Numerical Comparison of Muzzle Blast Loaded Structure

Xavier A. Quinn

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Aerospace Engineering

Rakesh K. Kapania, Chair
Mayuresh J. Patil
Joseph A. Schetz

February 18, 2022
Blacksburg, Virginia

Keywords: blast scaling, digital engineering, finite element analysis, modeling and simulation, muzzle blast, vehicle structures

Numerical Comparison of Muzzle Blast Loaded Structure

Xavier A. Quinn

(ABSTRACT)

Modeling and simulation have played an essential role in understanding the effects of blast waves on structures. However, a broad area of engineering problems, such as vehicle structures, buildings, bridges, or even the human body, can benefit by accurately predicting the response to blasts with little need for test or event data. This thesis reviews fundamental concepts of blast waves and explosives and discusses research in blast scaling. Blast scaling is a method that reduces the computational costs associated with modeling blasts by using empirical data and numerically calculating blast field parameters over time for various types and sizes of explosives. This computational efficiency is critical in studying blast waves' near and far-field effects. This thesis also reviews research to differentiate between free-air blasts and gun muzzle blasts and the progress of modeling the muzzle blast-structure interaction. The main focus of this thesis covers an investigation of different numerical and analytical solutions to a simple aerospace structure subjected to blast pressure. The thesis finally presents a tool that creates finite element loads utilizing muzzle blast scaling methods. This tool reduces modeling complexity and the need for multiple domains such as coupled computational fluid dynamics and finite element models by coupling blast scaling methods to a finite element model.

Numerical Comparison of Muzzle Blast Loaded Structure

Xavier A. Quinn

(GENERAL AUDIENCE ABSTRACT)

Numerical integration methods have helped solve many complex problems in engineering and science due to their ability to solve non-linear equations that describe many phenomena. These methods are beneficial because of how well they lend to programming into a computer, and their impact has grown with the increases in computing power. In this thesis, "modeling and simulation" refers to the characterization and prediction of an event's outcome through the use of computers and numerical techniques. Modeling and simulation play important roles in studying the effects of blast waves in many areas of engineering research such as aerospace, biomedical, naval, and civil. Their capability to predict the outcome of the interaction of a blast wave to vehicle structures, buildings, bridges, or even the human body while requiring limited experimental data has the chance to benefit a wide area of engineering problems. This thesis reviews fundamental concepts of blast waves, explosives, and research that has applied blast loading in modeling and simulation. This thesis describes the complexity of modeling an axially symmetric blast wave interaction by comparing the numerical and theoretical response blast loaded structure.

Dedication

I would like to dedicate this thesis to the men and women who serve in military alongside or operating weapon systems with substantial muzzle blast.

Acknowledgments

I would like to acknowledge the colleagues at the Naval Sufrace Warfare Center Dahlgren Division who have knowingly or unknowingly had a large impact on my unfolding interest and progressing knowledge of weapon system development, testing, and integration. I would specifically like the thank Dr. John Yagla for his encouragement and generous attitude, and for volunteering his time to pass on his experience and knowledge on the subject of experimental blast testing. I would also like to acknowledge my thesis advisor, Dr. Rakesh Kapania, for his patience and persistence in helping me achieve a new level of understanding and contribute to an already deep body of knowledge. I would also like to thank all of the committee members for their help and guidance. Finally, I would like to acknowledge and thank my wife and children for their tolerance and encouragement as I pursued my goals.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Blasts are an important load case for engineers to consider in many different areas. Explosive blasts have the capability to quickly impart large amounts of force and cause wide spread damage. Blasts are not only caused in the context of war, and can be used purposefully for many purposes or can be unanticipated accidents causing far-reaching destruction. Texas City, a port near Galveston, saw one of the worst examples of an unexpected explosion and experienced first hand the "worst industrial disaster in United States history." [32]

Two docked ships carrying ammonium nitrate fertilizer exploded on April 16th and 17th in the port of Texas City causing an estimated $50 to $70 million dollars of property damage in 1947 values [32]. The explosions threw debris as far as one mile, and the blast overpressure shattered windows and collapsed buildings, leaving 2,500 people homeless, causing over 3,000 injuries, and killing 600 [32]. More recent and similar examples include an explosion in Toulouse in 2001 and in the Port of Beirut in 2020 [31]. Both events were explosions of large amounts of ammonium nitrate, and video of the Beirut port explosion was captured from many sources due to a foregoing fire that had captured attention [31]. The explosion in Beirut, Lebanon caused huge amounts of damage and many casualties, leaving a crater with a diameter of 460 feet [23]. BBC News [23] reported that blast overpressure destroyed windows as far as 5 miles away from the blast center, and could be heard nearly 125 miles away in Cyprus. Stennett et al. [31] used calculators developed from blast scaling to estimate blast wave parameters such as the time of arrival, and predict that the explosion was the equivalent to roughly 600,000 kilograms of TNT.

Explosions are one cause of blasts, but blasts can come from shock waves caused by aircraft, gun systems, or instant release of pressure vessels. Engineers have many considerations when integrating gun weapon systems into aircraft. Some significant factors are the recoil loads transferred into the airframe and the muzzle blast's interaction with nearby aircraft structure areas. Understanding gun blast loads and their interaction with surrounding structures is critical to successfully integrating gun systems into lightweight vehicles. Engineers use and develop tools that highlight adjustable weapon system parameters to satisfy integration and mission requirements simultaneously.

Modeling and simulation can significantly improve prototyping cycles and reduce testing costs for engineers who integrate gun weapon systems onto aircraft. In addition, design studies, optimization, and life cycle analyses such as airframe fatigue due to gunfire are all areas that modeling and simulation can help improve the overall integration of gun systems on aircraft. This thesis presents blast analysis research and the different modeling and simulation techniques used, such as blast scaling, blast-structure interaction, and muzzle blast prediction. The complex nature of a blast event requires a multi-physics approach and the efficient interaction between computational fluid dynamics and finite element analysis to model it at the highest levels of fidelity. Blast scaling is a technique that can reduce some of that complexity in modeling.

Researchers have conducted comparisons of analytical and numerical techniques that predict structural responses to muzzle blasts. This thesis revisits a similar comparison due to the advancement of computational power and non-linear numerical techniques. In addition, this thesis discusses the limitations to the simplifications used in comparing various solutions and aims to contribute to computational mechanics by investigating solution methods for near-field muzzle blast loading on a finite element structural model. Finally, this thesis presents a new computer code that generates discrete gun blast loads based on a finite element model.

1.1 Overview of Blast Waves

Explosions are one example of an immense and quick release of energy that incites blast waves [4]. Blast waves consist of a shock wave and a blast wind blowing in the direction opposite to that of the shock wave characterized by negative pressure towards the point of explosion. A blast wave can be considered a fast traveling non-linearity caused by a relatively large and short pressure gradient, reducing in magnitude as it travels farther from its origin.

Mathematically, modeling the blast of an explosion has been done under essential assumptions such as; the source of high pressure being produced instantaneously at a point, the interior of the blast is homogeneous, and the blast takes a generally spherical shape. These approximations come from John von Neumann's discussion of characterizing the trajectory of a blast-induced shock wave and the continuous flow of air behind the shock wave. Von Neumann's discussion proposes more simplifications for blasts on a vast scale, like nuclear explosions. This proposal encouraged further investigations into the governing laws of blast wave decay due to a point explosion [35].

Sir Geoffrey (G. I.) Taylor was investigating the possibility of analytic solutions to the point source problem when an academic cohort presented a paper on sound pulse diffraction [6]. Blast waves are most often described by their pressure-time history, which Friedlander described mathematically in 1946 by equation 1.1, and shown in figure 1.1 [10]. In equation 1.1, the non-dimensional parameter z replaces time with a relation using the speed of sound c and a "pulse thickness" parameter, λ, as shown in equation 1.2 [10]. Freidlander [10] notes that the pulse thickness is a unit length chosen to non-dimensionalize the spatial aspects of his mathematical solution.

$$p_0(z) = (1 - z)e^{-z} \tag{1.1}$$

$$z = \frac{ct}{\lambda} \tag{1.2}$$

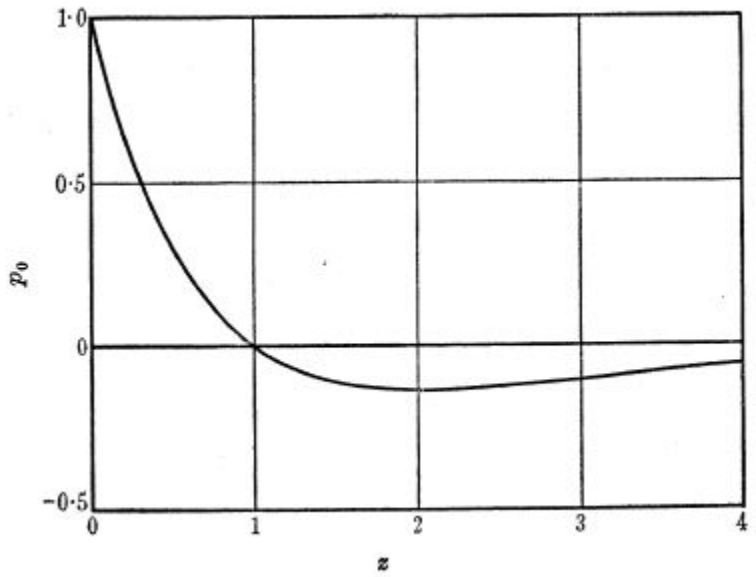

Figure 1.1: Non-dimensional pulse wave described by Equation 1.1 [10].

More recent research has used the modified Friedlander's equation, especially in curve fitting experimental data, to describe the positive phase of blasts and impulse [4].

$$p(t) = p_0 + P_s^+ \left(1 - \frac{t}{T^+}\right) e^{-b\frac{t}{T^+}} \tag{1.3}$$

Some critical parameters of the pressure-time history are peak incident (side-on) overpressure P_s^+, time of arrival T_a, and positive phase duration T^+. These parameters can be seen in figure 1.2; where arrow "A" denotes the time of arrival, arrow "B" points to the peak overpressure, and arrow "C" points to the beginning of the negative phase. The positive phase is the time between point "A" and "C," and the area under this section of the curve is the positive duration impulse. The modified Friedlander equation also contains a decay coefficient, b, that can be used as a fitting parameter along with peak incident

overpressure and positive phase duration. Side-on, or incident, overpressure is the blast wave's pressure above ambient, measured on a surface parallel to the instantaneous direction of travel. Generally, the negative phase of the blast is much smaller in magnitude and not always of concern in studies. Equations that describe the negative phase have been published, but they are often not implemented [4].

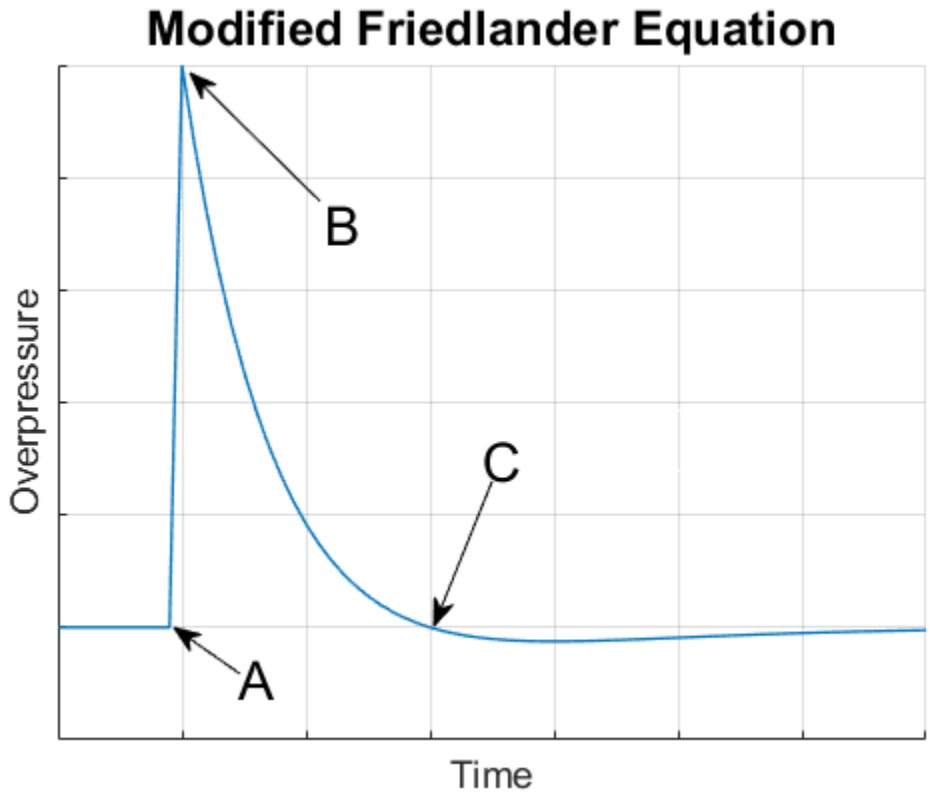

Figure 1.2: Non-dimensional pulse wave described by Equation 1.3 [10].

A blast's impulse describes the kinetic energy in a blast wave. The impulse describes a physical ability to impart a force on a body over some time [4]. The energy of different blasts can be compared using their impulses. The positive duration impulse can be found by:

$$I_s^+ = \int_{t_a}^{t_a+T^+} [p(t) - p_0]\, dt \tag{1.4}$$

1.2 Blast scaling

Blast scaling is a technique for modeling blasts of varying energy. Blast scaling calibrates the blast parameters and calculates a pressure history for a location based on its distance from the blast center. The blast parameters are generally determined using empirical data collected from tests. Two standard methods for blast scaling are the Hopkinson and Sachs methods. The Hopkinson method is based on a dimensional scaled distance. In contrast, the more general Sachs method states that pressure, time, impulse, and other blast parameters can be scaled by a dimensionless scaled distance [4]. In addition, the Sachs method is consistent at atmospheric pressures different from those at which the empirical data was captured [4].

It is essential to have a system that can classify different explosives by their output. Chapman-Jouguet pressure, detonation velocity, or heat of detonation are three analytical quantities that are used for this classification in the domain of explosives engineering [5]. In addition, explosives engineering uses three standard tests to measure empirical explosive output as percent TNT equivalent. These tests are the ballistic mortar test, the Trauzl test, and the plate dent test [5].

Table 1.1: Explosives and their TNT Equivalent by Test Method [5]

Explosive	Ballistic Mortar (% TNT)	Trauzl (% TNT)	Plate Dent (% TNT)
Comp C-3	126	117	118
RDX	150	157	135
EDNA C-3	139	122	122
Nitroguanidine C-3	104	101	95
PETN	145	173	129

1.3 Muzzle Blast Scaling

Gun firings have a unique signature that consists of a primary blast wave caused by the uncorking of the projectile and rapid release of air and propellant gases, and a secondary blast caused by muzzle flash which can be stronger than the primary blast [18]. Klingenberg and Heimerl [18] defines gun muzzle flash "as the radiation induced at various locations inside the propellant gas plume." This muzzle flash is caused by under-expanded gases leaving the muzzle and passing through shock fronts causing sudden deceleration, compression, and reheating causing after-burning and secondary blasts [18]. Muzzle blast waves are different than free air explosion blast waves because of their axial symmetry in the near to medium field. The gun system has many additional factors that influence the magnitude of the blast parameters, such as; barrel length, caliber, and propellant properties. These are some of the main characteristics that differentiate gun blasts from the assumptions made in the previous discussion of blast scaling for explosive charges.

Hopkinson's scaling was first applied to the muzzle blast problem by Reynolds and the U.S. Navy in the 1940's [36]. These works utilize additional geometric properties of the guns to correlate the overpressures of two different guns [36]. Other work represents muzzle blast fields with equivalent weight spherical explosive charge located at a distance from the muzzle along the bore axis of the gun [36]. However, this method has a specific shortfall by ignoring energy losses due to factors such as temperature [36].

Westine [36] proposes a Hopkinson scaling-based procedure, which was developed for scaling conventional explosive blast waves at sea-level conditions. In this method, the Friedlander equation is fit to experimental data to determine the shape of the transient load that can be applied to a structural element. This method non-dimensionalizes the data and expresses Hopkinson's law with the gun barrel length. The energy in the propellant charge, E, the mass of the projectile, M, and the velocity of the projectile, V approximate the blast's

energy in Equation 1.5.

$$W = E - \frac{1}{2}MV^2 \tag{1.5}$$

There are many simplifications made in the analysis of this method compared to experimental data, but Westine's work [36] produced a good correlation between the experimental and analytical model for a geometrically diverse sample of weapons. This work makes a significant assumption that the shock is "sufficiently weak so that the ratio of specific heats in air may be considered a constant" [36]. Westine's work [36] suffers from a scarcity of data on the specific impetus of various propellants and uses a constant value for all propellants. It is also worth noting that this scaling method does not consider the secondary flash over-pressure. This research is the cornerstone for the following muzzle blast scaling research that is covered in the literature review.

Chapter 2

Review of Literature

2.1 Review of Blast Scaling in FEA

Many finite element codes utilize blast scaling to analyze the mechanical response of blast-structure interaction. ConWep is one such code based on data and equations outlined in a U.S. Army technical manual, TM5-855-1 [13]. This routine assumes either a hemispherical charge shape for ground blasts or a spherical shape for air blasts [13]. Explosive weights of TNT form the basis for these calculations [13]. It uses an exponential decay for pressure-time history, as seen in Equation 2.1,

$$P(t) = P_{so}(1 - \frac{t - t_a}{t_d})e^{-A\frac{t-t_a}{t_d}} \tag{2.1}$$

where $P(t)$ is the pressure at time t, P_{so} is the peak incident overpressure at $t - t_a$, t_a is the time of arrival, t_d is the positive phase duration, and A is the decay coefficient [13]. This is also known as the modified Friedlander equation.

BLAST.F is a blast scaling routine developed by CHOCK and KAPANIA [3] and demonstrated its application to analyze structures modelled using NASTRAN. This program has two subroutines that calculate explosive blast pressure profiles. The first is a Hopkinson scaling method based on Kingery and Bulmash [17] Log-Log curve fitting. The second is a Sachs scaling-based method that uses scaled parameters from the U.S. Army Engineering Design Handbook, Explosions in Air [1]. CHOCK and KAPANIA [3] developed the code to

compare these two methods by outputting blast profiles based on user input loading areas. BLAST.F's output is agnostic to commercial finite element codes. CHOCK and KAPANIA [3] finds that the specific impulse of the blast profiles was the significant difference between the two methods when considering air blasts and hemispherical surface blasts.

CHOCK and KAPANIA [3] demonstrates the use of BLAST.F output in a finite element code, NASTRAN, by loading two different structures with BLAST.F output. The first structure simulated a portion of a ground vehicle's floor using a non-linear model of an aluminum plate parallel to the ground subjected to a hemispherical ground blast. CHOCK and KAPANIA [3] conducts a mesh convergence study with three simulations of the varying mesh size. The second model simulates a generic aircraft wing subjected to an air blast load generated by the Kingery and Bulmash routine and adjusted for the angle of incidence.

There is a deep body of research on the effects of blast loading using the commercial finite element solver LS-DYNA. LS-DYNA is a multi-physics code with many capabilities in modeling and simulating non-linear, highly dynamic events. The code originates from DYNA3D developed at Lawrence Livermore National Laboratory, which included explosive-structure interaction capability. As capabilities enhanced, the commercial version of the code, LS-DYNA, was released in 1989 [14]. This code's explicit nature and historical roots made it the primary focus for this thesis's review of research in modeling blast-structure interaction. In addition, LS-DYNA provides a specific blast scaling capability based on the ConWep empirical function called *LOAD_BLAST_ENHANCED [11].

LS-DYNA implements Multi-Material Arbitrary Lagrangian-Eulerian (MM-ALE) formulations that can be associated with equations of state to model fluids and explosive burn rates [14]. These element formulations allow scientists and engineers to model fluid-structure interaction, which provides a more direct capability for modeling blast-structure interaction. This technique involves explicit modeling of a fluid and explosive using arbitrary Lagrangian-Eulerian (ALE) element formulation with accompanying equations of state and

burn models. However, the strong discontinuous shock requires small ALE element sizes that drive the simulation's time step [29]. In addition, the distance between the origin of the blast may require large volumes of air to be modeled, which would become very computationally expensive [29].

Blast scaling reduces computational complexity by reducing a fluid-structure interaction model by applying the blast loads directly to Lagrangian element locations without the need to model the propagation of the blast wave in the fluid medium such as air [29]. However, the structural configuration can create a disadvantage to this method. For example, the *LOAD_BLAST_ENHANCED approach can shadow areas behind external corners and does not handle reflections or expansions of blast waves [27]. There have been many comparisons using LS-Dyna, and different techniques that work to avoid these issues while balancing the computational costs [27][29][33].

Slavik [29] demonstrates the benefits of a Blast Scaling-ALE coupling method over simple blast scaling. The model leverages work in Mullin and O'Toole [22] of a spherical charge of TNT located a short distance from the center of a Lagrangian plate. While in Mullin and O'Toole [22], the test is similar to the ballistic mortar test described above, Slavik [29] restricts the motion of the plate only in a direction normal to its surface, simulating a sled. Slavik [29] notes modeling challenges such as the angle of incidence varies across the plate and the absence of explosive particles impinging on the plate. Slavik [29] collects the pressure-time history at the center and corner of the plate to calculate the effective impulse, which is used with the plate's maximum velocity as primary response quantities [29].

Slavik [29] runs three simulations: one using the purely Lagrangian *LOAD_BLAST_ENHANCED method, one using an eighth-symmetric ALE air and explosive technique, and one using a coupled empirical *LOAD_BLAST_ENHANCED to ALE air interacting with the Lagrangian plate. The coupling simulation also exploits model symmetry. Slavik [29] conducts a mesh convergence study on the ALE air and explosive

simulation to ensure proper ALE element sizing. Table 2.1 shows the results of the three simulations. Slavik [29] notes that losses in energy due to advection in the ALE elements from artificial bulk viscosity and dissipation account for the lower plate velocities seen in the ALE models. The coupled approach has a larger difference in impulse than the ALE model at the center of the plate when compared to the response at the corner due to the shock traversing fewer ALE elements. This shows that the coupled model displays less attenuation of energy in the fluid elements. [29]

Table 2.1: Finite element results compared to empirical baseline model.[29]

	Plate Velocity (m/sec)	Center Impulse (Pa-sec)	Corner Impulse (Pa-sec)	ALE Element Count	CPU time (hours)
Coupled	94.5 (-6%)	2,123 (-4%)	296 (-9%)	393,300	1.6
ALE	91.1 (-10%)	1,958 (-12%)	294 (-10%)	807,300	3.4
Empirical	101	2,220	326	N/A	0.01

Schwer [27] expands on the work of Slavik [29], comparing the empirical ConWep method to coupled empirical-ALE methods in LS-DYNA utilizing different advection methods built into the ALE formulation to compare pressure-time histories in “ambient” and “air” elements. The ambient elements are a small layer of elements that communicate the empirical blast load to the rest of the ALE “air” domain. Slavik [29] concludes that the ALE air domain can have significant effects on the pressure-time history when the edges, faces, and the corners of the air domain can create reflections and expansions. The pressure histories close to the peak overpressure are sufficiently accurate for the ALE air elements with either Van Leer plus Half-Index-Shift or Donor Cell plus Half-Index-Shift. However, the ambient elements are most accurate with Van Leer plus Half-Index-Shift [27]. Schwer [27] also discusses the effects of grouping or not grouping the ALE elements.

Tabatabaei and Volz [33] pursues a similar methodology as Slavik [29] and Schwer [27], but utilizes experimental data as a basis for the comparison of empirical, ALE, and the coupled empirical-ALE methods. The model mimics a previous experimental test and exploits model symmetry for efficiency. The test and model consist of an explosive charge detonated above a steel-reinforced concrete simply supported panel. The study uses peak reflected pressure at the center of the concrete slab as the primary response quantity. The peak free field incident pressure is reported for the empirical model and experimental test but is at a distance that made the domain prohibitively large. Tabatabaei and Volz [33] reports substantial efficiency gains in computation time. The empirical approach underestimates peak free field incident pressure but overestimates the impulse. All methods underestimate the peak reflected pressure from the center of the plate. Some results are presented in Table 2.2 below.

Table 2.2: Concrete slab blast experiment compared to modeling results.[33]

	Peak Reflected (Mbar)	Peak Incident (Mbar)	No. of Elements	Total CPU time (hours)
Coupled	0.55E-3	N/A	12,930	41.70
ALE	0.71E-3	N/A	37,856	84.67
Empirical	0.29E-3	6.0E-6	4,179	2.23
Experiment	1.5E-3	7.3E-6	N/A	N/A

Gilson et al. [11] is another example of empirical-ALE coupling. This study focuses on how structural parameters influence a blast wave's reflected pressures and impulse. Gilson et al. [11] compare the blast interaction with elastic aluminum plate and a crush-able foam plate. [11]

Another example of blast modeling in LS-DYNA does not utilize empirical-ALE

coupling but presents a 2D to 3D mapping technique to reduce computation time. Kalra et al. [16] conducts parametric modeling studies in the 2D domain and 3D domain after mapping occurs. They also conducted a biomedical case study on an anatomical pig head to study pressure gradients in the brain caused by the blast. The case study finds a reasonable correlation with the data obtained from two experiments. The parametric studies conclude that mesh density, boundary conditions, constitutive models, and equations of state are critical to validating finite element blast simulations under this technique. [16]

2.2 Muzzle Blast Modeling

It is essential for weapon system designers to understand that the interaction of muzzle blasts with surrounding personnel and structures due to harsh transient loading [36]. Klingenberg and Heimerl [18] is a comprehensive and essential resource for understanding fundamental concepts of guns, muzzle blast, and flash. Klingenberg and Heimerl [18] also covers an exhaustive list of research, computational efforts, and experimental work, including common methods and measurement techniques. This book is critical in uncovering foundational works and is worthy of acknowledgment.

There is high variability in muzzle blast measurements that depend on many environmental and system parameters that are difficult to control. A great deal of work in modeling muzzle blast flows with numerical methods such as the finite-difference method has been conducted due to this experimental complexity. Ishiguro [15] simulates the two-dimensional unsteady inviscid flow of a perfect gas through a 30 degrees divergent nozzle. The problem is further simplified using an axis of symmetry along the nozzle's centerline. Moretti [21] relates the flow of muzzle blast to simplified cases of Prandtl-Meyer expansions of a flow-through channel. In this work, Moretti [21] covers the numerical treatment of the problem in steady quasi-one-dimensional, steady two-dimensional, and transient two-

dimensional forms using unique Eularian-based integration methods. Erdos and Del Guidice [7] furthers the line of reasoning set by Moretti [21], explaining that "the flow contained between the plume boundaries up to the Mach disk possesses the characteristics of a highly underexpanded supersonic exhaust jet," which can be modeled as a steady flow. Erdos and Del Guidice [7] conducts a comparative analysis of this supersonic plume model, blast wave theory with variable energy release, and unsteady shock-layer analysis, and validates the unsteady shock-layer model with experimental rifle data. Erdos and Del Guidice [7] deems the unsteady shock layer analysis to have excellent correlation to the experimental data. Taylor and Lin [34] use Gudnov's scheme with alternating direction implicit procedure to integrate and discretize the Euler equations of motion in their simulation of a muzzle flow. Taylor and Lin [34] compares the results of a rifle simulation to previously collected experimental data [34]. Taylor and Lin [34] conduct a different simulation of a mortar, with and without simulating the projectile, and find that the projectile significantly affected the blast origin's location.

Moore [20] show the capability of an Eulerian inviscid compressible two-dimensional hydrodynamic code, "SHELLTC," while investigating muzzle blast about a naval cannon. This report compares the simulation data to experimental pressure data. Moore [20] notes that the calculated velocity field did not have experimental data to compare against because most experiments only collect pressure fields [20]. However, Moore [20] argues that this is an added value of the simulation of the gun blast because it is not easily or generally measured in experimental testing. Moore [20] attributes the differences between the results to experimental data to the simplicity of the code and lack of projectile modeling. One unique characteristic of this investigation was that the entire gun barrel length was simulated with the grid, removing the need for assumptions regarding the change in fluid properties within the gun barrel over time [20].

These works are significant contributions to understanding the gas dynamics of

muzzle flow fields but are too computationally expensive to incorporate in a blast-structure interaction. Gun muzzle blasts are similar to free air explosions in that they exhibit similar phenomenon. The flow field about a gun muzzle during firing is a multi-phase, supersonic under-expanded jet, making it to be a transient flow that is very difficult to model [18]. A significant difference between free-air and muzzle blast fields is that the field is axi-symmetric about the bore of the gun, limiting reductions in model scope to quarter symmetry, rather than eighth symmetry for a spherical blast [20]. Blast scaling reduces the computational burden of a blast-structure interaction simulation, making muzzle blast scaling a critical extension of the work done with free-air spherical explosive blast scaling.

Fansler and Schmidt [9] developed a method of predicting the characteristics that describe a gun blast by parameterizing the weapon system to significant independent variables such as weapon geometry, emplacement, propellant, and projectile characteristics and launch conditions. Fansler and Schmidt [9] notes that there are many accurate numerical models of muzzle blast. However, his motivation is to create a model that requires less personnel and computer resources and can be helpful at the beginning of a gun's or structure's design cycle. The semi-empirical model is based on a power law and correlated to experimental data of bare muzzle guns to define scaling parameters with a least-squares fit. Fansler [8] further enhances his scaling models by comparing instantaneous energy release and constant energy deposition rate models. In addition, he finds that including gun-emptying and Mach disc location parameters improved data correlation [8].

2.3 Review of Muzzle Blast-Structure Interaction

Schmidt [26] led an experimental effort to investigate a 30mm chain-gun blast loading on the airframe of an attack helicopter. There are multiple experimental configurations of the gun, using various muzzle brakes to assess recoil reduction and blast-structure inter-

action. The experiments generate data on the interior ballistics of the gun and muzzle blast in the free field, and against a simulated aircraft surface, along with other measurements. Schmidt [26] compares the data with a scaling approach based on the energy rate of the blast from Fansler and Schmidt [9]. Schmidt [26] also cites excellent references on research in the area of muzzle blasts of moving guns [30][19].

Heaps et al. [12] works to extend the work of Fansler and Schmidt [9] by creating a computer code in FORTRAN that can generate field plots of the various blast parameters such as peak overpressure, time of arrival, and positive phase duration on a plane arbitrary to the orientation of the muzzle. The program created by Heaps et al. [12] includes an option for calculating reflected overpressure, lending it to calculating blast pressures on surfaces based on the angle of incidence and normal reflected shock theory.

Carson and Sahni [2] compares the muzzle-centered technique presented by Fansler and Schmidt [9] to their scaling law. Carson and Sahni [2] propose a scaling law that defines the polar angle from the blast center of the gun. Carson and Sahni [2] compare the models after tuning the parameters with numerical simulation data of a gun system. The approach is unique because it uses a numerical simulation rather than experimental data to tune the scaling law parameters. However, the bounding assumption of near field blast and the lack of variability in gun system parameters limits the comparison.

Singhal and Larson [28] compares dynamic reduction factors based on the cylindrical bending of flexible aerospace panels. They use many assumptions, such as a unique pressure history equation, linearized models, and constant spatial pressure distributions. This last assumption is plausible for only far-field muzzle blast. Singhal and Larson [28] uses three different approaches; finite element model using modal super-position and time integration to determine dynamic deflections, reducing the structure with an energy formulation, solving Duhamel's integral, and approximating a blast with a triangular pulse to calculate a load factor. Singhal and Larson [28] finds the triangular pulse method to be conservative

compared to the others. This thesis uses a similar approach to verify that the numerical modeling techniques would result in the best approximations of the time history response of an aerospace structure.

The following chapters present a numerical case study of various methods for calculating the response of a representative aerospace panel to a blast load and a mesh density study using a muzzle blast scaling tool. The comparison of analytical and numerical techniques show that the solution method can have a large impact on the response of a structure due to blast loading. The mesh convergence chapter shows the utility of a muzzle blast scaling tool coupled to a finite element solver, LS-DYNA. It also shows that the discretization of the blast pressure can be done using nodal point loads or element based distributed pressures with little impact to the results. Finally, a summary chapter draws on conclusions of the literature review, numerical case studies, and finite element model muzzle blast scaling tool, as well as discusses future research possibilities and applications of the tool in model based research.

Chapter 3

Numerical Modeling Verification

Many of the discussed research endeavors use modeling and computational simplifications out of necessity, while others conduct limited verification studies based primarily on mesh convergence. This thesis conducts a mesh convergence study and contributes to applied computational mechanics by comparing numerical solutions to a structural blast problem. One primary assumption made by most research using analytical methods is to ignore the negative phase of the blast pulse. This thesis does not ignore the negative phase of the blasts.

Oden and Bathe [25] define applied computational mechanics with an objective "to develop computational techniques that aid in the numerical analysis of various models of physical behavior and to use these analyses to design actual mechanical systems-structures, machines, processes, etc." Understanding the differences of the numerical solutions and how they compare to analytical solutions is very important to designing aerospace structures to withstand blast loading. This knowledge can also be applied to help determine the reasonability of solutions with more complex blast loading distributions such as near field gun muzzle blasts. This section compares the solutions of a simplified blast-loaded structure to understand the differences in results and the best method for this broad set of problems. This numerical experiment shows that the load application of a representative transient blast impulse in the finite element code is reasonably accurate, and more complex muzzle blast loads on larger structures will yield good responses within the applicable assumptions of specific modeling techniques.

This thesis conducts model verification by comparing the transient response of the central node in the finite element model to the transient response of an equivalent single degree of freedom system through analytical methods. This thesis compares two finite element models; a linear implicit and a non-linear explicit solution. The analytical method reduces the panel to a single degree of freedom system, solves Duhamel's integral for the initial conditions at the end of the blast loading, and calculates the free vibration after the end of blast loading. Peak pressures and blast impulse are varied to explore their effects on the results. The comparison investigates multiple parameters between the different solutions. These include comparing the FFT of the blast impulse to the FFT of the response and calculating the theoretical explicit time step with the Courant-Friedrichs-Lewy condition.

The structure is a simply supported panel that spans 36 inches, is 1 inch wide, and has one-tenth of an inch thickness. The calculations use linear elastic material models based on aluminum 6061. The table below shows the mechanical properties of the material.

Table 3.1: Linear Elastic Material Properties of Aluminum 6061

Density $\left[\frac{\text{lbf-sec}^{\,2}}{\text{in}^{\,4}}\right]$	Young's Modulus $\left[\frac{\text{lbf}}{\text{in}^2}\right]$	Poisson's Ratio
2.52E-04	9.701E+06	0.33

All models use a transverse pressure load that was constant across the length and width of the panel. The narrow width allows the assumption of constant pressure distribution in that direction to be reasonable. The transient pressure load for this comparison is calculated based on the following formula to allow the positive phase duration to be a controlled parameter rather than a calculated variable. The total duration is a value based on the positive phase duration and represented as $\alpha = \frac{1}{3.667\tau}$.

$$p(t) = \tilde{p}\left(1 - \frac{t}{\tau}\right)\left(1 - \frac{\alpha t}{\tau}\right)\left(1 - \frac{\alpha^2 t^2}{\tau^2}\right) \tag{3.1}$$

The numerical experiment compares results for three load cases. First, the blast impulse is held constant for two peak pressures by varying the positive phase duration. Second, the peak pressure will be varied with a constant positive phase duration, resulting in different impulses. Displacement and velocity comparisons use the central node at the mid-span of the panel. The velocity comparison is conducted at the end of the blast loading. The entire simulation time is used to compare overlays of the displacement histories. The final comparison is between the frequency content of the transient loads and displacement response at the mid-span. Figure 3.1 shows the blast pressure histories for the different numerical cases.

Table 3.2: Experimental Parameters

	Peak Pressure $\tilde{p}$ [psi]	Positive Phase Duration τ [s]	Impulse I [lbf-s]
Case 1	5	10E-04	-7.2561E-03
Case 2	2	25E-04	-7.2561E-03
Case 3	2	10E-04	-2.9025E-03

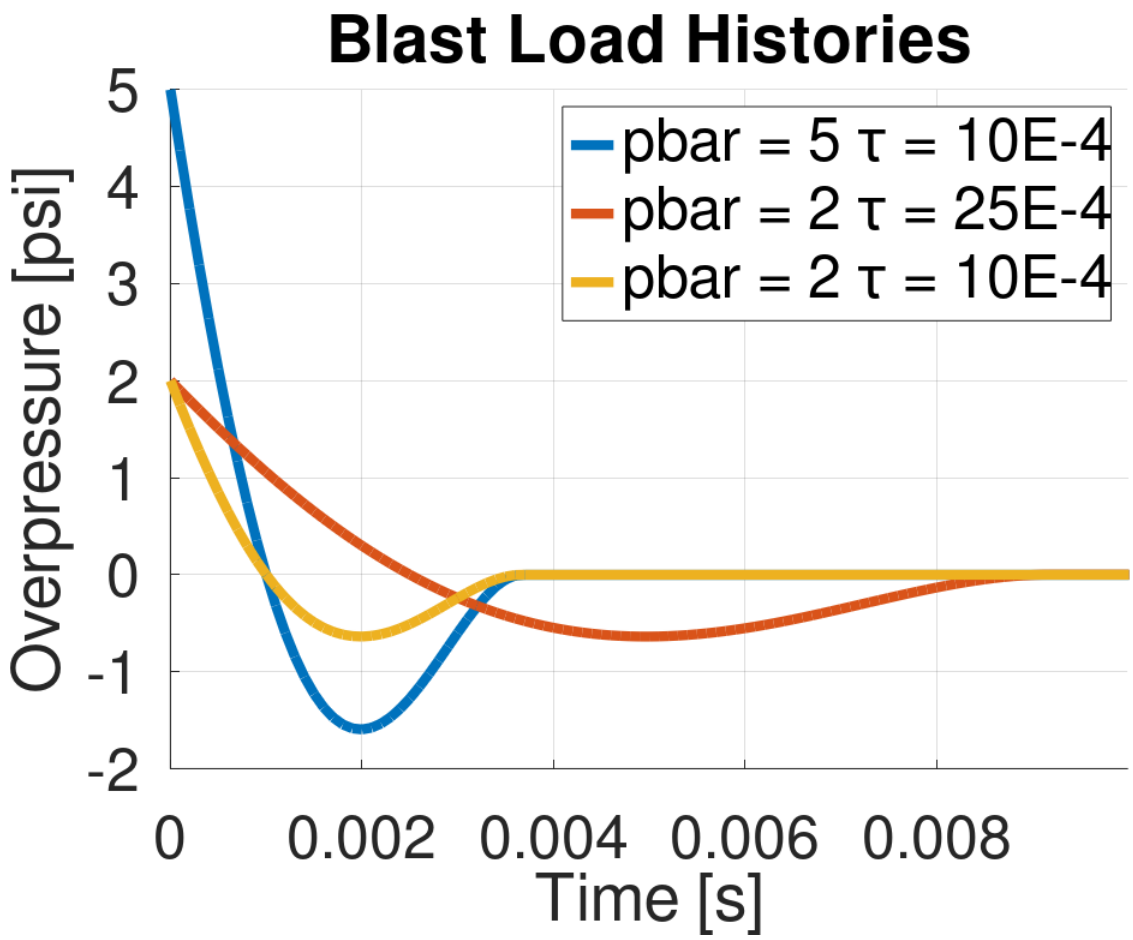

Figure 3.1: Comparison of various blast pressure histories.

3.1 Single Degree of Freedom System

The analytical solution reduces the structural panel to an undamped single degree of freedom system using the energy method. An equivalent stiffness and mass represent this single degree of freedom through the following integrals. Equations 3.4 and 3.5 represent the mode shape function and natural frequency, respectively.

$$M_e = \int_0^L m(x)\phi^2(x)dx = \frac{mL}{2} \tag{3.2}$$

$$K_e = EI \int_0^L \phi''(x)^2 dx = \frac{n^4 \pi^4 EI}{2L^3}, n = 1, 2, 3... \tag{3.3}$$

$$\phi(x) = sin\left(\frac{n\pi x}{L}\right), n = 1, 2, 3... \tag{3.4}$$

$$\omega_n = \sqrt{\frac{K_e}{M_e}} = n^2 \pi^2 \sqrt{\frac{EI}{mL^4}}, n = 1, 2, 3... \tag{3.5}$$

The following table shows the parameters of the structure's first ten modes.

Table 3.3: Energy Method Parameters of Structure

Mode	ω_n $(\frac{\text{rad}}{\text{s}})$	M_e $(\frac{\text{lbf-sec}^2}{\text{in}})$	K_e $(\frac{\text{lbf}}{\text{in}})$
1	45.071	0.0004662	9.47E-01
2	131.4	0.0004662	1.52E+01
3	257.6	0.0004662	7.67E+01
4	424.0	0.0004662	2.42E+02
5	630.67	0.0004662	5.92E+02
6	877.64	0.0004662	1.23E+03
7	1167.9	0.0004662	2.27E+03
8	1492.5	0.0004662	3.88E+03
9	1860.3	0.0004662	6.21E+03
10	2268.5	0.0004662	9.47E+03

The equivalent loading of the SDOF system is dependent on the system's mode, which is logical because an evenly distributed transverse pressure load across a beam would not excite antisymmetric mode shapes.

$$F_e = p(t) \int_0^L \phi(x) dx = p(t) \frac{2L}{n\pi}, n = 1, 2, 3, 4... \tag{3.6}$$

Duhamel's integral describes the single degree of freedom system's initial conditions at the end of the blast loading. The displacement history can then be determined after this point assuming free vibration.

3.1.1 Linear Implicit Model

The following section describes the finite element modeling techniques used to calculate the linear implicit transient response of the panel to the blast loads. Two-dimensional linear assumed strain C0 shell elements are used, with five through-thickness integration points. The elements are uniform squares, 1-inch long. The linear implicit model prescribes distributed pressure loads over each element. All pressure loads follow the same discrete pressure history curve, resulting in spatially constant distributed pressure. The finite element solutions run up to 0.14 seconds to capture at least one cycle of the lowest natural frequency. The displacement degrees of freedom are constrained at the ends, while the rotational degrees of freedom are set to be free. The aluminum is modeled as a linear elastic material, as noted above.

The Courant-Friedrichs-Lewy condition is a way to control the time step in small deformation theory based on the acoustic wave propagation through the material and the element dimensions. The method is shown below and calculated for the finite element model to compare with the actual solution time step used.

$$c = \sqrt{\frac{E}{\rho}} = 193,525 \quad \text{in/s}$$

$$\Delta t = \frac{L_{\text{element}}}{c} = 2.584\text{E-}06 \quad \text{s}$$

LS-Dyna uses a default scale factor of 0.9 on this value for conservatism. The values calculated above compare well with the explicit time step determined by the explicit solver

during the solution. The user's manual recommends reducing this scale factor to a lower limit of 0.67 for explosive loads. Linear implicit solutions are calculated using time steps of 2.0E-06 and 1.0E-06 and show that the results are the same for both step sizes. The linear implicit solutions use a constant time step of 1E-06 for the numerical experiments.

An implicit eigenvalue analysis of the model provides data to compare to the panel's energy method single degree of freedom approximation. The table below compares the natural frequencies of the single degree of freedom system to the modal analysis of the finite element model. The finite element model numbers do not align with the single degree of freedom system because there are bending modes in directions perpendicular to the loading and torsional modes. Visual inspection of the finite element model shapes easily rules out modes that do not apply to this comparison. These torsional or transverse modes would not be excited in a uniform pressure loading, and are not represented in the single degree of freedom system. The comparison of the natural frequencies for equivalent mode shapes is excellent between the single degree of freedom system and the finite element model.

Table 3.4: Modal Analysis Comparison

SDOF Mode	SDOF ω_n $(\frac{\text{rad}}{\text{s}})$	FEA Mode	FEA ω_n $(\frac{\text{rad}}{\text{s}})$	Δ
1	45.069	1	43.138	4.3%
2	180.28	2	172.612	4.3%
3	405.62	3	388.601	4.2%
4	721.1	4	691.399	4.1%
5	1126.7	6	1081.422	4.0%
6	1622.5	7	1559.200	3.9%
7	2208.4	9	2125.377	3.8%
8	2884.4	11	2780.713	3.6%
9	3650.6	13	3526.079	3.4%
10	4506.9	14	4362.459	3.2%

3.1.2 Nonlinear Explicit Model

Both finite element models use an offset of the pressure load's arrival to ensure solution stability at the beginning of the simulation. The blast arrival is offset by 1.0E-05 seconds to allow up to ten time-steps before the pressure begins increasing. The explicit model uses a non-linear, fully integrated two-dimensional shell element. The mesh, boundary conditions, material model, and pressure load are identical to the implicit model.

3.1.3 Structural Model Comparisons

The numerical experiment compares three models using three different metrics. Comparing the displacement and velocity at the end of the dynamic load, comparing the

displacement histories of the different systems, and comparing the frequency content of the systems' responses. The displacement and velocity of the central node of the finite element models at the end of the blast load is compared to the the initial conditions for free vibration response determined by Duhamel's Integral in the table below. The single degree of freedom (SDOF) system only uses the first natural frequency for this comparison. The displacements are reasonably similar for each case except for case 2. In case 2, the explicit model seems to have a much greater response to the negative phase pressure and has a positive displacement at the end of the blast, where the other two models are still negative displacements.

Table 3.5: Free Vibration Initial Conditions Comparison

Model	Case 1 Displacement [in]	Case 2 Displacement [in]	Case 3 Displacement [in]
SDOF Model	-0.18	-0.44	-0.07
Implicit FEA	-0.12	0.04	-0.05
Explicit FEA	-0.13	-0.57	-0.05
Model	Case 1 Velocity [in/s]	Case 2 Velocity [in/s]	Case 3 Velocity [in/s]
SDOF Model	10.8	15.7	4.3
Implicit FEA	16.4	-6.7	3.1
Explicit FEA	-26.7	157.3	6.6

Visualizing the displacement histories helps shed light on the differences in the velocity comparisons shown above. The figures below show the overlays of the displacement histories of the central node in the finite element models and the single degree of freedom system. The free vibration is calculated for modes 1, 3, and 5 of the single degree of freedom

system and super-positioned for the comparison.

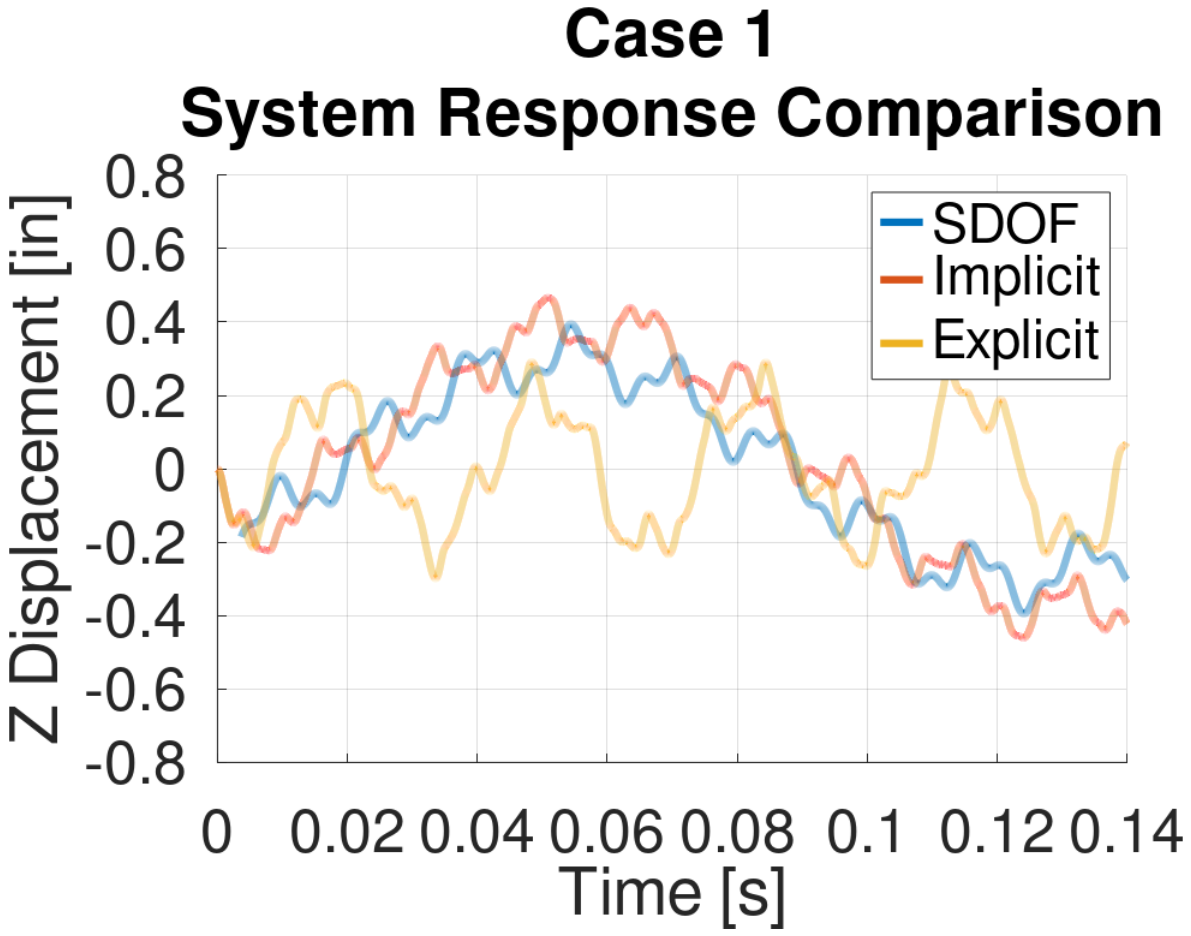

Figure 3.2: Comparison of displacement histories for blast 1.

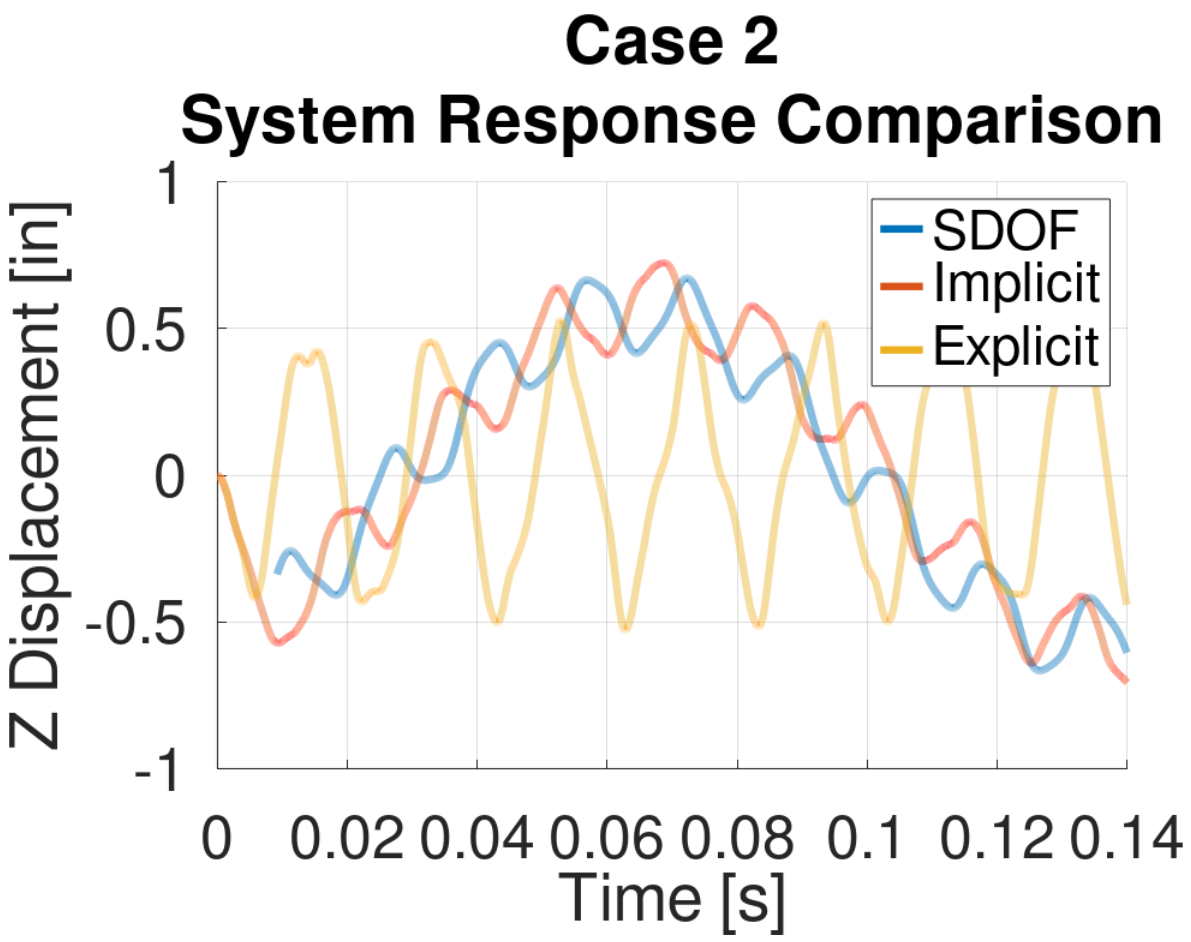

Figure 3.3: Comparison of displacement histories for blast 2.

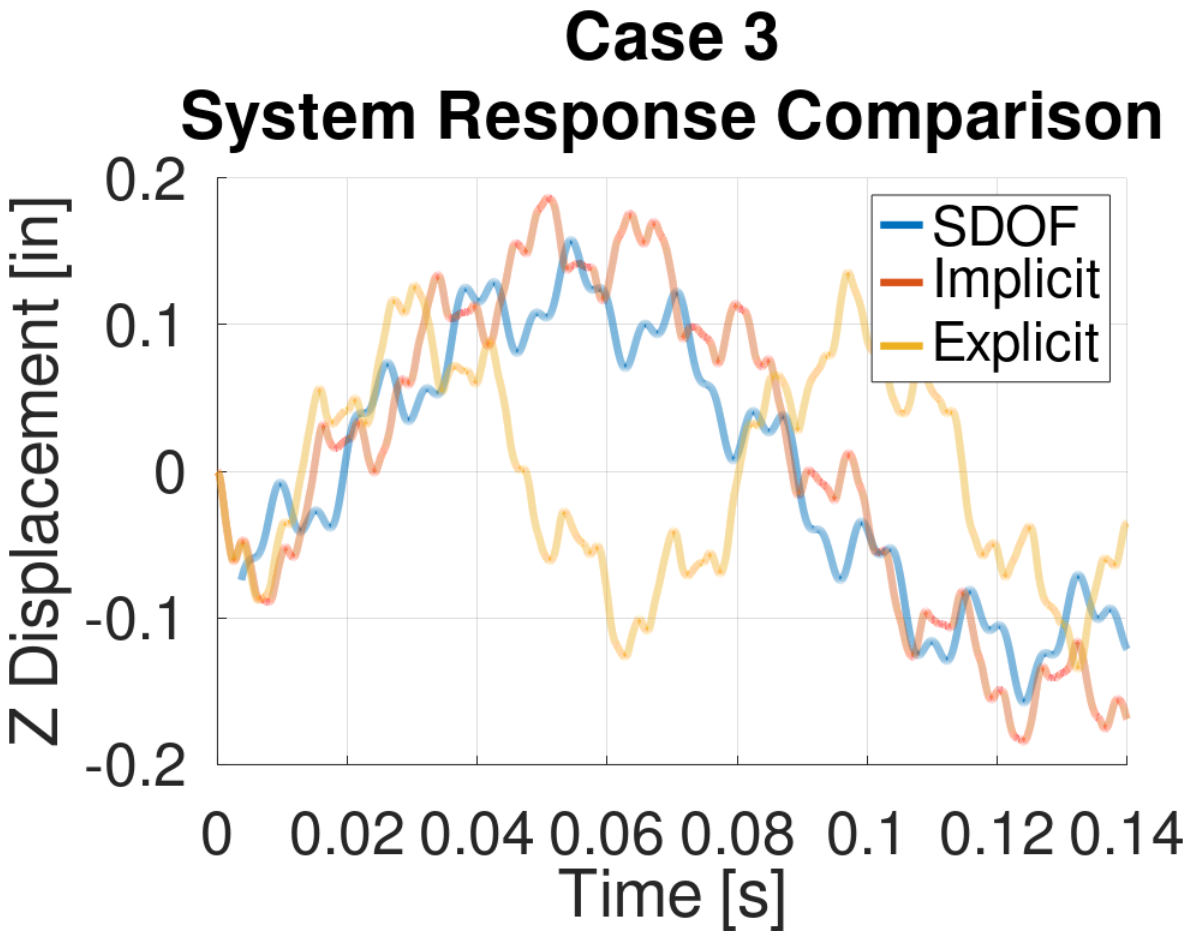

Figure 3.4: Comparison of displacement histories for blast 3.

The figures below show the displacement histories during the blast loading. These show that the implicit and explicit models behave similarly during the positive phase of the blast pressures, but rebound differently during the negative phase and at the end of the pressure load.

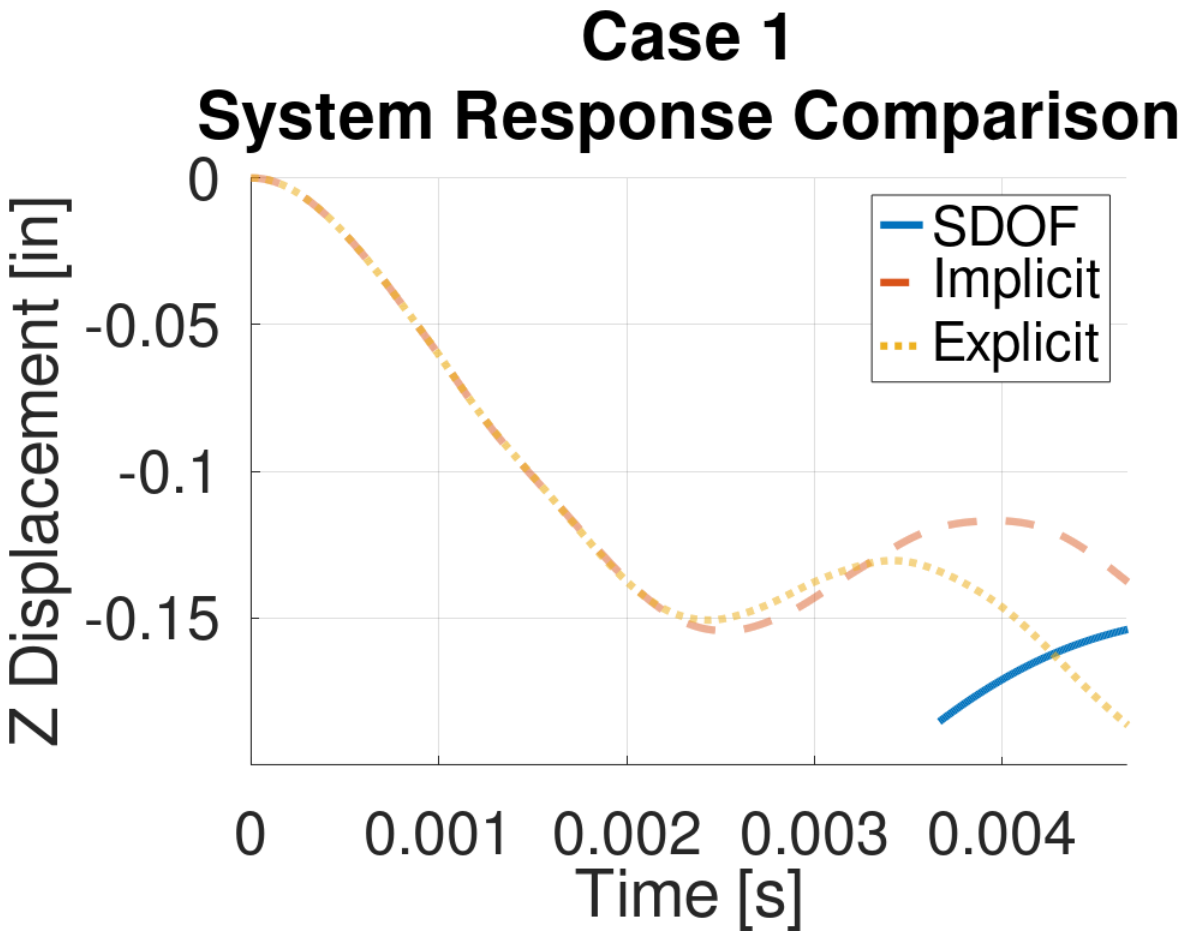

Figure 3.5: Comparison of displacement histories during blast loading for blast 1.

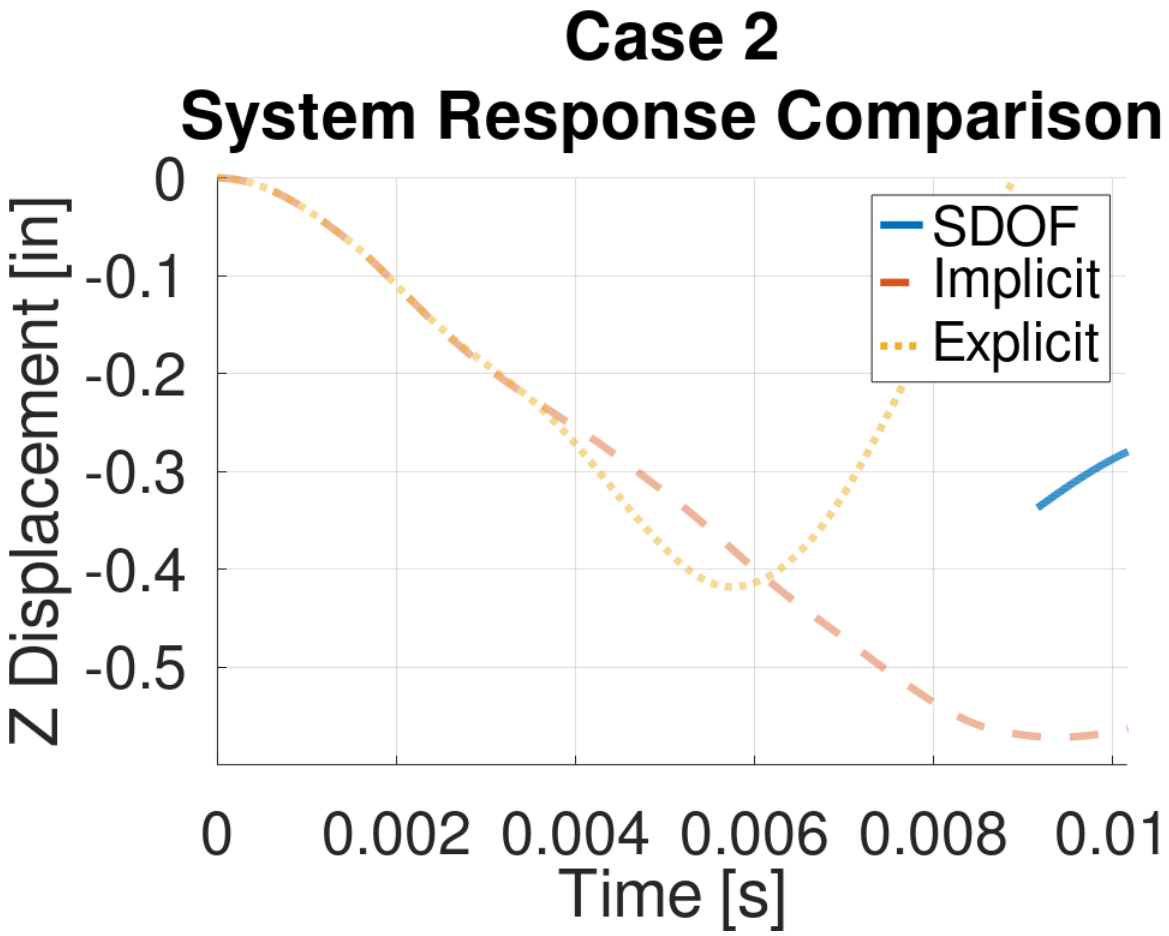

Figure 3.6: Comparison of displacement histories during blast loading for blast 2.

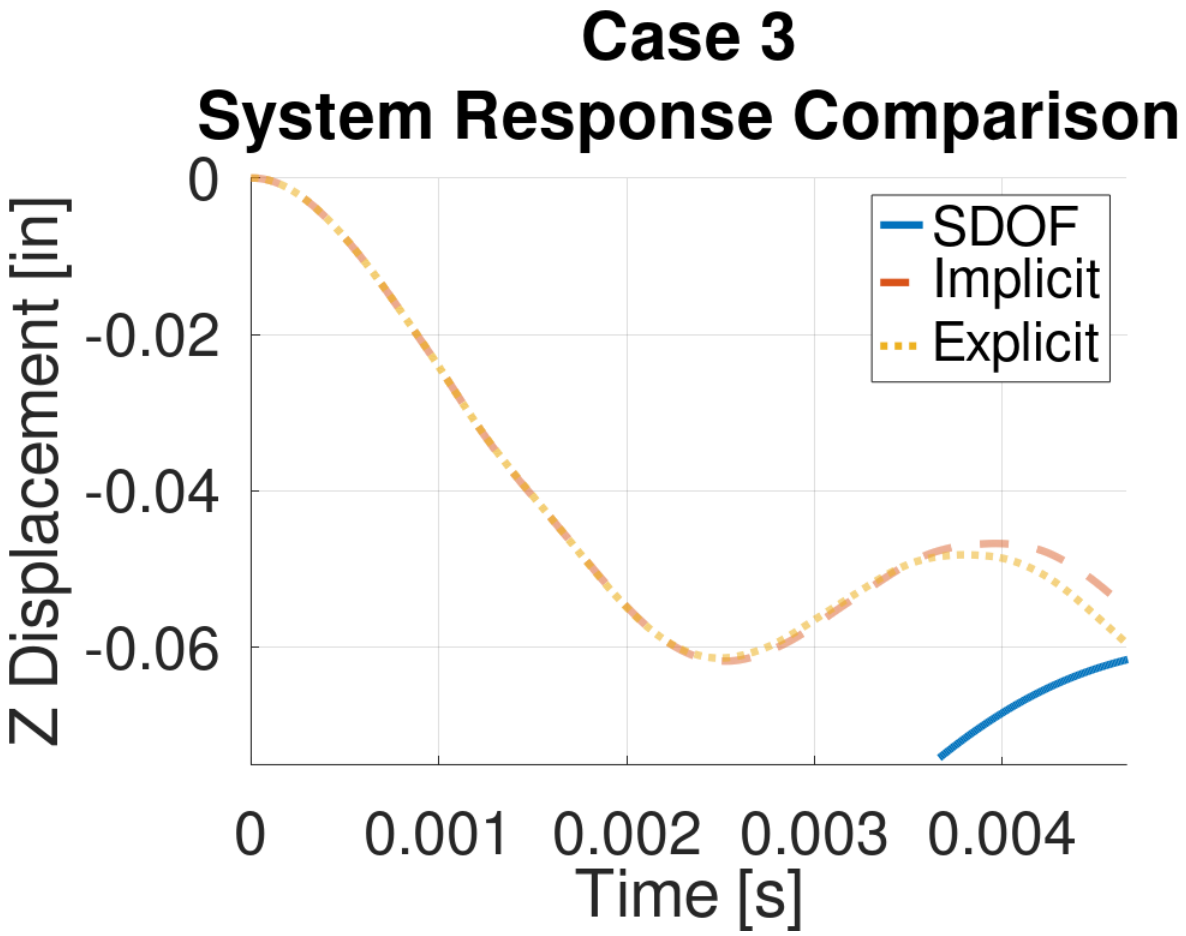

Figure 3.7: Comparison of displacement histories during blast loading for blast 3.

The implicit and analytical models have very similar peak displacements for all blasts. The explicit model shows slightly lower peak displacements across all blasts. The plots show a predominant excited frequency in the implicit and analytical model for all three blasts. Multiple frequencies are excited in the explicit model but different frequencies than the implicit and analytical. The displacement magnitude is highest for the implicit finite element model, while the analytical solution gives the smallest peak displacement. The lower peak displacement could be due to the higher predominant frequency content of the explicit response. Simulated blasts 1 and 2 give similar responses across the different models due to the similar impulse of the different blasts, while the smaller blast impulse gives smaller

displacements as expected.

Different periodic loads excite different natural frequencies of a structure. An impulse is a unique case because it excites continuous broadband of frequencies of the structure. Figure 3.8 below shows the FFT of the blast loads of the different experiments. The peak amplitudes are at 256.4 Hz, 106.4 Hz, and 256.4 Hz for experiments 1, 2, and 3. While blasts for experiments 1 and 3 have very different impulses and peak pressures, their positive phase duration is identical, resulting in the blasts' identical frequency content.

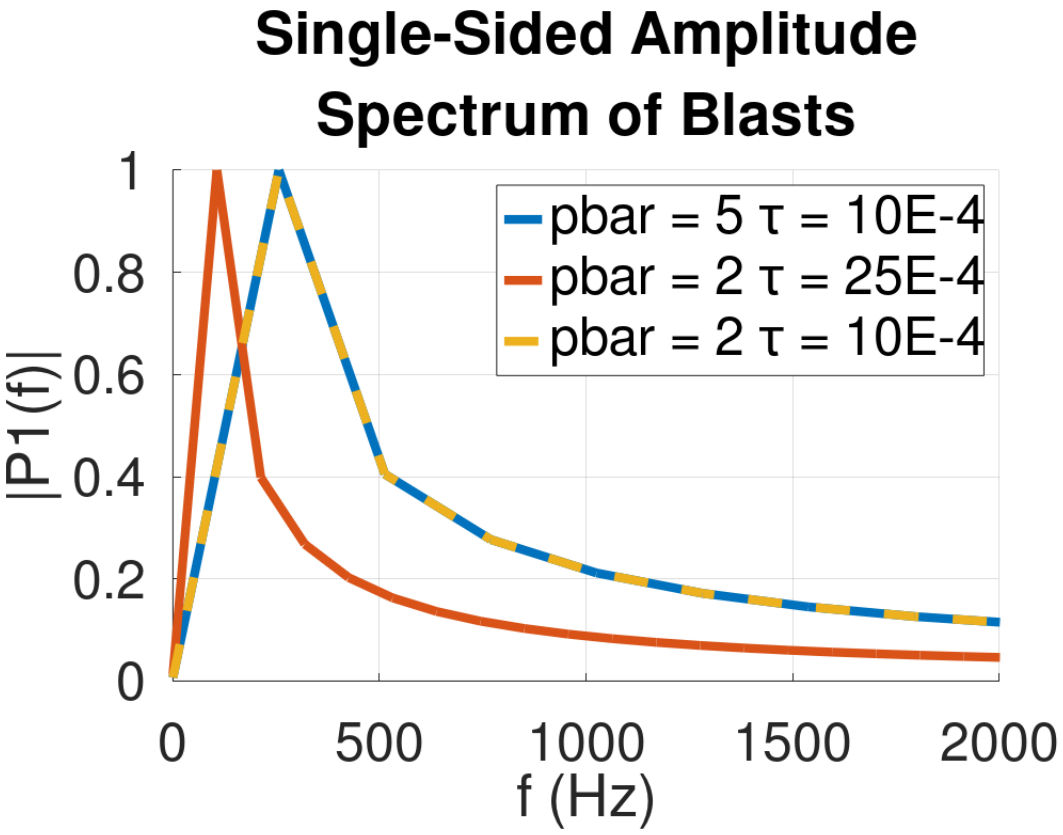

Figure 3.8: Comparison of frequency content of the different blast loads.

This numerical experiment compares the Fourier Transform of the finite element model results of the central node's Z-displacement. The displacement is in the same direction as the pressure load vector and analogous to the single degree of freedom system's displacement.

Figures 3.9, 3.10, 3.11 compare the finite element models' and single degree of freedom system's Fourier transforms of their displacement histories. The analytical method and implicit solution to the finite element model show very similar frequency content, which also corresponds to natural frequencies of the system in order of the frequency amplitude, e.g., the first mode is the highest and subsequent modes have less participation. The explicit solution to the finite element model shows a shift in the frequency content to higher frequencies. This disparity may be due to the higher frequency content of the blasts, which should excite frequencies closest to that blast, but still have substantial mass participation. The smaller impulse blast, case 3, shows the frequency content of the explicit solution to be most similar to the analytical and implicit solutions.

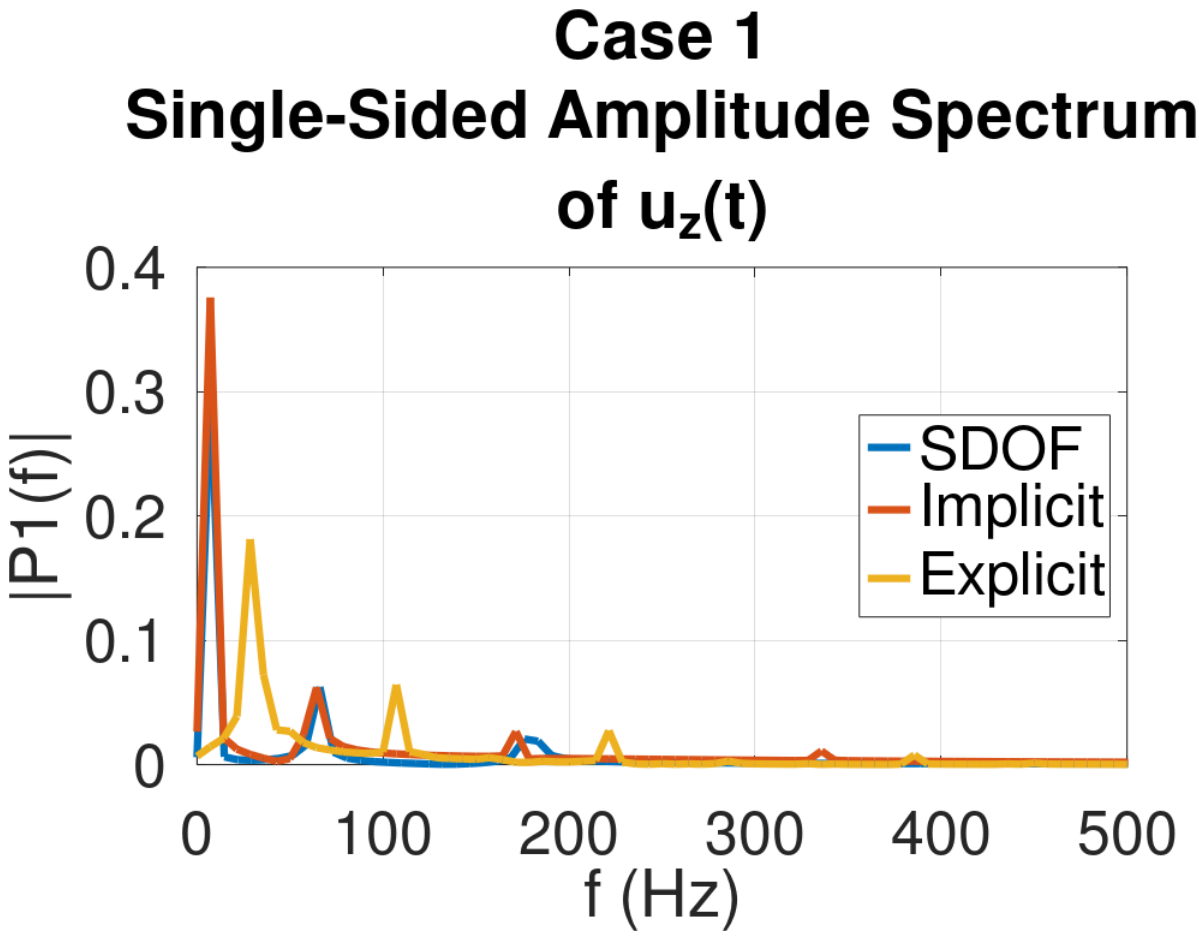

Figure 3.9: Comparison of frequency content of the responses in experiment 1.

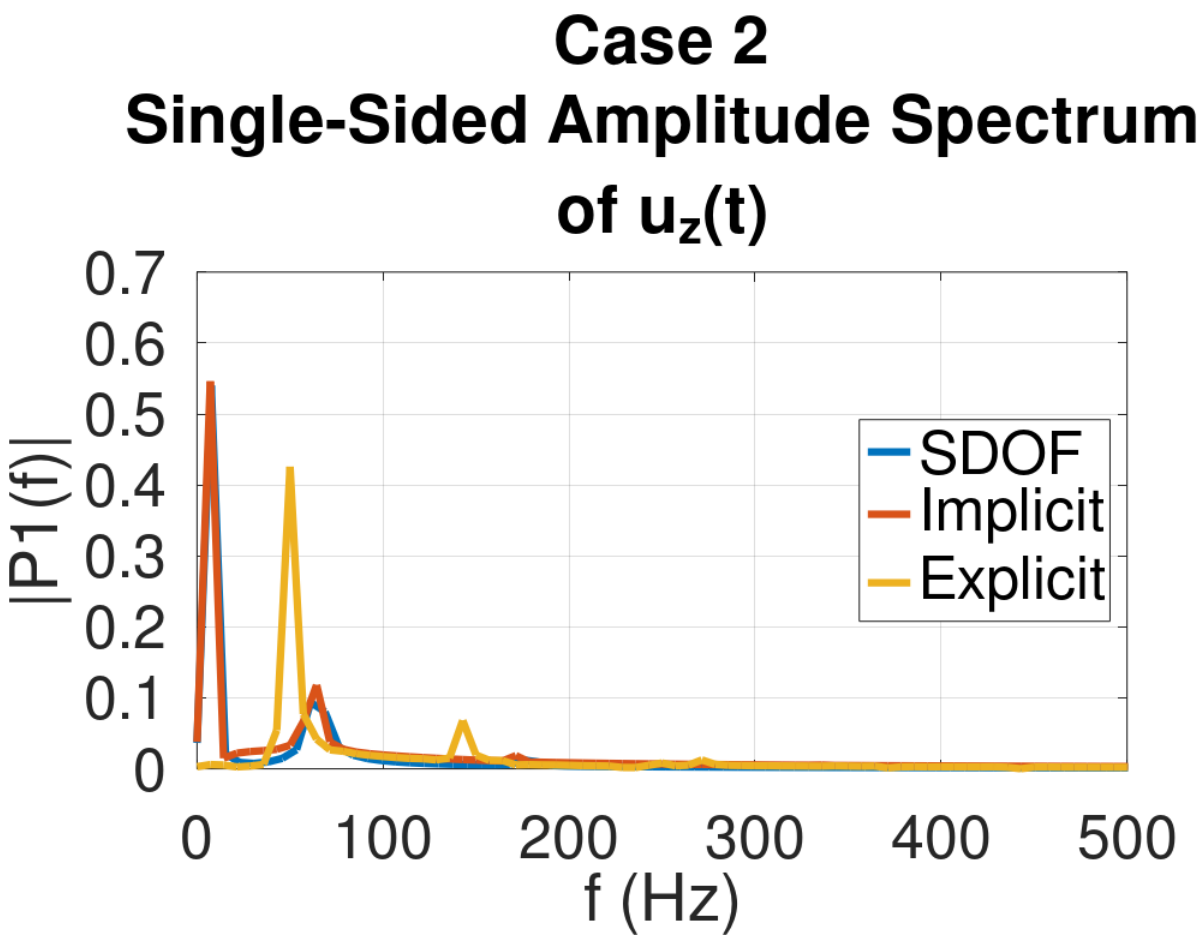

Figure 3.10: Comparison of frequency content of the responses in experiment 2.

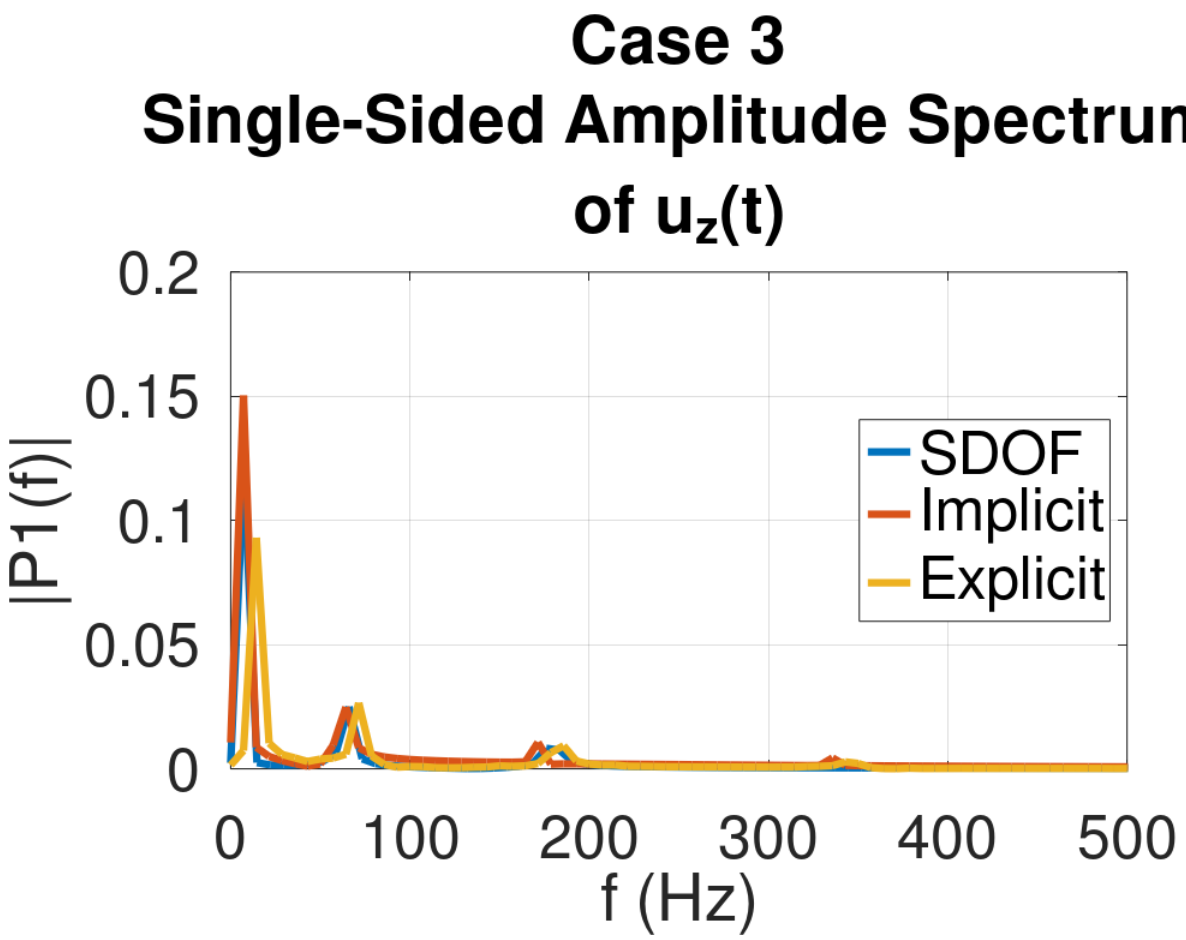

Figure 3.11: Comparison of frequency content of the responses in experiment 3.

Table 3.6 below shows the frequencies of the highest amplitude in the response of each system to the different blast loads. The Implicit model aligns with the first natural frequency of the system for each impulse. The analytical model shows the same peak frequencies for the two cases with the same impulse, and all are very close to the first natural frequency of the system. The explicit model shows different peak frequencies for each blast load, each being somewhere in between the first and second or second and third natural frequencies calculated with the implicit finite element modal analysis.

Table 3.6: Peak Frequency of Structural Response to Blast

Model	Case 1 Frequency [Hz]	Case 2 Frequency [Hz]	Case 3 Frequency [Hz]
SDOF Model	7.33	7.64	7.33
Implicit FEA	7.14	7.14	7.14
Explicit FEA	28.57	50.00	14.29

Most blast loading research neglects the negative phase of the blast due to its small magnitude. The relative length of time for the negative phase is much longer than the positive phase, and has a substantial impact on the impulse of the blast if it is neglected. Finite element models were run only using the positive phase of the blast to see if the results were still different. The figure below shows that the frequency content is still different between the explicit and implicit solutions, but similar to the simulations with the positive and negative phases. The main difference is that excluding the negative phase more than doubles the peak displacement of the responses making this an overly conservative solution.

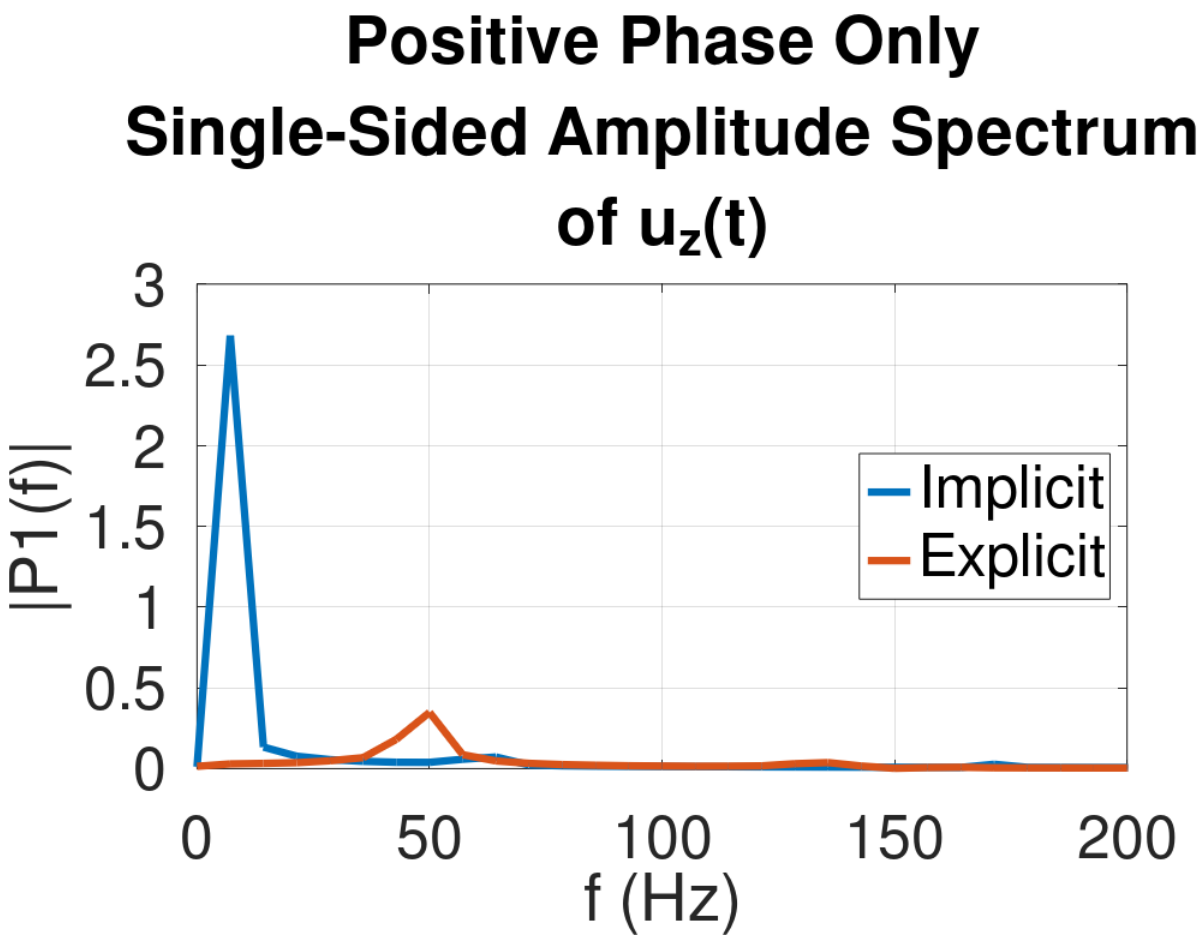

Figure 3.12: Comparison of frequency content of the responses to the blast's positive phase only.

The preceding numerical experiment does not include a spatial pressure distribution that represents a near field blast over time. This simplification helps solve the problem analytically and emphasizes using finite element models in near-field blast problems. While experimental data can help validate the varying results, the numerical comparison can shed light on some key differences without the experimental validation. The single degree of freedom system and implicit finite element model are very similar in frequency content and displacements, but the initial conditions are critical to matching the phase of the response. The explicit finite element model has a smaller peak displacement but is still reasonably close to the single degree of freedom and implicit models. The explicit finite element model

has a very different frequency content in its response than the single degree of freedom and implicit models which match the natural frequencies in order of decreasing amplitude with increasing natural frequency. The explicit model also has a different peak amplitude frequency for each different blast, while the single degree of freedom and implicit models remain virtually unchanged for different blasts. Finally, considering only the positive phase of the blast does not substantially affect the frequency content of the responses, but is overly conservative showing very large displacements relative to the entire transient blast loads.

The displacement magnitudes of all solutions are similar and would be helpful in an engineering calculation such as the dynamic load factor. However, the frequency content is very different and could affect fatigue or other normal modes based analyses. The implicit and analytical models give the most conservative solutions due to higher displacements resulting in higher stresses and strains. The explicit solution seems to be the most logical due to the non-linear loading and explicit mass and, therefore, momentum in the solution.

Chapter 4

Finite Element Blast Scaling Tool

The review of research does not show the coupling of finite element tools and muzzle blast scaling methods. Therefore, this thesis presents a tool that reads a finite element mesh, generates discrete transient loads on the finite element mesh, and outputs these loads in the native finite element solver language. Future work can be done to experimentally validate this tool, but it is recommended to utilize a diverse set of gun systems in the experimental data collection. This thesis uses LS-Dyna as the finite element solver for its non-linear capabilities, research history in blast loading, and ASCII-based input files. Figure 4.1 shows a flow diagram that describes the tool's process. The blue shapes signify a part of the tool that maps the blast to the finite element model. The green box encompasses the blast scaling calculations based on the method presented by Heaps et al. [12].

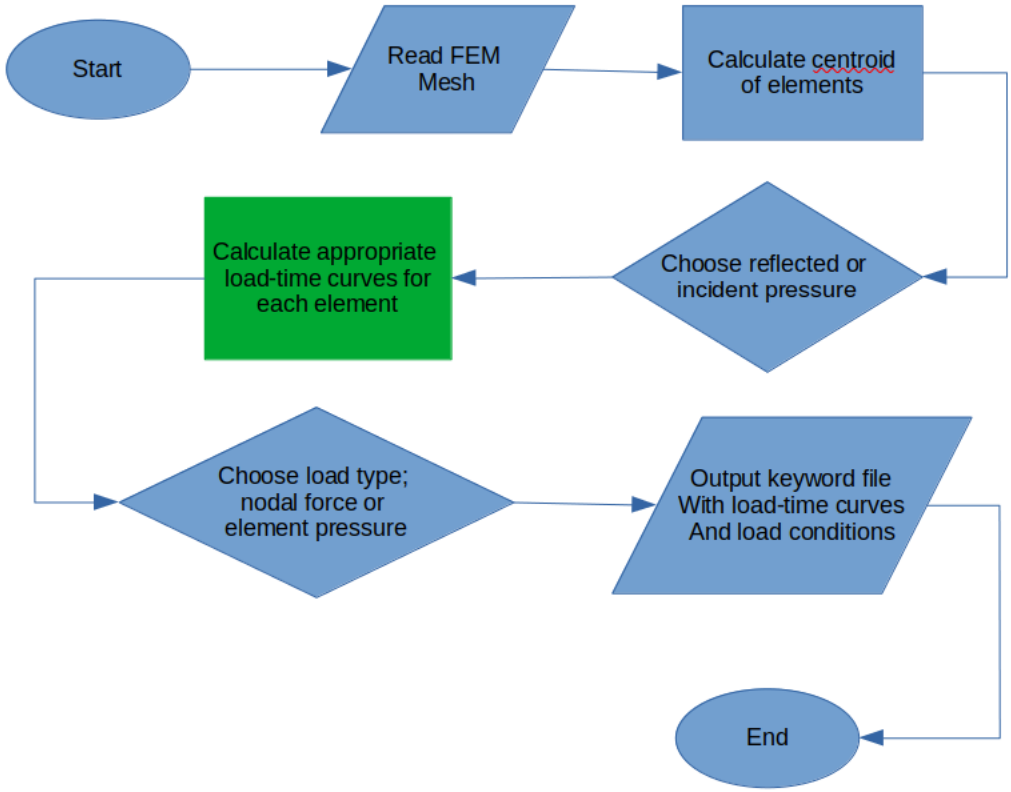

Figure 4.1: Flow diagram of Finite Element Muzzle Blast Tool.

The blast scaling methodology from Heaps et al. [12] is translated from FORTRAN to MATLAB and utilized in the larger program. The tool was written in MATLAB due to existing knowledge of the language. MATLAB also provides easy prototyping and debugging capabilities. The tool provides one option to load the finite elements with a distributed pressure load. The tool includes a second option has that converts the pressure-time history to a force by multiplying the pressure and the element's area. The force is then distributed equally to the element's nodes. The program then outputs a text keyword file that can be included in the original model when read into the LS-Dyna solver. Investigations show no discernible difference in the results between the loading methods.

The script calculates the blast using the scaled distance from the muzzle center to

the centroid of the element. Heaps et al. [12] created a FORTRAN program to calculate and plot arbitrary contour planes of a muzzle blast's peak overpressure, time of arrival, and positive phase duration. The scaling component of the tool uses these parameters and the modified Friedlander equation to create a pressure-time history as a transient load for an explicit finite element structural simulation. The key benefits of Heaps et al. [12]'s scaling routine are its ability to account for angle of incidence, and blast reflection pressures which are higher than incident pressures. The routine uses the flow deflection across the incident shock to determine if it must calculate for regular reflection with the theory of oblique shocks or single Mach stem reflection which has no analytical solution [12]. The scaling routine is valid for the near to medium field, or quantitatively with the non-dimensional scaling parameter $1 \leq \frac{r}{l'} \leq 100$ [12].

4.1 Mesh Convergence Study

The mesh sensitivity study examines finite element meshes of 24" x24" x0.1" Aluminum plate centered 36" from muzzle center and parallel to the bore axis. The edges of the plate that are parallel to the direction of fire have all translational and rotational degrees of freedom locked for a clamped boundary condition. Figure 4.2 shows the finite element model. The nominal element sizes are 2, 1, 0.5, and 0.25 inches. The mesh sensitivity study uses point and pressure loading techniques, as well as incident and reflected scaled pressures calculated with the finite element blast scaling tool that was created based on [12]. In addition, this thesis uses the grid convergence index (GCI) method, which researchers developed to uniformly report results of grid refinement studies [24].

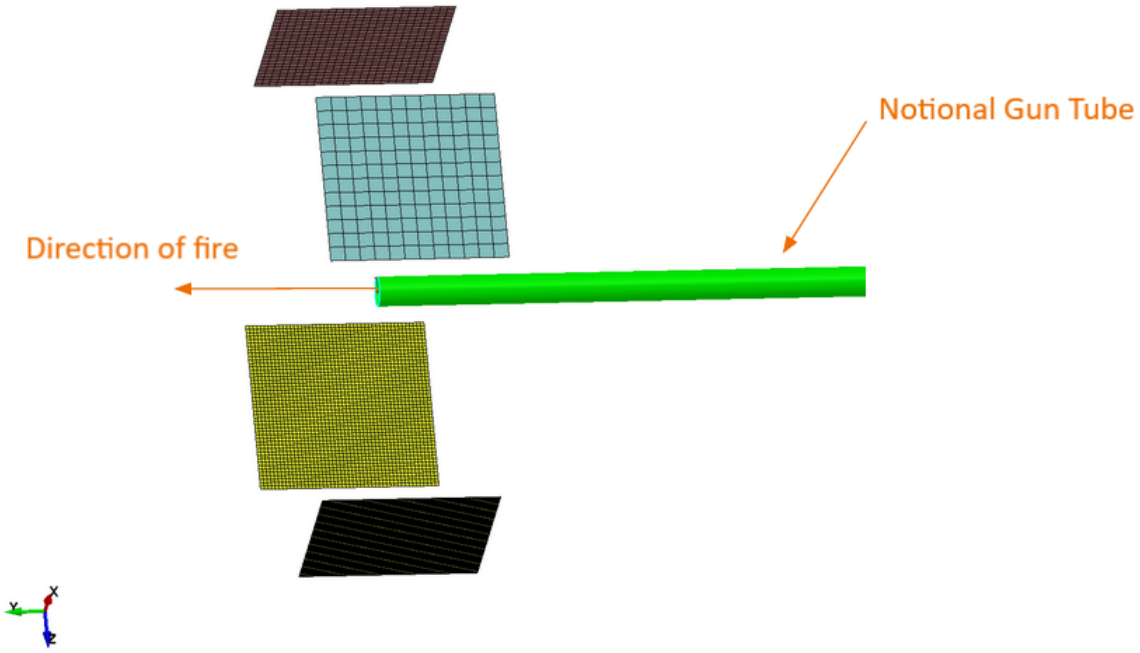

Figure 4.2: Finite element model used for mesh convergence calculations.

Discretization error estimates can be used in finite element simulations to correct the solutions. However, estimates must demonstrate that the solutions are in the asymptotic range before gaining confidence that the error estimate is reliable [24]. The GCI method is a fractional estimate of the relative uncertainty in a fine grid solution, as described by Equation 4.1. The GCI method accounts for the amount of grid refinement and the order of accuracy and converts the discretization error estimate into an uncertainty estimate [24]. Three different mesh sizes are required to determine the observed order of accuracy, and a fourth mesh can confirm that the asymptotic range has been reached. The difference between the formal and observed orders of accuracy drives the factor-of-safety value used for the GCI calculations. [24]

$$GCI = \frac{F_s}{r^p - 1} \left| \frac{f_2 - f_1}{f_1} \right| \tag{4.1}$$

The equations below show the general process that this thesis uses to calculate the uncertainty estimate of the blast models. First, the method chooses the central node of each mesh to calculate the maximum resultant displacement. Next, the process calculates the convergence rate exponent using the coarsest, F_1, the most refined mesh, F_2, and the mesh refinement ratio, r. The final step calculates an estimated numerically exact solution and the relative error to the estimated numerically exact.

$$u_x^2 + u_y^2 + u_z^2 = |u|^2 \tag{4.2}$$

$$p = \frac{\ln\left(\frac{F_1 - F_3}{F_1 - F_2}\right)}{\ln(r)} \tag{4.3}$$

$$F_0 = F_1 + \frac{F_1 - F_2}{r^p - 1} \tag{4.4}$$

$$\text{RelErr} = 100 \left| \frac{F_0 - F_1}{F_0} \right| \tag{4.5}$$

The models that use the incident pressure from the blast scaling tool do not show significant evidence that the mesh triplet was in the asymptotic range. This discrepancy was not a significant concern because the incident pressures are rarely found in structural interaction alone and are much smaller than reflected and total pressure. Both models that used the reflected pressure loads showed a 9% difference between observed and formal orders of accuracy, with a grid convergence index of 2% for a 2-inch mesh and 0.14% index for a 0.25-inch mesh. The figures below show pressure contours of the 0.25-inch mesh over time.

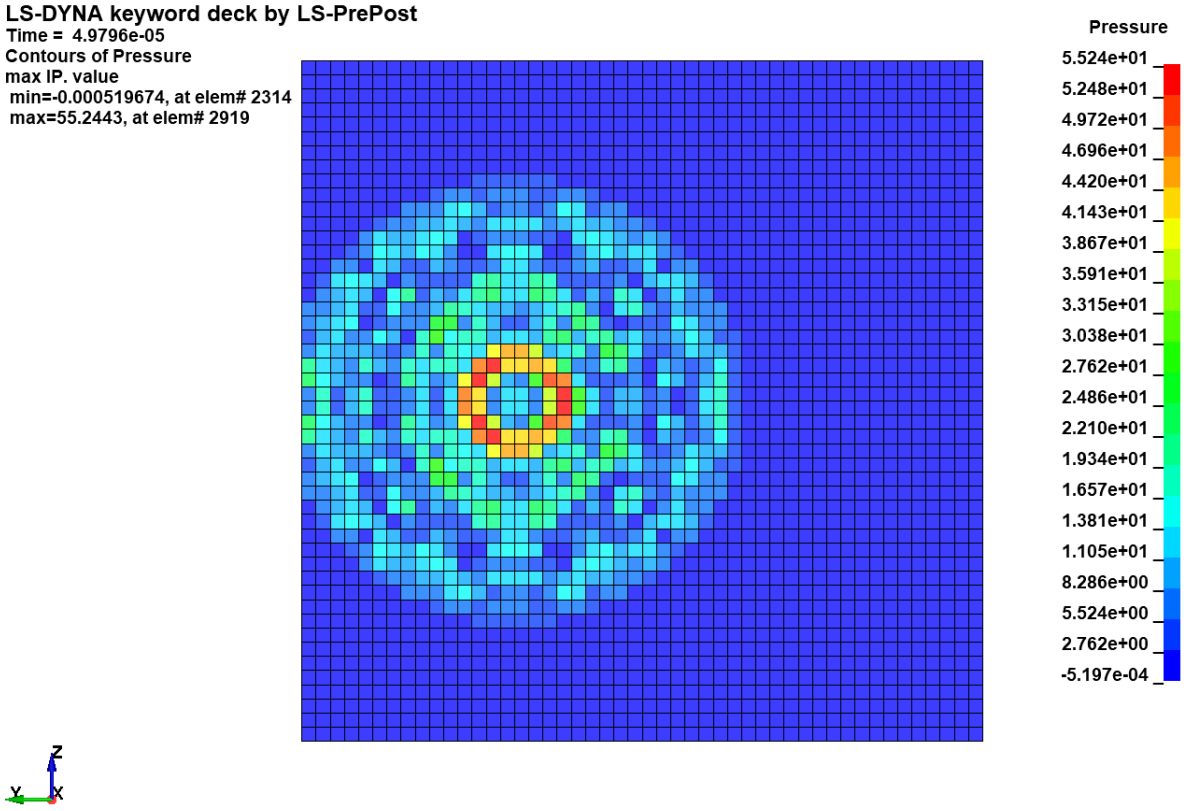

Figure 4.3: Pressure contour of muzzle blast tool loaded panel, $t = 0.000050s$.

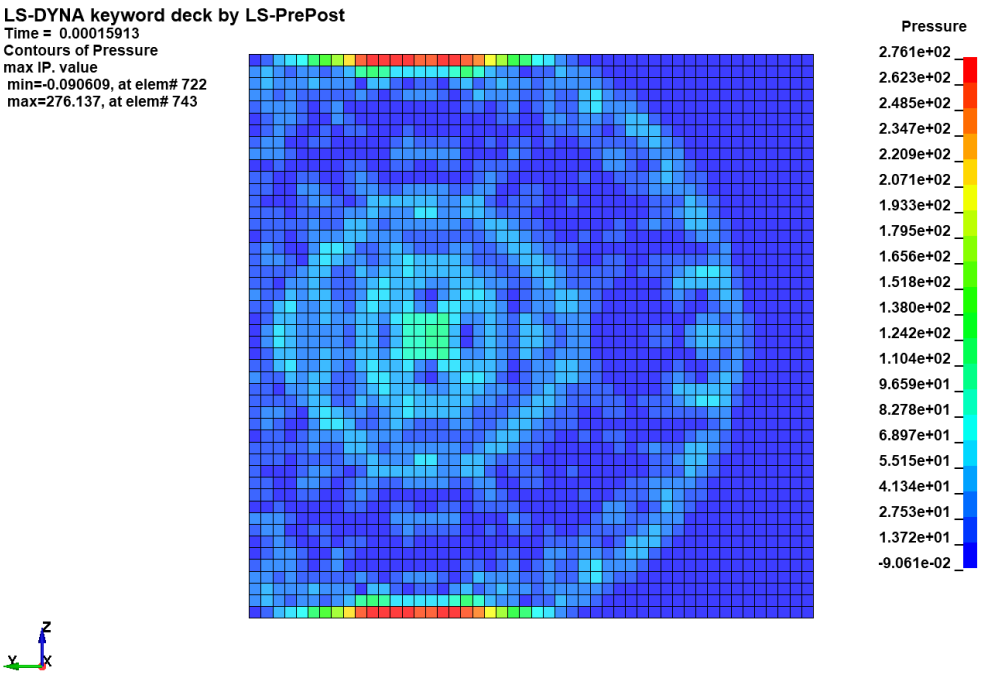

Figure 4.4: Pressure contour of muzzle blast tool loaded panel, $t = 0.000159s$.

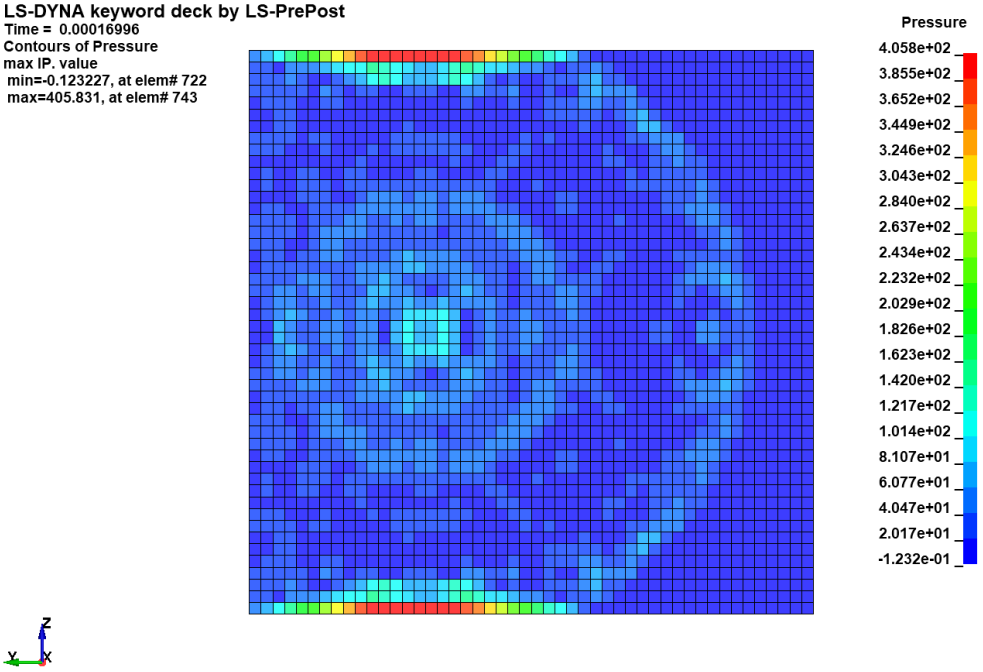

Figure 4.5: Pressure contour of muzzle blast tool loaded panel, $t = 0.000170s$.

Chapter 5

Summary

The research of blast-structure phenomenon is a broad and complex field. Researchers can explore new fields with the help of handy simulation tools such as muzzle blast scaling. Understanding gun blast loads and their interaction with surrounding structures is critical to successfully integrating gun systems into lightweight vehicles. Exploring the effects of large caliber muzzle blasts on human tissues is also an interesting application area. The impact of different numerical techniques on the simulation's solution should be clearly understood.

This thesis shows that the muzzle blast scaling finite element tool has is flexible and has a wide range of applications. Future work can show further application of this tool using MM-ALE domains for a more coupled FSI simulation. It also shows that the standard rules of thumb apply reasonably to mesh sizing with low errors found in the mesh convergence study. Finally, this thesis shows that different load application techniques, e.g. nodal point loads or element-based pressure loads, do not affect the solution of the finite element model.

Solution techniques for different problems have fewer barriers than before due to computer hardware and software advancement. Analytical and implicit numerical techniques used in past research can have different responses than explicit solutions to blast loading. Explicit solutions have a well-documented ability to handle non-linear problems more accurately and is the solution method chosen for conducting a mesh convergence study with the muzzle blast scaling tool. Testing data can be beneficial in further validation of numerical methods that are the most accurate at solving muzzle blast-structure interaction problems. Test data

of incident muzzle blast pressures can also be beneficial in understanding techniques and limitations for modeling more complex fluid interaction with finite element methods, such as MM-ALE models.

Bibliography

[1] A718300. Engineering design handbook. explosions in air. part one. *Army Materiel Command Alexandria, VA*, 1974.

[2] Robert A. Carson and Onkar Sahni. Scaling laws for the peak overpressure of a cannon blast. *Journal of Fluids Engineering*, 139(2), 2017.

[3] Jeffrey MK CHOCK and Rakesh K KAPANIA. Review of two methods for calculating explosive air blast. *The Shock and vibration digest*, 33(2):91–102, 2001.

[4] Jeffrey Mun Kong Chock. Review of methods for calculating pressure profiles of explosive air blast and its sample application. Master's thesis, Virginia Tech, 1999.

[5] Paul W. Cooper and Stanley R. Kurowski. *Introduction to the Technology of Explosives*. VCH Publishers, 1996.

[6] John M. Dewey. The friedlander equations. In *Blast Effects*, pages 37–55. Springer, 2018.

[7] J.I. Erdos and P.D. Del Guidice. Calculation of muzzle blast flowfields. *AIAA journal*, 13(8):1048–1055, 1975.

[8] Kevin S. Fansler. Description of muzzle blast by modified ideal scaling models. *Shock and Vibration*, 5(1):1–12, 1998.

[9] K.S. Fansler and E.M. Schmidt. The prediction of gun muzzle blast properties utilizing scaling. US Army ballistic research laboratory, aberdeen proving ground, md. Technical report, BRL Technical Report ARBRL-TR-02504 (AD B075859L), 1983.

[10] F.G. Freidlander. The diffraction of sound pulses. I. diffraction by a semi-infinite plane. *Proc. Roy. Soc. Lond. A*, 186:322–344, 1946.

[11] .L Gilson, J. Van Roey, C. Guéders, J. Gallant, and L. Rabet. A simple coupling of ALE domain with empirical blast load function in LS-DYNA. In *EPJ Web of Conferences*, volume 26, page 04018. EDP Sciences, 2012.

[12] C.W. Heaps, K.S. Fansler, and E.M. Schmidt. Computer implementation of a muzzle blast prediction technique. *The Shock and Vibration Bulletin*, 56:213–228, 1986.

[13] David W. Hyde. Microcomputer programs CONWEP and FUNPRO, applications of TM 5-855-1. Technical report, Army Engineer Waterways Experiment Station Vicksburg MS Structures Lab, 1988.

[14] LS-DYNA Keyword User's Manual Volume I. Version 971. *Livermore Software Technology Corporation*, 7374:354, 2007.

[15] Tomiko Ishiguro. Finite-difference calculations for two-dimensional unsteady expanding flows. *AIAA Journal*, 10(2):217–219, 1972.

[16] A. Kalra, F. Zhu, King H. Yang, and A.I. King. Key parameters in blast modeling using 2D to 3D ALE mapping technique. In *13th International LS-DYNA User Conference*, pages 1–8, 2014.

[17] C.N. Kingery and G. Bulmash. Technical report ARBRL-TR-02555: Air blast parameters from tnt spherical air burst and hemispherical burst. *AD-B082*, 713, 1984.

[18] Günter Klingenberg and Joseph M. Heimerl. Gun muzzle blast and flash. *Progress in astronautics and aeronautics*, 139, 1992.

[19] Dennis G. Mabey and Derek S. Capps. Blast from moving guns. *Journal of Aircraft*, 14(7):687–692, 1977.

[20] G.R. Moore. Finite difference calculations of the free-air gun blast about the muzzle of a 5in./54 naval gun. Technical report, Naval Weapons Lab Dahlgren VA, 1972.

[21] G. Moretti. Muzzle blast flow and related problems. In *11th Fluid and PlasmaDynamics Conference*, page 1190, 1978.

[22] Michael J. Mullin and Brendan J. O'Toole. Simulation of energy absorbing materials in blast loaded structures. In *8th International LS-DYNA Users Conference*, pages 2–7, 2004.

[23] BBC News. Beirut explosion: what we know so far, 2020.

[24] William L. Oberkampf and Christopher J. Roy. *Verification and validation in scientific computing*. Cambridge University Press, 2010.

[25] J.T. Oden and K.J. Bathe. A commentary on computational mechanics. *Appl. Mech. Rev*, 31(8):1053–1058, 1978.

[26] Edward M. Schmidt. Muzzle blast pressure loadings upon aircraft surfaces. Technical report, Army Ballistic Research Lab Aberdeen Proving Ground MD, 1984.

[27] Len Schwer. A brief introduction to coupling load blast enhanced with Multi-Material ALE: the best of both worlds for air blast simulation. In *LS-DYNA Forum. Bamberg*, 2010.

[28] A.C. Singhal and D.S. Larson. Computer simulation of weapon blast pressures on flexible surfaces. *Computers & structures*, 41(2):325–330, 1991.

[29] Todd P. Slavik. A coupling of empirical explosive blast loads to ALE air domains in LS-DYNA®. In *IOP Conference Series: Materials Science and Engineering*, volume 10-1, page 012146. IOP Publishing, 2010.

[30] F. Smith. A theoretical model of the blast from stationary and moving guns. In *First International Symposium on Ballistics, Orlando, FL*, pages 13–15, 1974.

[31] Chris Stennett, Sally Gaulter, and Jackie Akhavan. An estimate of the tnt-equivalent net explosive quantity (neq) of the beirut port explosion using publicly-available tools and data. *Propellants, Explosives, Pyrotechnics*, 45(11):1675–1679, 2020.

[32] Hugh W. Stephens. The texas city disaster: a re-examination. *Industrial & Environmental Crisis Quarterly*, 7(3):189–204, 1993.

[33] Zahra S. Tabatabaei and Jeffery S. Volz. A comparison between three different blast methods in LS-DYNA: LBE, MM-ALE, Coupling of LBE and MM-ALE. In *12th International LS-DYNA User Conference, Detroit, MI, June*, pages 3–5, 2012.

[34] T.D. Taylor and T.C. Lin. Numerical model for muzzle blast flowfields. *AIAA Journal*, 19(3):346–349, 1981.

[35] John Von Neumann. The point source solution. *Bethe [Bet47]*, page 13, 1941.

[36] P. Westine. The blast field about the muzzle of guns. *Shock Vib. Bull*, 39(6):139–149, 1969.

Appendices

Appendix A

LS-Dyna Input Decks

All LS-Dyna input decks can be retrieved by emailing me at xaquinn@vt.edu.

Appendix B

Analytical Model and Plotting

Identical code was used for each experimental blast load. Blast parameters were adjusted and respective binaries loaded for each case.

```
% This script calculates the free vibration response of a single degree of
% freedom system to a blast pulse using the Duhamel Integral
% Panel Geometry
s = 0.1;                % Panel thickness [in]
L = 36;                 % Panel span [in]
w = 1;                  % Panel width [in]
rho = 0.000259;         % Density [lbf-s^2/in/in^3]
nu = 0.33;              % Poisson's ratio
E = 9.7e6;             % Young's Modulus [psi]
mL = rho*s*w;           % Mass per unit length [lb_f-sec^2/in/in]
I = w*s^3/(12*(1-nu^2)); % Moment of Inertia [in^4]
% Blast Overpressure Impulse
pbar = -5*w; %Gives a uniformly distributed load across the panel in lbf/in.
% Assumptions (positive phase duration, total blast impulse time)
r = 10e-4;      % tau [sec]
a = 0.2728;  % alpha [unitless]
t_d = r/a;  % sec

for ii = [1 3 5]
% Energy Formulation for Equivalent System
Me(ii) = L*mL/2;
```

```
Ke(ii) = pi^4*ii^4*E*I/(2*L^3);
wn(ii) = ii^2*pi^2*sqrt(E*I/(mL*L^4)); % natural frequency
F0(ii) = pbar*L*2/(ii*pi);

% Duhamels Integral - Max Dynamic Load Factor
y_st(ii) = F0(ii)/Ke(ii); % [in]

ytd(ii) = (2*L*pbar*(-24*a^3-2*a*(-(-1+a)*r^2+3*a*(1+a)*r*t_d-6*a^2*t_d^2)*wn(
    ii)^2 +...
 (r-t_d)*(r-a*t_d)^2*(r+a*t_d)*wn(ii)^4 + (24*a^3-2*(-1+a)*a*r^2*wn(ii)^2 - r
    ^4*wn(ii)^4)*...
 cos(t_d*wn(ii)) + (1 + a)*r*wn(ii)*(6*a^2 + r^2*wn(ii)^2)*sin(t_d*wn(ii))))/(
    ii*Me(ii)*pi*r^4*wn(ii)^6);
vtd(ii) = (2*L*pbar*(-2*a*(3*a*(1+a)*r-12*a^2*t_d)*wn(ii)^2+a*(r-t_d)*(r-a*t_d
    )^2*wn(ii)^4-2*a*(r-t_d)*...
 (r-a*t_d)*(r+a*t_d)*wn(ii)^4-(r-a*t_d)^2*(r+a*t_d)*wn(ii)^4+(1+a)*r*wn(ii)
    ^2*(6*a^2 + r^2*wn(ii)^2)*...
 cos(t_d*wn(ii))-wn(ii)*(24*a^3-2*(-1+a)*a*r^2*wn(ii)^2-r^4*wn(ii)^4)*sin(t_d*
    wn(ii))))/(ii*Me(ii)*pi*r^4*wn(ii)^6);

t = linspace(t_d,0.14,14000);;
yt(ii,:) = ytd(ii).*cos(wn(ii)*(t-t_d))+(vtd(ii)./wn(ii)).*sin(wn(ii)*(t-t_d))
    ;

##maxDLF(ii) = max(abs(yt(ii,:)))./abs(y_st(ii));
##idx = yt(ii,:) == max(abs(yt(ii,:)));
##time_of_maxDLF = t(idx);
##fprintf('Max DLF is %f at a time of %f.\n',maxDLF(ii),time_of_maxDLF)
end

load .\Exp1\Analytical\ExplicitFEA1
```

```
load .\Exp1\Analytical\ImplicitFEA1

figure; hold on; grid on;
plot(t,yt(1,:),"linewidth", 3,...
timeFEAImp,dispFEAImp(:,3),"linewidth", 3,...
timeFEA,dispFEA(:,3),"linewidth", 3)
set(gca, "FontSize", 24)
title('System Response Comparison')
xlabel('Time [s]')
ylabel('Z Displacement [in]')
legend('Duhamel','Implicit','Explicit')
hold off;

figure; hold on; grid on;
plot(t,yt(1,:)+yt(3,:)+yt(5,:),"linewidth", 3,...
timeFEAImp,dispFEAImp(:,3),"linewidth", 3,...
timeFEA,dispFEA(:,3),"linewidth", 3)
set(gca, "FontSize", 24)
title({'Experiment 1';'System Response Comparison'})
xlabel('Time [s]')
ylabel('Z Displacement [in]')
legend('Duhamel','Implicit','Explicit')
hold off;
%~~~~~~~~~~~~~~~~~~~~~~~~~~~ Momentum-Impulse
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
idx = abs(timeFEA-(t_d+.00001)) == min(abs(timeFEA-(t_d+.00001)));
vtdFEA = veloFEA(idx,3);
idx = abs(timeFEAImp-(t_d+.00001)) == min(abs(timeFEAImp-(t_d+.00001)));
vtdFEAImp = veloFEAImp(idx,3);
I = -7.2561E-03;
vf= I/Me(1);
```

```
76 fprintf('Velocity of SDOF sytem according to Momentum-Impulse Theorem is %f in
       /s.\n',vf)
77 fprintf('Velocity of SDOF sytem at the end of the blast is %f in/s.\n',vtd)
78 fprintf('Velocity of Explicit Model at the end of the blast is %f in/s.\n',
       vtdFEA)
79 fprintf('Velocity of Implicit Model at the end of the blast is %f in/s.\n',
       vtdFEAImp)
80 %~~~~~~~~~~~~~~~~~~~~~~~~~~~ FFT Calcs
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
81 % Determine FRF from FFT of impulse and FFT of various methods' responses
82 timeDI = t;
83 load .\Exp1\Analytical\loadCurve1
84 %~~~~~~~~~~~~~~~~~~~~~~~~~~~ Impulse FFT
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
85 Ts = t(3) - t(2);; % sampling period
86 Fs = 1/Ts; % Sampling frequency
87 L = numel(t); % number of sampling points
88 X = F*pbar;
89 Y = fft(X);
90 P2 = abs(Y/L);
91 P1 = P2(1:L/2+1);
92 P1(2:end-1) = 2*P1(2:end-1);
93 f = Fs*(0:(L/2))/L;
94 %~~~~~~~~~~~~~~~~~~~~~~~~ Explicit FEA FFT
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
95 Ts = timeFEA(3) - timeFEA(2); % sampling period = fea timestep
96 Fs = 1/Ts; % Sampling frequency
97 L = numel(timeFEA); % number of sampling points
98 X = dispFEA(:,3);
99 Y = fft(X);
100 P2 = abs(Y/L);
```

```
P1_FEAExp = P2(1:L/2+1);
P1_FEAExp(2:end-1) = 2*P1_FEAExp(2:end-1);
f_FEAExp = Fs*(0:(L/2))/L;
%~~~~~~~~~~~~~~~~~~~~~~~~~ Linear Implicit FEA FFT
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Ts = timeFEAImp(3) - timeFEAImp(2); % sampling period = fea timestep
Fs = 1/Ts; % Sampling frequency
L = numel(timeFEAImp); % number of sampling points
X = dispFEAImp(:,3);
Y = fft(X);
P2 = abs(Y/L);
P1_FEAImp = P2(1:L/2+1);
P1_FEAImp(2:end-1) = 2*P1_FEAImp(2:end-1);
f_FEAImp = Fs*(0:(L/2))/L;
%~~~~~~~~~~~~~~~~~~~~~~~~~ Duhamel's Integral FFT
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Ts = timeDI(3) - timeDI(2); % sampling period = fea timestep
Fs = 1/Ts; % Sampling frequency
L = numel(timeDI); % number of sampling points
X = yt(1,:)+yt(3,:)+yt(5,:);
Y = fft(X);
P2 = abs(Y/L);
P1_DI = P2(1:L/2+1);
P1_DI(2:end-1) = 2*P1_DI(2:end-1);
f_DI = Fs*(0:(L/2))/L;
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Peaks
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
idx = P1 == max(P1);
peakFreqBI = f(idx);
idx = P1_FEAExp == max(P1_FEAExp);
peakFreqExp = f_FEAExp(idx);
```

```
idx = P1_FEAImp == max(P1_FEAImp);
peakFreqImp = f_FEAImp(idx);
idx = P1_DI == max(P1_DI);
peakFreqDI = f_DI(idx);
fprintf('Peak amplitude of the blast impulse is at %f Hz.\n',peakFreqBI)
fprintf('Peak amplitude of the explicit response is at %f Hz.\n',peakFreqExp)
fprintf('Peak amplitude of the implicit response is at %f Hz.\n',peakFreqImp)
fprintf('Peak amplitude of the analtical response is at %f Hz.\n',peakFreqDI)
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ PLOTS
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
figure;
plot(f_DI,P1_DI,"linewidth", 3,f_FEAImp,P1_FEAImp','"linewidth", 3,f_FEAExp,
    P1_FEAExp','"linewidth", 3)
grid on;
title({'Experiment 1';'Single-Sided Amplitude Spectrum of u_z(t)'})
xlabel('f (Hz)')
ylabel('|P1(f)|')
xlim([0 500])
h = legend('Duhamel','Implicit','Explicit');
legend(h,"location","east")
set(gca, "fontsize", 24)
set (h, "fontsize", 24);
```

Appendix C

Muzzle Blast Scaling Tools

C.1 FEA to Blast Scaling Script

```
%{
This script reads an LS-Dyna keyword file to find the coordinates of the
nodes, or the coordinates of the centroid of the elements. Then calculates
the blast scaled parameters; time of arrival, positive phase duration, and
peak over pressure, in order to create a pressure vs. time loading curve to
be input into a transient analysis.

Assumes 3 or 4 noded shell elements only
Assumes x-y axes makes up the horizontal plane (positive Z is up)
Assumes units follow MKS system based on Fansler et al. program
Simulation begins at the earliest time of arrival (tsuba) being t=0
%}
%
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%% User selects load application; 0 = *LOAD_NODE, 1 = *LOAD_SEGMENT
% loadtype = input('select load application;  0 = *LOAD_NODE, 1 = *
    LOAD_SEGMENT\n');
% reflyn = input('reflected press\n');
loadtype = 1;
reflyn = 1;
```

```
filename = 'ShellPanel.k';
%
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%% User inputs Model Properties
boreVec = [0,1,0];  % input unit vector in the direction of fire
origin = [0,0,0];  % input the coordinates of the center of the muzzle in the
    FEM global coordinate system (meters)
%
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%% Nodal Coordinates
fmt = ('%u %f %f %f %u %u');
N = {};
fid = fopen(filename);
while (!feof(fid))
    str = fgetl(fid);
    if strcmpi('*NODE',str)
        %temp = fgetl(fid); % Use this line if keyword file has comments
        %temp = fgetl(fid); % Use this line if keyword file has comments
        N{end+1} = textscan(fid,fmt);
    end
end
%% Element IDs
fmt = repmat('%8u ',1,10);
C = {};
frewind(fid);
while (!feof(fid))
    str = fgetl(fid);
    if strcmpi('*ELEMENT_SHELL',str)
        %temp = fgetl(fid); % Use this line if keyword file has comments
```

```
        %temp = fgetl(fid);
        C{end+1} = textscan(fid,fmt);
    end
end

%% Termination Time
fmt = ('%f %u %f %f %f %u');
T = {};
frewind(fid);
while (!feof(fid))
    str = fgetl(fid);
    if strcmpi('*CONTROL_TERMINATION',str)
        %temp = fgetl(fid); % Use this line if keyword file has comments
        %temp = fgetl(fid);
        T{end+1} = textscan(fid,fmt);
    end
end
fclose(fid);
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

%% Variable Declaration
term = T{1,1}{1,1}; % termination time of simulation

NodeID = N{1,1}{1,1};
NodeX = N{1,1}{1,2};
NodeY = N{1,1}{1,3};
NodeZ = N{1,1}{1,4};

ElemID = C{1,1}{1,1};
ElemNodes = [C{1,1}{1,3},C{1,1}{1,4},C{1,1}{1,5},C{1,1}{1,6}];
```

```
clear -v fid C N T temp str fmt ans

for ii = 1:numel(ElemID)
    %% Determine Centroid of Element
    if ElemNodes(ii,4) == 0
        n(ii) = 3;
    else
        n(ii) = 4;
    end
    for jj = 1:n(ii)
        temp = ElemNodes(ii,jj) == NodeID;
        vec(jj,:) = [NodeX(temp),NodeY(temp),NodeZ(temp)];
    end
    if abs(sum(vec(:,1))/size(vec,1) - vec(1,1)) < 1e-3
        %Element is in the Y-Z Plane
        y1 = vec(:,2)';
        z1 = vec(:,3)';
        [y,z] = polycent(y1,z1);
        A(ii) = polyarea(x1,y1);
        Cent(ii,:) = [vec(1,1) y z];
        nVec = [1,0,0];
    elseif abs(sum(vec(:,2))/size(vec,1) - vec(1,2)) < 1e-3
        %Element is in the X-Z Plane
        x1 = vec(:,1)';
        z1 = vec(:,3)';
        [x,z] = ploycent(x1,z1);
        A(ii) = polyarea(x1,y1);
        Cent(ii,:) = [x vec(1,2) z];
        nVec = [0,1,0];
    elseif abs(sum(vec(:,3))/size(vec,1) - vec(1,3)) < 1e-3
        %Element is in the X-Y Plane
```

```
105         x1 = vec(:,1)';
106         y1 = vec(:,2)';
107         [x,y] = polycent(x1,y1);
108         A(ii) = polyarea(x1,y1);
109         Cent(ii,:) = [x y vec(1,3)];
110         nVec = [0,0,1];
111     else
112         warning('No Centroid calculated')
113     end
114     clear -v vec
115     %% Determine Blast Parameters At Centroid of Element
116     [pbar(ii),tsuba(ii),tau(ii)] = blastScaleVec(boreVec,origin,[Cent(ii,1),
        Cent(ii,2),Cent(ii,3)],nVec,reflyn);
117     b = 2; % Decay parameter
118     %     pbarConv(ii) = pbar(ii)*101325; % convert units atm -> Pa
119     pbarConv(ii) = pbar(ii)*14.6959; % convert units atm -> psi
120     %% Determine Load vs Time
121     t = linspace(0,0.01,50);
122     p(ii,:) = pbarConv(ii).*(1-(t./tau(ii))).*exp(-b.*(t./tau(ii)));
123
124 end
125 ##fprintf('Max Overpressure = %f psi\n',max(pbar)*14.5038);
126 ##maxind = pbar == max(pbar);
127 ##fprintf('Location of Max Overpressure = [%f,%f,%f] inches\n',Cent(maxind,1),
        Cent(maxind,2),Cent(maxind,3));
128 %
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

129 % Countour Plot for Verification
130 tplot = linspace(min(tsuba),0.01,200);
131 pgrid = zeros(48,48);
```

```
inc = 1;
xcent = unique(round(Cent(:,1)*100)/100);
ycent = unique(round(Cent(:,2)*100)/100);
[xgrid,ygrid] = meshgrid(xcent,ycent);
 for kk = 1:numel(tplot)
     for ii = 1:48
         for jj = 1:48
             ind = (round(Cent(:,1)*100)/100 == xcent(ii) & round(Cent(:,2)
   *100)/100 == ycent(jj));
             pgrid(ii,jj) = pbar(ind)*14.5038;
             tagrid(ii,jj) = tsuba(ind);
             taugrid(ii,jj)= tau(ind);
             if tsuba(ind) > tplot(kk)
                 ptgrid(ii,jj,kk) = 0;
             else
                 ptgrid(ii,jj,kk) = 14.5038*pbar(ind)*(1-(tplot(kk)/tau(ind)))
   *exp(-b*(tplot(kk)/tau(ind)));
             end
         end
     end
 end
## figure;
##surf(xgrid,ygrid,ptgrid(:,:,1)'); % plot the first set of data
## axis manual      % this way the scales do not change
## for ii = 2:numel(t)   % now animate;
##     % ...time steps or data sets
##     surf(xgrid,ygrid,ptgrid(:,:,ii)'); % replace by pressure data
##     pause(1)   % nope. to slow it down. you loop through...
##     % ...draw > pause > draw > pause etc.
## end
## figure(1)
```

```
## filename = 'pressvtime.gif';
## for n = 1:length(t)
##       surf(xgrid,ygrid,ptgrid(:,:,n)') ;
##       zlim([-5 0])
##       drawnow
##       frame = getframe(1);
##       im = frame2im(frame);
##       [imind,cm] = rgb2ind(im,256);
##       if n == 1
##           imwrite(imind,cm,filename,'gif', 'Loopcount',inf);
##       else
##           imwrite(imind,cm,filename,'gif','WriteMode','append');
##       end
## end
 figure;
 surf(xgrid,ygrid,pgrid')
 title('FEM Muzzle Blast Scaling')
 zlabel('Peak Reflected Overpressure [psi]')
 ylabel('Direction of Fire [in]')
 xlabel('Orthogonal Distance from Bore Centerline [in]')
 figure;
 surf(xgrid,ygrid,tagrid')
 title('FEM Muzzle Blast Scaling')
 zlabel('Time of Arrival [s]')
 ylabel('Direction of Fire [in]')
 xlabel('Orthogonal Distance from Bore Centerline [in]')
 figure;
 surf(xgrid,ygrid,taugrid')
 title('FEM Muzzle Blast Scaling')
 zlabel('Positive Phase Duration [s]')
 ylabel('Direction of Fire [in]')
```

```
xlabel('Orthogonal Distance from Bore Centerline [in]')
% ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%% Write Load Curves based on application of load
if loadtype == 0
    % *Load_Node
    SF = -1*ones(1,numel(ElemID));
    DOF = 4*ones(1,numel(ElemID));
    CID = zeros(1,numel(ElemID));
    fid = fopen('includeBlastLoads.k','w+');
    fprintf(fid,'*KEYWORD\n');
    for ii = 1:size(p,1)
        fprintf(fid,'*SET_NODE\n');
        fprintf(fid,'%10.0u%10.0u%10.0u%10.0u%10.0u%10.0u%10.0u%10.0u\n', ElemID(ii)+100000,0,0,0,0,0,0,0);
        if n(ii) == 3
            fprintf(fid,'%10.0u%10.0u%10.0u%10.0u%10.0u%10.0u%10.0u%10.0u\n', ElemNodes(ii,1:3),0,0,0,0,0);
        elseif n(ii) == 4
            fprintf(fid,'%10.0u%10.0u%10.0u%10.0u%10.0u%10.0u%10.0u%10.0u\n', ElemNodes(ii,1:4),0,0,0,0);
        end
        fprintf(fid,'*DEFINE_CURVE\n');
        fprintf(fid,'%10.0u%10.0u%10.0u%10.0u%10.5f%10.0u%10.0u%10.0u\n', ElemID(ii),0,1,1,tsuba(ii)-min(tsuba),0,0,0);
        fprintf(fid,'%20.4f%20.4f\n',[t;(p(ii,:)*A(ii))/n(ii)]);
    end
    tab = [double(ElemID)+100000,DOF',double(ElemID),SF',CID',double(ElemNodes(:,1)),double(ElemNodes(:,2)),double(ElemNodes(:,3))];
    fprintf(fid,'*LOAD_NODE_SET\n');
```

```
    fprintf(fid,'%10.0u%10.0u%10.0u%10.1i%10.0u%10.0u%10.0u%10.0u\n',tab');
    fprintf(fid,'*END');
elseif loadtype == 1
    % *Load_Segment for Element
    SF = ones(1,numel(ElemID));
    n5 = zeros(1,numel(ElemID));
    fid = fopen('includeBlastLoads.k','w+');
    fprintf(fid,'*KEYWORD\n');
    for ii = 1:size(p,1)
        fprintf(fid,'*DEFINE_CURVE\n');
        fprintf(fid,'%10.0u%10.0u%10.0u%10.0u%10.5f%10.0u%10.0u%10.0u\n',
   ElemID(ii),0,1,1,tsuba(ii)-min(tsuba),0,0,1000);
        fprintf(fid,'%20.4f%20.4f\n',[t;p(ii,:)]);
    end
    tab = [double(ElemID),SF',n5',double(ElemNodes(:,1)),double(ElemNodes(:,2)
   ),double(ElemNodes(:,3)),double(ElemNodes(:,4)),n5'];
    fprintf(fid,'*LOAD_SEGMENT\n');
    fprintf(fid,'%10.0u%10.1f%9.4E%10.0u%10.0u%10.0u%10.0u%10.1u\n',tab');
    fprintf(fid,'*END');
else
    warning('unknown load application value')
end
fclose(fid);
```

C.2 Mesh Convergence Script

```
% This script reads in nodal displacements, calculates the maximum
% resultant displacement and conducts calculations for mesh refinement
% analysis of a muzzle blast scaling finite element model.
% Xavier Quinn, rev 20Jan2021
```

```
5
6  %% Input
7  inputdirs = {'C:\Analysis\Blast\MeshTest\LOADNODE_Incident';...
8      'C:\Analysis\Blast\MeshTest\LOADNODE_Reflected';...
9      'C:\Analysis\Blast\MeshTest\LOADSEGM_Incident';...
10     'C:\Analysis\Blast\MeshTest\LOADSEGM_Reflected'};
11
12 %% Loop through load cases
13 for kk = 1:size(inputdirs,1)
14     cd(inputdirs{kk,:})
15     clearvars -except inputdirs r kk
16
17     %% Read in ASCII Data - NODOUT File
18     filename = 'nodout';
19     fid = fopen(filename);
20     fmt = ('%u %f %f %f %f %f %f %f %f %f %f %f %f');
21     idx = 1;
22     while ~feof(fid)
23         str = fgetl(fid);
24         if numel(str) >= 11 && strcmpi(' n o d a l ',str(1:11))
25             [~,EInd] = regexp(str,'at time ','once');
26             t(idx) = str2double(str(EInd+1:EInd+13));
27             temp = fgetl(fid);
28             temp = fgetl(fid);
29             N = textscan(fid,fmt);
30             dspX(:,idx) = N{1,2};
31             dspY(:,idx) = N{1,3};
32             dspZ(:,idx) = N{1,4};
33             VelX(:,idx) = N{1,5};
34             VelY(:,idx) = N{1,6};
35             VelZ(:,idx) = N{1,7};
```

```
            AccX(:,idx) = N{1,8};
            AccY(:,idx) = N{1,9};
            AccZ(:,idx) = N{1,10};
            idx = idx +1;
        end
    end
    fclose(fid);

    %% Calculate resultant displacement
    for ii = 1:4
        dspR(ii,:) = sqrt(dspX(ii,:).^2+dspY(ii,:).^2+dspZ(ii,:).^2);
        accR(ii,:) = sqrt(AccX(ii,:).^2+AccY(ii,:).^2+AccZ(ii,:).^2)./386;
    end
    clearvars N

    %% Calculate Convergence Rate Exponent from displacement
    % Mesh tripplet begins at coarsest mesh, F1, to finest mesh F3.
    F1 = max(dspR(3,:));
    F2 = max(dspR(2,:));
    F3 = max(dspR(1,:));

    %% Order of Accuracy Calculations
    pf = 2; % Formal order of accuracy
    r = 2; % Mesh refinement ratio
    p = log((F1-F3)./(F1-F2))./log(r); % Observed order of accuracy

    %% Estimate Numerically Exact Solution
    F0 = F1 + (F1-F2)./(r^p-1);

    %% Calculate Grid Convergence Index
    if abs((p-pf)/pf) <= 0.1
```

```
67          Fs = 1.25; % Factor of safety
68          p_GCI = pf;
69          GCI(ii) = Fs/(r^p_GCI-1)*abs(F3-F2);
70      else
71          Fs = 3;
72          p_GCI = min(max(0.5,p),pf);
73          GCI(ii) = Fs/(r^p_GCI-1)*abs(F3-F2);
74      end
75
76      %% Calculate Relative Error to Estimated Numerically Exact Solution
77      RelErr(1,:) = 100*abs(F0 - max(dspR(1,:)))/F0;
78      RelErr(2,:) = 100*abs(F0 - max(dspR(2,:)))/F0;
79      RelErr(3,:) = 100*abs(F0 - max(dspR(3,:)))/F0;
80      RelErr(4,:) = 100*abs(F0 - max(dspR(4,:)))/F0;
81
82      %% Plots
83      % Label Plots with what load type
84      if strcmp(inputdirs{kk,1}(32:35),'NODE') && strcmp(inputdirs{kk,1}(37:end)
    ,'Incident')
85          tempTitle = 'Point Load, Incident Pressure';
86      elseif strcmp(inputdirs{kk,1}(32:35),'SEGM') && strcmp(inputdirs{kk,1}(37:
    end),'Incident')
87          tempTitle = 'Pressure Load, Incident Pressure';
88      elseif strcmp(inputdirs{kk,1}(32:35),'NODE') && strcmp(inputdirs{kk,1}(37:
    end),'Reflected')
89          tempTitle = 'Point Load, Reflected Pressure';
90      else
91          tempTitle = 'Pressure Load, Reflected Pressure';
92      end
93
94      % Plot Resultant displacement
```

```
95      figure; hold on;
96      plot(1000:size(dspR,2),dspR(1,1000:end))
97      plot(1000:size(dspR,2),dspR(2,1000:end))
98      plot(1000:size(dspR,2),dspR(3,1000:end))
99      plot(1000:size(dspR,2),dspR(4,1000:end))
100     xlabel('Timestep'); ylabel('Resultant Displacement [in]');
101     title({tempTitle;'Nodal Displacements'})
102     legend('2 Inch','1 Inch','1/2 Inch','1/4 Inch')
103     hold off;
104
105     % Plot Relative Error
106     figure; hold on;
107     bar(RelErr)
108     ylabel('Relative Error [%]');
109     xticks([1,2,3,4]);
110     xticklabels({'2 Inch','1 Inch','1/2 Inch','1/4 Inch'})
111     title({tempTitle;'Relative error of max resultant displacement';'to
    numerically exact estimation'; 'at center node of the panel'})
112     hold off;
113         %% Read in ASCII Data - ELOUTDET File (local strains)
114     %{
115     clearvars -except inputdirs r kk
116     filename = 'eloutdet';
117     fid = fopen(filename);
118     fmt1 = ('%d- ');
119     fmt2 = ('%s %s %f %f %f %f %f %f');
120     idx = 1;
121     nodeID = [10169;20625;32401;49409];
122     element = [];
123     while ~feof(fid)
124         str = fgetl(fid);
```

```
125         if numel(str) >= 24 && strcmpi(' n o d a l  s t r a i n ',str(1:24))
126             [~,EInd] = regexp(str,'at time ','once');
127             t(idx) = str2double(str(EInd+1:EInd+11));
128             temp = fgetl(fid);
129             temp = fgetl(fid);
130             temp = fgetl(fid);
131             temp = fgetl(fid);
132             while numel(temp) > 0 && ~feof(fid)
133                 nodeData = str2double(temp(1:end-1));
134                 tempstrain = textscan(fid,fmt2,2);
135                 strainArray = [tempstrain{1,3},tempstrain{1,4},tempstrain
    {1,5},...
136                     tempstrain{1,6},tempstrain{1,7},tempstrain{1,8}];
137                 ind = nodeData == nodeID;
138                 if sum(ind) == 1
139                     upperbytime(ind,:,idx) = strainArray(2,:);
140                     lowerbytime(ind,:,idx) = strainArray(1,:);
141                 end
142                 temp = fgetl(fid);
143                 temp = fgetl(fid);
144             end
145             idx = idx +1;
146         end
147     end
148     fclose(fid);
149     % Calculate Convergence Rate Exponent from displacement
150     Make sure the appropriate strain components are being compared based on
151     their output coordinate system
152     F1 = squeeze(upperbytime(4,1,:));
153     F2 = squeeze(upperbytime(3,3,:));
154     F3 = squeeze(upperbytime(2,1,:));
```

```
155     F4 = squeeze(upperbytime(1,3,:));
156     mesh tripplet begins at finest mesh, F1, to coarsest mesh F-3
157     p = log((F1-F3)./(F1-F2))./log(r);
158 p = 2;
159 Fs = 3;
160 GCI = Fs/(r^p-1)*abs(F2-F1);
161     % Estimate Numerically Exact Solution
162     F0 = F1 + (F1-F2)./(r^max(p)-1);
163 
164     RelErr(1,:) = (F0 - F1)./max(F0);
165     RelErr(2,:) = (F0 - F2)./max(F0);
166     RelErr(3,:) = (F0 - F3)./max(F0);
167     RelErr(4,:) = (F0 - F4)./max(F0);
168     %%
169     figure; hold on;
170     plot(F1(1000:end))
171     plot(F2(1000:end))
172     plot(F3(1000:end))
173     plot(F4(1000:end))
174     plot(real(F0(1,1000:end)))
175     xlabel('Timestep'); ylabel('Nodal Strain [in]');
176     title({[inputdirs{kk,1}(28:36),' ',inputdirs{kk,1}(37:end)];'Nodal Strain'
        })
177     legend('2 Inch','1 Inch','1/2 Inch','1/4 Inch','Numerically Exact
        Estimation')
178     hold off;
179 
180     figure; hold on;
181     plot(1000:size(RelErr,2),RelErr(1,1000:end))
182     plot(1000:size(RelErr,2),RelErr(2,1000:end))
183     plot(1000:size(RelErr,2),RelErr(3,1000:end))
```

```
184     plot(1000:size(RelErr,2),RelErr(4,1000:end))
185     xlabel('Timestep'); ylabel('Relative Error [pct]');
186     title({[inputdirs{kk,1}(28:36),' ',inputdirs{kk,1}(37:end)];'Relative
        Error to Numerically Exact Estimation'})
187     legend('2 Inch','1 Inch','1/2 Inch','1/4 Inch')
188     hold off;
189 %}
190 end
191 cd('C:\Users\xavier.quinn\Documents\MATLAB\ThesisCode\ModelingFiles')
```