A Finite Volume Approach to Modeling

Injury Mechanisms of Blast-Induced Traumatic Brain Injury

Randall Wade Hoberecht

A thesis

submitted in partial fulfillment of the

requirements for the degree of

Master of Science

University of Washington

2009

Program Authorized to Offer Degree:

Applied Mathematics

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this thesis is allowable only for scholarly purposes consistent with "fair use" as prescribed by the U.S. Copyright Law. Any other reproduction for any purposes or by any means shall not be allowed without my written permission.

Signature _____

Date _____

University of Washington

Graduate School

This is to certify that I have examined this copy of a master's thesis by

Randall W. Hoberecht

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

_____

Randall J. LeVeque

_____

J. Nathan Kutz

Date:_____

University of Washington

Abstract

A Finite Volume Approach to Modeling

Injury Mechanisms of Blast-Induced Traumatic Brain Injury

Randall Wade Hoberecht

Chair of the Supervisory Committee:
Professor Randall J. LeVeque
Department of Applied Mathematics

The development of improved personal protective armor has increased the survivability of

the modern soldier. Increased survivability against kinetic threats has resulted an increase in

the rate of surviving casualties with blast-induced Traumatic Brain Injury (bTBI).   The purpose

of this work is the development of a finite volume model, using equations of elasticity, to

model the propagation of pressure and shear waves through the brain resulting from

exposure to an explosive blast. The primary objectives of this research are to provide insight

into the complex nature of Traumatic Brain Injuries and aid in the development of higher

fidelity models. This work examines critical aspects for effective implementation of finite

volume methods, using CLAWPACK (Conservation LAWs PACKage), to numerically solve the

hyperbolic system of elastic equations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

DEDICATION

This work is dedicated to the soldiers who volunteer their service for an ideal greater than themselves and knowingly assume risks beyond any reasonable expectation. May we relentlessly pursue every means to provide for their well-being.

Introduction

> "Far better an approximate answer to the right question, which
> is often vague, than the exact answer to the wrong question,
> which can always be made precise."

> - John W. Tukey, 1962, *The Future of Data Analysis*

This thesis is the culmination of an initial investigation into the utility of Finite Volume methods for modeling shock wave propagation from air into biological materials. Specifically, this effort seeks to answer the question: can the implementation of Finite Volume methods, used to numerically solve the equations of linear elasticity, effectively model the propagation of shock waves caused by explosive blasts through the human head? The purpose of this work is to gain insight into the development of numerical models of shock wave propagation through the human head as a mechanistic cause of blast-induced Traumatic Brain Injury (bTBI).

There are two primary approaches to the research of blast injuries. One approach is the study of generalized blast response, of the entire body's response to a blast wave. This approach typically involves exposure of test animals to simulated or actual explosive blast waves. Due to the complex interaction between physical responses, it is very difficult to isolate the contribution of specific causes to the victim's overall condition. The second approach is to isolate a specific injury mechanism in order to understand its contribution to the blast injury.[1] While the former approach is important in understanding the overall susceptibility of humans to blast injuries, the latter approach is critical in understanding the contribution from individual mechanisms. The current focus of this research is in line with the latter approach, by attempting to isolate the pressure wave as a single injury mechanism. Care must be taken to ensure that results from either method are properly framed by the limitations of the approach.

An effective numeric model will be valuable in understanding the extent of injuries which can occur under a variety of conditions, and can provide insight into evaluating potential methods for reducing the occurrence and severity of such injuries. For example, the model could be used to evaluate materials and design of protective armor. Finite volume methods are well suited for this particular problem due to their ability to maintain accuracy where discontinuities exist in the solution; significant discontinuities in pressure are associated with the shock condition at the leading edge of the pressure wave generated by an explosive blast. While such discontinuities cause numerical difficulties when implementing finite difference methods, the conservative nature of finite volume methods provides a foundation for maintaining accuracy and stability.

This thesis primarily addresses two critical aspects of developing a finite volume numeric model. First, the model must accurately represent the complex geometry and material properties of the head and components therein. Second, the model should be able to effectively incorporate the appropriate range of pressure waveforms that match the conditions associated with occurrence of bTBI (i.e. magnitude, form, and duration). The finite volume methods implemented in this study are based on the material presented by Randy LeVeque in, *Finite Volume Methods for Hyperbolic Problems*, and implemented with the use of CLAWPACK (Conservation LAWs PACKage), a software application based on the methodologies outlined in the book.[2] The scope of work presented in this thesis is based on gaining fundamental understanding of how finite volume methods can be applied to the problem, and is limited primarily to two dimensions with simplified geometric representation of the skull and brain. Incorporation of increasingly accurate geometries and extension to three dimensions is the intended continuation of this research effort.

An underlying assumption in this effort is that useful results may be attained through limiting the domain of interest to the head and immediate surrounding area. This allows a significant reduction in computational cost by excluding numerical modeling of the detonation of an explosive or the interaction of the pressure wave with other body parts. Although some researchers hypothesize that a significant contribution to brain injury may be pressure waves initiated in the thoracic cavity, which propagate through the vascular system directly into the brain,[3,4] a reduced model including only the head may contribute to determination of the feasibility of these hypotheses; a failure of the reduced model to provide results consistent with the expected severity of injury would indicate the presence of other sources leading to injury. Additional hypothesized injury mechanisms associated with the blast wave include cavitations resulting from the period of relative negative present in an explosive blast pressure wave, as well as the presences of electromagnetic pulses generated by the blast.[5] It is foreseeable that the finite volume method could predict the occurrence of conditions leading to cavitation based on the presence of negative relative pressures within the brain.

This research supports current efforts at the Veterans' Affairs Puget Sound Medical Center in the study of bTBI, directly related to critical knowledge gaps identified during the May 2009 International State-of-the-Science Meeting on Non-Impact Blast-Induced Mild Traumatic Brain Injury. Two essential areas for current research identified in the meeting are:[6]

- "Clarify the pathophysiology in laboratory/computational modeling and clinical experiments."
- "Develop scaling relationships for animal models and validate models for human injury."

Background

According to the definition adhered to by the Brain Injury Association of America, Traumatic brain injury is:

> "an insult to the brain, not of a degenerative or congenital nature but caused by an external physical force, that may produce a diminished or altered state of consciousness, which results in an impairment of cognitive abilities or physical functioning. It can also result in the disturbance of behavioral or emotional functioning. These impairments may be either temporary or permanent and cause partial or total functional disability or psychosocial maladjustment."[7]

Traumatic Brain Injuries are not uncommon, with an estimated 1.4 million occurring each year in the United States.[8] Historically, relatively few of these injuries have been blast-induced; as a result, past TBI research efforts have focused primarily on understanding TBI resulting from the more common causes of sports, vehicle accidents and falls, and have not addressed blast-induced injuries to the same degree of academic rigor. The conduct of current military operations, in which our adversaries favor the use of land mines and remotely detonated explosives to direct confrontation, as well as increased armor effectiveness and immediate medical response capability, has led to an increased emphasis on the study of bTBI. A study recently completed in 2008 of approximately 2500 soldiers, representing two infantry brigades, following deployment to Iraq reported that 4.9% of the soldiers sustained injuries with loss of consciousness, and 10.3% reported injuries with altered mental status.[9] Although the actual number of soldiers with bTBI is a subject of some controversy, its prevalence has spurred significant research efforts throughout the country.

Injuries that result from explosive blasts are categorized into four categories; a victim of blast injury, however, is likely to sustain injuries in more than one, if not all, of the categories, making diagnosis of specific injury mechanisms extremely difficult. These categories are described as follows:

Primary: These injuries are a direct result of overpressure on the organs within the body. Due to the compressibility of air, primary blast injuries are typically associated with air-filled structures such as the lung, ear, and gastrointestinal tract; however, this does not preclude primary injury to other organs such as the brain.

Secondary: Explosive blasts generate flying debris (shrapnel from the explosive casing, or other objects near the blast) that causes this type of injury when it impacts people.

Tertiary: This type of injury occurs when a body, accelerated by the explosive blast, impacts a stationary structure or other object.

Miscellaneous: These injuries are all those that do not meet the criteria of the first three categories. Examples of miscellaneous injuries include: burns, crush injuries from structure collapse, and toxic inhalations.

Blast-induced TBI has previously been viewed as a result of secondary and tertiary blast injuries, however, recent research suggest that primary blast injury is likely a significant cause of bTBI.[10] Brain injury caused by secondary or tertiary blast injury mechanisms can be more closely compared to typical non-blast TBIs such as those resulting from the impacts and rapid head accelerations associated with automobile or sports-related accidents. Effective prevention of bTBI must address the contribution of all mechanisms of injury. Unfortunately, protective measures are not necessarily complementary, in that measures to protect against specific threats may increase susceptibility to other threats. A numerical study of bTBI conducted at Lawrence Livermore National Laboratories suggested that ballistic helmets, while protecting from secondary and tertiary injury, have the potential to enhance pressure waves between the helmet and skull, increasing the potential of primary blast injury to the brain.[11] Other experiments have demonstrated that the ballistic flak vest can actually increase the peak overpressure reaching the thorax under certain conditions.[12]

TBIs are categorized by severity as mild, moderate or severe, based on the associated duration of loss of consciousness (LOC) and post-trauma amnesia (PTA). Mild TBI is associated with LOC less than 30 minutes and PTA less than 24 hours; moderate TBI is based on LOC greater than 30 minutes but less than 24 hours and PTA between 1 and 7 days; severe TBI is anything greater.[13] Mild TBI is often difficult to diagnose because it often occurs in conjunction with more severe injuries, or in combination with other psychological disorders such as depression or post traumatic stress disorder (PTSD).[14] In fact, mild TBI and PTSD share many symptoms in common: insomnia, memory problems, poor concentration, depression, anxiety, and irritability. Distinguishing symptoms for bTBI are: headache, dizziness, fatigue and noise/light intolerance; while those suffering PTSD will exhibit: stress, emotional numbing, flashbacks, and nightmares.[15]

One particular challenge in current bTBI research efforts is the determination of appropriate biological material properties. Because most TBI research has been directed at non-blast injuries, typically blunt force trauma, in which the associated rate of strain is significantly ("orders of magnitude") smaller than those experienced due to blasts, extrapolation to the higher strain rates of blast injury may introduce error into the model. Currently, relatively little is known about the visco-elastic behavior of the brain at the higher strain rates associated with blast injuries. This is an important area of current research.[16]

Chapter 1: Explosive Blasts in Air

The detonation of an explosive generates a shock wave when the rate of combustion of the explosive material creates a pressure gradient sufficient that the speed of sound behind the shock front exceeds the speed of sound in front of the shock. Explosive materials such as trinitrotoluene (TNT), are characterized by a large amount of chemical potential energy that is nearly instantaneously converted to kinetic and heat energy during the process of detonation; the detonation speed for TNT is approximately 6900m/s. Although the amount of energy varies between explosives, TNT is commonly used as a reference for comparison. Through experimentation, the energy contained in TNT has been demonstrated to be approximately 4680 Joules/gram.[*][17] As the energy of the blast dissipates with increasing distance from the blast, the wave dissipates into a sound wave.

A typical pressure profile of a wave generated by an explosive blast is illustrated in Figure 1. Characteristics of the pressure profile are an initial jump discontinuity from ambient pressure, p_a, to a maximum overpressure, p_o. This is followed by a rapid decay of the positive overpressure back to the ambient level within a few milliseconds. The inertia of the motion of the expanded and displaced gas results in a negative pressure phase that lasts approximately three times as long as the positive phase, before returning to ambient pressure.[18] Further pressure oscillations are possible, but are not typically observed for blasts in air.

The self-similarity of the solution at different scales is an important characteristic of the blast wave pressure profile that aids in modeling. Within a broad range of conditions, the explosive blast waves are identical except for scaling in magnitude and duration. The primary exceptions to the scalability of blast pressure waves are ranges extremely close or far from the
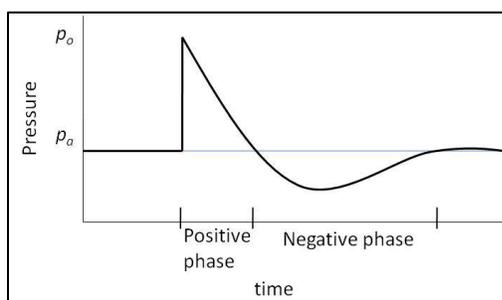
Figure 1: Typical pressure profile for an explosive blast in air. The graphic depicts the major characteristics of a blast wave: near instantaneous rise to maximum pressure, rapid decline back to ambient levels during positive phase, negative phase with pressure below ambient level lasting three times as long as the positive phase, and leveling off at ambient level.

---

[*]This value differs slightly from the definition for a standard ton of TNT used as a reference during early nuclear weapon development. At that time, one ton of TNT was defined as the energy release of one million kilocalories. To put this energy in physical perspective, 1 Joule is equivalent to the kinetic energy

point of detonation, and nuclear blasts. A mathematical description of the pressure waveforms that represent the typical ranges and sizes of explosives associated with bTBI can be accomplished by scaling from a reference explosion (typically 1kg of TNT) to the desired size of explosive and distance from the explosion. The scaling equations are a system of empirically developed equations, tailored for accuracy within specific ranges from the blast, as well as the type of blast (nuclear or conventional).[19] These equations are described in G.F. Kinney and K.J. Graham's book, *Explosive Shocks in Air*. Components of these equations critical to this research are summarized in this section.[20]

The first step in determining appropriate waveform is to determine a scaled distance, Z, that relates the modeled (actual) distance to a reference explosion (1kg TNT):

$$Z = \frac{f_d \times (actual\ \ distance\ )}{W^{1/3}} \tag{1.1}$$

In this calculation, W is the explosive yield, in kilograms of TNT, of the modeled explosive. The parameter, f_d, is considered the transmission factor, and determined by:

$$f_d = \left(\frac{\rho}{\rho_0}\right)^{1/3} \tag{1.2}$$

In the equation for the transmission factor, the ratio, ρ/ρ_0, is the ratio of air density at the point of interest to the air density of the location of the explosion. This value is typically one, except in cases of airburst, where an explosion occurs at a significant altitude difference from the point where the pressure profile is to be calculated. Most explosions leading to bTBI occur in near proximity to the victim, and therefore a transmission factor of $f_d = 1$ is assumed for this study.

The maximum overpressure, p_0, can then be calculated by:

$$p_0 = p_a \frac{808\left[1+\left(\frac{Z}{4.5}\right)^2\right]}{\sqrt{1+\left(\frac{Z}{0.0048}\right)^2}\sqrt{1+\left(\frac{Z}{0.32}\right)^2}\sqrt{1+\left(\frac{Z}{1.35}\right)^2}} \tag{1.3}$$

As previously noted, the speed that the shock front propagates is faster than the speed of sound. The speed of the shock front, as a mach speed, can be calculated at a specified point by (The variable, k, is the ratio of heat capacities, typically 1.4 for air):

$$M_x = \sqrt{1 + \frac{(k+1)p_0}{2kP_a}} \tag{1.4}$$

In the physical blast wave, the shock front is a region of finite thickness in which the physical properties of the air (pressure, temperature, velocity, etc) change at such a rate, that the front can be modeled as a mathematical discontinuity. The Rakine-Hugoniot relationships are a system of equations, based on conservation laws, which express the change in conditions

across the shock front, to include pressure, velocity, density, temperature, and sound speed. If x and y represent the two sides of the shock front, then the following relationship holds:

$$\frac{P_x u_x}{R T_x} = \frac{P_y u_y}{R T_y},$$

(1.5)

in which P is the absolute pressure, u is the velocity of the media, R is the specific gas law constant[†], and T is the absolute temperature. The thickness of the shock front increases as the distance from the blast increases and the waveform degenerates into a sound wave. This thickness can be estimated by:

$$Thickness(meters) = \frac{11+7M_x}{\rho_x(M_x-1)} \times 10^{-8}$$

(1.6)

Based on parameter values consistent with exposure leading to bTBI, this thickness is on the order of 10^{-3} mm. Using a uniform Cartesian grid, it would require a grid resolution of 10,000 cells per centimeter in order to resolve the discontinuity of the shock front across a single grid cell, creating unreasonable computational cost.

The two most critical components of the wave pressure profile that describe the potential of a wave to cause injury to personnel, or damage to structures, are the maximum overpressure and the duration of the positive phase. Together, these can be used to estimate the energy impulse associated with the blast. The impulse is defined as the integral of the pressure-time curve (Figure 1). The duration of the positive phase of the pressure profile is estimated by:

$$t_d = W^{\frac{1}{3}} \frac{980\left[1+\left(\frac{Z}{0.54}\right)^{10}\right]}{\left[1+\left(\frac{Z}{0.02}\right)^{3}\right]\left[1+\left(\frac{Z}{0.74}\right)^{6}\right]\sqrt{1+\left(\frac{Z}{6.9}\right)^{2}}}$$

(1.7)

In structural effects modeling, the blast waves are often modeled as a triangular pressure profile based only on the maximum overpressure and the positive phase duration. In structural analysis, this tends to provide a worst-case estimate of the destructive potential of the blast, because the negative phase can reverse the displacement to structural pieces caused by the positive phase; however, in a biological model, the negative pressures could potentially lead to cavitation.[21] The actual waveform, follows a curve approximated by the Friedlander equation:

$$p = p^0 \left(1 - \frac{t}{t_d}\right)e^{-\alpha\frac{t}{t_d}}$$

(1.8)

[†] For dry air, $R \approx 287.058 \frac{J}{kg \cdot K}$

The parameter, α, is the waveform parameter which is experimentally determined and available in reference tables for most conditions (i.e. Kinney and Graham, table XI). This curve is only valid for the positive phase of the pressure profile; the exact form of the negative phase is dependent on, and limited by, the ambient pressure.

The range of possible exposure to explosive blasts in current military operating environments is nearly unlimited. Improvised explosive devices range small man-carried devices less than one kilogram to large trucks capable of carrying in excess of 25,000 kilograms of explosives.[22] Additionally, some explosive weapons are designed specifically to maximize the energy of the pressure wave. These thermobaric weapons are characterized by slower detonation speeds, which results in a lower peak overpressure but significantly longer positive phase duration, increasing the associated impulse. Unlike conventional high explosives, thermobaric weapons typically rely on oxygen from the atmosphere for combustion; these are more commonly referred to as fuel-air explosives. Although the radius of physical destruction is limited, they are particularly effective against personnel in caves or other protective cover from ballistic threats.[23]

The key characteristics of the pressure profile, as calculated by the equations described above, are illustrated in Figure 2, which depicts the positive phase of the pressure wave as a function of time and distance (for 10kg TNT at ranges from 5 to 15 meters). This illustrates the initial cubic rate of decay for maximum pressure, decreasing mach speed, and increasing overpressure duration. Actual pressure histories using data from pressure sensors and actual explosive blasts have a much more erratic appearance, and deviate from this in some respects.
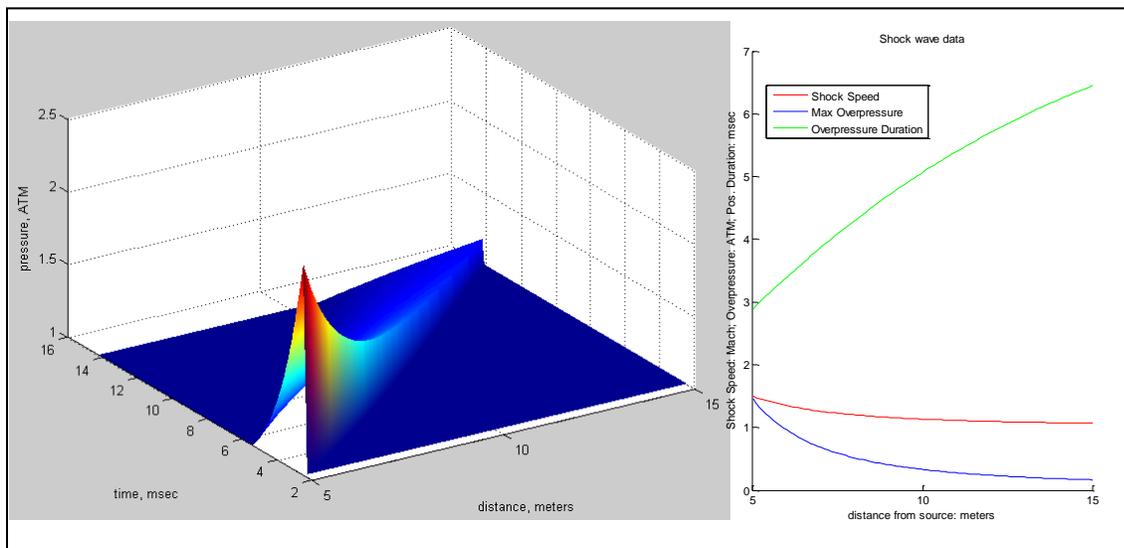
Figure 2: Plots representing open-air pressure profile of 10kg TNT at ranges from 5 to 15 meters from point of detonation. Left: Positive phase pressure profile as a function of time and distance based on ambient pressure of 1 atmosphere. Right: relationship of shock speed, maximum pressure, and duration of positive phase.

In addition to the error associated with sensitivity of sensors, data collected from a sensor placed a point will record the pressure history associated with numerous waves reflected off of structures or terrain. Additionally, it should be noted that these equations are based on unconfined detonation of TNT. The presence of a shell casing effectively reduces the amount of energy contributing to the pressure wave, by requiring energy to fracture and accelerate the physical structure of the casing, as shrapnel. If the explosive blast occurs over the ground, there is also the potential of the formation of a mach stem, when the shock wave reflected off of the ground merges with the shock wave propagating above the ground. This effect can increase the total shock pressure in the region affected by the mach stem. MATLAB Code used to create the plots in Figure 1 is included in the Appendix, MATLAB: Explosive Blast Waveform.

Chapter 2: Materials and the Human Response

The complexity of modeling the human head is expressed by Andrzej Przekwas, in his discussion on Multi-scale Computational Modeling of Lung Blast Injuries:

> "In spite of relatively uniform density and protective barriers, including the scalp, skull, meninges, and subarachnid cerebrospinal fluid, the brain is also susceptible to blast wave injuries. Highly anisotropic material properties in the brain and immense vascular perfusion will result in nonuniform absorption of the wave energy, stretching and breaking neural axons and the capillary blood brain barrier. Other homogeneous solid viscera transmit the pressure wave to the distal side of the body and are much less susceptible to blast wave injury. In general, the risk of injury is related to the blast wave energy delivered to the body and absorption by various tissues."[24]

Section 2.1: Material Properties

Accurate numerical modeling requires that the model reflects the physical geometry of the object as well as the material make-up of the object. For purposes of numerical modeling of linear elastic behavior, three parameters are required to describe the physical response of a material. These are the density of the material, ρ, and the Lamé parameters, λ and μ. The parameter μ is the shear modulus (commonly, the variable G is also used for this variable), and describes the linear restoring force (shear stresses; σ^{12}, σ^{23}, and σ^{13}) of a material to shear strains (ϵ^{12}, ϵ^{23}, and ϵ^{13}).[25]

$$\mu = \frac{\sigma^{12}}{2\epsilon^{12}} = \frac{\sigma^{23}}{2\epsilon^{23}} = \frac{\sigma^{13}}{2\epsilon^{13}} \qquad (2.1.1)$$

The parameter, λ, defines a relationship between the modulus of elasticity (Young's Modulus), E, and Poisson's Ratio, ν.

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \qquad (2.1.2)$$

The modulus of elasticity, E, is a measure of a material's linear response to axial forces such that $E = \frac{\sigma}{\epsilon}$, and is measured in units of pressure. E can also be expressed in terms of λ and μ as :

$$E = \frac{\mu(3\lambda+2\mu)}{\lambda+\mu} \qquad (2.1.3)$$

Poisson's ratio, ν, is a dimensionless quantity that describes the response of a material to compressive forces. Typically, $0 < \nu < 0.5$, and ν must be less than 0.5 for hyperbolic formulation. A material in which $\nu = 0.5$ is considered incompressible.

$$\nu = -\frac{\epsilon^{22}}{\epsilon^{11}} = -\frac{\epsilon^{33}}{\epsilon^{11}} \tag{2.1.4}$$

Poisson's ratio is also used to relate Young's modulus and the shear modulus for linear elastic materials:

$$\mu = \frac{E}{2(1+\nu)} \tag{2.1.5}$$

With advances in measurement and imaging technology, researchers have made significant progress in their ability to model the human body. Current efforts to model the human body, and specifically the head, provide a wide range of references to build from in the conduct of this research, and give insight into appropriate material parameters. Specifically, significant research efforts have been made in developing a high-resolution finite element model of the human head. Giovanni Belingardi describes one such model based on continued improvements from the first models created over 30 years ago.[26] Belingardi's finite element model of the skull consists of 55,264 elements with 26,000 nodes, using geometry based on CT scans and MR images. In Belingardi's model, the skull is modeled as two layers of compact (cortical) bone, surrounding a thicker interior of cancellous (trabecular) bone. Cerebral spinal fluid (CSF) is also included in the model. Both the bone and CSF are modeled as linear elastics, while the brain is modeled as a visco-elastic material; additionally, there is a 'sliding surface' included between brain and bone.

Numerous other efforts have been made to model the human head. Sarkar, Majumder and Roychowdhury developed a finite element model of the head, in order to model responses under static and dynamic loading. They describe the complex geometry of the head, structured with eight bones of irregular shape and thickness (approximately 6mm) and a thicker plate of bone at the base (approximately 12 mm). "The brain is neither fluid nor solid but a rather gel like medium, partitioned into many recesses by the folds."[27] In their model, the brain is modeled as a linear elastic material. A study by Liying Zhang, et al., using Finite Element Modeling to model concussion injuries resulting in professional football, attempted to account for the anisotropic nature of brain material by implementing differing visco-elastic material properties to describe white matter, grey matter, and the brain stem as separate materials.[28] Other references for material properties of the head include: a thesis by S.J.A. Michielsen, in which he references material parameters used in models by Ruan (elastic model) and Chen (visco-elastic model);[29] and an elastic model used by Moore to model primary blast effects on the central nervous system.[30] A summary of the references listed above is contained in Table 1.

Table 1: Summary of various sources describing material properties

	Young's Modulus (E) (kPa)	Density (ρ) (kg/m^3)	Poisson's Ratio (ν)	Shear Modulus (μ) (kPa)	Bulk Modulus (K) (kPa)
Source:	Belingardi, finite element model with visco-elastic brain				
Cortical Bone	15x10^6	1800	0.21		
Trabecular Bone	4.5x10^6	1600	0.01		
CSF	1.2	1040	0.49		
Brain		1140		$167 + (490 - 167)e^{-.145t/ms}$	
Source:	Linear Elastic Model, Sarkar et al.				
Bone	6.5x10^6	1412	0.22	2.66x10^6	
CSF	66.7	1040	0.499		
Brain	66.7	1040	0.48		
Source:	Ruan (reference Michielsen) Elastic brain response				
Brain	66.7	1040	0.49		
Source:	Chen (reference Michielsen) Visco-elastic brain response				
Brain				$(32.8 - 16.2) + 32.8e^{-t/0.0069s}$	
Source:	Zhang, et al.				
White Matter		1040		$7.8 + (41 - 7.8)e^{-400t/s}$	2.19x10^6
Grey Matter		1040		$6.4 + (34 - 6.4)e^{-400t/s}$	2.19x10^6
Brain Stem		1040		$7.8 + (58 - 7.8)e^{-400t/s}$	2.19x10^6
Source:	Moore				
Skull	6.5x10^6	1412	0.22		
Brain		1040		22.53	2.19x10^6
CSF		1040		22.53	2.19x10^6

In order to best develop an understanding of the limitations of the finite volume methods, this study begins with the use of a very simplified geometric representation of the skull, consisting of a circular or oval skull of constant thickness surrounding brain, and isotropic and elastic material properties. This allows the initial study to focus on the fundamental characteristics of the method before increasing the resolution of the model. Continued study in this area will seek to increase the accuracy of the geometric representation of each of the components as well as the representation of all components contained within the head. Even extremely simplified representations provide potential for relevant study towards gaining understanding of bTBI. For example, the numerical experiment conducted by Lawrence

Livermore National Laboratories, mentioned in the Background, included only three materials, skull, brain and cerebral spinal fluid, and used an ellipsoid as the geometric shape representing the skull.[31] Even with these significant simplifications, they were able to gain relevant insight into the nature of bTBI injury mechanisms.

Consistent with the independent linear elastic modeling of efforts of Moore and Sarkar, initial material parameters for this study are summarized in Table 2. The parameter λ can be determined from equation 2.1.2; however, due to the sensitivity this equation to nearly incompressible materials ($\nu \approx 0.5$), such as brain, λ is calculated directly from the bulk modulus, K, where

$$K = \lambda + \frac{2}{3}\mu \qquad (2.1.6)$$

The resulting speed of sound, $c_p = \sqrt{(\lambda + 2\mu)/\rho}$, for bone and brain is consistent with experimental studies. A speed of sound model for cranial bone by Conner, et al, determined appropriate values for the speed of sound based on density of the bone; 2302.6m/s to 2298.1m/s for densities of from 1400kg/m³ to 1500kg/m³ respectively.[32] Other sources cite Connor's work, using a range of 1953-3948m/s for sound speed in the skull and 1572m/s for sound speed in brain.[33] The property of bone represents a combination of the actual layered structure of cortical and trabecular bone. Experimental determination of the modulus of elasticity for cortical bone is approximately 20GPa with Poisson's ratio between 0.28 and 0.43.[34,35]

In order to maintain consistency with units of measurement, units of pressure used in numerical experimentation are based on:

$$1kPa = 10\frac{kg}{cm \cdot s^2}$$

These values are selected for consistency with other similar research efforts, and intended for use in testing the accuracy and stability of the finite volume methods. As researchers continue to study the physical response of these materials, the appropriate values will likely be refined.

Table 2: Summary of material properties used as initial values in this study

Material	Modulus of Elasticity $\left(\frac{kg}{cm \cdot s^2}\right)$	Density $\left(\frac{kg}{cm^3}\right)$	Poisson's Ratio, (ν)	Shear Modulus $\left(\frac{kg}{cm \cdot s^2}\right)$	λ $\left(\frac{kg}{cm \cdot s^2}\right)$	c_p $\left(\frac{cm}{s}\right)$	c_s $\left(\frac{cm}{s}\right)$
Bone	6.5×10^7	1.412×10^{-3}	0.22	2.66×10^7	1.77×10^7	2.24×10^5	1.37×10^5
Brain	667	1.04×10^{-3}	0.48	225.3	2.19×10^7	1.45×10^5	465

Section 2.2: Human Susceptibility to Brain Injuries

There are various types of traumatic brain injury, as described by the Brain Injury Association of America. Those that are particularly relevant to the study of bTBI are described as follows:[36]

- Diffuse Axonal Injury: Usually caused by shearing forces within the brain, resulting from strong shaking or rotation of the head, and characterized by tearing of nerve tissue throughout the brain
- Concussion: The most common type of TBI, occurs as a result of sudden impact or momentum change. Blood vessels may stretch and cranial nerves may be damaged. This type of trauma may cause diffuse axonal injury as well. Damage resulting from concussion may not show up on diagnostic imaging.
- Second Impact Syndrome (Recurrent TBI): Subsequent injury to the brain before symptoms from an initial injury have healed.

Diffuse axonal injury is the most frequently related to mild TBI, with injuries most commonly occurring at the junction between grey and white matter (corticomedulary), internal capsule and deep grey matter.[37] Susceptibility to blast-induced TBI is increased due to the visco-elastic property of the brain and the extreme deformation rates associated with the shock front of the blast wave. For higher strain rates, the brain tissue becomes exponentially more brittle, requiring less strain before permanent damage occurs. Due to its incompressibility ($\nu \approx 0.5$), the brain is much more resilient to compression strains than to tensile and shear strains. The amount of strain required to tear brain tissue is estimated at 10% to 20%.[‡38]

Traumatic Brain Injury resulting from sports-related injuries has a longer history of study than bTBI. Zhang, et al., conduct numerical experiments using a finite element model to recreate helmet to helmet collisions in football, in order to determine the best indicators of injury based on comparison to actual outcomes. They concluded that shear stress in the midbrain of the brainstem was the best predictor based on their sampling (n=24), and estimated that shear stresses of 6.0, 7.8, and 10.0kPa in the midbrain correlated to 25%, 50%, and 80% probability of mild TBI diagnosis respectively. This work also referenced intracranial pressure tolerance criteria established by Ward, based on animal tests as well as analytic experimentation. In this study, serious brain injury is correlated to peak intracranial pressures greater than 235kPa, while peak pressures below 173kPa resulted in minor or no injury.[39]

Experimental data and analysis of actual exposure of personnel to blast waves, has provide researchers with a general understanding of the body's ability to withstand pressure

---

‡ In Granacher's book on TBI, he states, "Bone will break at a strain of 1%-2%, whereas brain and vascular tissue may not tear until 10%-20% strain is applied. On the other hand, it takes considerably more force to cause a 1%-2% strain in bone than it does to produce a 10%-20% strain in brain tissue."

waves associated with explosive blasts. These references are typically based on conditions that are easily diagnosed (e.g. eardrum rupture, lung damage, and death), and do not address the more complex nature of brain injuries. Figure 3 is an example of a chart originally produced in 1988 by the Government Printing Office.[40] Charts such as this provide a reasonable means to evaluate the survivability of a specified blast, however, they do not account for multiple exposures nor do they assess the vulnerability of the brain. In developing a model for bTBI, these charts are useful in determining appropriate waveforms for modeling. Similar charts illustrate the effects of reflecting surfaces resulting in increased overpressures and durations. A study completed by the Defense Atomic Support Agency illustrates that the same injury potential exists at approximately one-half the incident maximum overpressure if the subject is in close proximity to a solid surface perpendicular to the direction of the incoming blast wave.[41]
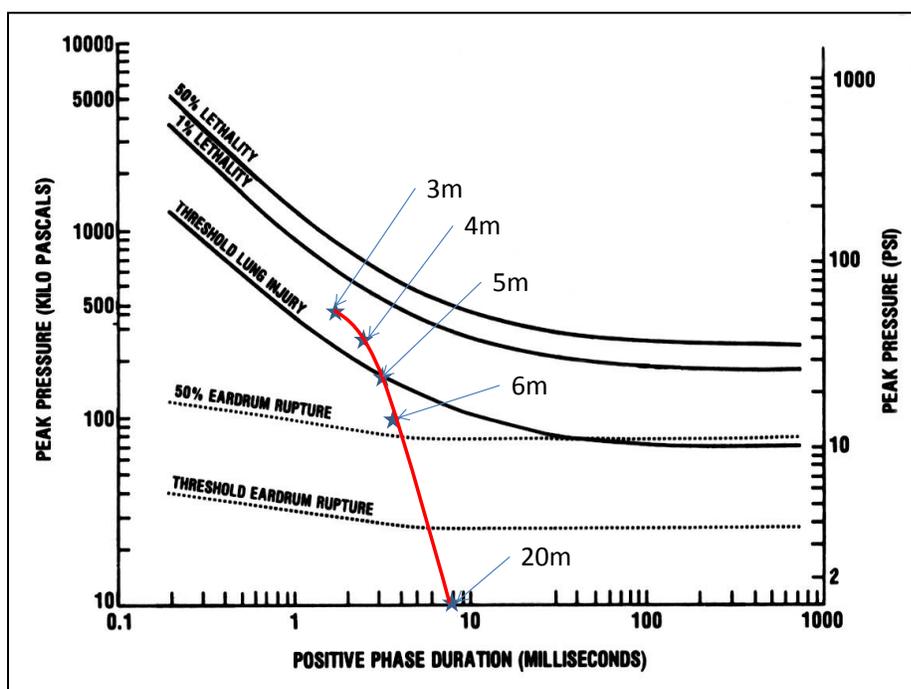
Figure 3: Human Injury Thresholds. Reprint of chart from Government Printing Office[40] depicting human thresholds for injury based on peak over-pressure and duration of the positive phase. Superimposed red line illustrates pressure and duration associated with 10kg TNT at distances from 3 to 20 meters from point of detonation.

Chapter 3: Equations of Linear Elasticity

Section 3.1: Linear Elastic System Derivation

The following is based on the derivation included by LeVeque in, *Finite Volume Methods for Hyperbolic Problems*.[42] Derivation of the differential equations governing conservative system of linear elasticity requires the assumption that displacement of any point is small enough that the restoring force of the material is linearly related to the deformation, Hooke's Law. Within the computational domain, this displacement can be described at a point by the vector $\vec{\delta}(x, y, z, t)$. Velocities, u, v and w, in the directions of the x, y and z axes respectively, are the time derivatives of displacement:

$$\delta_t^1 = u, \ \delta_t^2 = v, \ \delta_t^3 = w \tag{3.1.1}$$

The gradient of the displacement is used to define a strain tensor, ϵ, which includes components of normal strain, in the directions associated with a three-dimensional Cartesian mesh, as well as the associated shear strains. Because strain is determined from spatial derivatives of the actual displacement, strain is a dimensionless quantity.

$$\epsilon = \frac{1}{2}\left[\nabla\vec{\delta} + \left(\nabla\vec{\delta}\right)^T\right] = \begin{bmatrix} \epsilon^{11} & \epsilon^{12} & \epsilon^{13} \\ \epsilon^{21} & \epsilon^{22} & \epsilon^{23} \\ \epsilon^{31} & \epsilon^{32} & \epsilon^{33} \end{bmatrix} \tag{3.1.2}$$

For strains in the normal directions:

$$\epsilon^{11} = \delta_x^1, \ \epsilon^{22} = \delta_y^2, \ \epsilon^{33} = \delta_z^3, \tag{3.1.3}$$

and the shear strains are defined by:

$$\epsilon^{12} = \epsilon^{21} = \frac{1}{2}\left(\delta_y^1 + \delta_x^2\right), \quad \epsilon^{13} = \epsilon^{31} = \frac{1}{2}\left(\delta_z^1 + \delta_x^3\right), \quad \epsilon^{23} = \epsilon^{32} = \frac{1}{2}\left(\delta_y^3 + \delta_z^2\right) \tag{3.1.4}$$

From the spatial derivatives of 3.1.1 and time derivatives of 3.1.3 the following equalities are established (based on the equality, $(\delta_x^1)_t = (\delta_t^1)_x$):

$$u_x = \epsilon_t^{11}, \quad v_y = \epsilon_t^{22}, \quad w_z = \epsilon_t^{33} \tag{3.1.5}$$

Similarly, including the time derivatives of 3.1.4 yields:

$$\epsilon_t^{12} = \frac{1}{2}\left(v_x + u_y\right), \quad \epsilon_t^{13} = \frac{1}{2}(x_z + w_x), \quad \epsilon_t^{23} = \frac{1}{2}\left(w_y + v_z\right) \tag{3.1.6}$$

A stress tensor can also be defined, consisting of all the normal and shear stress components.

$$\sigma = \begin{bmatrix} \sigma^{11} & \sigma^{12} & \sigma^{13} \\ \sigma^{21} & \sigma^{22} & \sigma^{23} \\ \sigma^{31} & \sigma^{32} & \sigma^{33} \end{bmatrix} \tag{3.1.7}$$

The stress tensor represents all the internal forces acting at a given point. The total force per unit area acting on a point in a direction specified by a unit vector, $\vec{n}$, can be calculated by $\boldsymbol{\sigma} \cdot \vec{n}$. As an example, if a face is aligned with a standard Cartesian grid with $\vec{n}$ as a unit vector in the positive x direction, then σ^{11} represents the normal stress while σ^{21} and σ^{31} are the shear stresses.

The stress tensor represents the forces acting on each point, relationships between the stresses and acceleration can be established through the law of conservation of momentum. Conservation of momentum is based on Newton's Second Law, $\vec{F} = m\vec{a}$. The force vector is determined based on the sum of the spatial derivatives of the stresses acting on a point in the direction of the acceleration vector. In three-dimensional Cartesian coordinates, this is expressed as:

$$\sigma_x^{11} + \sigma_y^{12} + \sigma_z^{13} = \rho u_t$$

$$\sigma_x^{12} + \sigma_y^{22} + \sigma_z^{23} = \rho v_t \qquad (3.1.8)$$

$$\sigma_x^{13} + \sigma_y^{23} + \sigma_z^{33} = \rho w_t$$

The equalities of 3.1.5, 3.1.6, and 3.1.8 can then be combined into a single system of nine partial differential equations:

$$\epsilon_t^{11} - u_x = 0$$

$$\epsilon_t^{22} - v_y = 0$$

$$\epsilon_t^{33} - w_z = 0$$

$$\epsilon_t^{12} - \frac{1}{2}(v_x + u_y) = 0$$

$$\epsilon_t^{23} - \frac{1}{2}(v_z + w_y) = 0 \qquad (3.1.9)$$

$$\epsilon_t^{13} - \frac{1}{2}(u_z + w_x) = 0$$

$$\rho u_t - \sigma_x^{11} - \sigma_y^{12} - \sigma_z^{13} = 0$$

$$\rho v_t - \sigma_x^{12} - \sigma_y^{22} - \sigma_z^{23} = 0$$

$$\rho w_t - \sigma_x^{13} - \sigma_y^{23} - \sigma_z^{33} = 0$$

At this point, application of Hooke's Law, which defines a linear relationship between stress and strain, can be used to eliminate a variable by expressing stress as a linear function of

strain, or strain as a linear function of stress. These relationships are summarized in the following systems:

$$
\begin{bmatrix} \epsilon^{11} \\ \epsilon^{22} \\ \epsilon^{33} \\ \epsilon^{12} \\ \epsilon^{23} \\ \epsilon^{13} \end{bmatrix} = \begin{bmatrix} 1/E & -\nu/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & 1/E & -\nu/E & 0 & 0 & 0 \\ -\nu/E & -\nu/E & 1/E & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2\mu \end{bmatrix} \begin{bmatrix} \sigma^{11} \\ \sigma^{22} \\ \sigma^{33} \\ \sigma^{12} \\ \sigma^{23} \\ \sigma^{13} \end{bmatrix}
\tag{3.1.10}
$$

$$
\begin{bmatrix} \sigma^{11} \\ \sigma^{22} \\ \sigma^{33} \\ \sigma^{12} \\ \sigma^{23} \\ \sigma^{13} \end{bmatrix} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\mu \end{bmatrix} \begin{bmatrix} \epsilon^{11} \\ \epsilon^{22} \\ \epsilon^{33} \\ \epsilon^{12} \\ \epsilon^{23} \\ \epsilon^{13} \end{bmatrix}
\tag{3.1.11}
$$

With the substitutions of strain as a function of stress, the system 3.1.9 can be expressed in the standard hyperbolic form of:

$$
q_t + Aq_x + Bq_y + Cq_z = 0
\tag{3.1.12}
$$

$$
q = \begin{bmatrix} \sigma^{11} \\ \sigma^{22} \\ \sigma^{33} \\ \sigma^{12} \\ \sigma^{23} \\ \sigma^{13} \\ u \\ v \\ w \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -(\lambda + 2\mu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu \\ -1/\rho & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/\rho & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1/\rho & 0 & 0 & 0 \end{bmatrix}
$$

The coefficient matrices B, and C are structured similar to A, with appropriate transposition of rows and columns.

Behavior of the solution to this system of PDEs can be understood through examination of the eigenvalues and eigenvectors of the coefficient matrices (eigenvalues of all three matrices are identical, and eigenvectors only differ by row references). To avoid confusion with the Lamé parameter, λ, the variable s is used for the eigenvalues, representing the speed of propagation for the waves. The nine eigenvalues are:

$$
s^1 = -c_p, \qquad s^2 = c_p, \qquad s^3 = -c_s, \qquad s^4 = c_s
$$

$$
s^5 = -c_s, \qquad s^6 = c_s, \qquad s^7 = s^8 = s^9 = 0
\tag{3.1.13}
$$

The speeds, c_p and c_s are calculated by:

$$c_p = \sqrt{\frac{\lambda+2\mu}{\rho}}, \quad c_s = \sqrt{\frac{\mu}{\rho}} \tag{3.1.14}$$

The corresponding eigenvectors are:

$$r^{1,2} = \begin{bmatrix} \lambda + 2\mu \\ \lambda \\ \lambda \\ 0 \\ 0 \\ 0 \\ \pm c_p \\ 0 \\ 0 \end{bmatrix}, \quad r^{3,4} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mu \\ 0 \\ 0 \\ 0 \\ \pm c_s \\ 0 \end{bmatrix}, \quad r^{5,6} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \mu \\ 0 \\ 0 \\ \pm c_s \end{bmatrix}, \quad r^7 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad r^8 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad r^9 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.1.15}$$

The eigenvectors r^1 and r^2 represent the propagation of primary (pressure) waves along the x-axis at the speed of sound, c_p; r^3 and r^4 represent secondary (shear) wave propagation within the x-y plane, and; r^5 and r^6 represent secondary (shear) wave propagation within the x-z plane.

This system (3.1.12) can be simplified to two dimensions by assuming no variation in the z-direction ($q_z = 0$):

$$q_t + A q_x + B q_y = 0 \tag{3.1.16}$$

Representation of the coefficients can be consolidated by creating a single matrix, $\check{A}$, that can represent both A and B. This formulation becomes beneficial when presence of a non-uniform grid requires numerical evolution of the solution in non-Cartesian directions (see Section 4.5: Representation of Complex Material Geometry).

$$\check{A} = n^x A + n^y B \tag{3.1.17}$$

$$q = \begin{bmatrix} \sigma^{11} \\ \sigma^{22} \\ \sigma^{12} \\ u \\ v \end{bmatrix}, \quad \check{A} = - \begin{bmatrix} 0 & 0 & 0 & n^x(\lambda + 2\mu) & n^y \lambda \\ 0 & 0 & 0 & n^x \lambda & n^y(\lambda + 2\mu) \\ 0 & 0 & 0 & n^y \lambda & n^x \lambda \\ n^x/\rho & 0 & n^y/\rho & 0 & 0 \\ 0 & n^y/\rho & n^x/\rho & 0 & 0 \end{bmatrix}; \quad \vec{n} = (n^x, n^y)$$

$$\check{s}^1 = -c_p, \quad \check{s}^2 = c_p, \quad \check{s}^3 = -c_s, \quad \check{s}^4 = c_s, \quad \check{s}^5 = 0$$

$$\check{r}^{1,2} = \begin{bmatrix} \lambda + 2\mu(n^x)^2 \\ \lambda + 2\mu(n^x)^2 \\ 2\mu n^x n^y \\ \pm n^x c_p \\ \pm n^y c_p \end{bmatrix}, \qquad \check{r}^{3,4} = \begin{bmatrix} 2\mu n^x n^y \\ 2\mu n^x n^y \\ [(n^x)^2 - (n^y)^2]\mu \\ \mp n^y c_s \\ \pm n^x c_s \end{bmatrix}, \qquad \check{r}^5 = \begin{bmatrix} (n^y)^2 \\ (n^x)^2 \\ -n^x n^y \\ 0 \\ 0 \end{bmatrix}$$

The formulation of linear elasticity in terms of strain, instead of stress, requires some additional considerations. Formulation in this manner may be necessary if calculation of energy is a desired output of the model (see Section 3.2) and there exists media characterized by a shear modulus of zero (see Section 3.3). Using a 2-dimensional formulation as an example, from the original formulation (equation 3.1.9), only the conservation of momentum equations (equation 3.1.8) change, and the system becomes:

$$\epsilon_t^{11} - u_x = 0$$

$$\epsilon_t^{22} - v_y = 0$$

$$\epsilon_t^{12} - \frac{1}{2}(v_x + u_y) = 0 \tag{3.1.18}$$

$$\rho u_t - \sigma_x^{11} - \sigma_y^{12} = 0 \;\rightarrow\; \rho u_t - (\lambda + 2\mu)\epsilon_x^{11} - \lambda\epsilon_x^{22} - 2\mu\epsilon_y^{12} = 0$$

$$\rho v_t - \sigma_x^{12} - \sigma_y^{22} = 0 \;\rightarrow\; \rho v_t - (\lambda + 2\mu)\epsilon_y^{22} - \lambda\epsilon_y^{11} - 2\mu\epsilon_x^{12} = 0$$

The system with eigenvalues and eigenvectors can be expressed as:

$$q_t + Aq_x + Bq_y = 0 \tag{3.1.19}$$

$$q = \begin{bmatrix} \epsilon^{11} \\ \epsilon^{22} \\ \epsilon^{12} \\ u \\ v \end{bmatrix}, \qquad A = -\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 \\ (\lambda + 2\mu)/\rho & \lambda/\rho & 0 & 0 & 0 \\ 0 & 0 & 2\mu/\rho & 0 & 0 \end{bmatrix}$$

$$s^1 = -c_p, \qquad s^2 = c_p, \qquad s^3 = -c_s, \qquad s^4 = c_s, \qquad s^5 = 0$$

$$r^{1,2} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \pm c_p \\ 0 \end{bmatrix}, \qquad r^{3,4} = \begin{bmatrix} 0 \\ 0 \\ 1/2 \\ 0 \\ \pm c_s \end{bmatrix}, \qquad r^5 = \begin{bmatrix} \dfrac{1}{\lambda + 2\mu} \\ -\dfrac{\lambda}{} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

As expected the system retains the same eigenvalues as the stress system, representing the same propagation speed of pressure and shear waves. The significant difference is the fact that

the zero-speed wave results in a jump of ϵ^{11} and ϵ^{22}, whereas in the stress system, this wave represented a jump in only σ^{22}.

Section 3.2: Conservation Laws

Physical systems adhere to laws of conservation, namely conservation of mass, conservation of momentum, and conservation of energy. Conservation of momentum was required for the hyperbolic formulation in the establishing the relationships in 3.1.8. Conservation of mass ensures that the total mass of the system changes only due to mass entering or leaving the system (through the boundary or at a source/sink inside of the boundary). Finite volume methods applied to linear elastic systems ensures conservation of mass by defining the material properties associated with each finite volume (grid cell) within the domain. Specifically, an auxiliary variable associated with each cell contains the density of material in the cell, and therefore based on the volume of the cell, a mass can be calculated. Because there is no flow of mass in the linear elastic system, the mass of each cell remains constant and conservation of mass is ensured.

Conservation of energy includes all forms of energy (i.e. kinetic, potential, thermal); however, in the linear elastic system, total energy in the system can be viewed as the sum of only potential energy and kinetic energy. Kinetic energy, KE, is based on mass and velocity:

$$KE = \frac{1}{2}mv^2 \tag{3.2.1}$$

Potential energy in the system is closely related to the definition of potential energy in a spring, where PE_s, is defined as:

$$PE_s = \frac{1}{2}kx^2 \tag{3.2.2}$$

In a multi-dimensional elastic system, the relationships between normal and shear stresses result in a more complex expression. The formula for internal energy, $\mathcal{U}$, as referenced by Harris is:[43]

$$\mathcal{U} = \frac{1}{2}\sum_{i,j} \sigma^{ij}\epsilon^{ij} \tag{3.2.3}$$

Using relationships in 3.1.10, this formulation can be expressed solely in terms of strain.

$$\mathcal{U} = \frac{1}{2}[\sigma^{11}\epsilon^{11} + \sigma^{22}\epsilon^{22} + \sigma^{33}\epsilon^{33} + 2\sigma^{12}\epsilon^{12} + 2\sigma^{13}\epsilon^{13} + 2\sigma^{23}\epsilon^{23}]$$

$$\mathcal{U} = \frac{1}{2}\big[\epsilon^{11}\big((\lambda + 2\mu)\epsilon^{11} + \lambda\epsilon^{22} + \lambda\epsilon^{33}\big) + \epsilon^{22}(\lambda\epsilon^{11} + (\lambda + 2\mu)\epsilon^{22} + \lambda\epsilon^{33})$$
$$+ \epsilon^{33}(\lambda\epsilon^{11} + \lambda\epsilon^{22} + (\lambda + 2\mu)\epsilon^{33}) + 2\epsilon^{12}\,2\mu\epsilon^{12} + 2\epsilon^{13}\,2\mu\epsilon^{13} + 2\epsilon^{23}\,2\mu\epsilon^{23}\big]$$

By organizing by terms of like coefficients, this equation becomes:

$$\mathcal{U} = \frac{1}{2}[(\lambda + 2\mu)((\epsilon^{11})^2 + (\epsilon^{22})^2 + (\epsilon^{33})^2) + \tag{3.2.4}$$

$$\lambda(2\epsilon^{11}\epsilon^{22} + 2\epsilon^{11}\epsilon^{33} + 2\epsilon^{22}\epsilon^{33}) + 4\mu(\epsilon^{12})^2 + 4\mu(\epsilon^{13})^2 + 4\mu(\epsilon^{23})^2]$$

This is equivalent to the energy function determined by A.E.H. Love:[44§]

$$PE = \frac{1}{2}(\lambda + 2\mu)(\epsilon^{11} + \epsilon^{22} + \epsilon^{33})^2$$
$$+ 2\mu((\epsilon^{12})^2 + (\epsilon^{13})^2 + (\epsilon^{23})^2 - \epsilon^{11}\epsilon^{22} - \epsilon^{11}\epsilon^{33} - \epsilon^{22}\epsilon^{33}) \tag{3.2.5}$$

In order to demonstrate conservation of energy in two dimensions (for ease of demonstrating the calculation), this expression for internal energy (Potential Energy) can be simplified by assuming no variation in the z-direction: $\epsilon^{33} = \epsilon^{13} = \epsilon^{23} = 0$. The two-dimensional expression for potential energy becomes:

$$PE = \frac{1}{2}(\lambda + 2\mu)(\epsilon^{11} + \epsilon^{22})^2 + 2\mu((\epsilon^{12})^2 - \epsilon^{11}\epsilon^{22}) \tag{3.2.6}$$

To calculate the total energy in the two dimensional system, the kinetic energy, $KE = \frac{1}{2}\rho(u^2 + v^2)$, must be added to the potential energy; $E = KE + PE$ (equation 3.2.6 has been expanded in the following, in order to match like terms in later steps):

$$E = \left[\frac{1}{2}\rho u^2 + \frac{1}{2}\rho v^2\right] + \tag{3.2.7}$$

$$\left[\frac{1}{2}(\lambda + 2\mu)(\epsilon^{11})^2 + \frac{1}{2}(\lambda + 2\mu)(\epsilon^{22})^2 + 2\mu(\epsilon^{12})^2 + \lambda\epsilon^{11}\epsilon^{22}\right]$$

The total energy in the system can be calculated as the integral of this function over the entire domain; or in the discrete (finite volume) case, the total energy is the sum of the energy in each grid cell. Conservation of energy requires that the derivative of the total energy with respect to time is based only on energy flux through the boundary of the domain. First, by taking the derivative with respect to time, the rate of change of energy can be expressed as:

$$E_t = \rho u u_t + \rho v v_t + (\lambda + 2\mu)\epsilon^{11}\epsilon_t^{11} + \tag{3.2.8}$$

---

§ Different conventions are used by various authors in expressing strain. In Love's formulation of elastic equations, shear strains, expressed as e_{xy}, e_{xz}, and e_{yz} are half the magnitude used by LeVeque, such that $e_{xy} = 2\epsilon^{12}$

$$(\lambda + 2\mu)\epsilon^{22}\,\epsilon_t^{22} + 4\mu\epsilon^{12}\,\epsilon_t^{12} + \lambda\epsilon^{11}\,\epsilon_t^{22} + \lambda\epsilon^{22}\,\epsilon_t^{11}$$

Using the relationship between strain and acceleration (in two dimensions), and the stress-strain relationship, substitution of values for ρu_t, ρv_t, ϵ_t^{11} and ϵ_t^{22} into the equation yields:

$$E_t = u\left((\lambda + 2\mu)\epsilon_x^{11} + \lambda\epsilon_x^{22} + 2\mu\epsilon_y^{12}\right) + v\left((\lambda + 2\mu)\epsilon_y^{22} + \lambda\epsilon_y^{11} + 2\mu\epsilon_x^{12}\right) + \quad \text{(3.2.9)}$$

$$(\lambda + 2\mu)\epsilon^{11}u_x + (\lambda + 2\mu)\epsilon^{22}v_y + 4\mu\epsilon^{12}\left(\frac{1}{2}(v_x + u_y)\right) + \lambda\epsilon^{11}v_y + \lambda\epsilon^{22}u_x$$

When simplified, the resulting equation is demonstrated to be a sum of spatial derivatives:

$$E_t = (\lambda + 2\mu)(u\epsilon_x^{11} + \epsilon^{11}u_x) + (\lambda + 2\mu)\left(v\epsilon_y^{22} + \epsilon^{22}v_y\right) + \quad \text{(3.2.10)}$$

$$\lambda(u\epsilon_x^{22} + \epsilon^{22}u_x) + \lambda\left(v\epsilon_y^{11} + \epsilon^{11}v_y\right) + 2\mu\left(u\epsilon_y^{12} + \epsilon^{12}u_y\right) + 2\mu(v\epsilon_x^{12} + \epsilon^{12}v_x)$$

$$\frac{d}{dt}E = \frac{d}{dx}[(\lambda + 2\mu)(u\epsilon^{11}) + \lambda(u\epsilon^{22}) + 2\mu(v\epsilon^{12})]$$

$$+ \frac{d}{dy}[(\lambda + 2\mu)(v\epsilon^{22}) + \lambda(v\epsilon^{11}) + 2\mu(u\epsilon^{12})]$$

By defining a vector field, $\boldsymbol{F}$, such that

$$\boldsymbol{F} = (F_x, F_y); \quad \text{(3.2.11)}$$

$$F_x = (\lambda + 2\mu)(u\epsilon^{11}) + \lambda(u\epsilon^{22}) + 2\mu(v\epsilon^{12})$$

$$F_y = (\lambda + 2\mu)(v\epsilon^{22}) + \lambda(v\epsilon^{11}) + 2\mu(u\epsilon^{12}).$$

It then follows that using the gradient operator ($\nabla = \left[\frac{d}{dx}, \frac{d}{dy}\right]$), the energy function becomes:

$$\frac{d}{dt}E = \nabla \cdot \boldsymbol{F}, \quad \text{(3.2.12)}$$

and by divergence theorem

$$\int_S \frac{d}{dt}E\,dA = \int_{\partial S} \boldsymbol{F} \cdot \vec{n}\,ds$$

$$\frac{d}{dt}\int_S E\,dA = \int_{\partial S} \boldsymbol{F} \cdot \vec{n}\,ds \quad \text{(3.2.13)}$$

Therefore, the time rate of change of the total energy in the domain, S, can only change based on flux across the boundary of the domain, ∂S. Similar results can be obtained for both the one-

and three-dimensional cases as well. Conservation of energy for the PDEs does not ensure conservation of energy for the numerical method.

Section 3.3: Modeling Fluids in an Elastic System

The physical phenomenon in consideration consists of a pressure wave, initially propagating through the atmosphere (air) into the human body (this discussion extends to modeling other fluids (such as CSF) characterized by shear modulus of zero). The numerical model must therefore model propagation in air as well as the body, and more importantly, the interface between the two media. Propagation of pressure waves in air is modeled as an acoustic system, which is a simplified version of the linear elastic system. The p-waves in the elastic formulation are acoustic pressure waves, and shear waves are not present in the acoustic system. Behavior of the linear elastic model is based on the Lamé parameters, λ and μ, which are not typically associated with air; however, through examination of the model of acoustic wave propagation, appropriate parameters can be determined. Acoustic waves in two dimensions are modeled by the hyperbolic system:

$$\begin{bmatrix} p \\ u \\ v \end{bmatrix}_t + \begin{bmatrix} 0 & K & 0 \\ 1/\rho & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix}_x + \begin{bmatrix} 0 & 0 & K \\ 0 & 0 & 0 \\ 1/\rho & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix}_y = 0 \qquad (3.3.1)$$

In this system, K is the bulk modulus of compressibility, p is pressure, and u and v are velocities in the direction of the x- and y-directions, respectively. Based on the eigenvalues of the coefficient matrices, the speed of sound is defined as:

$$c_s = \sqrt{\frac{K}{\rho}} \qquad (3.3.2)$$

Using 20 degrees Celsius as a point of reference, the speed of sound in air is 343.3 m/s, and the density is 1.205kg/m^3. Used as a baseline for this study, this would suggest of bulk modulus of:

$$K = \rho c_s^2 \approx 1.42 \times 10^5 \, \frac{\text{kg}}{\text{m} \cdot \text{s}^2} \qquad (3.3.3)$$

Applying this to the linear elastic model, this becomes the equivalent of the Lamé parameter, λ. Because shear waves do not exist in air, the shear modulus, μ, must be equal to zero:

$$c_p = \sqrt{\frac{\lambda + 2\mu}{\rho}} = \sqrt{\frac{K}{\rho}} \qquad (3.3.4)$$

In air, the stress-strain relationship (equation 3.1.11) becomes

$$
\begin{bmatrix} \sigma^{11} \\ \sigma^{22} \\ \sigma^{33} \\ \sigma^{12} \\ \sigma^{23} \\ \sigma^{13} \end{bmatrix} = \begin{bmatrix} \lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \epsilon^{11} \\ \epsilon^{22} \\ \epsilon^{33} \\ \epsilon^{12} \\ \epsilon^{23} \\ \epsilon^{13} \end{bmatrix} \tag{3.3.5}
$$

Due to the singular nature of the matrix above, the inverse relationship cannot be defined. Directional strains cannot be determined from the directional stresses which are always equal, and shear stresses are always equal to zero.

$$
\sigma^{11} = \sigma^{22} = \sigma^{33} = \lambda(\epsilon^{11} + \epsilon^{22} + \epsilon^{33}) \tag{3.3.6}
$$

$$
\sigma^{12} = \sigma^{23} = \sigma^{13} = 0
$$

As a consequence, determination of the strains and cannot be accomplished with this formulation.

Section 3.4: Assumptions

The linear-elastic model of this phenomenon requires that the model adheres to certain assumptions which will inherently limit the accuracy of the numerical model in accurately representing the desired phenomenon. Discussion of these assumptions follows:

Assumption 1: Wave propagation through air at a constant speed of sound

By treating the air surrounding the head as an elastic material, the flow (blast wind) associated with movement of air particles displaced by the conversion of explosive solid into gas is not modeled. A consequence of this is the reduction of propagation speed of the shock wave from the actual supersonic speeds to the speed of sound; however, the rapid decay in speed of the shock front reduces the significance of this deviation. At a distance of eight meters from an explosive blast of 10kg TNT, the shock speed is approximately mach 1.2. The acceleration of the body, caused by the wind, is associated more directly with secondary and tertiary blast injury; therefore, this assumption ensures that primary injury mechanisms are isolated in the model and limits the model's use in more general applications of blast injury research. The pressure profile will reflect the appropriate pressure-time relationship at a point of impact with the skull, and the deviation in velocity of the air particles will be minimal because the speed of the shock is significantly greater in magnitude than the wind speed.

Assumption 2: Linear elastic biological materials

By treating the biological materials as linear elastics, the resulting system, will not account for loss of energy due to visco-elastic damping. The shaded region in the stress-strain diagram for a visco-elastic material (Figure 4) represents hysteresis, the amount of energy lost

as heat during a load/unload cycle. Additionally, energy will be absorbed through plastic deformation (representing physical damage to the cellular structure), or conversion to thermal energy. If the elastic limits of the cellular structures within the head are known, the linear elastic model can provide utility in identification of when and where an injury will occur, based on the when the limitations are exceeded; however numerical results for time periods after elastic limits are exceeded will be inherently erred. Furthermore, the fibrous nature of neurons within the brain results in material properties that are anisotropic and non-homogenous.

Assumption 3: Estimation of waveform

By reducing the computational domain to a small area, the wave must be initiated at the boundary. This is accomplished by defining a plane-wave with propagation consistent with a one-dimensional problem (no decay based on increasing radius, or curved shape of the front). At sufficient distance from the blast, the curvature of the front of the shock wave becomes insignificant, as illustrated in Figure 5. Furthermore, the change in pressure profile within the physical span of the domain will also be minimal. These factors create reasonable basis for use of a plane-wave generated at a single boundary as input to the system. Implementation of the plane wave (from the left boundary of the domain) requires use of boundary conditions on the top and bottom that ensure the waveform continues to propagate as a plane wave. This is accomplished with zero-order extrapolation boundary conditions . These boundaries prevent decay of the wave through expansion lateral to the direction of movement. Consequently, the wave will not decay as it crosses the domain. The actual decay across a span the width of a human skull is relatively small, motivating use of the average value of the peak pressure on either side of the skull (approximately 20cm) to define the modeled wave (see Figure 6).
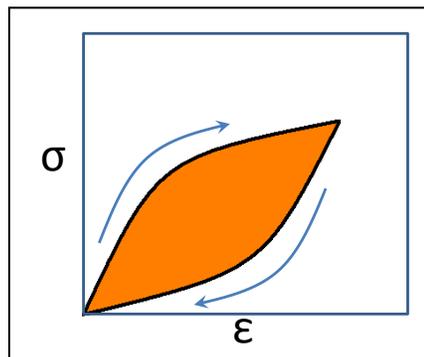
Figure 4: Stress-strain relationship in visco-elastic media. Shaded area in the graph represents hysteresis, the energy lost during each load/unload cycle.
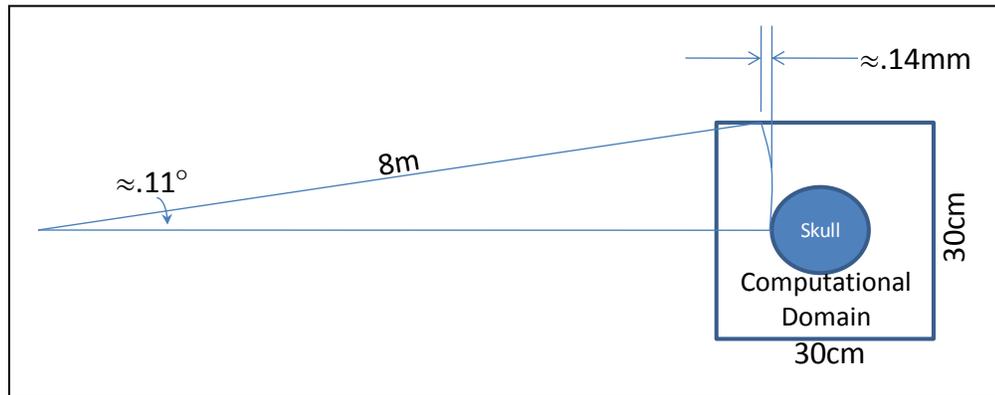
Figure 5: Plane wave justification. Illustration depicting curvature of the spherical expansion of a blast wave front at 8 meters from source.

It is important to ensure that the pressure waveforms used in the model appropriately reflect the actual conditions resulting in bTBI. Actual pressure histories of blast waves resulting in bTBI cannot be determined with a high degree of precision; therefore, the Friedlander equations provide a logical starting point for defining a waveform for numerical modeling. Continuation of the waveform into the negative phase results in an indefinite duration of the negative phase. Through experimentation, it is known that the negative phase of an explosive blast wave in air typically lasts approximately three times the duration of the positive phase.[45] Taking this into consideration, the following waveform is defined based on equation 1.8.

$$p(t) = p_o \left(1 - \frac{t}{t_d}\right) e^{\frac{-\alpha t}{t_d}} \times d \qquad (3.4.1)$$

This is the same as the Friedlander equation, with the addition of a multiplier, d. This variable represents a piecewise linear damping term that forces the continuation of the waveform expression, beyond the time boundaries of the positive phase, to represent a qualitatively appropriate form, by using the following:

$$d = \begin{cases} 0 & t < t_a \ or \ t \geq 4t_d \\ 1 & t_a \leq t < t_d \\ \frac{-t + 4t_d}{3t_d} & t_d \leq t < 4t_d \end{cases} \qquad (3.4.2)$$

Application of this waveform is illustrated in Figure 6, using the average of maximum overpressure that would be experienced on either side of a skull (~20cm). As illustrated, this difference is minimal.
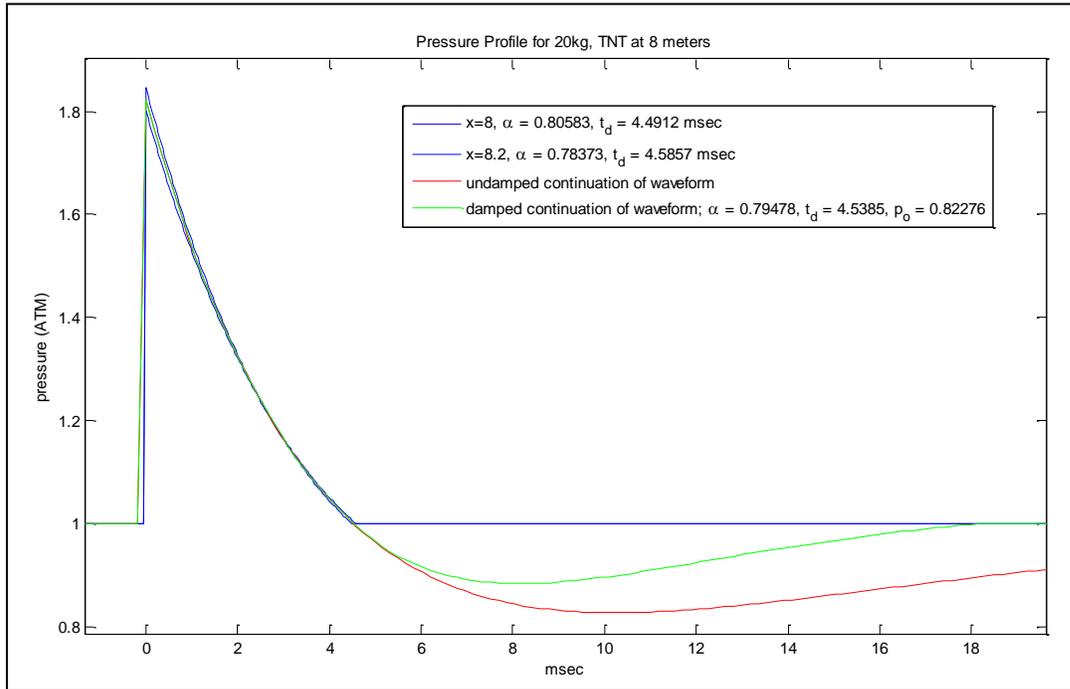
Figure 6: Modeled waveform. Plot illustrating Friedlander equation (blue) representing the decay of the blast wave as it crosses the span of the head (approximately 20cm), continuation of the Friedlander equation beyond t_d (red), and waveform implemented in model (green).

Chapter 4: CLAWPACK (Conservation LAWs PACKage)

CLAWPACK is a software application designed specifically to implement Finite Volume Methods used to calculate numerical solutions to systems of hyperbolic partial differential equations (PDEs), typically in the linear form

$$q_t + Aq_x = 0, \tag{4.1}$$

in which Aq is a linear flux function, $f(q)$. For a scalar, PDE, this is hyperbolic provided that A is real. If this represents a system of m PDEs, it is hyperbolic if all the eigenvalues of the $m \times m$ matrix A are real, and there exists a complete set of eigenvectors.

For physical systems based on conservation laws, the value of conserved quantities within a bounded region of a domain can only change in time based on flux occurring at through boundaries of the region. In one-dimension, this is expressed as:

$$\frac{d}{dt} \int_{x_1}^{x_2} q(x, t) dx = f\big(q(x_1, t)\big) - f\big(q(x_2, t)\big) \tag{4.2}$$

This formulation is the basis for finite volume numerical methods in which x_1 and x_2 represent boundaries of a grid cell, defining the region of a finite volume.

Section 4.1: Godunov's Method

Godunov's Method is an algorithm for evolving hyperbolic PDEs forward in time, and provides a basic means for expanding the method to non-linear conservation laws.[46] For this method, Q is the numerical approximation of the solution, q, and F is the numerical approximation of the flux function, f. The method begins by discretizing the solution to establish piecewise constant solution values representing the average value in each grid cell:

$$Q_i^n \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} q(x, t_n) dx \tag{4.1.1}$$

In this formulation, $x_{i-1/2}$ and $x_{i+1/2}$ are the left and right boundaries of cell i, respectively. The basic form for stepping forward in time is:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\big(F_{i+1/2}^n - F_{i-1/2}^n\big) \tag{4.1.2}$$

$F_{i-1/2}^n$ is the numerical flux at boundary $x_{i-1/2}$. This flux is calculated using a Riemann Solver, discussed in Section 4.2. Simply stated, this shows that the value of conserved quantities after a step forward in time is the previous value minus flux leaving the cell, and plus the flux entering the cell.

Godunov's method is implemented with a Reconstruct-Evolve-Average (REA) algorithm. First, a piecewise polynomial function is defined for each grid cell to approximate the solution value in that cell (the simplest being a constant function), then based on the discontinuities at the cell boundaries, waves are calculated which propagate into/out of the cells and define fluxes at the cell boundaries representing regions of updated values for the conserved variables. Finally, these regions are averaged across the entire cell to determine a new cell average. Other methods, such as the Lax-Wendroff Method can achieve second-order accuracy by basing the evolution of the solution on additional terms in a Taylor series expansion; however, because the Taylor expansion is not valid at points of discontinuity in the solution, first order accuracy is the best that can be expected. Godunov's method is therefore implemented in this study. Methods for increasing accuracy are discussed in Section 4.4: Limiters.

Godunov's method provides a means to numerically solve a system of hyperbolic PDEs throughout a discretized domain. The structure of the system allows the problem to be split into separate problems for each dimension, a process known as *dimensional splitting*.

$$q_t + Aq_x = 0$$

$$q_t + Bq_y = 0 \qquad\qquad (4.1.3)$$

$$q_t + Cq_z = 0$$

Because propagation of waves does not always occur in directions normal to the boundaries, dimensional splitting introduces a "splitting error." For special cases in which $AB = BA$ (in two dimensions), the system's behavior in the direction of each axis is independent of the other direction, and the splitting error is not present. However, in the linear elastic system, $AB \neq BA$, therefore, this is an expected source of error.

At each time step the evolution of the solution is based on the numerical solution of the PDEs corresponding to each spatial dimension. Let Q^n be the numerical approximation to the solution, at time t_n. For each time step of length Δt, the algorithm begins by evolving the problem $Q_t^n + AQ_x^n = 0$ a step forward in time by solving a Riemann problem at the two boundaries representing the cells boundary in the direction of the x-axis (see Section 4.2). The resulting solutions define an intermediate value for each grid cell, Q^*. Based on this solution, the system $Q_t^* + BQ_y^* = 0$ can be evolved a full time step, Δt, with the resulting solution of Q^{**}. Finally, the system $Q_t^{**} + CQ_z^{**} = 0$ is evolved a full time step, with the final result of Q^{n+1} to complete the time step.
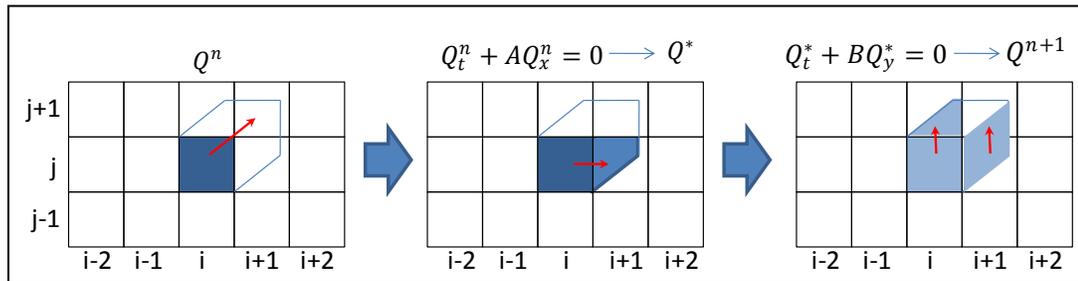
Figure 7: Dimensional splitting algorithm for 2-D wave propagation. Example problem in which initial condition (left) is evolved in x-direction to intermediate solution (middle) then evolved in y-direction for completion of time step (right)

For example, consider a two-dimensional problem, in which the solution in cell (i, j) consists of a wave propagating in the positive x and positive y directions (with $q = 0$ in all other cells) as depicted in Figure 7. During the first step, the solution to $Q_t^n + AQ_x^n$ will update cell $(i + 1, j)$, accounting for the rightward propagation of the wave. Next, the system, $Q_t^* + BQ_y^*$ will update cell $(i, j + 1)$, based on the upward propagation of the wave, and cell $(i + 1, j + 1)$ is updated based only on the value of Q^* in cell $(i + 1, j)$, however the solution, Q^{n+1} in cell $(i + 1, j + 1)$ will have no contribution from the upward component of the wave in originally in cell (i, j) and propagating through $(i, j + 1)$. This can be addressed through used of a second Riemann solver, described as a transverse solver, which decomposes the flux after the first step into eigenvectors associated with B, and calculates the contribution to cell $(i + 1, j + 1)$, before beginning the second step. Increased accuracy based on implementation of this proves negligible compared to the added computation cost of implementation, especially when considering a three dimensional model in which two transverse solvers are required for each wave at each interface. In application, the transverse solver has even contributed to sources of instability when modeling interfaces between significantly differing material properties with complex geometries (i.e. air/bone interface).

Section 4.2: Riemann Solvers

The finite volume method relies primarily on the use of Riemann Solvers which are used to solve a *Riemann Problem* at each boundary on each grid cells. A *Riemann Problem* is characterized by an initial condition in which a left state, q_l, and a right state, q_r, are distinguished with a single jump discontinuity between the two states.[47]

$$q = \begin{cases} q_l & x < x_o \\ q_r & x > x_o \end{cases} \tag{4.2.1}$$

The solution to the Riemann problem consists of a set of waves propagating from the point of discontinuity. Each wave represents a jump in solution value corresponding to a scalar multiple of the eigenvector associated with the wave. In the case of the linear elastic system in
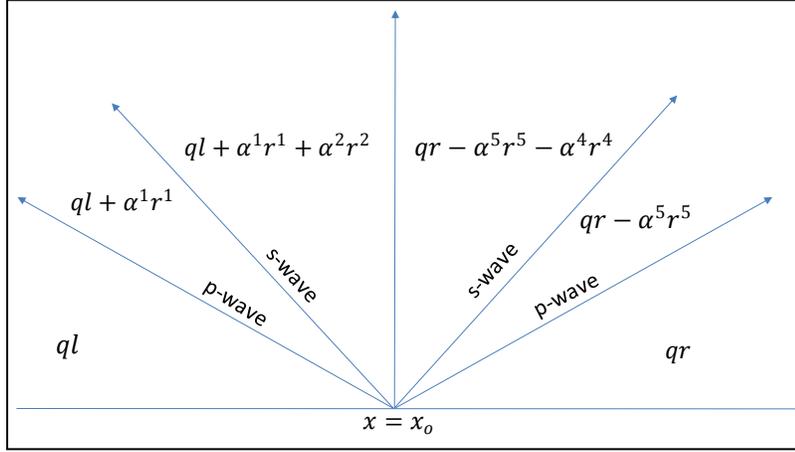
Figure 8: Depiction of solutions to the Riemann problem based on wave propagation. The difference between any two sectors is a scalar multiple of the wave between them.

two-dimensions, the solution consists of two pressure waves (also referred to as primary or p-waves), and two shear waves (secondary or s-waves). This is illustrated in Figure 8.

During each time step, the main CLAWPACK routine calls a user-programmed Riemann solver to solve Riemann Problems at every interface within the domain. Updating the solution in each grid cell is based on the solution of two Riemann problems for each spatial dimension of the domain.

Solution to the Riemann problem begins with decomposition of the jump in solution at interface into eigenvectors of the coefficient matrix, A. In the case of variable coefficient problems, the eigenvectors must be based on the coefficient matrix representing the cell that the wave is propagating into.

$$q_r - q_l = \alpha^1 r^1 + \cdots + \alpha^m r^m$$

$$\boldsymbol{\alpha} = R^{-1}(q_r - q_l) \tag{4.2.2}$$

Waves, $\mathcal{W}$, are then defined as the product of the wave strength and corresponding eigenvector:

$$\mathcal{W}^p = \alpha^p r^p \tag{4.2.3}$$

The net flux can then be defined as the sum of the original flux function ($f(q) = Aq$) and the rate of propagation of each wave multiplied by the corresponding wave:

$$F^n_{i\pm 1/2} = AQ_i + \sum_{p=1}^{m} (\lambda^p)^{\mp} \mathcal{W}^p_{i\pm 1/2} \quad ; \ (\lambda^p)^+ = \max(0, \lambda^p), \ (\lambda^p)^- = \min(0, \lambda^p) \tag{4.2.4}$$

By including the flux at both boundaries and averaging across the width of the cell, a formula for updating the solution in each cell is:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left[\sum_{p=1}^{m}(\lambda^p)^+\mathcal{W}_{i-1/2}^p + \sum_{p=1}^{m}(\lambda^p)^-\mathcal{W}_{i-1/2}^p\right] \tag{4.2.5}$$

The following demonstrates the Riemann Solver for an interface between two adjacent cells in the x-direction. Because material properties may differ between cells, it is important to include a reference to the appropriate side of the interface. The speed of wave propagation is based on the material properties of the cell it is propagating into (i.e. left propagating waves, identified by negative eigenvalues, have speeds, c_{pl} and c_{sl}, based on the properties of the left cell, μ_l, λ_l, and ρ_l). Similarly, the eigenvectors must also be based on the properties of the cell based on their associated eigenvalues. The eigenvalues are expressed in the matrix Λ, and the corresponding eigenvectors in matrix R.

$$\Lambda = \begin{bmatrix} -c_{pl} & 0 & 0 & 0 & 0 \\ 0 & -c_{sl} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{sr} & 0 \\ 0 & 0 & 0 & 0 & c_{pr} \end{bmatrix}, \quad c_p = \sqrt{\frac{\lambda+2\mu}{\rho}}, \quad c_s = \sqrt{\frac{\mu}{\rho}} \tag{4.2.6}$$

$$R = \begin{bmatrix} \lambda_l + 2\mu_l & 0 & 0 & 0 & \lambda_r + 2\mu_r \\ \lambda_l & 0 & 1 & 0 & \lambda_r \\ 0 & \mu_l & 0 & \mu_r & 0 \\ c_{pl} & 0 & 0 & 0 & -c_{pr} \\ 0 & c_{sl} & 0 & -c_{sr} & 0 \end{bmatrix}$$

The inverse of the eigenvector matrix is:

$$R^{-1} = \begin{bmatrix} \dfrac{c_{pr}}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} & 0 & 0 & \dfrac{(\lambda_r + 2\mu_r)}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} & 0 \\ 0 & 0 & \dfrac{c_{sr}}{c_{sr}\mu_l + c_{sl}\mu_r} & 0 & \dfrac{\mu_r}{c_{sr}\mu_l + c_{sl}\mu_r} \\ -\dfrac{c_{pr}\lambda_l + c_{pl}\lambda_r}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} & 1 & 0 & -\dfrac{2(-\lambda_r\mu_l + \lambda_l\mu_r)}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} & 0 \\ 0 & 0 & \dfrac{c_{sl}}{c_{sr}\mu_l + c_{sl}\mu_r} & 0 & \dfrac{\mu_l}{c_{sr}\mu_l + c_{sl}\mu_r} \\ \dfrac{c_{pl}}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} & 0 & 0 & -\dfrac{(\lambda_l + 2\mu_l)}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} & 0 \end{bmatrix}$$

$$\tag{4.2.7}$$

The difference between solution values in the adjacent cells is generally expressed as:

$$(q_r - q_l) = \begin{bmatrix} \sigma_r^{11} \\ \sigma_r^{22} \\ \sigma_r^{12} \\ u_r \\ v_r \end{bmatrix} - \begin{bmatrix} \sigma_l^{11} \\ \sigma_l^{22} \\ \sigma_l^{12} \\ u_l \\ v_l \end{bmatrix} = \begin{bmatrix} \Delta\sigma^{11} \\ \Delta\sigma^{22} \\ \Delta\sigma^{12} \\ \Delta u \\ \Delta v \end{bmatrix} \tag{4.2.8}$$

The wave strengths can then be calculated by $\boldsymbol{\alpha} = R^{-1}(q_r - q_l)$:

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha^1 \\ \alpha^2 \\ \alpha^3 \\ \alpha^4 \\ \alpha^5 \end{bmatrix} = \begin{bmatrix} \dfrac{\Delta\sigma^{11}c_{pr}}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} + \dfrac{\Delta\sigma^{12}(\lambda_r + 2\mu_r)}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} \\[3mm] \dfrac{\Delta\sigma^{12}c_{sr}}{c_{sr}\mu_l + c_{sl}\mu_r} + \dfrac{\Delta v \mu_r}{c_{sr}\mu_l + c_{sl}\mu_r} \\[3mm] -\dfrac{\Delta\sigma^{11}(c_{pr}\lambda_l + c_{pl}\lambda_r)}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} + \Delta\sigma^{22} - \dfrac{\Delta u 2(-\lambda_r\mu_l + \lambda_l\mu_r)}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} \\[3mm] \dfrac{\Delta\sigma^{12}c_{sl}}{c_{sr}\mu_l + c_{sl}\mu_r} + \dfrac{\Delta v \mu_l}{c_{sr}\mu_l + c_{sl}\mu_r} \\[3mm] \dfrac{\Delta\sigma^{11}c_{pl}}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} + \dfrac{\Delta\sigma^{12}(\lambda_l + 2\mu_l)}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} \end{bmatrix}$$

$$\tag{4.2.9}$$

The waves are then defined by 4.2.3. Because the third wave has zero speed it does not contribute to updating either cell value, and can be omitted, leaving the four waves required for updating the solution:

Left-propagating Pressure Wave:

$$\mathcal{W}^1 = \left(\frac{\Delta\sigma^{11}c_{pr}}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} + \frac{\Delta\sigma^{12}(\lambda_r + 2\mu_r)}{c_{pr}(\lambda_l + 2\mu_l) + c_{pl}(\lambda_r + 2\mu_r)} \right) \begin{bmatrix} \lambda_l + 2\mu_l \\ \lambda_l \\ 0 \\ c_{pl} \\ 0 \end{bmatrix} \tag{4.2.10}$$

Left-propagating Shear Wave

$$\mathcal{W}^2 = \left(\frac{\Delta\sigma^{12}c_{sr}}{c_{sr}\mu_l + c_{sl}\mu_r} + \frac{\Delta v \mu_r}{c_{sr}\mu_l + c_{sl}\mu_r} \right) \begin{bmatrix} 0 \\ 0 \\ \mu_l \\ 0 \\ c_{sl} \end{bmatrix} \tag{4.2.11}$$

Right-propagating Pressure Wave

$$\mathcal{W}^5 = \left(\frac{\Delta\sigma^{11}c_{pl}}{c_{pr}\,(\lambda_l+2\mu_l)+c_{pl}\,(\lambda_r+2\mu_r)} + \frac{\Delta\sigma^{12}(\lambda_l+2\mu_l)}{c_{pr}\,(\lambda_l+2\mu_l)+c_{pl}\,(\lambda_r+2\mu_r)} \right) \begin{bmatrix} \lambda_r+2\mu_r \\ \lambda_r \\ 0 \\ -c_{pr} \\ 0 \end{bmatrix} \qquad (4.2.12)$$

Right-propagating Shear Wave

$$\mathcal{W}^4 = \left(\frac{\Delta\sigma^{12}c_{sl}}{c_{sr}\mu_l+c_{sl}\mu_r} + \frac{\Delta v\mu_l}{c_{sr}\mu_l+c_{sl}\mu_r} \right) \begin{bmatrix} 0 \\ 0 \\ \mu_r \\ 0 \\ -c_{sr} \end{bmatrix} \qquad (4.2.13)$$

Finally, the cell value can be updated, provided Riemann solutions have been determined on both sides of the cell (based on updating in one dimension at a time).

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left[\sum_{p=1}^m (\lambda^p)^+ \mathcal{W}_{i-1/2}^p + \sum_{p=1}^m (\lambda^p)^- \mathcal{W}_{i+1/2}^p \right] \qquad (4.2.14)$$

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left[c_s \mathcal{W}_{i-1/2}^4 + c_p \mathcal{W}_{i-1/2}^5 - c_p \mathcal{W}_{i+1/2}^1 - c_s \mathcal{W}_{i+1/2}^2 \right]$$

The Riemann Solver for the strain system (3.1.19) is more complicated. In the eigenvector matrix, R, the values for μ and λ for the center eigenvector are not specified as being in the right state or the left state because they represent the boundary of the two sides.

$$\Lambda = \begin{bmatrix} -c_{pl} & 0 & 0 & 0 & 0 \\ 0 & -c_{sl} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{sr} & 0 \\ 0 & 0 & 0 & 0 & c_{pr} \end{bmatrix}, \qquad c_p = \sqrt{\frac{\lambda+2\mu}{\rho}}, \qquad c_s = \sqrt{\frac{\mu}{\rho}} \qquad (4.2.15)$$

$$R = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & -(\lambda+2\mu)/\lambda & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 \\ c_{pl} & 0 & 0 & 0 & -c_{pr} \\ 0 & c_{sl} & 0 & -c_{sr} & 0 \end{bmatrix}$$

The inverse of the eigenvector matrix can be determined, without specifying a reference to the unknown Lamé parameters.

$$R^{-1} = \begin{bmatrix} \dfrac{c_{pr}}{c_{pr}+c_{pl}} & \dfrac{\lambda c_{pr}}{(\lambda+2\mu)(c_{pr}+c_{pl})} & 0 & \dfrac{1}{c_{pr}+c_{pl}} & 0 \\[2ex] 0 & 0 & \dfrac{2c_{sr}}{c_{sr}+c_{sl}} & 0 & \dfrac{1}{c_{sr}+c_{sl}} \\[2ex] 0 & -\dfrac{\lambda}{\lambda+2\mu} & 0 & 0 & 0 \\[2ex] 0 & 0 & \dfrac{2c_{sl}}{c_{sr}+c_{sl}} & 0 & \dfrac{-1}{c_{sr}+c_{sl}} \\[2ex] \dfrac{c_{pl}}{c_{pr}+c_{pl}} & \dfrac{\lambda c_{pl}}{(\lambda+2\mu)(c_{pr}+c_{pl})} & 0 & \dfrac{-1}{c_{pr}+c_{pl}} & 0 \end{bmatrix} \tag{4.2.16}$$

The difference between solution values in the adjacent cells is:

$$(q_r - q_l) = \begin{bmatrix} \epsilon_r^{11} \\ \epsilon_r^{22} \\ \epsilon_r^{12} \\ u_r \\ v_r \end{bmatrix} - \begin{bmatrix} \epsilon_l^{11} \\ \epsilon_l^{22} \\ \epsilon_l^{12} \\ u_l \\ v_l \end{bmatrix} = \begin{bmatrix} \Delta\epsilon^{11} \\ \Delta\epsilon^{22} \\ \Delta\epsilon^{12} \\ \Delta u \\ \Delta v \end{bmatrix} \tag{4.2.17}$$

Without specified values for λ and μ for the center wave, only the strengths for the shear waves can be calculated by $\boldsymbol{\alpha} = R^{-1}(q_r - q_l)$. Similar to the stress model, it is not required to calculate α^3, corresponding to the zero-speed wave.

$$\alpha^2 = \frac{2c_{sr}\Delta\epsilon^{12}+\Delta v}{c_{sr}+c_{sl}} \tag{4.2.18}$$

$$\alpha^4 = \frac{2c_{sl}\Delta\epsilon^{12}-\Delta v}{c_{sr}+c_{sl}} \tag{4.2.19}$$

In order to calculate the strengths of the pressure waves, some knowledge of the stress system must be known. From the stress formulation (equation 3.1.17) the value of σ^{11} must be constant across the center wave, $\sigma^{11-} = \sigma^{11+}$, where σ^{11-} represents the value of σ^{11} on the left side of the zero-speed wave, and σ^{11+} is the value of σ^{11} on the right side (illustrated in Figure 9) In terms of strain, these are defined as:

$$\sigma^{11-} = (\lambda_l + 2\mu_l)\epsilon^{11-} + \lambda_l\epsilon^{22-} \tag{4.2.20}$$

$$\sigma^{11+} = (\lambda_r + 2\mu_r)\epsilon^{11+} + \lambda_l\epsilon^{22+} \tag{4.2.21}$$
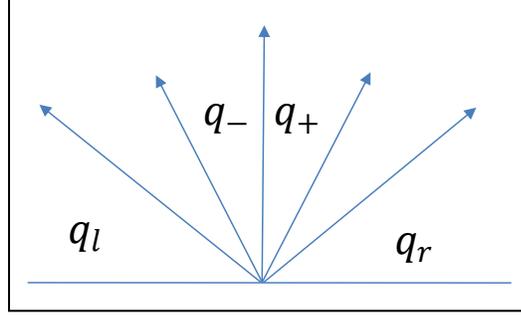
Figure 9: Solution regions associated with wave propagation in the linear elastic system.

From the eigenvectors in 4.2.15, the values for ϵ^{11+} and ϵ^{11-} can be expressed in terms of the initial conditions and wave strengths, α^1 and α^5:

$$\epsilon^{11+} = \epsilon_r^{11} - \alpha^5 \tag{4.2.22}$$

$$\epsilon^{11-} = \epsilon_l^{11} + \alpha^1 \tag{4.2.23}$$

From the third component, u, of the eigenvectors in 4.2.15, the following relationship can be established:

$$u_r - u_l = \alpha^1 c_{pl} - \alpha^5 c_{pr}$$

$$\alpha^1 = \frac{u_r - u_l + \alpha^5 c_{pr}}{c_{pl}}, \ \alpha^5 = \frac{u_r - u_l - \alpha^1 c_{pl}}{-c_{pr}} \tag{4.2.24}$$

Appropriate substitution yields equations for α^1 and α^5:

$$\alpha^1 = \frac{(\lambda_r + 2\mu_r)\left(\epsilon_r^{11} + \frac{\Delta u}{c_{pr}}\right) - (\lambda_l + 2\mu_l)\epsilon_l^{11} + \lambda_r \epsilon_r^{22} - \lambda_l \epsilon_l^{22}}{\dfrac{(\lambda_l + 2\mu_l)c_{pr} + (\lambda_r + 2\mu_r)c_{pl}}{c_{pr}}} \tag{4.2.25}$$

$$\alpha^5 = \frac{-(\lambda_l + 2\mu_l)\left(\epsilon_l^{11} + \frac{\Delta u}{c_{pl}}\right) + (\lambda_r + 2\mu_r)\epsilon_r^{11} + \lambda_r \epsilon_r^{22} - \lambda_l \epsilon_l^{22}}{\dfrac{(\lambda_l + 2\mu_l)c_{pr} + (\lambda_r + 2\mu_r)c_{pl}}{c_{pl}}} \tag{4.2.26}$$

At this point, analytic expressions for all required wave strengths have been established, and updated cell values can be determined in the same manner as the system in terms of stress.

Section 4.3: CFL Conditon

For the numerical method described above, a necessary condition for stability and convergence is the adherence to the CFL condition for the Courant number, ν:[48]

$$\nu = \frac{\Delta t}{\Delta x} \max_p |\lambda^p| \leq 1 \qquad (4.3.1)$$

This ensures that the wave solution to the Riemann Problem can contribute only to the solution of the cell adjacent to the interface. Courant numbers close to one are advantageous because they allow longer time steps; however, because the CFL condition is not a sufficient condition, stability can sometimes be enhanced by reducing the CFL below one. Besides requiring more time steps, reduction in the CFL number results in increased 'smearing' of solution values, especially at points where discontinuities exist. Unfortunately, when solving the Riemann problems throughout a larger domain, the choice of CFL must be based on the propagation of the fastest wave, in the fastest medium. In this case, the propagation speed of the pressure wave in bone is more than six times as fast as the propagation of the pressure wave in air (~2200m/s versus 343m/s). As a result, the effective maximum CFL when solving the Riemann problems with cells representing the media of air is approximately 0.15.

For example, by setting the Courant number (ν) to 0.9:

$$\nu = 0.9 = \frac{\Delta t}{\Delta x} cp_{bone} \qquad (4.3.2)$$

$$\frac{\Delta t}{\Delta x} = \frac{0.9}{2.24 \times 10^5 \frac{cm}{s}}$$

$$\frac{\Delta t}{\Delta x} \approx 4.018 \times 10^{-6} \frac{s}{cm} \qquad (4.3.3)$$

The resulting effective courant number associated with the propagation of the pressure wave through the air is:

$$\nu_{air} = \frac{\Delta t}{\Delta x} cp_{air} \qquad (4.3.4)$$

$$\nu_{air} \approx 4.018 \times 10^{-6} \frac{s}{cm} 34300 \frac{cm}{s}$$

$$\nu_{air} \approx 0.1376 \qquad (4.3.5)$$

Section 4.4: Limiters

The method outlined above is a first-order accurate. Although formal second-order accuracy cannot be achieved due to the discontinuous (shock) solution, accuracy can be improved through the use of wave limiters, which are designed to adjust the contribution of a specified wave, $\mathcal{W}^p$, based on the smoothness of the solution.[49] The limited wave, $\widetilde{\mathcal{W}}^p$, is defined as a scalar multiple of the corresponding wave determined with the Riemann Solver:

$$\widetilde{\mathcal{W}}^p_{i-1/2} = \phi\left(\theta^p_{i-1/2}\right)\mathcal{W}^p_{i-1/2} \tag{4.4.1}$$

The value $\phi(\theta)$ is determined from the dot-product of the wave with the corresponding wave at the upwind interface (behind the wave, as determined by the direction of propagation):

$$\theta^p_{i-1/2} = \frac{\mathcal{W}^p_{I-1/2} \cdot \mathcal{W}^p_{i-1/2}}{\mathcal{W}^p_{i-1/2} \cdot \mathcal{W}^p_{i-1/2}}, \quad I = \begin{cases} i-1, & s^p_{i-1/2} > 0 \\ i+1, & s^p_{i-1/2} < 0 \end{cases} \tag{4.4.2}$$

Common choices for limiter methods, defined by their associated function, $\phi(\theta)$, available in CLAWPACK are:

No Limiter: $\phi(\theta) = 1$

Minmod: $\phi(\theta) = \text{minmod}(1, \theta)$

$$\text{minmod}(a,b) = \begin{cases} a & if \ \ |a| < |b| \ and \ ab > 0 \\ b & if \ \ |b| < |a| and \ ab > 0 \\ c & if \ \ ab \leq 0 \end{cases}$$

Superbee: $\phi(\theta) = \max\left(0, \min(1, 2\theta), \min(2, \theta)\right)$

Monotonized Centered: $\phi(\theta) = \max\left(0, \min\left((1+\theta)/2, 2, 2\theta\right)\right)$

Van Leer: $\phi(\theta) = \frac{\theta + |\theta|}{1 + |\theta|}$

If the wave $\mathcal{W}^p_{i-1/2}$, and the upwind wave, $\mathcal{W}^p_{I-1/2}$, are identical, then $\theta^p_{i-1/2} = 1$, and all methods are equivalent, $\phi(\theta) = 1$. Effectively, this recognizes the smoothness of the solution and the resulting wave is not limited. If the waves are orthogonal, $\theta^p_{i-1/2} = 0$, and $\phi(\theta) = 0$ for all limiters except MC where $\phi(\theta) = 2$

An initial comparison of these methods in order to determine their appropriateness for implementation in modeling bTBI, is accomplished with a one-dimensional test problem. This problem consists of propagation of an appropriate waveform, defined by equation 3.4.1, from air (left), into a defined region of 6mm thickness of bone (middle), with brain material on the right side of the domain. Results are illustrated in Figure 10 at a point in time when the wave has reflected off the bone surface, and the transmitted portion has propagated into the brain. Optimally, the discontinuity should be represented by the sharpness of the curve in both materials (the term 'sharpness' is used to describe how closely the solution represents a step function at the appropriate location of the wave front). Due to the extreme discrepancies between material properties, the use of no limiter resulted in rapid instability at the material boundaries which dominated the domain prior to the time step illustrated in the figure. All four methods are characterized by varying amounts of smearing of the discontinuity, with the qualitatively best results achieved by the Superbee limiter. The minmod method proved the least capable in maintaining sharpness of the discontinuity. The extent of smearing present in Figure 10 is based exaggerated due to low resolution ($\Delta x = 0.1cm$).
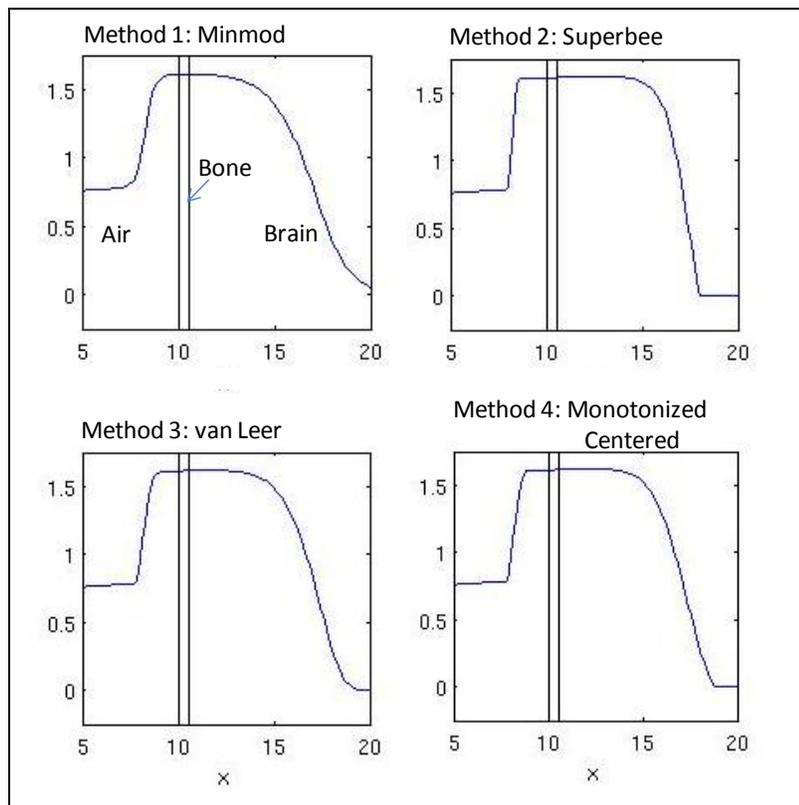
Figure 10: Comparison of limiter methods in 1-dimensional model of blast wave propagating from air (left side), through bone (6mm) and into brain (right side). Initial wave moving from left to right has reflected. Use of no limiter resulted in instability at the interface between materials which dominated the entire domain before the wave propagated to the interface. Grid resolution in x-axis is 10 cells per centimeter (x-axis units)
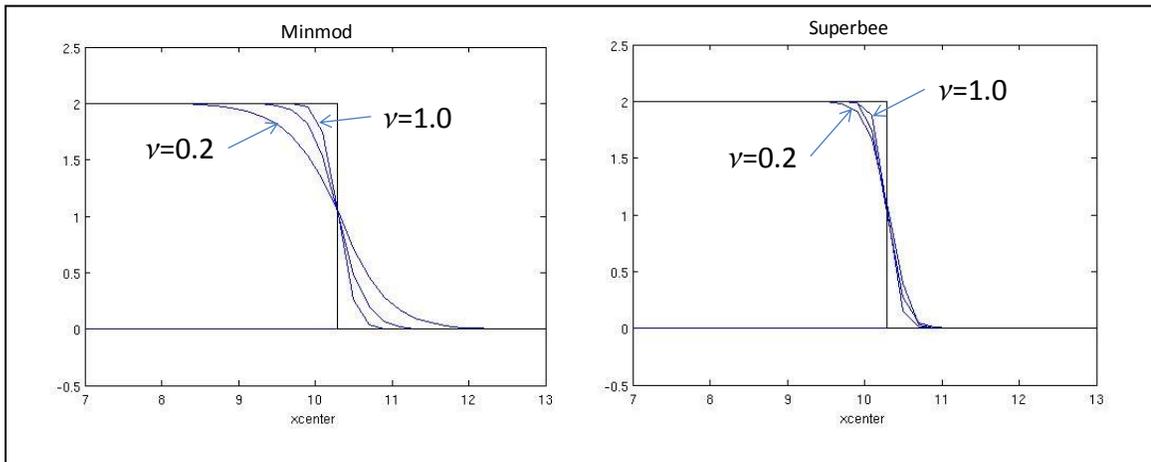
Figure 11: Comparison of minmod limiter (left) to superbee limiter (right). Plots include exact solution (perfect representation of the discontinuity) as well as solution values CFL numbers of 0.2, 0.9, and 1.0.

The effectiveness of the limiter should also be examined based on its performance with different values of the Courant number. The degree of smearing is expected to increase for lower Courant numbers. In a one-dimensional problem without material interfaces, the propagation of a discontinuity based on varying Courant number is illustrated in Figure 11. The Superbee limiter is much more effective at representing the discontinuity for smaller Courant numbers.

Section 4.5: Representation of Complex Material Geometry

Representation of complex geometries on discretized domains is a critical aspect to ensuring that the numerical modeling accurately represents the intended physical phenomenon. In this context, *complex geometry*, generally refers to boundaries of physical materials (i.e. brain, bone, etc.) that are not aligned with a Cartesian coordinate system. In a finite volume model, each computational cell is assigned a set of parameters (ρ, λ, and μ) that describes the material contained in that cell; boundaries between materials are therefore defined by varying the parameter values between adjacent cells. On a Cartesian grid system, it is impossible to exactly represent boundaries between materials that are not aligned with the coordinate system. As resolution is increased, accuracy can be expected to improve as the approximated boundary becomes closer to the intended physical boundary. Errors introduced by improper boundary representation can influence the behavior of the entire system; a higher-resolution grid essentially recreates a self-similar problem at a different scale, with the same sources of error. Appropriate boundary representation is essential in ensuring that the model represents the physical transmission, reflection, and refraction characteristics of the modeled interface.
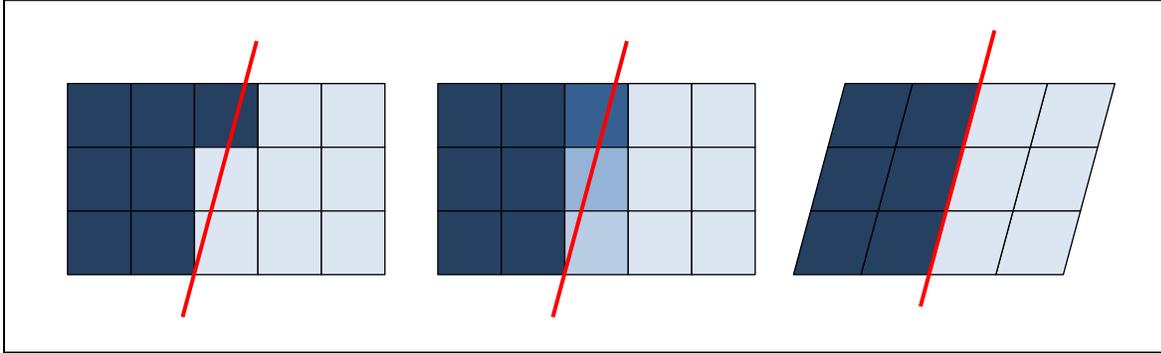
Figure 12: Pictorial representation of three methods for defining material boundaries within a computational domain; varying colors represent varying material properties. Left: Naïve approach; Middle: Cell Averaging; Right: Grid mapping.

A naïve approach to representing complex geometry would involve selecting material properties for each cell based on the location of a single point in the cell (such as the center) in relation to the boundary between materials, illustrated in Figure 12 (left). In this case, all boundaries are either aligned with the x or the y-axis, and significant error can be expected.

Perhaps a better approximation can be made by averaging the material properties for cells in which the boundary passes through, based on the percentage of the cell that lies on each side of the boundary (Figure 12 (middle)). This is not necessarily a straight-forward process, as there are various factors that must be considered in selecting an appropriate averaging technique. In CLAWPACK a subroutine, CELLAVE, computes the relative percentage of each cell that lies on each side of a defined material boundary (a separate subroutine, FDISC, must is used to define material boundaries). CELLAVE returns a value for a variable, w_1, that represents the percentage of the cell in the 'left' state defined in FDISC (the term, 'left,' is only used as a reference, and depending on the curve defined in FDISC, may not represent the physical left-right relationship at the boundary). Correspondingly, $w_2 = 1 - w_1$ is the percentage of the cell in the 'right' state.

The simplest averaging method is to take the arithmetic average of the material properties.

$$\rho_{average} = w_1\rho_l + w_2\rho_r, \ \mu_{average} = w_1\mu_l + w_2\mu_r, \ \lambda_{average} = w_1\lambda_l + w_2\lambda_r \quad (4.5.1)$$

Some literature suggests using an arithmetic average of density and a harmonic average of the Lamé parameters.[50] In the case where one of the materials is air with a shear modulus of zero, the harmonic average for the shear modulus is zero, determined as the limit as $\mu_* \to 0$ in the following calculation (the * symbol is used to generically refer to left or right).

$$\rho_{average} = w_1\rho_l + w_2\rho_r, \ \mu_{average} = \frac{1}{\frac{w_1}{\mu_l} + \frac{w_2}{\mu_r}}, \ \lambda_{average} = \frac{1}{\frac{w_1}{\lambda_l} + \frac{w_2}{\lambda_r}} \quad (4.5.2)$$

An alternate approach is to define cell properties such that the wave speed within the cell is the harmonic average of the wave speed of the two averaged properties. Depending on how the boundary bisects the averaged cell, the effective speed that a wave crosses the physical area modeled is the harmonic average of the wave's speed in the two materials. If the boundary is normal to the propagation of the wave, the time required for the wave to traverse the cell will be accurately represented; if the boundary is tangent to the direction of the incident wave, this method will not capture the splitting of the wave as it propagates at two different speeds in the different media.

Except in some specific cases, it is unlikely to be able to create cell properties such that the wave speeds of both the shear and the pressure waves can be made to equal the appropriate harmonic average. Additionally, the averaged cell should represent only a single reflecting surface. Standard averaging techniques, previously described, result in material properties that potentially create reflections on both sides of the averaged cell (based on a one-dimensional problem). This phenomenon can be avoided by defining the Lamé parameters such that the eigenvectors based on the averaged cell are a scalar multiple of the corresponding eigenvector for one of the materials. Figure 13 depicts possible wave propagation based on how a boundary intersects a cell, and the potential difference resulting from averaging (bottom).
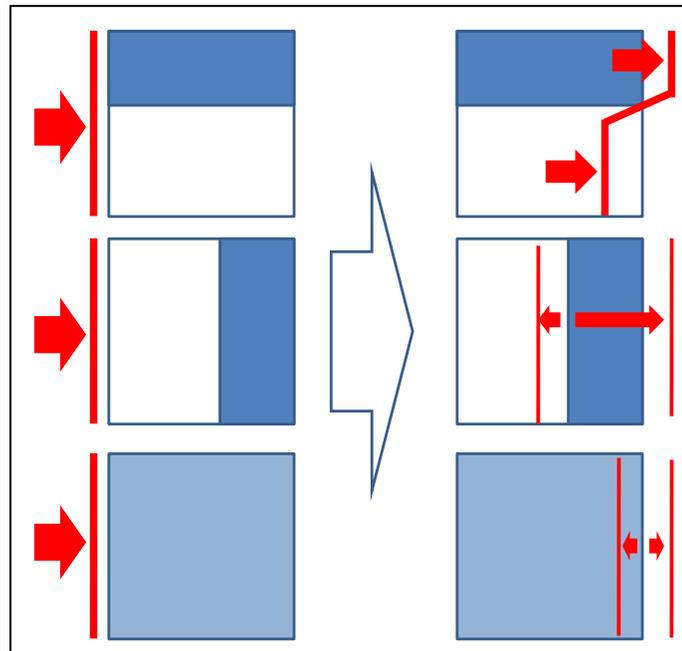
Figure 13: Illustration of potential errors associated with cell averaging, in the case of a plane wave propagating into a cell in which a boundary exists between two materials (darker shade represents material with greater speed of wave propagation). Top: Horizontal boundary; Middle: Vertical boundary; Bottom: Averaged cell properties result in same average regardless of boundary orientation.

Using the p-waves as the basis for defining the averaged properties, this method begins with determination of the harmonic average of the wave speed, $c_{p\ avg}$.

$$c_{p\ avg} = \frac{1}{\frac{w1}{c_{pl}}+\frac{w2}{c_{pr}}}$$

(4.5.3)

Determination of the material parameters necessary to ensure this condition is accomplished based on the eigenvector associated with the pressure wave. In two dimensions, the corresponding eigenvector for the p-wave is:

$$r^{1,2} = \begin{bmatrix} \lambda + 2\mu \\ \lambda \\ 0 \\ \pm c_p \\ 0 \end{bmatrix}$$

Consideration must be taken as to which property should be matched, based on percentage of each material within the cell. A logical approach is that the choice should be based on which material is contained in greater than 50% of the cell's volume. Through numerical experimentation, variations in this value did not significantly change results.

$$\lambda_{avg} = \frac{c_{p\ avg}}{c_{p*}} \lambda_*$$

(4.5.4)

$$\lambda_* = \begin{cases} \lambda_l & for\ w_1 > 0.5 \\ \lambda_r & for\ w_1 \le 0.5 \end{cases}$$

Using this method, care must be taken to ensure values are appropriate for the hyperbolic formulation of the system of PDEs. If the shear modulus, μ, is less than zero, hyperbolicity of the problem is lost based on the complex value for the eigenvalues associated with the shear waves.

$$\mu_{avg} = \max\ \left(0, \frac{1}{2}\left[\frac{c_{p\ avg}}{c_{p*}}(\lambda_* - 2\mu_*) - \lambda_{avg}\right]\right)$$

(4.5.5)

$$\rho_{avg} = \frac{\lambda_{avg} + 2\mu_{avg}}{c_{p\ avg}^2}$$

(4.5.6)

$$c_{s\ avg} = \sqrt{\frac{\mu_{avg}}{\rho_{avg}}}$$

(4.5.7)

Figures 14 and 15 depict parameter values and wave speeds for the averaged cell based on three methods just outlined. For the method based on the harmonic average of the p-wave speed, the density and the Lamé parameters are selected to match pressure wave eigenvectors for the state which occupies greater than 50% of the cell's volume. The computer code used to demonstrate the cell averaging methods is included in Appendix, MATLAB: Cell Averaging.
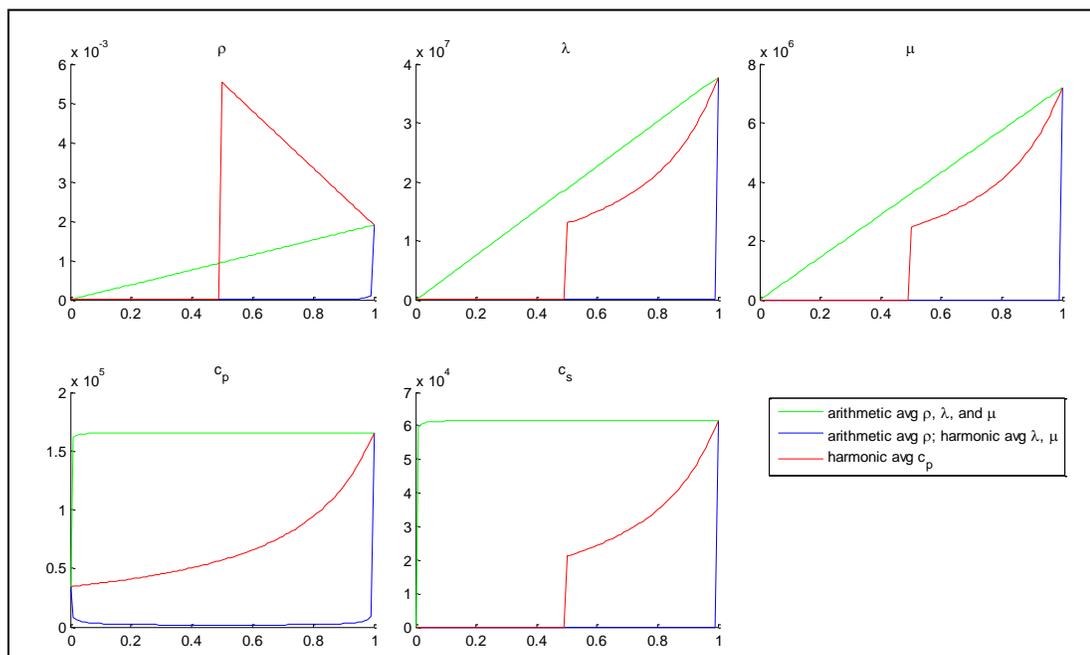
Figure 14: Comparison of methods for averaging material properties of bone and air. Graphs illustrate ρ (top left), λ (top middle), μ (top right), c_p (bottom left), and c_s (bottom right). x-axis represents percentage of bone.
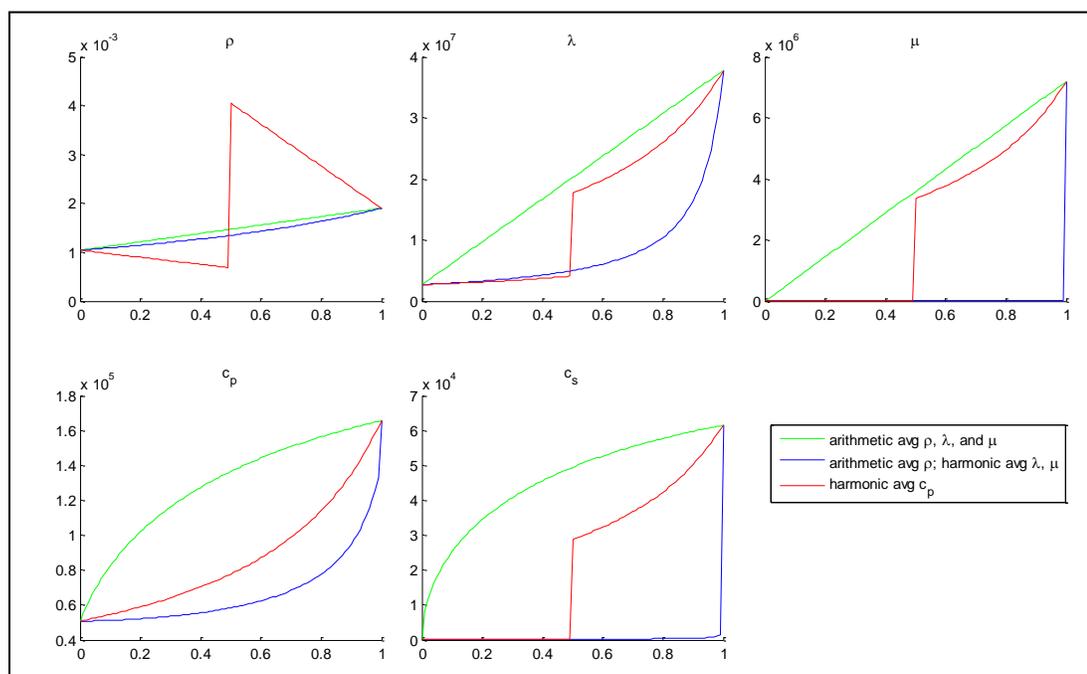
Figure 15: Comparison of methods for averaging material properties of bone and brain. Graphs illustrate ρ (top left), λ (top middle), μ (top right), c_p (bottom left), and c_s (bottom right). x-axis represents percentage of bone.

An alternative to the cell averaging is to map the vertices of each cell such that the boundary between materials lies along the cell boundaries, as described by Donna Calhoun, et al.[51] Creating a computational mesh that matches the geometry of a problem is frequently used in other high-resolution methods for solving PDEs, such as finite element methods. A significant benefit to the structure of the finite volume methods implemented in CLAWPACK is adherence to a logically rectangular structure. This reduces computational cost by establishing simple indexing references for each cell; in a two dimensional problem, each cell has exactly four neighbors, which can be referenced as left, right, top and bottom, relative to the cell.   With cell mapping, computational challenges can arise from irregular geometry and a large ratio between the sizes of cells.

Due to the complexity of grid mapping to complex shapes such as a skull, in order to study the effectiveness of this method, a circular representation of the skull is initially examined. This way, the effectiveness of grid mapping can be compared to other methods before determining the much more complex mappings required to reflect the actual shape of the skull. Because the mapping must be such that cell boundaries correspond with all material boundaries, a mapping technique is applied that maps the points of a Cartesian grid to concentric circles. A radial coordinate system cannot be implemented because, in addition to the fact that it is not logically rectangular, computational difficulties would arise at the center where all points converge.

The mapping implemented for a circular representation is based on designation of two radii from a center point: $r1$ is the radius inside of which points are mapped to circles, and $r2$ is the radius outside of which all points remain Cartesian. The mapping function maps a Cartesian point, (x_c, y_c), to a mapped point, (x_p, y_p). When implementing this mapping, it is important to consider that the number of grid cells must be selected such that the inner and outer boundaries of the skull region coincide with the cells, and that the width of the region is defined with a sufficient number of cells.

Using the origin of the Cartesian grid as the center reference point, the following algorithm is used to conduct the mapping: First, each point is designated a diameter, d, of the circle it will be mapped to, and a radius, r, of its position on the Cartesian grid.

$$d = \max(|x_c|, |y_c|) \tag{4.5.8}$$

$$r = \sqrt{x_c^2 + y_c^2} \tag{4.5.9}$$

Inside of $r1$, designated by $d \leq r1$, the vertices are mapped to a new location based on the required diameter of the mapped point:

$$x_p = d\frac{x_c}{r}, \; y_p = d\frac{y_c}{r} \tag{4.5.10}$$

Between $r1$ and $r2$ an interpolation factor, w, is required. In this case, a square function was found to have better results than a linear relationship, in reducing the ratio between largest and smallest cells in the domain. Because the CFL condition requires a time step based on cell size, if this ratio is large, a smaller time step will be required based only a few cells.

$$w = \left(\frac{r2-d}{r2-r1}\right)^2 \tag{4.5.11}$$

$$x_p = wd\frac{x_c}{r} + (1-w)x_c$$

$$y_p = wd\frac{y_c}{r} + (1-w)y_c$$

In the area outside of $r2$, the Cartesian coordinates remain unchanged:

$$x_p = x_c;\ y_p = y_c \tag{4.5.12}$$

This mapping is illustrated in Figure 16 which shows the mapped grid for different choices of $r2$, as well as ratio between the largest and smallest grid cells.

With some modifications, the Riemann problem can be solved for problems in which the discontinuity is not aligned with the Cartesian grid.[52] Implementation of a Riemann Solver on a mapped domain requires consideration of the non-uniformity of the grid, and must take into account differing: boundary lengths, boundary orientations, and cell sizes. The subroutine that implements this Riemann solver in CLAWPACK is included in Appendix, CLAWPACK: Riemann Solver on non-uniform grid. The mapping routine is included in Appendix, CLAWPACK: Grid Mapping.
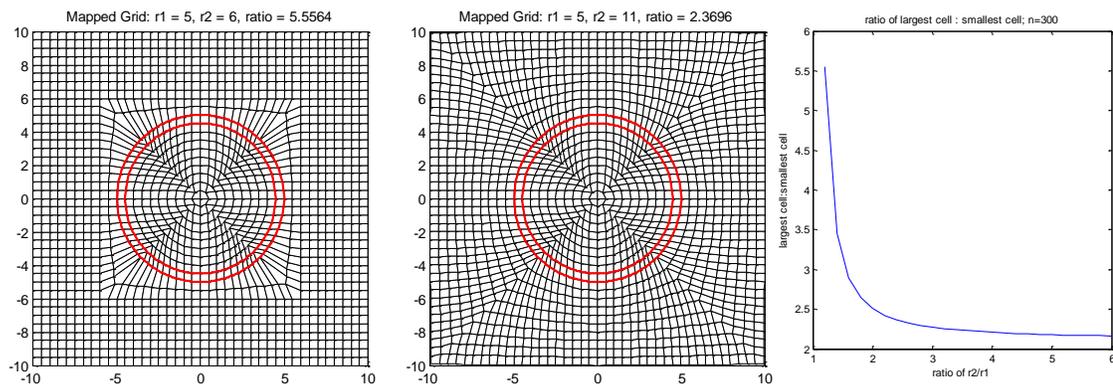
Figure 16: Effects of grid mapping on grid size of computational grid cells. Left: r1=5, r2=6; Center: r1=5, r2=10; Right: ratio between largest and smallest grid cells for r1=5 and 6<r2
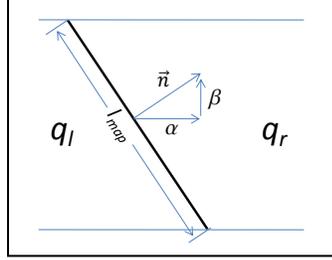
Figure 17: Solving the Riemann problem with non-Cartesian boundary. The variable, n, represents a unit normal to the mapped grid boundary,

First, the solution on each side of the boundary must be rotated based on the unit normal to the boundary, $\vec{n}$.

$$\begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_n \\ v_n \end{bmatrix} \tag{4.5.13}$$

$$\begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \begin{bmatrix} \sigma^{11} & \sigma^{12} \\ \sigma^{12} & \sigma^{22} \end{bmatrix} \begin{bmatrix} \alpha & -\beta \\ \beta & \alpha \end{bmatrix} = \begin{bmatrix} \sigma_{normal} & \sigma_{shear} \\ \sigma_{shear} & \sigma_{tangential} \end{bmatrix}$$

Determination of the wave strengths, $\vec{\alpha}$, is then accomplished using the same method as for the cartesian grid, based on the differences of the rotated solution on each side. The corresponding waves, however, must account for the rotation as well. In this case, the generalized form of the matrix, $\check{A} = \alpha A + \beta B$, with corresponding eigenvectors must be used:

$$\check{r}^{1,2} = \begin{bmatrix} \lambda + 2\mu\alpha^2 \\ \lambda + 2\mu\alpha^2 \\ 2\mu\alpha\beta \\ \pm\alpha c_p \\ \pm\beta c_p \end{bmatrix}, \quad \check{r}^{3,4} = \begin{bmatrix} 2\mu\alpha\beta \\ 2\mu\alpha\beta \\ [\alpha^2 - \beta^2]\mu \\ \mp\beta c_s \\ \pm\alpha c_s \end{bmatrix}, \quad \check{r}^5 = \begin{bmatrix} \beta^2 \\ \alpha^2 \\ -\alpha\beta \\ 0 \\ 0 \end{bmatrix} \tag{4.5.14}$$

In order to determine the flux, the speed of the wave must be scaled based on the ratio of the length of the side to the corresponding side of the Cartesian grid, as well as the ratio of the cartesian grid size ($dxdy$) to the area of the mapped grid cell, a.

$$s^{1,2} = \mp c_p \left(\frac{l_{map}}{d\star} \right) \left(\frac{dxdy}{a} \right) \tag{4.5.15}$$

$$s^{3,4} = \mp c_s \left(\frac{l_{map}}{d\star} \right) \left(\frac{dxdy}{a} \right)$$

$$d\star = \begin{cases} dx, & Riemann\ problem\ in\ x-direction \\ dy, & Riemann\ problem\ in\ y-direction \end{cases}$$

A simplified test problem is used to compare the five methods described above and summarized as follows:

1. Arithmetic Average: Arithemetic average of ρ, λ, and μ.

2. No Avaraging: Naïve approach, cell properties based on location of the center of the cell relative to boundary.

3. c_p Averaging: Parameters ρ, λ, and μ are selected to enforce c_p in the averaged cell is the harmonic average of wave speed in the two media.

4. Harmonic Averaging: Harmonic average is used for λ and μ; arithmetic average is used for ρ.

5. Grid Cell Mapping: Vertices are mapped such that boundary lies on interface between grid cells—no averaging is required.

The test problem consists of a square pulse pressure wave, propagating from a region represented with material properties of air, into a region represented by bone material properties. The boundary between the media consists of a piecewise linear function such that there is a area in which the boundary crosses cells with a slope of 3:1 , between two segments aligned with the Cartesian directions. Sample results from this test problem are contained in Figures 18 and 19.

The noticeable differences in results between the various methods illustrate the importance of capturing boundaries as accurately as possible. In theory, each method accurately reflects the wave propagation given the geometry and material properties defined by the material properties of the grid cells and the orientation of their boundaries. Only the cell mapping method accurately reflects both of these characteristics, and is therefore expected to provide the results most closely modeling the desired materials and boundary. Optimally, the test problem should match a physical experiment in which data collected would be representative of the true solution.

Qualitative analysis of the results from different methods indicates that the pressure wave averaging technique (Figure 18 (bottom)) is most consistent with the cell mapping technique (figure 19 (bottom)). Additionally, the detail views (right side plots in Figure 19 (middle), and Figure 19 (top)) illustrate the instability emerging, particularly in the harmonic averaging technique.
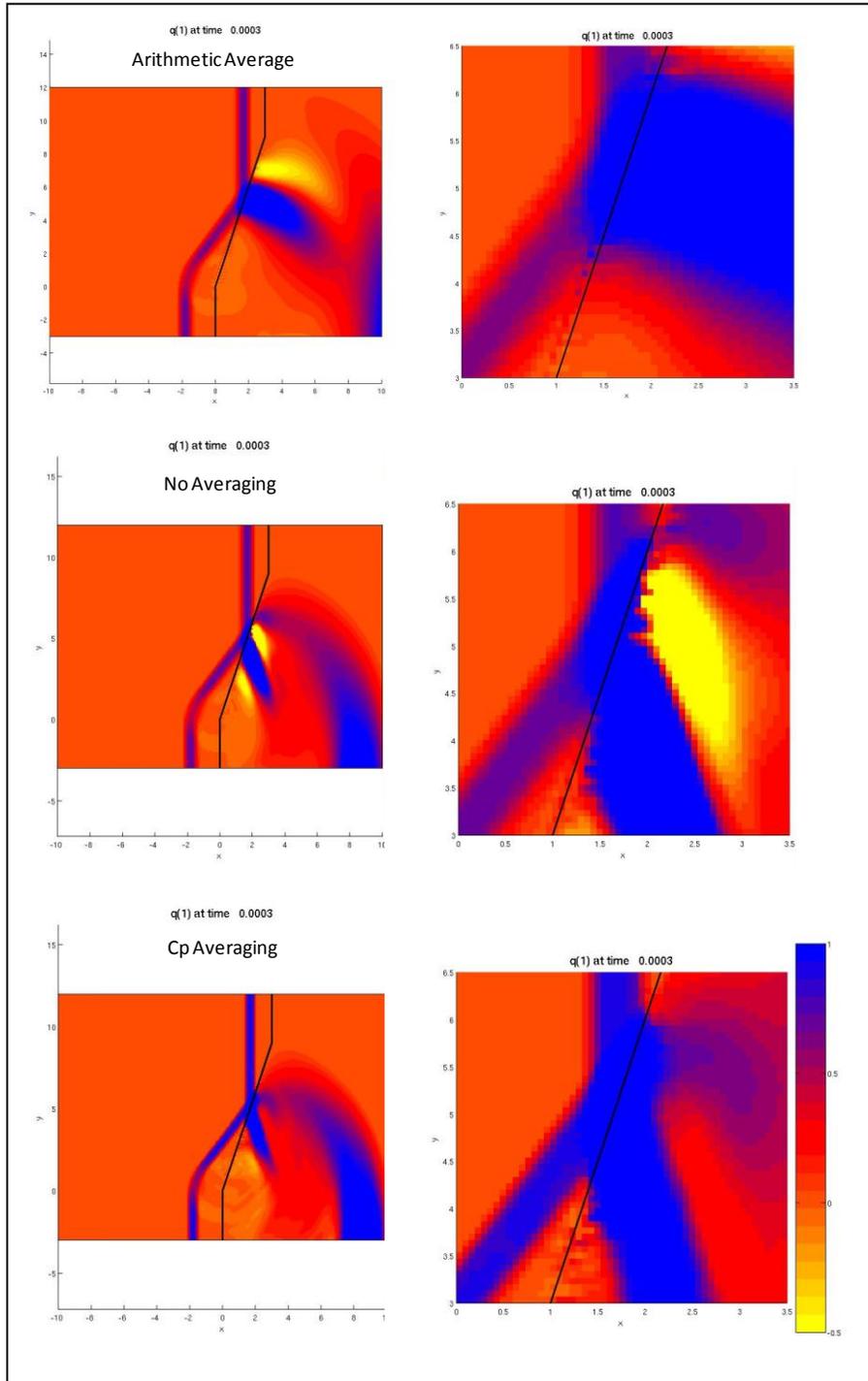
Figure 18: Comparison of methods (1). Propagation of a square pulse wave from left to right across the domain with boundary representing discontinuity between air (left side) and bone (right side). Top: Arithmetic average of ρ, λ, and μ. Middle: No averaging (naïve approach), cell values based on which side of boundary the center of the cell is located. Bottom: Parameters designed to achieve harmonic average of c_p in averaged cell.
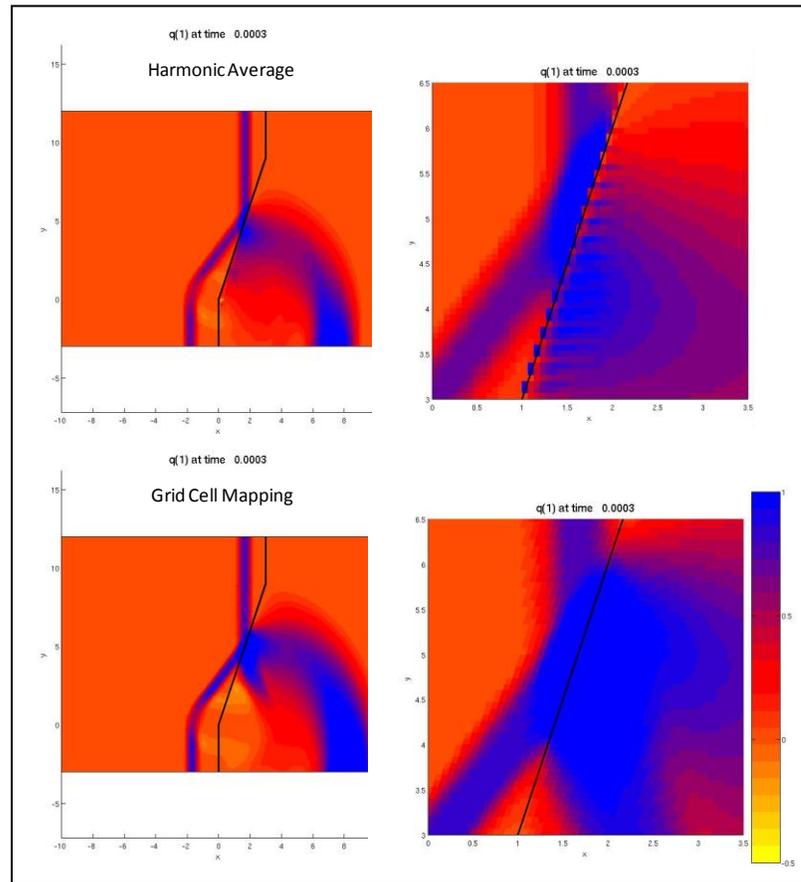
Figure 19: Comparison of methods (2). Top: Harmonic average of Lamé parameters, and arithmetic average of density. Bottom: Grid cell mapping to coincide with material boundary.

Results and Conclusions

For the initial numeric experiments, a waveform was selected to represent 20kg TNT at 8 meters distance, a blast that is consistent with a road-side improvised explosive device (IED), with the victim exposed in a vehicle passing by. The associated overpressure is .822atm≈83.3kPa[**] with an overpressure duration of 4.54msec. Based on the human threshold reference chart (Figure 3), the pressure wave from this blast is non-lethal, above the eardrum rupture threshold and below the 50% eardrum rupture line.

Using a plane wave and material properties constant in the y-dimension, it is possible to examine the reflection/transmission characteristics of the pertinent material interfaces as one-dimensional slices.   The linear-elastic response of the tissue results in almost no loss of energy within the head; waves propagating from bone into air are almost entirely reflected back into the skull. Initial reflection and transmission from air, through the skull, into the brain is illustrated in Figure 20.

Extension to a simplified skull model began with a circular representation of the skull with thickness of 6mm, and outer radius of 5cm. Figure 21 illustrates the progression of the pressure wave through the skull and brain. Pressure values represent the pressure trace, $\sigma^{11} + \sigma^{22}$ in units of atmospheres. The domain is a Cartesian grid with mx=my=600. This test problem was conducted using the various methods for establishing material boundaries as well as varying the Courant number, choice of limiter and inclusion of the transverse solver. Figure 21 illustrates progression of the wave from $t = 0.15$msec to $t = 0.57$msec. As this represents
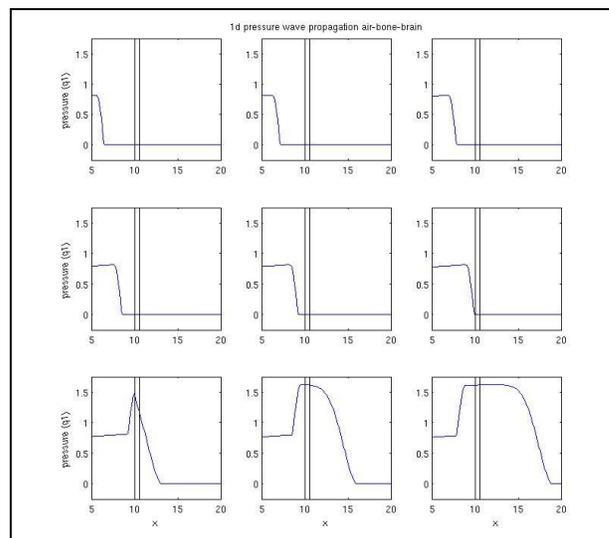
Figure 20: 1-Dimensional propagation of blast pressure wave from air through bone, and into brain. Grid resolution is 10 cells per centimeter.

[**] 100kPa≈ .9869atm

only approximately one-eighth of the time of the positive phase, ensuring stability in the method is essential if one desires to model the entire waveform.   Computational time varied significantly based on Courant number and inclusion of the transverse solver, and was not recorded, but was approximately three hours for the conditions above.
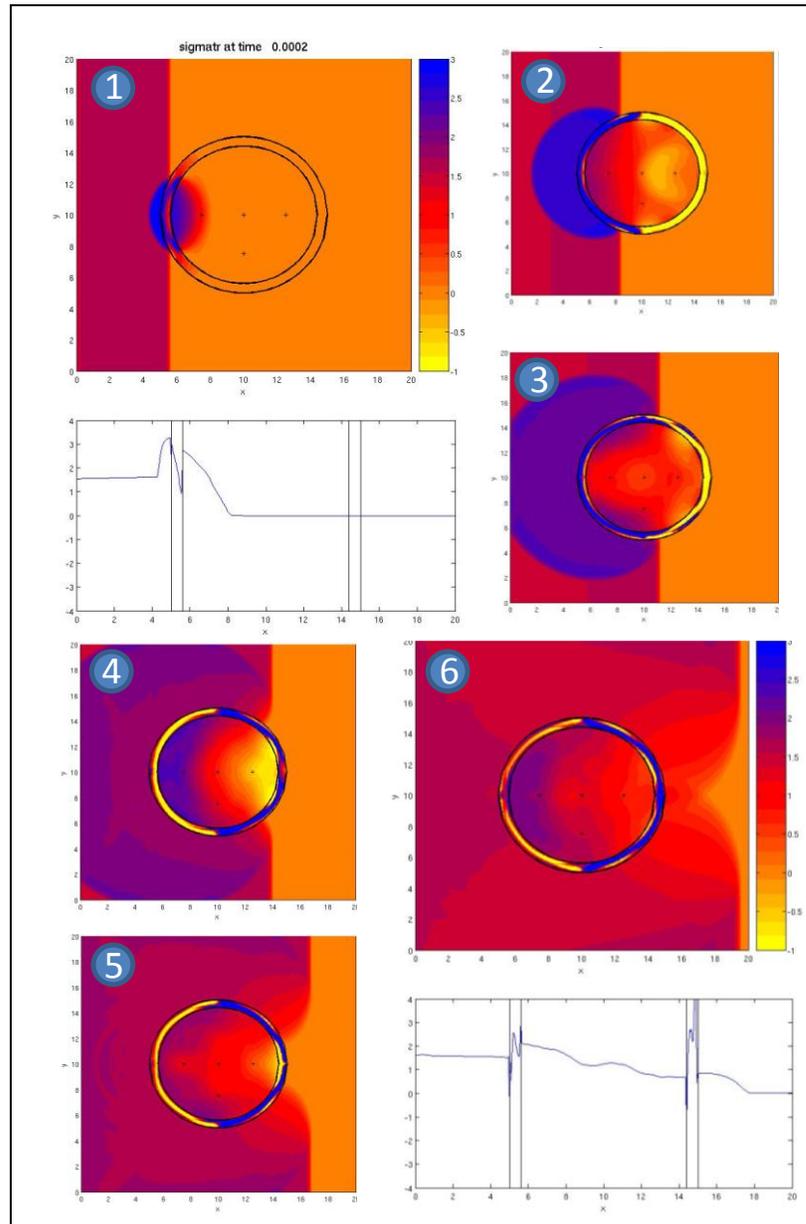
Figure 21: Explosive blast pressure wave propagating through circular representation of skull. Time progress: Frame 1: $t \approx 0.15msec$, Frame 2: $t \approx 0.24msec$, Frame 3: $t \approx 0.33msec$, Frame 4: $t \approx 0.41msec$, Frame 5: $t \approx 0.49msec$, Frame 6: $t \approx 0.57msec$. Frames 1 and 6 include cross section at $y = 10$. Grid resolution is mx=my=600.

This examination in not intended to draw conclusions directly regarding mechanism of bTBI, but to establish an understanding of the methods used to continue model development extending to three dimensions and including more accurate representations of material geometries and properties, from which more applicable results may be attained. Conclusions established based on the research completed are summarized as follows:

1. Choice of methods for representing material boundaries is critical. Consistent results were established between arithmetic averaging and using the harmonic average of the pressure wave speed; however, grid mapping and harmonic averaging of Lamé parameters proved extremely unstable. For the mapped grid, instability initially occurs at points where grid cells are nearly triangular, as illustrated in Figure 22. Instability in the harmonic averaging method is shown in Figure 23.

2. Inclusion of the transverse solver did not produce significantly different results and often increased the onset of instability
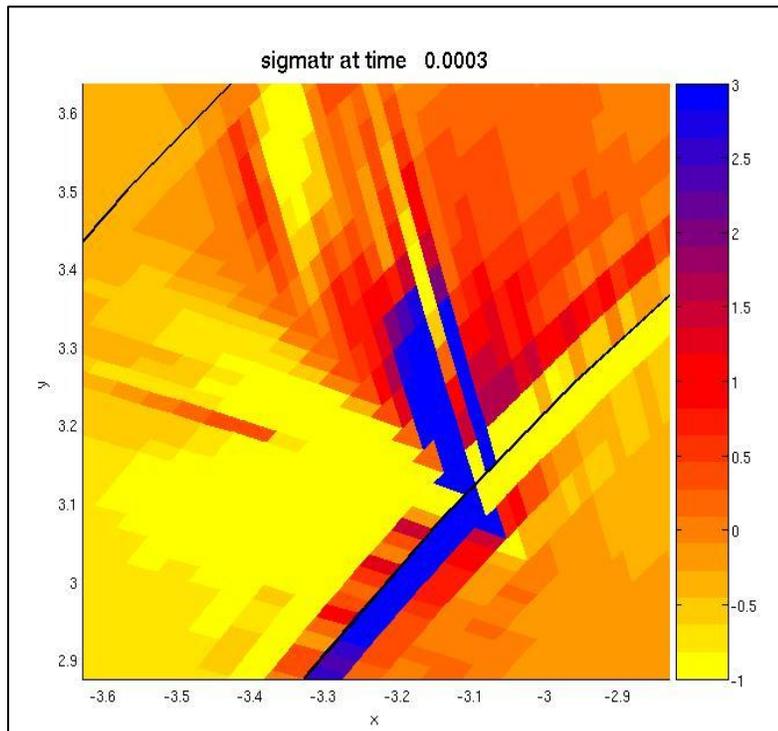
Figure 22: Instability in circular mapped grids: detail view of upper left quadrant of circular skull representation, where rows and columns converge, and the resulting quadrilateral cells become nearly triangular.
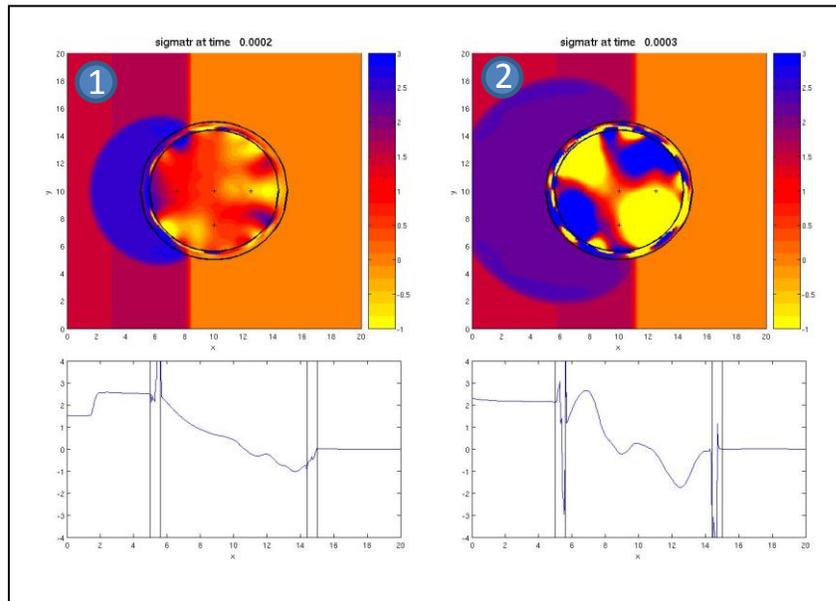
Figure 23: Instability in harmonic averaging method.

3.   Relatively thin width of skull, in terms of number of grid cells, is a source of instability, evidenced by the noticeable pattern of pressure values within the ring of the skull in Figure 21. Given longer simulation time periods, these values continued to grow to the point of causing instability throughout the domain. Higher resolution is required for better representation of the behavior within the bone.

4.   Resulting shear stress within the brain cavity were on the order of 10^{-5} atmospheres or 10^{-3} kPa. This is well below the injury thresholds identified by Zhang. This may be a result of various potential factors such as: homogeneous isotropic material properties, limitation of the model to two dimensions, regular geometry without modeling realistically all the intricacies of the material shapes, and reduction of strain rates due to smearing of the shock front across multiple grid cells.

5.   Reducing Courant number can increase stability by limiting the error associated with wave propagation beyond the boundary of adjacent cells. Courant number of 0.4 significantly improved stability over Courant number of 0.9.

6.   Choice of limiters is important, although superbee most effectively maintains a sharp discontinuity; it creates instability in the solution when propagating across boundaries between materials of vastly different properties. Without a limiter, rounding errors almost immediately become unstable at the material interfaces.
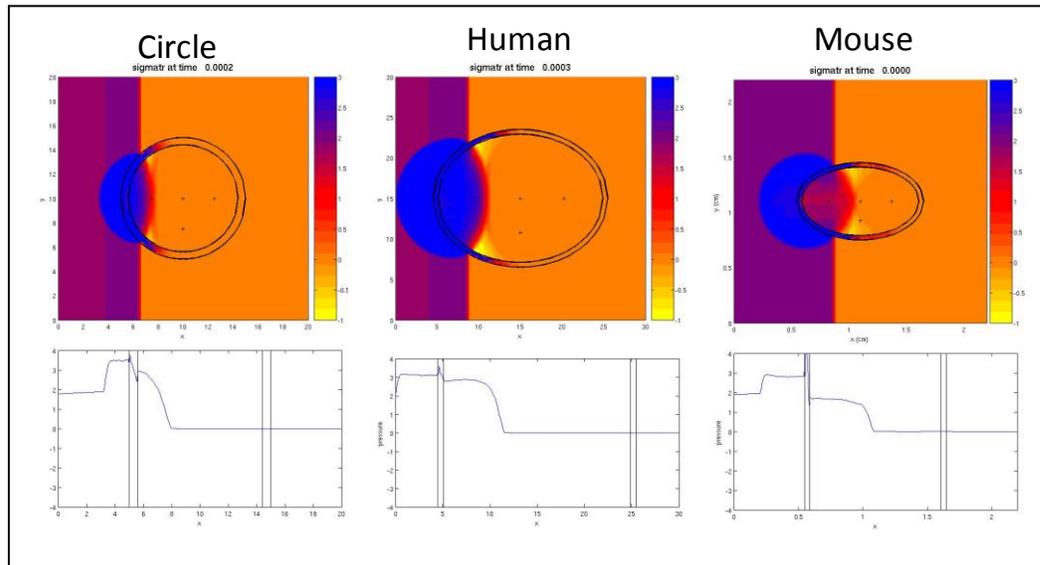
Figure 24: Comparison of circular and oval shapes representing appropriate scaling for human and mouse.

Parallel efforts between numerical modeling and live animal experimentation are mutually supporting. Effective numerical modeling can be used to determine the appropriate waveforms that should be created in a shock tube in order to cause injury to the test subject consistent with bTBIs in humans. Examination of data collected from the physical experiment can then be used to validate the efficacy of the model. Consistent results would support scaling of the numerical model to representation of the human skull. Consideration must be made for varying material properties between human and mouse skulls. Evidence indicates that the elastic response of the mouse brain is age-dependent,[53] therefore selection of wave characteristics and associated numeric modeling parameters, may vary between specimens. Additionally, the shape of the skull and relative thickness vary between the different species of mice. Figure 24 illustrates a relative comparison of geometries between the circular test problem and oval representations of a human skull and a mouse skull. Material properties are identical in all three examples.

An important objective of this research is to create a foundation for continued research and continue refinement of finite volume models. Specific areas for future work include:

1. Extension to three dimensions. Practical results cannot be obtained without including the three-dimensional aspects of the head.
2. Refinement of cell mapping algorithms. Grid cell mapping is a very effective concept for accurately capturing boundaries between materials within a domain; however, for this application it cannot be implemented unless a method is developed that allows its use on irregular geometry.

3.   Adaptive Mesh Refinement (AMR). Application of AMR algorithms to this problem can increase the accuracy of the solution at discontinuities. Due to the small physical domain and area occupied with the skull, coupled with numerous reflecting shocks, cost of implementing adaptive mesh refinement may be large compared to increasing resolution of entire domain.

4.   Increasing the accuracy of material properties. More realistic behavior can be achieved through better representation of the physical material properties as well as their geometries. Non-linear representation of visco-elasticity may be critical in accurately modeling the response of the brain.

5.   Experimental collaboration. Comparison of numeric results to results from physical experimentation is a vital step in validating the accuracy of the model. Validation of a numerical model can only be achieved through comparison to the physical phenomenon. An approach to accomplish this is by comparing results of the numerical model to those collected through controlled laboratory experiments. For this reason, it is prudent to include within the scope of numerical experimentation geometries and material properties of test animals. The Puget Sound Veterans' Affairs Medical Center is currently working to incorporate a shock tube into their study of bTBI, capable of creating injuries in mice similar to those experienced by soldiers in combat.[54] Additionally, physical experimentation with non-biological tissue can provide important data for comparison.

58

ENDNOTES

[1] Nabil M. Elsayed and James L. Atkins (eds), *Explosion and Blast-Related Injuries,* (Burlington, MA: Elsevier Academic Press, 2008), 109-119.

[2] Randall J. Leveque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge, (New York: University Press, 2002).

[3] Yudhijit Bhattachargee, "Neuroscience: Shell Shock Revisited: Solving the puzzle of Blast Trauma," *Science,* 319, (2008), 406-408.

[4] Amy Courtney and Michael Courtney "Links between traumatic brain injury and ballistic pressure waves originating in the thoracic cavity and extremities," *Brain Injury*, 21:7, (2007), 657 — 662.

[5] David F. Moore, et al., "Computational Biology – Modeling of Primary Blast Effects on the Central Nervous System," *NeuroImage*, (2009), doi: 10.1016/j.neuroimage.2009.02.019.

[6] Department of Defense, "Summary of Meeting Proceedings: Non-Impact Blast-Induced Mild Traumatic Brain Injury," Herndon, VA, (2009).

[7] Brain Injury Association of America, http://www. biausa.org/Pages/types_of_brain_injury.html#tbi, (accessed 10 May 2009).

[8] Brain Injury Association of America, "Traumatic Brain Injury in the United States: A call for Public/Private Cooperation," (2007).

[9] Charles W. Hoge, et al., "Mild Traumatic Brain Injury in U.S. Soldiers Returning from Iraq," *The New England Journal of Medicine,* 358 no. 5 (2008), 453-463.

[10] Elsayed, 17.

[11] William C. Moss, et al., "Skull Flexure from Blast Waves: A New Mechanism for Brain Injury with Implications for Helmet Design," (2008).

[12] J. Nerenberg, et al., "The Effectiveness of Different Personal Protective Ensembles in Preventing Blast Injury to the Thorax" (2008).

[13] 785th Medical Company, "Concussion Management in a Mature Theater: OIF 2007-2008," (2008).

[14] Brain Injury Association of America, "Traumatic Brain Injury in the United States: A Call for Public/Private Cooperation," (2007).

[15] 785th Medical Company.

[16] Michael Courtney, *Ballistics Testing Group*, http://www.ballisticstestinggroup.org/, (accessed May 2009).

[17] Kinney and Graham, 2.

[18] Jeff Slotnick, *Explosive Forces of Improvised Explosive Devices*, http://www.securitydiver.com/aic/stories/aritcle-114.html (accessed August 2008).

[19] Elsayed and Atkins, 167.

[20] Kinney and Graham, 94-100.

[21] Ray Bennett, (02 24 2009).

[22] John P. Sullivan, et al., *Jane's Unconventional Weapons Response Handbook*, (Alexandria, VA: Janes Information Group, 2002), 53.

[23] Anne E. Wildegger-Gassmaier, "Aspects of Thermobaric Weaponry," *ADF Health,*4 (2008), 3-6.

[24] Elsayed and Atkins, 172.

[25] Leveque, 493-497.

[26] Giovanni Belingardi, Giorgio Chiandussi, and Ivan Gaviglio, "Development and Validation of a New Finite Element Model of Human Head," (Politecnico di Torino, Dipartimento di Meccanica Italy, Paper Number 05-0441 http://www-nrd.nhtsa.dot.gov/pdf/esv/esv19/05-0441-O.pdf).

[27] S. Sarkar, S. Majumder, and A. Roychowdhury, "Response of human head under static & dynamic load using finite element method," Trends Biomater. Artif. Organs 17 (2004), 130-134.

[28] Liying Zhang, K.H. Yang, and A.I. King, "A Proposed Injury Threshold for Mild Traumatic Brain Injury," *Transactions of the ASME*, Vol 126, "2004", 226-236.

[29] S.J.A. Michielsen, "Mechanical properties of brain tissue under transient loading: a numerical-experimental approach," Eindhoven University of Technology, Department of Mechanical Engineering, Laboratory of Biomechanics (1996).

[30] Moore, et al.

[31] Moss, et al.

[32] Christopher W. Connor, Gret T Clement and Kullervo Hynynen, "A unified model for the speed of sound in cranial bone based on genetic algorithm optimization," *Physics in Medicine and Biology* 47 (2002) 3935.

[33] Xiangtao Yin, and Kullervo Hynynen, "A Numerical Study of Pulsed Sonication For Reducing Thermal Deposition in The Skull During Transcranial Focused Ultrasound Surgery," IEEE Ultrasonics Symposium (2005), 1242.

[34] F. Rupin, et al., "Determination of Poisson Ratio and Young Modulus by Scanning Acoustic Microscopy Combined with Nanoindentation," *Journal of Clinical Densitometry*, Volume 12, Issue 1, Pages 124-125

[35] R.M. Pidaparti and A. Vogt, "Experimental investigation of Poisson's ratio as a damage parameter for bone fatigue," *Journal of Biomedical Materials Research*, 59 (2002) 282-287.

[36] Brain Injury Association of America, http://www.biausa.org/Pages/types_of_brain_injury.html#symptoms.

[37] Katherine H. Taber, and Robin A. Hurley, "Traumatic Axonal Injury: Atlas of Major Pathways," *The Journal of Neuropsychiatry and Clinical Neurosciences*, 19 (2007), iv-104.

[38] Robert P. Granacher, *Traumatic Brain Injury*, 2nd Edition, (CRC Press, 2002), 6-7.

[39] Zhang, et al., 233.

[40] Andre Pennardt, "Blast Injuries," *eMedicine,* http://emedicine.medscape.com/article/822587-media

[41] I.G. Browen, E.R. Fletcher, and D.R. Richmond, "Estimate of Man's Tolerance to the Direct Effects of Air Blast," (Lovelace Foundation for Medical Education and Research, 1968), 26-28.

[42] LeVeque, 492-501.

[43] J.G. Harris, *Linear Elastic Waves,* (Cambridge University Press, 2001), 6.

[44] A.E.H. Love, *A Treatise on the Mathematical Theory of Elasticity*, (New York: Dover Publications, 1944), 102.

[45] Slotnick, http://www.securitydiver.com/aic/stories/aritcle-114.html (accessed August 2008).

[46] LeVeque, 64.

[47] Ibid. 52.

[48] Ibid. 68,78.

[49] Ibid. 181-183.

[50] Christoph T. Schröder and Waymond R. Scott Jr., "A Finite-Difference Model to Study the Elastic-Wave Interactions with Burried Land Mines*," IEEE Transactions on Geoscience and Remote Sensing,* 38 (2000) 1505-1512.

[51] Donna A. Calhoun, Chirstiane Helzel, and Randall J. LeVeque, "Logically Rectangular Grids and Finite Volume Mehtods for PDEs in Circular and Spherical Domains," *SIAM Review,* (2008), 723-752.

[52] LeVeque, 519-527.

[53] Amit Gefen, et al., "Strain Distribution in Brain Tissue of Rats Subjected to Closed-Head Injury is Age-Dependent," *2005 Summer Bioengineering Conference*, (June 22-26, Vail Colorado).

[54] David G. Cook, (02 24 2009).

BIBLIOGRAPHY

785th Medical Company. "Concussion Managment in a Mature Theater: OIF 2007-2008." 2008.

Belingardi, Giovanni, et al. "Development and Validation of a New Finite Element Model of Human Head." Politecnico di Torino, Dipartimento di Meccanica. http://www-nrd.nhtsa.dot.gov/pdf/esv/esv19/05-0441-O.pdf.

Bennett, Ray, Baker Engineering and Risk Consultants. 02 24, 2009.

Bhattachargee, Yudhijit. "Neuroscience: Shell Shock Revisited, Solving the Puzzle of Blast Trauma." *Science*, January 25, 2008: 406-408.

Bowen, I.G., E.R. Fletcher, and D.R. Richmond. *Estimate of Man's Tolerance to the Direct Effects of Air Blast.* Defense Atomic Support Agency, Albuquerque, NM: Lovelace Foundation for Medical Education and Research, 1968.

Brain Injury Association of America. "Traumatic Brain Injury in the United States: A call of Public/Private Cooperation." 2007.

—. *Types of Brain Injury.* http://www.biausa.org/Pages/types_of_brain_injury.html#tbi (accessed May 10, 2009).

Calhoun, Donna A., Christiane Helzel, and Randall J. LeVeque. "Logically Rectangular Grids and Finite Volume Methods for PDEs in Circular and Spherical Domains." *SIAM Review*, 2008: 723-752.

Connor, Christopher W., Gret T. Clement, and Kullervo Hynynen. "A unified model for the speed of sound in cranial bone based on genetic algorithm optimization." *Physics in Medicine and Biology* (Institute of Physics Publishing) 47 (2002): 3925-3944.

Cook, David G., VA Medical Center. 02 24, 2009.

Courtney, Amy, and Michael Courtney. "Links Between Traumatic Brain Injury and Ballistic Pressure Waves Originating in the Thoracic Cavity and Exremities." *Brain INjury* 21, no. 7: 657-662.

Courtney, Michael. *Ballistics Testing Group.* May 2009. http://www.ballisticstestinggroup.org/ (accessed May 2009).

Department of Defense. "Summary of Meeting Proceedings: Non-Impact Blast-Induced Mild Traumatic Brain INjury." *International State-of-the-Science Meeting.* Herndon, VA: Department of Defense, 2009. 1-13.

Elsayed, Nabil M., and James L. (editors) Atkins. *Explosion and Blast-Related Injuries.* Burlington, MA: Elsevier Academic Press, 2008.

Fagnan, Kirsten, Randall J. LeVeque, Thomas J. Matula, and Biran MacConaghy. "High-resolution finite volume methods for extracorporeal shock wave therapy."

Gefen, Amit, et al. "Strain Distribution in Brain Tissue of Rats Subjected to Closed-Head Injury is Age-Dependent." *2005 Summer Bioengineering Conference.* Vail, CO, 2005.

Granacher, Robert P. *Traumatic Brain Injury (2nd Ed).* CRC Press, 2002.

Harris, J. G. *Linear Elastic Waves.* Cambridge University Press, 2001.

Hoge, Charles W., et al. "Mild Traumatic Brain Injury in U.S. Soliders Returning from Iraq." *The New England Journal of Medicine* 358, no. 5 (January 2008): 358:453-463.

Kinney, Gilbert F., and Kenneth J. Graham. *Explosive Shocks in Air.* New York: Springer-Verlag New York Inc., 1985.

LeVeque, Randall J. *Finite Volume Methods for Hyperbolic Problems.* New York: Cambridge University Press, 2002.

Love, A.E.H. *A Treatise on the Mathematical Theory of Elasticity.* New York: Dover Publications, 1944.

Michielsen, S.J.A. "Mechanical properties of brain tissue under transient loading: a numerical-experimental approach." Eindhoven University of Technology, Depratment of Mechanical Engineering, Laboratory of Biomechanics, June 1996.

Moore, David F, et al. "Computational Biology - Modeling of Primary Blast Effects on the Central Nervous System." *NeuroImage*, 2009.

Moss, William C., et al. "Skull Flexure from Blast Waves: A New Mechanism for Brain Injury with Implications for Helmet Design." 2008.

Nerenberg, J, et al. "The Effectiveness of Different Personal Protective Ensembles in Preventing Blast Injury to the Thorax."

Pennardt, Andre. *Blast Injuries.* 12 19, 2007. http://emedicine.medscape.com/article/822587-media (accessed 06 12, 2009).

Pidaparti, R.M., and A. Vogt. "Experimental Investigation of Poisson's Ratio as a Damage Parameter for Bone Fatigue." *Journal of Biomedical Materials Research* 59 (2002): 282-287.

Rupin, F., et al. "Determination of Poisson Ratio and Young Modulus by Scanning Acoustic Microscopy Combined with Nanoindentation." *Journal of Clinical Densitometry* 12, no. 1: 124-125.

Sarkar, S., S. Majumder, and A. Roychowdhury. "Response of the Human Head Under Static and Dynamic Load Using Finite Element Method." *Trends Biomater. Artif. Organs* 17 (2004): 130-134.

Schröder, Christoph T., and Waymond R. Jr. Scott. "A Finite-Difference Model to Study the Elastic-Wave Interactions with Buried Land Mines." *IEEE Transactions on Geoscience and Remote Sensing*, July 2000: 1505-1512.

Schroder, Christoph T., and Waymond R. Scott Jr. "A Finite-Difference Model to Study the Elastic-Wave Interacations with Buried Land Mines." *IEEE Transactions on Geoscience and Remote Sensing* 38, no. 4 (July 2000): 1505-1512.

Slotnick, Jeff. *Explosive Forces of Improvised Explosive Devices.* 2004. http://www.securitydriver.com/aic/stories/article-114.html (accessed August 2008).

Sullivan, John P., et al. *Jane's Unconventional Weapons Response Handbook.* Alexandria, VA: Jane's Information Group, 2002.

Taber, Katherine H., and Robin A. Hurley. "Traumatic Axonal Injury: Atlas of Major Pathways." *The Journal of Neuropsychiatry and Clinical Neurosciences* 19 (2007): iv-104.

Wildegger-Gassmaier, Anne E. "Aspects of Thermobaric Weaponry." *ADF Health* 4 (April 2003): 3-6.

Yin, Xiangtao, and Hynynen Kullervo. "A Numerical Study of Pulsed Sonication For Reducing Thermal Deposition in The Skull During Transcranial Focused Ultrasound Surgery." *2005 IEEE Ultrasonics Symposiam.* IEEE, 2005. 1241-1244.

Zhang, Liying, King H. Yang, and Albert I. King. "A Proposed Injury Threshold for Mild Traumatic Brain Injury." *Transactions of the ASME*, April 2004: 226-236.

Appendix: Computer Code

MATLAB: Explosive Blast Waveform

The following code incorporates the ideal blast wave equations from Kinney and Graham (equations 1.1 through 1.8), in order to graphically depict pressure waveforms and quickly determine appropriate waveforms for any size of explosive and distance from the point of detonation.

```
%Blast Wave Equations
%reference: Kinney and Graham
%           Explosive Shocks in Air
close all; clear all; clc

dvals=linspace(10,20,50);

for m=1:size(dvals,2)

d0=dvals(2);
rho0=1;  %density of atmosphere for reference explosion
rho=1;    %density of atmosphere shock traves through
p=1;    %ambient air pressure in atm?
W=10;    %weight of explosive (TNT) in kg
d=dvals(m);  %distance in meters from explosion
k=1.4; %ratio of heat capacities (cp/cv) for air
Ztable=[.952 4 .461;
    1 3.71 .506;
    1.1 3.24 .606;
    1.25 2.68 .774;
    1.34 2.4 .897;
    1.6 1.86 1.242;
    1.75 1.63 1.474;
    1.9 1.44 1.723;
    2.0 1.29 1.897;
    2.15 1.21 2.171;
    2.25 1.14 2.361;
    2.4 1.04 2.656;
    2.6 .94 3.069;
    2.9 0.82 3.72;
    3.1 0.76 4.17;
    3.5 0.67 5.11;
    4 .6 6.34;
    4.75 0.52 8.27;
    5.5 .47 10.25;
    6.25 0.44 12.28;
    6.75 .42 13.64;
    8 0.38 17.1;
    9 0.36 19.9;
    11 .33 25.5;
    13 .30 31.2;
    16 .28 39.8;
    20 .25 51.4;
    27.5 .22 73.1;
    32.5 .21 87.6;
    40 .2 109;
    50 .18 139;
    70 .17 197;
    80 .16 226;
```

```
    100 .15 285;
    125 .14 358;
    150 .14 431;
    200 .13 578;
    300 .13 871;
    400 .12 1165;
    500 .12 1459];

fd=(rho/rho0)^(1/3);   %equation 7-3 p108

%scaled distance (sd)
Z=fd*d/(W^(1/3)); %eq 7-4 p109

%overpressure
pover=p*(808*(1+(Z/4.5)^2))/((sqrt(1+(Z/0.048)^2))*...
    (sqrt(1+(Z/0.32)^2))*(sqrt(1+(Z/1.35)^2))); %eq 6-2 p 94

%Mach number
Mx=sqrt(1+((k+1)/(2*k))*(pover/p));  %eq 4-26 p56

%duration of overpressure
td=W^(1/3)*(980*(1+(Z/0.54)^10))/((1+(Z/0.02)^3)*...
    (1+(Z/0.74)^6)*(sqrt(1+(Z/6.9)^2)));     %eq 6-10 p97

%duration of negative phase
tn=3*td;   %Jeff Slotnick article

%arrival time = ta
index=find(Ztable>=Z,1);
tarange=Ztable(index-1,3)-Ztable(index,3);
Zrange=Ztable(index,1)-Ztable(index-1,1);
dZ=(Ztable(index,1)-Z)/Zrange;
ta=Ztable(index,3)+dZ*tarange;

%wave form parameter
alpharange=Ztable(index-1,2)-Ztable(index,2);
Zrange=Ztable(index,1)-Ztable(index-1,1);
dZ=(Ztable(index,1)-Z)/Zrange;
alpha=Ztable(index,2)+dZ*alpharange;

%Thickness of shock front
th=10^-8*(11+7*Mx)/(rho*(Mx-1));

%plot wave form
bp=0;
t=linspace(0,td,100);
dt=t(2)-t(1);
%bp = blast pressure
for n=1:100
    bp(n)=pover*(1-t(n)/td)*exp(-(alpha*t(n)/td))+p;  %6-13 p100
    t(n)=t(n)+ta;
end

t=[0 ta-dt t];
bp=[p p bp];

Zdata(m)=Z;
OverpressData(m)=pover;
MachData(m)=Mx;
PressureData(m,:)=bp;
```

```
TimeData(m,:)=t;
thdata(m)=th;
tddata(m)=td;
tadata(m)=ta;
t3d(m,:)=tadata(m)*ones(1,size(dvals,2));

end

fh=figure(2);
pplot=p*ones(1,size(dvals,2));
pplot(1)=PressureData(1,3);
pplotdata(1,:)=pplot;
figure(2)
plot(dvals,pplot)
title(['t=',num2str(TimeData(1,3)),'msec'])
xlabel('distance')
ylabel('pressure')
axis([min(dvals) max(dvals) 0 max(PressureData(:,3))])


t1=min(TimeData(:,3));
tfinal=min(TimeData(:,102));

for j=2:size(dvals,2)
    time=TimeData(j,3);
    thickness=thdata(j);
    pplot(j)=PressureData(j,3);
    for k=1:j-1
        index2=find(TimeData(k,:)>time,1);
        index2=index2-1;
        if index2>0
            pplot(k)=PressureData(k,index2);
        else
            pplot(k)=p;
        end
    end
    pplotdata(j,:)=pplot;
    plot(dvals,pplot)
axis([min(dvals) max(dvals) 0 max(PressureData(:,3))])
    title(['Positive Phase Propagation, t=',num2str(time),'msec; th=',
        num2str(thickness*1000),'mm'])
    ylabel('pressure: ATM')
    xlabel('distance: meters')
    pause(.1)

end
hold on
figure(3)
dvals3d=ones(size(dvals,2),1)*dvals;
surf(dvals3d,t3d,pplotdata)
shading interp
xlabel('distance, meters')
ylabel('time, msec')
zlabel('pressure, ATM')

figure(4)
hold on
plot(dvals,MachData,'r')
plot(dvals,OverpressData,'b')
plot(dvals,tddata,'g')
```

```
%plot(dvals,tadata,'m')
ylabel('Shock Speed: Mach; Overpressure: ATM; Pos. Duration: msec')
legend('Shock Speed', 'Max Overpressure','Overpressure Duration')
title('Shock wave data')
xlabel('distance from source: meters')
%mov=close(mov);

OverpressData=OverpressData*101.3;
figure(5)
loglog(tddata,OverpressData)
axis([.1 1000 10 10000])
```

MATLAB: Cell Averaging

This code is implemented to illustrate the differences between methods of cell averaging, equations 4.5.1 through 4.5.7.

```
%Cell averaging Code
%Randall Hoberecht

clear all; close all; clc

%Material Properties
%AIR
rho1=0.0000012;
lam1=1420;
mu1=0;

%BRAIN
%rho1=0.00104;
%lam1=21900000;
%mu1=225.3;

bulk1=lam1+2*mu1;
cp1=sqrt(bulk1/rho1);
cs1=sqrt(mu1/rho1);
im1=sqrt(rho1*lam1);

%BONE
rho2=0.0019;
lam2=17700000;
mu2=26600000;
bulk2=lam2+2*mu2;
cp2=sqrt(bulk2/rho2);
cs2=sqrt(mu2/rho2);
im2=sqrt(rho2*lam2);

p=.5;
spaces=100;
x=linspace(0,1,spaces+1);

for i=0:spaces
    w1=(spaces-i)/spaces;
    w2=1-w1;
```

```
    rho(i+1)=w1*rho1+w2*rho2;
    rhoh(i+1)=1/((spaces-i)/(spaces*rho1)+(i/(spaces*rho2)));
    lam(i+1)=(spaces-i)*lam1/spaces+i*lam2/spaces;
    lamh(i+1)=1/((spaces-i)/(spaces*lam1)+(i/(spaces*lam2)));
    mu(i+1)=(spaces-i)*mu1/spaces+i*mu2/spaces;
    muh(i+1)=1/((spaces-i)/(spaces*mu1)+(i/(spaces*mu2)));
    if i==0
        muh(i+1)=mu1;
    end
    if i==spaces
        muh(i+1)=mu2;
    end
    bulkh1(i+1)=1/((spaces-i)/(spaces*bulk1)+(i/(spaces*bulk2)));
    bulka(i+1)=(spaces-i)*bulk1/spaces+i*bulk2/spaces;
    cph(i+1)=1/((spaces-i)/(spaces*cp1)+(i/(spaces*cp2)));
    cs(i+1)=sqrt(mu(i+1)/rho(i+1));
    csh(i+1)=sqrt(muh(i+1)/rho(i+1));
    cscp(i+1)=1/((spaces-i)/(spaces*cs1)+(i/(spaces*cs2)));
    if w1>p
    lamcp(i+1)=lam1*cph(i+1)/cp1;
    mucp(i+1)=.5*((bulk1*cph(i+1))/cp1-lamcp(i+1));
      if mucp(i+1)<0
          mucp(i+1)=0;
      end
    else
    lamcp(i+1)=lam2*cph(i+1)/cp2;
    mucp(i+1)=.5*((bulk2*cph(i+1))/cp2-lamcp(i+1));
      if mucp(i+1)<0
          mucp(i+1)=0;
      end
    end
    rhocp(i+1)=(lamcp(i+1)+2*mucp(i+1))/cph(i+1)^2;
    cscp(i+1)=sqrt(mucp(i+1)/rhocp(i+1));

end

bulk=lam+2*mu;
bulkh2=lamh+2*muh;
cp=sqrt(bulk./rho);
cs=sqrt(mu./rho);
cph1=sqrt(bulkh1./rho);
cph2=sqrt(bulkh2./rho);

figure(1)
subplot(2,6,1:2)
hold on
title('\rho')
plot(x,rho,'g')
plot(x,rhoh,'b')
plot(x,rhocp,'r')

subplot(2,6,3:4)
hold on
title('\lambda')
plot(x,lam,'g')
plot(x,lamh,'b')
plot(x,lamcp,'r')
%legend('arithmetic avg','harmonic average','harmonic c_p')

subplot(2,6,5:6)
```

```
hold on
title('\mu')
plot(x,mu,'g')
plot(x,muh,'b')
plot(x,mucp,'r')
%legend('arithmetic avg','harmonic average','harmonic c_p')

subplot(2,6,7:8)
hold on
plot(x,cp,'g')
plot(x,cph2,'b')
plot(x,cph,'r')
title('c_p')
%legend('a-mean \lambda, \mu, \rho','h-mean of c_p',...
%     'h-mean of \lambda and \mu')

subplot(2,6,9:10)
hold on
title('c_s')
plot(x,cs,'g')
plot(x,csh,'b')
plot(x,cscp,'r')
%legend('a')
legend('arithmetic avg \rho, \lambda, and \mu',...
    'arithmetic avg \rho; harmonic avg \lambda, \mu','harmonic avg c_p')
```

CLAWPACK

CLAWPACK can be obtained at www.amath.washington.edu/~claw/. This resource provides full documentation. This appendix highlights user-defined portions of subroutines used within CLAWPACK.

CLAWPACK: Riemann Solver for Elastics

The following is a subroutine incorporated into CLAWPACK to solve the Riemann problems associated with linear elastics. This code was obtained through examples present in existing CLAWPACK example problems. This code applies equations 4.2.8 through 4.2.13.

```
c     =======================================================
      subroutine rpn2(ixy,maxm,meqn,mwaves,mbc,mx,ql,qr,
     &                auxl,auxr,wave,s,amdq,apdq)
c     =======================================================
c
c     # Riemann solver for the elasticity equations in 2d, with varying
c     # material properties rho, lambda, and mu
c
c     # Note that although there are 5 eigenvectors, one eigenvalue
c     # is always zero and so we only need to compute 4 waves.
c     #
c     # solve Riemann problems along one slice of data.
c
```

```
c       # On input, ql contains the state vector at the left edge of each cell
c       #         qr contains the state vector at the right edge of each cell
c
c       # Note that the i'th Riemann problem has left state qr(i-1,:)
c       #                                    and right state ql(i,:)
c       # From the basic clawpack routines, this routine is called with ql = qr
c
c       # This data is along a slice in the x-direction if ixy=1
c       #                               or the y-direction if ixy=2.
c
c       # Contents of ql and qr:
c       #
c       # q(:,1) = sigma^{11} if ixy=1   or   sigma^{22} if ixy=2
c       # q(:,2) = sigma^{22} if ixy=1   or   sigma^{11} if ixy=2
c       # q(:,3) = sigma^{12} = sigma^{21}
c       # q(:,4) = u          if ixy=1   or   v          if ixy=2
c       # q(:,5) = v          if ixy=1   or   u          if ixy=2
c       #
c       # auxl and auxr hold corresponding slice of the aux array:
c       # Here it is assumed that auxl=auxr gives the cell values
c       # for this slice.
c       #
c       #  auxl(i,1) = rho, density
c       #  auxl(i,2) = lambda
c       #  auxl(i,3) = mu
c       #  auxl(i,4) = cp, P-wave speed
c       #  auxl(i,5) = cs, S-wave speed
c
c
c       # On output, wave contains the waves,
c       #            s the speeds,
c       #            amdq the  left-going flux difference  A^- \Delta q
c       #            apdq the right-going flux difference  A^+ \Delta q
c
c       # Note that the waves are *not* in order of increasing lambda.
c       # Instead the 1- and 2-waves are the P-waves and the 3- and 4-waves
c       # are the S-waves.   (The 5th wave has speed zero and is not used.)
c
c
c
        implicit double precision (a-h,o-z)
c
        dimension wave(1-mbc:maxm+mbc, meqn, mwaves)
        dimension awave(1-mbc:maxm+mbc,meqn)
        dimension    s(1-mbc:maxm+mbc, mwaves)
        dimension   ql(1-mbc:maxm+mbc, meqn)
        dimension   qr(1-mbc:maxm+mbc, meqn)
        dimension apdq(1-mbc:maxm+mbc, meqn)
        dimension amdq(1-mbc:maxm+mbc, meqn)
        dimension auxl(1-mbc:maxm+mbc, 5)
        dimension auxr(1-mbc:maxm+mbc, 5)
c
c
c       # set ku to point to  the component of the system that corresponds
c       # to velocity in the direction of this slice, kv to the orthogonal
c       # velocity.  Similarly ksig11 and ksig22 point to normal stresses.
c       # 3rd component is always shear stress sig12.
c
c
        if (ixy.eq.1) then
```

```
         ksig11 = 1
         ksig22 = 2
         ku = 4
         kv = 5
       else
         ksig11 = 2
         ksig22 = 1
         ku = 5
         kv = 4
       endif
c
c      # note that notation for u and v reflects assumption that the
c      # Riemann problems are in the x-direction with u in the normal
c      # direciton and v in the orthogonal direcion, but with the above
c      # definitions of ku and kv the routine also works with ixy=2
c
c
c      # split the jump in q at each interface into waves
c      # The jump is split into leftgoing waves traveling at speeds -cp, -cs
c      # relative to the material properties to the left of the interface,
c      # and rightgoing waves traveling at speeds +cp, +cs
c      # relative to the material properties to the right of the interface,
c
       do 20 i = 2-mbc, mx+mbc
          dsig11 = ql(i,ksig11) - qr(i-1,ksig11)
          dsig22 = ql(i,ksig22) - qr(i-1,ksig22)
          dsig12 = ql(i,3) - qr(i-1,3)
          du = ql(i,ku) - qr(i-1,ku)
          dv = ql(i,kv) - qr(i-1,kv)

c         # material properties in cells i (on right) and i-1 (on left):

         alamr = auxl(i,2)
          amur = auxl(i,3)
         bulkr = alamr + 2.d0*amur
         cpr = auxl(i,4)
         csr = auxl(i,5)

         alaml = auxr(i-1,2)
          amul = auxr(i-1,3)
         bulkl = alaml + 2.d0*amul
         cpl = auxr(i-1,4)
         csl = auxr(i-1,5)

c         # P-wave strengths:
c
         det = bulkl*cpr + bulkr*cpl
         if (det.eq.0.d0) then
            write(6,*) 'det=0 in rpn2'
            stop
            endif
          a1 = (cpr*dsig11 + bulkr*du) / det
          a2 = (cpl*dsig11 - bulkl*du) / det

c         # S-wave strengths:
c
         det = amul*csr + amur*csl
         if (det.eq.0.d0) then
c            # no s-waves
            a3 = 0.d0
```

```
              a4 = 0.d0
             else
               a3 = (csr*dsig12 + amur*dv) / det
               a4 = (csl*dsig12 - amul*dv) / det
             endif
c
c             a5 =dsig22-(alaml*a1)-(alamr*a2)
            a5 = -dsig11*(alaml*cpr+alamr*cpl)/(cpl*bulkr+bulkl*cpr)
     &            +dsig22
     &            +du*(bulkl*alamr-bulkr*alaml)/(cpl*bulkr+bulkl*cpr)
c             write(6,*)a5,ixy,i
c
c          # 5th wave has velocity 0 so is not computed or propagated.
c
c
c          # Compute the waves.
c
            wave(i,ksig11,1) = a1 * bulkl
            wave(i,ksig22,1) = a1 * alaml
            wave(i,3,1)  = 0.d0
            wave(i,ku,1) = a1 * cpl
            wave(i,kv,1) = 0.d0
            s(i,1) = -cpl
c
            wave(i,ksig11,2) = a2 * bulkr
            wave(i,ksig22,2) = a2 * alamr
            wave(i,3,2)  = 0.d0
            wave(i,ku,2) = -a2 * cpr
            wave(i,kv,2) = 0.d0
            s(i,2) = cpr
c
            wave(i,ksig11,3) = 0.d0
            wave(i,ksig22,3) = 0.d0
            wave(i,3,3)  = a3*amul
            wave(i,ku,3) = 0.d0
            wave(i,kv,3) = a3*csl
            s(i,3) = -csl
c
            wave(i,ksig11,4) = 0.d0
            wave(i,ksig22,4) = 0.d0
            wave(i,3,4)  = a4*amur
            wave(i,ku,4) = 0.d0
            wave(i,kv,4) = -a4*csr
            s(i,4) = csr
c
            awave(i,ksig11)=0.d0
            awave(i,ksig22)=a5
            awave(i,3)=0.d0
            awave(i,ku)=0.d0
            awave(i,kv)=0.d0
c
   20     continue
c
c
c
c      # compute the leftgoing and rightgoing flux differences:
c      # Note s(i,1),s(i,3) < 0   and   s(i,2),s(i,4) > 0.
c
       do 220 m=1,meqn
         do 220 i = 2-mbc, mx+mbc
```

```
            amdq(i,m) = s(i,1)*wave(i,m,1) + s(i,3)*wave(i,m,3)
            apdq(i,m) = s(i,2)*wave(i,m,2) + s(i,4)*wave(i,m,4)
c  220        continue
c
c
c       TROUBLE-SHOOTING
c       do 300 m=1,5
c          wsum=0.d0
c          do 305 mw=1,4
c            wsum=wsum+wave(i,m,mw)
c 305        continue
c            wsum=wsum+awave(i,m)
c          dq=ql(i,m)-qr(i-1,m)
c          if(dabs(wsum-dq).gt.1.d-12)then
c             write(6,*)'abs(wsum-dq)=',dabs(wsum-dq),ixy,i,m,ql(i,m),
c      &qr(i-1,m),awave(i,m)
c          endif
c 300      continue


 220      continue
c
      return
      end
```

CLAWPACK: Riemann Solver for Strain Model

The following code is adapted, based on the Riemann Solver for elastics, in order to solve for the system in terms of strain (ϵ) instead of stress (σ). This code incorporates equations 4.2.24 through 4.2.26.

```
c       ========================================================
      subroutine rpn2(ixy,maxm,meqn,mwaves,mbc,mx,ql,qr,
     &              auxl,auxr,wave,s,amdq,apdq)
c       ========================================================
c
c     # Riemann solver for the elasticity equations in 2d based
c     # on formulation based on strain, with varying
c     # material properties rho, lambda, and mu
c
c     # Note that although there are 5 eigenvectors, one eigenvalue
c     # is always zero and so we only need to compute 4 waves.
c     #
c     # solve Riemann problems along one slice of data.
c
c     # On input, ql contains the state vector at the left edge of each cell
c     #           qr contains the state vector at the right edge of each cell
c
c     # Note that the i'th Riemann problem has left state qr(i-1,:)
c     #                                    and right state ql(i,:)
c     # From the basic clawpack routines, this routine is called with ql = qr
c
c     # This data is along a slice in the x-direction if ixy=1
c     #                                 or the y-direction if ixy=2.
```

```
c
c      # Contents of ql and qr:
c      #
c      # q(:,1) = epsilon^{11} if ixy=1
c      # q(:,2) = epsilon^{22} if ixy=1
c      # q(:,3) = epsilon^{12} = sigma^{21}
c      # q(:,4) = u           if ixy=1  or   v          if ixy=2
c      # q(:,5) = v           if ixy=1  or   u          if ixy=2
c      #
c      # auxl and auxr hold corresponding slice of the aux array:
c      # Here it is assumed that auxl=auxr gives the cell values
c      # for this slice.
c      #
c      #  auxl(i,1) = rho, density
c      #  auxl(i,2) = lambda
c      #  auxl(i,3) = mu
c      #  auxl(i,4) = cp, P-wave speed
c      #  auxl(i,5) = cs, S-wave speed
c
c
c      # On output, wave contains the waves,
c      #            s the speeds,
c      #            amdq the  left-going flux difference  A^- \Delta q
c      #            apdq the right-going flux difference  A^+ \Delta q
c
c      # Note that the waves are *not* in order of increasing lambda.
c      # Instead the 1- and 2-waves are the P-waves and the 3- and 4-waves
c      # are the S-waves.   (The 5th wave has speed zero and is not used.)
c
c
c
       implicit double precision (a-h,o-z)
c
       dimension wave(1-mbc:maxm+mbc, meqn, mwaves)
       dimension    s(1-mbc:maxm+mbc, mwaves)
       dimension   ql(1-mbc:maxm+mbc, meqn)
       dimension   qr(1-mbc:maxm+mbc, meqn)
       dimension apdq(1-mbc:maxm+mbc, meqn)
       dimension amdq(1-mbc:maxm+mbc, meqn)
       dimension auxl(1-mbc:maxm+mbc, 5)
       dimension auxr(1-mbc:maxm+mbc, 5)
c
c      # set ku to point to  the component of the system that corresponds
c      # to velocity in the direction of this slice, kv to the orthogonal
c      # velocity.  Similarly ksig11 and ksig22 point to normal stresses.
c      # 3rd component is always shear stress sig12.
c
c
       if (ixy.eq.1) then
          keps11 = 1
          keps22 = 2
          ku = 4
          kv = 5
        else
          keps11 = 2
          keps22 = 1
          ku = 5
          kv = 4
        endif
c
```

```
c      # note that notation for u and v reflects assumption that the
c      # Riemann problems are in the x-direction with u in the normal
c      # direciton and v in the orthogonal direcion, but with the above
c      # definitions of ku and kv the routine also works with ixy=2
c
c
c      # split the jump in q at each interface into waves
c      # The jump is split into leftgoing waves traveling at speeds -cp, -cs
c      # relative to the material properties to the left of the interface,
c      # and rightgoing waves traveling at speeds +cp, +cs
c      # relative to the material properties to the right of the interface,
c
       do 20 i = 2-mbc, mx+mbc
         deps11 = ql(i,keps11) - qr(i-1,keps11)
         deps22 = ql(i,keps22) - qr(i-1,keps22)
         deps12 = ql(i,3) - qr(i-1,3)
         du = ql(i,ku) - qr(i-1,ku)
         dv = ql(i,kv) - qr(i-1,kv)

c         # material properties in cells i (on right) and i-1 (on left):

         alamr = auxl(i,2)
          amur = auxl(i,3)
         bulkr = alamr + 2.d0*amur
         cpr = auxl(i,4)
         csr = auxl(i,5)

         alaml = auxr(i-1,2)
          amul = auxr(i-1,3)
         bulkl = alaml + 2.d0*amul
         cpl = auxr(i-1,4)
         csl = auxr(i-1,5)

c         # P-wave strengths:
c
         det1 = (bulkl*cpr+bulkr*cpl)/cpr
          det2 = (bulkl*cpr+bulkr*cpl)/cpl

        a1=(bulkr*ql(i,keps11)+bulkr*du/cpr+
     &      alamr*ql(i,keps22)-bulkl*qr(i-1,keps11)-
     &      alaml*qr(i-1,keps22))/det1
        a2=(bulkr*ql(i,keps11)-bulkl*du/cpl+
     &      alamr*ql(i,keps22)-bulkl*qr(i-1,keps11)-
     &      alaml*qr(i-1,keps22))/det2

c         # S-wave strengths:
c
         det = csr+csl
         if (det.lt.1.d-13)then
c            NO SHEAR WAVES
            a3 = 0.d0
            a4 = 0.d0
         else
            a3 = (2.d0*csr*deps12+dv)/det
            a4 = (2.d0*csl*deps12-dv)/det
         endif
c
c         # 5th wave has velocity 0 so is not computed or propagated.
c
c         # Compute the waves.
```

```
c
         wave(i,keps11,1) = a1
         wave(i,keps22,1) = 0.d0
         wave(i,3,1)  = 0.d0
         wave(i,ku,1) = a1 * cpl
         wave(i,kv,1) = 0.d0
         s(i,1) = -cpl
c
         wave(i,keps11,2) = a2
         wave(i,keps22,2) = 0.d0
         wave(i,3,2)  = 0.d0
         wave(i,ku,2) = -a2 * cpr
         wave(i,kv,2) = 0.d0
         s(i,2) = cpr
c
         wave(i,keps11,3) = 0.d0
         wave(i,keps22,3) = 0.d0
         wave(i,3,3)  = a3/2.d0
         wave(i,ku,3) = 0.d0
         wave(i,kv,3) = a3 * csl
         s(i,3) = -csl
c
         wave(i,keps11,4) = 0.d0
         wave(i,keps22,4) = 0.d0
         wave(i,3,4)  = a4/2.d0
         wave(i,ku,4) = 0.d0
         wave(i,kv,4) = -a4 * csr
         s(i,4) = csr
c
   20    continue
c
c     # compute the leftgoing and rightgoing flux differences:
c     # Note s(i,1),s(i,3) < 0   and   s(i,2),s(i,4) > 0.
c
      do 220 m=1,meqn
        do 220 i = 2-mbc, mx+mbc
           amdq(i,m) = s(i,1)*wave(i,m,1) + s(i,3)*wave(i,m,3)
           apdq(i,m) = s(i,2)*wave(i,m,2) + s(i,4)*wave(i,m,4)
  220       continue
c
      return
      end
```

CLAWPACK: Riemann Solver on non-uniform grid

The following code implements a Riemann Solver for a logically rectangular non-uniform coordinate system incorporating equations 4.5.13 through 4.5.15. This code was provided by LeVeque, as an existing CLAWPACK subroutine.

```
c     =======================================================
      subroutine rpn2(ixy,maxm,meqn,mwaves,mbc,mx,ql,qr,
     &            auxl,auxr,wave,s,amdq,apdq)
c     =======================================================
c
c     # Riemann solver for the elasticity equations in 2d, with varying
```

```
c       # material properties rho, lambda, and mu
c
c       # On general quadrilateral grid, with variable coefficients
c
c       # Rotate velocity and stresses and then call standard Riemann solver.
c       # The resulting waves and flux differences are then rotated
c       # back to x-y.
c
c       # Note that although there are 5 eigenvectors, one eigenvalue
c       # is always zero and so we only need to compute 4 waves.
c       #
c       # solve Riemann problems along one slice of data.
c
c       # On input, ql contains the state vector at the left edge of each cell
c       #           qr contains the state vector at the right edge of each cell
c
c       # Note that the i'th Riemann problem has left state qr(i-1,:)
c       #                                     and right state ql(i,:)
c       # From the basic clawpack routines, this routine is called with ql = qr
c
c       # This data is along a slice in the x-direction if ixy=1
c       #                                 or the y-direction if ixy=2.
c
c       # Contents of ql and qr:
c       #
c       # q(:,1) = sigma^{11}
c       # q(:,2) = sigma^{22}
c       # q(:,3) = sigma^{12} = sigma^{21}
c       # q(:,4) = u
c       # q(:,5) = v
c       #
c       # auxl and auxr hold corresponding slice of the aux array:
c       # Here it is assumed that auxl=auxr gives the cell values
c       # for this slice.
c       #
c       #    aux(i,j,1)  = ax
c       #    aux(i,j,2)  = ay   where (ax,ay) is unit normal to left face
c       #    aux(i,j,3)  = ratio of length of left face to dyc
c
c       #    aux(i,j,4)  = bx
c       #    aux(i,j,5)  = by   where (bx,by) is unit normal to bottom face
c       #    aux(i,j,6)  = ratio of length of bottom face to dxc
c
c       #    aux(i,j,7)  = ratio of cell area to dxc*dyc
c       #                     (approximately Jacobian of mapping function)
c
c       #    auxl(i,8) = rho, density
c       #    auxl(i,9) = lambda
c       #    auxl(i,10) = mu
c
c
c       # On output, wave contains the waves,
c       #             s the speeds,
c       #             amdq the  left-going flux difference  A^- \Delta q
c       #             apdq the right-going flux difference  A^+ \Delta q
c
c       # Note that the waves are *not* in order of increasing lambda.
c       # Instead the 1- and 2-waves are the P-waves and the 3- and 4-waves
c       # are the S-waves.   (The 5th wave has speed zero and is not used.)
c
```

```
c
c
      implicit double precision (a-h,o-z)
c
      dimension wave(1-mbc:maxm+mbc, meqn, mwaves)
      dimension    s(1-mbc:maxm+mbc, mwaves)
      dimension   ql(1-mbc:maxm+mbc, meqn)
      dimension   qr(1-mbc:maxm+mbc, meqn)
      dimension apdq(1-mbc:maxm+mbc, meqn)
      dimension amdq(1-mbc:maxm+mbc, meqn)
      dimension auxl(1-mbc:maxm+mbc, 12)
      dimension auxr(1-mbc:maxm+mbc, 12)

      parameter (maxm4 = 1004)  !# assumes at most 1000x1000 grid with mbc=2
      dimension unl(-3:maxm4), unr(-3:maxm4)
      dimension usl(-3:maxm4), usr(-3:maxm4)
      dimension sigmanl(-3:maxm4), sigmanr(-3:maxm4)
      dimension sigmasl(-3:maxm4), sigmasr(-3:maxm4)
      dimension alf(-3:maxm4)
      dimension beta(-3:maxm4)

      if (mx.gt.maxm4 .or. mbc.gt.4) then
         write(6,*) ' increase maxm4 in rpn2'
         stop
         endif
c
c
c     # rotate the velocities q(4) and q(5) so that it is aligned with grid
c     # normal.  Rotate the stresses to compute normal and shear stress
c     # relative to the grid normal.
c     # The normal vector for the face at the i'th Riemann problem
c     # is stored in the aux array
c     # in locations (1,2) if ixy=1 or (4,5) if ixy=2.  The ratio of the
c     # length of the cell side to the length of the computational cell
c     # is stored in aux(3) or aux(6) respectively.
c
c
      if (ixy.eq.1) then
          inx = 1
          iny = 2
          ilenrat = 3
        else
          inx = 4
          iny = 5
          ilenrat = 6
        endif
c
c       # determine rotation matrix
c             [ alf  beta ]
c             [-beta  alf ]
c
c       # note that this reduces to identity on standard cartesian grid
c
c       # determine normal and shear velocity and stress components
c       # un = normal velocity,   us = shear velocity
c       # sigman = normal stress,   sigmas = shear stress
c       # at this edge:
c
        do i=2-mbc,mx+mbc
           alf(i) = auxl(i,inx)
```

```
            beta(i) = auxl(i,iny)
            unl(i) = alf(i)*ql(i,4) + beta(i)*ql(i,5)
            unr(i-1) = alf(i)*qr(i-1,4) + beta(i)*qr(i-1,5)
            usl(i) = -beta(i)*ql(i,4) + alf(i)*ql(i,5)
            usr(i-1) = -beta(i)*qr(i-1,4) + alf(i)*qr(i-1,5)
            sigmanl(i) = alf(i)**2*ql(i,1) + beta(i)**2*ql(i,2)
     &                  + 2.d0*alf(i)*beta(i)*ql(i,3)
            sigmanr(i-1) = alf(i)**2*qr(i-1,1) + beta(i)**2*qr(i-1,2)
     &                  + 2.d0*alf(i)*beta(i)*qr(i-1,3)
            sigmasl(i) = alf(i)*beta(i) * (-ql(i,1) + ql(i,2))
     &                  + (alf(i)**2 - beta(i)**2)*ql(i,3)
            sigmasr(i-1) = alf(i)*beta(i) * (-qr(i-1,1) + qr(i-1,2))
     &                  + (alf(i)**2 - beta(i)**2)*qr(i-1,3)
            enddo
c
c
c     # split the jump in q at each interface into waves
c     # The jump is split into leftgoing waves traveling at speeds -cp, -cs
c     # relative to the material properties to the left of the interface,
c     # and rightgoing waves traveling at speeds +cp, +cs
c     # relative to the material properties to the right of the interface,
c
      do 20 i = 2-mbc, mx+mbc
         dsig11 = sigmanl(i) - sigmanr(i-1)
         dsig12 = sigmasl(i) - sigmasr(i-1)
         du = unl(i) - unr(i-1)
         dv = usl(i) - usr(i-1)

c        # material properties in cells i (on right) and i-1 (on left):

         rhor = auxl(i,8)
         alamr = auxl(i,9)
          amur = auxl(i,10)
         bulkr = alamr + 2.d0*amur
          cpr = dsqrt(bulkr/rhor)
         csr = dsqrt(amur/rhor)

         rhol = auxl(i-1,8)
         alaml = auxr(i-1,9)
          amul = auxr(i-1,10)
         bulkl = alaml + 2.d0*amul
          cpl = dsqrt(bulkl/rhol)
         csl = dsqrt(amul/rhol)

c        # P-wave strengths:
c
         det = bulkl*cpr + bulkr*cpl
         if (det.eq.0.d0) then
            write(6,*) 'det=0 in rpn2'
            stop
            endif
         a1 = (cpr*dsig11 + bulkr*du) / det
         a2 = (cpl*dsig11 - bulkl*du) / det

c        # S-wave strengths:
c
         det = amul*csr + amur*csl
         if (det.eq.0.d0) then
c            # no s-waves
            a3 = 0.d0
```

```
               a4 = 0.d0
             else
               a3 = (csr*dsig12 + amur*dv) / det
               a4 = (csl*dsig12 - amul*dv) / det
             endif
c
c          # 5th wave has velocity 0 so is not computed or propagated.
c
c
c          # Compute the waves and transform back to x-y coordinates:
c
           wave(i,1,1) = a1 * (bulkl*alf(i)**2 + alaml*beta(i)**2)
           wave(i,2,1) = a1 * (bulkl*beta(i)**2 + alaml*alf(i)**2)
           wave(i,3,1) = a1 * alf(i)*beta(i)*(bulkl - alaml)
           wave(i,4,1) = a1 * cpl * alf(i)
           wave(i,5,1) = a1 * cpl * beta(i)
           s(i,1) = -cpl * auxl(i,ilenrat)
c          /auxl(i,7)
c
           wave(i,1,2) =  a2 * (bulkr*alf(i)**2 + alamr*beta(i)**2)
           wave(i,2,2) =  a2 * (bulkr*beta(i)**2 + alamr*alf(i)**2)
           wave(i,3,2) =  a2 * alf(i)*beta(i)*(bulkr - alamr)
           wave(i,4,2) = -a2 * cpr * alf(i)
           wave(i,5,2) = -a2 * cpr * beta(i)
           s(i,2) = cpr * auxl(i,ilenrat)
c           /auxr(i-1,7)
c
           wave(i,1,3) = -a3 * amul * 2.d0*alf(i)*beta(i)
           wave(i,2,3) =  a3 * amul * 2.d0*alf(i)*beta(i)
           wave(i,3,3) =  a3 * amul * (alf(i)**2 - beta(i)**2)
           wave(i,4,3) = -a3 * csl * beta(i)
           wave(i,5,3) =  a3 * csl * alf(i)
           s(i,3) = -csl * auxl(i,ilenrat)
c          /auxl(i,7)
c
           wave(i,1,4) = -a4 * amur * 2.d0*alf(i)*beta(i)
           wave(i,2,4) =  a4 * amur * 2.d0*alf(i)*beta(i)
           wave(i,3,4) =  a4 * amur * (alf(i)**2 - beta(i)**2)
           wave(i,4,4) =  a4 * csr * beta(i)
           wave(i,5,4) = -a4 * csr * alf(i)
           s(i,4) = csr * auxl(i,ilenrat)
c          /auxr(i-1,7)
c
   20     continue
c
c
c
c     # compute the leftgoing and rightgoing flux differences:
c     # Note s(i,1),s(i,3) < 0   and   s(i,2),s(i,4) > 0.
c
      do 220 m=1,meqn
        do 220 i = 2-mbc, mx+mbc
            amdq(i,m) = s(i,1)*wave(i,m,1) + s(i,3)*wave(i,m,3)
            apdq(i,m) = s(i,2)*wave(i,m,2) + s(i,4)*wave(i,m,4)
  220       continue
c
      return
      end
```

CLAWPACK: Boundary Conditions for incoming wave

The following is an extract from the subroutine BC2.f implemented in CLAWPACK. User defined boundary conditions have been included for left and bottom boundaries of the domain, in order to establish a pressure profile based on equation 3.4.1. Required inputs are angle (angle of incidence for wave $0 \leq angle \leq \pi/2$), α (waveform parameter), t_d (duration of positive phase), and p_{max} (maximum overpressure).

```
c       =========================================================
        subroutine bc2(maxmx,maxmy,meqn,mbc,mx,my,xlower,ylower,
      &                dx,dy,q,maux,aux,t,dt,mthbc)
c       =========================================================
c
c
c       # Standard boundary condition choices for claw2
c       # Modified for elasticity:
c       #   mthbc(k) = 3:  velocities u,v are components 4,5 of q
c       #   mthbc(k) = 4:  new choice for no stress boundaries
c
c       # At each boundary  k = 1 (left),  2 (right),  3 (top), 4 (bottom):
c       #   mthbc(k) =  0  for user-supplied BC's
c       #            =  1  for zero-order extrapolation
c       #            =  2  for periodic boundary conidtions
c       #            =  3  for fixed wall, u=v=0
c       #            =  4  for no stress, sig11=sig12=0  or  sig12=sig22=0
c       -------------------------------------------------
c
c       # Extend the data from the interior cells (1:mx, 1:my)
c       # to the ghost cells outside the region:
c       #   (i, 1-jbc)   for jbc = 1,mbc,  i = 1-mbc, mx+mbc
c       #   (i, my+jbc)  for jbc = 1,mbc,  i = 1-mbc, mx+mbc
c       #   (1-ibc, j)   for ibc = 1,mbc,  j = 1-mbc, my+mbc
c       #   (mx+ibc, j)  for ibc = 1,mbc,  j = 1-mbc, my+mbc
c
        implicit double precision (a-h,o-z)
        dimension q(1-mbc:maxmx+mbc, 1-mbc:maxmy+mbc, meqn)
        dimension aux(1-mbc:maxmx+mbc, 1-mbc:maxmy+mbc, *)
        dimension mthbc(4)
        common /wavebc/ angle,td,pmax,alpha
c
c
c-------------------------------------------------------------
c       # left boundary:
c-------------------------------------------------------------
        go to (100,110,120,130,140) mthbc(1)+1
c
  100 continue
        if(dabs(angle).lt.0.0001)then
c       damping function for t>td
        d1=dmin1(1.d0,(-t+4.d0*td)/(3.d0*td))
        d=dmax1(0.d0,d1)
        p=pmax*(1-t/td)*dexp(-alpha*t/td)*d
c
        do 106 ibc=1,mbc
           do 106 j=1-mbc, my+mbc
c              if(t.lt.td)then
              q(1-ibc,j,1)=p
              q(1-ibc,j,2)=p*aux(1-ibc,j,2)/(aux(1-ibc,j,2)+
```

```
     &2.d0*aux(1-ibc,j,3))
            q(1-ibc,j,3)=0
            q(1-ibc,j,4)=-p*aux(1-ibc,j,4)/(aux(1-ibc,j,2)+
     &2.d0*aux(1-ibc,j,3))
            q(1-ibc,j,5)=0
  106 continue
c
       else
         if(dabs(angle-3.141592/2.d0).lt.0.0001)then
           go to 110
         else
c
      anx=dcos(angle)
      any=dsin(angle)
      y0=t*aux(1,1,4)/any
      do 107 ibc=1,mbc
         do 107 j=1-mbc, my+mbc
         yc=ylower+(j-0.5d0)*dy
          if(yc.gt.y0)then
            q(1-ibc,j,1)=0
            q(1-ibc,j,2)=0
            q(1-ibc,j,3)=0
            q(1-ibc,j,4)=0
            q(1-ibc,j,5)=0
          else
            ty=(y0-yc)*any/aux(1,1,4)
            d1=dmin1(1.d0,(-ty+4.d0*td)/(3.d0*td))
            d=dmax1(0.d0,d1)
            p=pmax*(1-ty/td)*dexp(-alpha*ty/td)*d

      vm=p/(aux(1,1,2)+aux(1,1,3))
      r1=aux(1,1,2)+2.d0*aux(1,1,3)*anx**2
      r2=aux(1,1,2)+2.d0*aux(1,1,3)*any**2
      r3=2.d0*aux(1,1,3)*anx*any
      r4=-anx*aux(1,1,4)
      r5=-any*aux(1,1,4)

            q(1-ibc,j,1)=vm*r1
            q(1-ibc,j,2)=vm*r2
            q(1-ibc,j,3)=vm*r3
            q(1-ibc,j,4)=vm*r4
            q(1-ibc,j,5)=vm*r5
         endif
 107       continue

       endif
      endif

      go to 199

c
  110 continue
c     # zero-order extrapolation:
      do 115 m=1,meqn
         do 115 ibc=1,mbc
            do 115 j = 1-mbc, my+mbc
               q(1-ibc,j,m) = q(1,j,m)
  115       continue
      go to 199
```

```
  120 continue
c     # periodic:
      do 125 m=1,meqn
         do 125 ibc=1,mbc
            do 125 j = 1-mbc, my+mbc
               q(1-ibc,j,m) = q(mx+1-ibc,j,m)
  125       continue
      go to 199

  130 continue
c     # solid wall with no slip, u = v = 0 (components 4 and 5 of q)
      do 135 m=1,meqn
         do 135 ibc=1,mbc
            do 135 j = 1-mbc, my+mbc
               q(1-ibc,j,m) = q(ibc,j,m)
  135       continue
c     # negate the velocity:
      do 136 ibc=1,mbc
         do 136 j = 1-mbc, my+mbc
            q(1-ibc,j,4) = -q(ibc,j,4)
            q(1-ibc,j,5) = -q(ibc,j,5)
  136    continue
      go to 199

  140 continue
c     # no-stress boundary conditions sig12 = sig11 = 0
      do 145 m=1,meqn
         do 145 ibc=1,mbc
            do 145 j = 1-mbc, my+mbc
               q(1-ibc,j,m) = q(ibc,j,m)
  145       continue
c     # negate the sig12 and sig11 components in ghost cells:
      do 146 ibc=1,mbc
         do 146 j = 1-mbc, my+mbc
            q(1-ibc,j,1) = -q(ibc,j,1)
            q(1-ibc,j,3) = -q(ibc,j,3)
  146    continue
      go to 199

  199 continue
c
c-------------------------------------------------------
c     # right boundary:
c-------------------------------------------------------

CODE OMITTED (no user defined boundary conditions on right side)

c-------------------------------------------------------
c     # bottom boundary:
c-------------------------------------------------------
      go to (300,310,320,330,340) mthbc(3)+1
c
  300 continue
      if(dabs(angle).lt.0.00001)then
      go to 310
      else
         if(dabs(angle-3.141592/2.d0).lt.0.00001)then
c     damping function for t>td
      d1=dmin1(1.d0,(-t+4.d0*td)/(3.d0*td))
      d=dmax1(0.d0,d1)
```

```
      p=pmax*(1-t/td)*dexp(-alpha*t/td)*d
c
      do 306 jbc=1,mbc
         do 306 i=1-mbc, mx+mbc
            q(i,1-jbc,1)=p*aux(i,1-jbc,2)/(aux(i,1-jbc,2)+
     &2.d0*aux(i,1-jbc,3))
            q(i,1-jbc,2)=p
            q(i,1-jbc,3)=0
            q(i,1-jbc,4)=0
            q(i,1-jbc,5)=-p*aux(i,1-jbc,4)/(aux(i,1-jbc,2)+
     &2.d0*aux(i,1-jbc,3))
  306 continue

         else

      anx=dcos(angle)
      any=dsin(angle)
      x0=t*aux(1,1,4)/anx
      do 307 jbc=1,mbc
         do 307 i=1-mbc, mx+mbc
         xc=xlower+(i-0.5d0)*dx
          if(xc.gt.x0)then
            q(i,1-jbc,1)=0
            q(i,1-jbc,2)=0
            q(i,1-jbc,3)=0
            q(i,1-jbc,4)=0
            q(i,1-jbc,5)=0
         else
            tx=(x0-xc)*anx/aux(1,1,4)
            d1=dmin1(1.d0,(-tx+4.d0*td)/(3.d0*td))
            d=dmax1(0.d0,d1)
            p=pmax*(1-tx/td)*dexp(-alpha*tx/td)*d

      vm=p/(aux(1,1,2)+aux(1,1,3))
      r1=aux(1,1,2)+2.d0*aux(1,1,3)*anx**2
      r2=aux(1,1,2)+2.d0*aux(1,1,3)*any**2
      r3=2.d0*aux(1,1,3)*anx*any
      r4=-anx*aux(1,1,4)
      r5=-any*aux(1,1,4)

            q(i,1-jbc,1)=vm*r1
            q(i,1-jbc,2)=vm*r2
            q(i,1-jbc,3)=vm*r3
            q(i,1-jbc,4)=vm*r4
            q(i,1-jbc,5)=vm*r5
         endif
  307       continue

         endif
      endif

      go to 399
c
  310 continue
c     # zero-order extrapolation:
      do 315 m=1,meqn
         do 315 jbc=1,mbc
            do 315 i = 1-mbc, mx+mbc
               q(i,1-jbc,m) = q(i,1,m)
  315       continue
```

```
      go to 399

  320 continue
c     # periodic:
      do 325 m=1,meqn
         do 325 jbc=1,mbc
            do 325 i = 1-mbc, mx+mbc
               q(i,1-jbc,m) = q(i,my+1-jbc,m)
  325       continue
      go to 399

  330 continue
c     # solid wall with no slip, u = v = 0 (components 4 and 5 of q)
      do 335 m=1,meqn
         do 335 jbc=1,mbc
            do 335 i = 1-mbc, mx+mbc
               q(i,1-jbc,m) = q(i,jbc,m)
  335       continue
c     # negate the velocity:
      do 336 jbc=1,mbc
         do 336 i = 1-mbc, mx+mbc
            q(i,1-jbc,4) = -q(i,jbc,4)
            q(i,1-jbc,5) = -q(i,jbc,5)
  336    continue
      go to 399

  340 continue
c     # no-stress boundary conditions sig12 = sig22 = 0
      do 345 m=1,meqn
         do 345 jbc=1,mbc
            do 345 i = 1-mbc, mx+mbc
               q(i,1-jbc,m) = q(i,jbc,m)
  345       continue
c     # negate the sig12 and sig22 components in ghost cells:
      do 346 jbc=1,mbc
         do 346 i = 1-mbc, mx+mbc
            q(i,1-jbc,2) = -q(i,jbc,2)
            q(i,1-jbc,3) = -q(i,jbc,3)
  346    continue
      go to 399

  399 continue
c
c-------------------------------------------------------
c     # top boundary:
c-------------------------------------------------------

CODE OMITTED (no user-defined boundary conditions on top boundary)


      return
      end
```

CLAWPACK: Grid Mapping

The following code maps a Cartesian grid coordinate to a logically rectangular coordinate system described by equations 4.5.8 through 4.5.12.

```fortran
c      ====================================================
       subroutine mapc2p(xc,yc,xp,yp)
c      ====================================================
c
c      # on input,  (xc,yc) is a computational grid point
c      # on output, (xp,yp) is corresponding point in physical space
c
       implicit double precision (a-h,o-z)
       common / comaux / rout
c       dimension r(1),d(1),w(1), r1(1), r2(1), r2factor(1)
c
c      # radial coordinates, xc = r,  yc = theta
c
        xp=xc
        yp=yc

       r1=5
       r2=10
       r2factor=r2/r1

       xc = xc/r1
       yc = yc/r1
       r = sqrt(xc**2+yc**2)
       d = max(dabs(xc),dabs(yc))
       r = max(r,1d-10)

       if (d.le.1) then
          xp = d*xc/r
          yp = d*yc/r
        endif

       if (and(d.gt.1,d.lt.r2factor)) then
          xp = d*xc/r
          yp = d*yc/r
          w = ((r2factor-d)/(r2factor-1.0d0))**2.d0
          xp = (w*xp)+((1.d0-w)*xc)
          yp = (w*yp)+((1.d0-w)*yc)
         endif

       if (d.ge.r2factor) then
          xp=xc
          yp=yc
         endif

       xc=xc*r1
       yc=yc*r1
       xp=xp*r1
       yp=yp*r1

       return
       end
```