure, P_{cf} . The burn rate equation was used to determine the rate of regression of the surfaces of the solid propellant. The data attained from the burn rate equation was then used to approximate the remaining volume of the solid propellant and the mass of solid propellant that had been combusted, which was an important factor in thermodynamic analyses. Computations were inserted into the ballistics code using burn rate properties discovered in public domain documents.

The first step was to determine the time step in which all of the propellant had completely combusted, or burned out. This was accomplished by inserting the numerical values of the curve fit pressure into equation 3.2. The attained values were then multiplied by the time vector using the MATLAB array multiplication operator ‘.*’ (Matrix and Array Arithmetic). Performing this computation resulted in the regression of the propellant web at each time step interval. Using the MATLAB *cumsum* command, the cumulative sum of interval regressions was calculated, which computed the total regression at each time step (Cumulative Sum). Next, the MATLAB command *ones* was used to create an equally sized vector as the cumulative sum vector with all values set equal to 1 (Create array of all ones). This array was then multiplied by the initial charge radius. The cumulative sum array was then subtracted from the array containing values of the initial charge radius, which effectively computed the remaining radius of the charge at each individual time step. Finally, the MATLAB *find* command was utilized to determine the index number in which the charge radius from the previous calculation became negative. The index number computed could then be used to identify the time in which the propellant had exhausted. This computational procedure is summarized by the flowchart illustrated by figure C1 in the appendix.

The next phase in the burn rate computations was to compute the charge volume and mass at each time step. The volume of the charge was approximated by calculating an equivalent length of the charge. This length was determined by use of equation 3.3:

$$L_{eqv} = \frac{V_{ch,i}}{\pi r_{ch,i}^2} \quad 3.3$$

Where L_{eqv} is the equivalent length of the charge, $V_{ch,i}$ is the initial volume of the charge, and $r_{ch,i}$ is the initial radius of the charge. The volume of the charge, V_{ch} , at each time step was then calculated from:

$$V_{ch} = L_{eqv} \pi r_{ch}^2 \quad 3.4$$

Once the volume of the charge had been calculated, the mass of the charge remaining was easily computed by multiplying the volume by the solid propellant density:

$$m_{ch} = \rho_{ch} V_{ch} \quad 3.5$$

Where m_{ch} represents the charge mass and ρ_{ch} is the solid propellant density. Subsequently, it was desired to solve for the mass of propellant that had been burned, denoted by the symbol N , which was formulated by subtracting the charge mass remaining from the initial charge mass:

$$N = m_{ch,i} - m_{ch} \quad 3.6$$

Finally, the mass fraction, the percentage of charge mass that has oxidized, was computed by dividing the mass of burnt propellant by the initial charge mass:

$$mass\ fraction = \frac{N}{m_{ch,i}} \quad 3.7$$

Figure C2 of the appendix summarizes the propellant mass/geometry computational procedure. Upon completing the propellant burn rate calculations, the next step in creating the ballistic code was to derive and compute the equations of motion.

3.1.4– Equations of Motion

Derivation of the equations of motion required the equations to hold true to Newton's second law (Carlucci & Jacobson, 2007). For a closed system, Newton's second law can be

expressed as force (F) equals mass (m) multiplied by acceleration (a) as illustrated in equation 3.8 (Carlucci & Jacobson, 2007):

$$F = ma \quad 3.8$$

Another useful relationship used in the derivation was the definition of pressure (P) (Moran & Shapiro, 2008):

$$P = \frac{F}{A} \quad 3.9$$

The symbol “ A ” used in equation 3.9 denotes the cross sectional area of the cannon barrel. Rearranging equation 3.9 yielded an alternate representation of force which was then substituted into equation 3.8. By once again rearranging the equation, a relationship was formed between the pressure and acceleration of a tube launched projectile (U.S. Army Materiel Command, 1964):

$$a_{prj} = P_{cf} \left(\frac{A_{cb}}{m_{prch}} \right) \quad 3.10$$

The subscripts prj , cf , cb , and $prch$ presented in equation 3.10 denoted references to the projectile, curve fit expression, cannon barrel, and combined projectile and propellant charge respectively.

The projectile velocity was derived next from the mathematical relationship between acceleration and velocity (Hibbeler, 2007):

$$\frac{dV}{dt} = a \quad 3.11$$

By substituting equation 3.11 into equation 3.10 and integrating both sides with respect to time, the velocity of the projectile was determined to be:

$$V_{prj} = \left(\frac{A_{cb}}{m_{prch}} \right) \int_0^{t_{exit}} P_{cf} dt \quad 3.12$$

Note that the variable A_{cb} was considered to be constant and therefore could be pulled out of the integrand. Additionally, the combined mass of the projectile and charge was assumed to be a constant despite the minor changes in mass resulting from propellant combustion.

Next, the projectile displacement was derived using the mathematical relationship between velocity and displacement (Hibbeler, 2007):

$$\frac{dx}{dt} = V \quad \mathbf{3.13}$$

Following the same procedure used in the velocity derivation, the displacement of the projectile was calculated to be:

$$x_{prj} = \int_0^{t_{exit}} V_{prj} dt = \left(\frac{A_{cb}}{m_{prj}} \right) \iint_0^{t_{exit}} P_{cf} dt dt \quad \mathbf{3.14}$$

Encoding the expressions above into MATLAB required only a few lines of code. The expression used to solve the projectile acceleration was the simplest to encode since the cross sectional area and mass were known constants, and an expression describing the pressure had been obtained from the curve fit. Accelerations at each time step were solved by use of the MATLAB array multiplication operator, as previously used in the burn rate computations (Matrix and Array Arithmetic). Simply stated, the period placed before the multiplication symbol indicates that multiplication is desired to be performed at each time step. This operator was extensively used within the MATLAB routine. Solutions were then attained for the projectile velocity and displacement using the *int* command to symbolically integrate the pressure expression, the *sym2poly* command to change the integrated expression from a symbolic representation to a numerical representation, and the MATLAB array multiplication operator to record solutions for each time step. The flowchart illustrated in figure C3 in the appendix summarizes the projectile kinematics computational procedure. Once the motion dynamics of the system had been computed, the thermodynamics of the system were then evaluated.

3.1.5– Thermodynamics

Thermodynamic equations of state can be typically used to relate the pressure, volume (V), and temperature (T) within a system. Initially, the gas pressure (P) was the only of the three mentioned variables that had been accounted for, through the archived pressure data curve fit. The gas volume was the next variable to be evaluated. However, certain assumptions had to be made pertaining to the system:

1. The initial chamber volume was defined as the free volume behind the projectile prior to propellant ignition.
2. The gas created at the onset of propellant combustion instantaneously occupied the initial chamber volume.
3. The gas was assumed to be evenly distributed within the free volume behind the projectile.

Using these assumptions, the gas volume was obtained by multiplying the displacement of the projectile by the cross sectional area of the cannon barrel, which was then added to the initial chamber volume:

$$V_{gas} = V_{cham,i} + A_{cb}x_{prj} \quad \mathbf{3.15}$$

Implementation of equation 3.15 enabled the ballistic code to solve for the gas volume at each time step. The next phase was to solve for the temperature of the combustion gas.

At first, solving for the gas temperature in the system was attempted through the use of an equation of state, such as the ideal gas or Nobel-Abel equations. However, the solutions obtained from this approach resulted in unrealistically high initial gas temperature that rapidly transitioned to unreasonably low gas temperatures. The reason behind the erroneous temperatures was discovered to be the result of poorly defined variables necessary for the computation, including the gas co-volume and specific volume (Johnston, The Noble-Abel Equation of State: Thermodynamic Derivations for Ballistics Modelling, 2005). Therefore, a different approach to the problem was explored. Research revealed that the gas temperature could be equated through an energy balance equation. This equation stated that the projectile

kinetic energy was equal to “the loss of internal energy of the gas”, which is summarized in the following equation (U.S. Army Materiel Command, 1964):

$$N \int_T^{T_f} c_v dT = \frac{1}{2} m_{prj} V_{prj}^2 \quad 3.16$$

In which N denoted the mass of propellant burned, c_v represented the specific heat under constant volume, T was the temperature, T_f denoted the adiabatic flame temperature, m_{prj} was the mass of the projectile, and V_{prj} represented the velocity of the projectile. To further simplify this equation, the specific heat quantity, c_v , was assumed to be constant (U.S. Army Materiel Command, 1964):

$$N c_v (T_f - T_{gas}) = \frac{1}{2} m_{prj} V_{prj}^2 \quad 3.17$$

The absolute temperature in the system was then computed by rearranging equation 3.17:

$$T_{gas} = T_f - \frac{m_{prj} V_{prj}^2}{2N c_v} \quad 3.18$$

The temperatures output from this computation provided a more realistic representation of the temperature in the system. The same procedure involving the MATLAB array multiplication operator was used to implement the temperature expression into the ballistic code, as performed with other computations (Matrix and Array Arithmetic). The thermodynamics computational procedure is outlined in figure C4 of the appendix by means of a flowchart.

Another variable of interest was the local speed of sound. This variable was needed for computing the Helmholtz resonance of the projectile. It should be noted that the Helmholtz resonance has been included due to speculation that it may be associated with premature projectile fragmentation at the cannon muzzle; however, this topic will not be included in the scope of this thesis.

Research revealed that the local speed of sound was defined as (Carlucci & Jacobson, 2007);

$$c = \sqrt{\gamma R_{gas} T_{gas}} \quad 3.19$$

Where c represented the local speed of sound, γ was the specific heat ratio, R_{gas} denoted the specific gas constant of the combustion gas, and T_{gas} was the gas temperature. Since all of the necessary variables were available from public domain literature, approximated, or solved for, the local speed of sound was easily computed. Results attained were then applied to the Helmholtz resonance equation (Browning, Transient Flow Characteristics of a High Speed Rotary Valve, 2009):

$$f_H = \frac{c}{2\pi L_{cb}} \quad \mathbf{3.20}$$

The symbol f_H was used for the Helmholtz Resonance, and L_{cb} denoted the depth of the cannon bore. These expressions were inserted into the ballistic code in the same manner as previous expressions, while making use of the MATLAB command *sqrt* to account for the square root in equation 3.19 (Square Root). The Helmholtz Resonance computational procedure is outlined by means of a flowchart in figure C5 of the appendix.

3.1.6 – LS-DYNA Loading Conditions

The solutions determined by the MATLAB routines can be used as loading conditions of an explicit, three-dimensional simulation via ANSYS LS-DYNA. The main loading condition revealed by the MATLAB routines would enable the inclusion of a pressure-time curve into ANSYS. Such a curve can be incorporated into ANSYS by a variety of methods, this simplest of which involves the creation of a function.

In ANSYS, functions can be created by selecting parameters from the toolbar, then selecting functions, and clicking define/edit. Following this procedure will open the function editor. At this instance, the polynomial curve fit coefficients output from the MATLAB routines are needed. Creation of a pressure-time curve in ANSYS can now be performed by creating a function as illustrated in equation 3.1 in the input dialog box labeled '*Result =*'. The curve coefficients are implemented into the dialog box, and multiplied by the corresponding '*Time*' variable, resulting in a 9th order polynomial. Figure 3.2 summarizes the outlined procedure, note that the variables '*P1*' and '*P2*' represent where the values for the polynomial curve fit coefficients would be included:

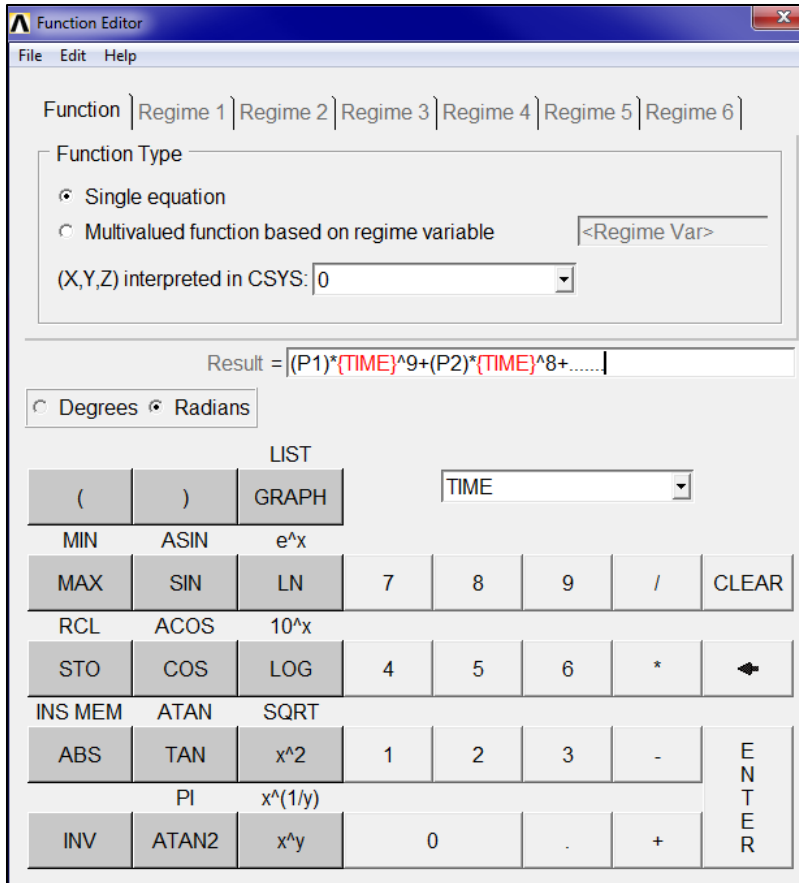


Figure 3.2 – ANSYS Function Editor Screenshot

CHAPTER 4: RESULTS and DISCUSSION

4.1 – Data Curve Fitting

MATLAB successfully produced a 9th order polynomial function in the form of equation 3.1 that correlated well to the archived pressure-time data provided by ARDEC. The strong relationship between the curve fit and archived data can be observed by plotting the curve fit and archived data sets on the same plot. This plot is illustrated in figure 4.1. The fitted curve represents the bulk fluid behavior with a reasonable degree of accuracy. Further validation of the curve fitting process can be achieved through comparison of the equation of motion results with experimentally recorded performance data, which will be discussed further.

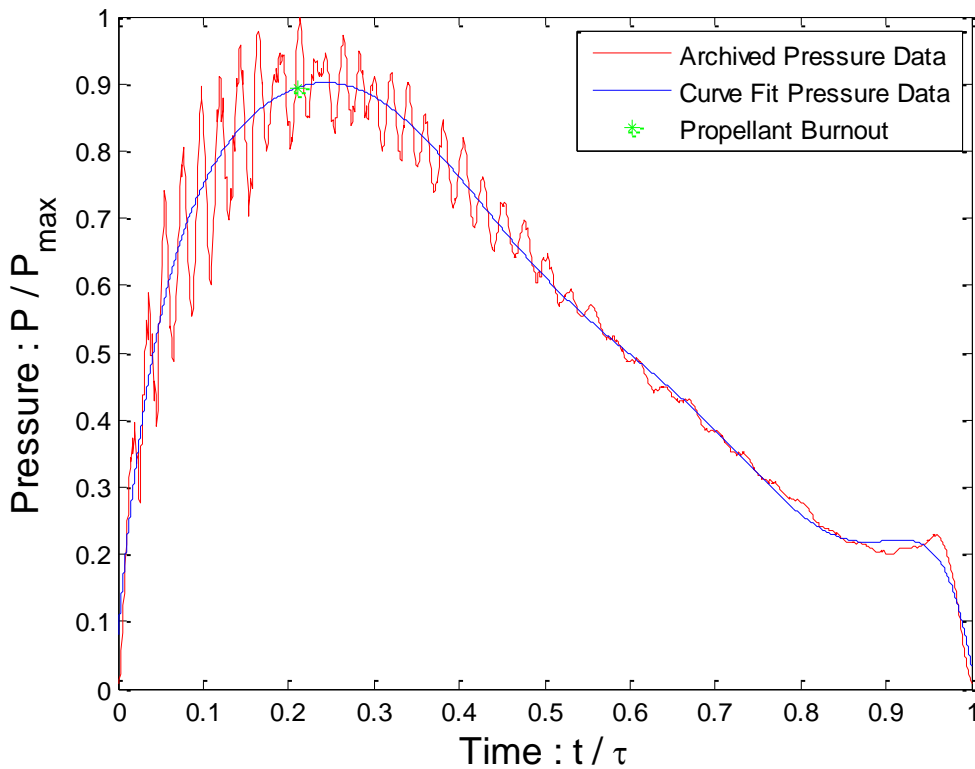


Figure 4.1 –Curve Fit and Archived Pressure Data Comparison (Non-Dimensional)

One additional note that must be addressed pertains to the precision of the polynomial curve fit coefficients. Experiments conducted on how rounding the coefficients impacted the accuracy of the polynomial curve fit data revealed significant deformation of the curves as the

number of significant digits was reduced. However, the degree of deformation varied considerably between the Imperial and SI data sets. For instance, if the polynomial curve fit coefficients were rounded to 3 decimal places in scientific notation, the corresponding curve fit for the Imperial units would exhibit minor deformation, as illustrated in figure 4.2:

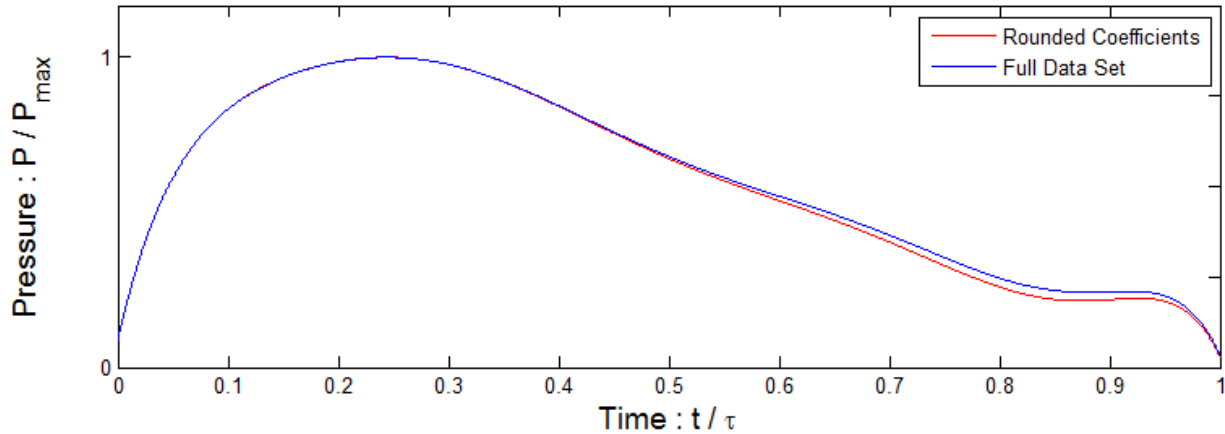


Figure 4.2 – Deformation of Imperial Curve Fit Plot Due to Rounding (Non-Dimensional)

Observation of the curve revealed that a majority of the deformation occurred at the latter half of the data set. Further observation of the data revealed a maximum difference between the two plots which corresponded to an 11.06% difference. Conversely, evaluation of the SI polynomial curve fit data revealed more sizeable deformations between the data sets. This relationship is exhibited in figure 4.3:

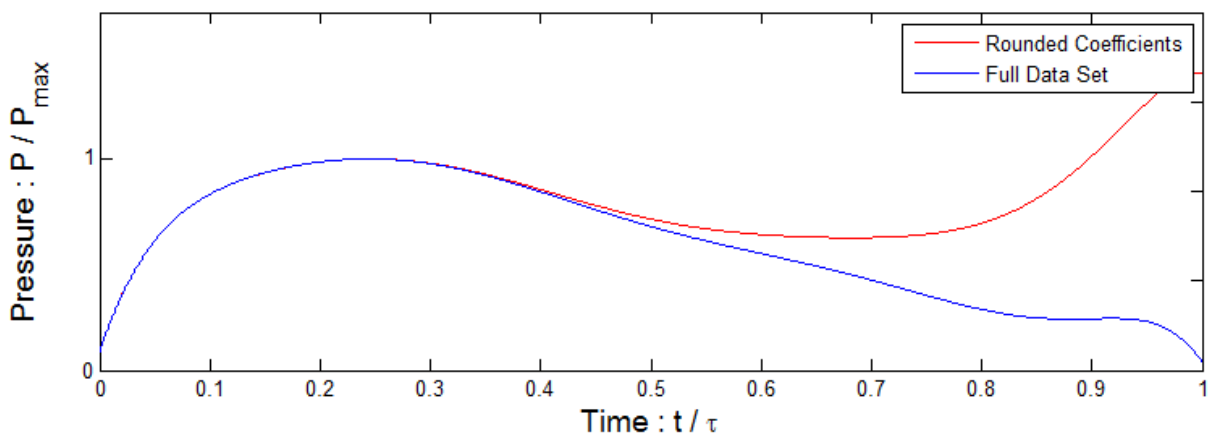


Figure 4.3 – Deformation of SI Curve Fit Plot Due to Rounding (Non-Dimensional)

The relationship demonstrated in figure 4.3 is clearly not representative of the actual data set. Further examination of the data sets revealed a maximum deformation between the two plots which corresponded to a 3622% difference.

With these findings in mind, the importance of the precision of the polynomial curve fit coefficients was clearly demonstrated. Since these coefficients are to be used in ANSYS for the creation of a pressure-time curve representative of archived data, it is essential that the precision of the coefficients be upheld, especially if the simulation is to be performed in SI units. As demonstrated, rounding errors prove to significantly alter the output of the polynomial function, which will ultimately result in non-representative loading conditions and unusable simulation results.

4.2– Propellant Burn Rate

Verification of the propellant burn rate results were difficult to provide. The reason behind this was due to the lack of quality sources pertaining to the propellant used in the system. A majority of the variables used in the burn rate calculations were either assumed or obtained from public domain literature, which was assumed to be a good approximation to the actual propellant data. Fortunately, the solutions obtained from the propellant burn rate section had very little impact on the results attained from the equation of motion and thermodynamic solutions. However, to conduct an accurate energy balance of the system, the propellant specifications used in the system would have to be representative of the true physical nature of the propellant to be used. Figure 4.4 was created to illustrate the change in the propellant charge's mass during the combustion process:

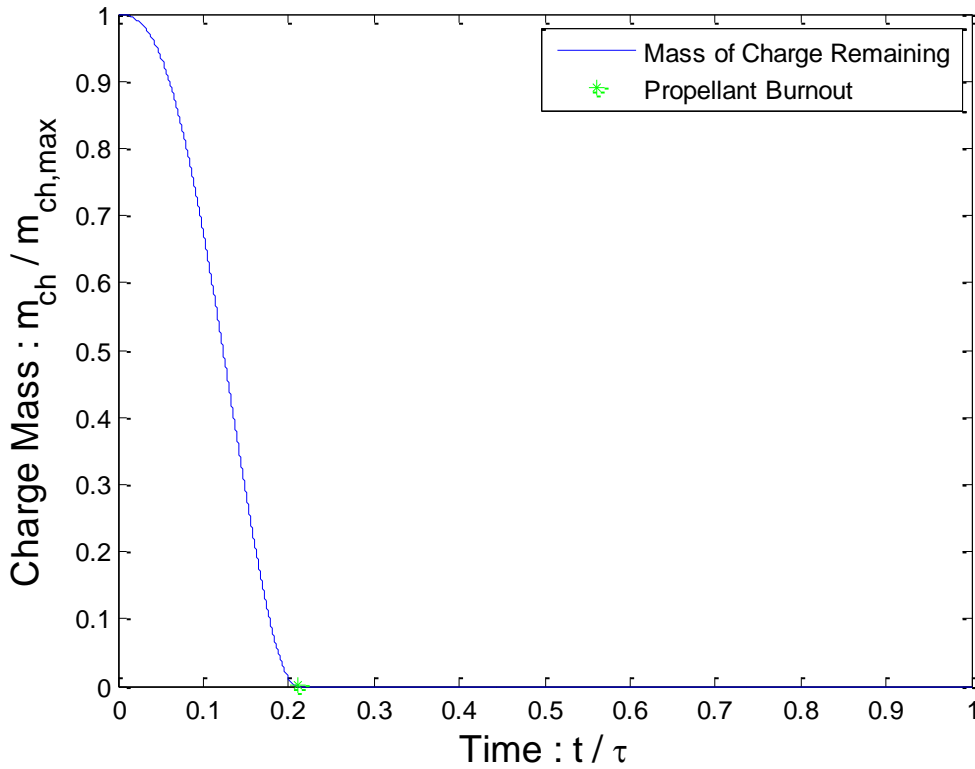


Figure 4.4 – Predicted Propellant Mass (Non-Dimensional)

Validation of the computational procedure used to create figure 4.4 was attained through comparison of the shape and trends of the curve to data reviewed in literature, such as figure 4.5 (Heiser, 1991):

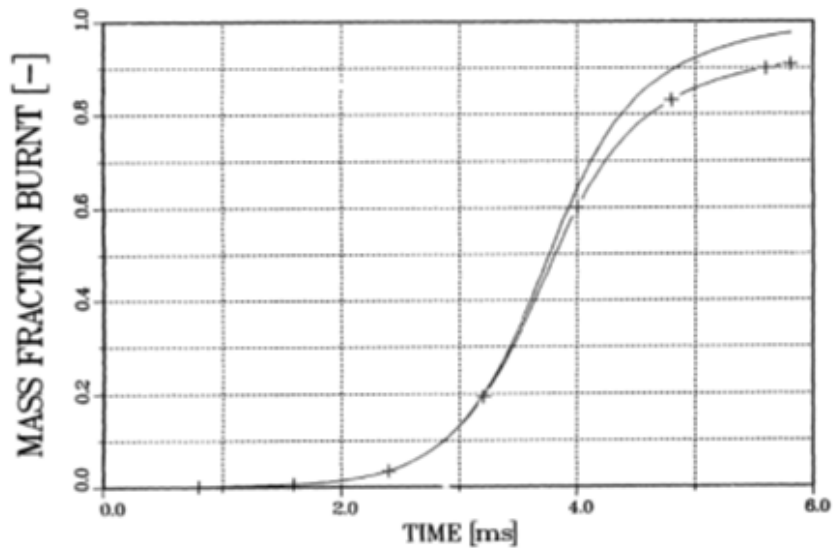


Figure 4.5 – Hypothetical Mass Fraction (Heiser, 1991)

The system represented by figure 4.5 consisted of a 120 gram projectile being fired from a cannon with a bore diameter of 20 mm and length of 2 m. The mass fraction burnt denoted in figure 4.5 represents the percentage of propellant that has combusted. Conversion of the propellant charge mass used in figure 4.4 to mass fraction burnt was achieved by means of the equation 3.7. The resulting plot is illustrated in figure 4.6.

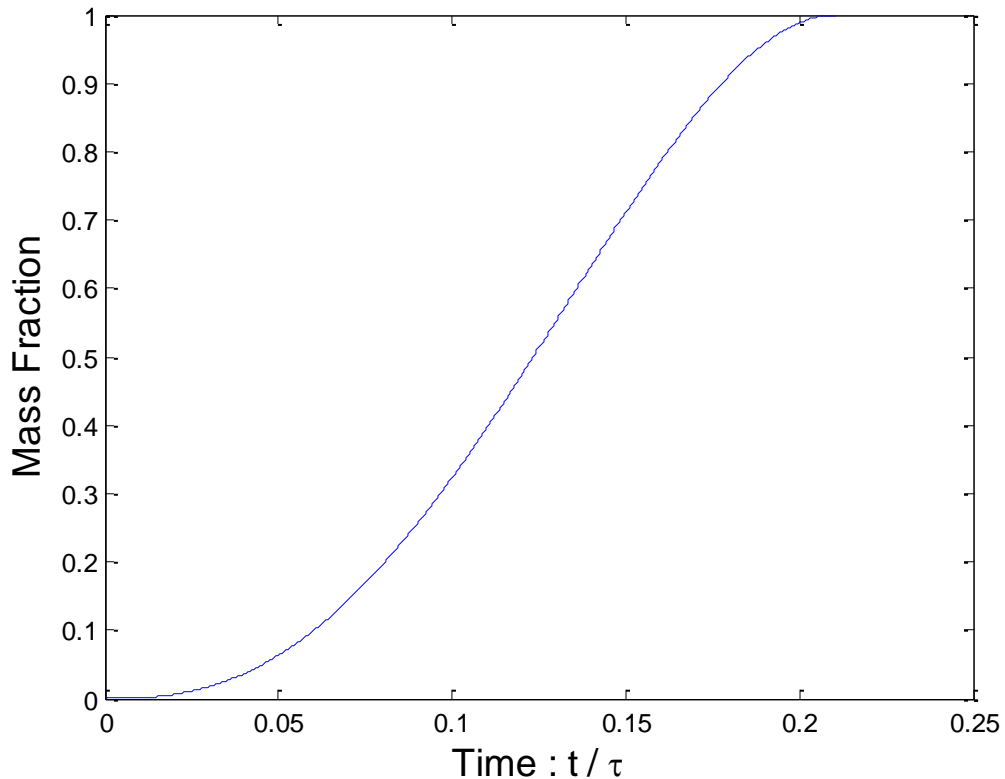


Figure 4.6 – Mass Fraction (Non-Dimensional)

Note that the shape of the curves, not the actual data points, was the focal point in this comparison. With this in mind, the shape of the mass fraction curves attained from the MATLAB codes compared extremely well to the trends observed in figure 4.5. This was explained by the relationship between propellant burn rate and pressure, as noted in equation 3.2. Because the shape and trends of the plot appear to be similar to the hypothetical scenario, it was assumed that the MATLAB code accurately simulated the propellant combustion process. It must be noted that the actual solution may vary from those computed by the MATLAB routine, but can be easily updated by inserting more accurate burn rate coefficients into the program.

Closer examination of the Imperial and SI unit results of the mass fraction of propellant burnt simulations revealed significant agreement pertaining to propellant burnout time. The burnout times simulated by each MATLAB routine were found to be exactly the same. Due to the exactness of the solutions attained, the propellant burn rate computations performed by the MATLAB routines were considered to be accurate with respect to the public domain information used to simulate the propellant geometry. It must be noted, that the actual solutions may vary from those computed in the MATLAB routines, but can be easily updated by inserting the actual burn rate coefficients into the programs.

4.3 – Equations of Motion

Verification of the results attained by the ballistics code was achieved by comparison to results from public domain information. As previously mentioned, a M721 mortar round has an approximate muzzle velocity of 124 m/s or 407 ft/s. The muzzle velocities attained from the SI & Imperial compared extremely well. The corresponding percentage differences were calculated to be less than 1% from each simulation, which demonstrated a significant degree of accuracy.

Additionally, the results attained from the interior ballistics simulation provided key information pertaining to the kinematics of the projectile during the act of launching. Plots generated by the MATLAB routine include results for projectile displacement, velocity, and acceleration versus time in both imperial and SI units. These results are displayed in figures 4.7 through 4.9.

Further computational procedure validation can be observed when comparing the projectile position at the muzzle and length of the cannon barrel from each MATLAB routine. First, however, it is important to note that the actual internal length of the cannon barrel could not be confirmed from public domain data, therefore it can only be safely assumed that this value is less than the external length of the cannon barrel, which was discovered to be 40.32 inches or 1.024 m from a public domain source (M224, 60mm Mortar 60mm Lightweight Mortar, 2012). Furthermore, the length from the base of the projectile to the obturating ring was determined from a visual approximation of the mortar round utilizing the given projectile dimensions. With

this information the interior length of the barrel, according to the MATLAB routines, could be approximated by adding the distance traveled by the projectile at the muzzle to the previously mentioned lengths describing the distance between the projectile base and obturating ring. However, due to export control regulations, the results computed by the simulation cannot be revealed. Based off of the simulation results, it was decided that the data pertaining to the distance traveled by the projectile at the muzzle was viable, but without actual data representative of the true nature of the system, the results cannot be verified with 100% assurance.

Figure 4.7 represents the simulated projectile non-dimensional displacement results. Note that the green star on the plots denotes the computed non-dimensional time at which the propellant has completely burned out. The actual projectile displacement results obtained from the simulations cannot be displayed due to export control regulations.

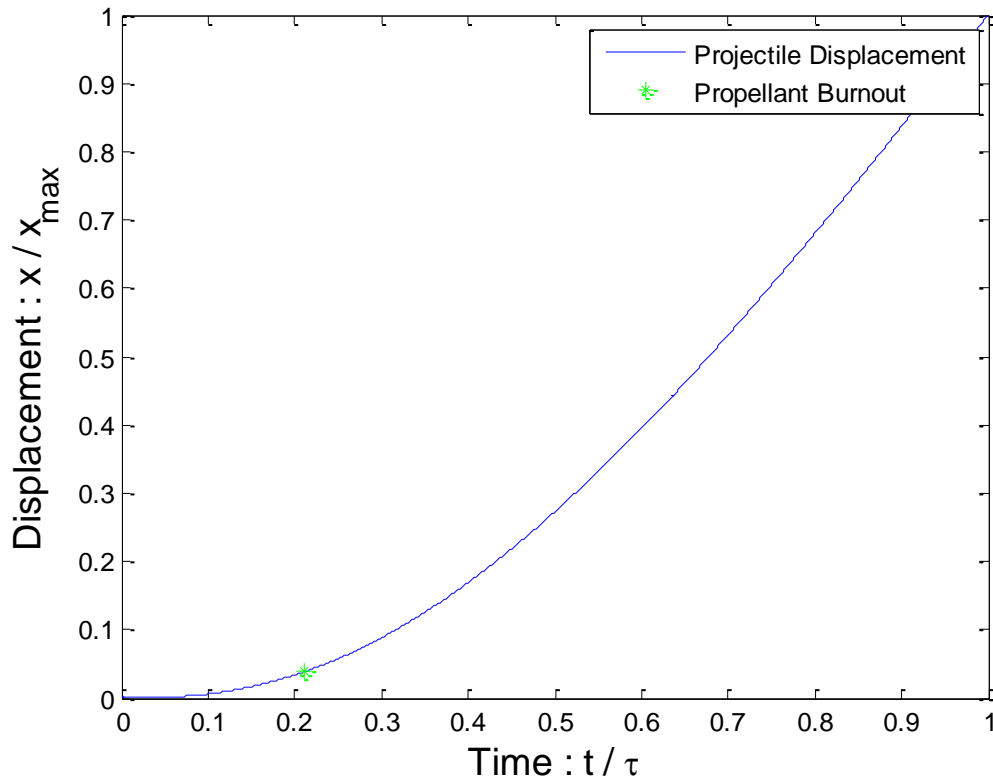


Figure 4.7 –Projectile Displacement (Non-Dimensional)

Figure 4.8 illustrates the computed projectile non-dimensional velocity results output by the simulation. As previously mentioned, verification of the computed muzzle velocities was achieved by comparison to performance data from public domain literature, which cited the muzzle velocities to be approximately 124 m/s or 407 ft/s.

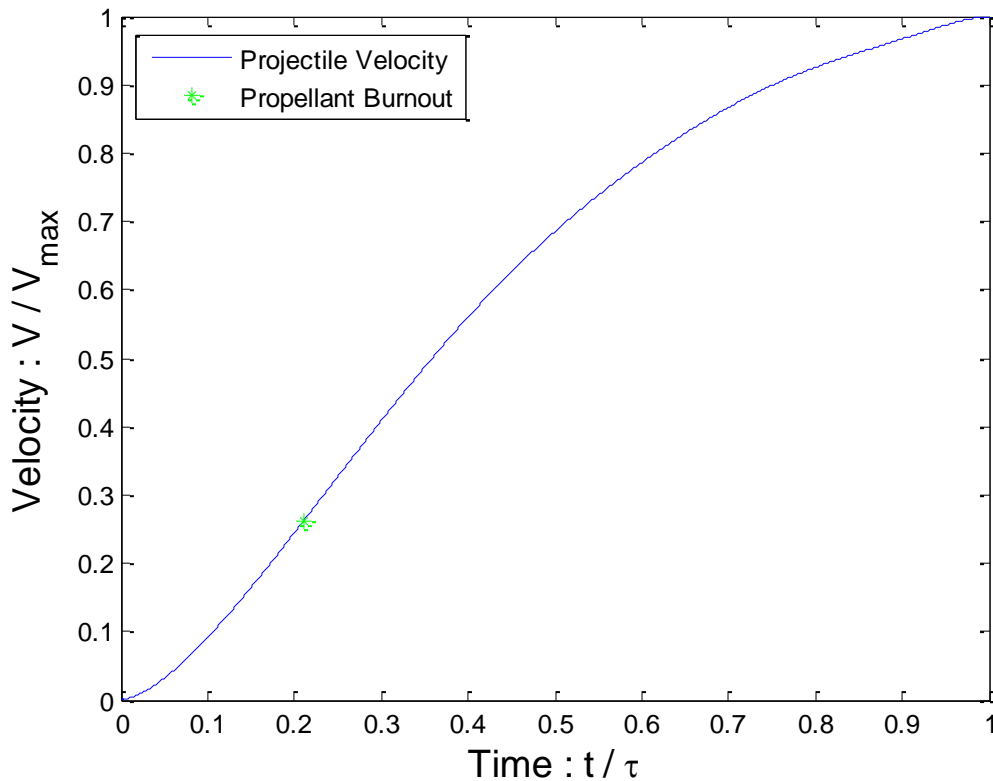


Figure 4.8 – Projectile Velocity (Non-Dimensional)

Further validation of the simplified interior ballistics simulation was attained through comparison of the shape and trends of the dynamics plots to existing literature. Figure 4.9 shows the non-dimensionalized dynamics results output by the simulation. Figure 4.10 displays projectile dynamics results from a 120-mm mortar round that was evaluated by a sophisticated three-dimensional interior ballistic simulation produced at Penn State University (Acharya, 2009).

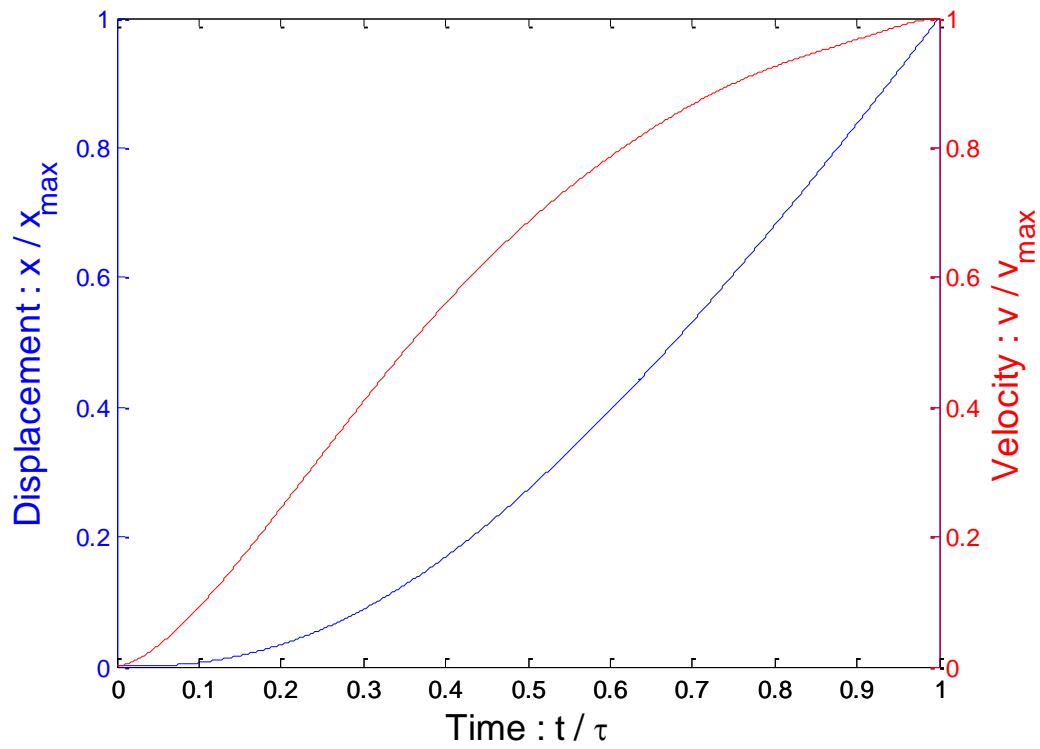


Figure 4.9 – Computed Projectile Dynamics (Non-Dimensional)

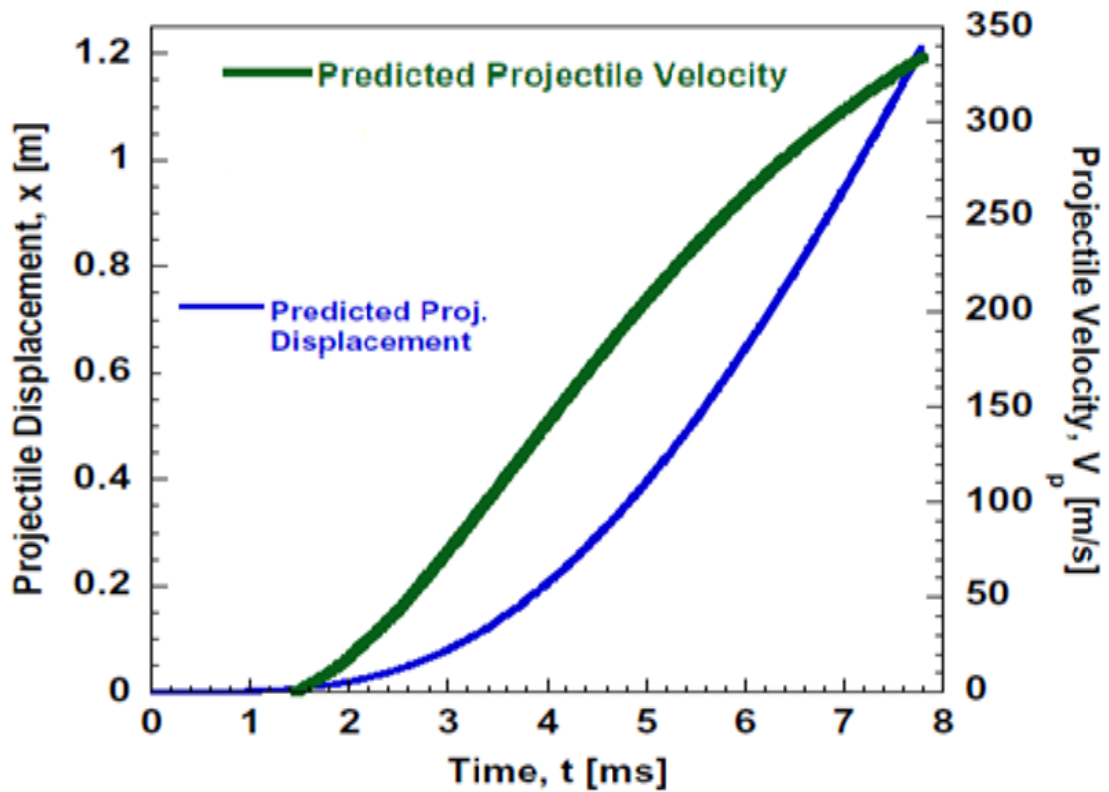


Figure 4.10 – Computed Projectile Dynamics of a 120-mm Mortar Round (Acharya, 2009)

Comparison of figures 4.9 and 4.10 revealed a strong correlation between the shapes and trends exhibited by the curves. The strong agreement corroborates the results obtained from the simplified interior ballistics analysis. Furthermore, comparison of the plots fortified the accuracy of the simplified ballistics simulation and reinforced the belief that neglecting solid propellant combustion modeling would not compromise the degree of accuracy for the motion dynamics results.

Figure 4.11 displays the non-dimensional acceleration of the projectile computed by the simulation. Comparison of the plots once again revealed a strong correlation between the Imperial unit and SI unit solutions. The maximum projectile accelerations from the Imperial unit and SI unit routines computed approximately the same number of g's. However, these values cannot be provided due to export control regulations.

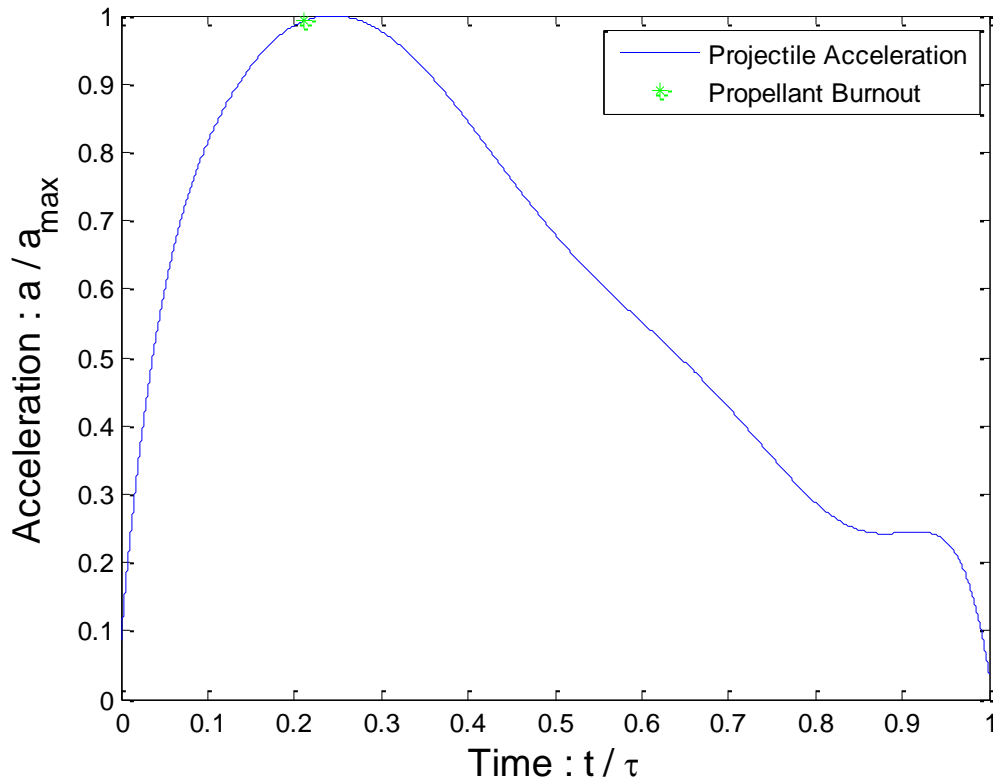


Figure 4.11 – Projectile Acceleration (Non-Dimensional)

An additional note of interest pertains to the shape of the acceleration curve. The shape computed corresponded extremely well to the shape of the curve fit pressure. This behavior

was anticipated due to the computational approach used to compute the acceleration, as displayed in equation 3.10.

4.4– Thermodynamics

Due to the lack of data available, the accuracy of the solutions attained from the thermodynamic portion of the MATLAB routines cannot be verified. Therefore, these solutions should be considered no more than approximations in the context of this paper. However, the behaviors exhibited by the obtained solutions can be validated through comparison of data examined in public domain literature with the output MATLAB results.

Figure 4.12 illustrates the computed volume behind the projectile. The shape and trends exhibited by the plot was comparable to the shape and trends of the projectile displacement. This behavior was anticipated since to the method used to calculate the volume was based off of a constant multiplied by the projectile displacement.

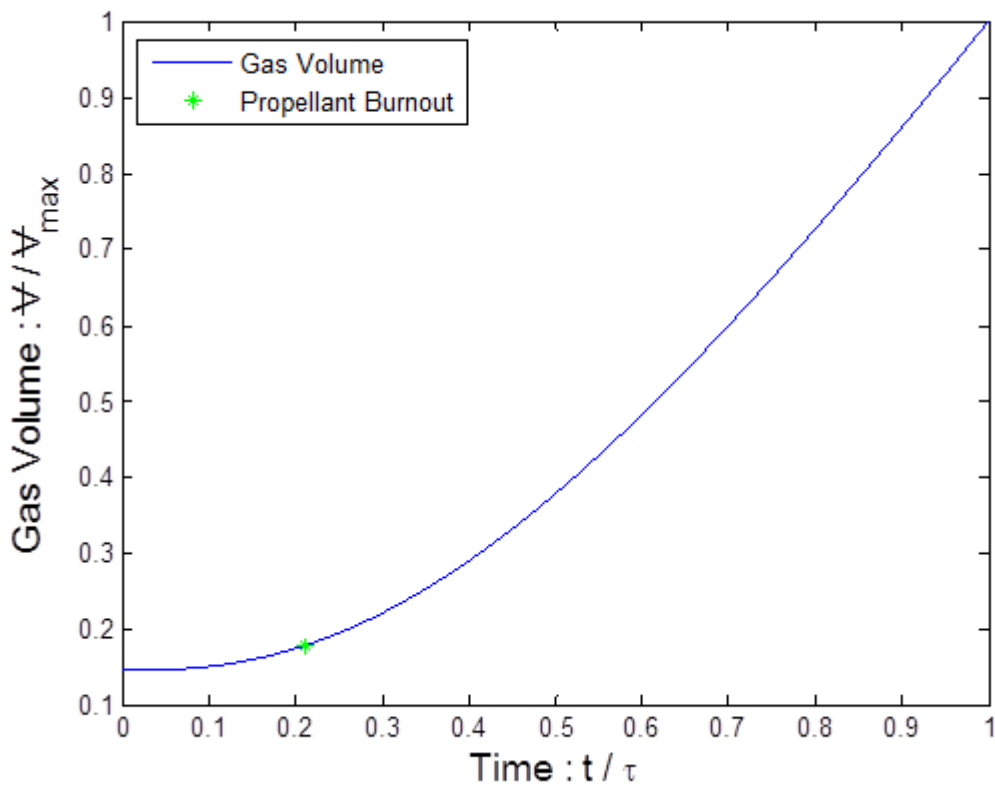


Figure 4.12 –Volume behind Projectile (Non-Dimensional)

Comparisons of the muzzle data and plots revealed a nearly perfect match between the imperial and SI MATLAB solutions. Taking this into consideration, the results obtained from the simulations were considered acceptable with respect to the data input into the MATLAB routines.

Figure 4.13 illustrates the computed solutions for the non-dimensional combustion gas temperature behind the projectile. Results obtained for the temperature solution could not be verified by any physical data. However, the trends and order of magnitude of the curves output by the MATLAB routines could be compared to a curve discovered within public domain literature, displayed in figure 4.14 (Shelton, Bergles, & Saha, 1973).

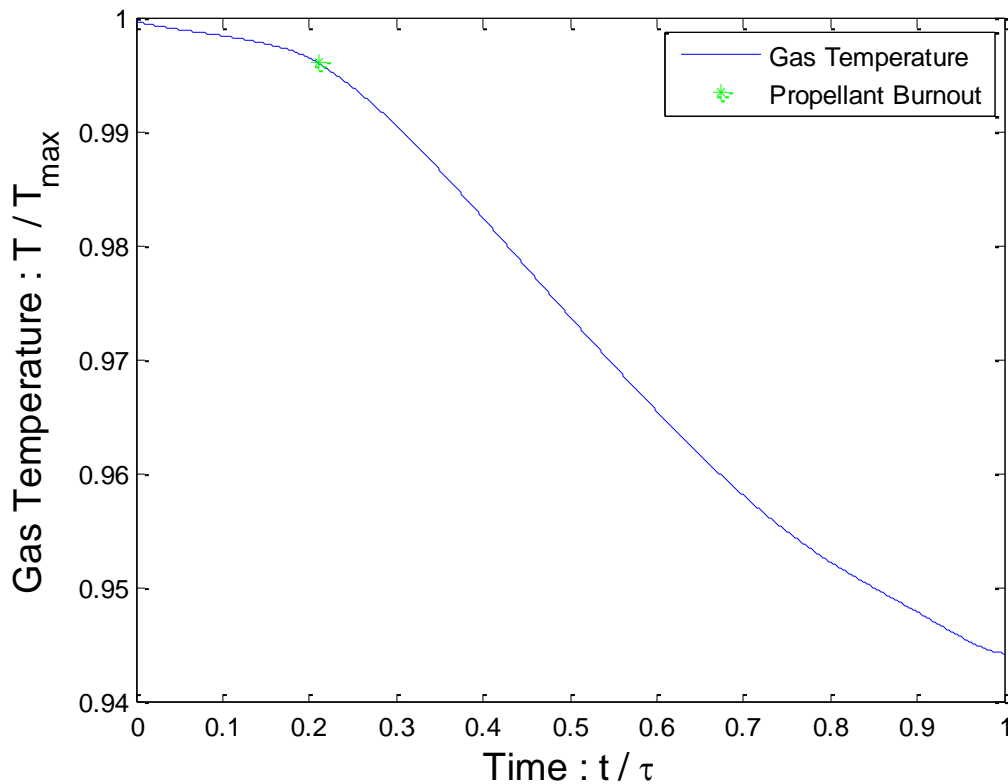


Figure 4.13 – Combustion Gas Temperature (Non-Dimensional)

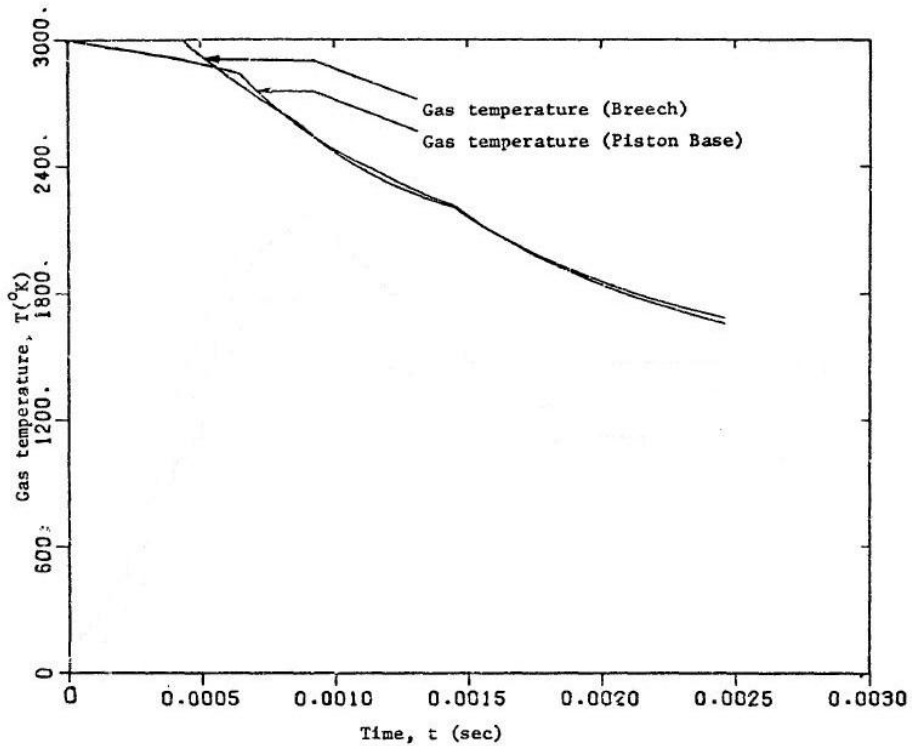


Figure 4.14 – Hypothetical Combustion Gas Temperature (Shelton, Bergles, & Saha, 1973)

Comparison of figure 4.13 to figure 4.14 revealed a distinct relationship pertaining to the shape of the curves, but a large variation in the ranges of temperatures observed. This is likely due to the scenarios represented by the plots. The plot of figure 4.14 represents a hypothetical scenario presented for the purposes of a study on heat transfer and gun barrel erosion (Shelton, Bergles, & Saha, 1973). Due to the hypothetical nature of the scenario, the actual data cannot be compared to any existing interior ballistic system; however, the shape of the plot can be used as a means of comparison since the system is theoretically based off of a propellant actuated interior ballistic system. With this in mind, the comparison of the hypothetical and computed plots shows very similar characteristics in describing the gas temperature of the system during the interior ballistic cycle. Therefore, it is assumed that the plots output by the MATLAB routines are representative of the nature of the systems based on the input data utilized in the routines.

One additional note of interest observed in figure 4.13 was the change in plot behavior at the indicated point of projectile burnout. The point of burnout indicated in the plots appeared to occur near an inflection point in the curve. Unfortunately, speculation pertaining to the

cause of this occurrence cannot be made since the propellant burnout time was not verified with physical data.

Approximations of the local speed of sound behind the projectile were computed by the program routines. Unfortunately, solutions obtained could not be verified with any physical data. Figure 4.15 illustrates the non-dimensional results obtained from the MATLAB simulation. Comparison of the solutions obtained from the Imperial Unit and SI Unit show a strong correlation in the trends exhibited by the plots, however, comparison of the physical data output by the plots revealed a percent difference of approximately 8.5% between the data sets. Considering the small variation between the minimum and maximum percentage differences, it was determined that the percent difference was the result of the specific gas constants used in the program routines. As previously mentioned, a majority of the data used in the program routines was obtained from public domain resources, and therefore could not be verified to be 100% representative of the actual physical data.

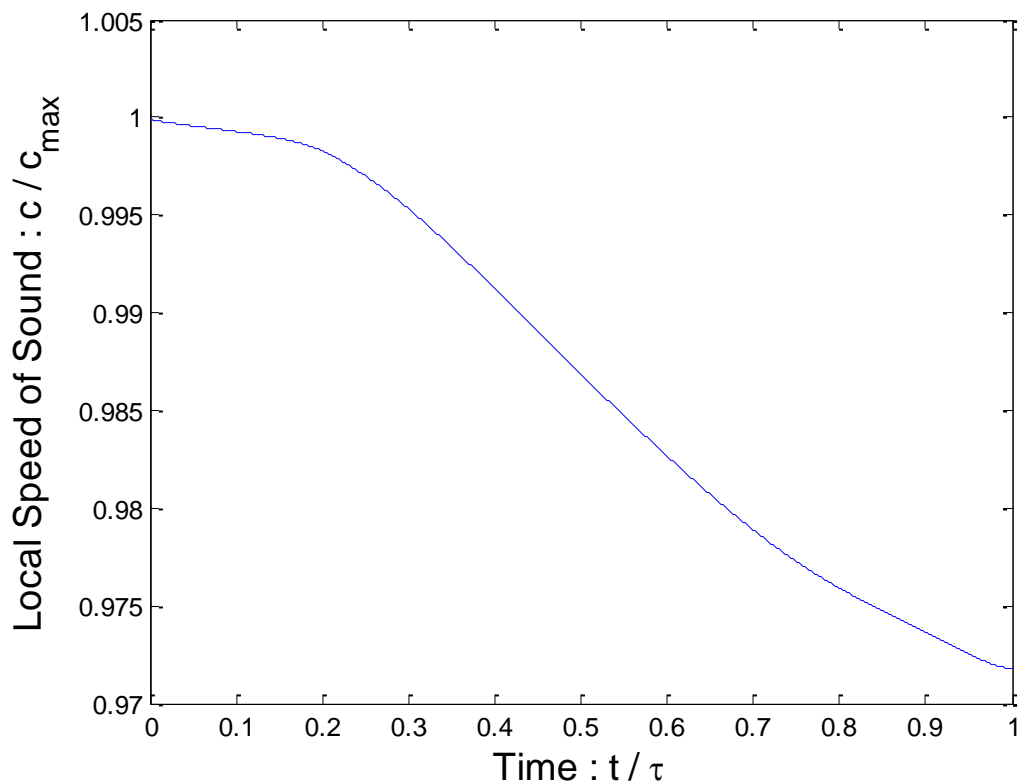


Figure 4.15 – Local Speed of Sound (Non-Dimensional)

The Helmholtz resonance was computed to be 185.2 Hz by the imperial unit routine and 170.1 Hz by the SI unit routine. Considering that the units describing the Helmholtz resonance were the same based for each of the routines, the solution output by each simulation was anticipated to be approximately equal. However, the imperial unit routine computed a Helmholtz resonance that was 8.1% higher than the SI simulation. The observed variation between the solutions was the result of variations between the local speed of sound solutions.

CHAPTER 5: CONCLUSION

Based upon the findings presented, an analysis conducted in the manner described in the context of this paper would prove to be beneficial by significantly simplifying the interior ballistics computations to obtain loading conditions for a highly efficient and highly accurate method to evaluate the integrity of the TL-UAV prototype. The interior ballistic model created in MATLAB will provide researchers with an effective design and optimization tool capable of evaluating the dynamics of the TL-UAV prototype from experimentally recorded pressure-time data. The contribution put forth in this thesis resides in the overall approach developed to produce effective simulation of the interior ballistics through a one-dimensional reduced order formulation of the thermodynamic-motion analysis of the projectile as described in the flow charts of figures C1 through C5 in the appendix.

Future work based of the results from this paper will enable users to conduct an explicit, three-dimensional simulation of the TL-UAV prototype by means of ANSYS LS-DYNA. Results obtained from such an analysis can be verified by comparison to results output by the interior ballistics simulation outlined in this thesis. Furthermore, by performing minor alterations to the MATLAB routines presented in this thesis, a variety of different scenarios can be described as long as archived pressure-time data is available. Additional modifications made to the base code would enable users to; conduct energy balances of the system, accurately evaluate the gas temperature and local speed of sound of the combustion gases, evaluate the momentum of the projectile, and to examine the Helmholtz Resonance. As for the analysis method, additional efforts put into this work would enable users to; evaluate the entire ballistics process of the system by coupling the interior ballistics program to intermediate and exterior ballistics routines, conduct thermal analyses of the system based off of updated thermodynamics simulations, and to conduct modal analyses of the system based off of Helmholtz resonance simulations.

REFERENCES

- M720/888 High Explosive Cartridges*. (1998, September 12). Retrieved 2012, from FAS Military Analysis Network.
- M721 60mm Illuminating Mortar Cartridge*. (1998, September 12). Retrieved 2011, from FAS Military Analysis Network: <http://www.fas.org/man/dod-101/sys/land/m721.htm>
- M224, 60mm Mortar 60mm Lightweight Mortar*. (2012). Retrieved 2012, from Military Factory: http://www.militaryfactory.com/smallarms/detail.asp?smallarms_id=18
- Acharya, R. (2009). *Modeling and Numerical Simulation of Interior Ballistic Process in a 120mm Mortar System*. University Park: Pennsylvania State University Department of Engineering.
- Anderson, R. D. (2006). *IBHVG2: Mortar Simulation With Interior Propellant Canister*. Aberdeen Proving Ground: Army Research Laboratory.
- Anderson, R. D., & Fickie, K. D. (1987). *IBHVG2 - A User's Guide*. Aberdeen Proving Ground: US Army Ballistic Research Laboratory.
- Asfaw, Y. (2008). *Structural Design and Analysis of an Existing Aerodynamically Optimised Mortar Shell*. Thesis, Addis Ababa University, Department of Mechanical Engineering.
- Brode, H. L., & Enstrom, J. E. (1970). *A Numerical Method for Calculating Interior Ballistics*. Santa Monica: The RAND Corporation.
- Browning, P. H. (2009). *Transient Flow Characteristics of a High Speed Rotary Valve*. Ph. D. Dissertation, West Virginia University, Department of Mechanical and Aerospace Engineering, Morgantown.
- Carlucci, D. E., & Jacobson, S. S. (2007). *Ballistics: Theory and Design of Guns and Ammunition*. Boca Raton: Taylor & Francis Group, LLC.
- Cooke, G. W. (2004, August 15). *60mm Mortar Ammunition and Fuzes*. Retrieved 2011, from Gary's U.S. Infantry Weapons Reference Guide: <http://www.inetres.com/gp/military/infantry/mortar/60mm.html>

- Cooke, G. W. (2004, August 15). *M224 60mm Lightweight Mortar*. Retrieved 2012, from Gary's U.S. Infantry Weapons Reference Guide:
<http://www.inetres.com/gp/military/infantry/mortar/M224.html>
- Create array of all ones*. (n.d.). Retrieved 2011, from MathWorks:
<http://www.mathworks.com/help/techdoc/ref/ones.html>
- Creating Symbolic Objects*. (n.d.). Retrieved 2011, from MathWorks:
<http://www.mathworks.com/help/toolbox/symbolic/sym.html>
- Cumulative Sum*. (n.d.). Retrieved 2011, from MathWorks:
<http://www.mathworks.com/help/techdoc/ref/cumsum.html>
- Department of the Army. (1984). *Military Explosives*. Washington D.C.: U.S. Army.
- Drew, D. A. (1983). Mathematical Modeling of Two-Phase Flow. *Annual Reviews Fluid Mechanics*.
- Erline, T. F., & Fischer, L. L. (1996). First Order Dynamic Tools for Rapid Assessment of Small Changes to Major Gun and Projectile Dynamic Parameters. In A. G. Pflegl (Ed.), *The Eighth U.S. Army Symposium on Gun Dynamics* (pp. 14-1 - 14-21). Newport: U.S. Army Armament Research, Development, and Engineering Center. Retrieved 2011
- Gonzalez, J. R. (1990). *Interior Ballistics Optimization*. Master's Thesis, Kansas State University, Department of Mechanical Engineering.
- Gough, P. S. (1990). *The XNOVAKTC Code*. Aberdeen Proving Ground: Ballistic Research Laboratory.
- Heiser, R. (1991). *Viscous Modeling of the Interior Ballistic Cycle*. European Research Office. Retrieved 2011
- Hibbeler, R. C. (2007). *Engineering Mechanics Dynamics* (11th ed.). Upper Saddle River: Pearson. Retrieved 2011
- Johnston, I. A. (2005). *The Noble-Abel Equation of State: Thermodynamic Derivations for Ballistics Modelling*. Edinburgh: Defence Science and Technology Organisation.

- Johnston, I. A. (2005). *Understanding and Predicting Gun Barrel Erosion*. Weapon System Division. Edinburgh: DSTO Defence Science and Technology Organisation.
- Matrix and Array Arithmetic* . (n.d.). Retrieved 2011, from MathWorks:
<http://www.mathworks.com/help/techdoc/ref/arithmeticooperators.html>
- McAssey, E. V. (1975). *Analysis of Flow and Heat Transfer in Small Arms*. Villanova University, Department of Mechanical Engineering. Durham: Army Research Office.
- Moran, M. J., & Shapiro, H. N. (2008). *Fundamentals of Engineering Thermodynamics* (6th ed.). Hoboken: John Wiley & Sons Inc. Retrieved 2011
- Nyberg, H. (2009). *Evaluation of Gun Propelling Charge Performance During the Life Cycle by Statistical Utilization of Data Collected in Test and Troop Gun Firings*. Helsinki University of Technology, Department of Biotechnology and Chemical Technology. Helsinki: Helsinki University of Technology. Retrieved from
<http://lib.tkk.fi/Diss/2009/isbn9789512297221/>
- Open Curve Fitting Tool*. (n.d.). Retrieved 2011, from MathWorks:
<http://www.mathworks.com/help/toolbox/curvefit/cftool.html>
- Polynomial Curve Fitting*. (n.d.). Retrieved 2011, from MathWorks:
<http://www.mathworks.com/help/techdoc/ref/polyfit.html>
- Ray, S. E., Newill, J. F., Nusca, M. J., & Horst, A. W. (2004). *Coupling of CFD and CSM Codes for the Study of Projectile Response to Ballistics Environments*. Aberdeen Proving Ground: Army Research Laboratory. Retrieved from <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA432903>
- Read Microsoft Excel Spreadsheet File*. (n.d.). Retrieved 2011, from MathWorks:
<http://www.mathworks.com/help/techdoc/ref/xlsread.html>
- Robbins, F. W., & Lynn, F. R. (1988). *Analytical Solutions to the Closed Bomb*. Aberdeen: Ballistic Research Laboratory.

- Schmidt, J. R., Nusca, M. J., & Horst, A. W. (2009). *Mortar Interior Ballistics: Sensitivity Studies Using IBHVG2 and Progress Toward a Multidimensional Representation*. Aberdeen Proving Ground: Army Research Laboratory. Retrieved 2011
- Shelton, S., Bergles, A., & Saha, P. (1973). *Study of Heat Transfer and Erosion in Gun Barrels*. Georgia Institute of Technology, School of Mechanical Engineering. Elgin Air Force Base: Air Force Armament Laboratory.
- Square Root*. (n.d.). Retrieved 2011, from MathWorks:
<http://www.mathworks.com/help/techdoc/ref/sqrt.html>
- Symbolic Integration*. (n.d.). Retrieved 2011, from MathWorks:
<http://www.mathworks.com/help/toolbox/symbolic/int.html>
- Symbolic Math Toolbox*. (n.d.). Retrieved 2011, from MathWorks:
http://www.mathworks.com/help/toolbox/symbolic/bs_fsi0.html
- Symbolic-to-Numeric Polynomial Conversion*. (n.d.). Retrieved 2011, from MathWorks:
<http://www.mathworks.com/help/toolbox/symbolic/sym2poly.html>
- U.S. Army. (2009). *M224 Mortar Fact File for the United States Army*. Retrieved 2011, from The United States Army: <http://www.army.mil/factfiles/equipment/indirect/m224.html>
- U.S. Army Materiel Command. (1964). *Engineering Design Handbook - Ammunition Series - Section 4, Design for Projection*. Washington D.C.: U.S. Army.
- U.S. Army Materiel Command. (1965). *Engineering Design Handbook - Ballistic Series - Interior Ballistics of Guns*. Washington D.C.: U.S. Army.
- U.S. Army Materiel Command. (1975). *Engineering Design Handbook - Propellant Actuated Devices*. Alexandria, VA: U.S. Army.
- Voldman, J. (2007). *Lumped-element Modeling with Equivalent Circuits*. Lecture Presentation, Massachusetts Institute of Technology. Retrieved 2012

Williams, A. W., Brandt, A. L., Kaste, P. J., & Colburn, J. W. (2006). *Experimental Studies of the No. 41 Primer and Ignition of 5.56-mm Ammunition*. Aberdeen Proving Ground: Army Research Laboratory.

APPENDICES

Appendix A – MATLAB Code (Imperial Units)⁷

```

% The following code solves the Interior Ballistics of mortars from the input data
clear all
clc
format long
syms t

% INPUT PARAMETERS DESCRIBING MORTAR
prompt_IMP={'Cannon Bore Diameter [in]:','Length of Cannon Barrel [in]:','Projectile Mass
[lb]:','Max Projectile Diameter [in]','Length from Base of Projectile to Obturating Ring
[in]:','Volume of Projectile from Base to Obturating Ring [in^3]:','Initial Mass of Propelling
Charge [lb]:','Initial Diameter of Propelling Charge [in]:','Propellant Web [in]:','Solid Density
of Propelling Charge [lb/in^3]:','Specific Heat Ratio of Propellant:','Adiabatic Flame
Temperature of Propellant [deg R]:','Covolume of Propellant [in^3/lb]:','Mean Molecular Weight of
Combustion Products [lb/mol]:','Propellant Burn Rate Coefficient [in/s/(psi)^n]:','Propellant
Burn Rate Exponent:','Ambient Pressure [psi]:','Ambient Temperature [deg R]:'};
ptitle_IMP='INPUT PARAMETERS FOR MORTAR INTERIOR BALLISTICS';
numlines_IMP=1;
PROJECTILE_IMP={0,'0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','14.7','540'};
options.Resize='on';
options.WindowStyle='modal';
user_input_IMP=inputdlg(prompt_IMP,ptitle_IMP,numlines_IMP,PROJECTILE_IMP,options);
%- - - - Launcher Properties - - - -%
d_cb_IMP=str2double(user_input_IMP{1}); % Cannon Barrel Diameter
L_cb_IMP=str2double(user_input_IMP{2}); % Cannon Barrel Depth
%- - - - Mortar Properties - - - -%
m_prj_IMP=str2double(user_input_IMP{3}); % Projectile Mass
d_prj_IMP=str2double(user_input_IMP{4}); % Projectile Diameter - - - - -
- - - - - UNUSED!!!
L_base_ob_IMP=str2double(user_input_IMP{5}); % Base to Obturating Ring Length
V_base_ob_IMP=str2double(user_input_IMP{6}); % Volume of Projectile behind Obturating Ring
%- - - - Propellant Properties - - - -%
m_ch_i_IMP=str2double(user_input_IMP{7}); % Initial Charge Mass
d_ch_i_IMP=str2double(user_input_IMP{8}); % Initial Charge Diameter
web_IMP=str2double(user_input_IMP{9}); % Propellant Web - - - - -
- - - - - UNUSED!!!
rho_ch_IMP=str2double(user_input_IMP{10}); % Solid Propellant Density
gamma_ch_IMP=str2double(user_input_IMP{11}); % Specific Heat Ratio
Tf_ch_IMP=str2double(user_input_IMP{12}); % Adiabatic Flame Temperature
b_ch_IMP=str2double(user_input_IMP{13}); % Covolume - - - - -
- - - - - UNUSED!!!
M_gas_IMP=str2double(user_input_IMP{14}); % Molecular Weight
%- - - - Propellant Burn Rate Data - - - -%
beta_IMP=str2double(user_input_IMP{15});
n_IMP=str2double(user_input_IMP{16});
%- - - - Ambient Conditions - - - -%
P_amb_IMP=str2double(user_input_IMP{17}); % Ambient Pressure
T_amb_IMP=str2double(user_input_IMP{18}); % Ambient Temperature - - - - -
- - - - - UNUSED!!!

% CONSTANTS AND VARIABLE CALCULATIONS FROM INPUTS
%- - - - Constants - - - -%
g_IMP=12*32.174; % Acceleration due to gravity [in/s^2]
R_univ_IMP=12*1545.38963; % Universal gas constant [in-lbf/(lb-mol*degR)]
%- - - - Launcher Calculations - - - -%
A_cb_IMP=(pi/4)*d_cb_IMP^2; % Cross sectional area of cannon barrel [in^2]
V_cb_IMP=L_cb_IMP*A_cb_IMP; % Volume of empty cannon barrel [in^3] - - - - -
- - - - - UNUSED!!!
%- - - - Propellant Calculations - - - -%
V_ch_i_IMP=m_ch_i_IMP/rho_ch_IMP; % Initial volume of charge [in^3]
r_ch_i_IMP=d_ch_i_IMP/2; % Initial radius of charge [in]
L_ch_IMP=V_ch_i_IMP/(pi*r_ch_i_IMP^2); % Equivalent length of charge [in]
%- - - - Combustion Gas Calculations - - - -%
R_gas_IMP=R_univ_IMP/M_gas_IMP; % Specific gas constant for combustion products [in-lbf/(lb-
degR)]

```

⁷ Variable quantities used in computations that represented prototype have been deliberately omitted due to export control regulations

```

R_air_IMP=0; % Specific gas constant for air [in-lbf/(lb-degR)] - - - - -
- - - - - UNUSED!!!
rho_air_IMP=0; % Density of air [lb/in^3] - - - - -
- - - - - UNUSED!!!
Cp_gas_IMP=(gamma_ch_IMP*R_gas_IMP)/(gamma_ch_IMP-1); % Specific Heat at Constant Pressure - - -
- - - - - UNUSED!!!
Cv_gas_IMP=R_gas_IMP/(gamma_ch_IMP-1); % Specific HEat at Constant Volume
F_gas_IMP=Tf_ch_IMP*R_gas_IMP; % Impetus of combustion gas [in-lbf/lbm]
V_gas_i_IMP=L_base_ob_IMP*pi/4*d_cb_IMP^2-V_base_ob_IMP-V_ch_i_IMP; % Initial chamber volume
prior to combustion [in^3]

% INPUT PRESSURE vs. TIME DATA
t_exp_IMP=transpose(xlsread('P-t Data.xlsx','Data','B:B')); % Retrieval of Time Data from Excel
File
P_exp_IMP=transpose(xlsread('P-t Data.xlsx','Data','D:D')); % Retrieval of ABSOLUTE Pressure Data
from Excel File

% CURVE FITTING OF PRESSURE VS. TIME DATA
[P_cf_IMP,R2_IMP]=polyfit(t_exp_IMP,P_exp_IMP,9); % Performs 9th order polynomial curve fit to
data
%- - - - Time Data - - - -%
t_last_IMP=max(t_exp_IMP); % Finds maximum value of time from input data
t_IMP=0:0.000002:t_last_IMP; % Creates the time vector from 0 seconds to maximum value of time
from input data
num_i_IMP=find(t_IMP,1,'last'); % Determines the number of time steps
t_sym_IMP=[t^9 t^8 t^7 t^6 t^5 t^4 t^3 t^2 t^1 t^0]; % Symbolic time coefficients for integration
purposes
%- - - - Pressure Attained from Curve Fit - - - -%
P_IMP=(P_cf_IMP(1).*t_IMP.^9+P_cf_IMP(2).*t_IMP.^8+P_cf_IMP(3).*t_IMP.^7+P_cf_IMP(4).*t_IMP.^6+P
cf_IMP(5).*t_IMP.^5+P_cf_IMP(6).*t_IMP.^4+P_cf_IMP(7).*t_IMP.^3+P_cf_IMP(8).*t_IMP.^2+P_cf_IMP(9)
.*t_IMP+P_cf_IMP(10)); % Pressure [Pa]
P_sym_IMP=dot(P_cf_IMP,t_sym_IMP); % Pressure as a function of symbolic time

% PROPELLANT BURN RATE CALCULATIONS
%- - - - Determination of Time of Propellant Burnout - - - -%
TEST1_IMP=beta_IMP.*P_IMP.^n_IMP; %|
TEST2_IMP=TEST1_IMP.*t_IMP; %|
TEST3_IMP=cumsum(TEST2_IMP); %|- Computations to determine time at which propellant is
completely combusted
TEST4_IMP=ones(1,num_i_IMP).*r_ch_i_IMP; %|
TEST5_IMP=TEST4_IMP-TEST3_IMP; %|
TEST6_IMP=find(TEST5_IMP>0,1,'last'); %|
t_BRN_IMP=0:0.000002:((TEST6_IMP-1)*0.000002); % Creates time vector corresponding to burning
propellant
%- - - - Burn Rate Calculations - - - -%
P_BRN_IMP=(P_cf_IMP(1).*t_BRN_IMP.^9+P_cf_IMP(2).*t_BRN_IMP.^8+P_cf_IMP(3).*t_BRN_IMP.^7+P_cf_IMP
(4).*t_BRN_IMP.^6+P_cf_IMP(5).*t_BRN_IMP.^5+P_cf_IMP(6).*t_BRN_IMP.^4+P_cf_IMP(7).*t_BRN_IMP.^3+P
_cf_IMP(8).*t_BRN_IMP.^2+P_cf_IMP(9).*t_BRN_IMP+P_cf_IMP(10)); % Pressure during propellant
combustion [Pa]
BR_IMP=beta_IMP.*P_BRN_IMP.^n_IMP; % Burn Rate computation for propellant [in/s]
r_ch_delta_IMP=BR_IMP.*t_BRN_IMP; % Change in propellant radius per time step [in]
r_ch_cumsum_IMP=cumsum(r_ch_delta_IMP); % Cumulative change in propellant radius [in]
r_ch_i_vec_IMP=ones(1,TEST6_IMP).*r_ch_i_IMP; % Vector with values equal to initial charge radius
r_ch_BRN_SI=r_ch_i_vec_IMP-r_ch_cumsum_IMP; % Radius of propellant during burning [in]
r_ch_BO_IMP=zeros(1,(num_i_IMP-TEST6_IMP)); % Radius of propellant after burnout (NO PROPELLANT
REMAINING) [in]
r_ch_IMP=[r_ch_BRN_SI r_ch_BO_IMP]; % Radius of propellant at each time step [in]
V_ch_IMP=L_ch_IMP.*pi.*r_ch_IMP.^2; % Volume of propelling charge at each time step [in^3]
m_ch_IMP=rho_ch_IMP.*V_ch_IMP; % Mass of propelling charge at each time step [lbm]
%- - - - Total Mass of Projectile and Charge Calculation - - - -%
m_prch_IMP=m_prj_IMP+m_ch_IMP; % Combined mass of projectile and propellign charge at each time
step [lbm]
N_ch_IMP=m_ch_i_IMP-m_ch_IMP; % Mass of Propellant Burnt [lbm]
m_fraction_IMP=(N_ch_IMP./m_ch_i_IMP); % Mass Fraction of Burnt Propellant [%]

% KINEMATICS FROM CURVE FIT PRESSURE
%- - - - Acceleration - - - -%
a_IMP=g_IMP.*(A_cb_IMP./m_prch_IMP).*P_IMP; % Acceleration of Projectile w.r.t. time [in/s^2]
a_ft_IMP=a_IMP./12; % Converts Acceleration to [ft/s^2]

%- - - - Velocity - - - -%

```

```

v_cof_IMP=sym2poly(int(P_sym_IMP,t)); % Velocity coefficients
v_IMP=g_IMP.*(A_cb_IMP./m_prch_IMP).*(v_cof_IMP(1).*t_IMP.^10+v_cof_IMP(2).*t_IMP.^9+v_cof_IMP(3)
.*t_IMP.^8+v_cof_IMP(4).*t_IMP.^7+v_cof_IMP(5).*t_IMP.^6+v_cof_IMP(7).*t_I
MP.^4+v_cof_IMP(8).*t_IMP.^3+v_cof_IMP(9).*t_IMP.^2+v_cof_IMP(10).*t_IMP+v_cof_IMP(11));
v_ft_IMP=v_IMP./12; % Converts Velocity to [ft/s]

%- - - - Position - - - -%
x_cof_IMP=sym2poly(int(int(P_sym_IMP,t),t)); % Position coefficients
x_IMP=g_IMP.*(A_cb_IMP./m_prch_IMP).*(x_cof_IMP(1).*t_IMP.^11+x_cof_IMP(2).*t_IMP.^10+x_cof_IMP(3)
).*t_IMP.^9+x_cof_IMP(4).*t_IMP.^8+x_cof_IMP(5).*t_IMP.^7+x_cof_IMP(6).*t_IMP.^6+x_cof_IMP(7).*t_I
MP.^5+x_cof_IMP(8).*t_IMP.^4+x_cof_IMP(9).*t_IMP.^3+x_cof_IMP(10).*t_IMP.^2+x_cof_IMP(11).*t_IMP
+x_cof_IMP(12));
x_ft_IMP=x_IMP./12; % Converts Displacement to [ft]

% THERMODYNAMICS FROM CURVE FIT PRESSURE
V_gas_IMP=V_gas_i_IMP+x_IMP.*A_cb_IMP; % Volume available to combustion gas behind projectile
[in^3]
V_gas_ft_IMP=V_gas_IMP./1728; % Converts Volume to [ft^3]
T_gas_calc_IMP=Tf_ch_IMP-
((m_prch_IMP(2:end).*v_IMP(2:end).^2)./(2.*N_ch_IMP(2:end).*Cv_gas_IMP)); % Calculates
Temperature of gas behind projectile !!!VALUE FOR INDEX 1 WILL BE NAN - NEED TO ACCOUNT FOR IN
NEXT STEP
T_gas_IMP=[Tf_ch_IMP T_gas_calc_IMP];
c_IMP=sqrt(gamma_ch_IMP.*R_gas_IMP.*T_gas_IMP);
c_ft_IMP=c_IMP./12;

% HELMHOLTZ RESONANCE
f_helm_IMP=c_IMP(num_i_IMP)/(2*pi*L_cb_IMP)

% GENERAL RESULTS
v_muz_IMP=v_IMP(num_i_IMP)

% GENERATION OF PLOTS
figure
plot(t_exp_IMP,P_amb_IMP+P_exp_IMP,'red',t_IMP,P_IMP,'blue',t_IMP(TEST6_IMP),P_IMP(TEST6_IMP),'g*
')
legend('Archived Pressure Data','Curve Fit Pressure','Propellant Burnout')
%title('Pressure vs. Time (Imperial Units)');
xlabel('Time (s)')
ylabel('Pressure (psi)')

figure
plot(t_IMP,x_IMP,t_IMP(TEST6_IMP),x_IMP(TEST6_IMP),'g*')
legend('Position','Propellant Burnout','Location','NorthWest')
%title('Position vs. Time (Imperial Units)')
xlabel('Time (s)')
ylabel('Position (in)')

figure
plot(t_IMP,v_ft_IMP,t_IMP(TEST6_IMP),v_ft_IMP(TEST6_IMP),'g*')
legend('Velocity','Propellant Burnout','Location','NorthWest')
%title('Velocity vs. Time (Imperial Units)')
xlabel('Time (s)')
ylabel('Velocity (ft/s)')

figure
plot(t_IMP,a_ft_IMP,t_IMP(TEST6_IMP),a_ft_IMP(TEST6_IMP),'g*')
legend('Acceleration','Propellant Burnout')
%title('Acceleration vs. Time (Imperial Units)')
xlabel('Time (s)')
ylabel('Acceleration (ft/s^2)')

figure
plot(t_IMP,V_gas_IMP,t_IMP(TEST6_IMP),V_gas_IMP(TEST6_IMP),'g*')
legend('Volume Behind Projectile','Propellant Burnout','Location','NorthWest')
%title('Volume Behind Projectile vs. Time (Imperial Units)')
xlabel('Time (s)')
ylabel('Volume (in^3)')

figure
plot(t_IMP,T_gas_IMP,t_IMP(TEST6_IMP),T_gas_IMP(TEST6_IMP),'g*')

```

```

legend('Gas Temperature Behind Projectile','Propellant Burnout')
%title('Gas Temperature Behind Projectile vs. Time (Imperial Units)')
xlabel('Time (s)')
ylabel('Temperature (degrees R)')

figure
plot(t_IMP,m_ch_IMP,t_IMP(TEST6_IMP),m_ch_IMP(TEST6_IMP),'g*')
legend('Charge Mass','Propellant Burnout')
%title('Charge Mass vs. Time (Imperial Units)')
xlabel('Time (s)')
ylabel('Mass (lbm)')

figure
plot(t_IMP(1:TEST6_IMP),m_fraction_IMP(1:TEST6_IMP))
%title('Mass Fraction of Propellant Burnt vs. Time (Imperial Units)')
xlabel('Time (s)')
ylabel('Mass Fraction Burnt')

figure
plot(t_IMP,c_IMP)
%title('Local Speed of Sound vs. Time (Imperial Units)')
xlabel('Time (s)')
ylabel('Speed of Sound (in/s)')

figure
plot(t_IMP,c_ft_IMP)
%title('Local Speed of Sound vs. Time (Imperial Units)')
xlabel('Time (s)')
ylabel('Speed of Sound (ft/s)')

```

Appendix B – MATLAB Code (SI Units)⁸

```

% The following code solves the Interior Ballistics of mortars from the input data
clear all
clc
format long
syms t

% INPUT PARAMETERS DESCRIBING MORTAR
prompt_SI={'Cannon Bore Diameter [m]:','Length of Cannon Barrel [m]:','Projectile Mass
[kg]:','Max Projectile Diameter [m]:','Length from Base of Projectile to Obturating Ring
[m]:','Volume of Projectile from Base to Obturating Ring [m^3]:','Initial Mass of Propelling
Charge [kg]:','Initial Diameter of Propelling Charge [m]:','Propellant Web [m]','Solid Density of
Propelling Charge [kg/m^3]:','Specific Heat Ratio of Propellant:','Adiabatic Flame Temperature of
Propellant [K]:','Covolume of Propellant [m^3/kg]:','Mean Molecular Weight of Combustion Products
[kg/mol]:','Propellant Burn Rate Coefficient [mm/s/(Mpa)^n]:','Propellant Burn Rate
Exponent:','Ambient Pressure [Pa]:','Ambient Temperature [K]:'};
ptitle_SI='INPUT PARAMETERS FOR MORTAR INTERIOR BALLISTICS';
numlines_SI=1;
PROJECTILE_SI={0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','101325','300'};
options.Resize='on';
options.WindowStyle='modal';
user_input_SI=inputdlg(prompt_SI,ptitle_SI,numlines_SI,PROJECTILE_SI,options);
%- - - - Launcher Properties - - - -%
d_cb_SI=str2double(user_input_SI{1}); % Cannon Barrel Diameter
L_cb_SI=str2double(user_input_SI{2}); % Cannon Barrel Depth
%- - - - Mortar Properties - - - -%
m_prj_SI=str2double(user_input_SI{3}); % Projectile Mass
d_prj_SI=str2double(user_input_SI{4}); % Projectile Diameter - - - - -
- - - - - UNUSED!!!
L_base_ob_SI=str2double(user_input_SI{5}); % Base to Obturating Ring Length
V_base_ob_SI=str2double(user_input_SI{6}); % Volume of Projectile behind Obturating Ring
%- - - - Propellant Properties - - - -%
m_ch_i_SI=str2double(user_input_SI{7}); % Initial Charge Mass
d_ch_i_SI=str2double(user_input_SI{8}); % Initial Charge Diameter
web_SI=str2double(user_input_SI{9}); % Propellant Web - - - - -
- - - - - UNUSED!!!
rho_ch_SI=str2double(user_input_SI{10}); % Solid Propellant Density
gamma_ch_SI=str2double(user_input_SI{11}); % Specific Heat Ratio
Tf_ch_SI=str2double(user_input_SI{12}); % Adiabatic Flame Temperature
b_ch_SI=str2double(user_input_SI{13}); % Covolume - - - - -
- - - - - UNUSED!!!
M_gas_SI=str2double(user_input_SI{14}); % Molecular Weight
%- - - - Propellant Burn Rate Data - - - -%
beta_SI=str2double(user_input_SI{15});
n_SI=str2double(user_input_SI{16});
%- - - - Ambient Conditions - - - -%
P_amb_SI=str2double(user_input_SI{17}); % Ambient Pressure
T_amb_SI=str2double(user_input_SI{18}); % Ambient Temperature

% CONSTANTS AND VARIABLE CALCULATIONS FROM INPUTS
%- - - - Constants - - - -%
g_SI=9.81; % Acceleration due to gravity [m/s^2] - - - - -
- - - - - UNUSED!!!
R_univ_SI=8.3144621; % Universal gas constant [J/(mol*K)]
%- - - - Launcher Calculations - - - -%
A_cb_SI=(pi/4)*d_cb_SI^2; % Cross sectional area of cannon barrel [m^2]
V_cb_SI=L_cb_SI*A_cb_SI; % Volume of empty cannon barrel [m^3] - - - - -
- - - - - UNUSED!!!
%- - - - Propellant Calculations - - - -%
V_ch_i_SI=m_ch_i_SI/rho_ch_SI; % Initial volume of charge [m^3]
r_ch_i_SI=d_ch_i_SI/2; % Initial radius of charge [m]
L_ch_SI=V_ch_i_SI/(pi*r_ch_i_SI^2); % Equivalent length of charge [m]
%- - - - Combustion Gas Calculations - - - -%
R_gas_SI=R_univ_SI/M_gas_SI; % Specific gas constant for combustion products [J/(kg*K)]
R_air_SI=287.058; % Specific gas constant for air [J/(kg*K)] - - - - -
- - - - - UNUSED!!!

```

⁸ Variable quantities used in computations that represented prototype have been deliberately omitted due to export control regulations

```

rho_air_SI=1.225; % Density of air [kg/m^3] - - - - -
-- UNUSED!!!
Cp_gas_SI=(gamma_ch_SI*R_gas_SI)/(gamma_ch_SI-1); % Specific Heat at Constant Pressure - - - - -
-- UNUSED!!!
Cv_gas_SI=R_gas_SI/(gamma_ch_SI-1); % Specific HEat at Constant Volume
F_gas_SI=Tf_ch_SI*R_gas_SI; % Impetus of combustion gas [J/kg]
V_gas_i_SI=L_base_ob_SI*pi/4*d_cb_SI^2-V_base_ob_SI-V_ch_i_SI; % Initial chamber volume prior to
combustion [m^3]

% INPUT PRESSURE vs. TIME DATA
t_exp_SI=transpose(xlsread('P-t Data.xlsx','Data','B:B')); % Retrieval of Time Data from Excel
File
P_exp_SI=transpose(xlsread('P-t Data.xlsx','Data','F:F')); % Retrieval of ABSOLUTE Pressure Data
from Excel File

% CURVE FITTING OF PRESSURE VS. TIME DATA
[P_cf_SI,R2_SI]=polyfit(t_exp_SI,P_exp_SI,9); % Performs 9th order polynomial curve fit to data
%- - - - Time Data - - - -%
t_last_SI=max(t_exp_SI); % Finds maximum value of time from input data
t_SI=0:0.000002:t_last_SI; % Creates the time vector from 0 seconds to maximum value of time from
input data
num_i_SI=find(t_SI,1,'last'); % Determines the number of time steps
t_sym_SI=[t^9 t^8 t^7 t^6 t^5 t^4 t^3 t^2 t^1 t^0]; % Symbolic time coefficients for integration
purposes
%- - - - Pressure Attained from Curve Fit - - - -%
P_SI=(P_cf_SI(1).*t_SI.^9+P_cf_SI(2).*t_SI.^8+P_cf_SI(3).*t_SI.^7+P_cf_SI(4).*t_SI.^6+P_cf_SI(5).
*t_SI.^5+P_cf_SI(6).*t_SI.^4+P_cf_SI(7).*t_SI.^3+P_cf_SI(8).*t_SI.^2+P_cf_SI(9).*t_SI+P_cf_SI(10)
); % Pressure [Pa]
P_sym_SI=dot(P_cf_SI,t_sym_SI); % Pressure as a function of symbolic time

% PROPELLANT BURN RATE CALCULATIONS
%- - - - Determination of Time of Propellant Burnout - - - -%
TEST1_SI=beta_SI.*(P_SI./1E6).^n_SI; %|
TEST2_SI=TEST1_SI.*t_SI; %|
TEST3_SI=cumsum(TEST2_SI); %|- Computations to determine time at which
propellant is completely combusted
TEST4_SI=ones(1,num_i_SI).*(r_ch_i_SI.*1000); %|
TEST5_SI=TEST4_SI-TEST3_SI; %|
TEST6_SI=find(TEST5_SI>0,1,'last'); %|
t_BRN_SI=0:0.000002:(TEST6_SI-1)*0.000002); % Creates time vector corresponding to burning
propellant
%- - - - Burn Rate Calculations - - - -%
P_BRN_SI=(P_cf_SI(1).*t_BRN_SI.^9+P_cf_SI(2).*t_BRN_SI.^8+P_cf_SI(3).*t_BRN_SI.^7+P_cf_SI(4).*t_B
RN_SI.^6+P_cf_SI(5).*t_BRN_SI.^5+P_cf_SI(6).*t_BRN_SI.^4+P_cf_SI(7).*t_BRN_SI.^3+P_cf_SI(8).*t_BR
N_SI.^2+P_cf_SI(9).*t_BRN_SI+P_cf_SI(10)); % Pressure during propellant combustion [Pa]
BR_SI=beta_SI.*(P_BRN_SI./1E6).^n_SI; % Burn Rate computation for propellant [mm/s]
r_ch_delta_SI=BR_SI.*t_BRN_SI; % Change in propellant radius per time step [mm]
r_ch_cumsum_SI=cumsum(r_ch_delta_SI); % Cumulative change in propellant radius [mm]
r_ch_i_vec_SI=ones(1,TEST6_SI).*(r_ch_i_SI.*1000); % Vector with values equal to initial charge
radius
r_ch_BRN_SI=r_ch_i_vec_SI-r_ch_cumsum_SI; % Radius of propellant during burning [mm]
r_ch_BO_SI=zeros(1,(num_i_SI-TEST6_SI)); % Radius of propellant after burnout (NO PROPELLANT
REMAINING) [mm]
r_ch_mm_SI=[r_ch_BRN_SI r_ch_BO_SI]; % Radius of propellant at each time step [mm]
r_ch_SI=r_ch_mm_SI./1000; % CONVERTS RADIUS OF PROPELLANT TO [m]
V_ch_SI=L_ch_SI.*pi.*r_ch_SI.^2; % Volume of propelling charge at each time step [m^3]
m_ch_SI=rho_ch_SI.*V_ch_SI; % Mass of propelling charge at each time step [kg]
%- - - - Total Mass of Projectile and Charge Calculation - - - -%
m_prch_SI=m_prj_SI+m_ch_SI; % Combined mass of projectile and propellign charge at each time step
[kg]
N_ch_SI=m_ch_i_SI-m_ch_SI; % Mass of Propellant Burnt [kg]
m_fraction_SI=(N_ch_SI./m_ch_i_SI); % Mass Fraction of Burnt Propellant

% KINEMATICS FROM CURVE FIT PRESSURE
%- - - - Acceleration - - - -%
a_SI=(A_cb_SI./m_prch_SI).*P_SI; % Acceleration of Projectile w.r.t. time
%a_mSI=(A_cb_SI./m_prj_SI).*P_SI;

%- - - - Velocity - - - -%
v_cof_SI=sym2poly(int(P_sym_SI,t)); % Velocity coefficients

```



```

v_SI=(A_cb_SI./m_prch_SI).*(v_cof_SI(1).*t_SI.^10+v_cof_SI(2).*t_SI.^9+v_cof_SI(3).*t_SI.^8+v_cof
_SI(4).*t_SI.^7+v_cof_SI(5).*t_SI.^6+v_cof_SI(6).*t_SI.^5+v_cof_SI(7).*t_SI.^4+v_cof_SI(8).*t_SI.
.^3+v_cof_SI(9).*t_SI.^2+v_cof_SI(10).*t_SI+v_cof_SI(11));
%v_mSI=(A_cb_SI./m_prj_SI).*(v_cof_SI(1).*t_SI.^10+v_cof_SI(2).*t_SI.^9+v_cof_SI(3).*t_SI.^8+v_co
f_SI(4).*t_SI.^7+v_cof_SI(5).*t_SI.^6+v_cof_SI(6).*t_SI.^5+v_cof_SI(7).*t_SI.^4+v_cof_SI(8).*t_SI
.^3+v_cof_SI(9).*t_SI.^2+v_cof_SI(10).*t_SI+v_cof_SI(11));

% - - - Position - - - -%
x_cof_SI=sym2poly(int(int(P_sym_SI,t),t)); % Position coefficients
x_SI=(A_cb_SI./m_prch_SI).*(x_cof_SI(1).*t_SI.^11+x_cof_SI(2).*t_SI.^10+x_cof_SI(3).*t_SI.^9+x_co
f_SI(4).*t_SI.^8+x_cof_SI(5).*t_SI.^7+x_cof_SI(6).*t_SI.^6+x_cof_SI(7).*t_SI.^5+x_cof_SI(8).*t_SI
.^4+x_cof_SI(9).*t_SI.^3+x_cof_SI(10).*t_SI.^2+x_cof_SI(11).*t_SI+x_cof_SI(12));
%x_mSI=(A_cb_SI./m_prj_SI).*(x_cof_SI(1).*t_SI.^11+x_cof_SI(2).*t_SI.^10+x_cof_SI(3).*t_SI.^9+x_c
of_SI(4).*t_SI.^8+x_cof_SI(5).*t_SI.^7+x_cof_SI(6).*t_SI.^6+x_cof_SI(7).*t_SI.^5+x_cof_SI(8).*t_S
I.^4+x_cof_SI(9).*t_SI.^3+x_cof_SI(10).*t_SI.^2+x_cof_SI(11).*t_SI+x_cof_SI(12));

% THERMODYNAMICS FROM CURVE FIT PRESSURE
V_gas_SI=V_gas_i_SI+x_SI.*A_cb_SI; % Volume available to combustion gas behind projectile
T_gas_calc_SI=Tf_ch_SI-((m_prch_SI(2:end).*v_SI(2:end).^2)./(2.*N_ch_SI(2:end).*Cv_gas_SI)); %
Calculates Temperature of gas behind projectile !!!VALUE FOR INDEX 1 WILL BE NAN - NEED TO
ACCOUNT FOR IN NEXT STEP
T_gas_SI=[Tf_ch_SI T_gas_calc_SI];
c_SI=sqrt(gamma_ch_SI.*R_gas_SI.*T_gas_SI);

% HELMHOLTZ RESONANCE
f_helm_SI=c_SI(num_i_SI)/(2*pi*L_cb_SI)

% GENERAL RESULTS
v_muz_SI=v_SI(num_i_SI)

% GENERATION OF PLOTS
figure
plot(t_exp_SI,P_amb_SI+P_exp_SI,'red',t_SI,P_SI,'blue',t_SI(TEST6_SI),P_SI(TEST6_SI),'g*')
legend('Archived Pressure Data','Curve Fit Pressure','Propellant Burnout')
%title('Pressure vs. Time (SI Units)');
xlabel('Time (s)')
ylabel('Pressure (Pa)')

figure
plot(t_SI,x_SI,t_SI(TEST6_SI),x_SI(TEST6_SI),'g*')
legend('Position','Propellant Burnout','Location','NorthWest')
%title('Position vs. Time (SI Units)')
xlabel('Time (s)')
ylabel('Position (m)')

figure
plot(t_SI,v_SI,t_SI(TEST6_SI),v_SI(TEST6_SI),'g*')
legend('Velocity','Propellant Burnout','Location','NorthWest')
%title('Velocity vs. Time (SI Units)')
xlabel('Time (s)')
ylabel('Velocity (m/s)')

figure
plot(t_SI,a_SI,t_SI(TEST6_SI),a_SI(TEST6_SI),'g*')
legend('Acceleration','Propellant Burnout')
%title('Acceleration vs. Time (SI Units)')
xlabel('Time (s)')
ylabel('Acceleration (m/s^2)')

figure
plot(t_SI,V_gas_SI,t_SI(TEST6_SI),V_gas_SI(TEST6_SI),'g*')
legend('Volume Behind Projectile','Propellant Burnout','Location','NorthWest')
%title('Volume Behind Projectile vs. Time (SI Units)')
xlabel('Time (s)')
ylabel('Volume (m^3)')

figure
plot(t_SI,T_gas_SI,t_SI(TEST6_SI),T_gas_SI(TEST6_SI),'g*')
legend('Gas Temperature Behind Projectile','Propellant Burnout')
%title('Gas Temperature Behind Projectile vs. Time (SI Units)')
xlabel('Time (s)')

```

```

ylabel('Temperature (K)')

figure
plot(t_SI,m_ch_SI,t_SI(TEST6_SI),m_ch_SI(TEST6_SI),'g*')
legend('Charge Mass','Propellant Burnout')
%title('Charge Mass vs. Time (SI Units)')
xlabel('Time (s)')
ylabel('Mass (kg)')

figure
plot(t_SI(1:TEST6_SI),m_fraction_SI(1:TEST6_SI))
%title('Mass Fraction of Propellant Burnt vs. Time (SI Units)')
xlabel('Time (s)')
ylabel('Mass Fraction Burnt')

figure
plot(t_SI,c_SI)
%title('Local Speed of Sound vs. Time (SI Units)')
xlabel('Time (s)')
ylabel('Speed of Sound (m/s)')

```

Appendix C – Flowcharts of Computational Procedures

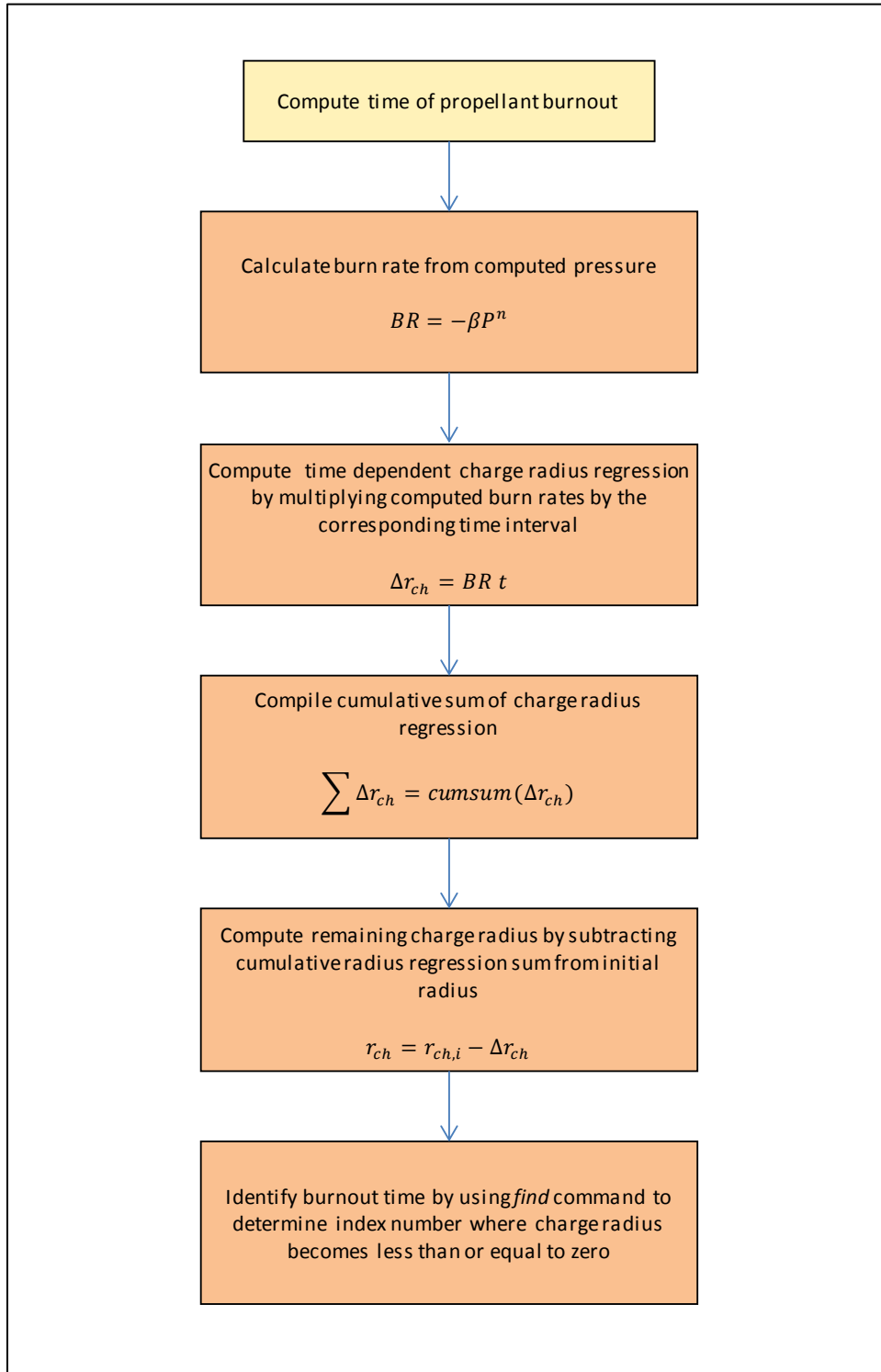


Figure C1 – Flowchart of Propellant Burnout Time Computation

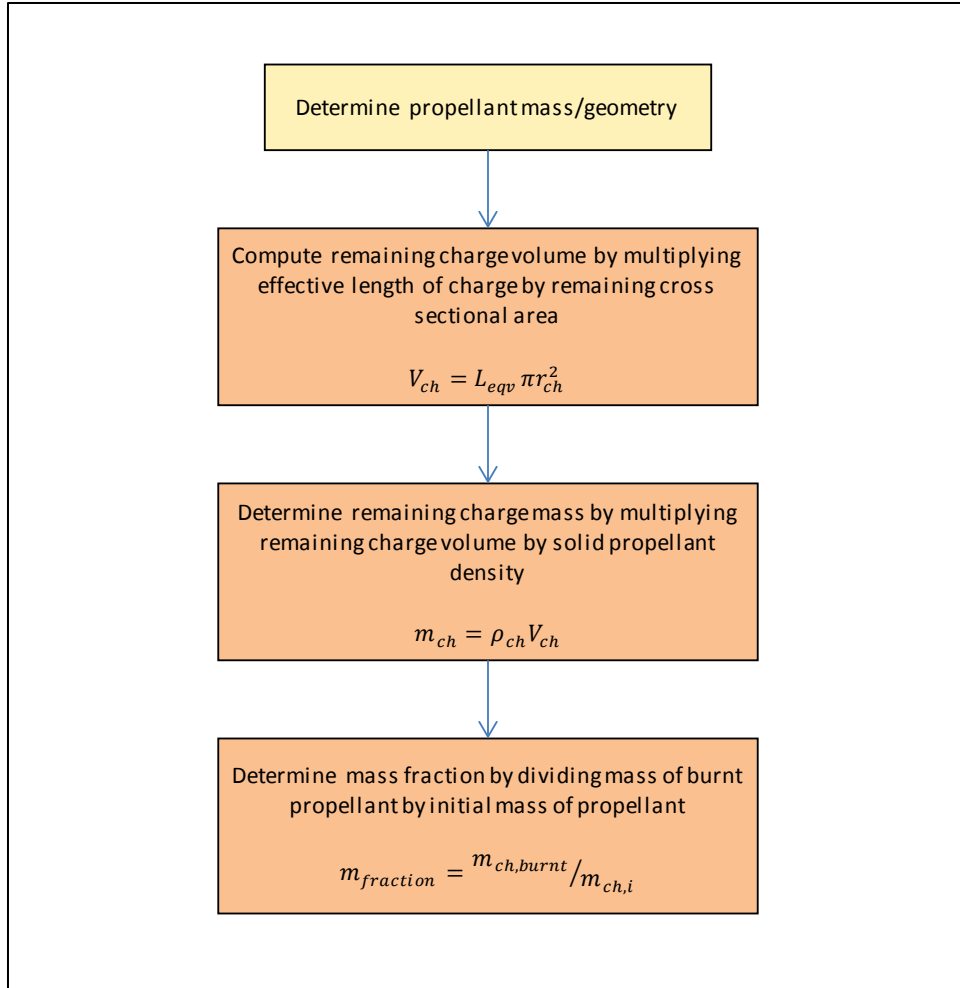


Figure C2 – Flowchart of Propellant Mass/Geometry Computations

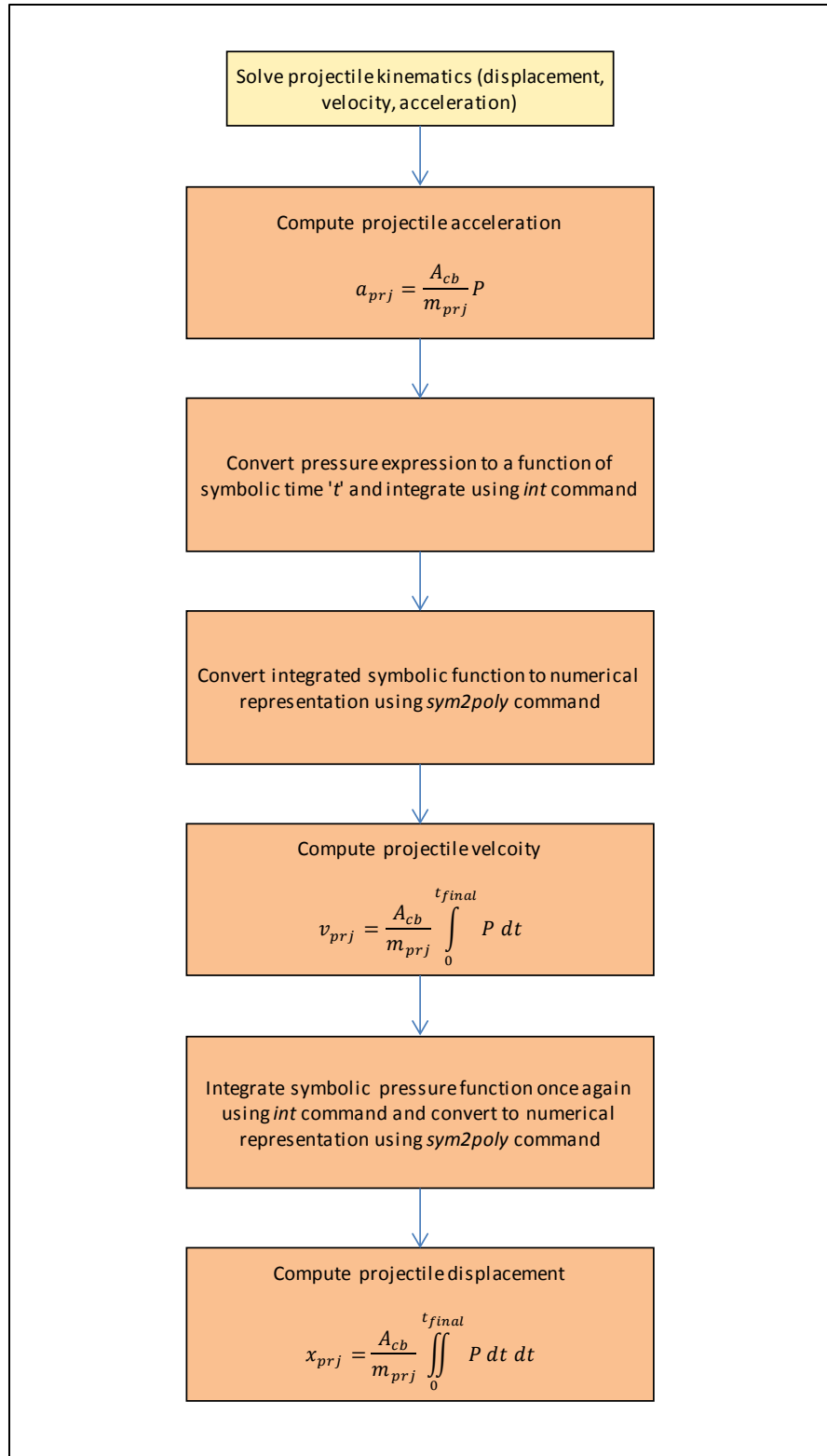


Figure C3 – Flowchart of Kinematic Computations

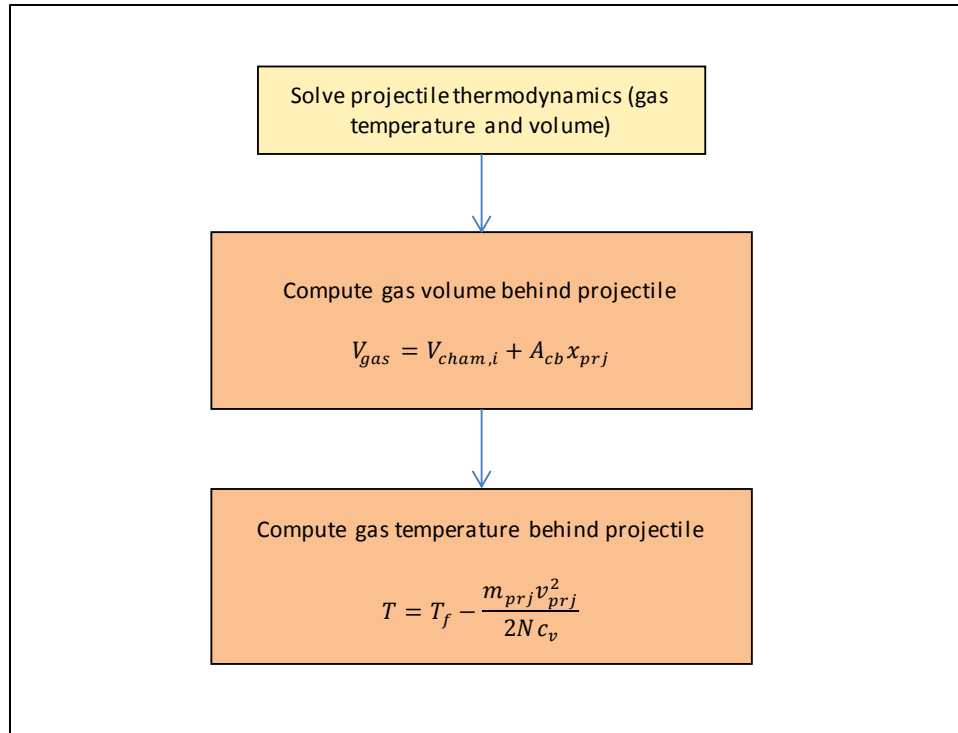


Figure C4 – Flowchart of Thermodynamic Computations

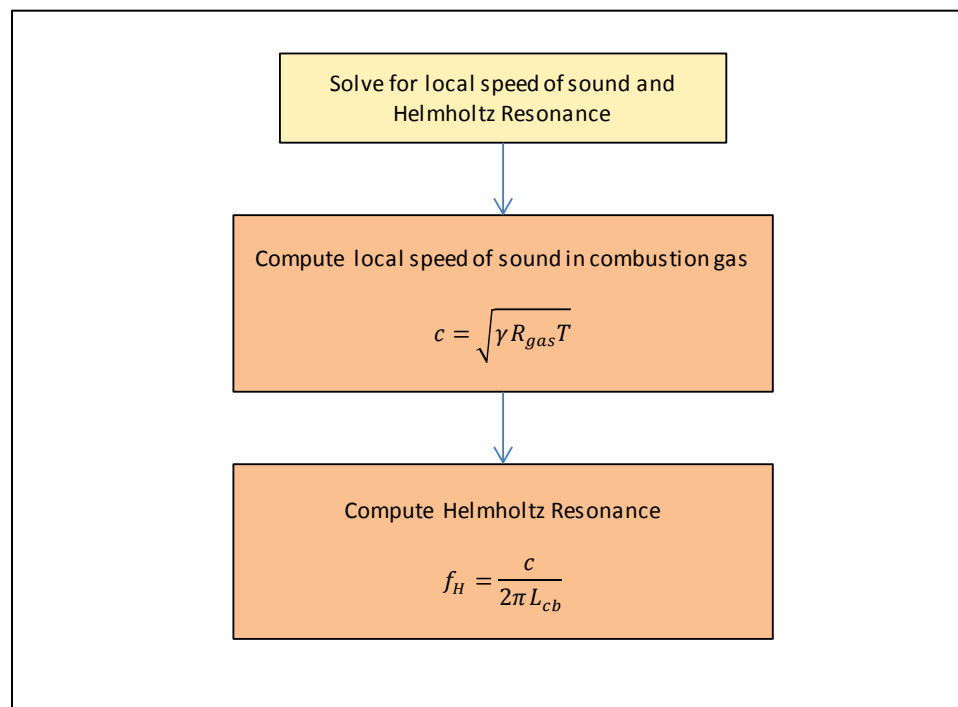


Figure C5 – Flowchart of Helmholtz Resonance Computation