Multiscale and multiphysics computational models of processes in shock wave lithotripsy

Daniel Fovargue

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mathematics.

Chapel Hill
2013

Approved by

Sorin Mitran

John Dolbow

Richard McLaughlin

Laura Miller

Pei Zhong

Abstract

DANIEL FOVARGUE: Multiscale and multiphysics computational models of processes in shock wave lithotripsy
(Under the direction of Sorin Mitran)

This thesis presents two computational models applied to processes in shock wave lithotripsy. The first is a multiphysics model of the focusing of an acoustic pulse and the subsequent shock wave formation that occurs in a refracting electromagnetic lithotripter. This model solves both the linear elasticity equations and the Euler equations with a Tait equation of state in arbitrary subsets of the full computational domain. It is implemented within BEARCLAW and uses a finite-volume Riemann solver approach. The model is validated using a standard lens design and is shown to accurately predict the effects of a lens modification. This model is also extended to include a kidney stone simulant in the domain in which a simple isotropic damage law is included. The second computational model is a 3D multiscale fracture model which predicts crack formation and propagation within a kidney stone simulant by utilizing a continuum-mesoscopic interaction. The simulant included in the model is realistic in that the data representing the stone is drawn from μCT image data. At the continuum scale the linear elasticity equations are solved while incorporating an anisotropic damage variable, again using a finite-volume Riemann solver within BEARCLAW. At the mesoscale, damage accumulates based on experimentally informed probability distributions and on predefined surfaces representing a granular structure. In addition to the computational models, some experimental results

are discussed. These include probability distributions of fracture properties found from μCT images of kidney stone simulants and corresponding image processing procedures.

ACKNOWLEDGMENTS

I first and most of all want to thank my wife and best friend, Lauren, for her continual support. My most valuable experience in graduate school was meeting her.

I would like to thank my advisor, Sorin Mitran, for his guidance and teaching. His explanations of mathematical and scientific concepts are unrivaled and my understanding is much broader and deeper as a result. I also very much appreciate his commitment to my success. I also want to thank my current and former committee members, John Dolbow, Richard McLaughlin, Laura Miller, Pei Zhong, and Michael Minion, for their time and effort.

I appreciate getting to work with Pei Zhong's lithotripsy group at Duke and would like to especially thank Jaclyn Lautz, Georgy Sankin, and Nathan Smith for their direct contributions to this work and their help along the way. I would also like to thank the members of Sorin Mitran's research group, Greg Herschlag, Anil Shenoy, Paula Vasquez, Christoph Kirsch, and Jennifer Young.

Thank you to the friends and officemates at UNC who helped make my time at graduate school the great experience it was, especially Emily Braley, Matthew Christian, J.D. Martindale, and Brandyn Lee, although there are many others too. Thank you as well to the many other professors, staff members, and students who make the UNC math department such an inspiring and friendly environment.

I would like to thank Grady Lemoine and Randy LeVeque for taking time to offer some help. Also, thank you to the Biomedical Research Imaging Center, especially Kevin Guley and Brandon Frederick, for the MicroCT imaging.

Thank you to all the amazing professors and teachers I've been lucky to have during all of my education. Finally, thank you to my parents, Kathy and Art, and my sister, Rachel, for their love and support, without which I couldn't have gotten this far.

This work has been supported by NIH grant 5R37DK052985-16.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

This thesis presents two computational models applied to extracorporeal shock wave lithotripsy (ESWL). ESWL is a common medical procedure used to break up kidney stones into small enough pieces for a patient to pass naturally. In an ESWL procedure, a strong acoustic pulse is generated outside the patient's body and is then focused towards the kidney stone. Depending on the type of lithotripter, the pulse begins as a shock wave or forms into one during transit due to nonlinear steepening effects. The strong compressive shock wave and trailing tensile wave that arrive at the stone cause damage to it through a variety of mechanisms. This process is repeated and eventually the stone will crack and break down into small pieces.

Both computational models use a finite-volume Riemann solver approach to numerically solve the partial differential equations (PDEs) chosen to model the physics. This method is implemented within the BEARCLAW software developed by Mitran [**86**] and based on the wave propagation algorithm of LeVeque [**69**]. Several modifications to the currently implemented method are also discussed in this thesis.

The first computational model is a 2D axisymmetric multiphysics simulation of acoustic pulse focusing and shock wave formation. An electromagnetic lithotripter, which uses a polystyrene acoustic lens for focusing, is simulated by this model. The simulation solves two sets of equations simultaneously within arbitrary subsets of the full computational

domain. The linear elasticity equations are used to model the refraction of the pulse within the lens and also the interaction of the pulse and kidney stone simulant, if a stone is included in the model. The Euler equations with a Tait equation of state are used to model the shock wave formation as the pulse transits from the lens towards the stone. This model is validated by comparing to experimental results with a standard lens design. The model is then shown to accurately predicts the effects of a lens modification. While both computational models presented in this work are applied to ESWL, they are developed in a general way in order to make them applicable to other application domains in potential future work.

The second computational model is a 3D multiscale simulation of fracture and damage within a kidney stone simulant. At the larger continuum scale, the heterogeneous linear elasticity equations are solved to model the p-wave and s-wave propagation through the stone as the shock wave hits. The shock wave input is taken directly from the multiphysics focusing model. The elasticity equations are extended to include an anisotropic damage variable to inform this scale of unresolved damage and fracture. The stone geometry in the computation is taken from μCT images of kidney stone simulants, so that realistic stone simulants are modeled.

At a smaller mesoscale, damage accumulates on predefined surfaces within each continuum cell, typically resembling a granular structure. The amount of damage is based on probability distributions found from experimental results. The continuum damage variables are updated according to the mesoscopic damage. Macroscopic fracture is modeled by allowing continuum cells to rupture based on the current stress and damage state. Results from solely the continuum model as well as the full multiscale model are compared

to experimental results. In addition, the mesoscopic structure is modified to simulate the mesoscopic structure of certain types of real kidney stones.

Experiments in which a stone was repeatedly shocked and imaged were also conducted as part of this work. Microscopic computed tomography, written as MicroCT or μCT, imaging was used, which reveals the internal structure of the stone. This process was applied to four stones and each resulted in several image sets of fracture within the stones. A 2D image processing procedure is presented which compiles probability distributions of crack lengths and widths. The volume of fracture and changes in these distributions as more shocks are applied are also analyzed.

The remainder of this introduction chapter is left for a description of some of the previously published work on topics presented in this thesis and contrasting that work with what is presented here. In Chapter 2, background on the application domain of ESWL is provided along with background information on continuum damage mechanics and fracture mechanics. In Chapter 3, finite-volume Riemann solver methods are described. Chapter 4 describes all the experimental methods used in this work and shows some purely experimental results. Chapter 5 contains a description of the multiphysics focusing model and results. Chapter 6 contains a description of the multiscale fracture model and corresponding results. Finally, a conclusion is provided in Chapter 7.

1.1. State of the Art

1.1.1. Computational Models of ESWL. Several studies exist on numerical models of focusing and shock wave formation during lithotripsy. Despite the prevalence of electromagnetic (EM) lithotipters, most works simulate focusing in either electrohydaulic (EH) or piezoelectric (PE) lithotripters and none model refracting EM lithotripters as is

done in this thesis. Coleman *et al.* solved the one dimensional Khokhlov-Zabolotskaya-Kuznetsov (KZK) equation, similar to Burgers' equation, with the HM3 geometry [**23**]. Hamilton developed a linear focusing solution on the axis of symmetry of a concave ellipsoidal mirror following the production of a spherical wave at the focus [**45**]. This model was later used by Sankin *et al.* to investigate optical breakdown as a shock wave generation mechanism [**110**]. Christopher developed a nonlinear acoustic model accounting for diffraction and attenuation and applied it to the HM3 [**21, 20**]. This model also solved Burgers' equation to account for nonlinear effects. Steiger [**117**] presented a finite difference model of a reflecting EM lithotripter and accounted for attenuation in tissue. Averkiou and Cleveland solved the 2D KZK equation to model an EH lithotripter [**8**]. Zhou and Zhong expanded on this model to investigate reflector geometry modifications [**137**]. Ginter *et. al.* [**40**] modeled a reflecting EM lithotripter by solving nonlinear acoustic equations by a 2D FDTD method.

Tanguay [**120**], in his disertation work, used a WENO method to solve the Euler equations for two phase flow in order to investigate the bubble cloud that forms due to the shock wave. Krimmel, Colonius, and Tanguay [**61**] expanded on this model to investigate the effect of bubbles on the focusing and shock wave formation in both ellipsoid reflector and spherical actuator. This work also incorporated the effect of tissue surrounding the stone. Iloreta [**50**] investigated possible inserts into an ellipsoidal reflector lithotripter and their effect on cavitation potential. CLAWPACK [**69**] was used in this work to solve the Euler equations.

The focusing model presented here is similar to recent work but incorporates a multiphysics aspect which allows it to model acoustic wave propagation in both solids and fluids simultaneously. This, in turn, allows for simulation of refracting electromagnetic

lithotripters which are a common type of lithotripter, and not previously modeled. This model will potentially aid in the design of new lenses for refracting electromagnetic lithotripters, and due to the relative ease of replacing lenses in these lithotripters, this is a promising route to improving the procedure. This model also allows for straightforward inclusion of additional linear elastic solids other than the lens. If there is future interest in including stone holders, add-ons to lenses, or other objects this model can be easily adapted.

Some models of fracture or stress distributions within kidney stones or kidney stone simulants have also been developed. Dahake and Gracewski [**27, 28**] used a finite difference scheme to analyze strain within circular and cylindrical stones and compared to experiment. They found locations of high strains due to focusing off of the distal surfaces of the stones. Lokhandwalla and Sturtevant [**77**] used a spring model to approximate the number of shocks required for the initial fracture of a kidney stone. Mihradi *et. al.* [**82**] used a finite element method to investigate stresses within stones. The authors looked at a variety of input pulses and also included a fracture model which compared well with experiments.

Works by Cleveland and Sapozhnikov [**22**] and Sapozhnikov et. al. [**111**] solved the axisymmetric elasticity equations using a centered finite difference scheme which modeled stones surrounded by fluid. The location of the greatest maximum principal stress over time was recorded and compared to the location of the initial fracture in an experimental setup. Several different configurations involving a cylindrical stone were tested. Wijerathne *et. al.* [**128**] applied a 3D dynamic fracture model called PDS-FEM to this problem and were able to predict the initial fracture of one stone configuration. Finally, Luo, in his Ph.D. thesis [**78**], solved the 3D linear elasticity equations for natural

stones taken from μCT imaging, and investigated stress distributions within. Similar to previous work on simulants, the conclusion was that strong shear waves created when a shock wave with large focal width hits the stone lead to high tensile stresses within the stone.

The fracture model presented in this work differs in several key ways from previous work. First, this work uses both realistic stone geometries taken from μCT images and realistic shock waves taken directly from the focusing model. Incorporation of more real world aspects should lend validity and accuracy to the model. In contrast to previous work, this work investigates the total number of shocks required for initial fracture of stones. This is accomplished by using a damage variable to model unresolved damage and fracture. Stone simulants rarely fracture after one shock and in many cases fracture after a repeatable number of shocks with very little variance. In previous work, if a shock passes through the stone without causing fracture, there will be no difference in the stone. Mihradi *et. al.* [**82**] and Wijerathne *et. al.* [**128**] do use the Tuler-Butcher criterion [**123**] which is based on accumulation of damage. However, tracking of damage itself and its effect on subsequent shocks is not included. These works also only investigate single shocks. The damage model in this work is informed by statistics collected from experiment while previous work is deterministic. Finally, this work includes mesoscopic structures able to simulate granular or other realistic mesostructures of simulants or real kidney stones.

1.1.2. μCT Imaging of Fracture. μCT imaging has been used successfully to image rocks and other brittle materials similar to the kidney stone simulants in this work [**130**]. Imaging has also been used to help analyze fracture, both brittle and ductile, over time.

Some examples of using μCT imaging to aid in fracture analysis of brittle materials before and after loading include work by Landis *et. al.* [**62, 63**] and Renard *et. al.* [**105**]. Other work involving imaging of brittle fracture includes [**3, 24, 58, 108, 124, 125**]. Although not directly relevant to this thesis, a few selected works on imaging of metals and ductile fracture include [**6, 9, 79, 88, 122**].

1.1.3. Computational Fracture. The two main goals of computational fracture mechanics are to solve for the distribution of stresses near a crack and to predict growth of a crack. Some computational fracture methods will now be discussed. This list is not meant to be exhaustive, but is presented as a sampling of existing methods.

The most widely used method in computational fracture is the finite element method (FEM) [**4**]. This method lends itself well to this problem because it is straightforward within the confines of the method to model the complicated geometry of existing cracks and to produce a finer mesh near crack tips. This can be done with finite difference and finite volume methods but these require extra implementation features to explicitly represent the crack and therefore aren't as well suited for modeling complicated geometries [**90**]. Remeshing is one drawback of FEM that comes about when modeling crack growth and propagation. Since typically a finer mesh is required near the crack tip than further away from it, the geometry is meshed in that way, but when the crack propagates a new mesh must be developed to correspond to the new location of the crack tip. This is especially difficult when modeling nonlinear materials where a stress history is important. Alternatively, the mesh can be refined where the crack is expected to propagate [**4**].

In an FEM implementation, fracture parameters can be computed using various methods and some examples will be listed here. The boundary collocation method can be used

to compute stress intensity factors [**57**]. The J integral can be computed through numerical integration [**17**]. The virtual crack extension method calculates the energy release rate [**46, 95**]. A more efficient and versatile approach developed more recently called the energy domain integral method also calculates the J integral [**113**]. When computing crack propagation a fracture criterion must be used and when this is met finite elements are either removed or separated.

Several methods exist which employ the FEM framework but do not require the finite elements to explicitly represent the crack. The eXtended Finite Element Method (X-FEM), developed by Belytschko, Black, Moës, and Dolbow in a series of works [**11, 31, 87**], allows for fracture modeling in a FEM implementation in which the cracks are independent of the finite element mesh. This is done by enriching the element nodes near cracks by a discontinuous function. A similar method called E-FEM, developed by Oliver [**91, 92, 93**], enriches the elements instead of the nodes. PDS-FEM, developed by Hori, Oguni, Sakaguchi, and Wijerathne [**48, 129**], as mentioned above, was applied to the problem of kidney stone fragmentation. Like the other methods, PDS-FEM uses functions with discontinuities to represent the effect of cracks. Meshless methods, such as the element-free Galerkin method [**12**], which discretize the domain with an unstructured set of nodes, have also been used to model fracture [**13**].

The previous methods solve continuum equations informed in some way of a crack discontinuity. Another set of methods model fracture by simulating a set of interacting objects, such as a mass-spring system, and are generally called lattice models. These are typically used for mesoscale simulations of materials and have been successfully applied to brittle fracture. An example would be a simulation of granular fracture in concrete, where grains are fully resolved, but the unresolved smaller scale dynamics are homogenized as

linear elasticity. Lattice methods were first described in 1941 by Hrennikoff [**49**]. They have been developed with both beam theory [**30, 47, 72, 112**] and with spring systems [**14, 26, 53, 56**].

Fracture has also been modeled with atomistic models. These models are very accurate due to the incorporation of the true small scale physics that leads to fracture. However, they are unable to model large scale fracture simply due to the vast number of atoms required. Even on the most powerful computers they are restricted to very small spatial and temporal domains. Some work on fracture using these methods includes [**29, 38, 39, 44, 54**].

1.1.4. Multiscale Fracture. Multiscale computations of fracture are an attractive option since they can theoretically achieve the accuracy of atomistic type methods over larger spatial and longer time scales. This is typically accomplished by only computing on the small scale when and where necessary and solving continuum equations away from cracks and defects.

Kohlhoff *et. al.* [**59**] proposed a finite element combined with atomistic (FEAt) model. This method joined a lattice atomistic model with a continuum finite element model by a transition region which used displacements to link the two regions. The original work applied the method to crack propagation in bcc crystals and later work by Gumbsch and Beltz [**43**] applied it to additional fracture problems. Tadmor, Ortiz, and Phillips [**119**] developed the Quasicontinuum (QC) method which also employs finite element and atomistic models. Here, the domain is discretized by finite elements and representative atoms, which represent many atoms, and macroscopic constitutive laws

are based on atomistic calculations. This method has been successfully applied to fracture as well [**84, 85**].

The coupling length scales (CLS) [**2, 16, 109**] and finite element, molecular dynamics, tight-binding (FE MD TB) methods [**1**] use a similar transition region as the FEAt model to connect finite element and atomistic regions, but employ a linear elastic approximation. The coupled atomistic and discrete dislocation (CADD) [**114**] method also employs a transition zone but includes a continuum representation of defects and discontinuities which allows for larger models of plastic flow [**25**]. The bridging scale decomposition method [**127**] relates MD and continuum mechanics by projecting the MD solution onto the coarse scale basis functions. Details of the above mentioned methods, as well as additional methods and comparisons between methods can be found in review papers such as [**25, 85, 94**].

The work presented here models fracture at the continuum level but informed by mesoscopic structures. In most multiscale fracture simulations, the model attempts to capture the true molecular physics leading to fracture, whereas in this work that level remains unresolved. Therefore, for this work, the multiscale model is described as being a continuum-mesoscopic interaction. This work also uses a finite volume implementation compared to most work that uses finite elements. The finite volume implementation is retained in order for the model to work seamlessly with the focusing model as well as accurately model the incoming shock wave, since this model includes a realistic shock.

CHAPTER 2

Background

This chapter contains a discussion on background material associated with this work. The first section describes the application domain of this project, which is shock wave lithotripsy. Next, background is provided on the solid mechanics aspects of the work, which includes a discussion on continuum damage mechanics and fracture mechanics.

2.1. Shock Wave Lithotripsy

During the 1990s 70–80% of kidney stones were treated using ESWL, with ureteroscopic stone removal and percutaneous nephrolithotomy accounting for the remainder [**75**]. The latter is a surgical technique reserved for large stones. One main and constant advantage of ESWL over the other two techniques is a lower complication rate and a shorter hospital stay. Recent technological advances have increased the efficacy of the ureteroscopic stone removal techniques, but ESWL remains the first treatment choice for most stones of size less than 2.5 cm [**75**].

ESWL was developed in 1980 [**19**] and first introduced clinically in 1984 with the Dornier HM3 lithotripter [**73, 74**]. This lithotripter was proven to be very effective and has stone-free rates, meaning the procedure was successful, of 77-90%. The disadvantages of this lithotripter, including the size, water bath, and the amount of anesthesia required caused many new lithotripters to be designed. Most of these machines are less effective, i.e. have stone-free rates anywhere from 45 to 60% and have a higher retreatment rate.

Only recently have modern lithotripters approached the stone-free rates of the Dornier HM3 [**83**].

There are three types of pulse generation used in lithotripters. These are electro-hydraulic (EH), piezoelectric (PE), and electromagnetic (EM). The Dornier HM3 is an example of an EH lithotripter. This type uses an ellipsoid reflector and a spark discharge at one focus of the ellipsoid to create a shock wave. The shock wave travels through the surrounding water, reflects off the ellipsoid reflector, travels into the patient and focuses at the kidney stone which is positioned at the second focus of the ellipsoid.

Piezoelectric actuators with a spherical shape are used in PE lithotripters. The actuator is only a portion of a sphere, and when activated, produces a pulse that travels through water, into the patient's body, and focuses at the center of the sphere, where the kidney stone has been positioned. Several methods of focusing have been introduced for EM lithotripters including reflection, refraction and spherical actuators [**33**]. Reflecting EM lithotripters use a cylindrical electromagnetic actuator and a paraboloid reflector. The kidney stone is positioned at the focus of the paraboloid.

This thesis presents a model for refracting EM lithotripters. The electromagnetic actuator creates a flat circular pulse which is then focused by means of an acoustic lens, typically made of polystyrene. Like the other methods, the actuator and lens are surrounded by water and the kidney stone is positioned at the geometrical focus of the lens.

Variability in shock features such as rise time and peak pressures, and the short lifetime of the electrodes in the original EH lithotripters led to the development of EM and PE lithotripters. Since PE lithotripters have had poor clinical performance [**75, 103**], EM lithotripters have become very common. In fact, most lithotripters developed during

the 1990s used EM pulse generation [**75**]. The three focusing methods are illustrated in Figure 2.1.

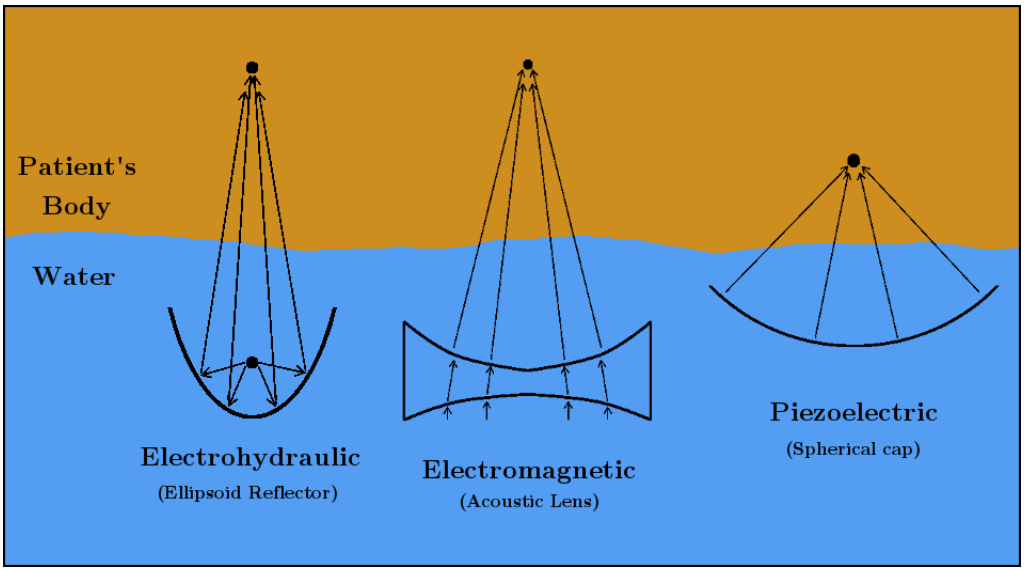


FIGURE 2.1. Illustration showing the three focusing methods used in lithotripsy: reflector, lens, and spherical actuator. Electromagnetic lithotripters are not restricted to refraction and have been used with parabolic reflectors and spherical actuators.

Effective transmission of acoustic energy to the patient in the Dornier HM3 requires immersion of the patients torso in a water bath which also contains the ellipsoid reflector and spark gap. This requires a large, unwieldy apparatus. In contrast, all modern lithotripters, regardless of the focusing method, house the pulse generating mechanism, the focusing apparatus, and the transmission medium (typically water) in a mobile arm pressed against the patient. The casing for the transmission medium is a soft rubbery material that can deform to the patient's body. Ultrasound gel is also used to aid in the transmission. To allow the urologist or technician to see the kidney stone and aim the device flouroscopic X-ray imaging is used [**74**]. An average patient is typically subjected to around 2000 pulses in one treatment session at 1 pulse per second, with varying

amplitude during the procedure [**75**]. Some sort of anesthesia is almost always used, whether general or local.

An example plot of the pulse at the focus is shown in Figure 2.2. This particular data is recorded using a refracting EM lithotripter but pulses from other types are similar. This was also recorded within an experimental setup which approximates the patient's body with additional water. The plot shows the pressure recorded by a hydrophone over time at the focus of the lens. The first portion of the wave is a high amplitude compressive shock. This is followed by a longer and relatively weaker tensile region, which is in turn followed by a compressive region and another tensile region. In addition to pulse profiles at the focus, profiles from other points in the focal plane (the plane perpendicular to the propagation axis) as well as other positions along the propagation axis will be shown in Chapter 5.

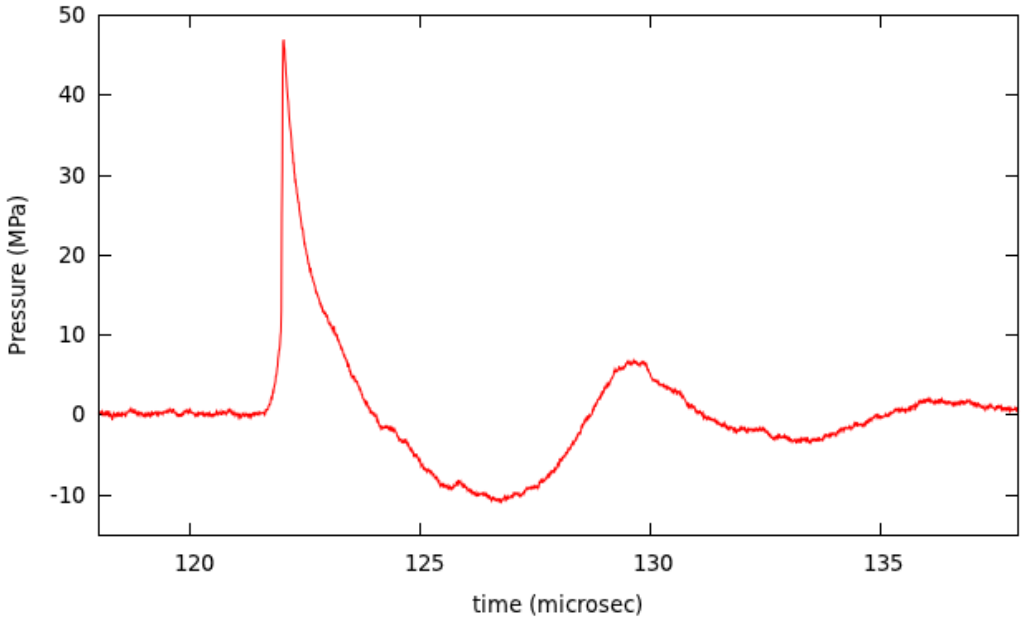


FIGURE 2.2. Plot of pressure over time at the focus of an electromagnetic lithotripter. (Data was recorded by Nathan Smith at Duke University using an optical hydrophone)

Several physical mechanisms contribute to the break up of the kidney stone. The main processes identified so far in the literature are tearing and shearing, spallation, quasi-static squeezing, cavitation effects, and dynamic squeezing [**103**]. Cavitation will be described first since it takes place in the liquid surrounding the stone. Cavitation refers to the formation of vapor bubbles in a liquid due to tensile forces and their subsequent collapse. Away from the stone this collapse is symmetric and creates a spherical shock wave that propagates in all directions and weakens rapidly. Bubbles can also form near the stone surface and when these bubbles collapse they do so asymmetrically, forming a jet that impinges on the surface of the stone, which leads to breaking and crack formation. Cavitation is a very important component of the overall kidney stone break up and the comminution into smaller pieces of stone [**103**]. Cavitation bubbles can also interfere with subsequent shock waves and lead to tissue damage. Much research is devoted to this subject and how to reduce cavitation away from the stones while maintaining the effect near the stone [**103**].

Tearing and shearing results from a large stress gradient within the stone as the strong leading compressive shock wave passes through. This effect is only usually relevant when the focal zone is small relative to the stone. Spallation occurs when a portion of the leading compressive shock wave reflects off the back of the stone as a tensile wave, due to the fluid behind the stone being acoustically softer. This reflected wave combines with the tensile portion of the incoming wave to create a region of high tensile stress. The distance of this location from the distal end of the stone can be calculated simply by knowing the time difference between the peak positive and peak negative points on the incoming wave and the longitudinal wave speed in the stone. For example, if the pulse shown in Figure 2.2 has of a duration of 4 μs between the positive and negative

peak and the speed of sound in the stone is 4159 m/s then within the stone these peaks would be separated by about 16.6 mm. Therefore, the spallation effect would occur at approximately 8.3 mm from the distal surface of the stone.

The quasi-static and dynamic squeezing effects require the focal zone to be large relative to the stone. Quasi-static squeezing occurs when the compressive wave passes more quickly through the stone than in the surrounding fluid. The compressive waves in the fluid surround and squeeze the stone. Dynamic squeezing also incorporates this effect but more specifically refers to the process of shear waves forming at leading corners of the stone, being driven by squeezing from the compressive waves outside the stone and creating strong tensile stresses within the stone [**103, 111**]. This process causes some of the highest tensile stresses found in the stone during the procedure, and for cylindrically shaped stones explains the consistent initial fragmentation of the distal third of the stone.

Recently, Smith and Zhong [**116**] and Zhong [**135**] have found single parameters that correlate well with overall stone comminution. These parameters are the average peak pressure incident on the stone and a non-dimensional based on the average peak pressure, a critical pressure value, and the shear modulus of the stone material. Published work also includes relating initial flaw distributions to probability of failure as well as other accepted methods in fracture mechanics [**135**].

2.2. Continuum Damage Mechanics

Continuum damage mechanics (CDM) is a homogenization technique which characterizes all microvoids and microcracks in a representative volume element (RVE) as a scalar or tensor. In this way these discontinuities can be represented in a continuous manner at a larger scale. Typical sizes in engineering applications for the RVE range

from $(0.1\,\text{mm})^3$ for metals to $(100\,\text{mm})^3$ for concrete [**66**]. CDM is often used up to the point of macrocrack initiation. Once macrocracks appear, fracture mechanics is used as a model instead and the cracks are modeled as discontinuities [**134**]. The development of CDM is mainly attributed to the work of Kachanov [**55**], Robotnov [**107**], Lemaitre [**64**], and Chaboche [**18**], among others.

Incorporating CDM into an existing continuum mechanical model requires three basic additions [**134**]:

(1) A damage variable, mentioned previously, which describes the amount of damage in each RVE,

(2) A damage growth law which is an equation typically dependent on the current stress or strain in the RVE as well as some sort of criterion for damage increase,

(3) A new constitutive relation that includes the damage variable.

The most general form of the damage variable is a fourth order tensor. The tensor relates a plane in reference configuration through the RVE with surface area, δS, to a plane in effective configuration with surface area, $\delta\tilde{S}$, by

$$(I_{ijkl} - D_{ijkl})\, v_k n_l \delta S = v_i \tilde{n}_j \delta\tilde{S}, \tag{2.2.1}$$

where $\vec{n}$ is the normal to the reference plane, $\tilde{\vec{n}}$ is the normal to the effective plane, and $\vec{v}$ is a reference vector. The surface areas are related by $\delta\tilde{S} = \delta S - \delta S_D$ where δS_D is the surface area of cracks and voids which intersect with the reference plane. Damage can also be represented by a second order tensor and the definition is similar to (2.2.1) and

is given by

$$(\delta_{ij} - D_{ij})\, n_j \delta S = \tilde{n}_i \delta \tilde{S}. \tag{2.2.2}$$

These tensor forms of the damage variable can represent anisotropic damage, which is often necessary as cracks generally form in specific directions like the direction perpendicular to the maximum tensile stress [**66**].

An isotropic damage law uses a scalar representation and it is assumed that the orientation of the microcracks and voids are uniformly distributed within the volume element. The scalar damage, D, is given by

$$D = \frac{\delta S_D}{\delta S}. \tag{2.2.3}$$

While this cannot account for anisotropic damage it can be used as an approximation for three dimensional problems [**66**]. As seen from Equation (2.2.3) the damage, D, ranges from 0 to 1, and 0 corresponds to a completely undamaged state. Usually some critical value, D_c, is specified which corresponds to the rupture of the element or the initiation of a macrocrack. This value typically ranges from 0.2 to 0.8 depending on the material [**134**]. The remainder of this section assumes isotropic damage. See references [**65, 66, 126, 134**] for descriptions of the following concepts incorporating anisotropic damage.

A damage growth law or equation is a differential equation that defines how the damage increases in time. One general example is the Isotropic Unified Damage Law

given by

$$\frac{dD}{dt} = \left(\frac{Y}{S}\right)^s \frac{\dot{\lambda}}{1-D}, \tag{2.2.4}$$

where Y is the energy density release rate, $\dot{\lambda}$ is a plastic multiplier, and S and s are material parameters [**66**]. As is common to many damage growth laws, a condition accompanies the equation. For this law, the equation is integrated forward in time if $\dot{\lambda}$ exceeds some value, otherwise there is no change in damage, i.e. $\frac{dD}{dt} = 0$ [**66**].

The stress which acts on the effective configuration (or area) is called the effective stress and is given component wise by

$$\tilde{\sigma}_{ij} = \frac{\sigma_{ij}}{1-D}. \tag{2.2.5}$$

Linear elastic models use Hooke's law as a constitutive relation

$$\epsilon = \frac{1+\nu}{E}\sigma - \frac{\nu}{E} Tr\left(\sigma\right)\mathbf{I}, \tag{2.2.6}$$

where $Tr(\sigma)$ is the trace of the stress tensor, $\mathbf{I}$ is the identity matrix, ν is Poisson's ratio, and E is Young's modulus. Adding a scalar damage variable changes equation (2.2.6) to

$$\epsilon = \frac{1+\nu}{E}\frac{\sigma}{1-D} - \frac{\nu}{E}\frac{Tr\left(\sigma\right)}{1-D}\mathbf{I}, \tag{2.2.7}$$

after substituting in the effective stress [**65**].

2.3. Fracture Mechanics

The field of fracture mechanics has a long history with many useful results. This section will focus only on concepts necessary for understanding this work which involves brittle dynamic fracture. First, the general process of fracture will be described and it will be shown that this is an inherently multiscale problem. Following this general description, some history, common parameters and fracture criteria will be discussed.

The occurrence of fracture due to a load on an object depends on the material properties and geometry of the object, the geometry of existing cracks, the distribution of the load, and the magnitude of the load, as well as other factors. This leads to a complex problem even before the issues of multiple scales are included. Dynamic fracture refers to a time dependence, as opposed to quasi-static fracture, and can incorporate inertia effects and stress waves created by propagating cracks [**4**].

There are three main types (or modes) of fracture. Mode I is the opening mode due to tensile stress normal to the crack plane, Mode II is the sliding mode due to shear stress parallel to the crack plane and perpendicular to the crack front, and Mode III is the tearing due to from shear stress parallel to the crack plane and parallel to the crack front. Crack growth can either be stable or unstable. Stable crack growth leads to very small additions to the size of a crack. Fracture is sometimes defined as unstable crack growth, since this process adds significant length to an existing crack over very short periods of time. The theoretical upper limits for the velocity of unstable mode I, II, and III crack growth are the Rayleigh wave speed, the longitudinal wave speed, and the shear wave speed of the material, respectively [**15**]. Crack velocities typically only reach

fractions of these wave speeds, but even so, can easily reach into the hundreds of meters per second for many materials.

Some important features of fracture are the direction of fracture, branching, and arrest. The general direction of crack growth is in the plane perpendicular to the direction of the maximum principal stress. The maximum principal stress is the largest eigenvalue of the stress tensor, and the direction it acts in is the associated eigenvector. Branching during crack growth can also occur. This typically only happens with mode I cracks and only if the crack velocity is a significant percentage of the Rayleigh wave speed [**15**]. Crack arrest may occur when the driving force of the crack falls below the material strength or if the propagating crack enters a region with a higher toughness [**4**].

The effect of fracture can clearly be macroscopic but the processes leading to fracture are typically microscopic. The initiation of a macroscopic crack results from the nucleation, growth, and eventual coalescence of microscopic cracks and voids. During loading, stress concentrations at the macroscopic crack edges and tips cause propagation of the crack. Again, this propagation is due to nucleation and growth of microscopic cracks and coalescence with the macroscopic crack. Many brittle materials will also have pre-existing microscopic cracks and voids that can grow and coalesce.

In brittle materials like ceramics, glass, or rocks the microscopic structures are rigid and internal slip cannot occur without permanent microcracks being formed [**15**]. Inhomogeneous brittle materials typically have a granular structure and the microscopic fracture process is more likely to occur along grain boundaries where the molecular bonds are weaker. There is also a greater likelihood of pre-existing microvoids along grain boundaries [**15**]. In a brittle material it is assumed that linear elasticity is valid very near existing cracks as well as up to the point of fracture. Figure 2.3 shows an example

stress-strain diagram for a brittle material. As seen in the figure, the linear relationship between stress and strain is valid almost to the point of fracture and therefore linear elasticity is typically accepted as a valid approximation.

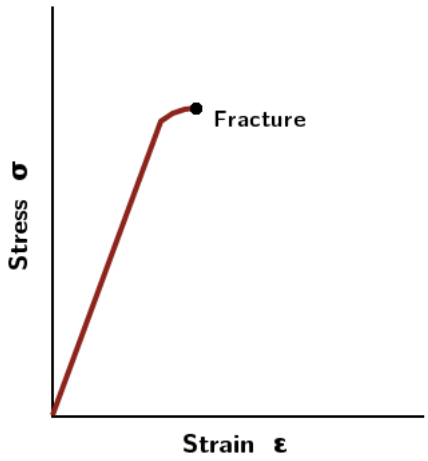


FIGURE 2.3. Stress strain diagram for a brittle material.

Linear elastic fracture mechanics (LEFM) was the earliest substantial work on fracture and is mainly attributed to the work of Griffith [**42**] and Irwin [**51, 52**]. Griffith produced the first fracture criterion which gave the critical stress required for fracture as

$$\sigma = \sqrt{\frac{2E\gamma}{\pi a}}, \tag{2.3.1}$$

where E is Young's modulus, γ is the surface energy density, and a is half the crack length [**42**]. Irwin modified Griffith's theory to account for failure in ductile materials [**51**]. Irwin also introduced the concept of stress intensity factors which describe the distribution of stress near a crack tip [**52**]. The stress intensity factors are defined as

limits

$$
\begin{aligned}
K_I &= \lim_{r\to 0} \sqrt{2\pi r}\sigma_{yy}\left(r,0\right) \\
K_{II} &= \lim_{r\to 0} \sqrt{2\pi r}\sigma_{yx}\left(r,0\right) \\
K_{III} &= \lim_{r\to 0} \sqrt{2\pi r}\sigma_{yz}\left(r,0\right),
\end{aligned}
\tag{2.3.2}
$$

where K_I, K_{II}, and K_{III} correspond to the three fracture modes, the stress fields are defined by a polar coordinate system (r, ϕ) with origin at the crack tip, and the crack is propagating along the x direction. The stress field can be written asymptotically in terms of the stress intensity factors as

$$
\sigma_{ij} = \frac{K_I}{\sqrt{2\pi r}} f_{ij}^{I}\left(\phi\right) + \frac{K_{II}}{\sqrt{2\pi r}} f_{ij}^{II}\left(\phi\right) + \frac{K_{III}}{\sqrt{2\pi r}} f_{ij}^{III}\left(\phi\right),
\tag{2.3.3}
$$

where the functions f are not material dependent [**15**].

Another commonly used parameter, also introduced by Irwin, is the energy release rate, denoted by $\mathcal{G}$. For a crack in an infinite plane loaded under tensile stress, σ, the energy release rate is

$$
\mathcal{G} = \frac{\pi\sigma^2 a}{E},
\tag{2.3.4}
$$

where E is Young's modulus and a is half the crack length. Similar to Equation 2.3.1, both the stress intensity factors or the energy release rate can be used as a fracture criterion by determining a critical value of either parameter, K_c or $\mathcal{G}_c$, which are both referred to as the fracture toughness [**4**]. LEFM models quasi-static fracture and when time dependencies are taken into account the domain is referred to as elastodynamic fracture mechanics.

One additional very common parameter, although developed for nonlinear materials, is the J integral, introduced by Rice [**106**]. The J integral is a measure of the nonlinear energy release rate and is defined by a contour integral around the crack tip. It also characterizes stresses and strains near the crack tip and so can be considered a stress intensity parameter in addition to an energy parameter [**4**]. This parameter is given by

$$J = \int_{\Gamma} \left(w dy - T_i \frac{\partial u_i}{\partial x} ds \right), \tag{2.3.5}$$

where Γ is an arbitrary path around the crack tip, w is the strain energy density, T_i are the components of the traction vector, and u_i are the displacement vector components. The strain energy density is

$$w = \int_0^{\epsilon_{ij}} \sigma_{ij} d\epsilon_{ij} \tag{2.3.6}$$

and the traction vector is

$$T_i = \sigma_{ij} n_j, \tag{2.3.7}$$

where σ_{ij} is the stress tensor, ϵ_{ij} is the strain tensor and n_i are the components of the unit normal vectors to Γ [**4**].

Many fracture criteria have been developed for predicting the onset of fracture in brittle materials. The simplest is the maximum stress criterion which states that material failure will occur if the maximum principal stress exceeds the tensile strength of the material, similar to the criteria presented in the preceding paragraphs. This criterion is often used for brittle materials. Other commonly used criteria are the Tresca, von Mises,

and Mohr-Coulomb criteria [**96**]. The Tresca and von Mises criteria are usually applied to ductile materials and the Mohr-Coulomb to brittle materials. Criteria have also been developed for anisotropic materials as well as specific materials like concrete, wood, or soil.

Kidney stone fracture during lithotripsy is classified as dynamic brittle fracture [**135**] and modeling requires a criterion reflecting this category of fracture. The Tuler-Butcher criterion [**123**], which is based on accumulation of damage, has been used in lithotripsy fracture modeling [**82, 128**]. While the criterion assumes damage accumulation, the damage itself is not modeled. In this work, the damage is modeled explicitly in an anisotropic fashion and modifies a simple maximum stress criterion.

CHAPTER 3

Finite-Volume Riemann-Solvers

This chapter discusses the main computational technique used in this work which is a finite-volume Riemann solver for wave propagation of hyperbolic systems of partial differential equations (PDEs) developed by LeVeque [**69**]. This method is used for the focusing model described in Chapter 5 and for the continuum level portion of the fracture model described in Chapter 6.

The main attractive feature of this type of method is its ability to accurately capture the formation and propagation of shocks. Since the lithotripsy focusing model requires a solution of exactly that, this method is employed. In addition, the input for the fracture model contains the shock wave from output of the focusing model. This numerical method is also designed to be applied to linear wave-propagation studies with heterogeneities, which is another aspect present in both the focusing and fracture models.

First, general background on the method is given in both an analytical and numerical context. Next, the systems of equations modeled by this method in this work are discussed. This includes the 2D axisymmetric and 3D linear elasticity equations, and the 2D axisymmetric Euler equations. Finally, modifications to the current implementations of the method are described and the solver is verified by a convergence test and by comparing reflection and transmission coefficients.

3.1. Analytical Considerations

The Riemann problem for a one-dimensional hyperbolic system of PDEs is defined by a discontinuous initial condition that is piecewise constant with a single discontinuity, given generally by

$$q_0(x) = \begin{cases} a & x \leq x_0 \\ b & x > x_0 \end{cases}, \tag{3.1.1}$$

where $x \in \mathbb{R}$, $q_0(x), a, b \in \mathbb{R}^m$, and m is the number of waves admitted by the PDEs. This problem and its solution become the basis for the numerical method described in this chapter, as it is solved at each interface of the discretized domain.

Consider the advection equation

$$v_t + cv_x = 0, \tag{3.1.2}$$

where $v = v(x,t)$, c is the advection speed, and subscripts denote differentiation, with initial condition

$$v(x,0) = f_0(x). \tag{3.1.3}$$

For this PDE, $m = 1$. By the method of characteristics, the solution is constant along characteristics, which gives

$$v(x,t) = f_0(x - ct), \tag{3.1.4}$$

as the solution.

The basics of a Riemann solver for linear hyperbolic systems are now presented with the 1D elasticity equations as an example,

$$\begin{aligned} \sigma_t^{11} - \lambda u_x &= 0 \\ \rho u_t - \sigma_x^{11} &= 0, \end{aligned} \tag{3.1.5}$$

where σ^{11} is the normal stress in the x-direction, u is the displacement velocity, λ is the first Lamé parameter, and ρ is density. In vector form the equations are

$$q_t + Aq_x = 0, \tag{3.1.6}$$

where

$$q = \begin{pmatrix} \sigma^{11} \\ u \end{pmatrix} \tag{3.1.7}$$

and

$$A = \begin{pmatrix} 0 & -\lambda \\ -1/\rho & 0 \end{pmatrix}. \tag{3.1.8}$$

Hyperbolic PDEs admit an eigenvalue decomposition,

$$AR = R\Lambda, \tag{3.1.9}$$

that decouples the system. The columns of R are the eigenvectors of A and Λ contains the eigenvalues along its diagonal,

$$R = \begin{pmatrix} c_p & c_p \\ 1/\rho & -1/\rho \end{pmatrix} \tag{3.1.10}$$

and

$$\Lambda = \begin{pmatrix} -c_p & 0 \\ 0 & c_p \end{pmatrix}, \tag{3.1.11}$$

where $c_p = \sqrt{\lambda/\rho}$ is the speed of sound in the medium. Introducing the characteristic variables, w, which are found from solving $Rw = q$, allows the system to be rewritten as a set of decoupled advection equations

$$w_t + \Lambda w_x = 0. \tag{3.1.12}$$

From the method of characteristics the exact solution is known. Each component of Equation (3.1.12) is an advection equation with known solution

$$w^p(x,t) = w_0^p(x - \lambda^p t) \tag{3.1.13}$$

where w_0 is found from $Rw_0 = q_0$, q_0 is the initial condition, $p = 1, 2$ denotes the component, and λ^p are eigenvalues. The solution in the original variables is then given by

$$q(x,t) = Rw(x,t). \tag{3.1.14}$$

This process can generally be applied to any linear hyperbolic system of PDEs. In addition, nonlinear systems may be approximated by a linear system

$$q_t + f(q)_x \approx q_t + Aq_x = 0. \tag{3.1.15}$$

If the approximation is made locally across a computational domain in a numerical algorithm, then this leads to a valid approximation of the solution.

3.2. Numerical Implementation

The underlying idea for the numerical implementation of Riemann solvers was first presented by Godunov [**41**] which included the essentials of the reconstruct-evolve-average (REA) algorithm. This was based on a finite volume discretization of the domain, so at the beginning of a time step each cell contained cell averages of the field variables. The algorithm is:

(1) Reconstruct a piecewise function from the cell averages. The simplest case, which is often employed, is a piecewise constant function.

(2) Evolve the solution over one time step. This process was alluded to in the previous section and the idea is to decompose the jump in the solution at the cell boundary into the eigenvectors of the system matrix. This is in order to advect these jumps in the proper direction and at the proper speed, given by the eigenvalues.

(3) Average the updated solution over each grid cell to attain the new cell averages.

The remainder of this section presents the wave propagation algorithm of LeVeque which is based on the general theory of Riemann solvers and the REA algorithm. More

details can be found in LeVeque's work [**69, 70**]. This method also forms the basis of implementation in the CLAWPACK software developed by Leveque [**68**], and the related BEARCLAW software by Mitran [**86**].

3.2.1. First Order Method. To proceed with the numerical description the first order method is described first. The update formula is

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(A^+ \Delta Q_{i-1/2} + A^- \Delta Q_{i+1/2}\right), \tag{3.2.1}$$

where Q_i^n refers to the solution value in spatial cell i and at time step n, Δt is the time step, Δx is the spatial step, $A^+ = R\Lambda^+ R^{-1}$, $A^- = R\Lambda^- R^{-1}$, and $\Delta Q_{i-1/2} = Q_i - Q_{i-1}$ is the jump in the solution values at the cell boundaries. Λ^+ contains the positive eigenvalues on the diagonal with the negative eigenvalues replaced by zero, and the opposite for Λ^-, $\Lambda = \Lambda^+ + \Lambda^-$.

Typically the terms are not computed this way but are found from

$$\begin{aligned} A^+ \Delta Q_{i-1/2} &= \sum_{p=1}^{m} (\lambda_p)^+ W_{p,i-1/2} \\ A^- \Delta Q_{i-1/2} &= \sum_{p=1}^{m} (\lambda_p)^- W_{p,i-1/2}, \end{aligned} \tag{3.2.2}$$

where m is the number of waves (the number of nonzero eigenvalues of A), and the speeds are

$$(\lambda_p)^+ = \begin{cases} \lambda_p & \lambda_p > 0 \\ 0 & \lambda_p \leq 0 \end{cases} \tag{3.2.3}$$

$$(\lambda_p)^- = \begin{cases} \lambda_p & \lambda_p < 0 \\ 0 & \lambda_p \geq 0 \end{cases}. \tag{3.2.4}$$

The waves, $W_{p,i-1/2}$, are the jumps at the cell boundaries decomposed into the eigenvectors of A,

$$W_{p,i-1/2} = \alpha_{p,i-1/2} r_p, \tag{3.2.5}$$

where $\alpha_{p,i-1/2} \in \mathbb{R}$, $W_{p,i-1/2}, r_p \in \mathbb{R}^m$, and $\alpha_{p,i-1/2}$ is found by solving

$$R\alpha_{i-1/2} = \Delta Q_{i-1/2} \tag{3.2.6}$$

where $\alpha_{i-1/2}, \Delta Q_{i-1/2} \in \mathbb{R}^m$ and $R \in \mathbb{R}^{m \times m}$. The eigenvectors of the system matrix remain the same throughout the computation so the linear solve is typically computed analytically before the simulation.

For the 1D elasticity example the wave coefficients are

$$\begin{aligned} \alpha_{1,i-1/2} &= \frac{\delta_1 + c_p \delta_2 \rho}{2c_p} \\ \alpha_{2,i-1/2} &= \frac{\delta_1 - c_p \delta_2 \rho}{2c_p}, \end{aligned} \tag{3.2.7}$$

where

$$\begin{pmatrix} \delta_1 \\ \delta_2 \end{pmatrix} = \Delta Q_{i-1/2} = \begin{pmatrix} \sigma_i^{11} - \sigma_{i-1}^{11} \\ u_i - u_{i-1} \end{pmatrix}. \tag{3.2.8}$$

So the update formula becomes

$$Q_t^{n+1} = Q_t^n - \frac{\Delta t}{\Delta x}\left(c_p \alpha_{1,i-1/2} \begin{pmatrix} c_p \\ -1/\rho \end{pmatrix} - c_p \alpha_{2,i+1/2} \begin{pmatrix} c_p \\ 1/\rho \end{pmatrix} \right). \tag{3.2.9}$$

This first order method is a generalization of the upwind method to systems of equations.

3.2.2. Higher Order Methods. To produce a second order method, a correction term is added to the update formula

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(A^+ \Delta Q_{i-1/2} + A^- \Delta Q_{i+1/2}\right) - \frac{\Delta t}{\Delta x}\left(\tilde{F}_{i+1/2} - \tilde{F}_{i-1/2}\right), \tag{3.2.10}$$

where the terms in the correction are given by

$$\tilde{F}_{i-1/2} = \frac{1}{2}\sum_{p=1}^{m} |\lambda_p| \left(1 - \frac{\Delta t}{\Delta x} |\lambda_p| W_{p,i-1/2}\right). \tag{3.2.11}$$

The above correction is a Lax-Wendroff scheme. This and some other second order schemes can introduce oscillations and so wave limiters are included to eliminate this effect. The correction formula written in the flux-limiter form is

$$\tilde{F}_{i-1/2} = \frac{1}{2}\sum_{p=1}^{m} |\lambda_p| \left(1 - \frac{\Delta t}{\Delta x} |\lambda_p| \tilde{W}_{p,i-1/2}\right) \tag{3.2.12}$$

where

$$\tilde{W}_{p,i-1/2} = \tilde{\alpha}_{p,i-1/2} r_p, \tag{3.2.13}$$

$$\tilde{\alpha}_{p,i-1/2} = \alpha_{p,i-1/2}\phi\left(\theta_{p,i-1/2}\right), \tag{3.2.14}$$

and

$$\theta_{p,i-1/2} = \frac{\alpha_{p,I-1/2}}{\alpha_{p,i-1/2}}, \tag{3.2.15}$$

where

$$I = \begin{cases} i-1 & \lambda_p > 0 \\ \\ i+1 & \lambda_p < 0 \end{cases}. \tag{3.2.16}$$

Many second order methods as well as high resolution slope limiters can be written in this form simply by changing the function ϕ. Table 3.1 is a selection of such methods.

TABLE 3.1. List of second order methods and slope limiters and their defining function.

Method	$\boldsymbol{\phi(\theta)}$
upwind	0
Lax-Wendroff	1
Beam-Warming	θ
Fromm	$\frac{1}{2}(1+\theta)$
minmod	$\text{minmod}(1,\theta)$
superbee	$\max(0,\min(1,2\theta),\min(2,\theta))$
monotized central-difference	$\max(0,\min((1+\theta)/\theta,2,2\theta))$
van Leer	$\frac{\theta+\lvert\theta\rvert}{1+\lvert\theta\rvert}$

3.2.3. Two and Three Dimensions. So far, this description has been of a system of equations in one dimension. This section will describe the implementation in two dimensions and briefly the extension to three dimensions. The general form of the system of equations being solved in 2D is

$$q_t + Aq_x + Bq_y = 0. \tag{3.2.17}$$

The eigenvalues and eigenvectors of A will be labeled by λ_p^x and r_p^x and for B these will be labeled by λ_p^y and r_p^y. Figure 3.1 contains a diagram showing the fluxes resulting from a normal solve in 2D and is included as a visual reference for the description in this section.

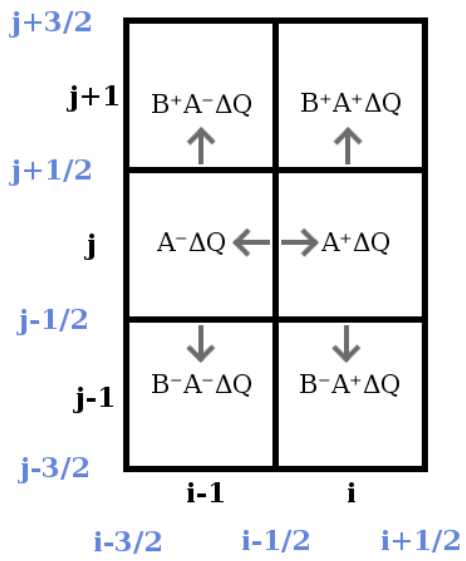


FIGURE 3.1. Reference diagram showing a finite volume grid in 2D with labeled cells and boundaries. Also shown is an illustration of the normal and transverse fluxes resulting from the normal Riemann solve at boundary $(i - 1/2, j)$.

There are known solutions to the 2D and 3D Riemann problems but the approach taken in LeVeque's wave propagation algorithm [**70**] is to use an approximation by dimensional splitting. In 2D, the approach solves the Riemann problem as it would in 1D in both the x and y-directions independently, but then also computes transverse fluctuations to update the correction terms. These transverse waves refer to portions of the left-going and right-going waves emanating from the cell boundary going into cells above or below the current slice of cells. Exclusion of the transverse waves leads to a stricter stability condition than the standard CFL ≤ 1 for hyperbolic systems. If $\Delta x = \Delta y$ then

the stability condition would become CFL < 0.5. Correction terms in the x-direction are given by $\tilde{F}_{i-1/2,j}$ and in the y-direction are given by $\tilde{G}_{i,j-1/2}$. The following provides a framework for computing a single time step update for a two dimensional solver.

First the correction terms, $\tilde{F}_{i-1/2,j}$ and $\tilde{G}_{i,j-1/2}$, are set to 0 at all cell boundaries. Then the fluctuations in the x-direction are computed as before by first decomposing the jumps at the cell boundaries into the eigenvectors of A by

$$\Delta Q_i - 1/2, j = \sum_{p=1}^{m} \alpha_{p,i-1/2,j} r_p^x \tag{3.2.18}$$

and

$$A^{\pm} \Delta Q_{i-1/2,j} = \sum_{p=1}^{m} \left(\lambda_p^x\right)^{\pm} \alpha_{p,i-1/2,j} r_p^x. \tag{3.2.19}$$

Next, the correction terms are updated by the limiter method being used by

$$\tilde{F}_{i-1/2,j} := \tilde{F}_{i-1/2,j} + \frac{1}{2} \sum_{p=1}^{m} |\lambda_p| \left(1 - \frac{\Delta t}{\Delta x} |\lambda_p| \tilde{W}_{p,i-1/2}\right). \tag{3.2.20}$$

The assignment symbol, :=, in the above equation indicates that this newly computed term should be added to the current value of $\tilde{F}_{i-1/2,j}$.

Next, the transverse fluctuations are computed by decomposing the x-direction fluctuations into the eigenvectors of B by

$$A^{+} \Delta Q_{i-1/2,j} = \sum_{p=1}^{m} \beta_p r_p^y, \tag{3.2.21}$$

and then setting

$$B^{\pm}A^{+}\Delta Q_{i-1/2,j} = \sum_{p=1}^{m} \left(\lambda_p^y\right)^{\pm} \beta_p r_p^y. \tag{3.2.22}$$

The same is done with $A^{-}\Delta Q_{i-1/2,j}$. The y-direction correction terms are then updated using these values by

$$\begin{aligned} \tilde{G}_{i,j\pm1/2} &:= \tilde{G}_{i,j\pm1/2} - \frac{\Delta t}{2\Delta x} B^{\pm}A^{+}\Delta Q_{i-1/2,j} \\ \tilde{G}_{i-1,j\pm1/2} &:= \tilde{G}_{i-1,j\pm1/2} - \frac{\Delta t}{2\Delta x} B^{\pm}A^{-}\Delta Q_{i-1/2,j}. \end{aligned} \tag{3.2.23}$$

At this point the process is repeated for the y-direction with transverse solves in the x-direction. The final update formula for two dimensions is

$$\begin{aligned} Q_{i,j}^{n+1} = Q_{i,j}^{n} &- \frac{\Delta t}{\Delta x}\left(A^{+}\Delta Q_{i-1/2,j} + A^{-}\Delta Q_{i+1/2,j}\right) \\ &- \frac{\Delta t}{\Delta y}\left(B^{+}\Delta Q_{i,j-1/2} + B^{-}\Delta Q_{i,j+1/2}\right) \\ &- \frac{\Delta t}{\Delta x}\left(\tilde{F}_{i+1/2,j} - \tilde{F}_{i-1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\tilde{G}_{i,j+1/2} - \tilde{G}_{i,j-1/2}\right). \end{aligned} \tag{3.2.24}$$

A very similar process is used to compute solutions in 3D, but when solving in the x-direction, transverse fluctuations would need to be computed in both the y and z-direction. In addition, these transverse waves must be split into the final direction meaning that a 3D solver must incorporate a second transverse solve. For example, when solving in the x-direction, the y-direction transverse waves, along with being included in the y-direction correction terms, must be split into the z-direction. These transverse waves will then be included in the z-direction correction terms. All combinations must be accounted for to ensure stability of CFL ≤ 1. This leads to 12 transverse solves per interface. In a heterogeneous model, care must be taken to select the correct material

parameters for the first and second transverse solves. A description of this is included in Section 3.3 on the elasticity equations.

3.2.4. Nonlinear Systems. The general form for a nonlinear system of equations in one dimension is

$$q_t + f(q)_x = 0. \tag{3.2.25}$$

An approximate Riemann solver must be used in these cases and the one described here is the linearized Riemann solver. This consists of approximating the nonlinear system by a linear system at each grid cell boundary by using the Jacobian of f,

$$q_t + f'(q)\, q_x = 0. \tag{3.2.26}$$

The Jacobian is approximated by a matrix, $\hat{A}_{i-1/2}$, valid at the cell boundary between Q_{i-1} and Q_i. The approximate problem that is actually solved is

$$\hat{q}_t + \hat{A}_{i-1/2}\hat{q}_x = 0. \tag{3.2.27}$$

At this point the method works very similarly to the linear case, i.e. the solution differences at the cell boundaries are decomposed into the eigenvectors of the Jacobian and the wave speeds are the eigenvalues.

The entries in $\hat{A}_{i-1/2}$ must be found from the field variables in the adjoining cells. Simple arithmetic averaging, $(Q_i + Q_{i+1})/2$, does not ensure conservation, but Roe averaging will ensure that the method remains conservative [**70**]. An example of a Roe average is given in section 3.4.4. Another special consideration to take into account with

nonlinear systems is an entropy fix. This corrects for rarefaction waves that move partially into both of the adjoining cells. If this is not taken into account the solution may converge to an entropy violating solution.

3.3. Elasticity Equations

The linear elasticity equations are used to model longitudinal and shear waves in solids. Within the lithotripsy focusing application these equations are used to model the refraction of the acoustic waves through the lens, as well as the propagation of stress through the stone. In the fracture model these equations are solved on the continuum scale.

3.3.1. Derivation. The elasticity equations are formed from the three following sets of equations [**7, 104**]:

- The equations of motion:

$$\rho \frac{\partial \mathbf{u}}{\partial t} = \nabla \cdot \sigma, \tag{3.3.1}$$

  where $\mathbf{u}$ is the time derivative of the displacements, σ is the stress tensor, and ρ is the density.

- Equations relating strain and displacement:

$$\epsilon = \frac{1}{2}\left(\nabla\delta + (\nabla\delta)^T\right), \tag{3.3.2}$$

  where ϵ is the strain tensor and δ is the displacement vector.

- Constitutive equations relating stress and strain. For the case of linear elasticity, Hooke's law is used:

$$\epsilon = \frac{1+\nu}{E}\sigma - \frac{\nu}{E}Tr\left(\sigma\right)\mathbf{I}, \tag{3.3.3}$$

where $Tr(\sigma)$ is the trace of the stress tensor, $\mathbf{I}$ is the identity matrix, ν is Poisson's ratio, and E is Young's modulus.

3.3.2. Cylindrical Coordinates. In cylindrical coordinates (z, r, θ) we define the unit basis vectors as $\left(\mathbf{e}^z, \mathbf{e}^r, \mathbf{e}^\theta\right)$. From equations (3.3.2) we get the following strain-displacement equations, noting that the gradient becomes $\nabla = \left(\frac{\partial}{\partial z}, \frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial \theta}\right)$ in cylindrical coordinates [**7**]:

$$\begin{aligned}
\epsilon^{zz} &= \frac{\partial \delta^z}{\partial z} \\
\epsilon^{rr} &= \frac{\partial \delta^r}{\partial r} \\
\epsilon^{\theta\theta} &= \frac{1}{r}\left(\frac{\partial \delta^\theta}{\partial \theta} + \delta^r\right) \\
\epsilon^{zr} &= \frac{1}{2}\left(\frac{\partial \delta^z}{\partial r} + \frac{\partial \delta^r}{\partial z}\right) \\
\epsilon^{z\theta} &= \frac{1}{2}\left(\frac{1}{r}\frac{\partial \delta^z}{\partial \theta} + \frac{\partial \delta^\theta}{\partial z}\right) \\
\epsilon^{r\theta} &= \frac{1}{2}\left(\frac{1}{r}\frac{\partial \delta^r}{\partial \theta} + \frac{\partial \delta^\theta}{\partial r} - \frac{\delta^\theta}{r}\right),
\end{aligned} \tag{3.3.4}$$

where superscripts denote direction. The equations of motion (3.3.1), with the velocity vector $\mathbf{u} = u\mathbf{e}^z + v\mathbf{e}^r + w\mathbf{e}^\theta$, become

$$\begin{aligned} \rho u_t - \sigma_z^{zz} - \sigma_r^{zr} - \frac{1}{r}\sigma_\theta^{z\theta} &= \frac{1}{r}\sigma^{zr} \\ \rho v_t - \sigma_z^{zr} - \sigma_r^{rr} - \frac{1}{r}\sigma_\theta^{r\theta} &= \frac{1}{r}\left(\sigma^{rr} - \sigma^{\theta\theta}\right) \\ \rho w_t - \sigma_z^{z\theta} - \sigma_r^{r\theta} - \frac{1}{r}\sigma_\theta^{\theta\theta} &= \frac{2}{r}\sigma^{r\theta}, \end{aligned} \tag{3.3.5}$$

where subscripts denote differentiation. Hooke's law can be written as [**70**]

$$\begin{pmatrix} \sigma^{zz} \\ \sigma^{rr} \\ \sigma^{\theta\theta} \\ \sigma^{zr} \\ \sigma^{z\theta} \\ \sigma^{r\theta} \end{pmatrix} = \begin{pmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\mu \end{pmatrix} \begin{pmatrix} \epsilon^{zz} \\ \epsilon^{rr} \\ \epsilon^{\theta\theta} \\ \epsilon^{zr} \\ \epsilon^{z\theta} \\ \epsilon^{r\theta} \end{pmatrix}, \tag{3.3.6}$$

where λ and μ are the Lamé parameters. The Lamé parameters relate to Young's modulus, E, and Poisson's ratio, ν, by

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{3.3.7}$$

and

$$\mu = \frac{E}{2(1+\nu)}. \tag{3.3.8}$$

In order to write system (3.3.4) in terms of stress and velocity instead of strain and displacements we use the transformation (3.3.6) and take a time derivative. When combined with (3.3.5) this results in

$$
\begin{aligned}
\sigma_t^{zz} - (\lambda + 2\mu)\, u_z - \lambda v_r - \frac{\lambda}{r} w_\theta &= \frac{\lambda}{r} v \\
\sigma_t^{rr} - \lambda u_z - (\lambda + 2\mu)\, v_r - \frac{\lambda}{r} w_\theta &= \frac{\lambda}{r} v \\
\sigma_t^{\theta\theta} - \lambda u_z - \lambda v_r - \frac{(\lambda + 2\mu)}{r} w_\theta &= \frac{\lambda + 2\mu}{r} v \\
\sigma_t^{zr} - \mu v_z - \mu u_r &= 0 \\
\sigma_t^{z\theta} - \mu w_z - \frac{\mu}{r} u_\theta &= 0 \\
\sigma_t^{r\theta} - \mu w_r - \frac{\mu}{r} v_\theta &= -\frac{\mu}{r} w \\
\rho u_t - \sigma_z^{zz} - \sigma_r^{zr} - \frac{1}{r}\sigma_\theta^{z\theta} &= \frac{1}{r}\sigma^{zr} \\
\rho v_t - \sigma_z^{zr} - \sigma_r^{rr} - \frac{1}{r}\sigma_\theta^{r\theta} &= \frac{1}{r}\left(\sigma^{rr} - \sigma^{\theta\theta}\right) \\
\rho w_t - \sigma_z^{z\theta} - \sigma_r^{r\theta} - \frac{1}{r}\sigma_\theta^{\theta\theta} &= \frac{2}{r}\sigma^{r\theta}.
\end{aligned}
\tag{3.3.9}
$$

Equations (3.3.9) are the full 3D cylindrical elasticity equations, where ρ, λ, and μ are given properties of the material being modeled. The material parameters can be spatially dependent so that

$$
\begin{aligned}
\rho &= \rho\,(z, r, \theta) \\
\lambda &= \lambda\,(z, r, \theta) \\
\mu &= \mu\,(z, r, \theta)\,.
\end{aligned}
\tag{3.3.10}
$$

To obtain the 2D axisymmetric elasticity equations we assume there is no variation in the θ direction and drop all corresponding derivatives. The equations split into two independent systems:

$$
\begin{aligned}
\sigma_t^{zz} - (\lambda + 2\mu)\, u_z - \lambda v_r &= \frac{\lambda}{r} v \\
\sigma_t^{rr} - \lambda u_z - (\lambda + 2\mu)\, v_r &= \frac{\lambda}{r} v \\
\sigma_t^{\theta\theta} - \lambda u_z - \lambda v_r &= \frac{\lambda + 2\mu}{r} v \\
\sigma_t^{zr} - \mu v_z - \mu u_r &= 0 \\
u_t - \frac{1}{\rho}\sigma_z^{zz} - \frac{1}{\rho}\sigma_r^{zr} &= \frac{1}{\rho r}\sigma^{zr} \\
v_t - \frac{1}{\rho}\sigma_z^{zr} - \frac{1}{\rho}\sigma_r^{rr} &= \frac{1}{\rho r}\left(\sigma^{rr} - \sigma^{\theta\theta}\right)
\end{aligned}
\tag{3.3.11}
$$

and

$$
\begin{aligned}
\sigma_t^{z\theta} - \mu w_z &= 0 \\
\sigma_t^{r\theta} - \mu w_r &= -\frac{\mu}{r} w \\
w_t - \frac{1}{\rho}\sigma_z^{z\theta} - \frac{1}{\rho}\sigma_r^{r\theta} &= \frac{2}{\rho r}\sigma^{r\theta}.
\end{aligned}
\tag{3.3.12}
$$

The first system models longitudinal waves and shear waves with motion in the zr plane and the second models shear waves with motion orthogonal to the zr plane. In the lithotripsy focusing application only the first system is solved as the latter waves are not expected to occur in any appreciable way. Therefore, the remainder of this section is devoted only to system (3.3.11).

3.3.3. Wave Propagation Implementation of Axisymmetric Equations. System (3.3.11) can be written as

$$q_t + Aq_z + Bq_r = Cq, \tag{3.3.13}$$

where

$$q = \begin{pmatrix} \sigma^{zz} & \sigma^{rr} & \sigma^{\theta\theta} & \sigma^{zr} & u & v \end{pmatrix}^T, \tag{3.3.14}$$

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & -\lambda - 2\mu & 0 \\ 0 & 0 & 0 & 0 & -\lambda & 0 \\ 0 & 0 & 0 & 0 & -\lambda & 0 \\ 0 & 0 & 0 & 0 & 0 & -\mu \\ -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\rho} & 0 & 0 \end{pmatrix}, \tag{3.3.15}$$

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -\lambda \\ 0 & 0 & 0 & 0 & 0 & -\lambda - 2\mu \\ 0 & 0 & 0 & 0 & 0 & -\lambda \\ 0 & 0 & 0 & 0 & -\mu & 0 \\ 0 & 0 & 0 & -\frac{1}{\rho} & 0 & 0 \\ 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3.3.16}$$

and

$$C = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \frac{\lambda}{r} \\ 0 & 0 & 0 & 0 & 0 & \frac{\lambda}{r} \\ 0 & 0 & 0 & 0 & 0 & \frac{\lambda+2\mu}{r} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\rho r} & 0 & 0 \\ 0 & \frac{1}{\rho r} & -\frac{1}{\rho r} & 0 & 0 & 0 \end{pmatrix}. \tag{3.3.17}$$

The implementation for the z-direction and r-direction Riemann solver, using the A and B matrix, relatively, are similar and their differences only involve the transposition of several elements. Therefore, only the derivation in the z-direction is shown. As seen in Section 3.1, the eigenvalues and eigenvectors of A found from the eigendecomposition $A = R\Lambda R^{-1}$ are used in a Riemann solver implementation. The eigenvalues of A, which correspond to the wave speeds at the cell boundaries, are

$$\text{diag}(\Lambda) = (-c_p, c_p, -c_s, c_s, 0, 0)\,, \tag{3.3.18}$$

where

$$c_p = \sqrt{\frac{\lambda + 2\mu}{\rho}} \tag{3.3.19}$$

and

$$c_s = \sqrt{\frac{\mu}{\rho}} \tag{3.3.20}$$

are the longitudinal and shear wave speeds in the material. From (3.3.18) it can be seen that the system models four waves, right-going and left-going longitudinal waves and right-going and left-going shear waves, even though there are six equations. The eigenvector matrix of A, where the eigenvectors are the columns, is

$$R = \begin{pmatrix} \lambda+2\mu & \lambda+2\mu & 0 & 0 & 0 & 0 \\ \lambda & \lambda & 0 & 0 & 0 & 1 \\ \lambda & \lambda & 0 & 0 & 1 & 0 \\ 0 & 0 & \mu & \mu & 0 & 0 \\ c_p & -c_p & 0 & 0 & 0 & 0 \\ 0 & 0 & c_s & -c_s & 0 & 0 \end{pmatrix}. \tag{3.3.21}$$

The matrix B has the same eigenvalues as A and the eigenvector matrix of B is simply (3.3.21) with rows permuted.

To find the wave coefficients we first define the jumps between solution values at the cell boundary as the following vector

$$\delta = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{pmatrix} = \begin{pmatrix} (\sigma_{zz})_R - (\sigma_{zz})_L \\ (\sigma_{rr})_R - (\sigma_{rr})_L \\ (\sigma_{\theta\theta})_R - (\sigma_{\theta\theta})_L \\ (\sigma_{zr})_R - (\sigma_{zr})_L \\ u_R - u_L \\ v_R - v_L \end{pmatrix}, \tag{3.3.22}$$

where R and L stand for the solution value in the cell to the right and left of the boundary. If the material parameters (ρ, λ, and μ) have a spatial dependence, so that

$$\begin{aligned} \rho &= \rho(z,r) \\ \lambda &= \lambda(z,r) \\ \mu &= \mu(z,r) \end{aligned} \tag{3.3.23}$$

we rewrite (3.3.21). The eigenvectors corresponding to negative wave speeds will use parameters from the cell to the left of the boundary. Likewise, eigenvectors corresponding to positive wave speeds will use parameters from the cell to the right. This is denoted as

$$\tilde{R} = \begin{pmatrix} (\lambda+2\mu)_L & (\lambda+2\mu)_R & 0 & 0 & 0 & 0 \\ \lambda_L & \lambda_R & 0 & 0 & 0 & 1 \\ \lambda_L & \lambda_R & 0 & 0 & 1 & 0 \\ 0 & 0 & \mu_L & \mu_R & 0 & 0 \\ (c_p)_L & -(c_p)_R & 0 & 0 & 0 & 0 \\ 0 & 0 & (c_s)_L & -(c_s)_R & 0 & 0 \end{pmatrix}. \tag{3.3.24}$$

Now $\delta = \tilde{R}\mathbf{a}$ is solved for $\mathbf{a}$, which are the coefficients of the jump vector decomposition into the eigenvectors of A. For slightly less cumbersome notation let $b = \lambda + 2\mu$.

This results in

$$
\begin{aligned}
a_1 &= \frac{(c_p)_R\delta_1 + b_R\delta_5}{b_R(c_p)_L + b_L(c_p)_R} \\
a_2 &= \frac{(c_p)_L\delta_1 - b_L\delta_5}{b_R(c_p)_L + b_L(c_p)_R} \\
a_3 &= \frac{(c_s)_R\delta_4 + \mu_R\delta_6}{(c_s)_R\mu_L + (c_s)_L\mu_R} \\
a_4 &= \frac{(c_s)_L\delta_4 - \mu_L\delta_6}{(c_s)_R\mu_L + (c_s)_L\mu_R}.
\end{aligned}
\tag{3.3.25}
$$

The formulas for a_5 and a_6 are not included because those correspond to the non-propagating waves, and that hence need not be included in a numerical simulation.

In two dimensions, a transverse solve, as described in Section 3.2.3, is required to maintain the full stability of the method. A 2D solver can be run without transverse solves, but would have to be run at a lower time step, increasing numerical dissipation. Here, a brief description of how to select material parameters for the transverse solve is provided. The transverse solve will model portions of the solution that are transmitted into cells perpendicular to the direction of the normal solve. For example, if the normal solve is in the x-direction, the values at cell boundary $(i-1/2, j)$ will be split into left-going and right-going parts affecting cells $(i-1, j)$ and (i, j). Portions of the right-going wave may be transmitted into cells $(i, j+1)$ and $(i, j-1)$ and portions of the left-going into cells $(i-1, j+1)$ and $(i-1, j-1)$. Performing the same Riemann solve in the y-direction will give the correct transmitted values into these cells. In the transverse solve the right-going wave becomes the up-going wave and the left-going wave becomes the down-going wave.

For the right-going wave from the normal solve, the up-going wave's speed and wave values will come from the material parameters of cell $(i, j+1)$ and the parameters in the

wave coefficients will come from cells (i, j) and $(i, j+1)$. The down-going wave's speed and wave values come from cell $(i, j-1)$ and coefficient parameters from cells (i, j) and $(i, j-1)$. The same is done for the left-going part of the normal solve except with row $i-1$ instead of i. If the $j+1$, j, and $j-1$ cells are refered to as the up (U), middle (M), and down (D) cells then the wave coefficients in Equation (3.3.25) become

$$\begin{aligned} a_1 &= \frac{(c_p)_M \delta_1 + b_M \delta_5}{b_M (c_p)_D + b_D (c_p)_M} \\ a_2 &= \frac{(c_p)_M \delta_1 - b_M \delta_5}{b_U (c_p)_M + b_M (c_p)_U} \\ a_3 &= \frac{(c_s)_M \delta_4 + \mu_M \delta_6}{(c_s)_M \mu_D + (c_s)_D \mu_M} \\ a_4 &= \frac{(c_s)_M \delta_4 - \mu_M \delta_6}{(c_s)_U \mu_M + (c_s)_M \mu_U}, \end{aligned} \tag{3.3.26}$$

where, in this case, δ_i are components of $A^+ \Delta Q_{i-1/2,j}$ or $A^- \Delta Q_{i-1/2,j}$. The δ_i can still be considered cell differences, where the middle cell contains the normal solve fluxes and the up and down cells contain zero.

3.3.4. Cartesian Coordinates. The 3D elasticity equations in Cartesian coordinates are derived in the same way as the cylindrical equations, shown in 3.3.2, using the strain-displacement equations, the equations of motion, and Hooke's law. The complete 3D

elasticity equations in Cartesian coordinates are

$$\begin{aligned}
\sigma_t^{xx} - (\lambda + 2\mu)\, u_x - \lambda v_y - \lambda w_z &= 0 \\
\sigma_t^{yy} - \lambda u_x - (\lambda + 2\mu)\, v_y - \lambda w_z &= 0 \\
\sigma_t^{zz} - \lambda u_x - \lambda v_y - (\lambda + 2\mu)\, w_z &= 0 \\
\sigma_t^{xy} - \mu v_x - \mu u_y &= 0 \\
\sigma_t^{yz} - \mu w_y - \mu v_z &= 0 \\
\sigma_t^{xz} - \mu w_x - \mu u_z &= 0 \\
u_t - \frac{1}{\rho}\sigma_x^{xx} - \frac{1}{\rho}\sigma_y^{xy} - \frac{1}{\rho}\sigma_z^{xz} &= 0 \\
v_t - \frac{1}{\rho}\sigma_x^{xy} - \frac{1}{\rho}\sigma_y^{yy} - \frac{1}{\rho}\sigma_z^{yz} &= 0 \\
w_t - \frac{1}{\rho}\sigma_x^{xz} - \frac{1}{\rho}\sigma_y^{yz} - \frac{1}{\rho}\sigma_z^{zz} &= 0.
\end{aligned} \tag{3.3.27}$$

3.3.5. Wave Propagation Implementation of 3D Cartesian Equations. The system of equations (3.3.27) can be written as

$$q_t + Aq_x + Bq_y + Cq_z = 0, \tag{3.3.28}$$

where

$$q = \begin{pmatrix} \sigma^{xx} & \sigma^{yy} & \sigma^{zz} & \sigma^{xy} & \sigma^{yz} & \sigma^{xz} & u & v & w \end{pmatrix}^T. \tag{3.3.29}$$

As in section 3.3.3, the eigenvalues and eigenvectors of one of the system matrices will be presented as well as the wave coefficients. The system matrices are similar and the process for deriving these values is the same, so only matrix A will be presented, given

by

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -\lambda - 2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\lambda & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu \\ -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\rho} & 0 & 0 & 0 \end{pmatrix}. \tag{3.3.30}$$

The eigenvalues of A are

$$\operatorname{diag}(\Lambda) = (-c_p, c_p, -c_s, c_s, -c_s, c_s, 0, 0, 0), \tag{3.3.31}$$

where the sound speeds, c_p and c_s, are given by (3.3.19) and (3.3.20). The eigenvector matrix is

$$R = \begin{pmatrix} \lambda+2\mu & \lambda+2\mu & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda & \lambda & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \lambda & \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \mu & \mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & \mu & 0 & 0 & 0 \\ c_p & -c_p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_s & -c_s & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_s & -c_s & 0 & 0 & 0 \end{pmatrix}. \tag{3.3.32}$$

Modeling the variable coefficient equations requires the parameters in (3.3.32) to be drawn from either the left or right finite volume cell depending on the direction of the

wave. Much like the matrix in (3.3.24), this eigenvector matrix becomes

$$\tilde{R} = \begin{pmatrix} (\lambda+2\mu)_L & (\lambda+2\mu)_R & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_L & \lambda_R & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \lambda_L & \lambda_R & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \mu_L & \mu_R & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu_L & \mu_R & 0 & 0 & 0 \\ (c_p)_L & -(c_p)_R & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & (c_s)_L & -(c_s)_R & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (c_s)_L & -(c_s)_R & 0 & 0 & 0 \end{pmatrix}. \tag{3.3.33}$$

Finally, $\delta = \tilde{R}\mathbf{a}$ is solved for the wave coefficients $\mathbf{a}$ whose elements are given by

$$\begin{aligned} a_1 &= \frac{(c_p)_R\delta_1 + b_R\delta_7}{b_R(c_P)_L + b_L(c_p)_R} \\ a_2 &= \frac{(c_p)_L\delta_1 - b_L\delta_7}{b_R(c_P)_L + b_L(c_p)_R} \\ a_3 &= \frac{(c_s)_R\delta_4 + \mu_R\delta_8}{(c_s)_R\mu_L + (c_s)_L\mu_R} \\ a_4 &= \frac{(c_s)_L\delta_4 - \mu_L\delta_8}{(c_s)_R\mu_L + (c_s)_L\mu_R} \\ a_5 &= \frac{(c_s)_R\delta_6 + \mu_R\delta_9}{(c_s)_R\mu_L + (c_s)_L\mu_R} \\ a_6 &= \frac{(c_s)_L\delta_6 - \mu_L\delta_9}{(c_s)_R\mu_L + (c_s)_L\mu_R}, \end{aligned} \tag{3.3.34}$$

where $b = \lambda + 2\mu$ and

$$\delta = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \\ \delta_7 \\ \delta_8 \\ \delta_9 \end{pmatrix} = \begin{pmatrix} \sigma_R^{xx} - \sigma_L^{xx} \\ \sigma_R^{yy} - \sigma_L^{yy} \\ \sigma_R^{zz} - \sigma_L^{zz} \\ \sigma_R^{xy} - \sigma_L^{xy} \\ \sigma_R^{yz} - \sigma_L^{yz} \\ \sigma_R^{xz} - \sigma_L^{xz} \\ u_R - u_L \\ v_R - v_L \\ w_R - w_L \end{pmatrix}. \tag{3.3.35}$$

A 3D Riemann solver requires a secondary transverse solve to split the first transverse solutions into the 3rd direction. The selection of material parameters is essentially the same as in the first transverse solve shown in Section 3.3.3. For example, if the normal solve is in the x-direction at cell boundary $(i - 1/2, j, k)$ then the normal solve uses material parameters from cells $(i - 1, j, k)$ and (i, j, k). The first transverse solve of the right-going wave uses material parameters from cells $(i, j-1, k)$, (i, j, k), and $(i, j+1, k)$. The second transverse solve of the up-going wave found from the right-going wave uses material parameters from cells $(i, j+1, k-1)$, $(i, j+1, k)$, and $(i, j+1, k+1)$. From one cell boundary there are four primary transverse solves and eight secondary transverse solves. All must be computed to ensure full stability and material parameters are found in the same way as in the example.

3.4. Euler Equations

The Euler equations are used to model fluid motion and sound wave propagation in compressible inviscid fluids. They are derived from the compressible Navier-Stokes equations with viscosity terms removed. These equations are used in the lithotripsy application to model the shock formation in the water region after the lens. The shock forms because the speed of sound depends on pressure, which is a nonlinear effect.

3.4.1. General Form. The Euler equations consist of equations modeling conservation of mass, momentum, and energy. In general these are derived by computing the flux in and out of an infinitesimal volume cell. The conservation of mass (or continuity) equation is given by

$$\rho_t + \nabla \cdot (\rho \mathbf{u}) = 0. \tag{3.4.1}$$

The momentum equations are given by

$$(\rho \mathbf{u})_t + \nabla \cdot (\mathbf{u} \otimes (\rho \mathbf{u})) + \nabla p = 0, \tag{3.4.2}$$

where p is the pressure and $\otimes$ is the dyadic product. The energy equation is given by

$$(\rho E)_t + \nabla \cdot (\mathbf{u} (\rho E + p)) = 0. \tag{3.4.3}$$

To close this system equations of state must be incorporated, which relate the pressure to one or more of the conserved quantities.

3.4.2. Cylindrical Coordinates. As before, for cylindrical coordinates, let the unit basis vectors be $\left(\mathbf{e}^z, \mathbf{e}^r, \mathbf{e}^\theta\right)$ and the velocity vector be $\mathbf{u} = u\mathbf{e}^z + v\mathbf{e}^r + w\mathbf{e}^\theta$. So, in 3D cylindrical coordinates the Euler equations become

$$
\begin{aligned}
\rho_t + (\rho u)_z + (\rho v)_r + \frac{1}{r}(\rho w)_\theta &= -\frac{1}{r}(\rho v) \\
(\rho u)_t + \left(\rho u^2 + p\right)_z + (\rho uv)_r + \frac{1}{r}(\rho uw)_\theta &= -\frac{1}{r}(\rho uv) \\
(\rho v)_t + (\rho uv)_z + \left(\rho v^2 + p\right)_r + \frac{1}{r}(\rho vw)_\theta &= -\frac{1}{r}\left(\rho v^2 - \rho w^2\right) \\
(\rho w)_t + (\rho uw)_z + (\rho vw)_r + \frac{1}{r}\left(\rho w^2 + p\right)_\theta &= -\frac{2}{r}(\rho vw) \\
(\rho E)_t + (u(\rho E + p))_z + (v(\rho E + p))_r + \frac{1}{r}(w(\rho E + p))_\theta &= -\frac{1}{r}(v(\rho E + p)).
\end{aligned}
\tag{3.4.4}
$$

To form the axisymmetric equations, derivatives in the θ direction are dropped and it is also assumed there is no velocity in the θ direction, that is $w = 0$. This results in

$$
\begin{aligned}
\rho_t + (\rho u)_z + (\rho v)_r &= -\frac{1}{r}(\rho v) \\
(\rho u)_t + \left(\rho u^2 + p\right)_z + (\rho uv)_r &= -\frac{1}{r}(\rho uv) \\
(\rho v)_t + (\rho uv)_z + \left(\rho v^2 + p\right)_r &= -\frac{1}{r}\left(\rho v^2\right) \\
(\rho E)_t + (u(\rho E + p))_z + (v(\rho E + p))_r &= -\frac{1}{r}(v(\rho E + p)).
\end{aligned}
\tag{3.4.5}
$$

3.4.3. Equations of State. As mentioned earlier an equation of state (EOS) is needed to complete the system of equations given in Sections 3.4.1 and 3.4.2. In numerical code this equation is used to replace the pressure, p, with the variables being solved for.

3.4.3.1. *Ideal Gas Law.* A very commonly used EOS is the ideal gas law. For a derivation see for example [**70, 121**]. In forms applicable to numerical models it is typically written

as

$$\frac{p}{p_0} = \left(\frac{\rho}{\rho_0}\right)^{\gamma} \tag{3.4.6}$$

or

$$p = (\gamma - 1)\,\rho\left(E + \frac{1}{2}\left(u^2 + v^2\right)\right). \tag{3.4.7}$$

The value of γ, the adiabatic index, typically used for air at room temperature is 1.4 [**121**].

3.4.3.2. *Modified Tait Equation of State.* When modeling compressible liquids, for example the water in the lithotripsy application, a different EOS is needed. A common EOS used for compressible water is the modified Tait EOS. In the following forms it is equivalent to the Tammann (or stiffened) EOS:

$$\frac{p + B}{p_0 + B} = \left(\frac{\rho}{\rho_0}\right)^{\gamma}, \tag{3.4.8}$$

$$p = (\gamma - 1)\,\rho\left(E + \frac{1}{2}\left(u^2 + v^2\right)\right) - B. \tag{3.4.9}$$

By comparing equations (3.4.8) and (3.4.9) with (3.4.6) and (3.4.7) it's seen that the modified Tait equation is simply a translation of the ideal gas law. In general terms, the EOS assumes water behaves as an ideal gas already subjected to B pressure. Values for γ and B typically used for water are 7.0 and 300 MPa [**121**].

3.4.4. Wave Propagation Implementation. The Riemann solver implementation will be derived using the ideal gas law. Because the modified Tait EOS is a constant

translation by B of the ideal gas law, the Riemann solver with the ideal gas law can be used to model compressible liquids requiring the modified Tait EOS, as long as the values are initialized using the the modified Tait EOS.

System (3.4.5) can be written as

$$q_t + A(q)_z + B(q)_r = C(q), \tag{3.4.10}$$

where

$$q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix}, \tag{3.4.11}$$

$$A(q) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho E + p) \end{pmatrix} = \begin{pmatrix} q_2 \\ q_2^2/q_1 + p \\ q_2 q_3/q_1 \\ q_2/q_1 (q_4 + p) \end{pmatrix}, \tag{3.4.12}$$

$$B(q) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \end{pmatrix} = \begin{pmatrix} q_3 \\ q_2 q_3/q_1 \\ q_3^2/q_1 + p \\ q_3/q_1 (q_4 + p) \end{pmatrix}, \tag{3.4.13}$$

$$C(q) = -\frac{1}{r}\begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 \\ v(\rho E + p) \end{pmatrix}, \tag{3.4.14}$$

and p is given by (3.4.7) or, if written in terms of the subscripted q's, by

$$p = (\gamma - 1)\left(q_4 + \frac{1}{2q_1}\left(q_2^2 + q_3^2\right)\right). \tag{3.4.15}$$

The vector functions A and B are given in terms of q_i to help illustrate the computation of the Jacobians.

To linearize the system the Jacobians of A and B are needed. Only the derivation of the Jacobian of A and its eigensystem will be shown. The computations for B are very similar. The Jacobian of A is

$$A'(q) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2}(\gamma-3)u^2 + \frac{1}{2}\tilde{\gamma}v^2 & (3-\gamma)u & -\tilde{\gamma}v & \tilde{\gamma} \\ -uv & v & u & 0 \\ -\gamma uE + \tilde{\gamma}(u^3 + uv^2) & \gamma E - \frac{1}{2}\tilde{\gamma}(3u^2 + v^2) & -\tilde{\gamma}uv & \gamma u \end{pmatrix}, \tag{3.4.16}$$

where $\tilde{\gamma} = \gamma - 1$ is used to simplify the matrix.

The eigenavalues of (3.4.16), and the wave speeds at the cell boundaries, are

$$\text{diag}(\Lambda) = (u - c, u, u, u + c), \tag{3.4.17}$$

where $c = \sqrt{\frac{\gamma p}{\rho}}$ is the speed of sound in the fluid. These wave speeds make intuitive sense as one would expect information to travel in a fluid at the fluid velocity, u, or the fluid velocity plus or minus the speed of sound. The eigenvector matrix is given by

$$R = \begin{pmatrix} 1 & 0 & 1 & 1 \\ u-c & 0 & u & u+c \\ v & 1 & v & v \\ H-uc & v & \frac{1}{2}(u^2+v^2) & H+uc \end{pmatrix}, \tag{3.4.18}$$

where $H = E + \frac{p}{\rho}$ and is called the total specific enthalpy.

Since this Riemann solver uses a linearization of a nonlinear system Roe averages should be used to ensure that the method remains conservative. The following averages are used in equations (3.4.17) and (3.4.18) when solutions are computed at each cell interface:

$$\hat{u} = \frac{\sqrt{\rho_{i-1}}u_{i-1} + \sqrt{\rho_i}u_i}{\sqrt{\rho_{i-1}} + \sqrt{\rho_i}}, \tag{3.4.19}$$

$$\hat{v} = \frac{\sqrt{\rho_{i-1}}v_{i-1} + \sqrt{\rho_i}v_i}{\sqrt{\rho_{i-1}} + \sqrt{\rho_i}}, \tag{3.4.20}$$

$$\hat{H} = \frac{\left(\sqrt{\rho_{i-1}}E_{i-1} + p_{i-1}/\rho_{i-1}\right) + \left(\sqrt{\rho_i}E_i + p_i/\rho_i\right)}{\sqrt{\rho_{i-1}} + \sqrt{\rho_i}}, \tag{3.4.21}$$

and

$$\hat{c} = \sqrt{(\gamma-1)\left(\hat{H} - \frac{1}{2}\left(\hat{u}^2 + \hat{v}^2\right)\right)}. \tag{3.4.22}$$

The coefficients that result from the decomposition of the jump δ into the eigenvectors of R, solving $\delta = R\mathbf{a}$, are

$$\begin{aligned} a_3 &= \frac{\gamma - 1}{\hat{c}^2}\left(\hat{H} - \left(\hat{u}^2 + \hat{v}^2\right)\delta_1 + \hat{u}\delta_2 + \hat{v}\delta_3 - \delta_4\right) \\ a_2 &= \delta_3 - \hat{v}\delta_1 \\ a_4 &= \frac{\delta_2 + (\hat{c} - \hat{u})\,\delta_1 - \hat{c}a_3}{2\hat{c}} \\ a_1 &= \delta_1 - a_3 - a_4, \end{aligned} \tag{3.4.23}$$

where

$$\delta = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{pmatrix} = \begin{pmatrix} \rho_R - \rho_L \\ (\rho u)_R - (\rho u)_L \\ (\rho v)_R - (\rho v)_L \\ (\rho E)_R - (\rho E)_L \end{pmatrix}. \tag{3.4.24}$$

3.5. Modifications to Current Implementations

This section discusses several modifications to currently available implementations of CLAWPACK and BEARCLAW. The first is described in work by Lemoine and Ou [**67**] and concerns the transverse solves in heterogeneous materials. Although this was developed by others, it is discussed in some detail here since it was imperative in achieving a working 3D elasticity solver. The second modification is a slight change in implementation which reduces the total number of transverse solves. The final modification was implemented for the focusing model, described in Chapter 5. This modification provides a new framework within BEARCLAW that allows for solving two sets of equations within

arbitrary subsets of the full computational domain. A brief explanation is provided here, while the full discussion is left for Chapter 5.

When modeling heterogeneous systems, as currently implemented in CLAWPACK and BEARCLAW, the transverse solves are only approximations when solving in heterogeneous portions of the domain, such as near material boundaries. For the following description Figure 3.1 provides reference. Let boundary $(i-1/2, j)$ be an interface for a normal solve in 2D, so that cell (i, j) contains the right-going $A^+\Delta Q$. The transverse solver correctly solves the transmission of $A^+\Delta Q$ into cells $(i, j+1)$ and $(i, j-1)$ and subtracts these values from the cell (i, j). This is valid away from heterogeneities. Near heterogeneities, however, for a completely valid solution, a full Riemann solve is needed at both interface $(i, j-1/2)$ and $(i, j+1/2)$ to correctly account for the reflection back into cell (i, j). In 2D this would add two additional transverse solves for each interface, from 2 to 4. This can be potentially reduced, however, by combining nearby solves. For example, both cells (i, j) and $(i, j-1)$ have an $A^+\Delta Q$ term from the normal solves at boundaries $(i-1/2, j)$ and $(i-1/2, j-1)$, respectively. These could be used as the left and right states for the transverse solve at interface $(i, j-1/2)$. Unfortunately, implementation of this method to reduce the total number of solves is not straightforward in the current CLAWPACK or BEARCLAW frameworks.

For small differences in wave speeds the approximation does not cause instabilities. Also, the shock forming nonlinear systems, that Riemann solvers are often used for, do not typically have strong heterogeneities. For these reasons the full Riemann solves accounting for transverse reflections are often not included in software such as CLAWPACK and BEARCLAW. Unfortunately, in 3D the ratio between material parameters which can cause instabilities shrinks significantly. A quick numerical experiment showed

that for acoustics in 2D, a ratio of wave speeds of approximately 10:1 was needed before instabilities developed. In 3D this ratio dropped to near 2:1.

To correctly account for all reflections in 3D over twice as many solves would be needed for each interface, as reflections of reflections would need to be calculated. The total number of transverse solves increases from 12 to 40. As stated before, combining transverse solves across interfaces would reduce work but is not straightforward in the current framework. Future work may include reworking aspects of BEARCLAW to allow for more efficient 3D calculations which include full transverse solves. Fortunately, not all reflections need to be included to decrease the error seen at material boundaries. For the kidney stone fracture model, described in Chapter 6, BEARCLAW was modified to include reflection solves to increase the stability limit.

The second modification involves combining flux terms before computing the resulting transverse fluctuations. For example, as illustrated in Figure 3.1, cell (i, j) is affected by the normal solve at boundary $(i - 1/2, j)$ and the resulting flux, $A^+\Delta Q_{i-1/2}$. This cell is also affected by the normal solve at boundary $(i + 1/2, j)$ and the resulting flux, $A^-\Delta Q_{i+1/2}$. Current implementations of CLAWPACK and BEARCLAW then apply the transverse solve to each of the terms individually. Letting

$$A^s\Delta Q_{i-1/2} = A^+\Delta Q_{i-1/2} + A^-\Delta Q_{i+1/2}, \tag{3.5.1}$$

the transverse solves can be applied to $A^s\Delta Q_{i-1/2}$ alone. For linear systems, this results in exactly the same solution. This results in the number of transverse solves per interface being halved.

In 2D, where there are only 2 transverse solves per interface, the resulting speedup in a sample problem was about 6%. In 3D this increased to about 19%, as the 12 transverse solves per interface become a more significant portion of the work. In a 3D implementation which includes the reflections from the transverse solves, so 40 transverse solves per interface, the speedup is closer to about 30%. To reiterate, this is not a trade-off due to an approximation. For linear systems, the solution is identical, as the speedup comes solely from a rearrangement of the order of operations.

In 3D with the full transverse solves, this can also be applied to the reflections of the first transverse solve. For example, let $A^+\Delta Q$ be the flux affecting cell (i, j, k) from the normal solve at boundary $(i - 1/2, j, k)$. The first transverse solves will split this term in both the y and z directions. The y transverse solve will create terms $B^-A^+\Delta Q$ and $B^+A^+\Delta Q$ which affect cells $(i, j - 1, k)$ and $(i, j + 1, k)$, respectively. When including the full transverse solves, two additional terms are created as reflections which affect cell (i, j, k). Let these be called $R^-A^+\Delta Q$ and $R^-A^+\Delta Q$. At this point, the second transverse solves need to be applied to these four terms, but since the two reflection terms affect the same cell these can be combined into one term. When combining both the normal solves and the first transverse reflections, the number of transverse solves per interface (when including all full transverse solves) is reduced from 40 to 16.

The third modification was developed for the focusing model described in Chapter 5, and is described in more detail there. Essentially, a framework was incorporated into BEARCLAW which allows two sets of equations to be solved simultaneously within arbitrary subsets of the full computational domain. Two root level grids both cover the entirety of the domain, one for each equation set. An auxiliary variable defines which portions of the domain belong to which equation set. Solution values are either kept or

replaced with solution values from the other grid depending on which subset they fall into. The user defines the subsets and functions to convert solution values between the two equation sets. This method works seamlessly with AMR.

3.6. Verification of Method

In this section, the 3D solvers for acoustics and elasticity, which were added to the BEARCLAW framework for this work, are verified. First the solver is shown to accurately predict reflection and transmission coefficients at oblique material interfaces. The materials for this test do not support shear waves. Secondly, a convergence test is performed on the the full elasticity solver with shear waves. These tests were performed on solvers which include the full transverse solves and flux combinations described in the previous section.

The setup to test transmission and reflection coefficients includes two materials with an oblique plane as the boundary between them and an initial pressure pulse in one of the regions. The materials do not support shear waves and so the equations essentially revert to acoustics. The domain is $1 \times 1 \times 1$ where each coordinate runs from 0.0 to 1.0. The initial pulse is a wide step function, with pressure set to 2.0 within $0.05 < x < 0.2$. Elsewhere in the domain, pressure is set to 0.0. Within an elasticity solver, setting the pressure actually means setting each of the three normal stresses. Likewise, when computing pressure later in the computation, the pressure is given by the average of the three normal stresses.

The initial pulse splits into left and right-going halves with amplitude 1.0. Therefore, the amplitudes of the reflected and transmitted pulses are exactly the values of the reflection and transmission coefficients. The boundary between the two materials can

be set flat or to any arbitrary angle. For these tests the boundary is rotated in 3D (over two angles), so that the full 3D aspects of the solver are tested. If the boundary was only rotated in 2D then the second transverse solve would never be tested. Also, for calculating the coefficients, the two angle rotation is still equivalent to a one angle relation between the pulse and boundary. Since the initial pulse is flat and perpendicular to the x-axis, then the normal of the oblique boundary can be rotated around the x-axis until a single angle is found between the pulse and the boundary. This single angle is used to compare coefficients even though the simulation remains fully 3D.

The density, ρ, is set to 1.0 for both materials, so the p-wave speed equals the acoustic impedance, $c_i = z_i$, where $i = 1, 2$. The pulse begins in material 1 and moves toward material 2. The formula for the reflection and transmission coefficients of an acoustic pulse encountering a boundary with angle, θ, are

$$R = \frac{z_2 \sec\left(\arcsin\left(\frac{c_2 \sin\theta}{c_1}\right)\right) - z_1 \sec\theta}{z_2 \sec\left(\arcsin\left(\frac{c_2 \sin\theta}{c_1}\right)\right) + z_1 \sec\theta} \tag{3.6.1}$$

and

$$T = \frac{2 \sec\left(\arcsin\left(\frac{c_2 \sin\theta}{c_1}\right)\right)}{z_1 \sec\theta + \sec\left(\arcsin\left(\frac{c_2 \sin\theta}{c_1}\right)\right)}. \tag{3.6.2}$$

The coefficients are also related by $T = R + 1$, which is derived from continuity at the boundary [**34**]. Four cases were tested by comparing plots of Equations (3.6.1) and 3.6.2 against values found from the computation. Results of these tests are shown in Figure 3.2. The first two cases test different impedance ratios for a flat and oblique boundary. The second two cases test different boundary angles for set impedance ratios.

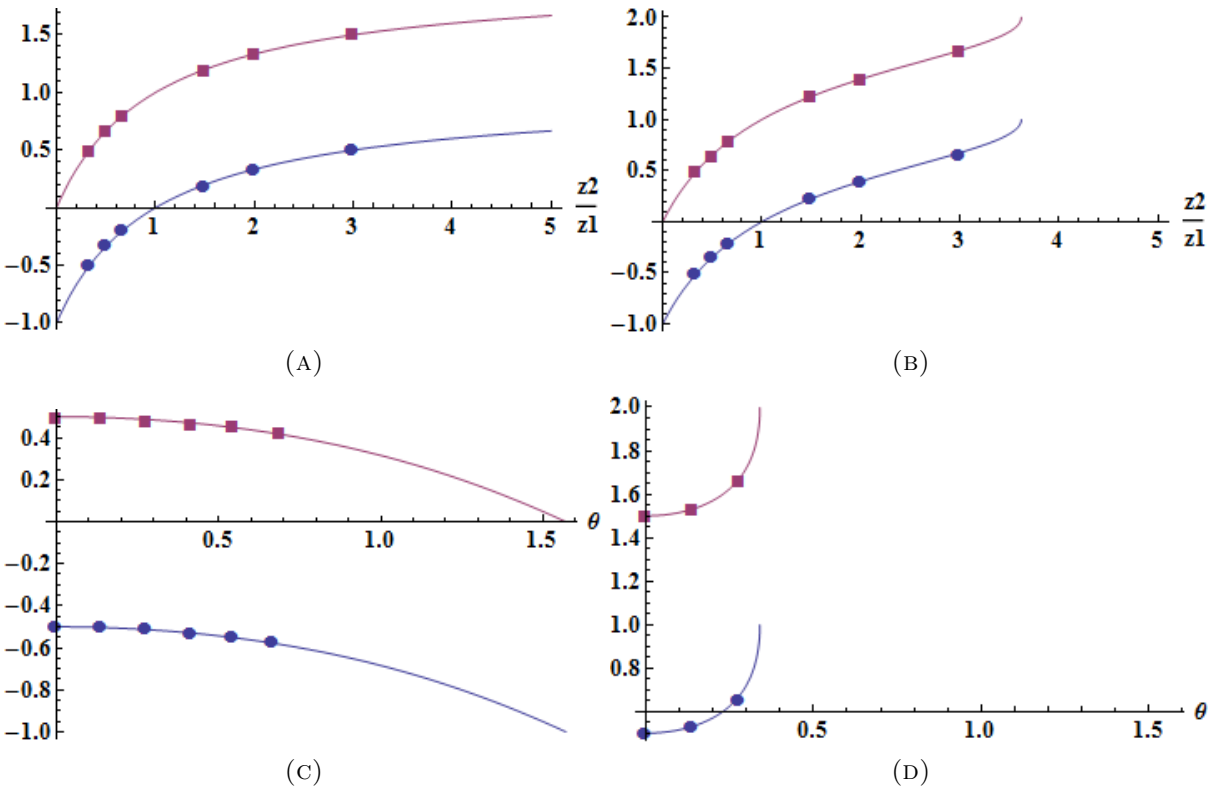


FIGURE 3.2. Plots of analytically and numerically computed reflection and transmission coefficients for an acoustic pulse encountering an oblique material interface. Red corresponds to transmission coefficients and blue corresponds to reflection coefficients. Points are numerical results and lines are plots of the analytic equations. (A) Coefficients versus impedance ratios for a flat boundary, $\theta = 0$. (B) Coefficients versus impedance ratios for an oblique boundary, $\theta = 0.28$ radians. (C) Coefficients versus angles for $z2/z1 = 1/3$. (D) Coefficients versus angles for $z2/z1 = 3/1$.

The simulations were run with CFL= 0.99 on $90 \times 90 \times 90$ grids or sometimes $120 \times 120 \times 120$ grids for higher impedance ratios. These grid sizes were used to keep runtime relatively low, but these sizes still led to considerable numerical diffusion for higher impedance ratios. Also, material averaging was not used at the boundary for these tests. For these reasons, finding a single value for the amplitude of the reflected and transmitted pulses was not straightforward. To compute the numerical values, histograms of slices of the data were created once the pulse had completely reflected and transmitted. Values at

spikes in the histogram data were then used as the reflection and transmission coefficients. As seen in Figure 3.2 the model shows excellent agreement with the analytic coefficients.

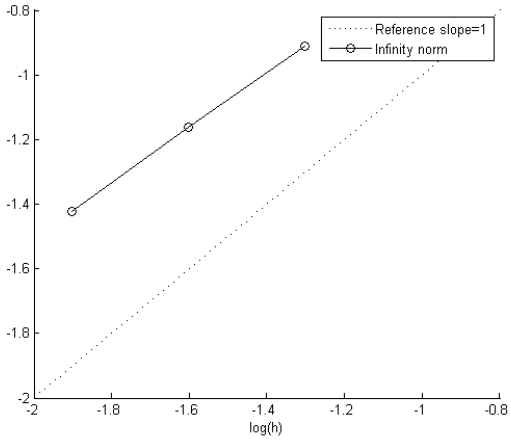


FIGURE 3.3. Convergence plot of the 3D elasticity solver implemented in a finite volume Riemann solver context in BEARCLAW.

Convergence of the full 3D elasticity solver with support for shear waves is also tested. In this problem an initial pulse in a Gaussian shape encounters a rectangular inclusion in the domain. The rectangle is fully 3D and its material parameters are $c_p = 4$, $c_s = 2$, and $\rho = 1$. The parameters outside the inclusion are $c_p = 2$, $c_s = 1$, and $\rho = 1$. The domain size is $1 \times 1 \times 1$ and discretized by n finite volume cells in each direction. For the convergence test, simulations were run with n =20, 40, 80, and 160. Figure 3.3 shows the results of this test which is first-order convergence. As explained by LeVeque [**69**], the second-order corrections remove the first order diffusive error, which causes the rampant diffusion of a method like upwind. The corrections leave a first order error term in the acoustic speeds, which leads to small shifts in the solution and formal first-order convergence. The slope in the convergence plot is slightly less than one, at 0.86.

CHAPTER 4

Experimental Methods

This chapter describes the experimental methods used to collect data for incorporation into and validation of the focusing and fracture models presented in later chapters. In Section 4.3 some purely experimental results of kidney stone simulant fracture using micro-CT scans are shown.

4.1. Experimental Data Acquisition - Focusing Model

The development and experimental validation of the focusing model, to be described in Chapter 5, is a collaborative effort with Nathan Smith, Georgy Sankin, Walter N. Simmons, and Pei Zhong at Duke University. The pressure profiles used to verify the model are recorded within an experimental setup used to test lens designs. The processes in an experimental refracting EM lithotripter can be segmented into a few distinct stages. First is the creation of the acoustic pulse by the electromagnetic actuator. After traveling through a small portion of water the acoustic pulse enters the lens and refracts. Upon exiting the lens the pulse has been directed towards the geometrical focus of the lens. As the pulse proceeds through the water and converges towards the focus the amplitude increases. Eventually, due to nonlinear effects, the pressures are high enough to cause significant steepening of the pulse and finally shock wave formation.

The essentials of the setup include the electromagnetic pulse generator and polystyrene lens which are submerged in a 40 x 30 x 30 cm Lucite tank filled with 0.2 micrometer-filtered and degassed water (<3 mg/L Oxygen concentration, 23°C). The acoustic lens fits directly on top of the actuator with a small fraction of water in between. The lenses are made from polystyrene and its material properties are given in Section 5.2.

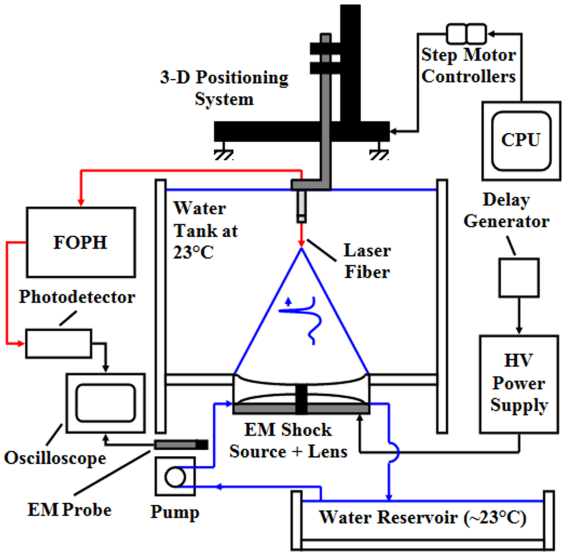


FIGURE 4.1. Diagram of the experimental setup with the tank, actuator, and lens in the center. Red arrows show the FOPH setup and blue arrows show the flow of water to fill the space behind the lens. Also shown is the 3-D positioning system used to position the fiber optic probe hydrophone for pressure measurement.

An FOPH 500 fiber optical hydrophone by RP Acoustics (Leutenbach, Germany) is used to record the pressure data at certain points over time. An optical hydrophone works by pulsing a laser through a fiber optic strand placed in the water. Pressure differences in the water will cause slight variations in the density which causes the reflection coefficients between the fiber optic strand and the water to change. The photovoltage signal is sampled at 100 MHz by a LeCroy oscilloscope. Figure 2.2 shows an example plot of this type of data. Other parameters of the pulse can be derived from this data including beam width, peak pressures, and acoustic energy. Figure 4.1 shows a diagram

of the experimental setup. Dimensions in the experiment and computation are given in cylindrical coordinates (z, r, θ), where $r = 0$ is the center axis of the actuator and lens and $z = 0$ is the surface of the actuator. The actuator extends from $r = 15$ to $r = 70$ mm, the lens extends to $r = 72$ mm, and the geometrical focus of the lens is at $z = 181.8$ mm.

A similar experimental setup is used to create the pulse input for the focusing model. This input consists of pressure data as a function of the radial coordinate (r), time (t), and the source voltage (V). This data corresponds to the direct wave created by the actuator. In this experiment the lens is removed and the optical fiber of the hydrophone is placed close to the actuator at $z \approx 5$ mm.

To create the input pressure data three source voltages (12.8, 15.8, and 18.8 kV) were used. The radial profile of the pulse was characterized by FOPH pressure measurements at $\Delta r = 5$ mm steps over the interval $25 \leq r \leq 60$ mm. Near the edges of the actuator where the profile changes more rapidly, $15 \leq r \leq 25$ mm and $60 \leq r \leq 70$ mm, a smaller step size of $\Delta r = 2.5$ mm was used. Elsewhere, $r \leq 15$ mm and $r \geq 70$ mm, the incoming pressure is assumed to be zero. This data was curve fitted as functions of r, t, and V in order to interpolate and extrapolate input pressure data over these variables. The functions are presented in Section 5.3.

4.1.0.1. *Post-processing of data.* The hydrophone supplies data as pressure over time at certain spatial coordinates. All other parameters presented in the results section of the focusing model (Section 5.3), such as beam width and pulse durations, are derived from these pressure profiles. The experimental pressure profiles are averages over four runs which increases the signal-to-noise ratio sufficiently for comparison to the model.

The hydrophone measurements are post-processed using MATLAB and the lithotripter field parameters are calculated following the IEC standard 61846. The compressive and tensile pulse durations, t_+ t_-, respectively, are calculated based on the first and last point where 10% of the peak pressure of that portion of the wave is encountered. The rise time, t_r, is calculated as the time from when the leading compressive wave increases from 10% to 90% of the peak pressure. Beam width, BW, is calculated as the diameter of the circle in the focal plane, perpendicular to the propagation axis, defined by where the pressure is 50% of the peak pressure of the leading compressive wave.

The effective acoustic pulse energies are calculated by approximating the integral

$$E_{\mathrm{Eff}} = 2\pi \int_0^{R_h} \mathrm{PII}\,(r)\, r\, dr \tag{4.1.1}$$

where PII is the pulse intensity integral. This integral is also evaluated approximately and is given by

$$\mathrm{PII}\,(r) = \frac{1}{Z_0} \int_{t_1}^{t_2} P\,(z,r,t)^2\; dt \tag{4.1.2}$$

where $P(z,r,t)$ is pressure, Z_0 is the acoustic impedance of water, and t_1 and t_2 are the first and final crossing points of 10% of the peak pressure of the region in question. Here, acoustic energies are calculated only in the geometric focal plane of the lens so that $z = 181.8$ mm. Numerical data is produced in the same format (pressure over time at certain (z,r) coordinates) and therefore the same post-processing of parameters is used.

4.2. Experimental Data Acquisition - Fracture Model

The computational fracture model, to be described in Chapter 6, is verified by comparing to micro-CT (μCT) images of kidney stone simulants that have been subjected to lithotripter shock waves. Statistics are also derived from the images and are used to inform the model. The shocking of stones is a collaborative effort with Jacyln Lautz and Pei Zhong at Duke University. In addition, the model is verified by comparing the number of shocks required for initial fracture to experimental results by Jaclyn Lautz.

The kidney stone simulants are made of BegoStone, a hard plaster originally developed for dental applications. BegoStone has been shown to have similar mechanical properties to calcium oxalate monohydrate kidney stones [**76**]. The Begostone powder is mixed with water and setting takes about 10 minutes. The Begostone attains its full hardness after about 24 hours. The stones used in this work are created from a 15:3 powder-to-water ratio. Material properties resulting from this ratio can be found in [**35**] and are shown in Table 4.1.

TABLE 4.1. Experimentally found material properties of a 15:3 powder-to-water ratio BegoStone simulant [**35**].

		15:3 BegoStone
Density	(kg/m^3)	1995 ± 18
P-wave speed	(m/s)	4159 ± 114
S-wave speed	(m/s)	2319 ± 34
Tensile Faliure Strength (Dry)	(MPa)	16.3 ± 1.8
Tensile Faliure Strength (Wet)	(MPa)	7.12 ± 0.27

4.2.1. Number of Shocks Required for Fracture. This section presents results by Jaclyn Lautz regarding the number of shocks required for initial fracture of a 7 mm cylindrical BegoStone kidney stone simulant. The fracture model is both informed and verified by this data. The data compares the peak positive pressure, P_+, of the pulse and

the number of shocks required for fracture. The P_+ data is derived from a combination of several input voltages as well as translations along the propagation axis. While there are differences between a pulse from a higher input voltage shifted along the axis and a lower input voltage at the focus, these differences are minor and considered negligible for this study.

The data contains values for both water and butanedial as the fluid surrounding the stone. Butanediol is a viscous fluid with similar sound speeds to water. The greater viscosity, with respect to water, reduces cavitation. As seen in the measurements this increases the number of shocks required for initial fracture. This is due to the lack of surface damage to the stone from bubble collapse. Since the numerical model does not incorporate cavitation or cavitation effects at this point, comparisons are made to the butanediol results.

Figure 4.2 shows the results of Jaclyn Lautz's study with power fits. As seen in the plot, the number of shocks required clearly decreases with increased P_+. A few conclusions can be drawn from this data. For lower P_+, cavitation plays a greater role in initial fracture than stresses in the stone. As P_+ increases, the number of shocks for water and butanediol begin to converge as the effect from stresses in the stone begins to increase. Finally, for large P_+, the cavitation effects become essentially negligible with respect to initial fracture. For values of P_+ above about 55 MPa the number of shocks required is about the same for water and butanediol.

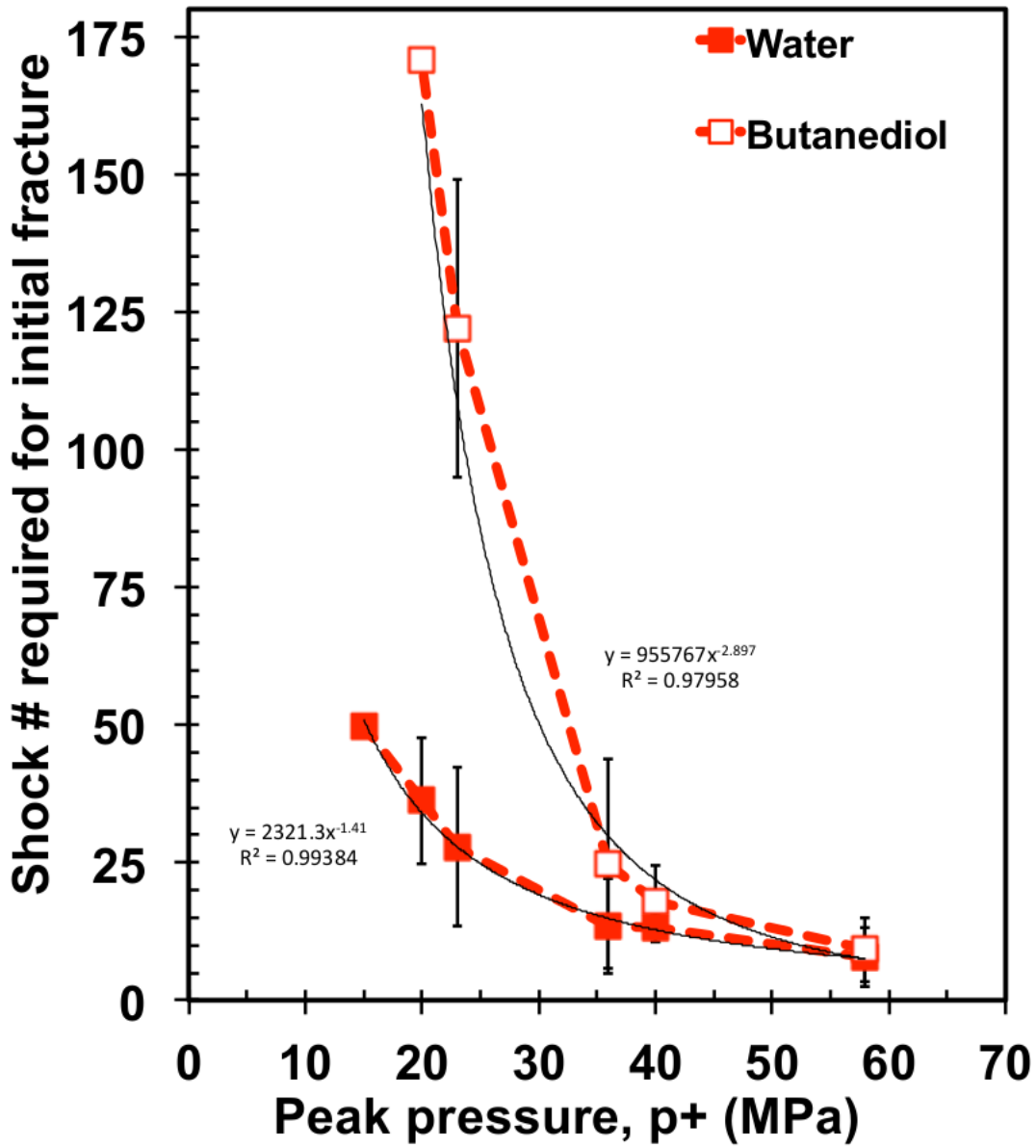


FIGURE 4.2. Plot showing the number of shocks required for initial fracture of a 7 mm cylidrical 15:3 BegoStone kidney stone simulant versus the peak positive pressure of the shock. Results for both water and butanediol as the transmission medium are included. Power fits for both sets of data are also shown. (Results recorded and plot constructed by Jaclyn Lautz)

Table 4.2 shows some values collected using the power fits from Figure 4.2 to be used later when verifying the fracture model. The table lists input voltage, corresponding P_+ from experiment, and the average number of shocks required for initial fracture in both water and butanediol using the power fits. Extrapolation of the power fit leads to

fewer shocks for butanediol at 18 kV than for water, but this is assumed to be incorrect. As mentioned, above 55 MPa, or 16 kV input, it is assumed that butanediol and water produce equivalent results. Therefore, 13, 8, and 7 shocks to 14, 16, and 18 kV input, respectively, will be used to compare model and experiment.

TABLE 4.2. Number of shocks required for initial fracture of a 7 mm cylindrical BegoStone kidney stone simulant over several input energies. Number of shock data for both water and butanediol is found from power fits of experimental data.

Input Voltage (kV)	P_+ (MPa)	Water	Butanediol
14	48	9.8	12.8
16	58	7.7	7.8
18	61	7.1	6.4

4.2.2. Micro-CT Images of Kidney Stone Simulants. These stones are shocked in a similar experimental setup as described for collecting FOPH data, with essentially just the FOPH replaced with the stone holder and stone. The stone is placed at the focus on a thin film within a cylindrical holder located above the lens and actuator. The stone is repositioned between shocks if knocked out of place. 14 kV shocks were used for all shocks in these experiments.

The μCT images were created at the Biomedical Research Imaging Center (BRIC) at the University of North Carolina. The stones were scanned at a resolution of 6 microns (cylindrical stones) and 8 microns (spherical stone). First, an untreated stone would be scanned, i.e. one subjected to zero shocks, in order to have an initial condition that could be used in the computational model. The initial voids and defects in the stone are presumably where cracks would tend to initialize due to the high stresses that typically surround these areas. For true comparison to experiment it is then imperative to include these initial features in the model. After the initial scan the stone would be subjected to

a number of lithotripter shock waves and then rescanned. This process of shocking and scanning would be continued until the stone breaks.

Due to the expense of the imaging, only four stones were put through the imaging and shocking procedure. Three stones were cylindrical and one was spherical. Two of the cylindrical stones had a diameter of 5 mm and a height of 7 mm and the other had a diameter of 9 mm and height of 10 mm. The 7 mm stones have a rounded cap on the distal end with a height of about 0.8 mm in addition to the 7 mm. The spherical stone had a diameter of 10 mm. This stone was scanned at the slightly lower resolution of 8 microns because a ridge on its surface would not allow it to fit in the 6 micron resolution holder which is 10 mm across.

Table 4.3 shows the shock numbers at which each stone was imaged and highlight the image sets with clearly visible cracks. The first 7 mm cylindrical stone broke at 14 shocks. The other stones were not shocked after their final imaging. 7 mm stone 1 and 10 mm stone 1 both developed cracks in the expected location within the stone, approximately one third of the distance from the distal end of the stone, whereas 7 mm stone 2 developed cracks towards the center of the stone. It is fairly consistent in experiments to see the one third distance fracture, so 7 mm stone 2 is somewhat of an outlier. This stone is still included in the analysis and it is assumed that the fracture parameters derived from the imaging are just as valid as the other stones.

Figure 4.3 shows an example 3D rendering of a spherical stone from MicroCT images. This stone was imaged as a preliminary test of whether the MicroCT scans would show the areas and values of interest and is not the spherical stone subjected to the shocking and imaging procedure.

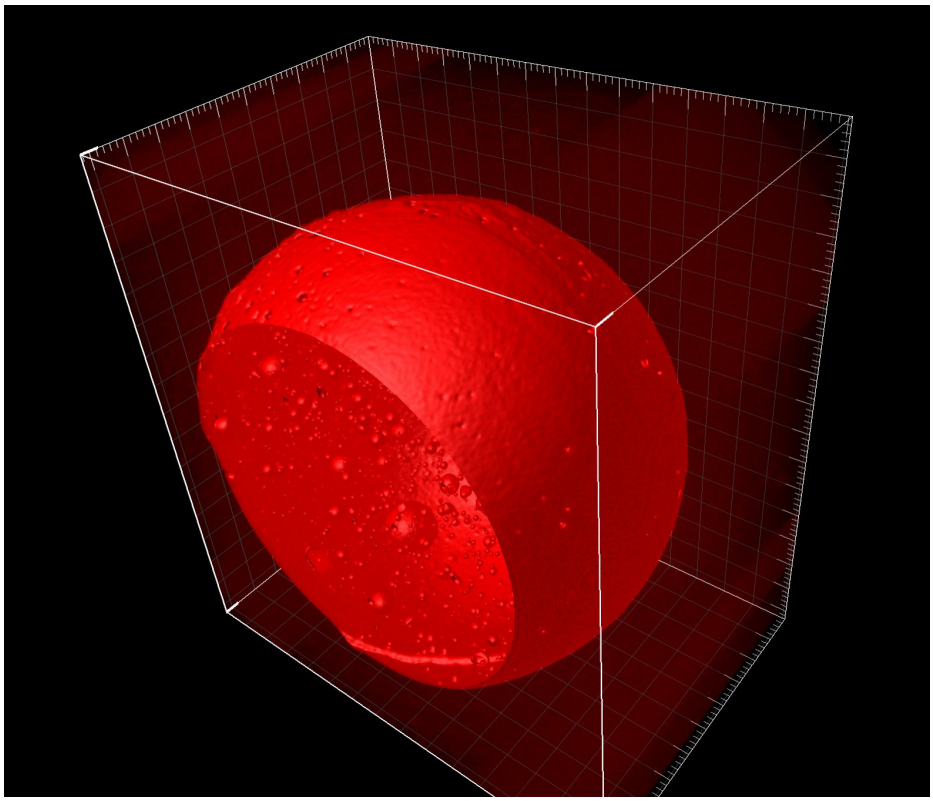


FIGURE 4.3. An example 3D rendering from MicroCT images of a spherical stone. This rendering shows the contour between stone and air values. The data is clipped to enable viewing of the air pockets inside the stone. (Image produced by Jaclyn Lautz.)

TABLE 4.3. The number of shocks each stone was subjected to for each μCT imaging. Bold text denotes image sets that contain clearly visible cracks.

Stone	Number of Shocks											
7 mm cylindrical stone 1	0	2	3	4	5	6	7	8	**9**	**10**	**11**	**12**
7 mm cylindrical stone 2	0	3	4	**5**	**6**	**7**	**9**	**11**	**13**	**15**		
10 mm cylindrical stone	0	5	**10**	**15**	**16**	**18**	**20**	**22**	**24**			
10 mm spherical stone	0	**10**	**14**									

The image sets for the 7 mm stones, for example, each contain about 1200 image slices, totalling about 3 GB of data in Tiff format. The amount of data makes computationally expensive 3D fracture analysis of the images prohibitive. As shown in the following section, 2D fracture analysis of the image slices is chosen. The high resolution of 6 and 8 microns, and therefore the large data sets, are required to capture the fracture as early

as possible. A portion of the goal of the imaging is to see how fracture increases and proceeds as more shocks are applied. Each shock that a stone is subjected to causes some amount of damage. The damage sustained from the first few shocks is on a scale smaller than the resolution of the imaging process. As the stone is subjected to more shocks, more bonds break leading to more damage and wider fractures which are then resolved in the images. To have as many image sets with visible fracture in succession as possible it was imperative to use the highest resolution.

4.2.3. Post-processing of images. Post-processing of the images is done with MATLAB (code is included in Appendices 8.1 and 8.2). The main goal of the image processing is to determine statistical distributions of crack lengths and widths to inform the computational model. The challenge is to quantify the cracks from the image slices in an automated way. Secondary goals include determining how much damage and fracture is added with each shock and if the distributions change with additional shocks. Post processing was only done with the cylindrical stones since these have more successive image sets. This also allows the spherical images to be used to validate the model. The model is developed and parameters are selected with the cylindrical stones. Then a spherical stone will be modeled and a comparison can be made to the experimental images.

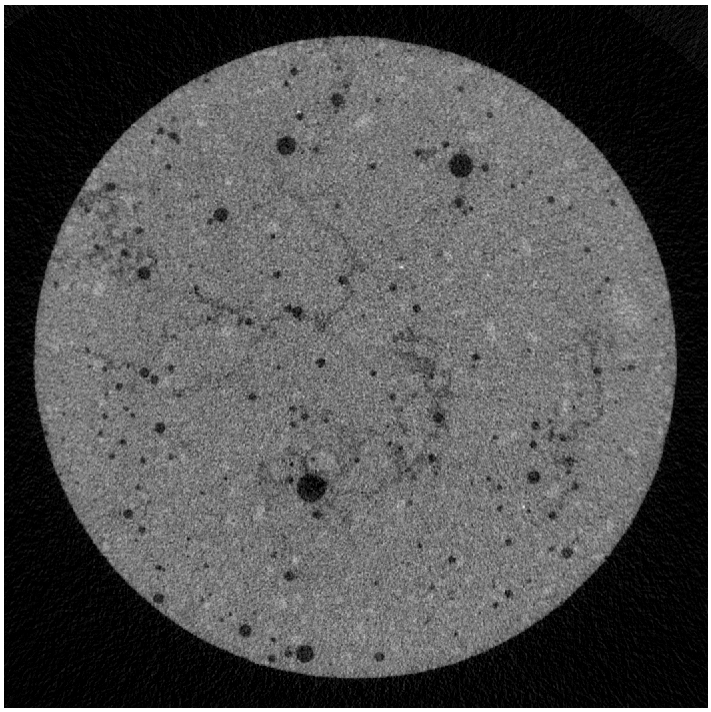


FIGURE 4.4. An example MicroCT image slice of a 7 mm cylindrical stone in which cracks are visible.

The post-processing is split generally into two procedures. The first enhances the cracks and the second quantifies the cracks by overlaying a set of line segments on them. A typical image slice with clearly visible cracks is shown in Figure 4.4 before any image processing occurs. While the cracks are visible they do not stand out as much as some of the other defects in the stone. The dark circles, which are air pockets, are particularly noticeable.

The first image processing procedure enhances the cracks by eliminating other defects and increasing the contrast of the cracks. The first step is to perform a slight low-pass filter (blur) to remove the high frequency noise. Next an equation for the outer border of the stone is computed by finding the edges and assuming the shape of an ellipse. After this the pixels are divided into four categories: outside the stone, darker than a certain

gray value, ligher than another gray value, and other. These categories for the example image are shown in Figure 4.5.

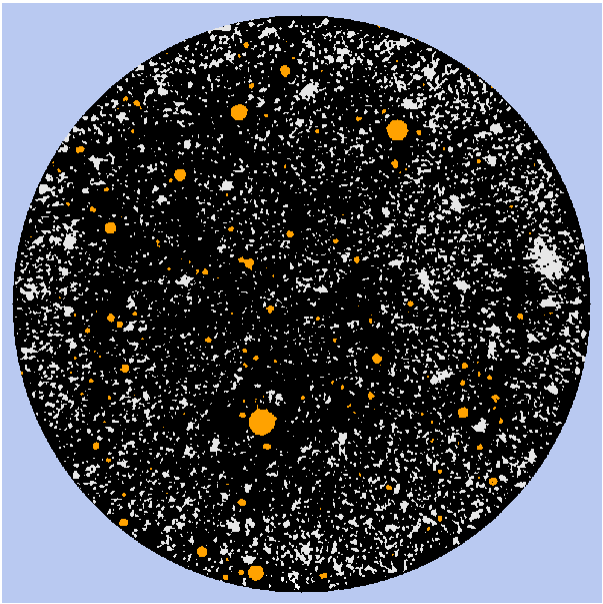


FIGURE 4.5. An example MicroCT image showing the four regions that the image is divided into for the crack enhancing procedure. Blue is outside the stone, gray is the lighter pixels, orange is the dark pixels, and black is the remainder which will contain the cracks.

A new image is created with ones where any dark pixels are and zeros elsewhere. This image is blurred and any non-zero pixels are converted to ones. These new pixels are added as new dark pixels into the categories image effectively expanding the covering of the air pockets. This subprocess ensures that the air pockets are completely covered by the dark pixels.

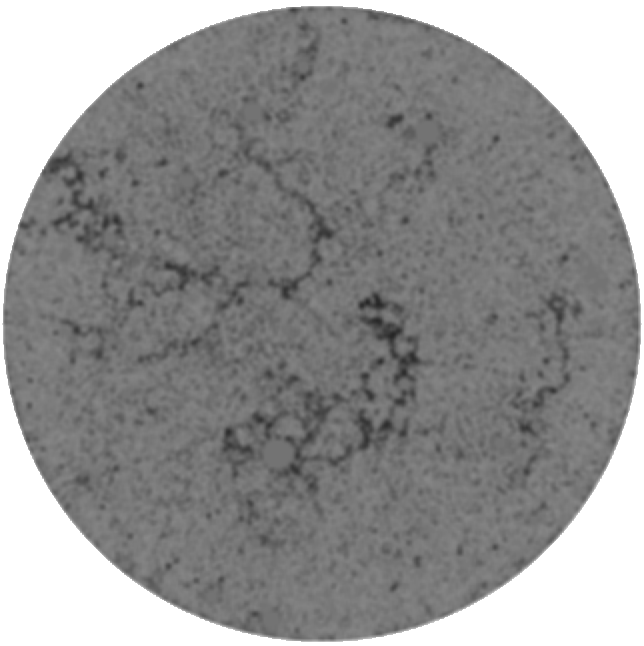


FIGURE 4.6. An example MicroCT image slice after being subjected to the crack enhancing image processing algorithm.

The next step is to replace the dark pixels and light pixels with an average gray value which removes these defects. The background color is also now changed to white. These steps will introduce sharp edges which may be undesirable so another low-pass filter is performed. The final step is to rescale all the gray values into the full [0,255] range. Since the background color has been changed to white this causes the cracks to become very dark features. The enhanced image corresponding to Figure 4.4 is shown in Figure 4.6. The code for the crack enhancing procedure is included in Appendix 8.1.

The second image processing procedure quantifies the cracks by covering them with a set of line segments. This process begins with the enhanced cracks image. The first step is to apply MATLAB's edge detect algorithm to the image. The cracks are wide enough that this finds both sides of each crack as seen in Figure 4.7. Next, midpoints are found between these edges. Using the gradient of the image, the algorithm traces

from edge points through cracks until the other edge is hit, then marks the pixel in the middle. This process results in a set of points covering the cracks.

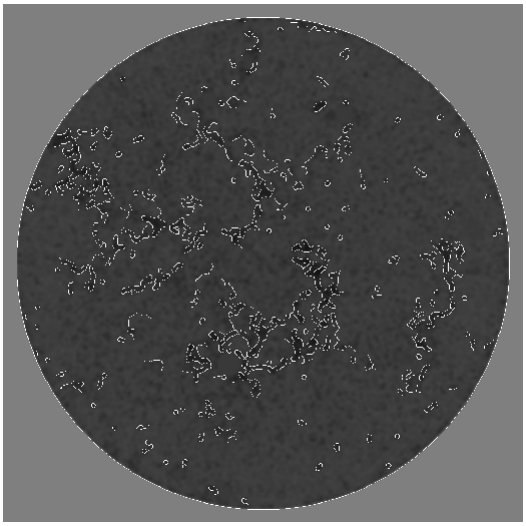


FIGURE 4.7. An example MicroCT image with edges shown in white from MATLAB's edge detection algorithm. The background is a darkened version of the crack enhanced image to help highlight the edges.

The next step is to connect these points with line segments. Essentially, the algorithm loops over the crack points and for each point finds the closest crack point (greater than some distance away) and adds a line segment. This gives an initial covering. Next, each line segment is checked to see if it lays over bright pixels which means the algorithm has connected two separate dark areas. These lines are then removed. To help capture the ends of cracks the segments are randomly lengthened by a small amount. Next, line segment end points that are within some distance of one another are placed at the same point so that the process results in connected segments. Finally, any repeated line segments are removed. Figure 4.8 shows the line segment covering for the example image. The code contains some subtleties that aren't described here but the interested reader may view the code in Appendix 8.2.

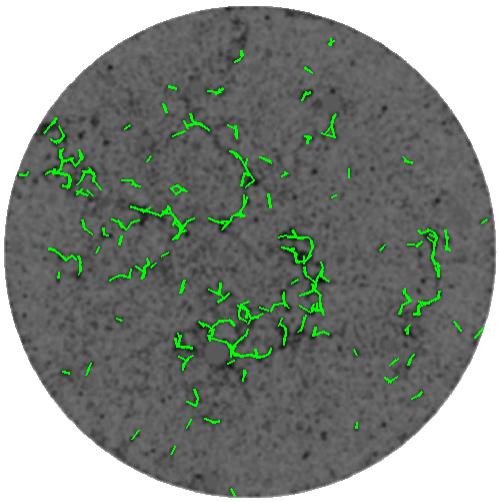


FIGURE 4.8. An example MicroCT image with line segments representative of cracks shown in green. The background is the crack enhanced image.

Crack lengths and widths are extracted from the data provided by the previous algorithm. Crack length data is determined by looping over all line segments and examining any connected line segments. If a connected line segment is in the same direction as the original line segment, within a certain tolerance, then these lengths are added together and the two segments are essentially treated as one. This process continues so that any arbitrary number of segments can be combined as long as they all lay in the same direction. If a segment does not connect to any other segments or only connects to segments that lay in different directions then that segment's length is counted by itself. This process gives a set of crack lengths that can be used to create a histogram for example.

Crack widths are found by first determining the midpoint of each line segment. From the midpoints, the algorithm moves along the path perpendicular to the direction of the line segment. Using data from the enhanced crack image (Figure 4.6), the algorithm stops once it encounters a pixel that is light enough. The width is calculated as the sum

of the distances from the midpoint to the two light pixels. As with the length data, this process gives a set of width data that can be used to create histograms.

4.3. Results

The lognormal probability distribution appears to be a good fit for both the length and width data. The inverse Gaussian, generalized extreme value, and log-logistic distributions also provide good fits of the data, but the lognormal distribution has somewhat of a precedence for being used for fracture parameters in fatigue [**99, 131, 133**] and brittle fracture [**10, 124**].

The equation for the lognormal probability distribution function (PDF) is

$$f\left(x;\mu,\phi\right)=\frac{1}{x\sqrt{2\pi\sigma^{2}}}\exp\left(-\frac{\left(\ln x-\mu\right)^{2}}{2\sigma^{2}}\right). \tag{4.3.1}$$

This distribution results from taking the exponential of a normal random variable with mean μ and variance σ. Families of this distribution over the two parameters are shown in Figures 4.9 and 4.10.

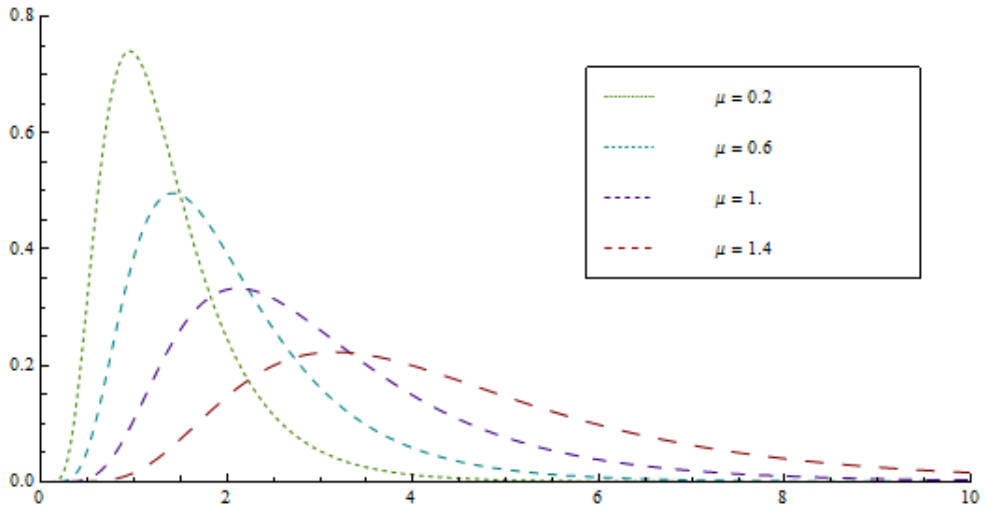


FIGURE 4.9. Lognormal probability distribution function for various values of μ with $\phi = 0.5$.

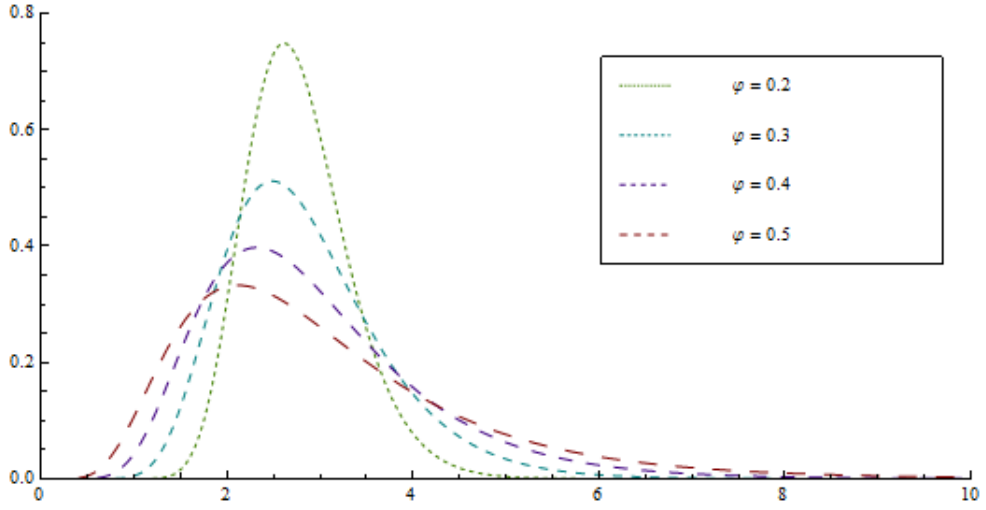
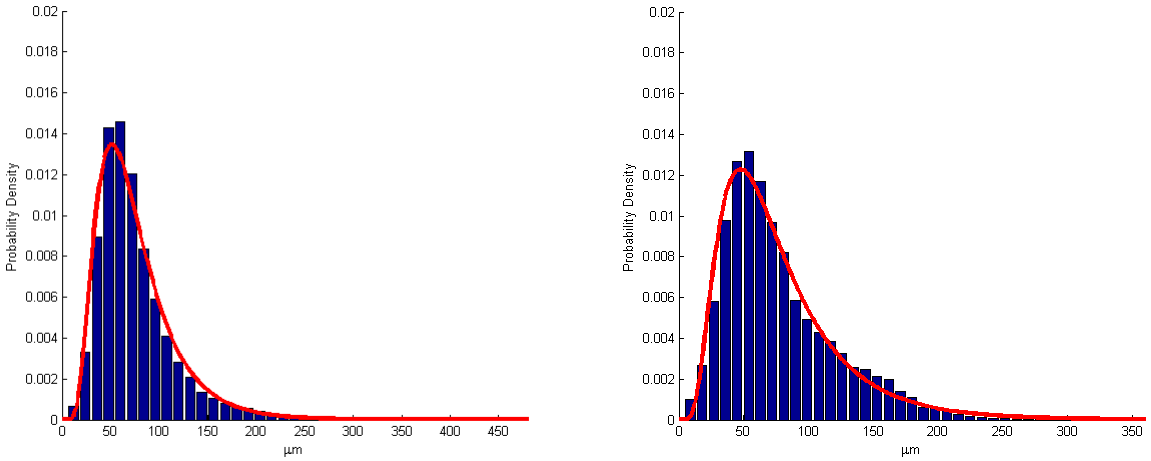


FIGURE 4.10. Lognormal probability distribution function for various values of ϕ with $\mu = 1.0$.

The process described in the previous section essentially creates a set of length and width data for each 2D image slice. To group this data for one image set, i.e. one complete image of the stone at a certain shock number, the data is collected over the visible crack. Since the cracks in the cylindrical stones run predominantly perpendicular

to the axis of the cylinder, which is the same direction as the image slices, only the data from relatively few image slices needs to be accumulated. Data from any one image slice appears to tend towards a lognormal distribution but amassing data over the visible crack provides a smooth and consistent distribution. The histograms of length and width from the 7 mm cylindrical stone 1 shock 10 are shown in Figure 4.11 with lognormal fits. To improve these probability distribution fits and all others shown in this section the data is shifted so that the smallest data is translated to zero. Length data is shifted by -24 μm and width data is shifted by -4 μm.



(A) (B)

FIGURE 4.11. Crack length (A) and width (B) data from 7 mm cylindrical stone 1 at shock 10 meant as an example to show the lognormal fit close up.

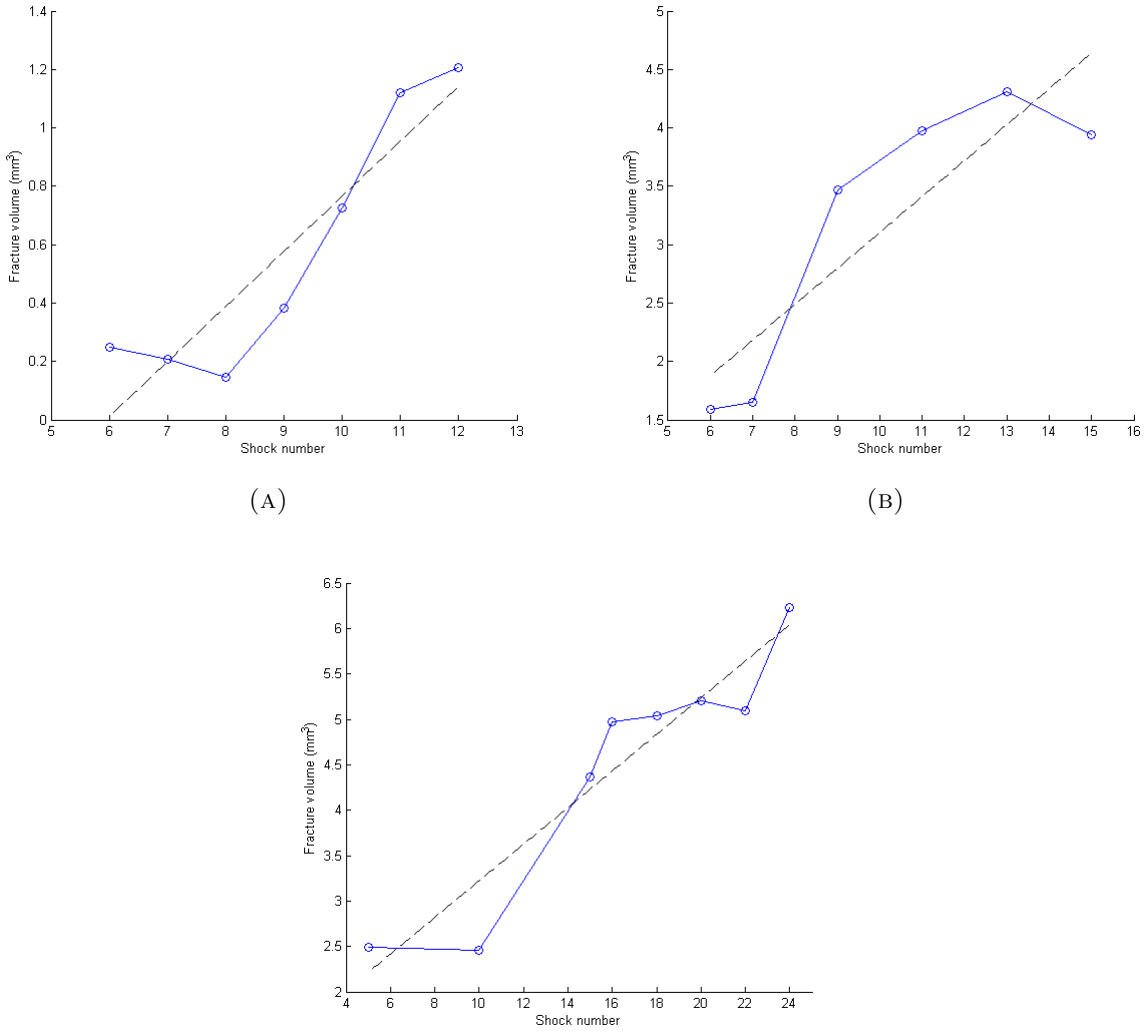
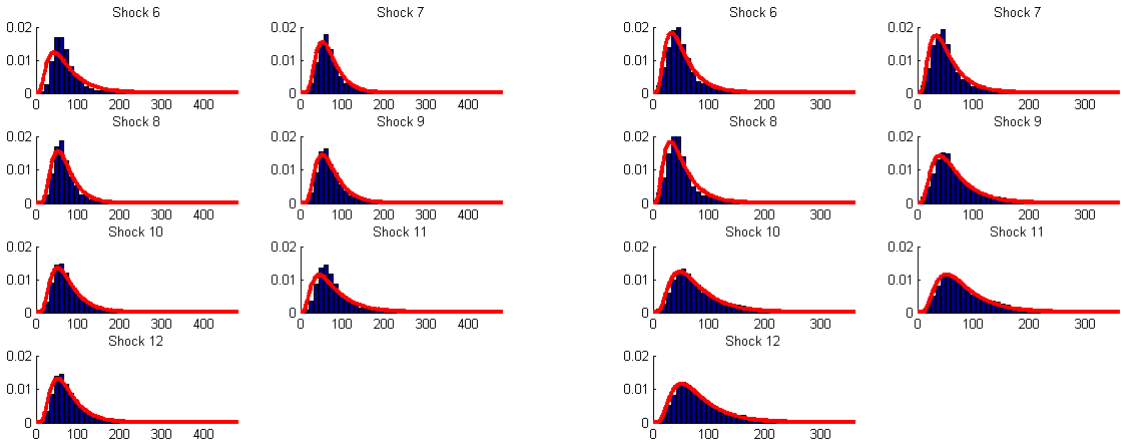


FIGURE 4.12. Volume of fracture calculated by automated procedure for each image set with a linear fit. (A) 7 mm cylindrical stone 1. (B) 7 mm cylindrical stone 2. (C) 10 mm cylindrical stone 1.

In Figure 4.12 the volume of fracture for each image set is shown. This is calculated simply by summing the volume of each individual crack, approximated by the crack length times the crack width times the height of one image slice, 6 μm. The brittle stone should not heal over time and so fracture volume should only increase. As seen in the figures though, fracture volume decreases occasionally when the stone is shocked.

It is assumed that this is not truly happening but is a by-product of the complicated procedure for calculating this data. This includes the imaging procedure itself which can result in different overall image parameters dependent on the orientation and makeup of the stone. Regardless, a general increase in fracture volume is seen in each stone as more shocks are applied.

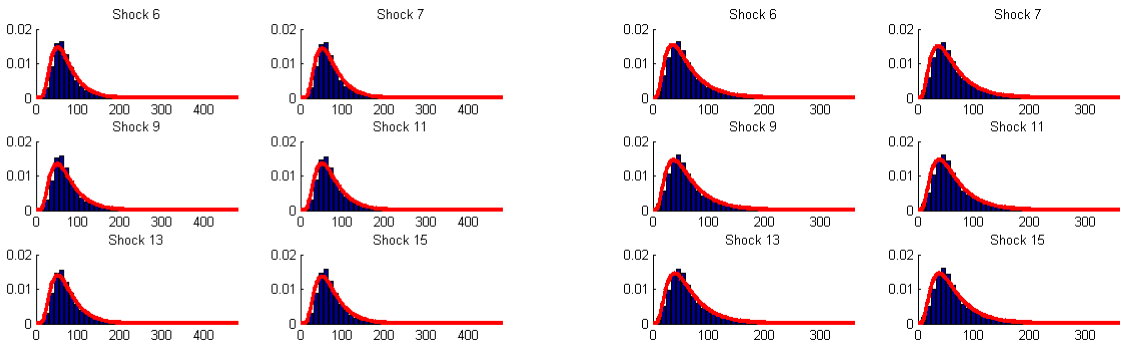


(A) (B)

FIGURE 4.13. Normalized crack length and width data from 7 mm cylindrical stone 1 plotted as histograms for each shock number image set and compared to lognormal distributions. Length and width is measured in μm. Length data is shifted 24 μm left and width data is shifted 4 μm left. (A) Length. (B) Width.

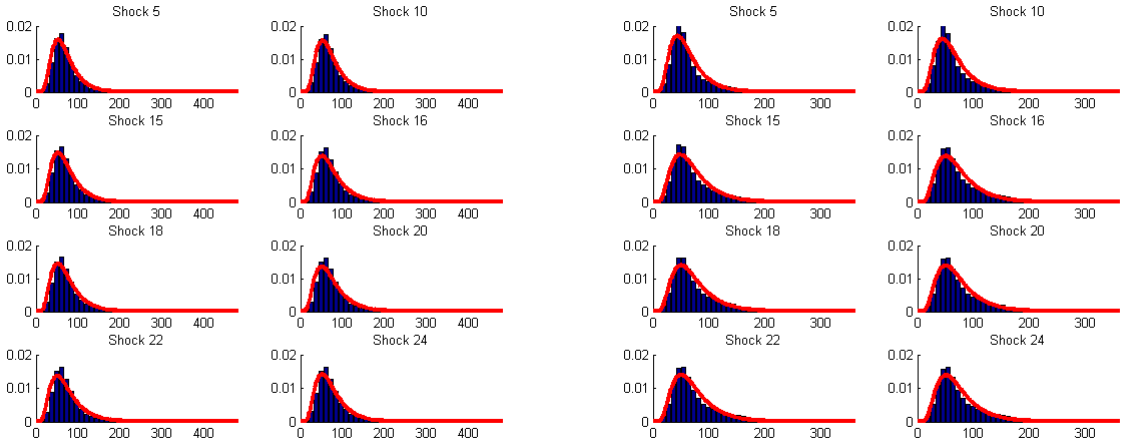
A linear fit is also provided in the figures to help determine some average amount of fracture that is added with each shock. Although its not clear that this increase should be linear, there is not enough data to determine any other specific rate of increase and so for now it is approximated linearly. From the linear fits it is determined that the rate of increase for each stone is 0.189, 0.307, and 0.202 mm^3 per shock for 7 mm stone 1, 7 mm stone 2, and 10 mm stone 1, respectively. For reference the approximate total volumes of the 7 mm and 10 mm stones are about 140 and 640 mm^3, respectively.

In Figures 4.13, 4.14, and 4.15 the length and width histograms for each shock number are plotted against a lognormal fit. The lognormal fit appears to be valid for all histograms. The length data seems to maintain approximately the same distribution as more shocks are applied and more fracture occurs. The width data distribution, on the other hand, seems to flatten out with more shocks.



(A) (B)

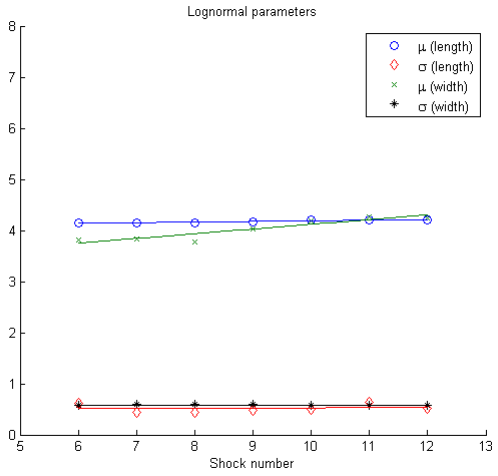
FIGURE 4.14. Normalized crack length and width data from 7 mm cylindrical stone 2 plotted as histograms for each shock number image set and compared to lognormal distributions. Length and width is measured in μm. Length data is shifted 24 μm left and width data is shifted 4 μm left. (A) Length. (B) Width.



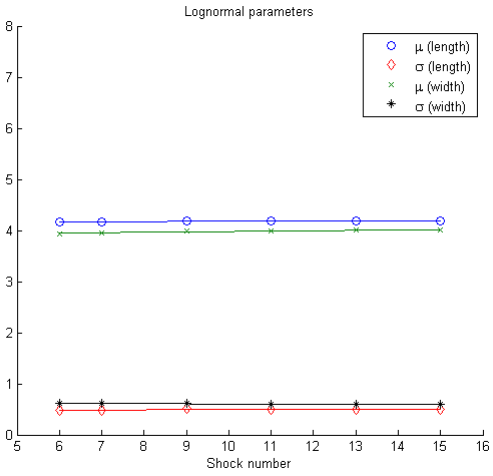
(A) (B)

FIGURE 4.15. Normalized crack length and width data from 10 mm cylindrical stone 1 plotted as histograms for each shock number image set and compared to lognormal distributions. Length and width is measured in μm. Length data is shifted 24 μm left and width data is shifted 4 μm left. (A) Length. (B) Width.

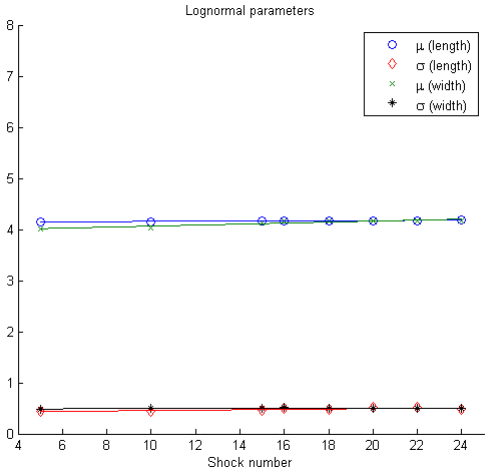
Figure 4.16 shows the lognormal distribution parameters, μ and σ for each lognormal fit seen in Figures 4.13, 4.14, and 4.15. The figure also shows a linear fit for each parameter. The length parameters remain more or less constant over the shocks. The width σ parameter also seems to remain constant, but the width μ parameter increases for every stone as more shocks are applied.



(A)

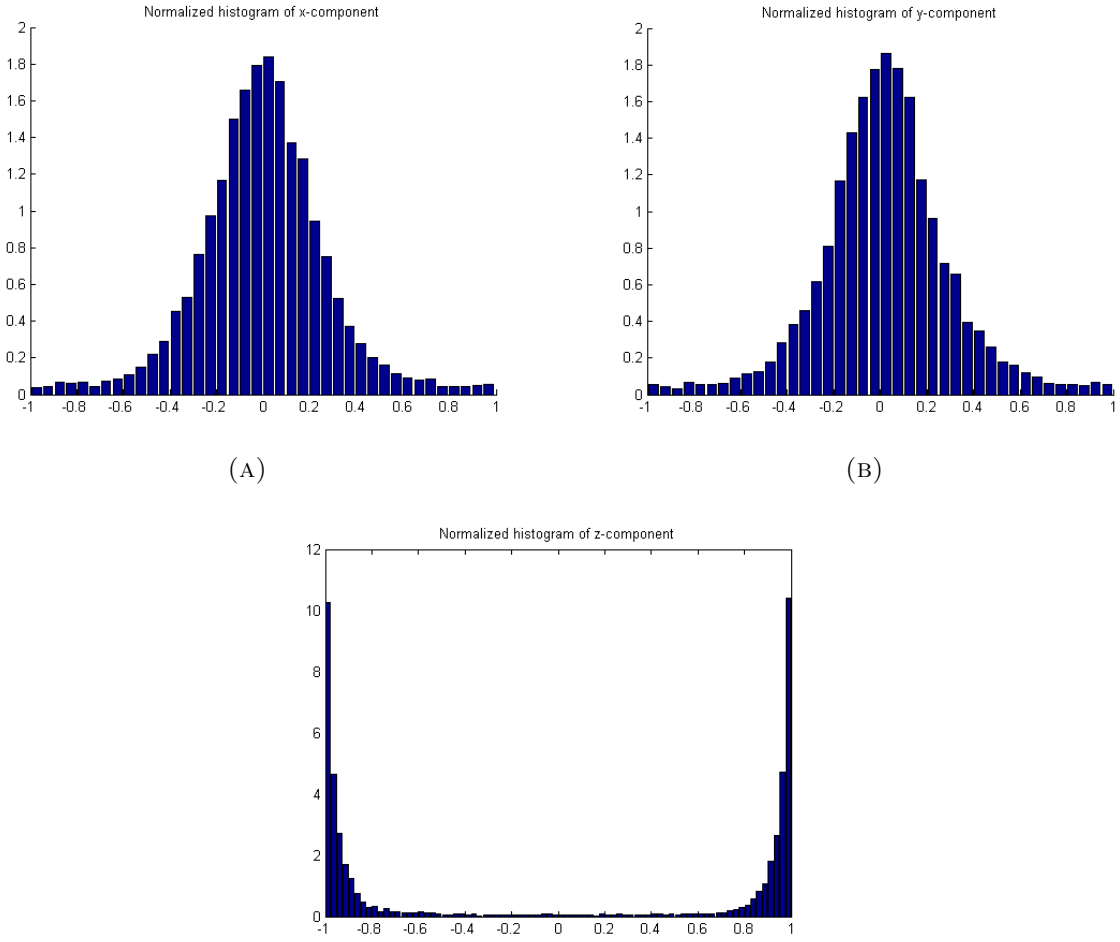


(B)



(C)

FIGURE 4.16. Calculated parameters of the lognormal fits of the crack length and width data for each stone and shock number. (A) 7 mm cylindrical stone 1. (B) 7 mm cylindrical stone 2. (C) 10 mm cylindrical stone 1.



(A) (B)

(C)

FIGURE 4.17. Distributions of the x (A), y (B), and z (C) components of the crack orientations vectors.

Distributions of crack orientation in 3D are also calculated. Cracks in 3D are expected to be planes so in the images it is expected that the cracks will be thinner in one direction than the others. The distributions are calculated by using the crack enhanced images described in the previous section. Several of these images are stacked and a 3D gradient of the image is computed. The gradient vectors point from dark to light pixels, so in 3D the gradient vector within a crack points in the thinnest direction and therefore gives the crack orientation. Figure 4.17 shows distributions of the x, y, and z components of the

orientation vectors over the visible crack for the 7 mm stone 1 shock 10. As mentioned earlier, the main visible cracks in the image sets predominantly run perpendicular to the axis of symmetry of the cylinder, which corresponds to the z-axis in the figures. It is assumed that this is due to the maximum principal stress direction being in the z direction. Therefore, as seen in the figures, it is determined that the crack orientations are normally distributed with mean in the direction of the maximum principal stress.

4.4. Discussion

To briefly summarize this chapter the important points and findings are repeated:

(1) It has been shown that the crack length and width data appear to be lognormally distributed, or at the very least, a lognormal distribution is a good approximation of the data.

(2) An estimate of the fracture volume increase during each shock has been given.

(3) The changes in the lognormal distribution fits as shocks are applied and damage accumulates have been shown.

The length distributions appear constant over time. Even though additional fracture is occurring, as more cracks are seen over time, not just wider cracks, the crack lengths do not seem to be influenced by this. As expected though, the crack width distributions change as more shocks are applied. While perhaps the new cracks being formed are no wider initially than old cracks were initially, the old cracks would be expected to widen as more shocks pass through the stone. Cracks would also, of course, be expected to lengthen, but recall that unless the crack lengthens in the exact same direction it is not counted as the same crack in this analysis.

CHAPTER 5

Multiphysics Focusing Model

This section describes a simulation developed to accurately model the focusing of an electromagnetically generated acoustic pulse through a lens and the subsequent shock wave formation in water, that is, the process in a refracting electromagnetic (EM) lithotripter. It is meant to model an experimental setup used to test lens designs, described in the previous chapter, and thus does not make any attempt to include a patient's body in the domain. The experiment approximates the patient's body with additional water. The model is verified by comparing experimental FOPH measurements and equivalent numerical output from the water near the focus. This output type is also used when predicting effects of lens modifications. In addition, the model can include a kidney stone simulant in the domain. Finally, a simple damage model is incorporated within the linear elasticity equations used to model wave propagation in the stone. The majority of the work presented in this chapter is also included in a published manuscript [**37**]. The model was also used to help validate a new lens design in additional work [**89**].

The simulation is 2D axisymmetric, written in FORTRAN, and uses the BEARCLAW software by Mitran [**86**]. BEARCLAW provides adaptive mesh refinement (AMR) and time integration in a conservative finite-volume Riemann-solver context. For example, to model a hyperbolic system of PDEs, a user of BEARCLAW would specify the waves and speeds of the system, in the Riemann solver sense, and provide other aspects of the

simulation like boundary conditions, initial conditions, source term solutions, and values for coefficients. Then BEARCLAW can solve the system over time while utilizing AMR.

This model incorporates a multiphysics aspect to accurately capture the behavior of the pulse in both the elastic solids and the water as well as the interaction between the two. The axisymmetric linear elasticity equations, shown in Section 3.3, are used to model the focusing of the pulse through the acoustic lens as well as the wave propagation within the stone. These equations are needed to correctly model mechanical waves passing through solids, which includes both p-waves and s-waves. The axisymmetric Euler equations, shown in Section 3.4, are used to model the wave propagation in water. These equations are needed to correctly model the shock steepening that occurs as the pulse strengthens away from the lens. The elasticity equations cannot account for this nonlinear behavior. A diagram of the domain of this computation is shown in Figure 5.1.

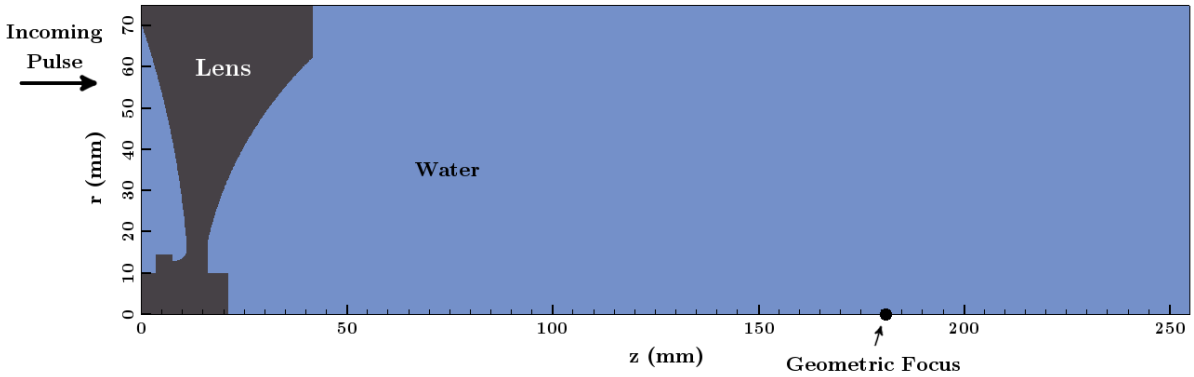


FIGURE 5.1. Diagram showing the domain of the focusing model when not including a kidney stone simulant. The z axis is the axis of symmetry. The incoming pulse enters along the left boundary. The geometric focus of the acoustic lens is labeled.

5.1. Multiphysics

Two different implementations of the multiphysics were used in this work. The first is efficient and is used for the comparisons to experimental FOPH data since this required many runs with varying parameters. The second is more general, but not as efficient in its current form, and is used for simulations that include the kidney stone simulant.

The first implementation solves one equation set at a time. It begins by solving the elasticity equations across the entire domain from the initial time, $t_0 = 0$, to a preset time later in the simulation, $t = t_S$, once the main pulse has passed through the lens. For the results presented in this work $t_S = 28$ μs At this point the current elasticity solution values are transformed to Euler values and the Euler equations are solved for the remainder of the simulation. This implementation of the multiphysics is specific to the EM lithotripter model.

The second method solves both equation sets simultaneously and both across the entire domain. Both domains are divided into elasticity and Euler regions. Solution values are only kept within the associated region and solution values in the other regions are transformed. For example, within the elasticity solver, elasticity values in the Euler domain are discarded and replaced with transformed versions of the Euler solutions. To be more specific, consider an elasticity cell at coordinate (i, j) as pictured in Figure 5.2. Of the eight surrounding cells, cells $(i+1, j)$, $(i+1, j-1)$, and $(i, j-1)$ are Euler cells and the remainder are elasticity. During each time step within the elasticity solver, the values from the Euler solver in cells $(i+1, j)$, $(i+1, j-1)$, and $(i, j-1)$ are transformed to elasticity values and placed in the corresponding cells in the elasticity solver. At this point the actual elasticity solve can take place. The Euler values simply overwrite the

existing elasticity values in those cells. This same process occurs for the Euler solver but with the complement set of cells.

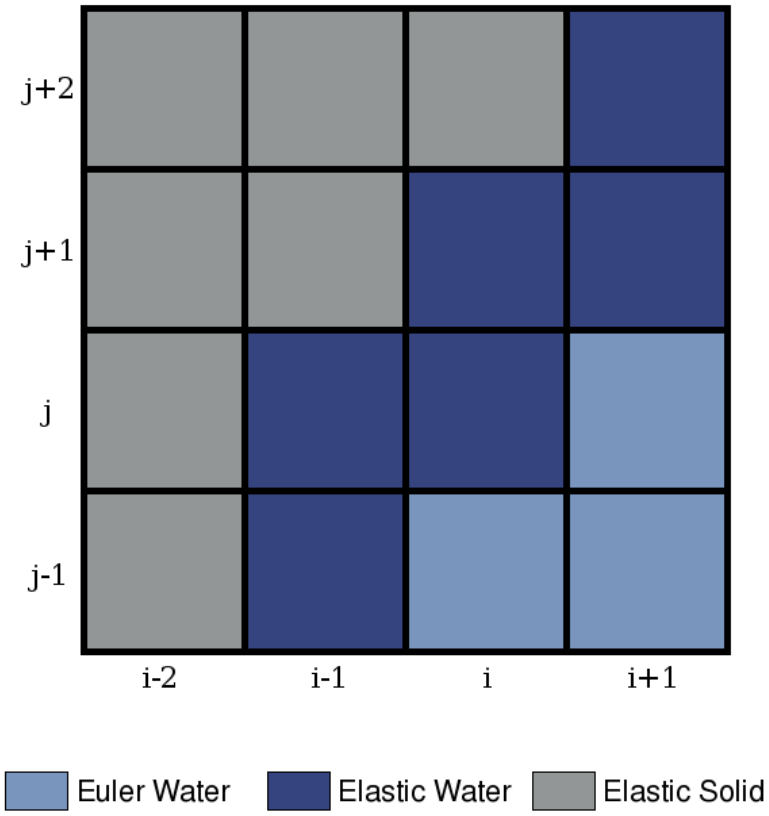


FIGURE 5.2. Reference diagram showing finite volume cells divided into Euler water, elastic water, and elastic solid regions.

This method allows for the whole domain to be divided into arbitrary subsets of elasticity and Euler regions. A diagram of the EM lithotripter computation divided in this way is provided in Figure 5.3. For this application this allows for straightforward inclusion of different lens and stone shapes, as well as other elastic solids such as a stone holder, for example, if the effects from that were decided to not be negligible. This method is much more general and could be applied to other problems requiring the interaction of shock waves in fluids with linear elastic solids. Although, the solids must remain stationary in the current iteration of the model.

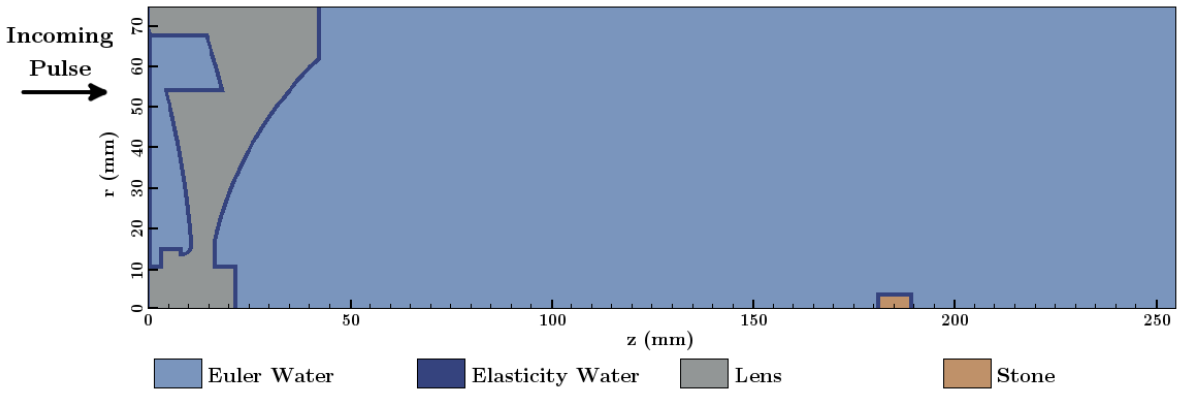


FIGURE 5.3. Diagram showing the domain of the focusing model when using the general multiphysics implementation. The proximal surface of the stone is located at the geometric focus of the lens. The thickness of the elasticity water region is exaggerated to aid in viewing.

This method also requires a transition layer of elastic water between the elastic solids and the Euler water. This is because the transformation between the solution values is only valid for water (or whatever fluid is being studied), not for the solid. The requirement is that the eight cells surrounding any Euler cell must either be Euler cells or elastic water cells. For example, consider cell $(i, j-1)$ in Figure 5.2. The solution values in this cell will be affected firstly by the normal Riemann solves along its four boundaries. Therefore, the four corresponding adjacent cells must be elastic water or Euler water to ensure that these solves are physically correct. In addition, nearby transverse solves also affect cell $(i, j-1)$. The normal solve at boundary $(i-1/2, j)$, for example, will affect cell $(i, j-1)$ through the transverse correction. Therefore, the diagonal cells must also be Euler or elastic water.

This method is inefficient in its current form due simply to the fact that both equation sets are solved across the entire domain. Only near the boundaries between the two regions do both equation sets need to be solved. Away from boundaries, only one equation set needs to be solved since the other set's solution values are discarded. So ideally, this

method would only add a small amount of extra computation time, to compute the double solves at the boundaries, compared to a single equation set solver. Unfortunately, this isn't easily attainable in BEARCLAW, as it was not designed with this process in mind. A Riemann solver built from the ground up to work in this manner could theoretically see the ideal efficiency of the algorithm.

Another option would be to use a single grid but implement three Riemann solvers within it: one for elasticity when both cells are elastic, one for Euler when both cells are Euler, and a third for a mixed solution when one cell is Euler and one is elastic. This method requires more initial coding and so has not been implemented at this point. On the other hand, this method would achieve the ideal efficiency, since only one grid is being used, without having to rewrite large portions of BEARCLAW to work with the dual grid procedure. If future work requires an efficient elastic-Euler solver then this procedure will be implemented. For now, the dual grid implementation is used for the EM lithotripter computations when a stone is included in the domain. The method works seamlessly with AMR and extends to 3D.

The equations for converting the elasticity values to and from the Euler values are shown below. The E subscript denotes elasticity values and the F subscript denotes fluid or Euler values. To compute Euler from elasticity, the pressure is first calculated from the average of the normal stresses. This value is negated to account for the opposite conventions in fluids and elasticity. The equations for the density and energy are the modified Tait equation of state and the momentum values simply come from the density

and velocities.

$$p = -\left(\sigma_E^{zz} + \sigma_E^{rr} + \sigma_E^{\theta\theta}\right)/3$$

$$\rho_F = \rho_0 \left(\frac{p+B}{p_0+B}\right)^{1/n}$$

(5.1.1)
$$(\rho u)_F = \rho_F u_E$$

$$(\rho v)_F = \rho_F v_E$$

$$(\rho E)_F = \frac{p+B}{\gamma-1} + \frac{1}{2}\left((\rho u)_F u_F + (\rho v)_F v_F\right)$$

where ρ_0 and p_0 refer to the initial water density and initial pressure, respectively.

To convert Euler to elasticity the displacement velocities are first calculated and then the EOS is used to determine the pressure. The normal stresses are set to the negative of the pressure and the shear stress is set to zero. The transformation is

$$u_E = \frac{(\rho u)_F}{\rho_F}$$

$$v_E = \frac{(\rho v)_F}{\rho_F}$$

(5.1.2)
$$p = (\gamma-1)\left[(\rho E)_F - \frac{1}{2}\left((\rho u)_F u_E + (\rho v)_F v_E\right)\right] - B$$

$$\sigma_E^{zz} = \sigma_E^{rr} = \sigma_E^{\theta\theta} = -p$$

$$\sigma_E^{zr} = 0.$$

5.2. Description of Simulation

5.2.1. Input. The simulation does not model the electromagnetic pulse generator explicitly, but rather sets the values at the boundary to be the incoming wave created by the pulse generator. So the pulse generator can be thought of as being just outside the

domain, specifically the left side of the diagrams in Figures 5.1 and 5.3. The incoming pressure pulse is an approximation of experimentally measured pressure from near the surface of the pulse generator, described in Section 5.3.

The initial pulse enters entirely into the elasticity domain in either multiphysics implementation. The pulse is modeled by setting the values of the ghost cells along the $z = 0$ boundary. The input is generally formatted as pressure values for certain radial positions and times. These values are interpolated in space and time to match the current time of the simulation and the radial positions of the cell centers. Let p_j^n be the interpolated pressure value for the n^{th} time step and j^{th} finite volume cell along the boundary. Assuming isotropy, the solution values in the ghost cell region are set by

$$\begin{aligned} &(\sigma^{zz})_j^n = 2p_j^n\,,\ (\sigma^{rr})_j^n = 2p_j^n\,,\ (\sigma^{\theta\theta})_j^n = 2p_j^n\,, \\ &(\sigma^{zr})_j^n = 0\,,\ u_j^n = 0\,,\ v_j^n = 0\,. \end{aligned} \tag{5.2.1}$$

The pressure values are doubled because the initial pulse will split into left-going and right-going halves and only the right-going half will enter the domain.

Input also includes the lens geometry which is the main application interest. This is set by various parameters including those to control the curvature on either side of the lens, cuts and other modifications, and material parameters. Finally, the stone shape and material properties can also be set by the user. In this work only cylindrical stones are included, but the general multiphysics implementation allows for the straightforward inclusion of any axisymmetric shaped stone.

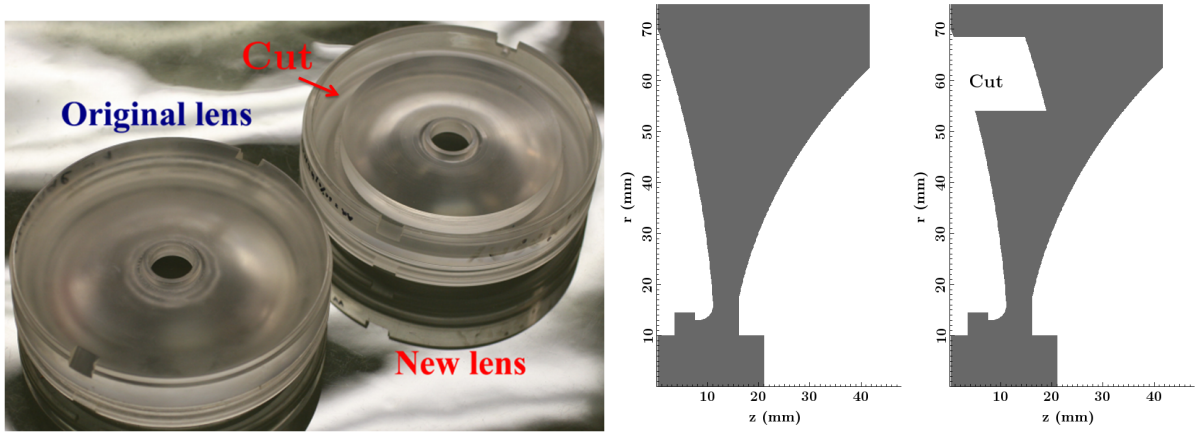
5.2.2. Modeling the Lens and Stone. The elasticity equations are used in this computation to model wave propagation in the elastic solids (lens and stone) as well as in the

small regions of water surrounding these solids. By modeling water using these equations it is assumed that under the conditions in this simulation that water behaves like a solid. In other words, there is no significant flow when sound waves travel through water. Another concern is the absence of nonlinear effects, but since the acoustic waves have either relatively low amplitude or are traveling over very short distances when being modeled by the elasticity equations, these effects are assumed to be negligible.

An important feature of the water is that it doesn't support shear waves. Simply setting the shear wave speed to zero leads to instabilities at the lens-water boundary at high resolutions. As shown in Table 5.1 the shear wave speed is set to 10^{-4} which is approximately what is found in the literature for the elastic properties of water at very small scales [**60**]. Within this simulation the exact value of this number will not noticeably affect the outcome. This is because the other wave speeds are so much larger relatively and the timestep calculated to ensure a CFL close to one is based on the largest wave speed. So, the numerical diffusion of the shear waves within the water will overcome any wave propagation and there will essentially be no shear waves in the water. Again, the only requirement of this parameter is that it is a small positive number, but a value consistent with what is found in the literature is chosen regardless.

The two polystyrene lenses used to validate this model are shown in Figure 5.4 along with cross section diagrams. This figure also points out the difference in the lenses which is an annular ring cut designed to improve efficacy of the lithotripsy procedure. The representation of the original (or standard) lens within the computational domain can be seen in Figures 5.1, 5.3, and 5.5. The material parameters which define the lens, water, and stone in the computation are shown in Table 5.1. The stone material properties are

based on Ultracal-30 gypsum, which has been shown to have similar material properties as struvite and uric acid kidney stones [**81**].



(A) (B)

FIGURE 5.4. (A) Photograph of the two lenses used. On the left is a standard lens design. On the right is the new lens modified by an annular ring cut. (B) Diagrams of the cross sections of the two lenses, original on the left and modified on the right. $r = 0$ corresponds to the central axis of the lenses.

TABLE 5.1. Material properties used in the 2D axisymmetric elasticity equations for water, lens (polystyrene), and stone (Ultracal-30).

			Water	**Lens**	**Stone**
Density	ρ	(kg/m^3)	1000	1060	1700
P-wave speed	c_p	(m/s)	1489	2337	2840
S-wave speed	c_s	(m/s)	10^{-4}	1157	1430
First Lamé parameter	λ	(Pa)	2.217×10^9	2.951×10^9	6.759×10^9
Second Lamé parameter	μ	(Pa)	10^{-5}	1.418×10^9	3.476×10^9

At the lens-water boundary averaging of the material properties is used in cells which contain both lens and water. This is done along the curved edges of the lens to ensure the full accuracy of the method is attained. This is not done along straight edges since the cell size is within the measurement error of the lens. In each of these cells the fraction of the cell containing water, f_W, and the fraction of the cell containing lens, $f_L = 1 - f_W$, are calculated. The density in the cell is computed by an arithmetic average, however

the Lamé parameters must be computed by harmonic averaging [**36**]. New sound speeds are then computed from the averaged density and Lamé parameters. The algorithm for defining the averaged values in these cells is

$$
\begin{aligned}
\rho_A &= f_L \rho_L + f_W \rho_W \\
\lambda_A &= 1/\left(\frac{f_L}{\lambda_L} + \frac{f_W}{\lambda_W}\right) \\
\mu_A &= 1/\left(\frac{f_L}{\mu_L} + \frac{f_W}{\mu_W}\right) \\
c_p &= \sqrt{\frac{\lambda_A + 2\mu_A}{\rho_A}} \\
c_s &= \sqrt{\frac{\mu_A}{\rho_A}}.
\end{aligned}
\tag{5.2.2}
$$

The stone boundary is defined simply by setting any cell whose center falls within the stone to have stone properties, otherwise the cell is given water properties. Within the stone a simple isotropic damage law is implemented. At each time step and in each cell the maximum principal stress, σ_1, is calculated. The maximum principal stress is the largest eigenvalue of the stress tensor. The damage growth model used here is

$$
\frac{dD}{dt} = \alpha\left(\sigma_1 - \sigma_c\right), \tag{5.2.3}
$$

where D is the damage variable, σ_c is a critical stress value, and α is a scalar multiplier. The damage can range from an undamaged state to some critical value, i.e. $0 \leq D \leq D_c < 1$. The damage only increases if $\sigma_1 > \sigma_c$ and $\frac{d\sigma_1}{dt} > 0$, otherwise $\frac{dD}{dt} = 0$. The inclusion of damage here is preliminary and is meant to show capabilities of the multiphysics focusing model. For that reason, little emphasis is placed on selecting the damage parameters in this chapter. For these computations, $D_c = 0.5$, $\sigma_c = 10$ MPa, and $\alpha = 0.004$ are chosen.

The damage variable affects the wave propagation through the Lamé parameters. The two parameters become functions of time as the damage changes

$$\begin{aligned} \lambda &= \lambda(z,r,t) = \lambda(z,r)(1-D(t)) \\ \mu &= \mu(z,r,t) = \mu(z,r)(1-D(t)). \end{aligned} \tag{5.2.4}$$

5.2.3. Initial Conditions and Boundary Conditions. The initial conditions in this simulation are hydrostatic atmospheric conditions. In the elasticity regions the three normal stresses are set to be atmospheric pressure, $p_0 = 0.1$ MPa, and the shear stress and displacement velocities are set to zero. In the Euler regions the variables are set using $\rho_0 = 1000$ kg/m^3 and $p_0 = 0.1$ MPa with zero initial momentum.

Surrounding the domain of the simulation there is a region of ghost cells whose values are set at each time step to enforce a particular boundary condition. The directions in this section refer to the orientation shown in Figures 5.1 and 5.3. As discussed above, the left boundary condition at $z = 0$ serves as the input for the simulation. The three normal stress variables are set to double the pressure values of the incoming pulse within the ghost cell region. After the pulse has completely entered the domain this boundary uses a solid wall boundary condition.

The bottom boundary condition at $r = 0$ requires a mirrored boundary condition due to the axisymmetry. All ghost cell values are set to the values in their mirrored cell across $r = 0$, except for the velocity in the r direction which is negated. The mirrored boundary condition and the solid wall boundary condition are equivalent. The right and top boundary conditions are set to a zero order extrapolation outflow boundary condition. The ghost cells are set to the value of the closest domain cell at every time step. An

outflow boundary condition is chosen to mimic the computational domain being within a tank of water larger than it.

5.2.4. Adaptive Mesh Refinement. This model is a prime candidate for adaptive mesh refinement (AMR), as the most significant part of the solution is a pulse, much smaller than the domain, travelling across the length of the domain. In addition, a very fine spatial step is required to capture the shock steepening whereas a very coarse grid is sufficient away from the pulse. Dynamic AMR is used in this simulation in a physically-inspired way. The refined grid is manually controlled to move along with the pulse since its speed and position can be estimated. Any measured output specific to the pulse is taken from the finest grid. Both implementations of the multiphysics work with AMR.

5.2.5. Other details. The root level grid has a grid spacing of 1.5 mm. The refinement ratio is 48 leading to a grid spacing of 31.25 μm on the fine grid. Any results that include a stone were run with a refinement ratio of 32 which corresponds to a grid spacing of 46.875 μm. The lower refinement ratio was chosen to reduce computational time. As these results are not compared to experiment, but are meant to showcase capabilities of the multiphysics model, the full resolution isn't used.

As stated earlier the time step is chosen based on a desired CFL and the maximum wave speed encountered on the previous time step. With a refinement ratio of 48, this leads to a time step of about 13.1 ns during the elasticity portion. Early in the Euler portion the time step is about 20.7 ns but as the pulse focuses and the shock develops the time step decreases to account for the increase in wave speed.

Both sets of equations have source terms due to the axisymmetry. These are updated with Strang splitting using the exact solutions of the ODEs after removing spatial

derivatives from the PDEs. The focusing model is second order and uses the monotized central-difference wave limiter. The simulations are run in serial on a linux machine with a 3.3 GHz Intel Xeon X5680 CPU and the efficient model takes about 70 hours to complete.

5.2.6. Output. Many different forms of output can be extracted from this model depending on what is needed because the solution values at every spatial point and at every time step are known. The most obvious form of output are the visualizations shown in Figure 5.5 where the values over portions of the domain at select time steps are shown. However, to compare to experiment the output is set up to mimic the way the FOPH records values. A certain number of locations are set and pressure at those locations is recorded over time. Another helpful form of output is a map of peak positive and peak negative pressures near the focus. Examples of these output forms are shown in Figures 5.6 and 5.7. When a stone is included, the computation tracks the maximum of the maximum principal stress over time and the damage, both within the stone. These variables show the effect of the pulse over the duration of the simulation.

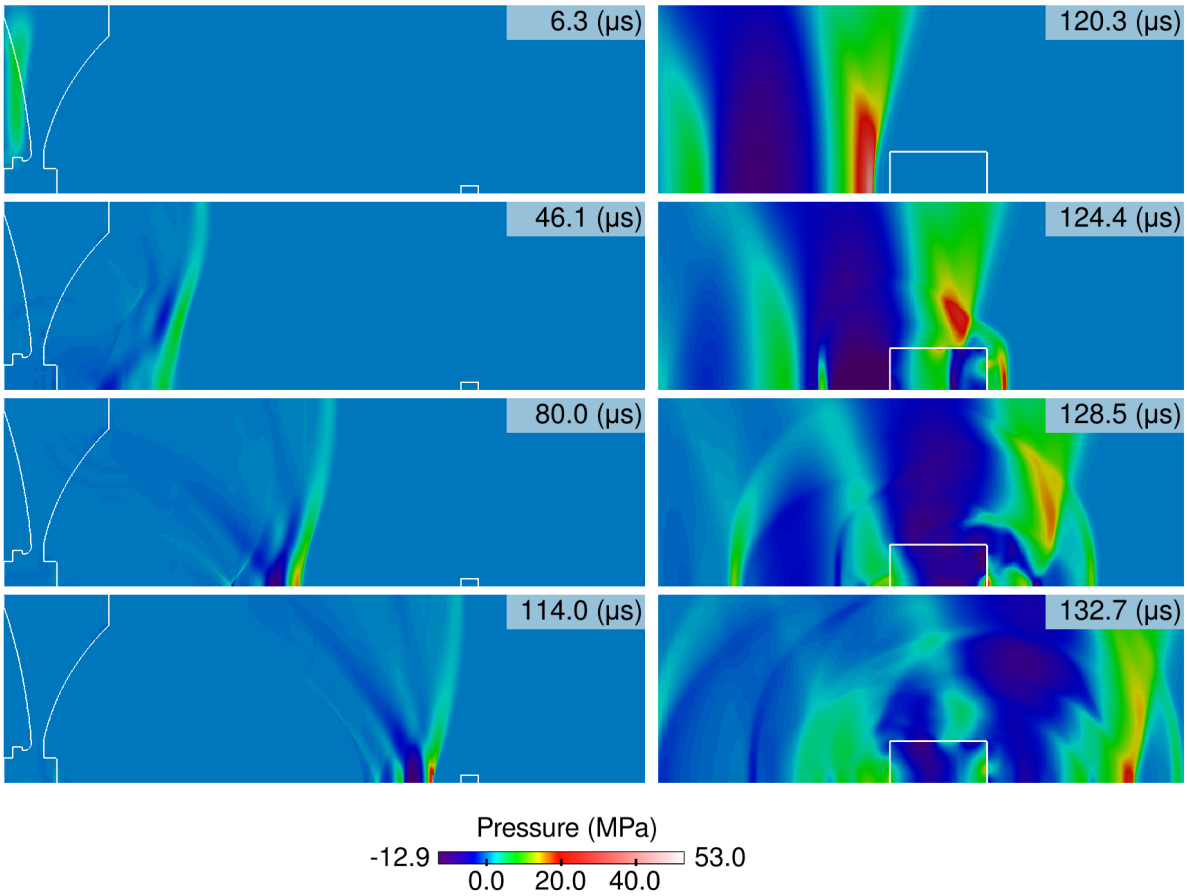


FIGURE 5.5. Progression of the computational solution at selected times. On the left the focusing and shock wave formation over the entire computational domain is shown. On the right the interaction of the shock wave in the fluid region and elastic stone is shown. The original lens and 15.8 kV input is used here. Within the elastic regions the average of the normal stresses is displayed as pressure.

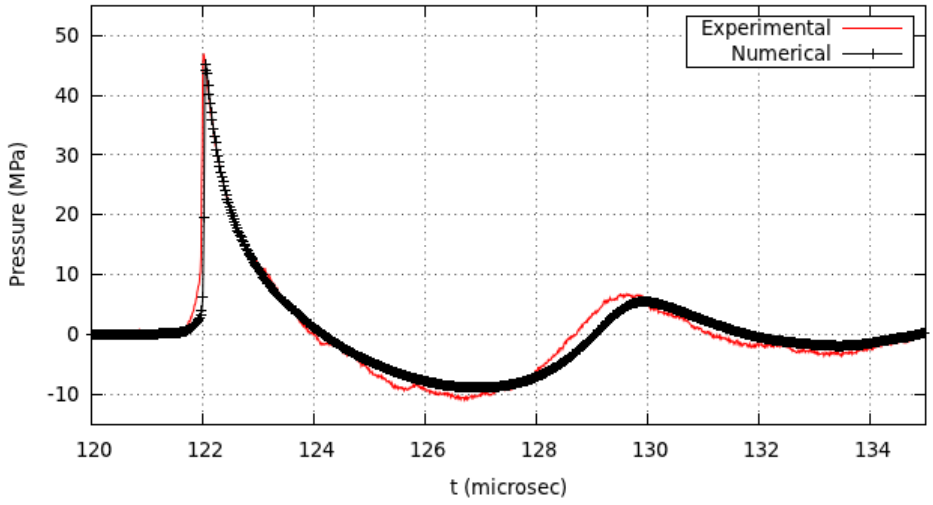


FIGURE 5.6. Example comparison of the experimentally recorded hydrophone data and the numerically recorded data at the geometric focus of the lens.

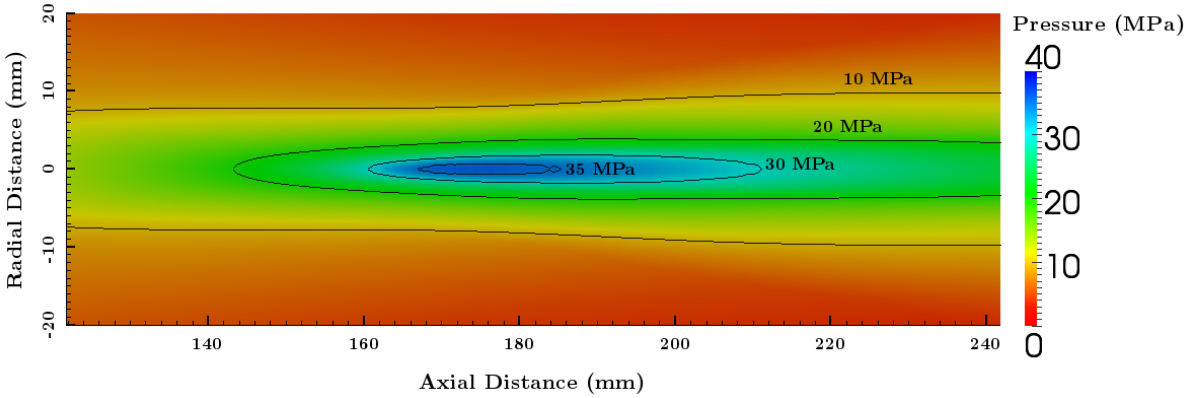


FIGURE 5.7. Example plot of maximum pressure readings near the focus with contour lines.

Finally, output from this model is used as input for the fracture model described in Chapter 6. Values are recorded along a slice at a specified z location, $z = z_1$, and over a range of r, $r = [r_1, r_2]$, and time $t = [t_1, t_2]$. z_1 is chosen to be just ahead of where the stone would be located and $r_1 = 0$ and r_2 are chosen so that when the data is converted to input it will cover the input side of the domain. The times are chosen to simply cover

the range of the pulse. The values currently being used are $z_1 = 179.8$ mm, $r_1 = 0$, $r_2 = 15$ mm, $t_1 = 119$ μs, and $t_2 = 134$ μs.

5.3. Results

This section discusses results of the multiphysics focusing model and shows comparisons to experiments. First, the functions found to approximate the input are presented. Next, the model is validated against pressure measurements recorded using the original lens. The model is also shown to correctly predict the effects of an annular ring cut. This lens modification was advanced by Qin [**100**] as a method to allow refracting EM lithotripters to mimic the pressure profiles of the HM3 near the focus. Recall from Section 2.1 that EM lithotripters, while having some desirable qualities compared to the HM3, have lower stone-free rates and increased retreatment rates. It is hypothesized that this is mainly due to differences in the pulse near the focus. The annular ring cut modification creates a delayed wave which interferes with the main wave causing a lengthening and weakening of the tensile portion and second compressive portion of the pulse as well as an increase in the beam width. Zhong *et. al.* [**136**] and Mancini *et. al.* have shown that this new lens demonstrates improved stone comminution both *in vitro* and *in vivo* compared to the original lens design.

5.3.1. Model Input. The first result presented is the characterization of the direct pulse produced by the electromagnetic actuator, as mentioned in Section 4.1. This is used to create the input for the computational model. The peak pressure of the plane wave created by the actuator,

$$p_0(V) = 5.16 \times 10^{-4} V^{1.895}, \tag{5.3.1}$$

is approximately proportional to the square of the source voltage (V). The radial profile of the pulse is fit by

$$p_r(r,V) = p_0(V)\left(1 + \frac{(r-r_0)^2}{r_1^2} - \frac{(r-r_0)^4}{r_2^4}\right), \tag{5.3.2}$$

where $r_0 = 43.5$ mm, $r_1 = 93.5$ mm, and $r_2 = 28.0$ mm. Finally, the function

$$p_{\text{input}}(r,V,t) = \begin{cases} a_1 p_r(r,V)\sin^2(a_2 t)\exp(a_3 t) & , \quad p_r \geq 0 \\ 0 & , \quad p_r < 0 \end{cases} \tag{5.3.3}$$

is used to define the pressure over the time interval $0 \leq t \leq 20$ mm, where $a_1 = 147$, $a_2 = 0.454 \times 10^6$, and $a_3 = -0.25 \times 10^6$. Example plots of the pressure over time and radial distance are shown in Figure 5.8.

5.3.2. Validation using original lens. Comparisons of pressure profiles, peak pressures, and calculated lithotripter parameters are presented in order to validate the computational model against experiment. In this section and the next the plots showing pressure profiles have had the numerical data shifted slightly left or right to align the shock front for aiding visualization. These shifts in time vary from plot to plot and are less than 0.3 μs. No significant change in the shape of the pulse would occur from correcting for this by using small changes in the wave speed parameters, so a simple translation is used.

Figure 5.9 shows good agreement between experimental and numerical pressure profiles including easily discernible parameters such as peak pressures, P_+ and P_-, and pulse durations, t_+ and t_-. Figures 5.9a and 5.9b show pressure profiles along the propagation axis ($r = 0$ mm), at $z = 121.8$, 151.8, 181.8 (focus), 211.8, and 241.8 mm, with 13.8

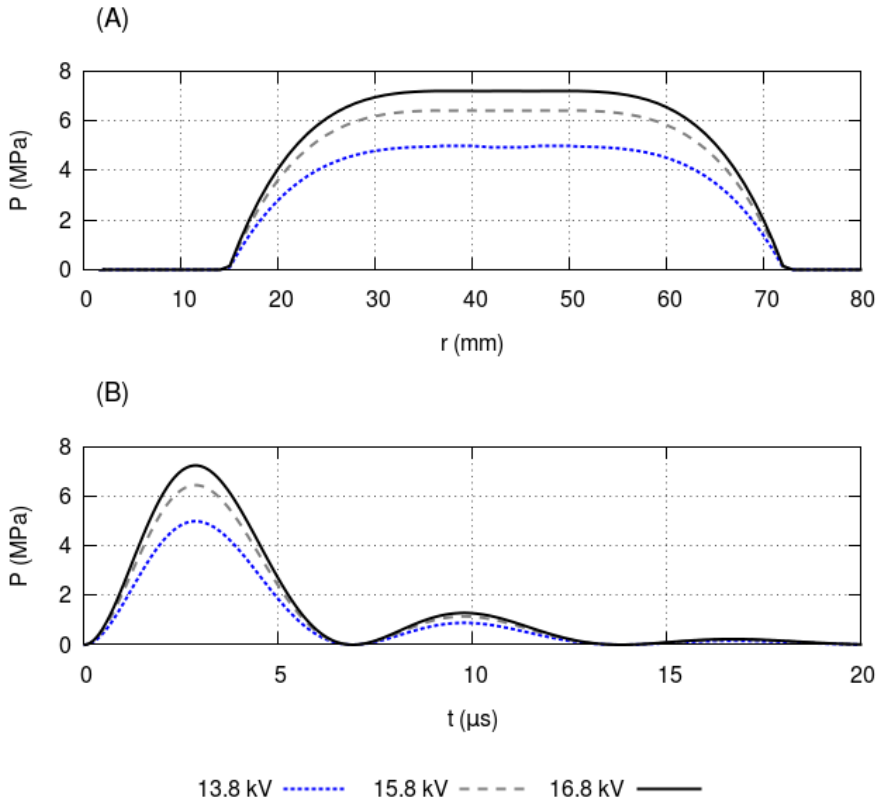


FIGURE 5.8. Example plots of the incoming pulse for the three voltage levels predominantly used here. (A) Pressure distribution in the radial direction at $t = 3$ μs. (B) Pressure over time at $r = 40$ mm.

kV and 15.8 kV input, respectively. Figures 5.9c and 5.9d show pressure profiles in the focal plane ($z = 181.8$ mm), at $r = 0$, 2, 4, 8 mm, with 13.8 kV and 15.8 kV input, respectively. In these latter images it is apparent that the duration of the tensile portion of the pulse (t_-) is less in experiment than in the model. This may be due to the effect of tensile wave shortening from the cavitation that is produced by the tensile portion [**5, 71, 97, 115**]. Since the computation does not include cavitation this effect is not modeled and the numerical result contains the full non-attenuated tensile portion.

Figure 5.10 shows that the distribution of peak pressures in the focal plane for 13.8 and 15.8 kV input is well captured by the model except for an approximate 10% difference in P_+ near the focus. These plots can also provide a visual estimate of the beam width.

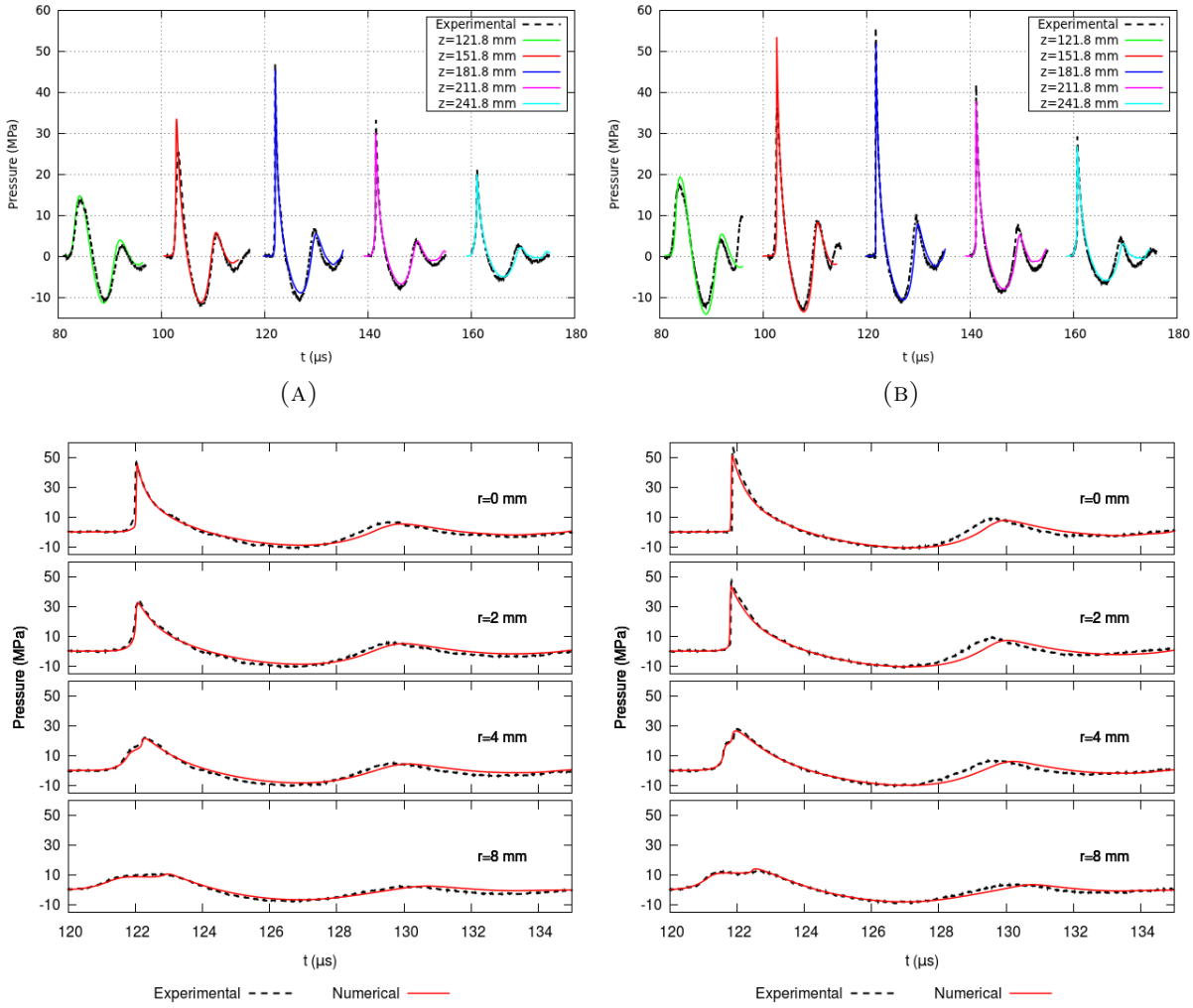
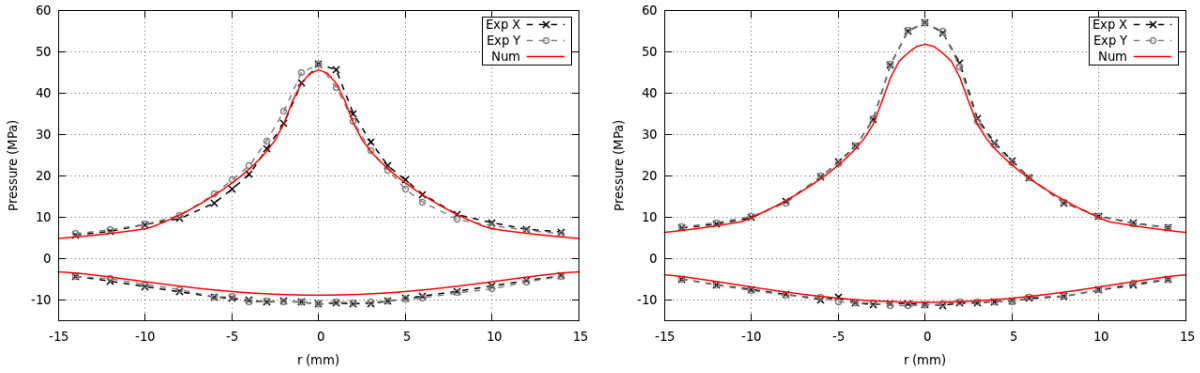


FIGURE 5.9. Plots of experimental and numerical pressure profiles along the propagation axis, $r = 0$, and in the focal plane, $z = 181.8$ mm for the original lens. (A) Propagation axis with 13.8 kV input. (B) Propagation axis with 15.8 kV input. (C) Focal plane with 13.8 kV input. (D) Focal plane with 15.8 kV input.

Figure 5.11 plots the peak positive and negative pressures and the beam width in the focal plane over the dynamic range of the lithotripter. This plot also shows fitted polynomial curves of the data. Peak negative pressures are very well matched with numerical values consistently only slightly lower (in absolute value) than experimental values. Although less data is available, beam width values match very well. Peak positive pressure matches

well for the mid range input voltages which are typical of the source voltages used in the medical procedure [**32**]. Experimental P_+ is up to 30% lower than numerical for weaker input pulses which may be due to extrapolation error in the numerical input. For the strongest input pulses the experimental P_+ is up to 13% higher than the numerical P_+. This may be improved by further refinement of the finite volume grid. Though in order to retain moderate runtime for the most relevant input voltages, finer grids were not used.



(A) (B)

FIGURE 5.10. Plots of peak positive and peak negative pressure in the focal plane ($z = 181.8$ mm) for the original lens. Experimental data is recorded in four directions from the z-axis ($x+, x-, y+, y-$). Numerical data is mirrored across $r = 0$ to aid in visualization. (A) 13.8 kV. (B) 15.8 kV.

Table 5.2 presents lithotripter parameters calculated from the experimental and numerical pressure profiles at the focus and in the focal plane. The pulse parameters, P_+, P_-, t_+, and t_-, match very well for both input voltages. Percent error for these parameters range from 2.4% to 12.7%. The larger discrepancy in the rise time may be attributed to the chosen coarseness of the grid since this involves a measurement of the shock. Beam width error ranges from 2.7% to 10.7% over the input values and acoustic energy error ranges from 4.3% to 34.9%. The FOPH pressure measurements are estimated to have

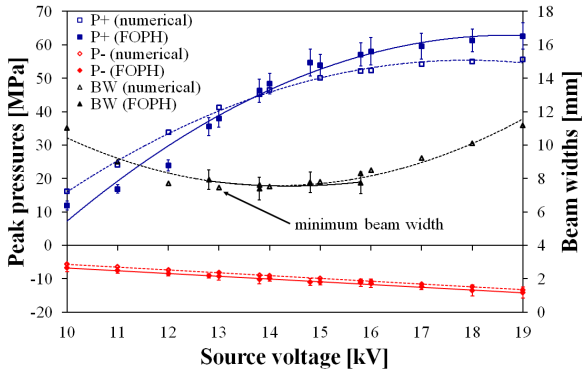


FIGURE 5.11. Comparison of peak positive pressure (P_+), peak negative pressure (P_-), and beam width for the original lens over the dynamic range of the lithotripter. Polynomial fits are also shown (dotted for experiment and solid for numerical).

at least 5% error [**115**]. Since FOPH measurements were used to create the input, the numerical model is considered to carry the same degree of uncertainty.

TABLE 5.2. Comparison of lithotripter parameters calculated from experimental and numerical pressure profiles at the focus for the original lens design. Energy subscripts +1, −1, and +2 refer to the first compressive, first tensile, and second compressive wave, respectively. $R_h = 6$ mm was used for all pulse energy calculations. SV is source voltage and BW is beam width.

	SV (kV)	P_+ (MPa)	P_- (MPa)	t_+ (μs)	t_- (μs)	t_r (ns)
Experimental	13.8	46.3	-10.2	1.62	3.49	145.0
	15.8	56.4	-11.0	1.59	3.31	16.2
Numerical	13.8	45.2	-8.9	1.57	3.40	50.2
	15.8	52.2	-10.6	1.63	3.69	37.3
	SV (kV)	BW (mm)	E_{+1} (mJ)	E_{-1} (mJ)	E_{+2} (mJ)	E_{total} (mJ)
Experimental	13.8	7.4	33.2	17.2	1.5	53.2
	15.8	7.5	51.7	18.3	4.3	74.7
Numerical	13.8	7.6	30.1	13.5	1.6	45.3
	15.8	8.3	48.0	20.3	2.8	71.5

5.3.3. Prediction of new lens parameters. In this section the model is shown to accurately predict pressure profiles near the focus with the new lens design. This model was developed and its parameters were established using the original lens geometry. For modeling the shock wave focusing produced by the new lens, the only model parameters

that are changed govern the geometry of the lens. All other aspects of the model remain the same. The new lens geometry is tested using 15.8 and 16.8 kV input, as opposed to the lower amplitude input used for the original lens. The interference from the delayed wave caused by the lens cut leads to reduced acoustic pressures at the focus. In order to compare pulses with similar focal pressures higher amplitude inputs are used.

This section will present data in the same manner as in the original lens section. Figure 5.12 shows pressure profiles along the propagation axis and in the focal plane for 15.8 and 16.8 kV input. As with the original lens there is good agreement between the overall shapes of the profiles. The model accurately captures the weakening and elongation of the tensile portion caused by the lens cut. Also noticeable in the radial plots, 5.12c and 5.12d, is the agreement of the suppressed secondary compressive wave. Except for a small spike in the numerical solution better overall agreement is seen compared to the same plots for the original lens. The numerical spike does not substantially contribute to the effective acoustic energy as seen in Table 5.3 and appears exaggerated in the propagation axis plots, 5.12a and 5.12b.

Figure 5.13 shows the peak positive and peak negative pressures in the focal plane, again for 15.8 and 16.8 kV input. These plots show that the model correctly predicts the increase in beam width caused by the lens cut. Figure 5.14 shows numerical peak positive pressure, peak negative pressure and beam width over the dynamic range of the lithotripter. The numerical parameter results appear to match the available experimental data.

Lithotripter parameters calculated from pressure profiles taken from the focus and focal plane are presented in Table 5.3. There is good agreement between model and experiment for P_+, t_+, and t_-. Error ranges from 1.1% to 8.3%. Error for P_- is slightly

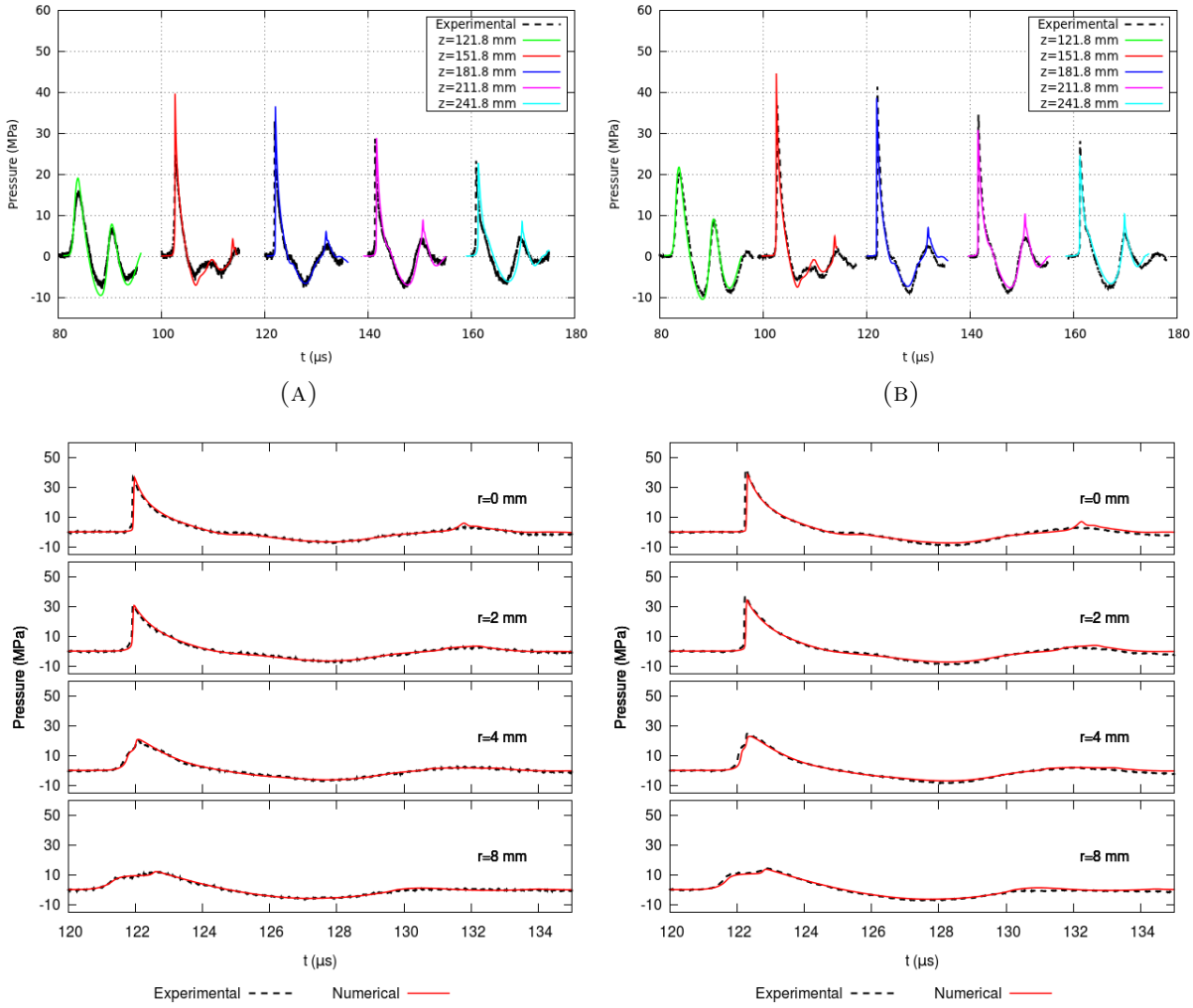
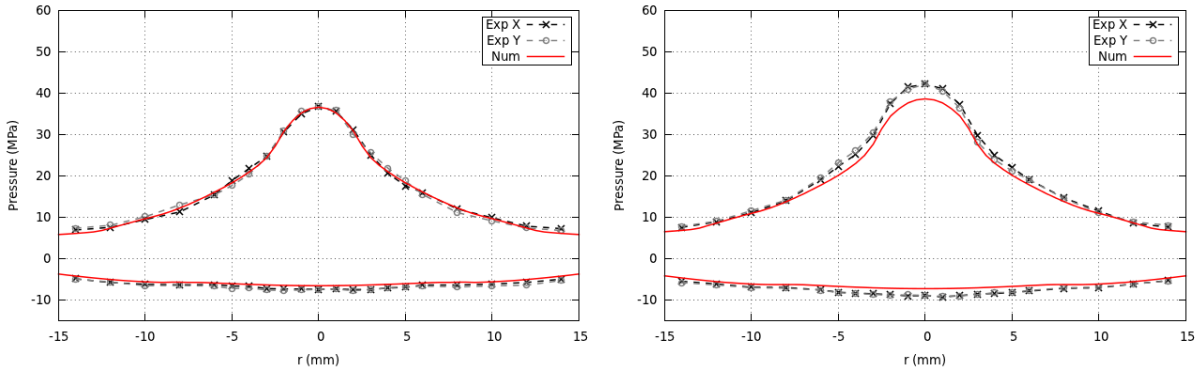


FIGURE 5.12. Plots of experimental and numerical pressure profiles along the propagation axis, $r = 0$, and in the focal plane, $z = 181.8$ mm for the new lens. (A) Propagation axis with 15.8 kV input. (B) Propagation axis with 16.8 kV input. (C) Focal plane with 15.8 kV input. (D) Focal plane with 16.8 kV input.

higher at 18.9% and 20.5%. As with the original lens, error is high for rise time presumably due to grid refinement. Beam width is captured very well at 0.9% and 3.8% error and acoustic energy has error ranging from 8.0% to 22.3%.

5.3.4. Inclusion of kidney stone simulant. Lastly, results which include a cylindrical kidney stone simulant in the domain are shown. The stone has height 7 mm and radius



(A) (B)

FIGURE 5.13. Plots of peak positive and peak negative pressure in the focal plane ($z = 181.8$ mm) for the new lens. Experimental data is recorded in four directions from the z-axis ($x+, x-, y+, y-$). Numerical data is mirrored across $r = 0$ to aid in visualization. (A) 15.8 kV. (B) 16.8 kV.

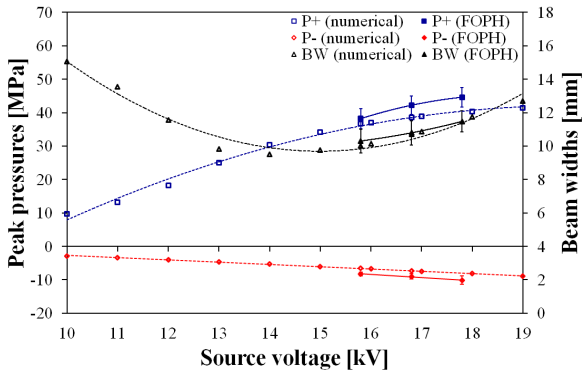


FIGURE 5.14. Comparison of peak positive pressure (P_+), peak negative pressure (P_-), and beam width for the new lens with available experimental data over the dynamic range of the lithotripter. Polynomial fits are also shown (dotted for experiment and solid for numerical)

3 mm and is placed so that the proximal surface is at the geometrical focus of the lens, z =181.8 mm, as shown in Figures 5.1 and 5.3. Figure 5.15 shows a comparison of the maximum of the maximum principal stress over time and the damage for the original lens with 13.8 kV input and the new lens with 16.8 kV input. These inputs are used since they provide similar acoustic energy at the focus. The original lens appears to cause more damage in the stone, but this is expected in this case. While the acoustic energy is similar for $R_h = 6$ mm, the stone has a radius of 3 mm and so the smaller beam width from the

TABLE 5.3. Comparison of lithotripter parameters calculated from experimental and numerical pressure profiles at the focus for the new lens design. Energy subscripts +1, −1, and +2 refer to the first compressive, first tensile, and second compressive wave, respectively. $R_h = 6$ mm was used for all pulse energy calculations. SV is source voltage and BW is beam width.

	SV (kV)	P_+ (MPa)	P_- (MPa)	t_+ (μs)	t_- (μs)	t_r (ns)
Experimental	15.8	38.2	-8.3	1.82	3.11	51.8
	16.8	42.2	-9.0	1.80	3.13	18.6
Numerical	15.8	36.7	-6.6	1.80	3.07	44.7
	16.8	38.7	-7.3	1.89	3.24	40.9
	SV (kV)	BW (mm)	E_{+1} (mJ)	E_{-1} (mJ)	E_{+2} (mJ)	E_{total} (mJ)
Experimental	15.8	10.4	29.9	9.8	0.0	39.9
	16.8	10.8	38.7	12.1	0.0	51.3
Numerical	15.8	10.0	27.5	7.7	0.1	35.6
	16.8	10.7	34.0	9.4	0.1	43.9

original lens leads to more acoustic energy encountering the stone. Also recall that the greater efficacy of the new lens includes cavitation effects which are not modeled here. These stress and damage results are preliminary work meant to show the capabilities of the model. Nevertheless, this model gives similar distributions of maximum principal stress to that of Cleveland and Sapozhnikov [**22**] and Sapozhnikov *et. al.* [**111**]

5.4. Discussion

The multiphysics focusing model was applied to a refracting electromagnetic lithotripter and was validated by comparing to experimental data. Many computational models of the wave propagation and nonlinear shock wave formation in ESWL have been developed, but none up to now have modeled this common type of lithotripter. This is most likely due to the fact that the focusing occurs by refraction inside a solid lens compared to all other lithotripter types in which the focusing occurs in water, usually by reflection. This focusing type required a multiphysics approach in order to combine these two domains. Arbitrary boundaries between the elasticity and Euler regions in the model allow for

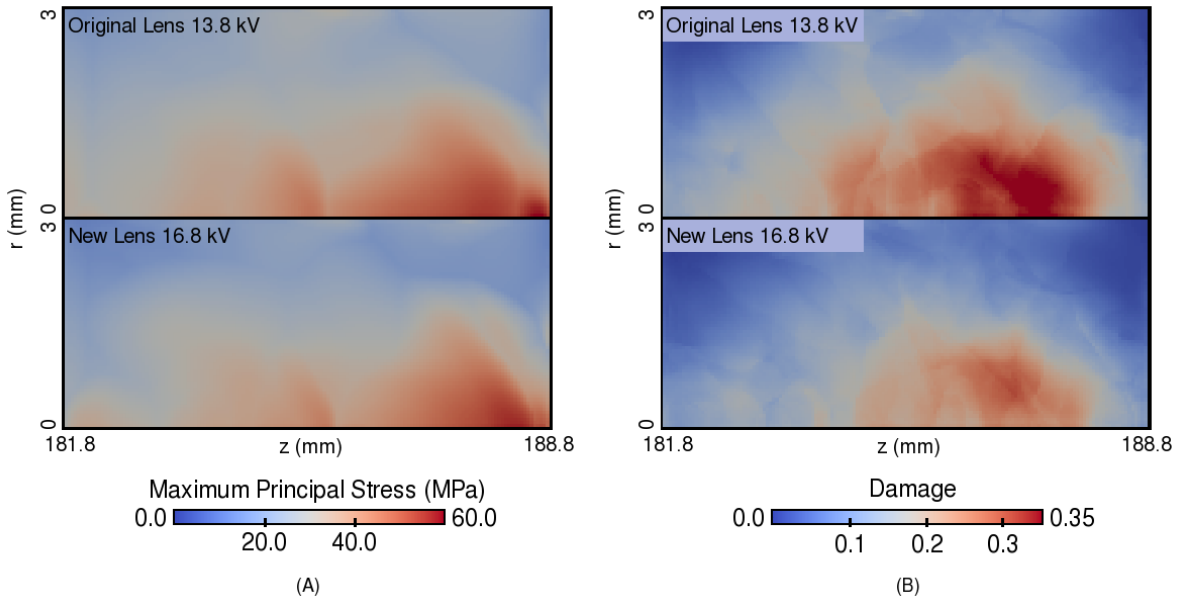


FIGURE 5.15. Comparison of the maximum over time of the maximum principal stress (A) and the damage (B) in a cylindrical kidney stone simulant with height 7 mm and radius 3 mm for the original lens with 13.8 kV input versus the new lens with 16.8 kV input after the pulse has passed completely through.

straightforward inclusion of lens modifications, different stone shapes, or any additional elastic solids in future work, e.g. a stone holder. A framework for BEARCLAW, allowing for these arbitrary regions, was developed. The model works seamlessly with AMR and could be applied to other domains requiring interaction of shock waves in fluids and linear elastic solids.

The focusing model was successfully validated against experimental results for a standard lens design. The predictive capabilities of the model were also shown by comparing to experiment with a modified lens. Numerical and experimental pressure profiles match well and most calculated lithotripter parameters fall within the error estimates of the FOPH and model input. With regards to the chosen lens modification, the model correctly predicts the weakening and lengthening of the tensile wave, suppression of the second compressive wave, and the increase in beam width caused by the lens cut. This

modified lens has been shown in other work to create pressure distributions similar to the HM3 which improve the efficacy of refracting EM lithotripters [**100, 136, 80**]. Further modifications of the lens, including sweeps of lens geometry parameters, and the potential effect on stones can now be tested without requiring the fabrication of physical lenses.

The model was developed with clinically relevant source voltages in mind, 13.8 to 16.8 kV. For better accuracy with stronger input a finer grid may be required. This would substantially increase the runtime of the computation which could risk its usefulness. Another option and possible future work is to implement a parallelized version. Other possible improvements include modeling the effect of bubbles due to cavitation in a manner like that of Tanguay [**120**] and modeling attenuation in the lens. A 3D version of the code may also be useful for modeling non-axisymmetric lenses.

CHAPTER 6

Multiscale Fracture Model

In this chapter a multiscale and probabilistic computational fracture model and its application to kidney stone breakup during lithotripsy are presented. First, the continuum scale model is described, which simulates the interaction of the shock wave and stone by solving the heterogeneous linear elasticity equations. Stress distributions are tracked during the computation. An anisotropic damage variable is also included to inform the continuum equations of unresolved fracture. Finally, the geometry of the stone is realistic, in that it is based on μCT imaging data of a kidney stone simulant. Pre-existing flaws in the stone are captured by the initial values of the damage variable.

The multiscale procedure extends the model to include a two scale interaction between the continuum scale model and a mesoscale model. The mesoscale model incorporates a structure, typically granular, into each continuum cell defined by a set of surfaces. Damage is allowed to accumulate on these surfaces based on probability distributions found from experiment. Following a description of the mesoscale, a discussion of the multiscale interaction is provided. Macroscopic fracture is modeled by allowing continuum cells to rupture in a manner dependent on the current stress and damage state. Finally, results and comparisons to experiment will be shown.

6.1. Continuum Model

The continuum scale model solves the 3D variable-coefficient linear elasticity equations using a finite-volume Riemann solver approach. These equations are used to model the shock wave transit through the stone and nearby water. The input to the model includes the Euler solution values recorded from the focusing model as well as the stone geometry. Many of the concepts in this model are similar to the focusing model described in Chapter 5, so that will be referred to often.

As in the focusing model, BEARCLAW, along with the wave propagation context provided by it, is used for this model. The implementation for the 3D linear elasticity equations in Cartesian coordinates is shown in Section 3.3.5 and validated in Section 3.6. As before, the water surrounding the stone is also modeled by the elasticity equations, but again the distances are very small so this is a reasonable approximation. In addition, capturing nonlinear effects here is not a priority. The baseline values for the elasticity parameters used for water and stone are shown in Table 6.1. The stone properties are based on 15:3 BegoStone, which has been shown to have similar material properties as calcium oxalate monohydrate kidney stones [**35**].

As mentioned in Section 3.5, this model required modifying BEARCLAW to include at least some of the transverse reflections normally not accounted for. Since there is a large ratio between the wave speeds in this problem, the standard implementation of BEARCLAW leads to instabilities at material boundaries. Accounting for all reflections leads to a large increase in the number of solves per interface. Fortunately, not all reflections (or full transverse solves) need to be accounted for, in order to increase the stability of the method. For results given in this work, reflections were added to the first

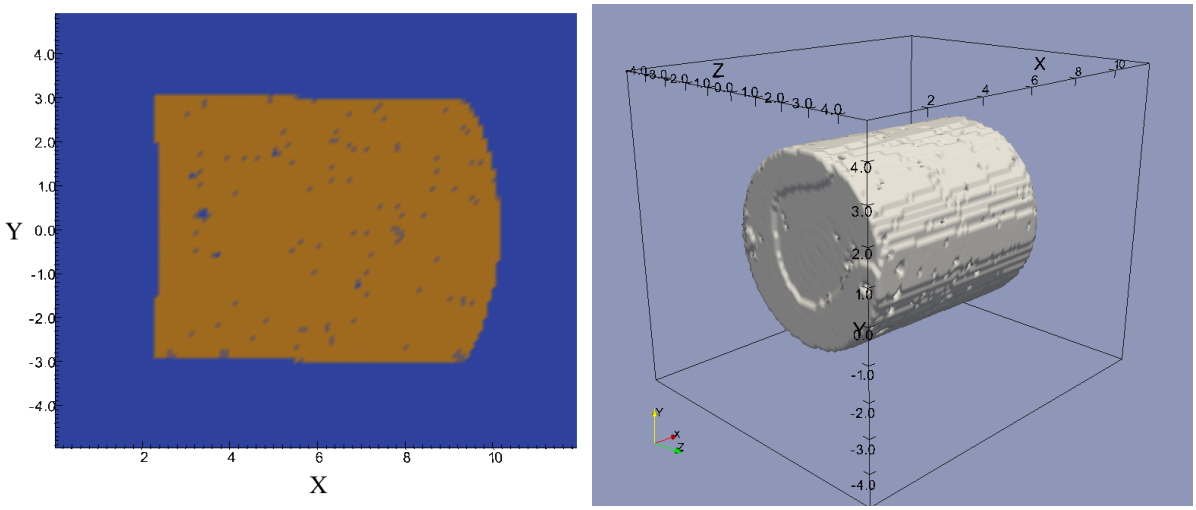
transverse solves, and the second transverse solves of the first reflections, but not for the second transverse solves of the first transmissions. In addition, the second modification proposed in Section 3.5 is also included to decrease total computational time. This modification combines the left-going and right-going fluxes entering the same cell before transverse corrections are applied.

TABLE 6.1. Material properties used in the 3D elasticity equations for water and kidney stone simulant (BegoStone).

			Water	**Stone**
Density	ρ	(kg/m^3)	1000	1995
P-wave speed	c_p	(m/s)	1489	4159
S-wave speed	c_s	(m/s)	10^{-4}	2319
First Lamé parameter	λ	(Pa)	2.217×10^9	1.305×10^{10}
Second Lamé parameter	μ	(Pa)	10^{-5}	1.0729×10^{10}

Coordinates have changed slightly from the focusing model. The x-axis is now the propagation axis, the main direction that the pulse travels in, and the approximate central axis of the cylindrical stones. Since the stone data is taken directly from the μCT images the stones may be slightly translated or rotated off this central axis. $x = 0$ mm in the fracture model corresponds to $z = 179.8$ mm in the focusing model. The stone is positioned so that the proximal surface is approximately at $x = 2$ mm, which corresponds to $z = 181.8$ mm in the focusing model and the geometric focus of the lens. Diagrams showing the basic setup of the domain and coordinates are provided in Figure 6.1.

The initial conditions are atmospheric conditions, where all the normal stresses are set to atmospheric pressure, 100 kPa, and the shear stresses and displacement velocities are set to 0. The pulse input data comes directly from output from the focusing model as described in section 5.2.6. The output data is from a 1D slice of the 2D axisymmetric focusing model domain and contains Euler solution values, $(\rho, \rho u, \rho v, \rho E)$, over time for each point on the slice. This data must be transformed from the 1D form to a 2D form.



(A) (B)

FIGURE 6.1. Diagrams of the fracture model domain. All values are in millimeters. The stone shown here corresponds to the 7 mm cylindrical stone 1 from the experimental results. (A) A slice of the 3D domain, at $z = 0$. (B) A contour of the edge of the stone within the full domain.

This is accomplished by drawing values from the data and placing them in a circular form over the $x = 0$ input boundary. The radius value is simply found from the cell coordinates, $r = \sqrt{y^2 + z^2}$, as $r = 0$ in the input data corresponds to $(y, z) = (0, 0)$ on the input boundary. Values are linearly interpolated from the two closest times and radial positions in the data.

If an incoming pulse completes before the simulation finishes, the input boundary at $x = 0$ is switched to a zero order extrapolation outflow boundary condition. An option is provided to allow for multiple pulses to enter the domain, since in experiments many pulses are required to fracture the stone. In the experimental procedure there is approximately a one second delay between shocks. The reason for the delay is mainly to let bubbles from cavitation subside, so as not to interfere with the subsequent shock. It is assumed that any stresses within the stone which lead to initial fracture have long since subsided after one second [**132**]. The largest stresses within the stone occur only a

few microseconds after the shock enters the stone. For this reason the stone simulations are run only slightly longer than this, approximately 16 μs per pulse, to ensure that the important stresses are captured but to limit unnecessary computational work. To approximate the long delay in experiment, the domain is wiped and atmospheric conditions are restored before the next pulse enters. The boundary conditions at all other boundaries are zero-extrapolation ouflow to simulate the stone being suspended in water.

6.1.1. Principal Stresses. As discussed in Section 2.3 the principal stresses are often important factors in determining whether a crack forms or propagates. In this model the maximum principal stress is used as an indicator of when damage or fracture will occur, as well as input into the damage growth law. In other words, if the maximum principal stress is high enough, then it is assumed that there may exist enough tension to either cause damage, crack formation, or crack propagation. Therefore, the principal stresses along with their directions are computed at every time step in every finite volume cell.

To compute these values the iterative QR eigenvalue algorithm is used to decompose the stress tensor into eigenvalues and eigenvectors. Recall that the eigenvalues are the principal stresses and the eigenvectors are the directions in which the principal stresses act. Since the stress tensor is a 3×3 matrix this algorithm does not significantly slow the overall computation even when computing at every grid cell.

6.1.2. Anisotripic Damage Variable. In this model a vector representation of damage is used where a damage value is stored for each of the three directions in each finite

volume cell. This can be thought of as a simplification of the second-order tensor representation of damage in which only the normal directions are retained,

$$\mathbf{D} = \begin{pmatrix} D_x & 0 & 0 \\ 0 & D_y & 0 \\ 0 & 0 & D_z \end{pmatrix}. \tag{6.1.1}$$

This representation is a natural way to implement anisotropic damage within the numerical scheme used here because of the dimensional splitting. For example, when computing the waves in the x-direction the corresponding damage variable, D_x, is used to affect the material parameters. All components of the damage, D_x, D_y, and D_z, have the range $[0, 1)$, but here a limiting critical value, D_c, is used as a maximum for each damage variable, so that $0 \leq D_i \leq D_c < 1$ where $i = x, y,$ or z.

This representation of damage is chosen because the damage within the stone is not expected to be isotropic. From preliminary runs of the algorithm, which only included solving the elasticity equations, it was seen that the direction of the maximum principal stress in a cylindrical stone was typically parallel to the x-axis. In addition, the corresponding experimental results showed that the cracks which caused the initial break of the stone ran perpendicular to the x-axis, which is expected from the orientation of maximum principal stress found during the computations. Since most of the crack surfaces should form perpendicular to the x-axis, isotropic damage is not correct. Waves propagating in the x-direction should be affected to a greater degree by the unresolved damage than waves propagating in either the y or z-directions. It is proposed then, that anisotropic damage is a more accurate representation of the unresolved fracture.

Since the damage in each direction is computed independently, the affect of the damage variables on the material parameters can be derived using the isotropic damage variable discussed in Section 2.2. Equation (2.2.7) shows how to incorporate damage into Hooke's law, which is the constitutive equation that defines the linear elasticity equations. Effectively, the addition of the isotropic damage variable into Hooke's law transforms the Young's modulus,

$$E := E\,(1 - D)\,. \tag{6.1.2}$$

This means that the stiffness of the material decreases with increased damage. Rederiving the elasticity equations with the damage variable results in it being multiplied into the matrix in equation (3.3.6). This equation becomes

$$\begin{pmatrix} \sigma^{xx} \\ \sigma^{yy} \\ \sigma^{zz} \\ \sigma^{xy} \\ \sigma^{xz} \\ \sigma^{yz} \end{pmatrix} = (1 - D) \begin{pmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\mu \end{pmatrix} \begin{pmatrix} \epsilon^{xx} \\ \epsilon^{yy} \\ \epsilon^{zz} \\ \epsilon^{xy} \\ \epsilon^{xz} \\ \epsilon^{yz} \end{pmatrix}. \tag{6.1.3}$$

As shown by this equation, to incorporate damage into the elasticity equations the Lamé parameters are simply multiplied by $(1 - D)$. The following diagram shows how each of

the five material parameters are affected by a damage variable.

$$\rho \rightarrow \rho$$

$$\lambda \rightarrow \lambda(1 - D_i)$$

$$\mu \rightarrow \mu(1 - D_i)$$

$$c_p \rightarrow c_p\sqrt{1 - D_i}$$

$$c_s \rightarrow c_s\sqrt{1 - D_i}$$

where D_i is either D_x, D_y, or D_z depending on which direction is currently being solved. Density in this derivation of damage is not affected. In the stiff brittle stone in this work, damage is used to homogenize the breakage of molecular bonds at an unresolved microscale. A macroscopic change in density will not occur from these broken bonds. In addition, mass is not transferred between cells and so any cell within the rigid stone should retain its mass. In other studies and in other materials, for example a metal being pulled apart, one would expect the density of a representative cell to decrease with increasing damage as voids open.

During the computation the damage variables change and this change is based either on a damage growth law, as seen for the focusing model, or in the case of this model, on what occurs at the mesoscale and the chosen method to relate the two scales. These processes will be discussed in later sections. Damage should not decrease in the kidney stone simulants over the time scales involved in this application, so each damage variable is restricted by $\frac{dD_i}{dt} \geq 0$. The material parameters in the elasticity equations are therefore

functions of space and time and can be written as

$$\begin{aligned} \rho &= \rho_0(x,y,z) \\ \lambda &= \lambda(x,y,z,t) = \lambda_0(x,y,z)(1-D_i(t)) \\ \mu &= \mu(x,y,z,t) = \mu_0(x,y,z)(1-D_i(t)) \end{aligned} \tag{6.1.4}$$

A purely continuum based damage growth law is proposed in order to compare to the multiscale approach discussed later. This growth law is based on the maximum principal stress, a critical stress, and the direction of the maximum principal stress. This law is similar to the law used in the focusing model but includes anisotropy. The proposed law is given by

$$\frac{dD}{dt} = \alpha(\sigma_1 - \sigma_c)^s, \tag{6.1.5}$$

where D is the total damage in a finite volume cell, σ_1 is the maximum principal stress, $\sigma_c = 7.1$ MPa is a critical stress value, and α and s are material parameters. This law is only applied if $\sigma_1 > \sigma_c$, otherwise $\frac{dD}{dt} = 0$. Let $\hat{e}_1 = (e_1^1\ e_1^2\ e_1^3)^T$ be the eigenvector corresponding to (and therefore the direction of) the maximum principal stress. Then the anisotropic damage variables are updated by

$$\begin{pmatrix} D_x \\ D_y \\ D_z \end{pmatrix} = \begin{pmatrix} D_x \\ D_y \\ D_z \end{pmatrix} + D_{\text{add}} \begin{pmatrix} e_1^1 \\ e_1^2 \\ e_1^3 \end{pmatrix}, \tag{6.1.6}$$

where D_{add} is the total damage added to a single finite volume cell over a single time step according to Equation (6.1.5).

6.1.3. Stone Input. In order to best compare to experimental results, realistic stone simulants are included in the domain. The stone geometries are taken directly from the μCT images of the four stones described in Section 4.2.2, all at zero shocks. These μCT images provide data on the internal structure of the stone before it is subjected to any shocks. The grayscale data in the images is used to define the stone-water boundaries and to initialize the damage variables.

The data is converted from a set of grayscale Tiff images to input files for the computation using a MATLAB script. The original image files define a 3D rectangular region of grayscale voxels encompassing the stone. Each voxel contains an integer value from 0 (black) to 255 (white). The values in the stone typically range from 70 to 130, while any values under 70 correspond to air around the stone or air pockets in the stone. The script first decreases the size of the bounding rectangle by calculating the start and end of the stone in each direction. This reduces the size of the files, the number of files, and the amount of imaging artifacts away from the stone. The script then uses a set ratio, r, to scale the images down in size. The total 3D rectangle is divided into sets of $r \times r \times r$ blocks of voxels and then the average of the r^3 values is given to the corresponding voxel in the scaled down version of the image set. This again creates smaller images and fewer images, which leads to the stone input being more manageable. In addition, the image data inherently contains high frequency noise which is reduced by the averaging. Finally, for results presented in this work, where $r = 4$, the scaled down image data still retains a higher resolution than the model. For future work, perhaps with parallelism implemented, the model resolution could increase, warranting image data with $r = 2$.

The model uses this scaled down image data to define material parameters across the entire domain. Since the variable coefficient linear elasticity equations are solved

everywhere, these parameters define the stone within the water as well as any air pockets inside the stone. It is assumed that the stones within the experiment are submerged long enough that any air pockets are filled with water. Damage variables are also initiated using this data. For now, this model does not use AMR and so a single root-level grid is defined. The material parameters across this grid are therefore set only once at the beginning of the computation.

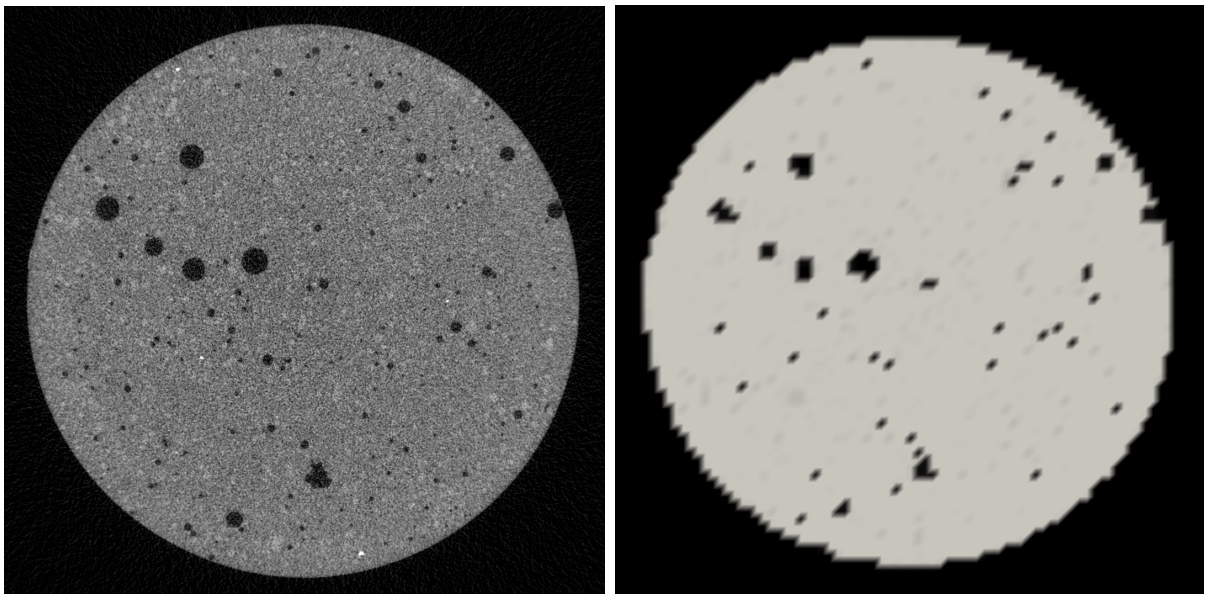
To set the parameters, the model loops over all finite volume cells. If the cell is not within the stone input data then it is assigned water values. If it is within the data then the grayscale value, gv, of the closest voxel is drawn, where $0 \leq gv \leq 255$. The material parameters are then set by

$$\text{Material} = \begin{cases} \text{Water} & \text{if} \quad gv < 70 \\ \text{Stone with damage} & \text{if} \quad 70 \leq gv \leq 110 \\ \text{Stone} & \text{if} \quad gv > 110 \end{cases}, \tag{6.1.7}$$

where the damage in the second case is initialized by a linear interpolation,

$$D_x = D_y = D_z = \frac{D_{init}}{70 - 110} \left(gv - 110 \right). \tag{6.1.8}$$

It is assumed that microscopic unresolved damage in the stone leads to darker pixels. No one direction should be predominant, so the initial damage is assumed to be isotropic and each direction is set to the same value. In this work, $D_{init} = 0.04$ is chosen. Example images, comparing a slice of the original μCT image data to a slice of the simulation domain, are shown in Figure 6.2. The model representation of the realistic stone can also be seen in Figure 6.1.



(A) (B)

FIGURE 6.2. A single slice of a kidney stone simulant from both the μCT images (A) and the representation of the stone in the computational model (B). The stone shown is the 7 mm cylindrical stone 1.

6.1.4. Macroscale fracture. Finally, macroscale fracture is tracked in the simulation to determine the initial fracture of the stone. This is accomplished by rupturing finite volume cells, which here means changing material properties drastically relative to the small changes due to increasing damage. This causes the ruptured cells to become stress concentrators and leads to propagation of the macroscale crack. The rupture procedure is also anisotropic and dependent on the direction of the underlying damage.

The rupture criterion used here is based on a simple isotropic criteria [**66**],

$$\sigma_1 > \sigma_R (1 - D) \tag{6.1.9}$$

where σ_R is the tensile stress required to rupture a sample with no damage. This is extended here to incorporate anisotropic damage,

$$\sigma_1 > \sigma_R \left(1 - \tilde{D}\right), \tag{6.1.10}$$

where

$$\tilde{D} = e_1^1 D_x + e_1^2 D_y + e_1^3 D_z, \tag{6.1.11}$$

and e_1^i are components of the direction of the current maximum principal stress in the finite volume cell in question.

The effect of the rupture on the continuum equations is also anisotropic. Each cell is allowed to rupture once and the direction of the rupture, $\hat{R} = (R_x, R_y, R_z)$, is based on the direction of the underlying damage. This will be discussed for the multiscale procedure later, but in a solely continuum sense the rupture direction would be set to the damage vector, (D_x, D_y, D_z), and normalized. This will often be equivalent to setting the rupture direction to the current direction of the maximum principal stress, as damage often accrues based on that. In other cases, including the case of underlying anisotropic microstructures this may not be correct, and so the direction of the damage is used.

The material parameters for the wave propagation solves are affected by the anisotropic rupture in a similar way as the anisotropic damage. When computing in the x-direction, for example, the Lamé parameters for a ruptured cell are changed by

$$\begin{aligned} \lambda &:= (1 - r_p R_x)\, \lambda \\ \mu &:= (1 - r_s R_x)\, \mu \end{aligned} \tag{6.1.12}$$

where r_p and r_s are chosen parameters, and the sound speeds are recalculated from the updated Lamé parameters. For this work, $r_p = 0.7$ and $r_s = 0.99$. These values essentially revert the material properties to that of water in the direction of the rupture. It is assumed if a cell ruptures, it is indicative of a macroscale crack, which is a void relative to the stone. Since the stone is soaked then this void is presumably filled with water and stone powder.

For example, if the rupture is purely in the x-direction, $\hat{R} = (1, 0, 0)$, then the cell will effectively no longer support shear waves in the x-direction. Again, this is because the assumed void that has opened in the yz-plane is filled with fluid that does not support shear waves. The p-wave speed in the x-direction is also significantly reduced, but the waves propagating in the y and z-directions are not affected.

It may be appropriate in future work to allow cells to rupture multiple times, especially in directions perpendicular to the original rupture, but for now cells are limited to one rupture. The rupture stress, σ_R, is found by comparing the simulation to experiment, whereas the critical stress value used in the damage law is based on the tensile strength of BegoStone found in the literature. A value of 90 MPa is used for the rupture stress in this work. A smaller value, like the critical stress, would lead to the immediate rupture of almost all the cells in the stone.

6.2. Mesoscale Model

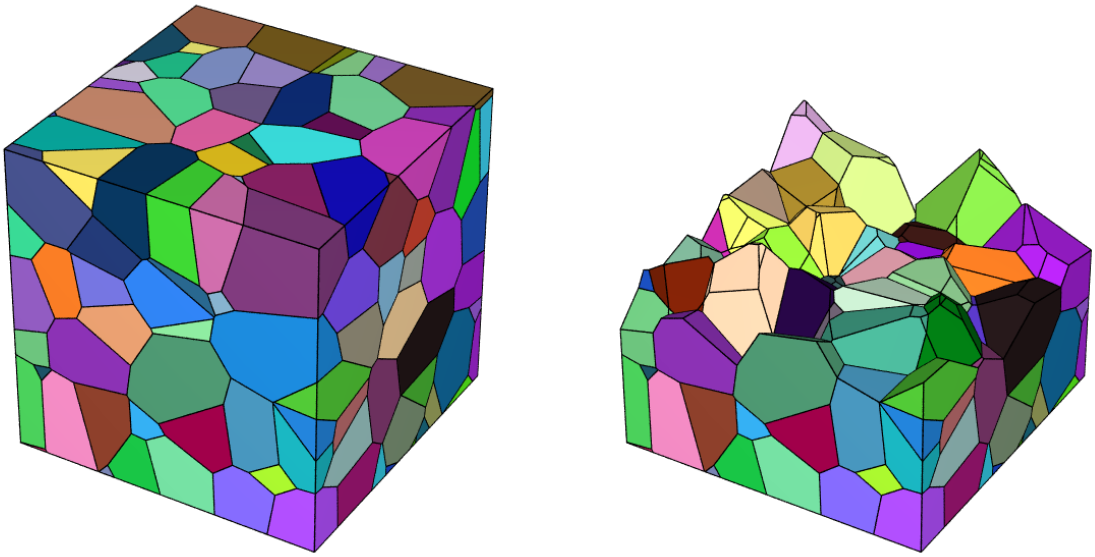
At the mesoscale of the model, a structure is defined by sets of surfaces within each continuum cell. For most results given in this work, these surfaces will represent a granular structure. While grain size may vary between materials this is a common structure at small scales of brittle materials and this assumption shouldn't limit this model. In

granular structures the molecular bonds between grains can be much weaker than bonds within grains. In addition, more imperfections exist along grain boundaries causing high tensile stresses in those areas. Therefore, it is expected that the vast majority of fracture will occur along grain boundaries. Here, this effect is approximated by assuming that fracture will only occur along grain boundaries.

To model a granular structure the software Neper is used [**101, 102**]. This software creates a random set of three dimensional grains within a cube shape. Each finite volume cell comprising the stone in the continuum model contains a Neper structure. The grains are all similar in size and each are defined by a set of surfaces. An example configuration produced by Neper is shown in Figure 6.3. This approximation differs from the real stone in that the surfaces defining the real grains would clearly not be flat. The idea, at this scale, is not to model the grains exactly but to provide preferred orientations for possible cracks. During the simulation, damage will only be allowed to accumulate along these surfaces.

Figure 6.4 shows, in a 2D example, a granular material and an overlaying 2D equivalent of a Neper structure. As seen in the figure, a single Neper surface (line in the 2D equivalent) represents the interface connecting two grains. This interface area may contain a different material than the grains, very small grains, or both. This type of structure could also be used to approximate a more crystalline solid without an interface material.

The grains in the model are larger than the true size of grains in BegoStone. Future work may address this through refinement of the continuum grid and utilization of Neper structures with more grains. In this work though, the method of the predefined fracture surfaces is tested as is. It is not an unreasonable approximation to use larger surfaces,



(A) (B)

FIGURE 6.3. Example output from Neper software. Both subfigures show the same instance. (A) The entire cube. (B) Grains with center above 0.5 are not shown in order to view inner grains.

as it would essentially correspond to collecting grains together. Each finite volume cell still contains many surfaces with many possible directions and so this shouldn't limit the direction or propagation of the cracks. In the next section, it will be shown that the Neper grains do approximate the size of grains in certain types of real kidney stone.

Each cell stores a list of information about the Neper structure it contains. This includes structure information such as a normal vector and surface area, SA, for each surface. A point located on each surface is also stored, as well as a list of every other surface that each surface connects to. Every surface also has an associated probability of fracture stored. For BegoStone, all these probabilities are equal, but as will be discussed later, this provides an option to prescribe an anisotropic microstructure within each cell. As a simple example consider a cell with surfaces with normals in a particular direction having a higher probability of damage accumulation than other surfaces. Finally,

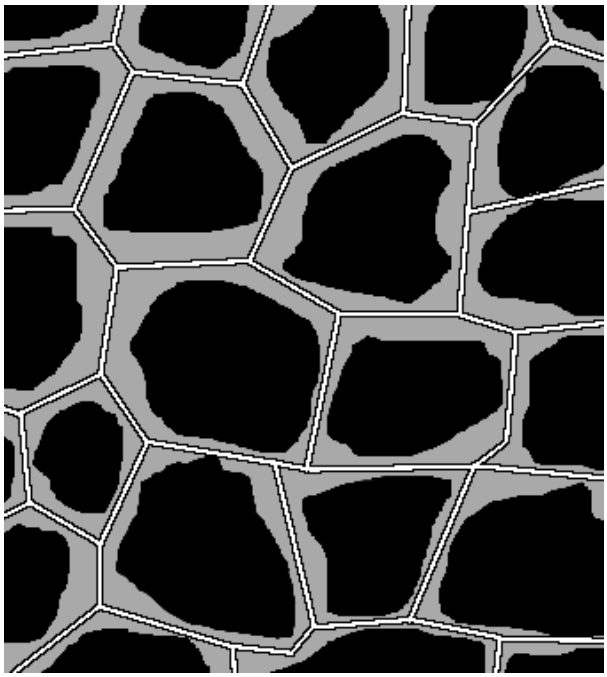


FIGURE 6.4. Diagram showing a 2D example of how the Neper structure is used to represent grains. Black represents grains. Grey represents the interface between grains. The white lines illustrate a 2D version of the Neper structure.

each surface contains information on the amount of damage accumulated. This includes whether the surface has been damaged at all, and if it has then by how much.

6.3. Multiscale Interaction

6.3.1. Continuum to Mesoscale. The continuum scale to mesoscale interaction is essentially a damage growth law. At the continuum level a damage growth law is updated in every finite volume cell at every time step,

$$\frac{dD}{dt} = \alpha\, Y\, (\sigma_1 - \sigma_c)^s\,, \tag{6.3.1}$$

if $\sigma_1 > \sigma_c$, where Y is a random value drawn from a lognormal distribution and the remainder of the parameters are the same as in Equation (6.1.5). The parameters for this lognormal distribution are based on the experimentally found values of crack length

and width, as presented in Section 4.3, so $\mu = 4$ and $\sigma = 0.5$. The chosen material parameters are $\alpha = 3 \times 10^{-8}$ and $s = 1.4$. Equation (6.3.1) is updated to find the maximum total damage that could accumulate in the cell during one time step,

$$D_{\text{add}} = \Delta t \; \alpha \; Y \left(\sigma_1 - \sigma_c\right)^s . \tag{6.3.2}$$

The actual amount of damage that accumulates is a fraction of this total and depends on probabilities associated with each surface.

Surfaces within the cell's granular structure are selected randomly. Damage is then added to that particular surface if two conditions are met. The first is the intrinsic probability associated with that surface's fracture. For the basic case, all surfaces have equal probability but later these probabilities will define anisotropic microstructures. The second involves the direction of the surface with respect to the direction of the maximum principal stress. This probability is calculated by

$$P = \left|\hat{n} \cdot \hat{\sigma}_1\right|^k , \tag{6.3.3}$$

where k is a parameter to control the probability. Large values of k force the surface normal to be closer to parallel with the direction of the maximum principal stress. For most results in this work $k = 3$. In the implementation, two uniformly distributed random numbers are drawn and compared to the fracture probability and the direction probability, respectively. If both values are less than the respective probability then the surface is allowed to accumulate damage.

The amount of damage on a surface will be referred to as the width of the surface. In this way, the volume found from multiplying the surface area and the width will

intuitively give the surface's contribution to the total damage of the cell. In actuality though, since the surface is representative of a number of molecular bonds, the width is better thought of as a measure of the number of broken bonds. If a surface is selected to have damage accumulate then the width of surface i is increased by

$$W_i := W_i + \beta Y, \tag{6.3.4}$$

where Y is a random number drawn from a lognormal distribution with parameters $\mu = 4$ and $\sigma = 0.5$, and $\beta = 0.01$ is a scaling parameter.

Let D_T represent the total damage accumulated during a a time step for one cell. So at the beginning of a time step $D_T = 0$. Surfaces are selected and damage is added until $D_T \geq D_{\text{add}}$. The process progresses in the following way:

- A surface, i, is randomly selected and the potential new damage is calculated as $D_P = \beta \times Y \times SA_i$,
- If the surface passes both probability tests then damage is accrued on surface, i.e. Equation (6.3.4) is updated and $D_T := D_T + D_P$,
- If the surface fails either tests then damage is not accrued on the surface but the total is updated, $D_T := D_T + D_P$.

The total damage is increased even for failing surfaces so that the method is more general, and can be applied to simulations with anisotropic structures. As an example, consider a cell, A, with only surfaces with normal near $(0, 1, 0)$ having a high intrinsic probability to accumulate damage, and a similar cell B but with $(1, 0, 0)$. Now consider the maximum principal stress to be oriented in the $(1, 0, 0)$ direction for both cells. The continuum level will calculate the same amount of damage for both cells, but it would be expected

that cell A would accumulate less damage than cell B. If the process above is followed then this will indeed be the case, since in cell A every surface will most likely fail one of the two tests. In cell B many surfaces will still fail but some will also pass both tests.

6.3.2. Mesoscale to Continuum. Informing the continuum equations of the underlying damage at the mesoscale involves updating the anisotropic damage variable. In each cell, at the beginning of every time step the three components of the damage variable are updated by

$$D_i = \sum_{j=1}^{b} A_j W_j \left|\hat{n}_{i,j}\right| \tag{6.3.5}$$

where i is the direction, x, y, or z, j counts over the b broken surfaces, A_j and W_j are the area and width of the j^{th} broken surface, and $\hat{n}_{i,j}$ is the i^{th} component of the normal vector of the j^{th} broken surface.

6.4. Results

This section presents results from the multiscale fracture model. The first results only solve the continuum level without damage accumulation in order to verify continuum level aspects of the model. The remainder of results utilize the full continuum-mesoscopic interaction. The majority of these results are for the BegoStone simulants taken from the μCT images. Some results with mesoscopic structures based on real kidney stones are also presented.

Figure 6.5 shows the progression of one shock through the stone. This result includes damage growth but is set low so the effect on the pulse is negligible. While this result does not show or imply fracture it is included as a general verification that the pulse

behaves as expected. Maximum principal stress is shown in the figure at several time steps.

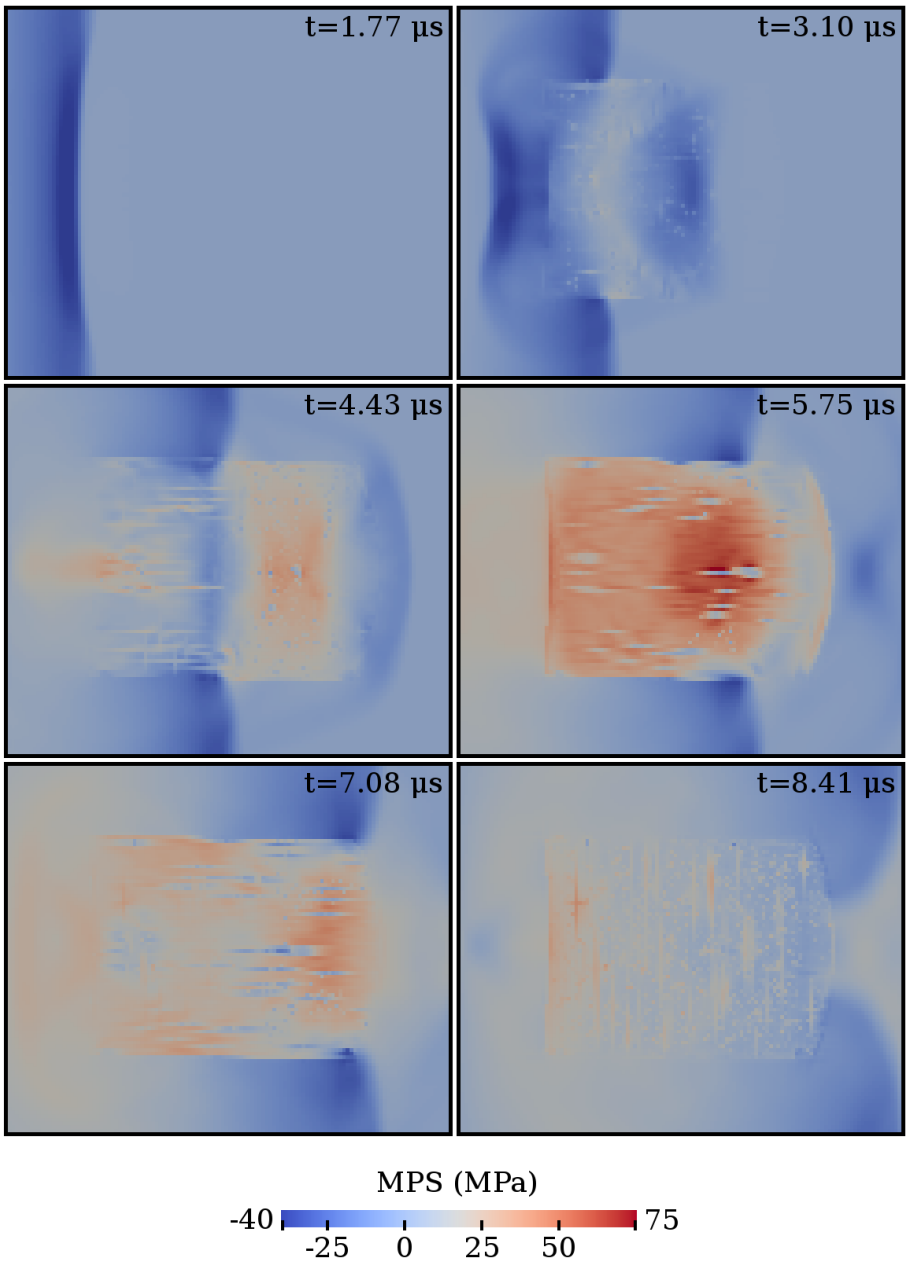


FIGURE 6.5. Time series of one shock encountering a stone in the multi-scale computational fracture model. Maximum principal stress is displayed on a slice of the 3D domain. The stone is the 7 mm cylindrical stone 1 and the shock is from 18 kV input. In the $t = 5.75$ μs frame the high tensile stress near one third the distance from the distal end of the stone is apparent.

6.4.1. Continuum Model. The first results of the kidney stone model are simply maximum pricipal stress (MPS) distributions within the stone and do not include any fracture or damage growth. This version of the model does include initial damage taken from the μCT images, with $D_{\text{init}} = 0.1$ from Equation (6.1.8). The model is tested on all four of the scanned stones. Figure 6.6 shows the maximum of the MPS over time in each stone over 2D slices of the domain.

Figure 6.7 shows potential locations and directions of cracks based on the MPS. While calculating the MPS, the model also calculates the direction of the MPS. Within the visualization program ParaView, circular glyphs are added where the maximum of the MPS over time occurs. They are oriented to be in the plane perpendicular to the direction of the MPS, which is also where cracks would be expected to propagate. Values of the maximum MPS over 50 MPa (45 MPa for the spherical stone) are candidates for the glyphs. Five glyphs are then randomly placed on the candidate cells. The figures also include a contour of the stone-water boundary. The glyphs aren't meant as true predictions of cracks, but as a visualization of the potential orientations and locations of cracks in later simulations.

6.4.2. Multiscale Model. This section describes results of the full continuum-mesoscopic interaction presented in this chapter. As in the previous section, results for all four scanned stones are shown. For 7 mm cylindrical stones 1 and 2 results for both 16 kV and 18 kV input are shown. For the remaining two stones (10 mm cylindrical and 10 mm spherical) only 18 kV input results are shown. For these results the spatial step $\Delta x = 100$ μm. Neper structures with 250 grains are used which leads to an average grain diameter of approximately 15 μm. BegoStone material properties are used and all

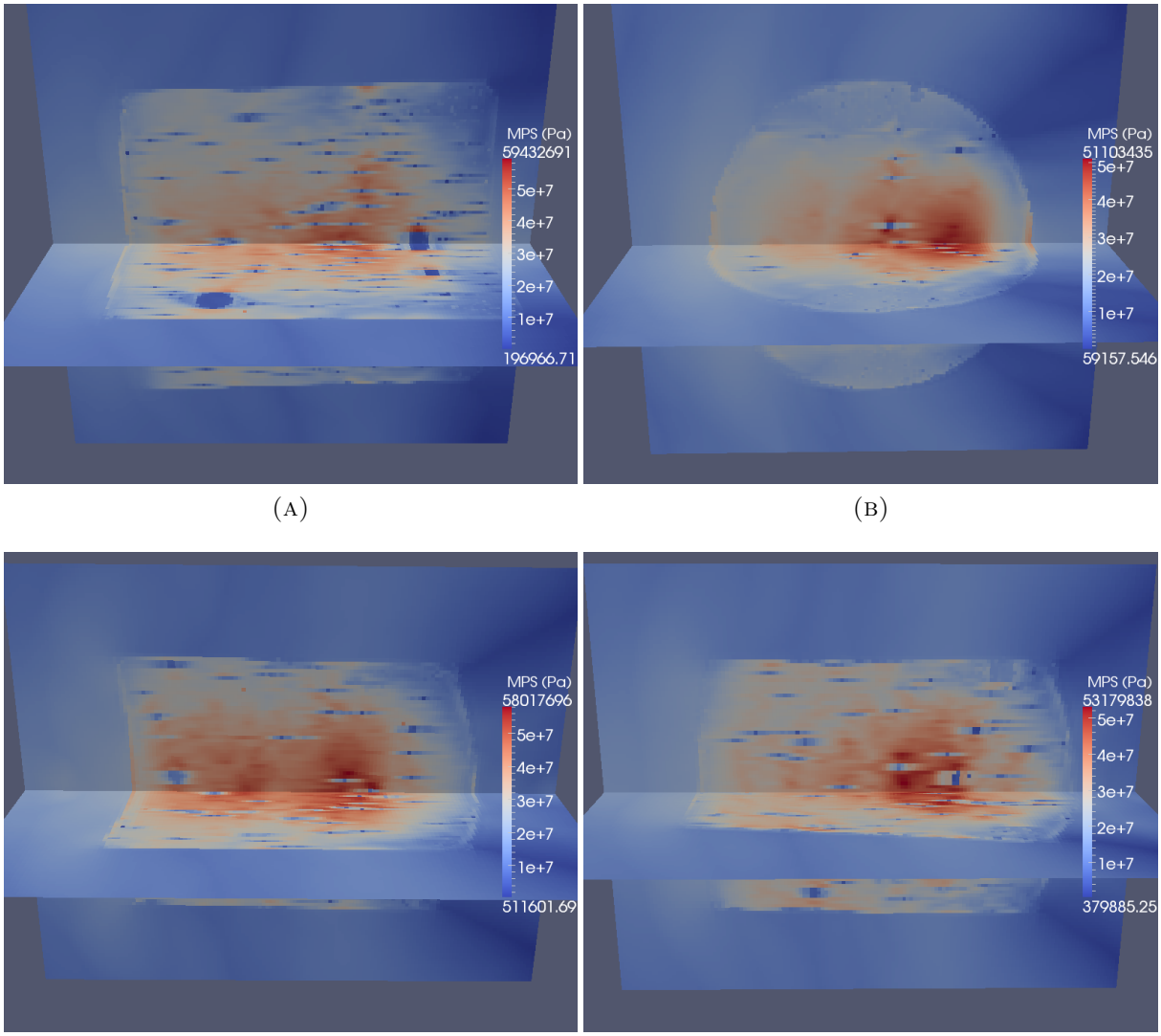
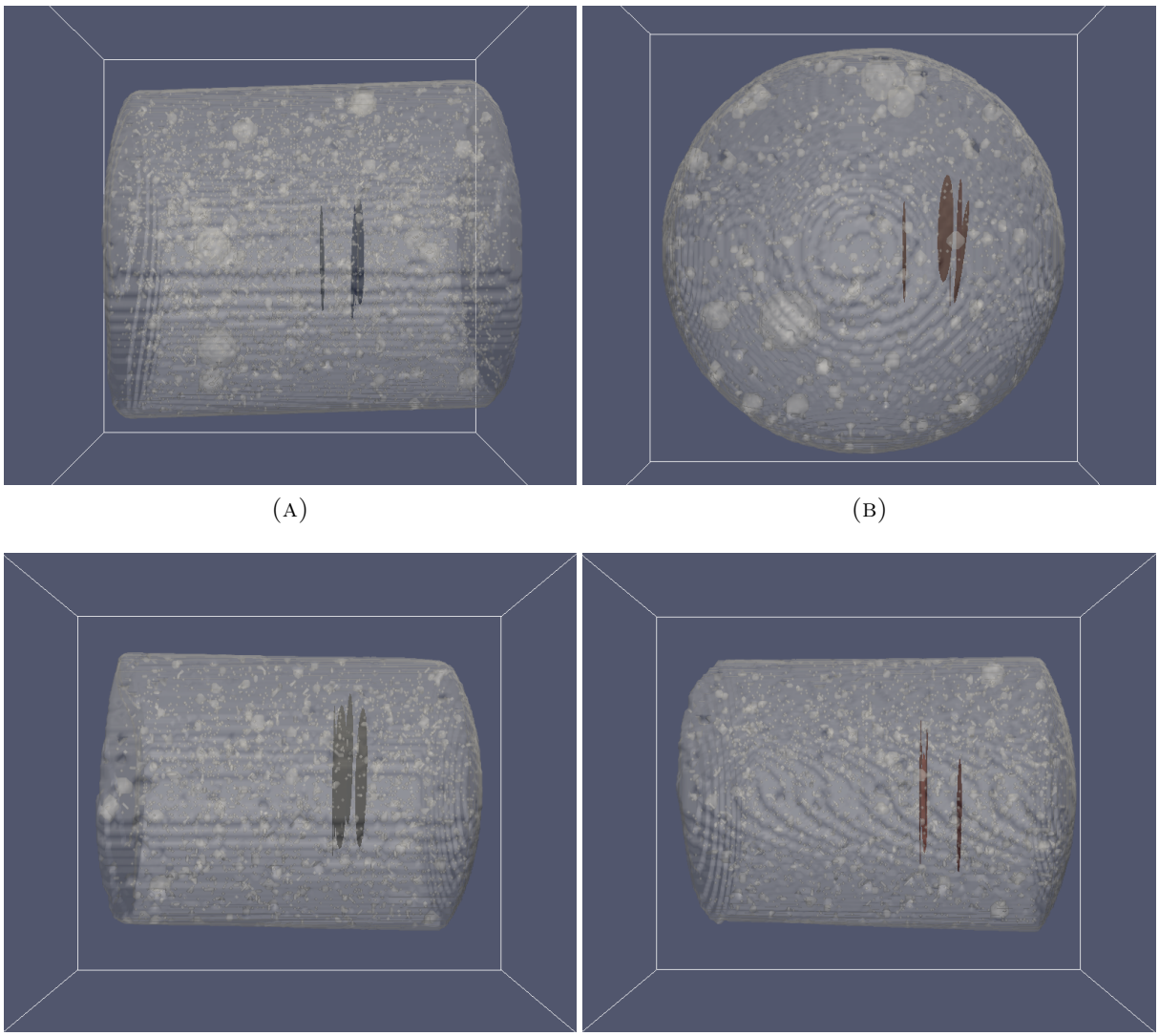


FIGURE 6.6. Maximum principal stress distributions in realistic stones with no damage or fracture growth law. (A) 10 mm cylindrical stone. (B) 10 mm spherical stone. (C) 7 mm cylindrical stone 1. (D) 7 mm cylindrical stone 2.

mesoscale structure surfaces have equal intrinsic probability. All other values used for these results were the values given in this chapter.

First, a time series of the damage and ruptured cells for 7 mm cylindrical stone 1 with 18 kV input is shown in Figure 6.8. The figures show the state of the stone after each shock, ending after 5 shocks. From the images it is reasonable to assume the stone



(A) (B)

(C) (D)

FIGURE 6.7. Potential locations and directions of cracks resulting from high maximum principal stress. (A) 10 mm cylindrical stone. (B) 10 mm spherical stone. (C) 7 mm cylindrical stone 1. (D) 7 mm cylindrical stone 2.

fractures after 4 or 5 shocks. While this value is lower than the 6 to 7 shocks expected it still shows how the damage which accumulates after the first shock reduces the stiffness of the stone and leads to subsequent fracture. The fracture also occurs in approximately the correct location along the propagation axis and progresses in the correct direction

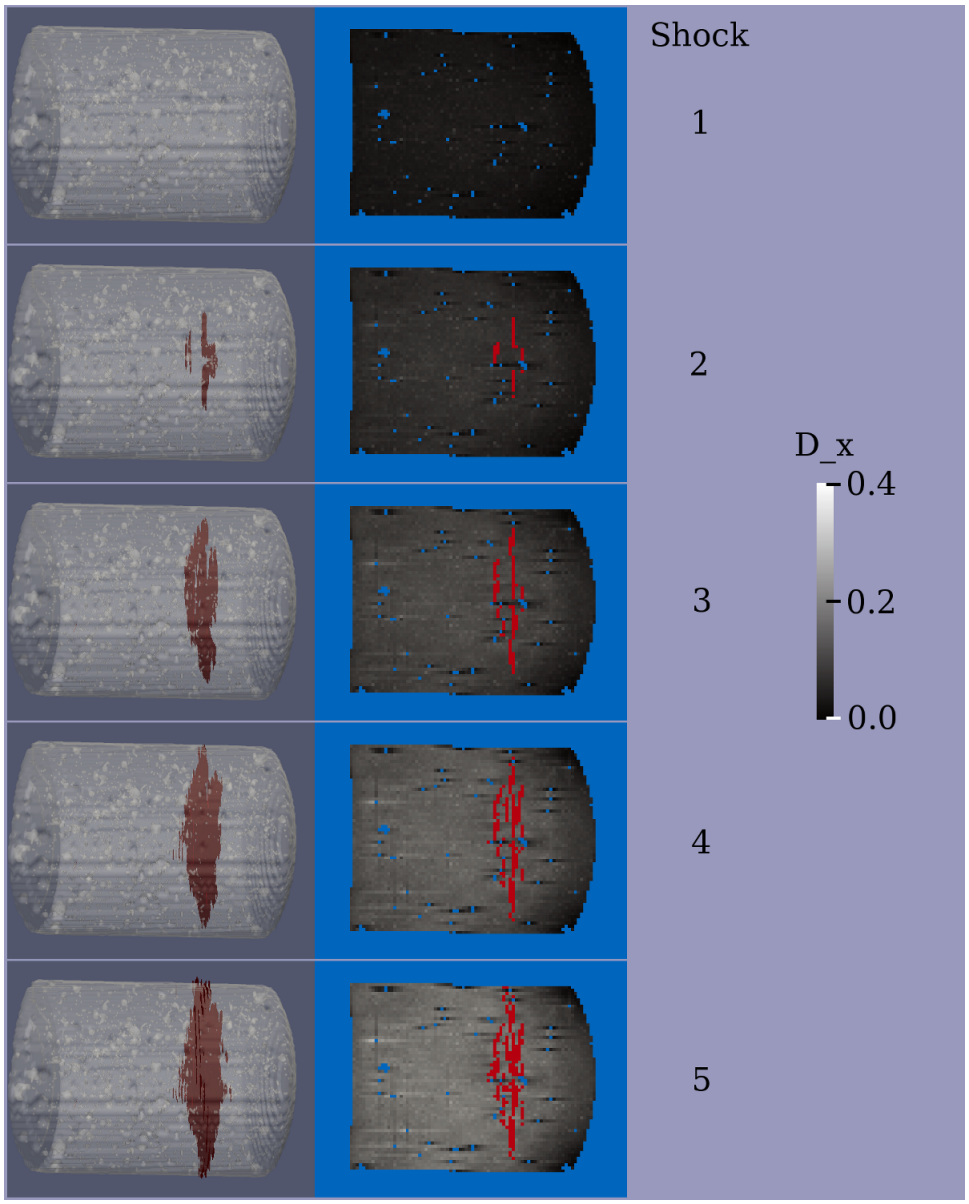


FIGURE 6.8. The state of the 7 mm cylindrical stone 1 with 18 kV shock input after shocks 1 through 5. The images on the left show a 3D contour of the stone-water boundary and flat circular glyphs denote ruptured cells. The glyphs are oriented in the direction of the rupture. The images on the right show the damage in the x direction along a slice at $z = 0$. Water cells are colored blue and ruptured cells are colored red.

perpendicular to the propagation axis. Figure 6.9 shows the typical fracture of a 7 mm cylindrical stone for comparison.

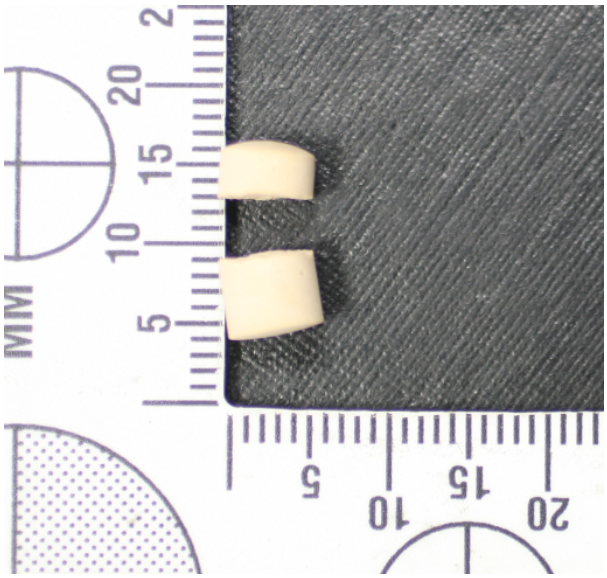
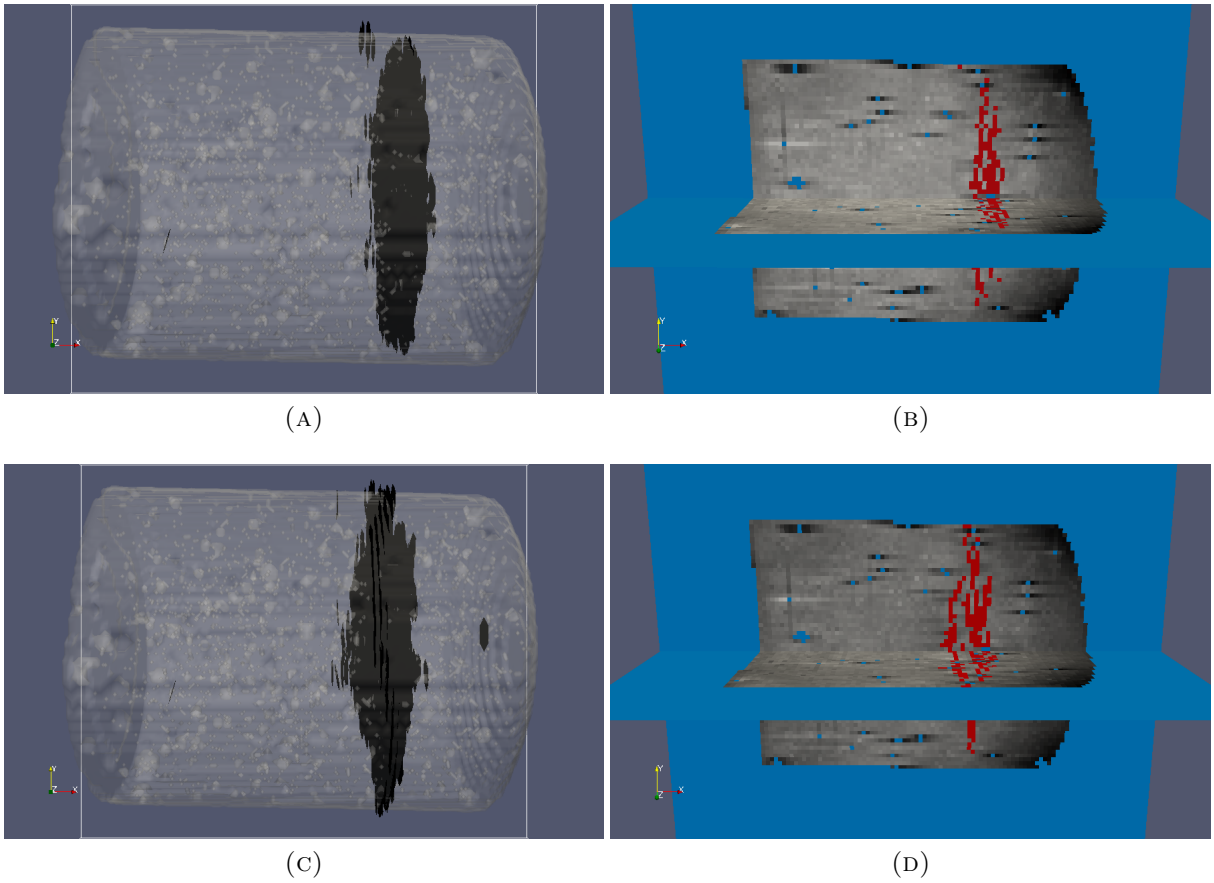


FIGURE 6.9. A 7 mm cylindrical BegoStone kidney stone simulant after initial fracture. Fracture reliably occurs in this stone geometry perpendicular to the propagation axis and approximately a third of the distance from the distal end of the stone. (Image by Jaclyn Lautz)

Figures 6.10, 6.11, 6.12, and 6.13 show final fracture configurations for the 7 mm cylindrical stone 1, 7 mm cylindrical stone 2, 10 mm cylindrical stone, and 10 mm spherical stone, respectively. Each fracture pattern is shown in two forms. The first is by glyphs oriented in the direction of the rupture along with a 3D contour of the stone. The second is two slices showing the damage and ruptured cells. Damage ranges from 0 to $D_c = 0.4$ and only the damage in the x-direction, D_x, is shown since this is the principal direction in which the damage develops. Table 6.2 summarizes the stones, kV input, and the shocks required for initial fracture. The stone was considered to be fractured if the ruptured cells reached within 3 cells of the stone-water boundary along slices in the y and z-directions.

Figures 6.14, 6.15, 6.16, and 6.17 show orthogonal μCT images of each stone's final image set for comparison to the model. The μCT images are arranged to match the

model figures, so that the shock would enter from the left. Each of these figures also contains a duplicate with the crack outlines in red to aid in viewing.

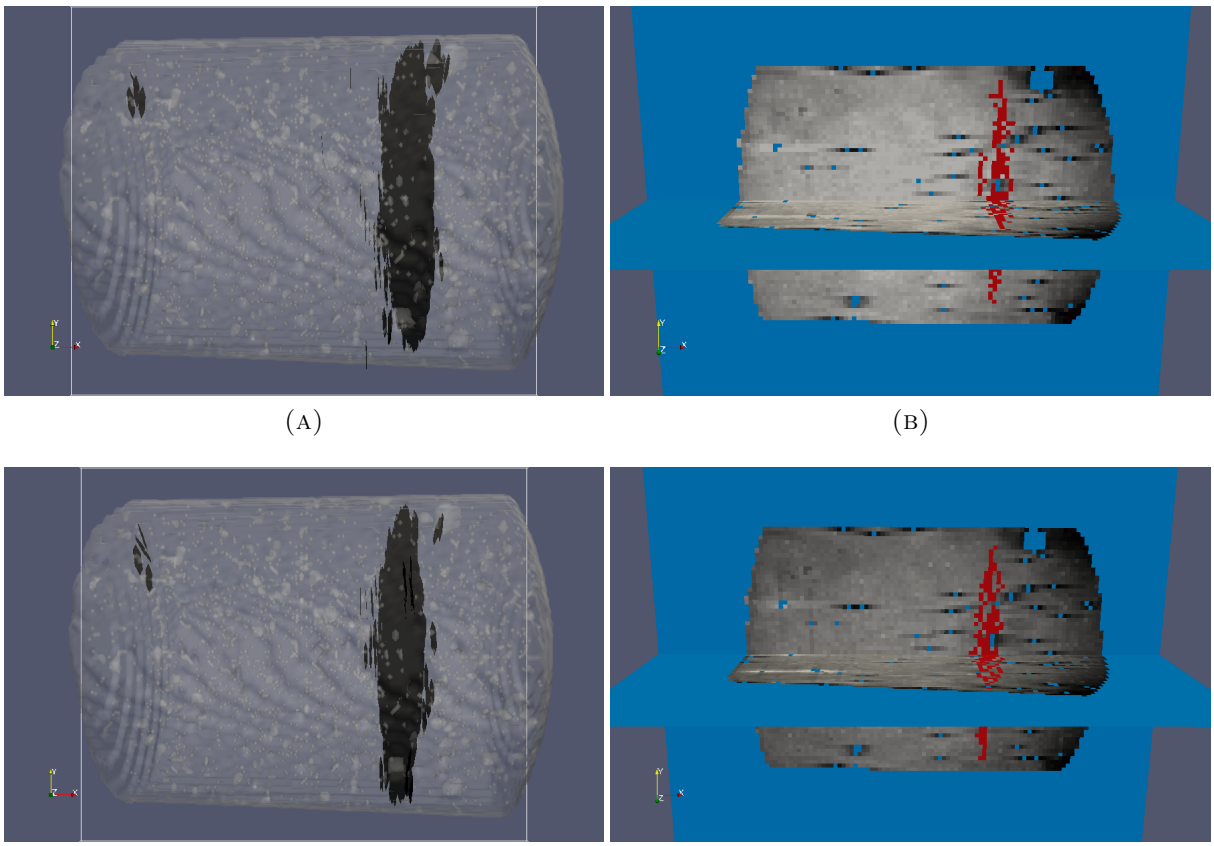


(A) (B)

(C) (D)

FIGURE 6.10. Final stone states of 7 mm cylindrical stone 1. (A) Glyphs and 3D contour for 16 kV and 8 shocks. (B) Damage in x direction and rupture on slices for 16 kV and 8 shocks. (C) Glyphs and 3D contour for 18 kV and 5 shocks. (D) Damage in x direction and rupture on slices for 18 kV and 5 shocks.

6.4.3. Real Kidney Stone Microstructure. This section describes results for two simulations with mesoscale structures based on real kidney stones. Both are given a general spherical shape with 8 mm diameter. The first simulation is based on a struvite stone which has been shown to have a granular structure with grain sizes of approximately 20-50 μm [**118**] and usually fractures along these granular surfaces [**138**]. The spatial step in the model is $\Delta x = 100$ μm and Neper structures with 50 grains are used leading to an

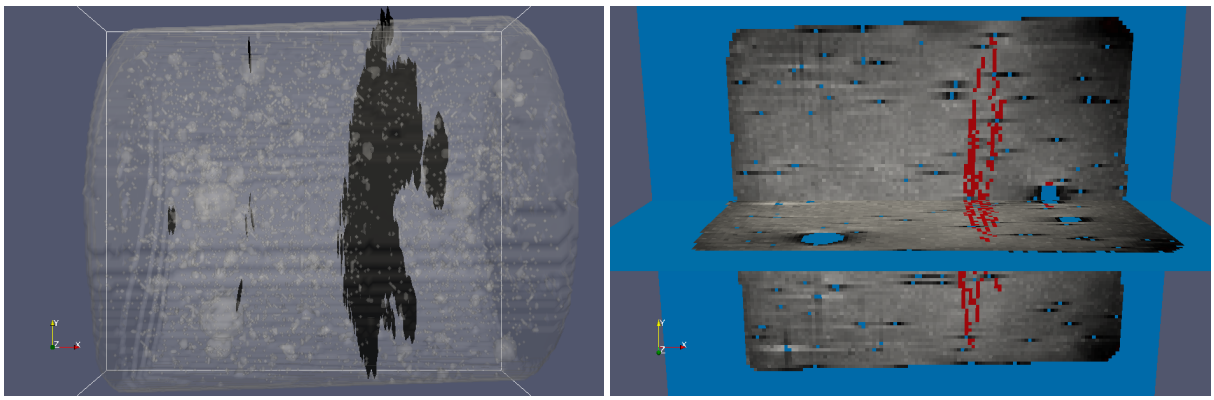


(A) (B)

(C) (D)

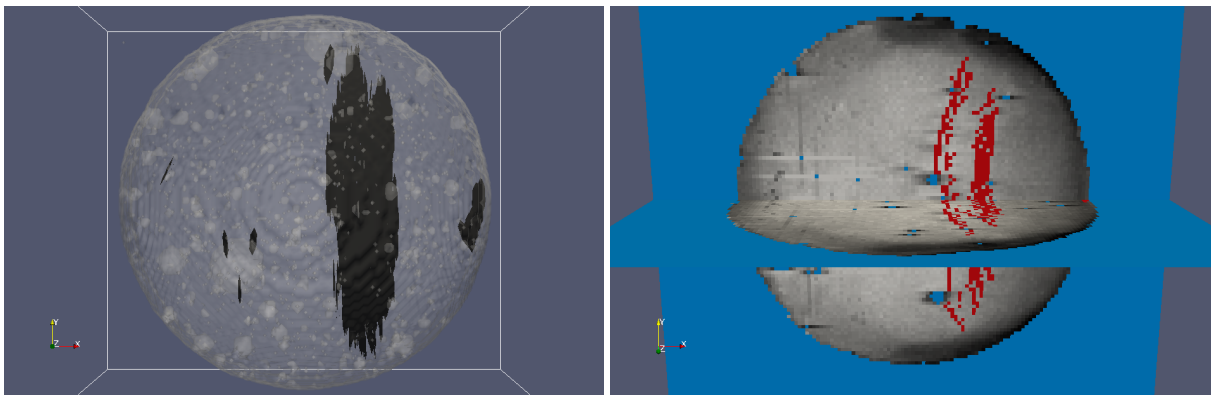
FIGURE 6.11. Final stone states of 7 mm cylindrical stone 2. (A) Glyphs and 3D contour for 16 kV and 9 shocks. (B) Damage in x direction and rupture on slices for 16 kV and 9 shocks. (C) Glyphs and 3D contour for 18 kV and 6 shocks. (D) Damage in x direction and rupture on slices for 18 kV and 6 shocks.

average grain diameter of approximately 27 μm. Struvite stones have material properties similar to Ultracal-30 simulants and so those material properties are chosen [**35**]. Since the wave speeds are closer to water, more energy from the shock enters the stone, and higher amplitude tensile stresses are observed. Therefore, for these test $\sigma_R = 95$ (MPa). Results from this simulation are shown in Figure 6.18 which, as before, shows glyphs in a 3D contour and damage and ruptured cells along slices of the domain. This simulation was run for 5 shocks at 18 kV input.



(A) (B)

FIGURE 6.12. Final stone states of the 10 mm cylindrical stone. (A) Glyphs and 3D contour for 18 kV and 5 shocks. (B) Damage in x direction and rupture on slices for 18 kV and 5 shocks.



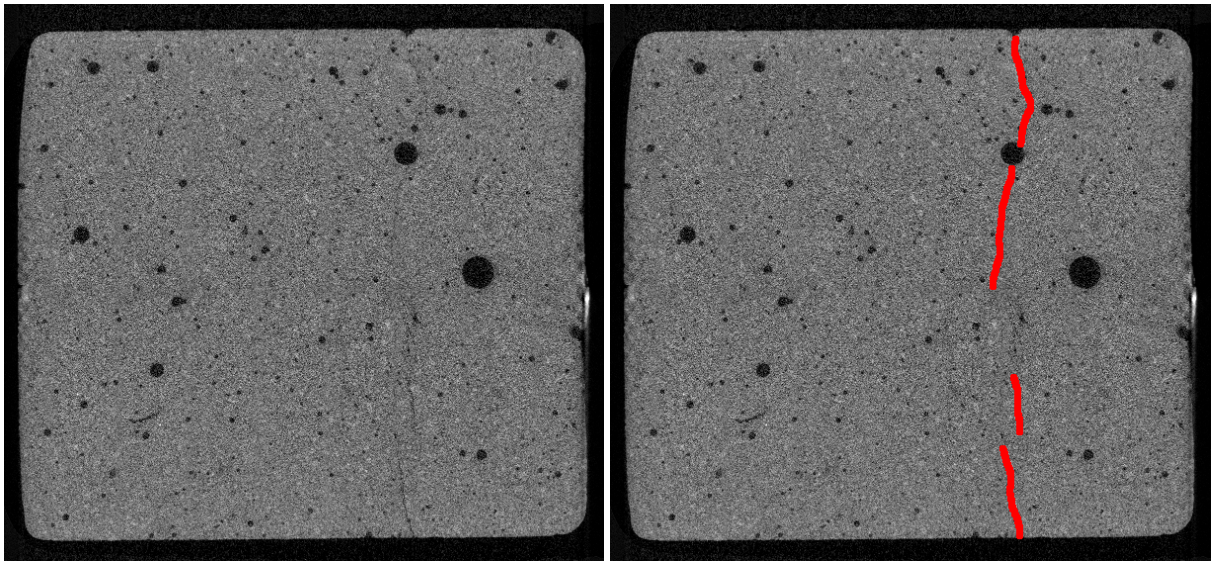
(A) (B)

FIGURE 6.13. Final stone states of the 10 mm spherical stone. (A) Glyphs and 3D contour for 18 kV and 9 shocks. (B) Damage in x direction and rupture on slices for 18 kV and 9 shocks.

The second simulation is based on a calcium oxalate monohydrate (COM) stone. This stone has similar material properties as BegoStone and so those properties are used for this test. COM stones, as well as uric acid stones, have a concentric laminated structure in addition to the crystalline structure of the grains [**98, 138**]. In this model Neper grains of approimately 15 μm diameter, as in the BegoStone model, were used, but the surfaces were given intrinsic probabilities based on their orientations. To approximate the concentric laminated structure in a spherical stone, surfaces with orientation aligned

TABLE 6.2. Number of shocks required to cause initial fracture in simulations with various stones and kV input. At 8 shocks the spherical stone did not seem to be quickly approaching full fracture.

Stone	kV	Shocks
7 mm cylindrical stone 1	14	14
7 mm cylindrical stone 1	16	8
7 mm cylindrical stone 1	18	5
7 mm cylindrical stone 2	14	14
7 mm cylindrical stone 2	16	9
7 mm cylindrical stone 2	18	6
10 mm cylindrical stone	18	5
10 mm spherical stone	18	8+



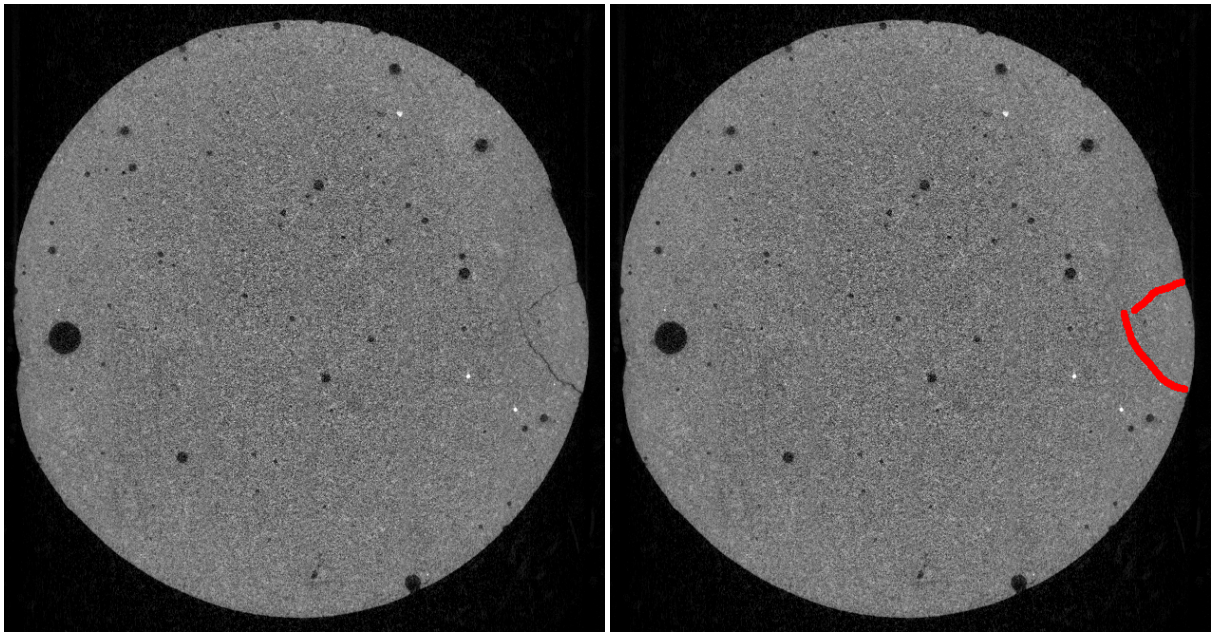
(A) (B)

FIGURE 6.14. Orthogonal slice of μCT image data for the 10 mm cylindrical stone. (A) Original image. (B) Crack covered in red to aid in viewing.

with a sphere centered at the stone center were given higher probabilities to accumulate damage. The intrinsic probability is given by

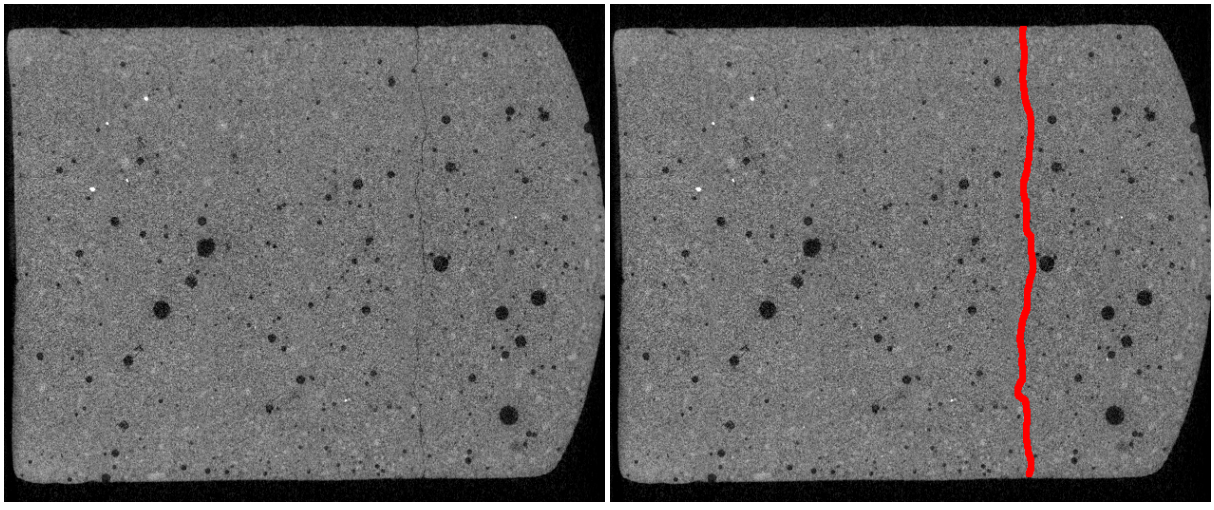
$$P = |\hat{p}_1 \cdot \hat{n}|\,, \tag{6.4.1}$$

where $\hat{p}_1$ is a unit vector from the stone center towards the surface and $\hat{n}$ is the unit normal vector of the surface. For this simulation $k = 1$ in Equation (6.3.3) and $\alpha =$



(A) (B)

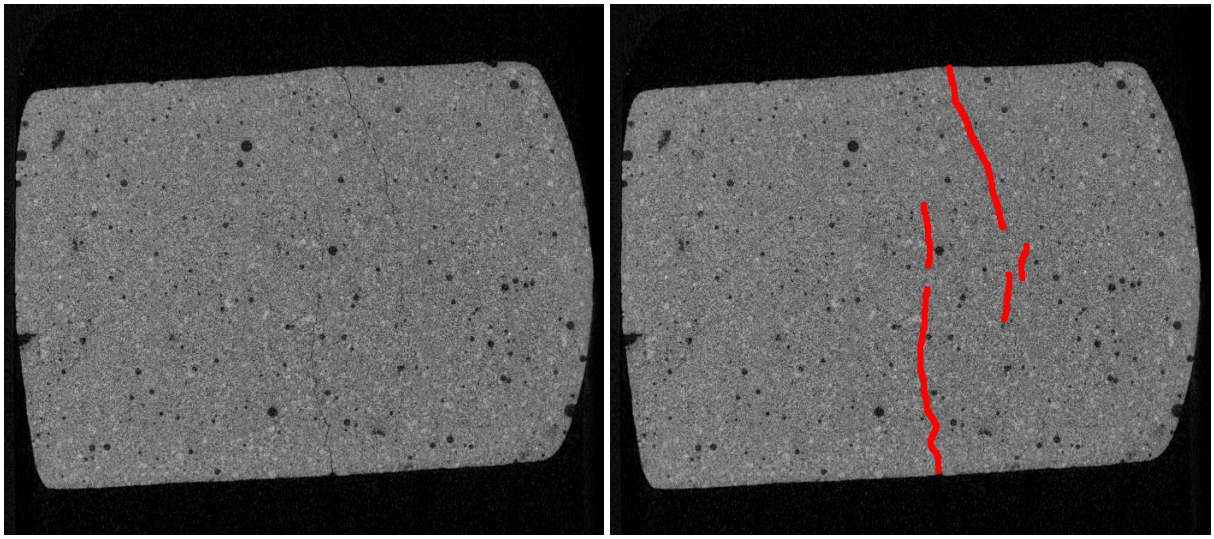
FIGURE 6.15. Orthogonal slice of μCT image data for the 10 mm spherical stone. (A) Original image. (B) Crack covered in red to aid in viewing.



(A) (B)

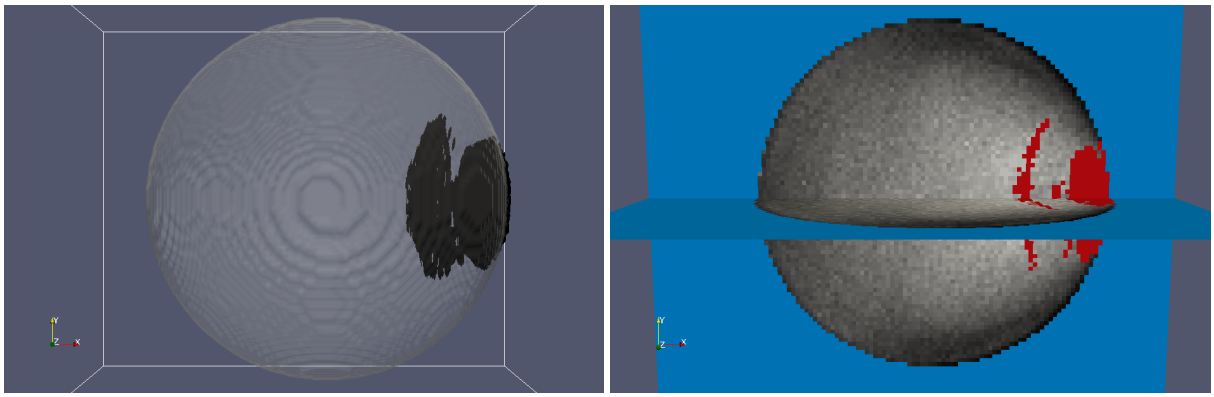
FIGURE 6.16. Orthogonal slice of μCT image data for 7 mm cylindrical stone 1. (A) Original image. (B) Crack covered in red to aid in viewing.

10^{-7} in Equation (6.3.2). Results from this simulation are shown in Figure 6.19. This figure displays damage in all three directions to illuminate the effect of the mesoscopic anisotropy. This simulation was run for 6 shocks at 18 kV.



(A) (B)

FIGURE 6.17. Orthogonal slice of μCT image data for 7 mm cylindrical stone 2. (A) Original image. (B) Crack covered in red to aid in viewing.

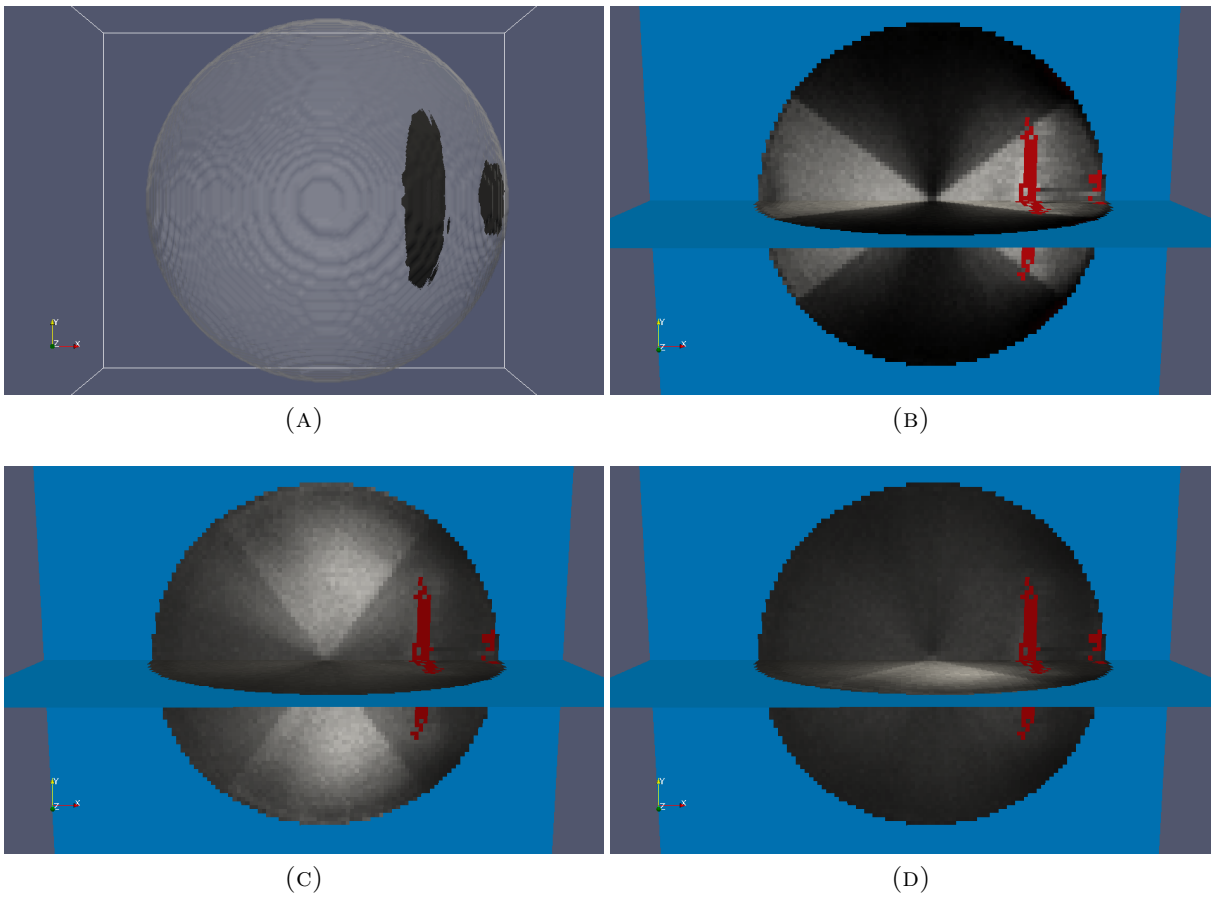


(A) (B)

FIGURE 6.18. Fracture state for the 8 mm spherical struvite approximation after 5 shocks at 18 kV. (A) Glyphs and 3D contour. (B) Damage in x direction and rupture on slices.

6.5. Discussion

This chapter presented a computational model able to predict the initial fracture of kidney stone simulants. This model incorporated a continuum-mesoscopic interaction which allowed for the inclusion of effects due to a mesoscopic structure. The mesoscopic



(A) (B)

(C) (D)

FIGURE 6.19. Fracture state for the 8 mm spherical COM approximation after 6 shocks at 18 kV. (A) Glyphs and 3D contour. (B) Damage in x-direction and rupture on slices. (C) Damage in y-direction and rupture on slices. (D) Damage in z-direction and rupture on slices.

structure represents a granular structure by default but through assigning probabilities to portions of the structure can represent other configurations.

The fracture model was applied specifically to the application of shock wave lithotripsy. With regards to the application domain, this model has advanced beyond the current state of the art in four distinct ways. The first two are somewhat trivial but add to the realism and therefore the potential accuracy of the method. First, a realistic shock wave, taken directly from the multiphysics focusing model, is used. The focusing model has been verified against experiment and therefore this incoming pulse can be assumed to

be realistic. Second, fracture can be computed in a realistic kidney stone simulant, with water pockets and initial damage, based on data from μCT images. Third, inclusion of a damage variable allows the model to make predictions of the number of shocks required for initial fracture of stones. Finally, the extension to anisotropic damage and mesoscopic interaction add additional accuracy and realism to the model. This also allows for simulation of real kidney stones such as COM and struvite and their associated mesoscopic structures.

The continuum results shown in Figures 6.5, 6.6, and 6.7 serve to validate the continuum aspects of the model such as the shock wave and realistic stone geometry input. The final distributions of the maximum of the maximum principal stress over time shown in Figure 6.6 match similar results in the literature [**22, 111**]. The calculated direction of the maximum principal stress, as illustrated in Figure 6.7, approximately match the direction and location of known initial fracture in cylindrical BegoStone simulants.

The first and second 7 mm cylindrical stone resulted in the fracture patterns shown in Figures 6.10 and 6.11. These show very good agreement in crack location and propagation direction with the μCT results in Figures 6.16 and 6.17. In addition, these results show good agreement with the number of shocks required for initial fracture as shown in Figure 4.2. The number of shocks for 18 kV may be one or two shocks too low, but the experimental data was based on a power fit for this kV so additional experiments should be run to confirm. The number of shocks for 14 and 16 kV agree nicely.

The location and propagation direction of the fracture in the 10 mm cylindrical stone, shown in Figure 6.12, matches well with experiment, shown in Figure 6.14. A similar, and expected, fracture pattern perpendicular to the propagation axis and appoximately one third the distance from the distal surface of the stone. At 14 kV the stone took

at least 24 shocks to break, but this was just one data point. The simulation predicts 5 shocks to initial fracture at 18 kV. More experiments need to be run on this stone size to determine if the damage parameters selected based on the 7mm cylindrical stone translate to this stone size.

The 10 mm spherical stone does not show good matching between fracture in experiment and simulation. In the simulation, Figure 6.13, the fracture pattern resembles the pattern in the cylindrical stones, in that the main fracture occurs approximately one third the diameter of the stone from the distal surface and propagates perpendicular to the propagation axis. In the μCT image, Figure 6.15, the fracture extends inward from the distal surface. The discrepancy in fracture pattern could be due to the differences in the experimental and simulation parameters. Specifically, the experimental fracture could be due in part to cavitation occuring on the distal surface. Also, the experiment uses 14 kV and the simulation uses 18 kV. The higher kV should lead to internal stresses playing a larger roll versus cavitation on fracture. To determine if this is the case, experiments should be run on 10 mm spherical stones in butanediol at larger kV. After 8 shocks the fracture propagation had slowed significantly and so this simulation was stopped in interest of running other simulations.

The model was also applied to two realistic kidney stones, by modifying the mesoscale structure and material properties. The struvite stone simulation showed in Figure 6.18 fracture developing nearer the distal surface of the stone than in the BegoStone simulations. This is due to the change in material properties not the slight change in mesoscopic structure. There is certainly very little difference between a cell with 250 grains versus 50 grains in this simulation. In either case, there are many surfaces per cell oriented in many different directions, and the effect on the continuum scale is negligible. In future work,

with more refined grids and possibly AMR, this effect could begin to cause differences since there may be very few surfaces in each cell.

The COM stone approximation, shown in Figure 6.19, developed fracture in a similar location as the BegoStone simulants which is expected given that they were assigned equivalent material parameters. An effect of the underlying anisotropic mesoscale structure possibly appears in the propagation direction of the macroscale crack. As opposed to the spherical BegoStone and struvite stones, in which the crack curves towards the stone surface almost in a purely radial direction, the crack in the COM stone curves slightly in the tangential direction. The concentric spherical mesoscale structure seems to influence the macoscale crack to at least partially move along the concentric spheres. These two simulations of realistic kidney stones are very preliminary and are meant to showcase the potential capabilities of this model. Future work may compare simulated fracture in these stones to experimental data.

Overall, this model appears to correctly predict the location and propagation direction of fracture within BegoStone kidney stone simulants. Additional experiments should be carried out to verify results of the 10 mm cylindrical and spherical stones. A forthcoming publication will extend the model to include AMR and parallelism which will greatly enhance the model. This should allow for much more accuracy of the location and propagation of the macroscale fracture through grid refinement. The added speedup will also allow for testing lower kV input, i.e. more shocks. Finally, future work may include cavitation effects. Some experimental data may need to be collected, but cavitation effects on the stone surfaces could be approximated. This could be accomplished by initiating macroscopic fracture along the stone surface when large tensile stresses exist outside the stone.

CHAPTER 7

Conclusion

In this work, two computational models were developed and applied to shock wave lithotripsy. Some experimental results on the shocking of kidney stone simulants and image processing of μCT data were also presented. Four stones were subjected to the shocking and scanning procedure and many subsequent images sets with visible fracture were collected. Development of an image processing procedure to quantify the fracture led to results on fracture volume per shock, and fitted probability distributions for crack length, width, and orientation.

Both computational models utilize the finite-volume conservative-law wave propagation algorithm of LeVeque [**69**]. Several modifications to the current implementation of the method in BEARCLAW were presented. These include computing the full transverse solves, combining normal solves prior to transverse solves for speedup of the method, and a framework for multiphysics utilizing arbitrary subsets of the domain.

The first computational model is a multiphysics model applied to the focusing and subsequent shock wave formation of an acoustic pulse in a refracting electromagnetic lithotripter. Through a multiphysics framework capable of solving the linear elasticity equations and Euler equations simultaneously with AMR, the model is verified against experimental results. The model is also able to predict the effects of a lens modification [**37**] and has been used to validate a new lens design [**89**]. Possible future work includes

adding cavitation and attenuation, a parallel implementation, and a 3D version for non-axisymmetric lenses.

The second model is a multiscale fracture model in which a continuum-mesoscopic interaction was developed which combines an elasticity solver with damage and an underlying mesoscopic structure. This model was specifically applied to kidney stone fracture and was able to correctly predict the location and direction of the initial fracture common in cylindrical BegoStone simulants. By including damage variables the model is also able to predict the number of shocks required for initial fracture of 7 mm cylindrical BegoStone simulants. Finally, the general mesoscopic framework allows for the modeling of real kidney stones with known mesoscopic structures. Immediate future work on this model includes AMR and parallelism as well as implementation of simple test cases to be included in a forthcoming publication. Further refinement of the method in the lithotripsy application and more comparisons to experiment will also be investigated.

CHAPTER 8

Appendix

8.1. Image Processing Procedure Code for Fracture Analysis 1

This appendix presents the MATLAB code developed to modify the μCT images of kidney stone simulants in a way that enhanced cracks. Essentially, dark and light regions of the imaged are removed allowing for a greater contrast between the cracks and stone. The process is described in Section 4.2.3.

```
function [ yfinal, a, b, cx, cy ] = illumCracks( y, bgcolor, userAvgVal,
  userScale, userMin, grayLow )
%   Highlights the cracks in the
%   stone images by filling in areas of light or dark with an average
%   color. Also blurs and rescales the color in the image. Either can
%   calculate values for filling in and rescaling directly from the image
%   or can use user specified values for processing many images in the
%   same set, so that exactly the same processes are applied.
%
%  Input:
%      y          -  image as 2d double array
%      bgcolor    -  0 will give black background, else white background
%      userAvgVal -  Average gray value. -1 will calculate this value
%                    from the current image. Otherwise uses avgGray as
```

```
%                     value.
%       userScale   -  Scaling factor. -1 will calculate this value from
%                     the current image. Otherwise uses userScale.
%       userMin     -  Other parameter in scaling equation. -1 uses value
%                     calculated from current image. Otherwise use userMin


% Apply a low pass filter to remove noise
gsize = 20;
gsigma = 1.5;
myfilt = fspecial('gaussian', gsize, gsigma);
y = imfilter(y,myfilt);
% Find dark pixels and light pixels
% Also identify 'outside the stone' region.
% Gray values range from 0 to 256 (lower numbers are darker).
% Lower threshold is defined as very dark to pick up inside of air
% pockets and not cracks.
% But as seen in image this won't cover all of air pockets.
% Also find average non dark value
avgVal = 0;
numAvgVal = 0;
% Define lower threshold
% grayLow = 65;
% Define higher threshold
grayHigh = 130;
% Find top, bottom, right and left of stone
grayOut = 110;
```

```
p = 1;
while max(y(p,:))<grayOut && p<size(y,1)-1
    p = p+1;
end
sl = p;
p = size(y,1);
while max(y(p,:))<grayOut && p>2
    p = p-1;
end
sr = p;
p = 1;
while max(y(:,p))<grayOut && p<size(y,2)-1
    p = p+1;
end
sb = p;
p = size(y,2);
while max(y(:,p))<grayOut && p>2
    p = p-1;
end
st = p;
a = (sr-sl)/2;
b = (st-sb)/2;
dab = 1;
cx = sl+a;
cy = sb+b;
thres = zeros(size(y));
```

```
dark = zeros(size(y));
% Determine the average gray value of this picture
% and use that to replace
if userAvgVal == -1
    [X Y] = ndgrid(1:size(y,1),1:size(y,2));
    % Loop over y and extract dark pixels
    for i=1:size(y,1)
        for j=1:size(y,2)
            if ((i-cx)/(a-dab))^2+((j-cy)/(b-dab))^2 > 1
                thres(i,j) = 3;
            elseif y(i,j) <= grayLow
                dark(i,j) = 1;
            elseif y(i,j) >= grayHigh
                thres(i,j) = 2;
            else
                numAvgVal = numAvgVal + 1;
                avgVal = avgVal + y(i,j);
            end
        end
    end
    % Calculate average non dark value
    avgVal = avgVal/numAvgVal;
% Otherwise use supplied gray value, when not doing sum
% for average value
else
    [X Y] = ndgrid(1:size(y,1),1:size(y,2));
```

```
    % Logical matrix operations
    inc1 = ((X-cx)./(a-dab)).^2 + ((Y-cy)./(b-dab)).^2 > 1;
    thres(inc1) = 3;
    inc = (y <= grayLow) & ~(inc1);
    dark(inc)  = 1;
    inc = (y >= grayHigh) & ~(inc1);
    thres(inc) = 2;
    avgVal = userAvgVal;
end
% Blur dark pixel map
% Now instead of using 1 to denote dark pixels use nonzero, which
% has increased the size of the area covering the air pockets.
gsize = 5;
gsigma = 20;
myfilt = fspecial('gaussian', gsize, gsigma);
dark2 = imfilter(dark,myfilt);
% Go over new dark map and change values
darkThres = 0.02;
y3 = y;
inc = dark2 > darkThres;
y3(inc) = avgVal;
inc = thres == 2;
y3(inc) = avgVal;
inc = thres == 3;
if bgcolor == 0
    y3(inc) = 40;
```

```
else
    y3(inc) = 180;
end
% More blur to reduce edges from filling in areas
myfilt = fspecial('gaussian', 10, 4);
y4 = imfilter(y3,myfilt);
% Make sure outside of stone is white
inc = ((X-cx)/(a-dab)).^2 + ((Y-cy)/(b-dab)).^2 > 1;
y4(inc) = 180;
% Rescale grays
if userScale == -1 && userMin == -1
    y5 = y4;
    cmax = max(max(y5));
    cmin = min(min(y5));
    dc1 = 1/(cmax-cmin);
    yfinal = 256*dc1*(y5-cmin);
    if bgcolor == 0
        yfinal(inc) = 0;
    end
else
    yfinal = 256*userScale*(y4-userMin);
    if bgcolor == 0
        yfinal(inc) = 0;
    end
    % Make sure no values are out of bounds
    inc = yfinal<0;
```

```
    yfinal(inc) = 0;
    inc = yfinal>256;
    yfinal(inc) = 256;
end
```

8.2. Image Processing Procedure Code for Fracture Analysis 2

This appendix presents the MATLAB code developed to cover the cracks in the μCT images of kidney stone simulants with a set of line segments. This process assumes the images have already had the illuminate cracks code applied, as presented in Appendix 8.1. The process shown here requires several steps and is described in Section 4.2.3.

```
function [ segs widths ] = crackSegments( y2 )
% Assumes input image has already had illumCracks
% applied to it
% Function is divided into 6 main parts:
%   1) Compute gradient and magnitude, and edge detection
%   2) Find midpoints of edge lines using gradient
%   3) Create lines representative of cracks
%   4) Remove lines which include bright areas
%   5) Match any close points together
%   6) Check for and remove doubles
%   7) Recalculate widths
%-------------------------------------------------------------
% 1) Compute gradient and magnitude, and edge detection
%-------------------------------------------------------------
% Gradient assumes columns are x and rows are y
```

```
% so the outputs are switched
[imy,imx] = gradient(y2);
mag = sqrt(imx.^2 + imy.^2);
edges = edge(y2,'canny',0.11);
%-------------------------------------------------------------
% 2) Find midpoints of edge lines using gradient
%-------------------------------------------------------------
% Declare line information arrays
lines = zeros(size(mag));
linesx = zeros(size(mag));
linesy = zeros(size(mag));
linesw = zeros(size(mag));
% Lower and upper width of crack in pixels
dl = 2;
du = 25;
% Step of crack width calculator
dxy = 0.5;
% Points have to be at least this dark
pointGrayThres = 110;
% Thresholds for opposite edge values
angThres = 0.5;
magThres = 20;
edgeN = 1;
for i=1:size(y2,1)
  for j=1:size(y2,2)
    % Check that gradient magnitude is above a certain threshold but
```

```
% not too high (which corresponds to the edge of the stone)
%if mag(i,j)>magt1 && mag(i,j)<magt2
% Check if edge is here
if edges(i,j)
  % Only operate on "half" of the vectors, to not double up
  % cracks
  if imx(i,j) < 0
    % Move in direction opposite of gradient and look for
    % similar magnitude
    % Normalize gradient vector
    nx = -imx(i,j)/mag(i,j);
    ny = -imy(i,j)/mag(i,j);
    % Current position is at starting position
    cx = dl*nx;
    cy = dl*ny;
    dc = dl;
    % Step length for each component
    dx = dxy*nx;
    dy = dxy*ny;
    % Final position for each component
    dux = du*nx;
    duy = du*ny;
    done = false;
    while ~done && dc<du
      % Check current pixel
      cpx = round(i+cx);
```

```
cpy = round(j+cy);
% If pixel has similar magnitude and opposite gradient
% then assume this is other side of crack
if cpy<=size(mag,2) && cpy>1 && cpx>1 && ...
   cpx<size(mag,1) && ...
   abs(mag(cpx,cpy)-mag(i,j))<magThres
  % Check angle between gradient vectors
  theta = acos(([-imx(i,j) -imy(i,j)]*[imx(cpx,cpy); ...
          imy(cpx,cpy)])/(mag(i,j)*mag(cpx,cpy)));
  if abs(theta)<angThres
    % Make sure there is a crack edge nearby
    if max(max(edges(cpx-edgeN:cpx+edgeN, ...
      cpy-edgeN:cpy+edgeN)))>0
      % Then this is probably a crack, mark line in middle
      i1 = round(0.5*(cpx+i));
      j1 = round(0.5*(cpy+j));
      % Make sure that its actually a dark pixel
      if y2(i1,j1)<y2(i,j) && y2(i,j)<pointGrayThres
        lines(i1,j1) = 1;
        linesw(i1,j1) = dc;
        linesx(i1,j1) = -imy(i,j);
        linesy(i1,j1) = imx(i,j);
        done = true;
      end
    end
  end
```

```
            end
            % Update search
            cx = cx + dx;
            cy = cy + dy;
            dc = dc + dxy;
          end
        end
      end
    end
end
% Normalize linesx,y
lmag = 1./sqrt(linesx.^2 + linesy.^2);
linesx = linesx.*lmag;
linesy = linesy.*lmag;
%-------------------------------------------------------------
% 3) Create lines representative of cracks
%-------------------------------------------------------------
numsegs = 0;
segs = zeros(sum(sum(lines)),4);
widths = zeros(size(segs,1),1);
doneLines = zeros(size(lines));
connect = zeros(size(lines));
closeFarR = 8;
for i=1:size(lines,1)
  for j=1:size(lines,2)
    if lines(i,j) && ~doneLines(i,j)
```

```
% Set current point to done
doneLines(i,j) = 1;
% Find angle of direction vector
dirx = linesx(i,j);
diry = linesy(i,j);
% Move in circle around point
for r=1:0.5:20
  for th=-pi/2:pi/(3*r):3*pi/2
    % Calculate current direction vector (cdirx,cdiry)
    % and current point (i1,j1)
    cdirx = cos(th);
    cdiry = sin(th);
    i1 = round(i + r*cdirx);
    j1 = round(j + r*cdiry);
    % If line point is close and points in same
    % direction just include it with this one
    if r<closeFarR
      if lines(i1,j1)
        thetad = acos([dirx diry]*[linesx(i1,j1); ...
                 linesy(i1,j1)]);
        if thetad<0.4*pi || theta>2.75*pi
          doneLines(i1,j1) = 1;
        end
      end
    % Else we're far enough to start looking
    % for connections
```

```
else
  % In first ('1') direction (towards dirx diry)
  if ([dirx diry]*[cdirx; cdiry])>0.7071 && ...
     (connect(i,j) == 0 || connect(i,j) == -1)
    % Find next point to connect to
    if lines(i1,j1) && ~doneLines(i1,j1)
      % Add line segment
      numsegs = numsegs + 1;
      segs(numsegs,1:2) = [i j];
      segs(numsegs,3:4) = [i1 j1];
      widths(numsegs) = 0.5*(linesw(i,j)+ ...
          linesw(i1,j1)) + 0.75*randn;
      if connect(i,j) == 0
        connect(i,j) = 1;
      elseif connect(i,j) == -1
        connect(i,j) = 2;
      end
      % Check new point for similar direction
      thetad = acos([dirx diry]*[linesx(i1,j1); ...
                linesy(i1,j1)]);
      if thetad<0.25*pi
        if connect(i1,j1) == 0
          connect(i1,j1) = -1;
        elseif connect(i1,j1) == 1
          connect(i1,j1) = 2;
        end
```

```
    elseif thetad>0.75*pi
      if connect(i1,j1) == 0
        connect(i1,j1) = 1;
      elseif connect(i1,j1) == -1
        connect(i1,j1) = 2;
      end
    end
  end
  % In second ('-1') direction (away from dirx diry)
elseif ([dirx diry]*[cdirx; cdiry])<-0.7071 && ...
       (connect(i,j) == 0 || connect(i,j) == 1)
  % Find next point to connect to
  if lines(i1,j1) && ~doneLines(i1,j1)
    % Add line segment
    numsegs = numsegs + 1;
    segs(numsegs,1:2) = [i j];
    segs(numsegs,3:4) = [i1 j1];
    widths(numsegs) = 0.5*(linesw(i,j)+...
        linesw(i1,j1)) + 0.75*randn;
    if connect(i,j) == 0
      connect(i,j) = -1;
    elseif connect(i,j) == 1
      connect(i,j) = 2;
    end
    % Check new point for similar direction
    thetad = acos([dirx diry]*[linesx(i1,j1); ...
```

```
                        linesy(i1,j1)]);
                if thetad<0.25*pi
                  if connect(i1,j1) == 0
                    connect(i1,j1) = 1;
                  elseif connect(i1,j1) == -1
                    connect(i1,j1) = 2;
                  end
                elseif thetad>0.75*pi
                  if connect(i1,j1) == 0
                    connect(i1,j1) = -1;
                  elseif connect(i1,j1) == 1
                    connect(i1,j1) = 2;
                  end
                end
              end
            end
          end
        end
      end
    end
  end
end

%-------------------------------------------------------------
% 4) Remove lines which include bright areas
%-------------------------------------------------------------
```

```
ns2 = 0;
segs2 = zeros(numsegs,4);
widths2 = zeros(numsegs,1);
lightThres = 105;
for n=1:numsegs
    segdir = [segs(n,3)-segs(n,1), segs(n,4)-segs(n,2)];
    seglength = sqrt(segdir(1)^2+segdir(2)^2);
    segdir = segdir/seglength;
    dc = 1;
    lightCross = false;
    while dc<seglength && ~lightCross
        i1 = round(segs(n,1) + dc*segdir(1));
        j1 = round(segs(n,2) + dc*segdir(2));
        if y2(i1,j1)>lightThres
            lightCross = true;
        end
        dc = dc+1;
    end
    if ~lightCross
        ns2 = ns2 + 1;
        segs2(ns2,:) = segs(n,:);
        widths2(ns2) = widths(n);
    end
end
segs = segs2;
numsegs = ns2;
```

```
% Randomly lengthen some cracks
% so that lines extend closer to crack edges
for n=1:numsegs
    dirx = segs(n,1)-segs(n,3);
    diry = segs(n,2)-segs(n,4);
    dirn = norm([dirx diry]);
    dirx = dirx/dirn;
    diry = diry/dirn;
    randlen1 = abs(3*randn);
    if randlen1>6
        randlen1 = 6;
    end
    segs(n,1) = round(segs(n,1) + dirx*randlen1);
    segs(n,2) = round(segs(n,2) + diry*randlen1);
    randlen2 = abs(3*randn);
    if randlen2>6
        randlen2 = 6;
    end
    segs(n,3) = round(segs(n,3) - dirx*randlen2);
    segs(n,4) = round(segs(n,4) - diry*randlen2);
end
%-------------------------------------------------------------
% 5) Match any close points together
%-------------------------------------------------------------
% Boolean array saying if a point has had another point
% moved to it, it should then not move
```

```
movedto = zeros(numsegs,2);
% Distance threshold that two points should be the same
disThres = 5;
% Loop over all line segments
for n=1:numsegs
    % If first point of line segment hasn't had a point
    % moved to it
    if ~movedto(n,1)
        % Loop over all other segments
        for ncheck=[1:n-1 n+1:numsegs]
            % Check first point of other segments
            if norm([segs(n,1)-segs(ncheck,1), ...
               segs(n,2)-segs(ncheck,2)])<disThres
                segs(n,1:2) = segs(ncheck,1:2);
                movedto(ncheck,1) = true;
            % Check second point of other segments
            elseif norm([segs(n,1)-segs(ncheck,3), ...
                   segs(n,2)-segs(ncheck,4)])<disThres
                segs(n,1:2) = segs(ncheck,3:4);
                movedto(ncheck,2) = true;
            end
        end
    end
    % If second point of line segment hasn't had a
    % point moved to it
    if ~movedto(n,2)
```

```
        % Loop over all other segments
        for ncheck=[1:n-1 n+1:numsegs]
            % Check first point of other segments
            if norm([segs(n,3)-segs(ncheck,1), ...
               segs(n,4)-segs(ncheck,2)])<disThres
                segs(n,3:4) = segs(ncheck,1:2);
                movedto(ncheck,1) = true;
            % Check second point of other segments
            elseif norm([segs(n,3)-segs(ncheck,3), ...
                   segs(n,4)-segs(ncheck,4)])<disThres
                segs(n,3:4) = segs(ncheck,3:4);
                movedto(ncheck,2) = true;
            end
        end
    end
end
%-------------------------------------------------------------
% 6) Check for and remove doubles
%-------------------------------------------------------------
toBeRemoved = zeros(numsegs);
nRemoved = 0;
for n=1:numsegs
    % First check if segment is just a point
    if segs(n,1) == segs(n,3) && segs(n,2) == segs(n,4)
        toBeRemoved(n) = true;
        nRemoved = nRemoved + 1;
```

```
    % Check segment against all other segments
    else
        for ncheck=[1:n-1 n+1:numsegs]
            if ~toBeRemoved(ncheck)
                if (segs(n,1)==segs(ncheck,1) && ...
                    segs(n,2)==segs(ncheck,2) && ...
                    segs(n,3)==segs(ncheck,3) && ...
                    segs(n,4)==segs(ncheck,4)) || ...
                   (segs(n,1)==segs(ncheck,3) && ...
                    segs(n,2)==segs(ncheck,4) && ...
                    segs(n,3)==segs(ncheck,1) && ...
                    segs(n,4)==segs(ncheck,2))
                    toBeRemoved(n) = true;
                    nRemoved = nRemoved + 1;
                    break;
                end
            end
        end
    end
end
% Create new line segment structured with doubles removed
ns2 = 0;
segs2 = zeros(ns2,4);
widths2 = zeros(ns2,1);
for n=1:numsegs
    if ~toBeRemoved(n)
```

```
        ns2 = ns2 + 1;
        segs2(ns2,:) = segs(n,:);
        widths2(ns2) = widths(n);
    end
end
segs = segs2;
widths = widths2;
%-------------------------------------------------------------
% 7) Recalculate widths
%-------------------------------------------------------------
% by looping over segments, and looking from midpoint
% of segment towards crack edge.
numsegs = size(segs,1);
widths2 = zeros(numsegs,1);
wthres = 100;
widthPlus = 0;
widthMinus = 0;
for i=1:numsegs
    % Find midpoint
    mx = 0.5*(segs(i,1) + segs(i,3));
    my = 0.5*(segs(i,2) + segs(i,4));
    % Calc direction of seg
    dx = segs(i,1) - segs(i,3);
    dy = segs(i,2) - segs(i,4);
    dn = 1/norm([dx dy]);
    dx = dx*dn;
```

```
dy = dy*dn;
% Rotate direction
dirx = -dy;
diry =  dx;
% Move in direction until edge is found
dx = 0.5;
dc = dx;
done = false;
if y2(round(mx),round(my))>wthres
    widthPlus = 0;
    widthMinus = 0;
else
    while ~done && dc<25
        px = mx + dc*dirx;
        py = my + dc*diry;
        if y2(round(px),round(py))>wthres
            done = true;
            widthPlus = dc;
        else
            dc = dc + dx;
        end
    end
    dx = 0.5;
    dc = dx;
    done = false;
    while ~done && dc<25
```

```
            px = mx - dc*dirx;
            py = my - dc*diry;
            if y2(round(px),round(py))>wthres
                done = true;
                widthMinus = dc;
            else
                dc = dc + dx;
            end
        end
    end
    widthTot = widthMinus + widthPlus;
    if widthMinus > 24 || widthPlus > 24 || widthTot < 2
        widths2(i) = -1;
    else
        widths2(i) = widthTot + 0.5*randn;
    end
end
notwidths = (widths2 == -1);
widths = widths2(~notwidths);
end
```

References

[1] FF Abraham, JQ Broughton, N Bernstein, and E Kaxiras, *Spanning the continuum to quantum length scales in a dynamic simulation of brittle fracture*, Europhysics Letters **44** (1998), no. 6, 783–787.

[2] FF Abraham, JQ Broughton, N Bernstein, and E Kaxiras, *Spanning the length scales in dynamic simulation*, Computers in Physics **12** (1998), no. 6, 538–546.

[3] Alireza Akhavan, Seyed-Mohammad-Hadi Shafaatian, and Farshad Rajabipour, *Quantifying the effects of crack width, tortuosity, and roughness on water permeability of cracked mortars*, Cement and Concrete Research **42** (2012), no. 2, 313–320.

[4] T. L. Anderson, *Fracture mechanics fundamentals and applications*, CRC Press, 2006.

[5] M. Arora, C. D. Ohl, and D. Lohse, *Effect of nuclei concentration on cavitation cluster dynamics*, J. Acoust. Soc. Am. **121** (2007), 3432–3436.

[6] DRB Aroush, E Maire, C Gauthier, S Youssef, P Cloetens, and HD Wagner, *A study of fracture of unidirectional composites using in situ high-resolution synchrotron x-ray microtomography*, Composites Science and Technology **66** (2006), no. 10, 1348–1353.

[7] Robert J. Asaro and Vlado A. Lubarda, *Mechanics of solids and materials*, Cambridge University Press, 2006.

[8] Michalakis A. Averkiou and Robin O. Cleveland, *Modeling of an electrohydraulic lithotripter with KZK equation*, J. Acoust. Soc. Am. **106** (1999), 102–112.

[9] L Babout, W Ludwig, E Maire, and JY Buffiere, *Damage assessment in metallic structural materials using high resolution synchrotron x-ray tomography*, Nuclear Instruments & Methods in Physics Research Section B-Beam Interactions with Materials and Atoms **200** (2003), 303–307.

[10] Zdenek P. Bazant and Emilie Becq-Giraudon, *Statistical prediction of fracture parameters of concrete and implications for choice of testing standard*, Cement and conrete research **32** (2002), no. 4, 529–556.

[11] T Belytschko and T Black, *Elastic crack growth in finite elements with minimal remeshing*, International Journal for Numerical Methods in Engineering **45** (1999), no. 5, 601–620.

[12] T Belytschko, L Gu, and YY Lu, *Fracture and crack growth by element free galerkin methods*, Modelling and Simulation in Materials Science and Engineering **2** (1994), no. 3A, 519–534.

[13] T Belytschko, Y Krongauz, D Organ, M Fleming, and P Krysl, *Meshless methods: An overview and recent developments*, Computer Methods in Applied Mechanics and Engineering **139** (1996), no. 1-4, 3–47.

[14] JE Bolander and S Saito, *Fracture analyses using spring networks with random geometry*, Engineering Fracture Mechanics **61** (1998), no. 5-6, 569–591.

[15] K. Bertram Broberg, *Cracks and fracture*, Academic Press, 1999.

[16] JQ Broughton, FF Abraham, N Bernstein, and E Kaxiras, *Concurrent coupling of length scales: Methodology and application*, Physical Review B **60** (1999), no. 4, 2391–2403.

[17] W. C. Carpenter, D. T. Read, and R. H. Dodds, *Comparison of several path independent integrals including plasticity effects*, International journal of fracture **31** (1986), no. 4, 303–323.

[18] J. Chaboche, *Continuum damage mechanics: a tool to describe phenomena before crack initiation*, Nucl. Eng. Des. **64** (1981), no. 2, 233–247.

[19] CH. Chaussy, Walter Brendel, and E. Schmiedt, *Extracorporeally induced destruction of kidney stones by shock waves*, The Lancet **316** (1980), no. 8207, 1265 – 1268.

[20] P. T. Christopher, *Modeling the Dornier HM3 lithotripter*, J. Acoust. Soc. Am. **96** (1994), 3088–3095.

[21] P. T. Christopher and K. J. Parker, *New approaches to nonlinear diffractive field propagation*, J. Acoust. Soc. Am. **90** (1991), 488–499.

[22] Robin O. Cleveland and Oleg A. Sapozhnikov, *Modeling elastic wave propagation in kidney stones with application to shock wave lithotripsy*, J. Acoust. Soc. Am. **118** (2005), no. 4, 2667–2676.

[23] A. J. Coleman, M. J. Choi, and J. E. Saunders, *Theoretical predictions of the acoustic pressure generated by a shock wave lithotripter*, Ultrasound in Med. & Biol. **17** (1991), no. 3, 245–255.

[24] Dustin Crandall, Grant Bromhal, and Zuleima T. Karpyn, *Numerical simulations examining the relationship between wall-roughness and fluid flow in rock fractures*, International Journal of Rock Mechanics and Mining Sciences **47** (2010), no. 5, 784–796.

[25] WA Curtin and RE Miller, *Atomistic/continuum coupling in computational materials science*, Modelling and Simulation in Materials Science and Engineering **11** (2003), no. 3, R33–R68.

[26] WA Curtin and H Scher, *Brittle-fracture in disordered materials: A spring network model*, Journal of Materials Research **5** (1990), no. 3, 535–553.

[27] G Dahake and SM Gracewski, *Finite difference predictions of p-sv wave propagation inside submerged solids .1. liquid-solid interface conditions*, Journal of the Acoustical Society of America **102** (1997), no. 4, 2125–2137.

[28] G Dahake and SM Gracewski, *Finite difference predictions of p-sv wave propagation inside submerged solids .2. effect of geometry*, Journal of the Acoustical Society of America **102** (1997), no. 4, 2138–2145.

[29] MS Daw and MI Baskes, *Embedded-atom method - derivation and application to impurities, surfaces, and other defects in metals*, Physical Review B **29** (1984), no. 12, 6443–6453.

[30] L de Arcangelis, A Hansen, HJ Herrmann, and S Roux, *Scaling laws in fracture*, Physical Review B **40** (1989), no. 1, 877–880.

[31] J Dolbow, N Moës, and T Belytschko, *An extended finite element method for modeling crack growth with frictional contact*, Computer Methods in Applied Mechanics and Engineering **190** (2001), no. 51-52, 6825–6846.

[32] Tulga Egilmez, Mehmet Ilteris Tekin, Murat Gonen, Ferhat Kilinc, Resit Goren, and Hakan Ozkardes, *Efficacy and safety of a new-generation shockwave lithotripsy machine in the treatment of single renal or ureteral stones: experience with 2670 patients*, Journal of Endourology **21** (2007), no. 1, 23–27.

[33] W Eisenmenger, XX Du, C Tang, S Zhao, Y Wang, F Rong, D Dai, M Guan, and A Qi, *The first clinical results of "wide-focus and low-pressure" ESWL*, Ultrasound in Medicine and Biology **28** (2002), no. 6, 769–774.

[34] William Cronk Elmore and Mark A. Heald, *Physics of waves*, McGraw-Hill, 1969.

[35] Eric Esch, Walter Neal Simmons, Georgy Sankin, Hadley F. Cocks, Glenn M. Preminger, and Pei Zhong, *A simple method for fabricating artifcial kidney stones of different physical properties*, Urological Research **38** (2010), 315–319.

[36] Tiernan R. Fogarty and Randall J. LeVeque, *High-resolution finite-volume methods for acoustic waves in periodic and random media*, J. Acoust. Soc. Am. **106** (1999), 17.

[37] Daniel E. Fovargue, Sorin Mitran, Nathan B. Smith, Georgy N. Sankin, Walter N. Simmons, and Pei Zhong, *Experimentally validated multiphysics computational*

model of focusing and shock wave formation in an electromagnetic lithotripter, Journal of the Acoustical Society of America (Accepted 2013).

[38] K Gall, MF Horstemeyer, M Van Schilfgaarde, and MI Baskes, *Atomistic simulations on the tensile debonding of an aluminum-silicon interface*, Journal of the Mechanics and Physics of Solids **48** (2000), no. 10, 2183–2212.

[39] PC Gehlen, MF Kanninen, and GT Hahn, *Crack extension by bond rupture in a model of bcc iron*, Scripta Metallurgica **6** (1972), no. 11, 1087.

[40] Siegfried Ginter, Marko Liebler, Eckard Steiger, Thomas Dreyer, and Rainer E. Riedlinger, *Full-wave modeling of therapeutic ultrasound: Nonlinear ultrasound propagation in ideal fluids*, The Journal of the Acoustical Society of America **111** (2002), no. 5, 2049–2059.

[41] S. K. Godunov, *A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics*, Mat. Sb. **47** (1959), 271–306.

[42] A. A. Griffith, *The phenomena of rupture and flow in solids*, Philosophical Transactions of the Royal Society **A221** (1920), 163–198.

[43] P Gumbsch and GE Beltz, *On the continuum versus atomistic descriptions of dislocation nucleation and cleavage in nickel*, Modelling And Simulation In Materials Science And Engineering **3** (1995), no. 5, 597–613.

[44] P Gumbsch and RM Cannon, *Atomistic aspects of brittle fracture*, MRS Bulletin **25** (2000), no. 5, 15–20.

[45] Mark F. Hamilton, *Transient axial solution for the reflection of a spherical from a concave ellipsoidal mirror*, J. Acoust. Soc. Am. **93** (1993), 1256–1266.

[46] T. K. Hellen, *On the method of virtual crack extensions*, International journal for numerical methods in engineering **9** (1975), 187–207.

[47] HJ Herrmann, A Hansen, and S Roux, *Fracture of disordered, elastic lattices in two dimensions*, Physical Review B **39** (1989), no. 1, 637–648.

[48] M Hori, K Oguni, and H Sakaguchi, *Proposal of fem implemented with particle discretization for analysis of failure phenomena*, Journal of the Mechanics and Physics of Solids **53** (2005), no. 3, 681–703.

[49] A. Hrennikoff, *Solution of problems of elasticity by the framework method*, Journal of applied mechanics **12** (1941), 169–175.

[50] Jonathan I. Iloreta, Yufeng Zhou, Georgy N. Sankin, Pei Zhong, and Andrew J. Szeri, *Assessment of shock wave lithotripters via cavitation potential*, Physics of Fluids **19** (2007), no. 8.

[51] G. R. Irwin, *Fracture dynamics*, Fracturing of Metals, American Society for Metals (1948), 147–166.

[52] G. R. Irwin, *Analysis of stresses and strains near the end of a crack traversing a plate*, Journal of Applied Mechanics **24** (1957), 361–364.

[53] A Jagota and SJ Bennison, *Element breaking rules in computational models for brittle fracture*, Modelling and Simulation in Materials Science and Engineering **3** (1995), no. 4, 485–501.

[54] EAA Jarvis, RL Hayes, and EA Carter, *Effects of oxidation on the nanoscale mechanisms of crack formation in aluminum*, CHEMPHYSCHEM **2** (2001), no. 1, 55–59.

[55] L. Kachanov, *Time of the rupture process under creep conditions*, TVZ Akad Nauk S.S.R. Otd. Tech. Nauk **8** (1958), 26–31.

[56] T Kawai, *New discrete models and their application to seismic response analysis of structures*, Nuclear Engineering and Design **48** (1978), no. 1, 207–229.

[57] A. S. Kobayashi, R. D. Cherepy, and W. C. Kinsel, *A numerical procedure for estimating the stress intensity factor of a crack in a finite plate*, Journal of Basic Engineering **86** (1964), no. 4, 681–684.

[58] Maya Kobchenko, Hamed Panahi, Francois Renard, Dag K. Dysthe, Anders Malthe-Sorenssen, Adriano Mazzini, Julien Scheibert, Bjorn Jamtveit, and Paul Meakin, *4d imaging of fracturing in organic-rich shales during heating*, Jounal of Geophysical Research-Solid Earth **116** (2011).

[59] S Kohlhoff, P Gumbsch, and HF Fischmeister, *Crack propagation in b.c.c. crystals studied with a combined finite-element and atomistic model*, Philosophical Magazine A-Physics of Condensed Matter Structure Defects and Mechanical Properties **64** (1991), no. 4, 851–878.

[60] A. E. Korenchenko and V. P. Beskachko, *Determining the shear modulus of water in experiments with a floating disk*, Journal of Applied Mechanics and Technical Physics **49** (2008), no. 1, 80–83.

[61] Jeff Krimmel, Tim Colonius, and Michel Tanguay, *Simulation of the effects of cavitation and anatomy in the shock path of model lithotripters*, 3rd International Urolithiasis Research Symposium **38** (2010), 505–518.

[62] E. N. Landis, T. Zhang, E. N. Nagy, G. Nagy, and W. R. Franklin, *Cracking, damage and fracture in four dimensions*, Materials and Structures **40** (2007), no. 4, 357–364.

[63] EN Landis, EN Nagy, and DT Keane, *Microstructure and fracture in three dimensions*, Engineering Fracture Mechanics **70** (2003), no. 7-8, 911–925.

[64] J. Lemaitre, *Evaluation of dissipation and damage in metals submitted to dynamic loading*, Proceedings of the ICM-1, Kyoto, Japan (1971).

[65] Jean Lemaitre and Jean-Louis Chaboche, *Mechanics of solid materials*, Cambridge University Press, 1990.

[66] Jean Lemaitre and Rodrigue Desmorat, *Engineering damage mechanics*, Springer, New York, 2005.

[67] Grady I. Lemoine and M. Yvonne Ou, *Finite volume modeling of poroelastic-fluid wave propagation with mapped grids*, (Submitted 2013).

[68] Randall J. LeVeque, *Clawpack software*, `http://depts.washington.edu/clawpack/`.

[69] Randall J. LeVeque, *Wave propagation algorithms for multidimensional hyperbolic systems*, Journal of Computational Physics **131** (1997), 327–353.

[70] Randall J. LeVeque, *Finite volume methods for hyperbolic problems*, Cambridge University Press, New York, 2002.

[71] Marko Liebler, Thomas Dreyer, and Rainer E. Riedlinger, *Modeling of interaction between therapeutic ultrasound propagation and cavitation bubbles*, Ultrasonics **44, Supplement** (2006), no. 0, e319 – e324.

[72] G Lilliu and JGM van Mier, *3d lattice type fracture model for concrete*, Engineering Fracture Mechanics **70** (2003), no. 7-8, 927–941.

[73] J. E. Lingeman, D. Newman, J. H. Mertz, P. G. Mosbaugh, R. E. Steele, R. J. Kahnoski, T. A. Coury, and J. R. Woods, *Extracorporeal shock wave lithotripsy: the methodist hospital of indiana experience.*, J Urol **135** (1986), 1134–1137.

[74] James E. Lingeman, Samuel C. Kim, Ramsay L. Kuo, James A. McAteer, and Andrew P. Evan, *Shockwave lithotripsy: Anecdotes and insights*, Journal of Endourology **17** (2003), no. 9.

[75] James E. Lingeman, James A. McAteer, Ehud Gnessin, and Andrew P. Evan, *Shock wave lithotripsy: advances in technology and technique*, Nature Review Urology **6** (2009), 660–670.

[76] Yunbo Liu and Pei Zhong, *BegoStone - a new stone phantom for shock wave lithotripsy research*, Journal of the Acoustical Society of America **112** (2002), no. 4.

[77] Murtuza Lokhandwalla and Bradford Sturtevant, *Fracture mechanics model of stone comminution in ESWL and implications for tissue damage*, Phys. Med. Biol. **45** (2000), 1923–1940.

[78] Haibiao Luo, *Three-dimensional model of elastic wave propagating in kidney stones with application to shock wave lithotripsy*, Ph.D. thesis, Boston University, 2011.

[79] E. Maire, O. Bouaziz, M. Di Michiel, and C. Verdu, *Initiation and growth of damage in a dual-phase steel observed by x-ray microtomography*, Acta Materialia **56** (2008), no. 18, 4954–4964.

[80] John G. Mancini, Andreas Neisius, Nathan Smith, Georgy Sankin, Gaston M. Astroza, Michael E. Lipkin, Walther N. Simmons, Glenn M. Preminger, and Pei Zhong, *Assessment of a modified acoustic lens for electromagnetic shock wave lithotripters in a swine model*, Journal of Urology (In press 2013).

[81] JA McAteer, JC Williams, RO Cleveland, J Van Cauwelaert, MR Bailey, DA Lifshitz, and AP Evan, *Ultracal-30 gypsum artificial stones for research on the mechanisms of stone breakage in shock wave lithotripsy*, Urological Research **33** (2005), no. 6, 429–434.

[82] S Mihradi, H Homma, and Y Kanto, *Numerical analysis of kidney stone fragmentation by short pulse impingement*, JSME International Journal Series A-Solid Mechanics and Material Engineering **47** (2004), no. 4, 581–590.

[83] Nicole L. Miller and James E. Lingeman, *Treatment of kidney stones: current lithotripsy devices are proving less effective in some case*, Nature Clinical Practice Urology **3** (2006), 5.

[84] R Miller, EB Tadmor, R Phillips, and M Ortiz, *Quasicontinuum simulation of fracture at the atomic scale*, Modelling and Simulation in Materials Science and Engineering **6** (1998), no. 5, 607–638.

[85] RE Miller and EB Tadmor, *The quasicontinuum method: Overview, applications and current directions*, Journal of Computer-Aided Materials Design **9** (2002), no. 3, 203–239.

[86] S. Mitran, *Bearclaw software*, `http://mitran.web.unc.edu/codes`.

[87] N Moës, J Dolbow, and T Belytschko, *A finite element method for crack growth without remeshing*, International Journal for Numerical Methods in Engineering **46** (1999), no. 1, 131–150.

[88] T. F. Morgeneyer, M. J. Starink, and I. Sinclair, *Evolution of voids during ductile crack propagation in an aluminium alloy sheet toughness test studied by synchrotron radiation computed tomography*, Acta Materialia **56** (2008), no. 8, 1671–1679.

[89] Andreas Neisius, Nathan B. Smith, Georgy Sankin, Nicholas J. Kuntz, John F. Madden, Daniel E. Fovargue, Sorin Mitran, Michael E. Lipkin, Walter N. Simmons, Glenn M. Preminger, and Pei Zhong, *In situ lens modification for confocal alignment of peak cavitation activity and pressure in an electromagnetic shock wave lithotripter*, Science Translational Medicine (In Preperation).

[90] T. Nishioka, *Computational dynamic fracture mechanics*, International Journal of Fracture **86** (1997), 127–159.

[91] J Oliver, *Modelling strong discontinuities in solid mechanics via strain softening constitutive equations .1. Fundamentals*, International Journal for Numerical Methods in Engineering **39** (1996), no. 21, 3575–3600.

[92] J Oliver, *Modelling strong discontinuities in solid mechanics via strain softening constitutive equations .2. numerical simulation*, International Journal for Numerical Methods in Engineering **39** (1996), no. 21, 3601–3623.

[93] J. Oliver, A. E. Huespe, and P. J. Sanchez, *A comparative study on finite elements for capturing strong discontinuities: E-fem vs x-fem*, Computer Methods in Applied Mechanics and Engineering **195** (2006), no. 37-40, 4732–4752.

[94] HS Park and WK Liu, *An introduction and tutorial on multiple-scale analysis in solids*, Computer Methods in Applied Mechanics and Engineering **193** (2004), no. 17-20, 1733–1772.

[95] D. M. Parks, *A stiffness derivative finite element technique for determination of crack tip stress intensity factors*, International Journal of Fracture **10** (1974), 487–502.

[96] Walter D. Pilkey and Deborah F. Pilkey, *Peterson's stress concentration factors*, John Wiley & Sons, Inc, 2008.

[97] YA Pishchalnikov, OA Sapozhnikov, MR Bailey, IV Pishchalnikova, JC Williams, and JA McAteer, *Cavitation selectively reduces the negative-pressure phase of lithotripter shock pulses*, ARLO **6** (2005), no. 4, 280–286.

[98] G Pittomvils, H Vandeursen, M Wevers, JP Lafaut, D Deridder, P Demeester, R Boving, and L Baert, *The influence of internal stone structure upon the fracture-behavior of urinary calculi*, Ultrasound In Medicine And Biology **20** (1994), no. 8, 803–810.

[99] James W. Provan, *Probabilistic fracture mechanics and reliability*, M. Nijhoff, 1986.

[100] Jun Qin, *Performance evaluation and design improvement of electromagnetic shock wave lithotripters*, Ph.D. thesis, Duke University, 2008.

[101] R. Quey, P.R. Dawson, and F. Barbe, *Large-scale 3d random polycrystals for the finite element method: Generation, meshing and remeshing.*

[102] Romain Quey, *Neper software*, `http://neper.sourceforge.net/`.

[103] Jens J. Rassweiler, Thomas Knoll, Kai-Uwe Kohrmann, James A. McAteer, James E. Lingeman, Robin O. Cleveland, Michael R. Bailey, and Christian Chaussy, *Shock wave technology and application: An update*, European Urology **59** (2011), 784–796.

[104] J. N. Reddy, *An introduction to continuum mechanics*, Cambridge University Press, 2008.

[105] Francois Renard, Dominique Bernard, Jacques Desrues, and Audrey Ougier-Simonin, *3d imaging of fracture propagation using synchrotron x-ray microtomography*, Earth and Planetary Science Letters **286** (2009), no. 1-2, 285–291.

[106] J. R. Rice, *A path independent integral and the approximate analysis of strain concentration by notches and crack*, Journal of Applied Mechanics **35** (1968), 379–386.

[107] Y. N. Robotnov, *Creep rupture in applied mechanics*, Proceedings of the 12th International Congress on Applied Mechanics (1968), 342–349.

[108] Simon Rouchier, Hans Janssen, Carsten Rode, Monika Woloszyn, Genevieve Foray, and Jean-Jacques Roux, *Characterization of fracture patterns and hygric properties for moisture flow modelling in cracked concrete*, Construction and Building Materials **34** (2012), 54–62.

[109] RE Rudd and JQ Broughton, *Concurrent coupling of length scales in solid state systems*, Physica Status Solidi B-Basic Research **217** (2000), no. 1, 251–291.

[110] Georgy N. Sankin, Yufeng Zhou, and Pei Zhong, *Focusing of shock waves induced by optical breakdown in water*, J. Acoust. Soc. Am. **123** (2008), 4071–4081.

[111] Oleg A. Sapozhnikov, Adam D. Maxwell, Brian Macconaghy, and Michael R. Bailey, *A mechanistic analysis of stone fracture in lithotripsy*, Journal of the Acoustical Society of America **121** (2007), 1190–1202.

[112] E Schlangen and EJ Garboczi, *Fracture simulations of concrete using lattice models: Computational aspects*, Engineering Fracture Mechanics **57** (1997), no. 2-3, 319–332.

[113] C. F. Shih, B. Moran, and T. Nakamura, *Energy release rate along a three-dimensional crack front in a thermally stressed body*, International Journal of Fracture **30** (1986), 79–102.

[114] LE Shilkrot, RE Miller, and WA Curtin, *Coupled atomistic and discrete dislocation plasticity*, Physical Review Letters **89** (2002), no. 2.

[115] N. Smith, G. N. Sankin, W. N. Simmons, R. Nanke, J. Fehre, and P. Zhong, *A comparison of light spot hydrophone and fiber optic hydrophone for lithotripter field characterization*, Rev. Sci. Instrum. **83** (2012), 014301.

[116] N. Smith and P. Zhong, *Stone comminution correlates with the average peak pressure incident on a stone during shock wave lithotripsy*, Journal of Biomechanics **45** (2012), no. 15, 2520–2525.

[117] E. Steiger, *FD-TD-modeling of propagation of high energy sound pulses in lithotripter-tissue-arrangements*, Ultrasonics Symposium, 1997. Proceedings., 1997 IEEE, vol. 2, oct 1997, pp. 1361 –1364 vol.2.

[118] ET Sylven, S Agarwal, CL Briant, and RO Cleveland, *High strain rate testing of kidney stones*, Journal of Materials Science-Materials in Medicine **15** (2004), no. 5, 613–617.

[119] EB Tadmor, M Ortiz, and R Phillips, *Quasicontinuum analysis of defects in solids*, Philosophical Magazine A-Physics of Condensed Matter Structure Defects and Mechanical Properties **73** (1996), no. 6, 1529–1563.

[120] Michel Tanguay, *Computation of bubbly cavitating flow in shock wave lithotripsy*, Ph.D. thesis, California Institute of Technology, 2004.

[121] Philip A. Thompson, *Compressible-fluid dynamics*, McGraw-Hill, Inc, 1972.

[122] H. Toda, E. Maire, S. Yamauchi, H. Tsuruta, T. Hiramatsu, and M. Kobayashi, *In situ observation of ductile fracture using x-ray tomography technique*, Acta Materialia **59** (2011), no. 5, 1995–2008.

[123] F. R. Tuler and B. M. Butcher, *A criterion for the time dependence of dynamic fracture*, International Journal of Fracture Mechanics **5** (1968), no. 4, 431–437.

[124] M Van Geet and R Swennen, *Quantitative 3D-fracture analysis by means of microfocus X-ray computer tomography (mu CT): an example from coal*, Geophysical Research Letters **28** (2001), no. 17, 3333–3336.

[125] M Van Geet, R Swennen, and M Wevers, *Quantitative analysis of reservoir rocks by microfocus X-ray computerised tomography*, Sedimentary Geology **132** (2000), no. 1-2, 25–36.

[126] George Z. Voyiadjis and Peter I. Kattan, *Damage mechanics*, Taylor and Francis, Boca Raton, FL, 2005.

[127] GJ Wagner and WK Liu, *Coupling of atomistic and continuum simulations using a bridging scale decomposition*, Journal of Computational Physics **190** (2003), no. 1, 249–274.

[128] M. L. L. Wijerathne, Muneo Hori, Hide Sakaguchi, and Kenji Oguni, *3D dynamic simulation of crack propagation in extracorporeal shock wave lithotripsy*, IOP Conf. Series: Materials Science and Engineering **10** (2010), no. 1.

[129] M. L. L. Wijerathne, Kenji Oguni, and Muneo Hori, *Numerical analysis of growing crack problems using particle discretization scheme*, International Journal for Numerical Methods in Engineering **80** (2009), no. 1, 46–73.

[130] Philip J. Withers and Michael Preuss, *Fatigue and damage in structural materials studied by X-ray tomography*, Annual Review of Materials Research, Vol 42 (DR Clarke, ed.), Annual Review of Materials Research, vol. 42, 2012, pp. 81–103.

[131] W. F. Wu and C. C. Ni, *Probabilistic models of fatigue crack propagation and their experimental verification*, Probabilistic Engineering Mechanics **19** (2004), no. 3, 247–257.

[132] XF Xi and P Zhong, *Dynamic photoelastic study of the transient stress field in solids during shock wave lithotripsy*, Journal of the Acoustical Society of America **109** (2001), no. 3, 1226–1239.

[133] J. N. Yang and S. D. Manning, *A simple second order approximation for stochastic crack growth analysis*, Engineering Fracture Mechanics **53** (1996), no. 5, 677–686.

[134] Wohua Zhang and Yuanqiang Cai, *Continuum damage mechanics and numerical applications*, Zhejiang University Press, Hangzhou and Springer, Heidelberg, 2011.

[135] Pei Zhong, *Shock wave lithotripsy*, Bubble Dynamics and Shock Waves (Can F. Delale, ed.), Springer, Heidelberg, 2013, pp. 291–338.

[136] Pei Zhong, Nathan Smith, Walter N. Simmons, and Georgy Sankin, *A new acoustic lens design for electromagnetic shock wave lithotripters*, 10th International Symposium on Therapeutic Ultrasound (ISTU 2010) AIP Conference Proceedings, vol. 1359, 2011, pp. 42–47.

[137] Y. Zhou and P. Zhong, *The effect of reflector geometry on the acoustic field and bubble dynamics produced by an electrohydraulic shock wave lithotripter*, J. Acoust. Soc. Am. **119** (2006), 3625–3636.

[138] TI Zohdi and AJ Szeri, *Fatigue of kidney stones with heterogeneous microstructure subjected to shock-wave lithotripsy*, Journal of Biomedical Materials Research Part B-Applied Biomaterials **75B** (2005), no. 2, 351–358.