RESONANT ULTRASOUND SPECTROSCOPY

IN COMPLEX SAMPLE GEOMETRY

by

Matthew Kenneth Fig

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Mechanical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

November 2005

APPROVAL

Of the Thesis Submitted by

Matthew Kenneth Fig

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

Dr. R. Jay Conant, Committee Chair

Approved for the Department of Mechanical and Industrial Engineering

Dr. Christopher. H. M. Jenkins, Department Head

Approved for the College of Graduate Studies

Dr. Joseph J. Fedock, Graduate Dean

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements of a master's degree at Montana State University, I agree that the library shall make it available to borrowers under rules of the library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as described in the U.S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Matthew Kenneth Fig

November, 2005

AKNOWLEDGEMENTS

The experimental measurements and data analysis for this work were completed at the Idaho National Laboratory (INL) during the summers of 2004/05 and were sponsored by the Laboratory Directed Research and Development program "Acousto-Optic MEMS." The INL is operated by Battelle Energy Alliance under DOE Idaho Operations Office Contract DE-AC07-05ID14517.

I also would like to thank the members of my graduate committee, Dr. Conant, Dr. Telschow, and Dr. Cairns for their support and encouragement over the course of this undertaking. The feedback and time spent reading drafts of this thesis is much appreciated.

Additionally, I would like to thank all those who helped me with the analytical, experimental, and computational aspects of this project. Dr. Telschow, Dr. John Walters, Dave Cottle, and Rob Schley all had a hand in making this project a success, and in making me feel welcome and valuable at the INL. Thank you.

TABLE OF CONTENTS

TABLE OF CONTENTS – CONTINUED

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF FIGURES – CONTINUED

LIST OF FIGURES – CONTINUED

ABSTRACT

Resonant Ultrasound Spectroscopy (RUS) is the study of the mechanical resonances, or normal modes, of elastic bodies to infer material properties such as the elasticity matrix. This powerful technique is based on two physical facts, the first of which is that the resonant response of an elastic object depends on several parameters intrinsic to the object, such as the object's shape, density, elastic constants, and crystallographic orientation. The second is that using these parameters, the resonant spectrum of an object can be calculated. This method has widely been applied to rectangular parallelepipeds (RPPDs) because the use of such simple geometry frees an investigator interested only in acquiring the elastic constants of a particular material from the hindrance of dealing with the additional computational difficulty imposed by more complex sample geometry. In addition to the use of RPPDs, some work has been done with other objects of high symmetry such as cylinders and spheres. The goal of this research was to explore the extension of RUS techniques to objects exhibiting more complex shape. Toward this end, a computational method was developed for handling the addition of complex geometry. This computational scheme was then verified experimentally through the examination of several objects exhibiting complex shapes.

INTRODUCTION

Basics

Resonant Ultrasound Spectroscopy (RUS) is a means of determining material properties of an elastic object by exciting the resonant frequencies (normal modes) of the object. The spectrum of modes of an elastic object contains much information about the object, both microscopic and macroscopic. The information that can be derived from such a spectrum includes the entire elasticity matrix as well as information about the object's geometry and density. In principle, all of this information can be acquired from a single, accurate measurement of an object's resonant spectrum [1]. In theory, this is done by analytically solving the inverse problem, that is, calculating an object's shape and elastic constants based on knowledge of the experimentally measured resonant spectrum.

Unfortunately, the inverse problem is not solved for most shapes. Thus the usual method of extracting the needed information from an object's resonant spectrum is to iteratively solve the forward problem [2]. This means that a numerical approximation is used to predict the resonant spectrum of the object under consideration and this approximation is then compared to the actual, measured spectrum to see how well the prediction fits. If the measured and calculated spectrums don't agree to within acceptable limits, the parameters used in the calculation are adjusted and the computation is redone. This process is repeated until the criterion for judging the goodness of fit is met. This criterion is usually that the RMS relative error of the first 20 or so frequencies is less

than 1%, preferably less than .5% [3].

This iterative approach forms the basis of the RUS method. It has widely been used to determine the material properties of objects exhibiting high symmetry, such as rectangular parallelepipeds (RPPDs), cylinders and spheres [3]. Unfortunately most objects of interest aren't of such simple shape. Thus to determine properties of practical engineering objects, the method needs to be extended to account for more complex sample geometry. This would allow for such things as inspection of objects in service to determine if there has been a fundamental change in the object that might compromise its usability. Such compromising changes might include the presence of fatigue cracks or a change in microstructure that alter the elastic properties of the object to the point where its continued use should be questioned.

History

Prior to any serious scientific inquiry, the sound of a vibrating object was known to convey information about the object. For instance, bell makers have long known that different alloys forged into the same shape make bells of different sounds, and that the presence of a crack or other imperfection, such as a large deviation from the desired shape can be detected by the differing harmonics a bell exhibits from a bell that isn't cracked or deformed. These two examples show that the resonance of an object depends not only on the material with which the object is made, but the geometry that defines the object. Knowledge of the relationship between material, shape and resonant frequency was used in a variety of fields, not just bell making. It is said that when

designing the first jet engines during World War II, German engineers employed a violinist to rub his bow over different turbine blade designs so that the resonant frequencies of the chosen blades would be very different from the angular speed of the engine itself, thus avoiding the destruction that could result from driving the blades at a resonant frequency [4]. The qualitative knowledge of the phenomena of natural resonance is useful for examples like these, but a more quantitative approach is needed for material characterization and engineering use.

The laboratory use of resonance techniques to infer material properties dates back to the 1920s [5]. These experiments used long (length/diameter>10) cylindrical bars, as the resonant spectrum for this shape can be attained analytically from first principles. This type of investigation has a serious limitation, namely that the material used for testing must be available in such an awkward shape. Later, others investigated the use of spherical samples. The focus of these investigations was the free resonance of the Earth after a large earthquake. It was found that one could determine the radial density variation of the Earth with these models. These investigators did not solve the inverse problem for this shape but relied on tables of special functions to calculate the expected resonant modes [6].

The real breakthrough came in the late 1960s with the work of Holland and Demarest, who approached the problem by numerical methods based on variational principles [7-8]. The applicability of these methods was, however, limited to samples with rectangular geometry, as the basis functions chosen were the normalized Legendre polynomials. Still a complete set of lower frequency resonant modes could accurately

be predicted, and even the mode shapes could be calculated with this method. Even with the limitation to rectangular geometry, this technique soon proved its usefulness to materials scientists who could apply it to samples of virtually any available size, as long as the shapes were rectangular. In fact the biggest hurdle during the early years was the limited computational resources available to carry out the required number of calculations. With the advent of modern PC computing power, this is no longer the case.

Mathematical Development

As stated above, the mathematical foundation of RUS is the classical variational approach to the problem taken by Holland [7] and Demarest [8]. An approach similar to this will be used here [9].

Classical mechanics tells us that the general form of the Lagrangian L is

$$\hat{L} = \int_V (KE\text{-}PE)dV \tag{1.1}$$

where KE is the kinetic energy and PE is the potential energy of the body with volume V. Now for a linear elastic body of arbitrary shape, the kinetic energy is given by

$$KE = \frac{1}{2}\rho \frac{\partial \hat{u}_i}{\partial t}\frac{\partial \hat{u}_i}{\partial t} \tag{1.2}$$

where the displacements are given by the u_i terms and ρ is the material density. Here and throughout the rest of the development, Einstein notation is used. This means that repeated indices in a single term of an equation indicate summation over the range of each index. The indices i,j,k,l have a range from 1 to 3 corresponding to the 3

coordinate directions. If one assumes the time dependence of the displacement is that of harmonic motion given by

$$\hat{u}_i(\vec{x},t) = u_i(\vec{x})e^{i\omega t} \tag{1.3}$$

then the expression for the kinetic energy becomes

$$KE = \frac{1}{2}\rho\omega^2 u_i u_i e^{2i\omega t} \tag{1.4}$$

The potential energy is, in the absence of external forces, the strain energy **U**, given by [25],

$$PE = \mathbf{U} = \frac{1}{2}C_{ijkl}\varepsilon_{ij}\varepsilon_{kl} \tag{1.5}$$

If (1.4) and (1.5) are substituted into (1.1), the Lagrangian becomes

$$\hat{L} = \frac{1}{2}\int_V \left(\rho\omega^2 u_i u_i e^{2i\omega t} - \frac{1}{2}C_{ijkl}\varepsilon_{ij}\varepsilon_{kl}\right)dV \tag{1.6}$$

In what follows it will be convenient to transform this equation to matrix form. To this end, note that if we define

$$\{\hat{u}\} = \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} e^{i\omega t} = \{u\}e^{i\omega t} \tag{1.7}$$

then

$$u_i u_i = \left(u_1^2 + u_2^2 + u_3^2\right) = \{u\}^T\{u\} \tag{1.8}$$

which can be substituted into the first term of (1.6). To transform the second term of (1.6), a more circuitous route must be taken. First it is noticed that the second term

contains part of Hooke's law for a 3-dimensional linearly elastic solid, given by

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl} \tag{1.9}$$

where σ_{ij} is the stress tensor, C_{ijkl} is the elasticity (or stiffness) tensor, and ε_{kl} is the strain tensor. Thus the strain energy term can be written as

$$U = \frac{1}{2}C_{ijkl}\varepsilon_{ij}\varepsilon_{kl} = \frac{1}{2}C_{ijkl}\varepsilon_{kl}\varepsilon_{ij} = \frac{1}{2}\sigma_{ij}\varepsilon_{ij} \tag{1.10}$$

For the current problem, it is assumed that there are no body torques so that the stress tensor is symmetric. This means that $\sigma_{ij} = \sigma_{ji}$ so that i and j may be interchanged in (1.9). Additionally, since the engineering strain tensor is given by [25],

$$\varepsilon_{kl} = \frac{1}{2}\left(\hat{u}_{k,l} + \hat{u}_{l,k}\right) = \frac{1}{2}\left(u_{k,l} + u_{l,k}\right)e^{i\omega t} \tag{1.11}$$

the strain tensor is symmetric as well. These two observations taken together imply that the elasticity tensor is symmetric in its first two indices as well as in its second two indices, which means $C_{ijkl} = C_{jikl} = C_{ijlk} = C_{jilk}$ in (1.9). It also can be shown from energy considerations [24] that for the elasticity tensor

$$C_{ijkl} = C_{klij} \tag{1.12}$$

Since all three tensors that appear in Hooke's law are symmetric, the Voight contraction notation can be used. The Voight notation simplifies tensor representation by reducing the dimension of the contracted tensors. This is done through the use of specific rules for grouping pairs of indices so that when certain combinations of indices occur they are replaced with a single index. The rules for the Voight contraction notation apply only to symmetric tensors. These index rules are given below in (1.13).

$$\begin{aligned} 11 &\to 1 \qquad & 12 = 21 &\to 6 \\ 22 &\to 2 \quad \text{and} \quad & 23 = 32 &\to 4 \\ 33 &\to 3 \qquad & 13 = 31 &\to 5 \end{aligned} \tag{1.13}$$

After applying these rules, the elasticity tensor transforms from a 3x3x3x3 hypercube to a 6x6 square matrix because the indices of this tensor are taken two at a time from left to right. Both the stress tensor and the strain tensor transform from a 3x3 tensor to a 6x1 vector. The alternative representation of Hooke's law is then

$$\{\sigma\} = [C]\{\varepsilon\} \tag{1.14}$$

where

$$\{\sigma\} = \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{Bmatrix} \tag{1.15}$$

and the strain is written as the vector

$$\{\varepsilon\} = \begin{Bmatrix} u_{1,1} \\ u_{2,2} \\ u_{3,3} \\ \frac{1}{2}\left(u_{2,3} + u_{3,2}\right) \\ \frac{1}{2}\left(u_{1,3} + u_{3,1}\right) \\ \frac{1}{2}\left(u_{1,2} + u_{2,1}\right) \end{Bmatrix} e^{i\omega t} \tag{1.16}$$

and with the elasticity tensor as below in (1.17).

$$[C] = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \tag{1.17}$$

Next, the expression for the strains can be put in terms of the displacements by rewriting (1.16) as a product of the symmetric gradient operator and the displacement vector (1.7) as below

$$\{\varepsilon\} = [B]\{\hat{u}\} = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} \\ 0 & \frac{1}{2}\frac{\partial}{\partial x_3} & \frac{1}{2}\frac{\partial}{\partial x_2} \\ \frac{1}{2}\frac{\partial}{\partial x_3} & 0 & \frac{1}{2}\frac{\partial}{\partial x_1} \\ \frac{1}{2}\frac{\partial}{\partial x_2} & \frac{1}{2}\frac{\partial}{\partial x_1} & 0 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} e^{i\omega t} \tag{1.18}$$

Finally, to transform the expression for the strain energy to the needed matrix form, substitute (1.15) and (1.16) into (1.10), then use (1.14) to get

$$\mathbf{U} = \frac{1}{2}\sigma_{ij}\varepsilon_{ij} = \frac{1}{2}\{\sigma\}^T\{\varepsilon\} = \frac{1}{2}\{\varepsilon\}^T[C]^T\{\varepsilon\} = \frac{1}{2}\{\varepsilon\}^T[C]\{\varepsilon\} \tag{1.19}$$

Then (1.18) can be substituted for the strains in (1.19) to yield

$$\mathbf{U} = \frac{1}{2}\{\varepsilon\}^T[C]\{\varepsilon\} = \frac{1}{2}\{u\}^T[B]^T[C][B]\{u\}e^{2i\omega t} \tag{1.20}$$

so that the expression for the Lagrangian becomes

$$\hat{L} = \frac{1}{2}\int_V \left(\rho\omega^2 \{u\}^T \{u\} - \{u\}^T [B]^T [C][B]\{u\}\right) dV e^{2i\omega t} \tag{1.21}$$

Since the right hand side of (1.21) is harmonic, the left hand side must also be harmonic. Consequently (1.21) can be rewritten as

$$\hat{L} = Le^{2i\omega t} \tag{1.22}$$

where

$$L = \frac{1}{2}\int_V \left(\rho\omega^2 \{u\}^T \{u\} - \{u\}^T [B]^T [C][B]\{u\}\right) dV \tag{1.23}$$

The expression (1.23) is the required Lagrangian of the body in matrix form. With the Lagrangian in this form, the next task is to put the displacements in a form that will allow them to be evaluated numerically. Perhaps the easiest way to do this is to expand each component of the displacement in a Cartesian power series of the form

$$(\Psi_\lambda) = \left(x^p y^q z^r\right) \tag{1.24}$$

where $p,q,r \in \mathbb{N}$. With this choice of basis the form of the displacements becomes

$$u_i = \sum_{(p,q,r)}^{\infty} a_{i(p,q,r)} \Psi_{(p,q,r)} = a_{i\lambda} \Psi_\lambda \tag{1.25}$$

where (p,q,r) are discrete elements of the basis set, λ and the $a_{i\lambda}$ coefficients are constants. In general the expansion (1.25) cannot be summed to infinity, thus a truncation condition will be imposed. The common practice is to limit the highest exponent in (1.24) by the condition given as in 1.26 below.

$$p+q+r \leq N \tag{1.26}$$

This is the needed truncation condition. Thus for $N \to \infty$, (1.25) exactly represents the displacements [9]. In practice, N is limited to the lowest number that will allow a good compromise between accuracy and computational time. This truncation condition limits the number of terms in the polynomial appearing in (1.25). To find out how many terms, as a function of N, will appear in the polynomial expansion of each component of the displacement, one must determine the number of combinations of exponents p,q,r that obey (1.26) for a given N. Since finding the needed number is a combination problem, the solution will be of the form

$$\text{Number of Combinations} = \binom{\gamma}{\zeta} = \frac{\gamma!}{\zeta!(\gamma-\zeta)!} \tag{1.27}$$

where the middle term is the binomial coefficient. For N = 0, the number of combinations is 1, given by (p,q,r) = (0,0,0). For N = 1, the number of combinations is 4, given by (p,q,r) = (0,0,0) (1,0,0) (0,1,0) and (0,0,1). As can easily be checked, the number of combinations for N = 2 and N = 3 is 10 and 20 respectively. The sequence of numbers 1,4,10,20… is recognized as the tetrahedral sequence, found on the fourth diagonal of Pascal's triangle. Thus the correct number of combinations for (1.26) is given by

$$\text{Number of Combinations} = \binom{N+3}{3} = \frac{(N+3)!}{3!\big((N+3)-3\big)!} \tag{1.28}$$

which can be rewritten as in (1.29) below.

$$\binom{N+3}{3} = \frac{(N+3)!}{3!N!} = \frac{(N+3)!}{6N!} \tag{1.29}$$

With (1.25) the vector $\{u\}$ is redefined as the product of a matrix $\{\Phi\}$ of monomials and a column vector of coefficients. The matrix of monomials will be $3\times R$, with R derived from (1.29). Since there are three components of the displacement vector, R is given as

$$R = 3\frac{(N+3)!}{6N!} = \frac{(N+3)!}{2N!} \tag{1.30}$$

Furthermore $\{\Phi\}$ is block diagonal so that each row contains a vector $\{\varphi\}^T$ and two zero vectors, all of length R/3. Thus the general form of the displacement vector is written as

$$\begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{bmatrix} \{\varphi\} & \{0\} & \{0\} \\ \{0\} & \{\varphi\} & \{0\} \\ \{0\} & \{0\} & \{\varphi\} \end{bmatrix}^T \begin{Bmatrix} \{\alpha_1\} \\ \{\alpha_2\} \\ \{\alpha_3\} \end{Bmatrix} = \{\Phi\}\{a\} \tag{1.31}$$

For example, if N = 2, the row vector $\{\varphi\}^T$ is of the form (notice there are 10 terms)

$$\{\varphi\}^T = \begin{Bmatrix} 1 & x & y & z & xy & xz & yz & x^2 & y^2 & z^2 \end{Bmatrix} \tag{1.32}$$

and the column vectors $\{\alpha_i\}$ are the coefficients corresponding to each term in (1.32).

If the representation of the displacements (1.25) is substituted into (1.23), the Lagrangian becomes

$$L = \frac{1}{2}\int_V \left(\rho\omega^2 \{a\}^T [\Phi]^T [\Phi]\{a\} - \{a\}^T [\Phi]^T [B]^T [C][B][\Phi]\{a\}\right) dV \tag{1.33}$$

or, since the coefficient vectors $\{a\}$ are constants with respect to the volume integration,

$$L = \frac{1}{2}\omega^2 \{a\}^T \int_V [\Phi]^T \rho [\Phi]\, dV \{a\} - \frac{1}{2}\{a\}^T \int_V [\Phi]^T [B]^T [C][B][\Phi]\, dV \{a\} \tag{1.34}$$

or more compactly,

$$L = \frac{1}{2}\omega^2 \{a\}^T [E]\{a\} - \frac{1}{2}\{a\}^T [\Gamma]\{a\} \tag{1.35}$$

where the matrices $[E]$ and $[\Gamma]$ are defined as:

$$[E] = \int_V [\Phi]^T \rho [\Phi] dV \tag{1.36}$$

and

$$[\Gamma] = \int_V [\Phi]^T [B]^T [C][B][\Phi] dV \tag{1.37}$$

It is noteworthy that both the $[E]$ and $[\Gamma]$ matrices are symmetric and square, and also $[E]$ is positive definite. Also, since the elasticity matrix and the density have been left inside the volume integrals, bodies for which these properties exhibit spatial variation can be handled with these expressions.

Now that the correct form of the Lagrangian has been found, one can find the variation of the Lagrangian with variation in displacement. First it is noticed that the Lagrangian in (1.35) is a function of the coefficients only. Consequently, the variation of L is written as in (1.38) below.

$$\begin{aligned}\delta L = &\frac{\partial L}{\partial a_{10}}\delta a_{10} + ... + \frac{\partial L}{\partial a_{1(R/3)}}\delta a_{1(R/3)} \\ &+ \frac{\partial L}{\partial a_{20}}\delta a_{20} + ... + \frac{\partial L}{\partial a_{2(R/3)}}\delta a_{2(R/3)} \\ &+ \frac{\partial L}{\partial a_{30}}\delta a_{30} + ... + \frac{\partial L}{\partial a_{3(R/3)}}\delta a_{3(R/3)}\end{aligned} \tag{1.38}$$

In the equilibrium state, L must be stationary for any variation. But in order for the variations of each component of the displacements to be arbitrary, each coefficient must vary arbitrarily. Since each coefficient is free to vary arbitrarily, the only way for $\delta L = 0$ to be true is if each term in (1.38) is independently zero. This can be summarized as

$$\frac{\partial L}{\partial \{a\}} = \omega^2 \{a\}^T [E] - \{a\}^T [\Gamma] = \{0\}^T \tag{1.39}$$

which can be rewritten

$$\omega^2 \{a\}^T [E] = \{a\}^T [\Gamma] \tag{1.40}$$

If the transpose of both sides of (1.40) is taken, and the symmetry of $[E]$ and $[\Gamma]$ is exploited, (1.40) can be written in the form commonly found in the RUS literature. The result of these manipulations is given by

$$\omega^2 [E]\{a\} = [\Gamma]\{a\} \tag{1.41}$$

Equation (1.41) is recognized as a generalized eigenvalue equation with eigenvalues ω^2 and eigenvectors $\{a\}$. It is this equation that will be solved in order to acquire both the normal mode frequencies, obtained from the eigenvalues ω^2, and the physical displacements of the oscillating body, given by the eigenvectors $\{a\}$. Thus the problem

of finding the resonant frequencies of elastic bodies has been reduced to the evaluation of the volume integrals (1.36) and (1.37).

As previously mentioned, there are an infinite number of eigenvalues for any particular problem, and as no closed form expression is available for most shapes, a numerical solution to (1.41) is sought. This necessitates a truncation of the infinite set to one that is of manageable size for a computer to solve. It is known that when an infinite eigenvalue equation is solved on a truncated set, the ordered solutions of the truncated set are greater than the actual eigenvalues [10]. Thus N is usually varied until the first 50 or so modes appear to converge. The standard value for N when using simple geometry is around 10 but can be as high as 20, which leads to matrices with rank that ranges from 858 to 4620. The number of modes needed to determine a set of material parameters will depend on the number of parameters one is investigating. For example, a highly anisotropic material may have many (up to 21 are possible) independent elastic constants that are unknown, whereas an isotropic material will only have two independent elastic constants. It has been suggested that the number of modes one needs to accurately fit the elastic constants of a material under investigation should be at least 5 and preferably 8-10 times the number of independent elastic constants [1]. This suggestion is not based in theory, but rather based on the consensus of experienced RUS practitioners.

Expressions (1.36) and (1.37) can be evaluated analytically for RPPDs, cylinders, spheroids and ellipsoids. For a homogeneous RPPD with sides of length 2α, 2β and 2γ, both expressions (1.36) and (1.37) are of the same general form. Let the generic form of these expressions be given by f as shown below in (1.42).

$$f = \int_{-\alpha}^{\alpha}\int_{-\beta}^{\beta}\int_{-\gamma}^{\gamma} x^p y^q z^r dxdydz \tag{1.42}$$

This integral is easily evaluated in closed form, given by

$$f = \frac{8\alpha^{p+1}\beta^{q+1}\gamma^{r+1}}{(p+1)(q+1)(r+1)} \tag{1.43}$$

For a homogeneous right circular cylinder of length 2Δ and radius r, the correct integral is given by

$$f = \int_{-\Delta}^{\Delta}\int_{-r}^{r}\int_{-\sqrt{y^2-r^2}}^{\sqrt{y^2-r^2}} x^p y^q z^r dxdydz \tag{1.44}$$

which also has a closed form solution [11]. This solution is

$$f = \frac{4\pi r^{p+q+2}\Delta^{r+1}(p-1)!!(q-1)!!}{(r+1)(p+q+2)!!} \tag{1.45}$$

where the double factorial represents the product of the alternate positive integers up to the argument. Notice in both of these examples that the limits of integration are the equations that define the surface of the body, because the integration is performed over the sample volume.

The value of these integrals has been found analytically for other simple geometries [9]. The RUS method has enjoyed a large degree of success with simple geometries such as those listed above. Materials that have been investigated by this method include such common materials as aluminum, glass and steel, but also more exotic, anisotropic metals like plutonium and super-hard materials like decagonal Al-Ni-

Co and icoshedral Y-Mg-Zn have successfully been characterized by RUS [3]. Thus as a tool for the materials scientist, RUS has established itself. The goal of the current research is to extend this method to investigate its usefulness for the engineering or NDE community by evaluating objects with more complex geometry.

Thesis Goal

The goal of this thesis is to investigate the applicability of RUS methods to objects with complex shape. This will be accomplished by choosing several sample objects of complex geometric shape but known material properties, specifically isotropic aluminum, and using numerical integration techniques to evaluate the integrals appearing in (1.36) and (1.37). The approach used here to verify the RUS method as a tool for complex sample geometry will be taken in two steps. First a plane, right circular cylinder will be machined from the test aluminum and the elastic constants C_{11} and C_{44} will be found using standard RUS techniques. With this information, the resonant frequencies and mode shapes of several geometrically complex samples made from the same material will then be predicted and verified experimentally.

INVESTIGATION OF NUMERICAL INTEGRATION

The first step in seeking to apply the RUS method to samples with complex geometry is to find a way to evaluate the integrals appearing in (1.36) and (1.37) numerically because for most shapes, there will be no closed form solutions such as that found in (1.45). To find the best numerical approach for this application, several techniques were examined, including the trapezoidal rule, Simpson's rule, Monte-Carlo integration, Sinc function integration and Gaussian quadrature. Each of these methods was in turn demonstrated and finally compared so as to arrive at the optimal method for the present problem. The results of this process are given below.

Trapezoidal Rule

The trapezoidal rule is one of the oldest and most easily understood methods used to evaluate integrals numerically. To begin an examination of this technique, start by discussing the approximation of the integral of a general function f(x). The integral of f(x) from x_0 to x_n is given by the area between the curve and the x-axis between these two points. Choosing any integral number $n > 1$ of equal length intervals between these two points and evaluating the function at the endpoints of these intervals allows one to approximate this area by forming a series of trapezoids and summing their areas. This is done by forming the sum as given by

$$\int_A^B f(x)dx \cong \frac{1}{2}\sum_{j=0}^{n-1}(x_{j+1}-x_j)\left[f(x_{j+1})+f(x_j)\right] \qquad (2.1)$$

An example for n=2 is illustrated below in Figure 2.1. With two intervals between x_0 and x_n the approximation is poor. As the number of intervals between the two endpoints increases, so does the accuracy of the approximation.

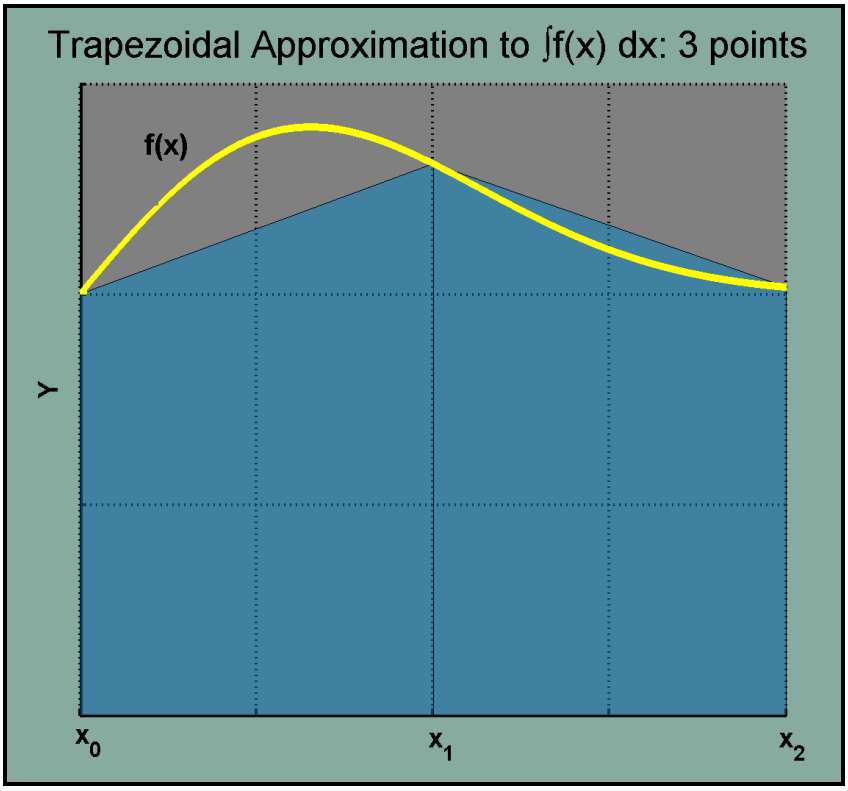

Figure 2.1 Trapezoidal Approximation to the Integral f(x). Here two intervals are used between the endpoints. The area in blue is the area of the approximation.

Notice here that the first trapezoid underestimates the area and the second trapezoid overestimates the area. This is a prominent feature of the trapezoidal rule. Sometimes the areas of overestimation cancel the areas of underestimation and sometimes the two act together to give an overall poor estimate of the integral. As the number of intervals is increased indefinitely, convergence to the true value of the integral is guaranteed [12]. It is, however, impossible to calculate an infinite number of terms so in practice the number of terms calculated is usually limited by choosing a

specified tolerable error and calculating the difference between successive terms to estimate this error. If the number of intervals is increased to 4, the approximation is much better. The new approximation is shown in Figure 2.2.

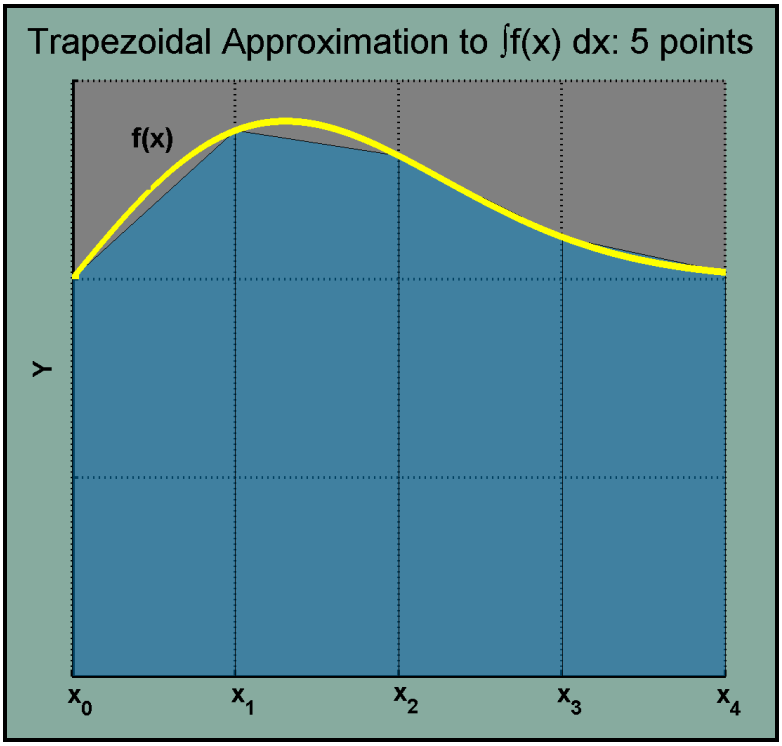

Figure 2.2 New Trapezoidal Approximation to the Integral f(x). Here two more intervals have been added, resulting in a much better approximation to the definite integral. Notice, however, that there are still gaps in the left intervals.

The trapezoidal rule is known as a 2^{nd} order method. This is because the error incurred when using this method is a 2^{nd} order error [27]. As such, the error behaves as

$$\text{Error} \approx O(h^2) \tag{2.2}$$

where the h is the interval length. Thus for this method, the error can be made arbitrarily small with the use of small intervals. The drawback to using extremely small intervals is that, with smaller intervals, more function evaluations are needed. With (2.2) as a guide, an estimate of the error incurred when using the trapezoidal rule can be attained.

Simpson's Rule

Simpson's rule in general gives better results than the trapezoidal rule for the same number of terms. This is because Simpson's rule is a 4^{rth} order method [27]. As such, the error resulting from use of Simpson's rule behaves as

$$\text{Error} \approx O(h^4) \tag{2.3}$$

where again, h is the size of the interval used to partition the domain of integration. Thus, as was the case with the trapezoidal rule, the error can be estimated and controlled by using (2.3) as an approximation. A comparison of equations (2.2) and (2.3) reveals the advantage of Simpson's rule over the trapezoidal rule. Namely, for the same partition the Simpson's rule has a smaller associated error.

Simpson's rule also uses the function values at specified points, but instead of interpolating between two points with a straight line, this method uses a quadratic polynomial (parabolic arc) to interpolate between the points [12]. Thus every interval needs three points to successfully describe the interpolation, the endpoints can be the same as those from the intervals used with the trapezoidal rule, but there must also be a point chosen between each set of endpoints; the usual choice is the midpoint. The mathematical basis of this process is described by considering the definite integral of f(x) from x_0 to x_n. In order to find a quadratic interpolation, a polynomial must be used which, considering the first interval, has the form

$$g(x) = \alpha x^2 + \beta x + \gamma \tag{2.4}$$

with the constants α, β, γ chosen to satisfy the constraint equations given in (2.5) below.

$$f(x_0) = g(x_0), \quad f\left(\frac{x_0+x_1}{2}\right) = g\left(\frac{x_0+x_1}{2}\right) \text{ and } f(x_1) = g(x_1) \tag{2.5}$$

The equations in (2.5) are the interpolation conditions. Solving for α, β and γ gives the coefficients for use on this interval. This process must then be repeated for every interval used. The forgoing can be avoided by setting

$$\int_{x_0}^{x_1} f(x)\,dx = \int_{x_0}^{x_1} g(x)\,dx = \int_{x_0}^{x_1} \alpha x^2+\beta x+\gamma\,dx \tag{2.6}$$

After evaluation of the third integral in (2.6) using (2.5), the approximation becomes

$$\int_{x_0}^{x_1} f(x)\,dx = \frac{x_1 - x_0}{2}\left[f(x_0)+4f\left(\frac{x_0+x_1}{2}\right)+f(x_1)\right] \tag{2.7}$$

Now if the interval between x_0 and x_n is broken up into n smaller intervals, a sum must be formed to get an expression for the total approximate integral. This sum is given by

$$\int_{x_0}^{x_n} f(x)\,dx \cong \frac{h}{3}\sum_{j=0}^{n-2}\left[f(x_0+2jh)+4f\left(x_0+(2j+1)h\right)+f\left(x_0+2(j+1)h\right)\right] \tag{2.8}$$

where

$$h = \frac{x_n - x_0}{2n} \tag{2.9}$$

Figure 2.3 shows a two interval, five point Simpson's rule approximation to the same function f(x) as is shown in Figure 2.1. Notice how much better the approximation of the area is with the Simpson's rule than with the trapezoidal rule.

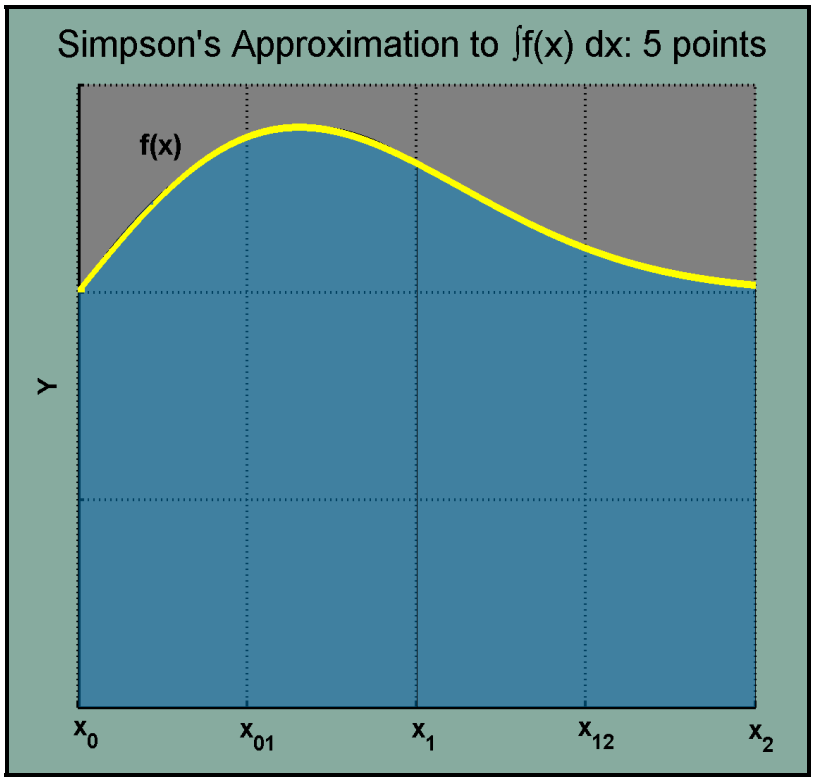

Figure 2.3 Simpson's Approximation to the integral f(x). This is the two interval, 5-point approximation for comparison with the two interval trapezoidal approximation shown in Figure 3.1.

Sinc Function Integration

Another, newer method of numerical integration was investigated in looking for the most efficient approach for the current problem, namely the Sinc function approximation. This method involves the Sinc function, defined by

$$\mathrm{Sinc}(x) = \frac{\sin(x)}{x} \tag{2.10}$$

which is used to expand a function f(x) in the form of an infinite series, given by

$$f(x) = \sum_{k=-\infty}^{\infty} f(kh)\,\mathrm{Sinc}\left(\frac{\pi(x-kh)}{h}\right) \tag{2.11}$$

For appropriately chosen h, (2.11) converges to f(x) for all x [13]. The principle advantage of using such a series is that it can handle singularities in f(x), whereas other types of approximations cannot. Using this expansion, the definite integral of f(x) can be found by integrating (2.9) term by term. This is possible because (2.10) has a power series representation, which is

$$\mathrm{Sinc}(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n+1)!} \tag{2.12}$$

The integral of f(x), written in terms of these series, becomes

$$\int f(x)dx = \sum_{k=-\infty}^{\infty} \left[f(kh) \sum_{n=0}^{\infty} \left(\frac{\pi}{h}\right)^{2n} \frac{(-1)^n (x-kh)^{2n+1}}{(2n+1)(2n+1)!} \right] \tag{2.13}$$

Although this method the advantage that it can handle irregular functions, there are serious drawbacks to using (2.13) for the current work. The first is that the h appearing in equations (2.10) and (2.13) must be carefully chosen to ensure convergence of (2.13) on all x. Second, for each function f(x), h must be chosen separately in order to ensure optimal convergence [14]. Thirdly, if h is not optimally chosen, (2.13) can be extremely slow to converge, much slower than even the trapezoidal rule described earlier. In addition, the ability to handle singularities is of little value for the present endeavor because the power series in (1.25) introduces no singularities to either (1.36) or (1.37).

Gaussian Quadrature

Another method of numerical integration investigated in detail was Gaussian quadrature. This form of numerical integration has its basis in a rather simple idea. The idea is that one should not be restricted to sampling the function to be integrated at the endpoints of equal length intervals, like the trapezoid rule and Simpson's rule require, but rather one should pick the optimal points to evaluate the function so that the error in approximating the integral is minimized. These optimal points are called the Gauss-points of the function and are defined by the fundamental theorem of Gaussian quadrature. This theorem states that the optimal sampling points are the roots of the orthogonal polynomial for the same interval and weighting function [15]. The most commonly used family of polynomials for Gaussian quadrature are the normalized Legendre polynomials, given by the solutions of the Legendre differential equation

$$\frac{d}{dx}\left[\left(1-x^2\right)\frac{dy}{dx}\right]+m\left(m+1\right)y=0 \tag{2.14}$$

for integer m. Legendre polynomials have an associated weighting equal to one. This makes the formula for Gauss-Legendre quadrature easier to derive. The interval on which the Legendre polynomials are used is [-1,1], thus the function to be integrated must also be mapped to this interval. One way to accomplished this mapping is to make the following substitution

$$\int_{x_0}^{x_1} f(x)dx = \int_{-1}^{1} f\left[\left(\frac{x_1+x_0}{2}\right)+u\left(\frac{x_1-x_0}{2}\right)\right]du = \int_{-1}^{1} g(u)du \tag{2.15}$$

With this new function definition, the integral can be analyzed with the Gauss-Legendre quadrature by

$$\int_{-1}^{1} g(u)du = \sum_{j=1}^{\infty} w_j g(x_j) \tag{2.16}$$

where the x_j are the roots of the Legendre polynomials called the abscissa, and the w_j are the corresponding weights. The weights are calculated with the following

$$w_j = \frac{2}{\left(1 - x_j^2\right)\left[P_n'(x_j)\right]^2} \tag{2.17}$$

where the P_n' is the derivative of the n^{th} polynomial, evaluated at its zero, x_j. An alternate formulation that is easier to use for finding the abscissa and weights of an n-term quadrature is to numerically solve the following eigenvalue problem

$$A\lambda = V\lambda \tag{2.18}$$

where the eigenvalues, λ, are the abscissa and the weights are derived from the first element v_j of each normalized eigenvector as

$$w_j = v_j^2 \sum_{k=1}^{n} w_k = 2v_j^2 \tag{2.19}$$

The matrix **A** in (2.18) is a tridiagonal n×n matrix with zeros on the main diagonal. The form of this matrix is given in (2.20) below.

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{1}{\sqrt{4(1)^2-1}} & 0 & \cdots & 0 \\ \frac{1}{\sqrt{4(1)^2-1}} & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \frac{(n-1)}{\sqrt{4(n-1)^2-1}} \\ 0 & \cdots & 0 & \frac{(n-1)}{\sqrt{4(n-1)^2-1}} & 0 \end{bmatrix} \quad (2.20)$$

The formulation (2.18) is easy to employ as part of the integration routine as (2.20) is easily constructed and most mathematical programming environments have built-in functions to solve eigensystems like this [16].

Monte-Carlo Integration

Monte-Carlo integration methods are different from other the other numerical methods in that the integral of the function isn't evaluated over the domain of the integrand. In one variation of this technique, the function is instead evaluated over a simpler domain Ω, for which the domain Θ of the integrand is a subdomain. The method works by generating a set of N points, usually randomly but sometimes on a grid, within Ω and calculating the number H of these points that lie within Θ. The needed integral is then H/N multiplied by the integral of the function over Ω [28]. Since Ω is chosen to be as simple as possible, the problem is reduced to the calculation of a simpler integral and then figuring out a way to tell when a particular point is in Θ. The process is illustrated

in Figure 2.4

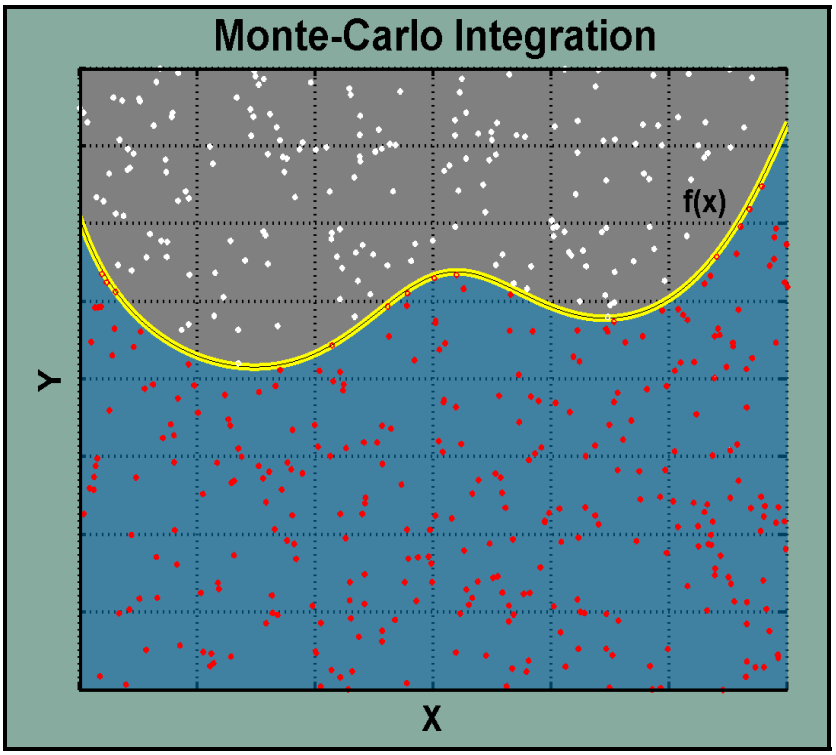

Figure 2.4 Monte-Carlo Integration. Here the sampling points that are in Θ are red, the true area of integration, called Θ in the text, is in blue and the simple domain Ω, is the entire plotting area box, including subdomain Θ.

Comparison of Quadrature Methods and Selection

All of the quadrature methods discussed above can be generalized to work for three dimensional regions, like those that appear in equations (1.36) and (1.37). This was done and a comparison was made between the methods for a region of the general form described in Figure 3.1 and associated text. The criteria used for determining which method worked best for the current problem were computational speed and adaptability, meaning the ease with which the computer code can be altered to allow integration of different shapes.

The computational speed was judged based on the time to achieve the same degree of accuracy for each numerical method. For each method, a successful trial consisted of calculating a volume within 1% relative error of the actual sample volume for a circular cylinder. All tests were run on the same machine using Matlab code written specifically for this purpose. The results are summarized in Table 2.1.

Table 2.1 Numerical Integration Comparisons. To obtain these results, each method was executed 900 times in a loop, thus the runtime represents the total time through the loop. The error listed for the Monte-Carlo method was the average error for the 900 executions.

	Monte-Carlo	Sinc	Simpson's Rule	Trapezoid Rule	Gauss-Legendre
Runtime (s)	12.21	3.10	0.36	0.21	0.13
Error (%)	0.46	0.45	0.46	0.45	0.46

As shown in Table 2.1, for the same level of accuracy Gaussian quadrature is faster than the next fastest method by a factor of 1.6. The reason the Monte-Carlo method is so much slower is that the error resulting from this method is different each time it is executed. This is due to the statistical nature of the method, sometimes the randomly generated field gives a better answer than other times. Thus in order to achieve an average error of less than 1%, a large number of sampling points had to be used, well beyond the number of iterations required by the next slowest method, the Sinc method.

The Sinc method requires a large number of iterations because, as can be seen by examining (2.13), this method requires a nested summation. This nested summation not only increases the time of execution, but also increases the number of iterations required to arrive at any degree of accuracy. To achieve the results in Table 2.1, the outer sum in (2.13) was truncated to $k = 25$, while the inner sum was truncated to $n = 100$

which means that a total of 5100 iterations was needed. Comparing this to 690 iterations needed by the trapezoid rule and 180 needed by Simpson's rule, it is seen that the Sinc method is very inefficient, but this is not the only problem with the Sinc method. Part of the difficulty in using the Sinc method is that the value for h appearing in (2.13) is not prescribed, rather it is chosen to optimize the calculation speed by trial and error. After investigating the use of (2.13) for several different h values, it was found that for even a simple function like

$$\int x^2 dx \tag{2.21}$$

the series (2.13) needs 950 iterations to reach 1% relative error.

The other criterion used to evaluate the different numerical integration methods, that of adaptability, turned out to provide less insight than the first. All of these schemes are about equally adaptable. Each requires only moderate changes in the program for use on different functions, with the exception of the Monte-Carlo method. This method is a bit more cumbersome to alter because, in order to achieve the best results, the domain Ω should enclose the domain Θ as tightly as possible. So when a new domain of integration is chosen, one must recalibrate Ω to obtain the closest fit. Thus, the numerical integration scheme chosen for the present purpose, based primarily on the computational speed comparison, is Gaussian quadrature using Legendre polynomials. A three dimensional generalization of this method was implemented in a Fortran routine so that the integrals appearing in (1.36) and (1.37) could be evaluated for the sample geometries

illustrated in Figures 3.1 and 3.2. The Gaussian quadrature algorithm implemented for these evaluations used 22 Gauss points. Although the number of Gauss points needed was tested up to 40, little change in the predicted frequencies was noticed beyond the 22 Gauss points finally chosen for use in the algorithm.

METHODS AND MATERIALS

Samples

The samples chosen for this thesis were grouped into two classes. Each class was chosen to investigate a different aspect of the use of RUS for complex sample geometry. The purpose of both classes as well as a physical description is given below.

The first class, which will be called the 'Plane Cut' class, consists of three right circular cylinders with a plane cut parallel to the axis of the cylinder so that one symmetry plane is destroyed. The purpose of destroying the sample symmetry is that this is one of the simplest ways of creating more complexity in the geometry. Three samples of this general shape were prepared, each with the same length and diameter (2.54 cm and 1.27 cm respectively), but with differing depth of cut, as illustrated in Figure 3.1.

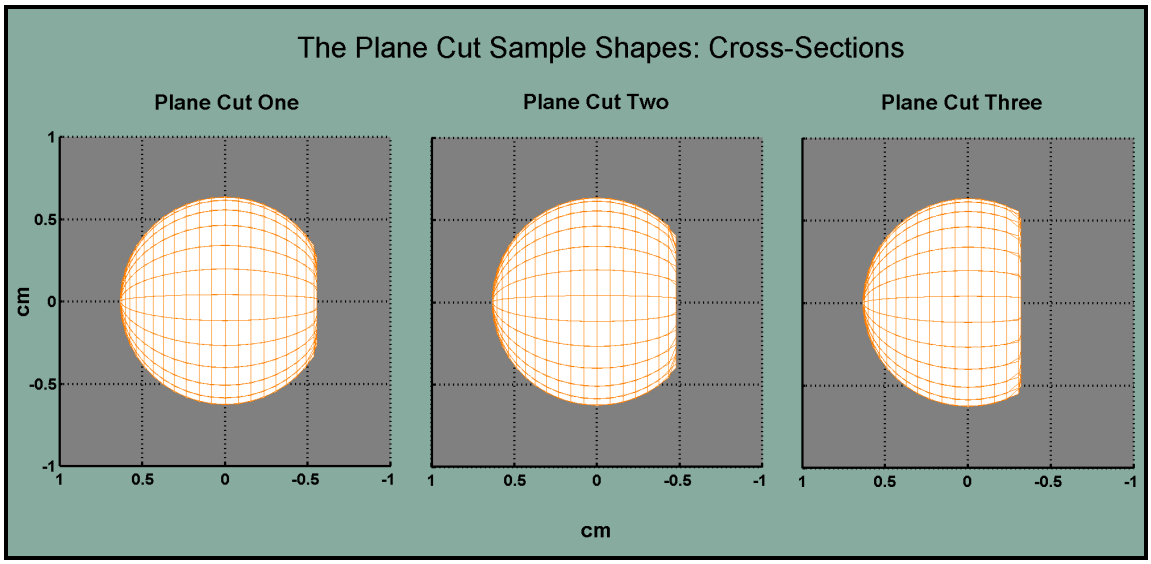

Figure 3.1 Plane Cut Geometry. The cross-sectional shape of each of the three Plane Cut samples is shown. Each sample is essentially a right circular cylinder that has a plane cut into its side so that the normal of the plane is normal to the cylindrical axis.

A complete description of all relevant geometrical information for the three Plane Cut samples is listed in Table 3.1.

Table 3.1 Plane Cut Geometric Data. This table contains the dimensions of the three Plane Cut samples. Note Depth of Cut values are rounded.

	Plane Cut 1	Plane Cut 2	Plane Cut 3
Sample Length (cm)	2.54	2.54	2.54
Sample Diameter (cm)	1.27	1.27	1.27
Depth of Cut (cm)	0.08	0.16	0.32

The purpose of the Plane Cut samples is to see if certain low-lying modes, meaning modes that occur at the low frequency (<240 kHz) end of the resonant spectrum, can be tracked as a function of sample shape. In other words, as the plane cut goes deeper into the cylinder, can the change in frequency of particular modes be predicted and verified? This will show not only if the RUS method can handle this small step away from simple sample symmetry, but also if details of the mode spectrum can be discerned when samples are complex but similar to each other. The modes to be studied include the first and second bending, torsional, and barrel modes. These modes were chosen because, being low frequency modes and the lowest overtones of their respective types, they are easily detectable with the equipment used for the experimental verification.

The second class of complex shape will be a right circular cylinder with a plane cut into it as above, only with the addition of a notch cut out of the side to destroy all symmetry planes. This sample, having no symmetry plane and exhibiting sharp internal edges and corners, will be considered a piece closer to general mechanical engineering interest than any that is known to have been investigated by the RUS technique before. This sample, which will be called the ‘Notch Cut’ class, is used to test the idea that the

RUS method can apply to samples with no symmetry and very complex geometric features. This sample was prepared the same way as Plane Cut 3, with the same depth of cut, initial diameter and initial length. The notch was added later and is made so that there is no symmetry plane in the sample. The depth of cut for the notch is the same as the depth of cut for the plane, .32 cm. An illustration of this sample is found in Figure 3.2.

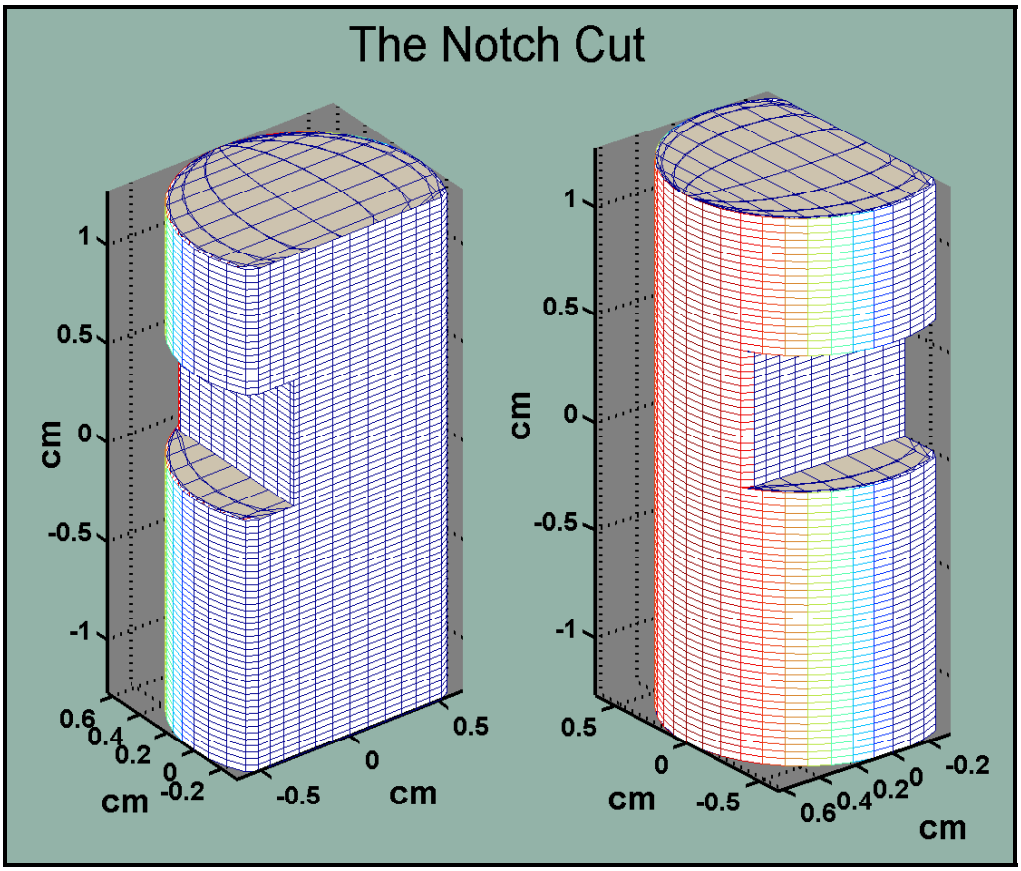

Figure 3.2 The Notch Cut. This sample has all of the same geometric dimensions as Plane Cut 3 listed above, with the addition of a square notch cut out at 90° to the plane face. The depth of the notch cut is equal to the depth of the plane cut.

The purpose of the notch cut is to test just how well the RUS method works when the sample geometry is significantly more complex. The presence of the notch represents

a large departure from the simple geometries for which RUS has been tested, and as stated above, this sample is more representative of a component an engineer might have to deal with than a cube, a cylinder or a sphere. Thus the applicability of RUS to real world components can be explored with this piece.

EXPERIMENT

Equipment

The experimental component of this project consisted of using piezoelectric transducers to excite and detect the resonant frequencies of the sample objects. To accomplish this the samples were held in between the transducers while a network analyzer was used to excite one of the transducers through a range of frequencies (a process called sweeping) while the other read the response of the sample to the excitation. The frequency range was chosen to be broad at first, covering about 10 kHz at a time, until the resonant peaks were identified. Once a peak was identified, the frequency range of the sweep was narrowed to 400 Hz. The 400 Hz range was used because the network analyzer used is limited to plotting 400 points, thus one point per Hz was plotted, which also is the effective resolution of the signal generator.

The standard transducer used for RUS is a specially constructed, .1 mm thick $LiNbO_3$ crystal made by Dynamic Resonance Systems (DRS) for this work [17]. These transducers weren't applicable for this experiment, thus a DRS horizontal mount longitudinal generator type was used. The reason that the first DRS transducers are the standard is that they are made to work well with the tiny (<5mm) RPPD samples that are normally used for RUS investigations. They are made so that the lowest resonance of the transducer is higher than the highest measured resonance of the sample. This prevents the resonance of the transducer from being mistaken for the resonance of the sample. The transducers used for the current project are made for RUS with larger objects (>1cm)

and as such there were no noticeable interactions between the resonant frequencies of the two objects, consequently, the interaction between the transducers and the samples was considered negligible. The experimental setup is pictured below.

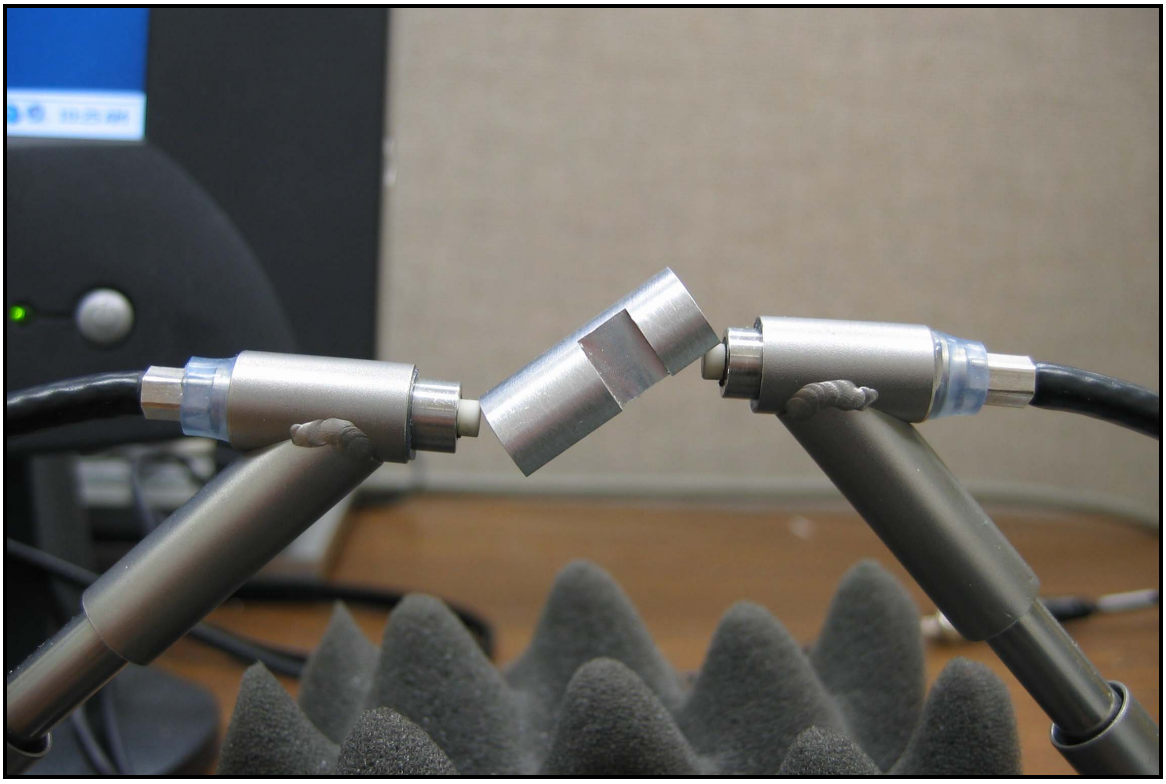

Figure 4.1 Experimental Setup. This image shows the Notch Cut sample held between the two DRS transducers.

Two factors that did turn out to influence the resonant frequency measurement in an appreciable way were the force between the transducer and the sample and the contact point of the transducer on the surface of the sample. Various investigators have found that a force of 3/100 Newton can shift a measured frequency response up to 1 kHz [18]. Great care was taken to avoid putting pressure on the sample during measurement in order to avoid this sort of effect. This was avoided by mounting the transducers horizontally so that there was a small gap between the pair. The sample was then

balanced between the transducers so that the only pressure on the surface of the transducers was due to the weight of the sample. This minimized the amount of contact pressure between the transducers and the samples without introducing a support which would also alter the resonance of the system.

The location of the contact point between the transducer and the sample also had an effect on the measured resonant frequency. It was found that by moving the contact point from one surface to another, the measured frequency could be altered by up to 300 Hz. In order to deal with this problem, the transducer was simply placed as close to a predicted anti-node for the frequency under consideration as possible, thus allowing for the largest signal, which was taken for the correct frequency.

The next factor that influenced the frequency measurements was that of temperature. When a sample was initially positioned between the transducers, it was noticed that the measured frequencies gradually rose as the object cooled down from the temperature it had reached while being handled. The cooling time was generally between 10 and 15 minutes. This was the easiest effect to deal with, although it created considerable delay when taking measurements.

Geometric imperfections of the samples also had an effect on the measured frequencies. All samples were made to the average standards of an average machine shop; the specified tolerances were ±.001 in. for all dimensions. As previously mentioned the samples used with RUS by materials scientists usually are more highly accurate than this, but for the present purpose, the normal tolerances were deemed acceptable. Two samples of each shape were constructed in an attempt to minimize this

problem. The maximum difference observed between the frequencies of two seemingly identical samples was around 300 Hz for low-lying modes. Thus all reported frequencies are an average of the frequencies measured for the two copies of each sample.

Procedure

As mentioned above, the first step of the project was to determine the elastic constants C_{11} and C_{44} using the RUS iteration method for a right circular cylinder and then use these constants to see if the resonant frequency spectrum for more complex geometry could be predicted. To this end a computer program was written in Matlab that could calculate these frequencies. This program was used to predict all frequencies under 240 kHz for the sample, 20 resonant frequencies and their mode shapes were predicted this way. The constants used in the initial program were those for generic 6061-T6 aluminum [19]. Once a set of frequencies was predicted, the resonant frequencies of the sample were measured. The constants used in the calculation were then varied until the RMS of the relative error for the frequencies under 240 kHz was minimized.

The method used to find the minimum is similar no matter how many independent elastic constants, or other parameters such as geometry, the material has. First one parameter is varied by small amounts until a minimum RMS error is found, then another constant is varied to see if the RMS error can be minimized further. This process continues, sometimes with backtracking in the case of many independent parameters, until the parameter space for the problem has essentially been ‘mapped’. This corresponds with the typical procedure for the use of RUS [20].

Using this procedure, the elastic constants corresponding to the minimum RMS error were found for the present case. These constants were then used to calculate the resonant frequency spectrum for all of the remaining samples. The computer program used for the second set of calculations was written in Fortran, as the computational speed of the Fortran routine was found to be an average of 30 times faster than the Matlab code. No iteration was done for the remaining samples, as they were all machined from the same piece of 6061-T6 bar stock.

Where performed, matching of individual measured modes with individual predicted modes was accomplished mainly by frequency mapping. That is, a one-to-one correspondence between these sets was assumed, and the order of the modes was assumed to be the same. In addition to this rather crude method, a technique was developed so that a little more confidence could be had in the frequency mapping. This technique involved placing the transducers on the sample surface as close to an expected anti-node as possible and looking for the expected increase in signal strength. When the signal strength increased near a predicted anti-node, the mode was considered to be identified. This procedure is examined more closely in the results section.

RESULTS

Initial Fitting

The fitting procedure for the cylinder required 7 iterations. That is, the elastic constants were varied 7 times in order to find the smallest RMS error. The procedure for this variation was as follows. First, initial values for Young's modulus (Y), Poisson's ratio (ν) and the Shear modulus (G) were obtained. The values for C_{11} and C_{44} can then be obtained through the relations

$$C_{11} = \frac{Y(1-\nu)}{(1+\nu)(1-2\nu)} \tag{5.1}$$

and

$$C_{44} = G \tag{5.2}$$

These values were used in the computer program with the N in (1.26) set to 12. The calculation was then repeated with N set to 14, and no significant changes in predicted frequency for the modes less than 240 kHz were observed. Next C_{11} was varied until the smallest RMS error was found. During this process of searching for the minimum RMS, Poisson's ratio was kept at a constant value of .33. Because of the relationship between Y and G, given by

$$Y = 2G(1+\nu) \tag{5.3}$$

C_{11} and C_{44} are related by

$$C_{44} = \frac{C_{11}(1-2\nu)}{2(1-\nu)} \tag{5.4}$$

so varying C_{11} but keeping ν constant also means varying C_{44}. This was done because, though different sources list different values for Y and G of 6061-T6 aluminum, all of the sources consulted list Poisson's ratio as this same value. Once the best-fit constants were obtained, the value used for the density was allowed to vary slightly. This was done to ensure that the best possible fit was obtained. The density was calculated from a mass measurement and measurements of the sample geometry. Variation of the density did not improve the RMS error from the fitting procedure. The iteration steps for the fitting process , including the RMS error for each step, and the guess values for C_{11} and C_{44} are listed below in Table 5.1.

Table 5.1 Iteration Summary. This table summarizes the process of finding the best elastic constants for fitting the data in a least-squares sense. The maximum single frequency relative error for the first 20 frequencies is .76%.

	C_{11} (GPa)	C_{44} (GPa)	RMS(%)
Iteration 1 (initial data)	103.22	26.19	.48
Iteration 2	104.0	26.39	.64
Iteration 3	102.0	25.88	.72
Iteration 4	103.6	26.29	.52
Iteration 5	103.1	26.16	.47
Iteration 6	103.05	26.15	.48
Iteration 7	103.15	26.17	.48

As revealed in the table, the generic data for this kind of aluminum provided a good initial guess value for the needed elastic constants. The elastic constants found by this process were then used to predict the frequency spectrum for the other sample pieces for frequencies under 240 kHz. The number of frequencies predicted here is higher because many of the modes for the cylinder are degenerate, meaning that they exist in pairs that share the same frequency. Thus once the geometric symmetry is destroyed, the degenerate modes become distinct and individually observable. The results for the

simple cylinder are summarized in Figure 5.1.

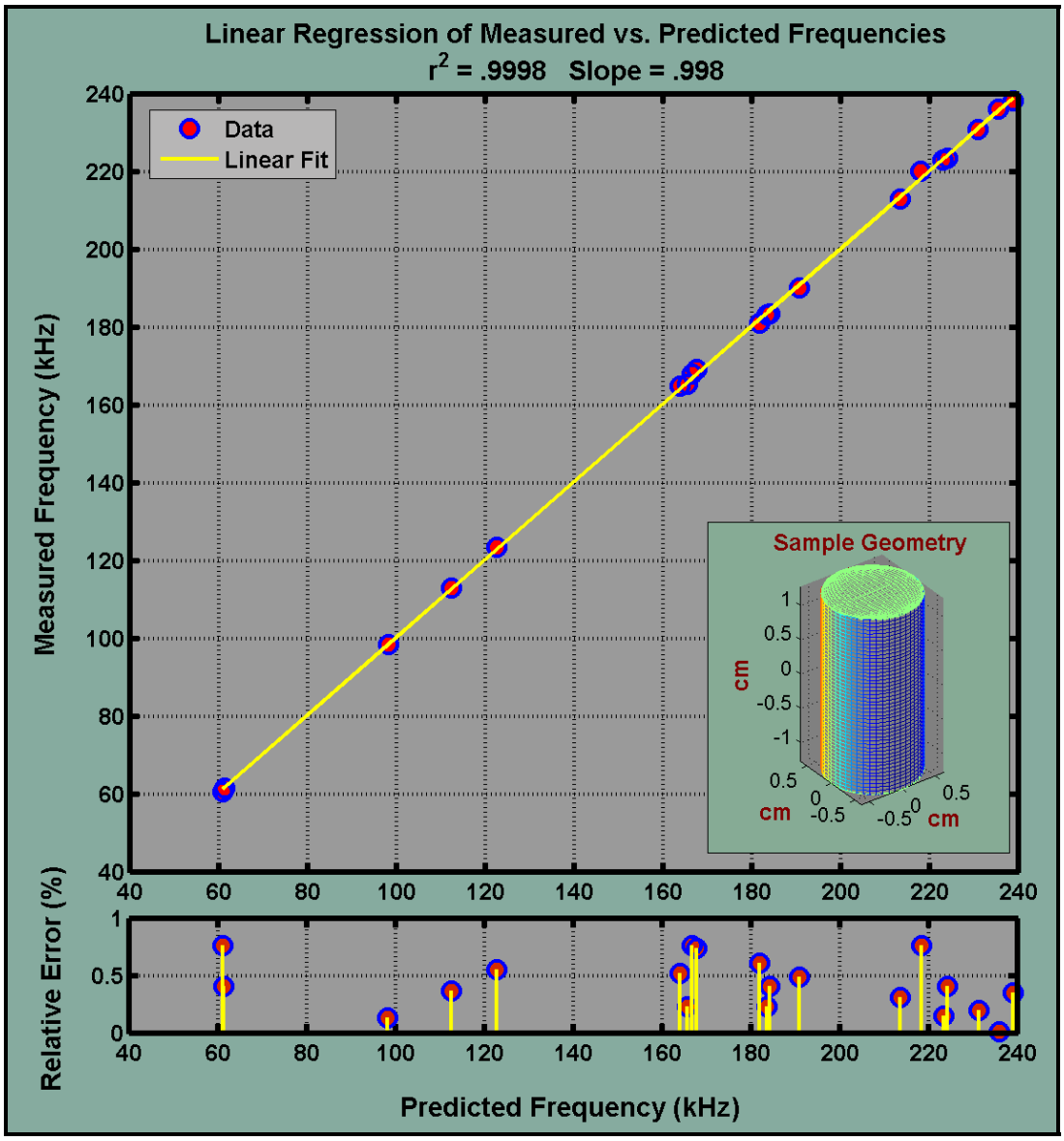

Figure 5.1 Linear Regression of Measured vs. Predicted Frequencies: Cylinder. Here the measured frequencies were plotted against those predicted by the model. If the fit were perfect, all points would be on a line of slope 1. The actual line of the regression has a slope of .998. The largest deviation from the line is less than 1% relative error. The correlation coefficient of .9998 suggests a high level of agreement for the fit. For this model, the RMS error is .47%.

Figure 5.1 shows the process of comparing the predicted frequency spectrum to the measured frequency spectrum. The common practice is to look for the minimum

RMS relative error for at least the first 5n frequencies, where n is the number of independent elastic constants of the material under consideration. Thus a linear regression was performed and the point-wise relative error to this linear regression was also calculated. This was also done for the other samples, as is shown below.

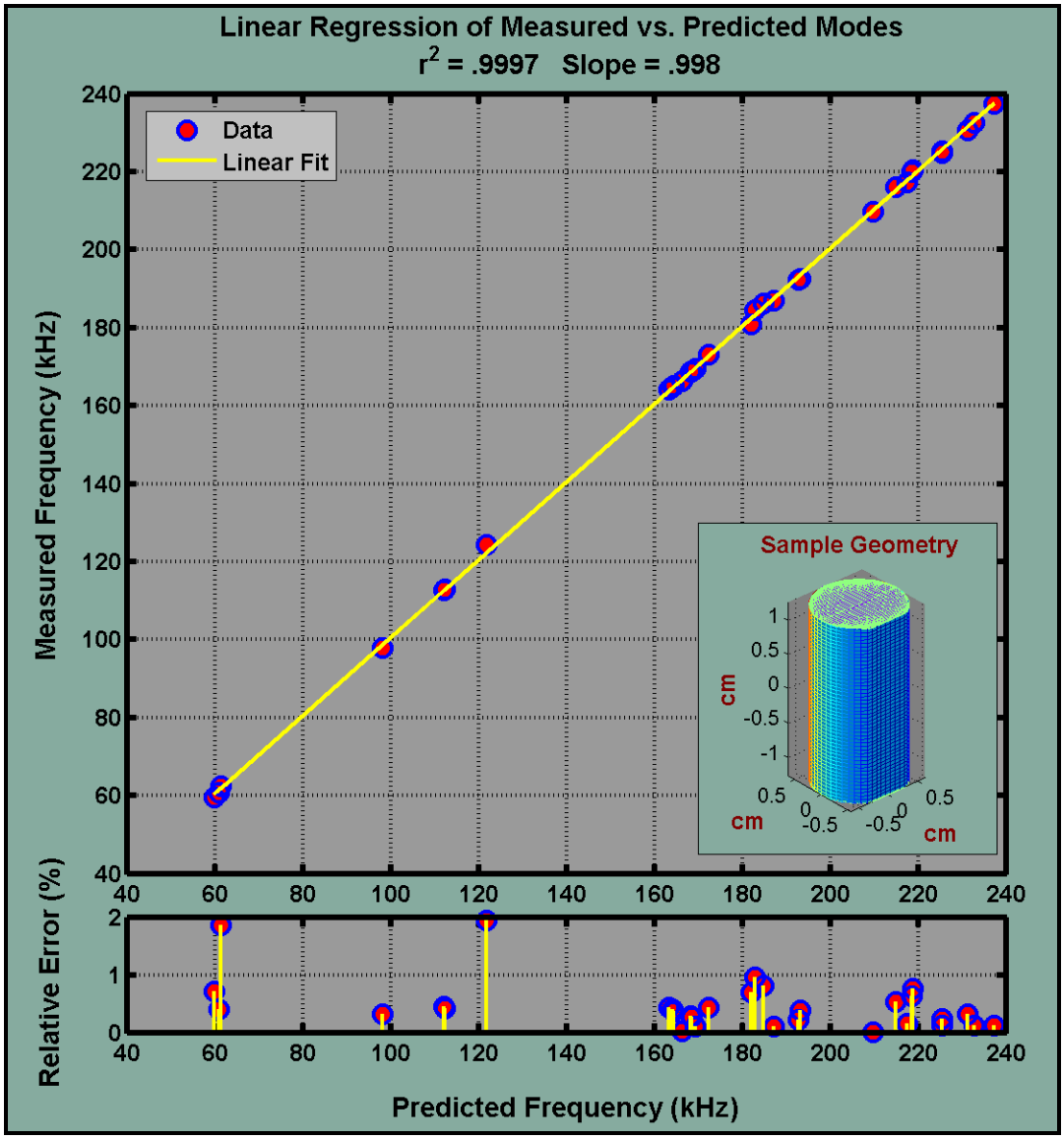

Figure 5.2 Linear Regression of Measured vs. Predicted Frequencies: Plane Cut 1. Here the measured frequencies were plotted against those predicted by the model. The line has a slope of .998, the maximum relative error is 2.0 % and the RMS value is .62%. Notice how many more modes are in this diagram than the one above.

The results for comparing the calculated resonant frequency spectrum to that measured for Plane Cut 2 are shown in Figure 5.3

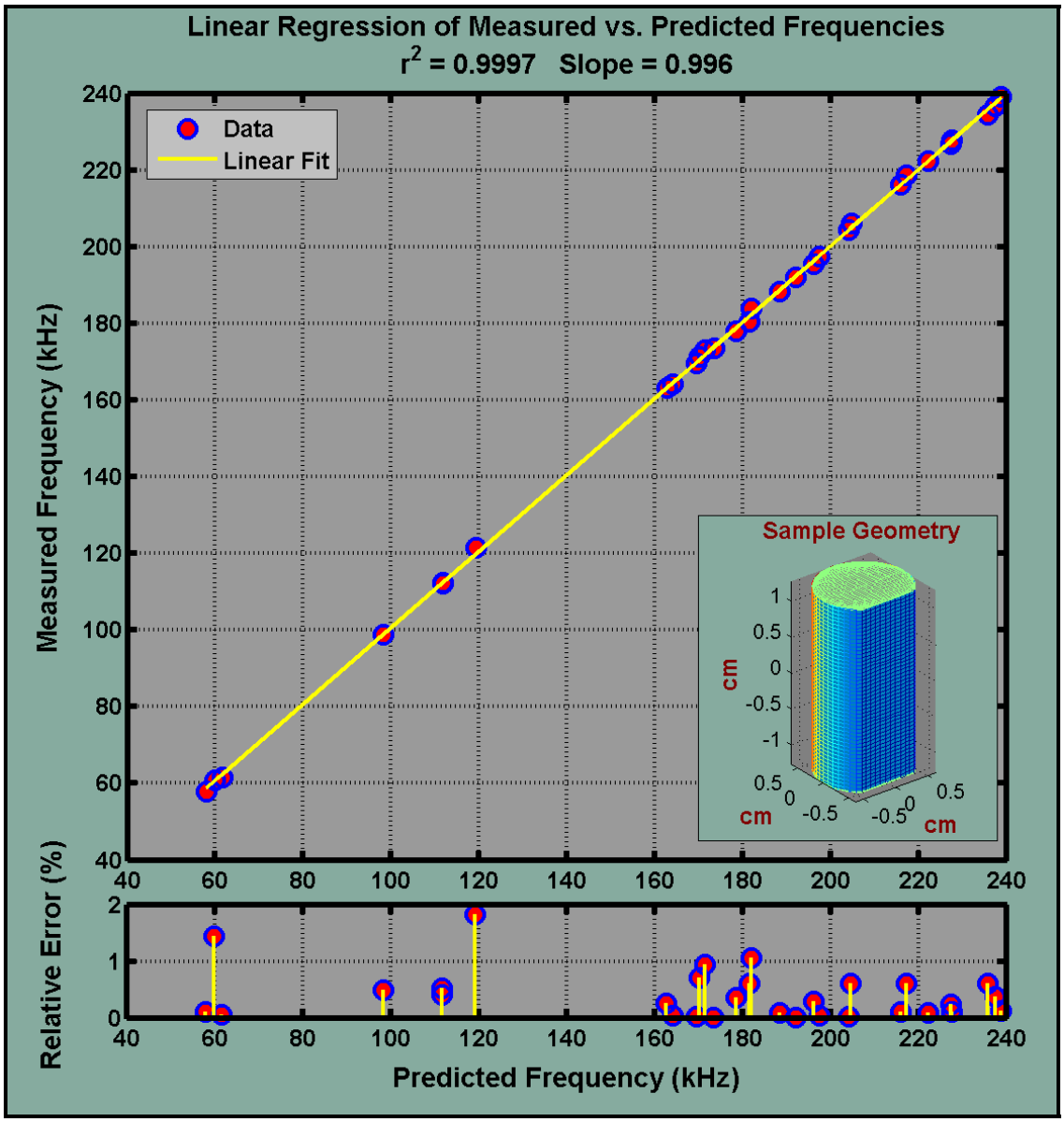

Figure 5.3 Linear Regression of Measured vs. Predicted Frequencies: Plane Cut 2. Here the measured frequencies were plotted against those predicted by the model. The line has a slope of .996, the maximum relative error is 1.8 % and the RMS value is .59%. This RMS error also isn't as low as that for the cylinder, but the correlation coefficient is still very close to 1, which indicates a good fit. This sample also has many more modes under 240 kHz than the cylindrical sample did. This fit is even better than the previous fit, although the reason for this is not known.

The results for comparing the calculated resonant frequency spectrum to that measured for Plane Cut 3 are shown in Figure 5.4.

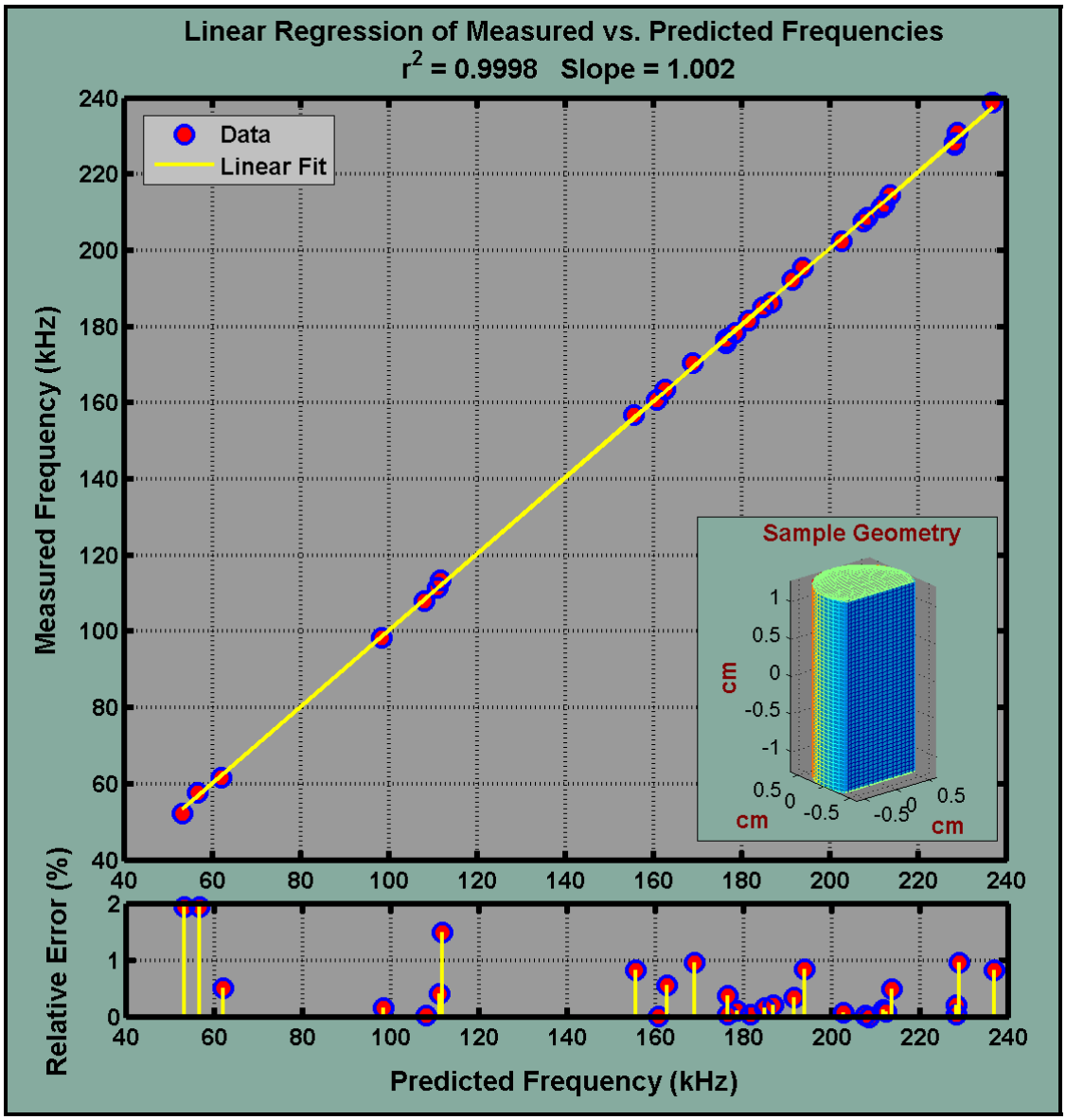

Figure 5.4 Linear Regression of Measured vs. Predicted Frequencies: Plane Cut 3. Here the measured frequencies were plotted against those predicted by the model. The line has a slope of 1.002, the maximum relative error is 1.9 % and the RMS value is .61%. This RMS error also isn't as low as that for the cylinder, but the correlation coefficient is still very close to 1, which indicates a good fit. This sample, too, has many more modes under 240 kHz than the cylindrical sample did. This fit is not as good as the fit for Plane Cut 2 but is better than that for Plane Cut 1.

The results for comparing the calculated resonant frequency spectrum to that measured for the Notch Cut are shown in Figure 5.4

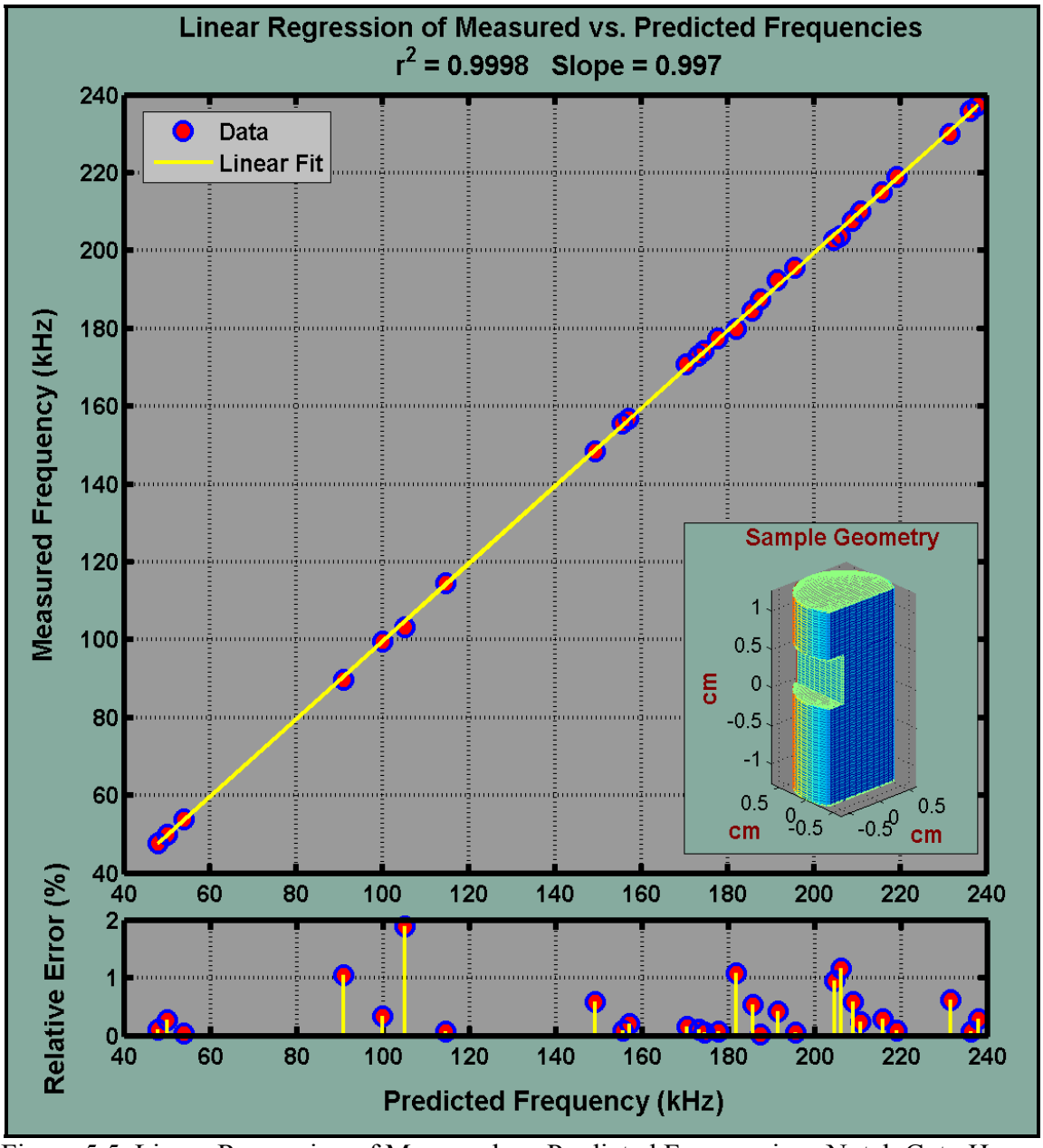

Figure 5.5 Linear Regression of Measured vs. Predicted Frequencies: Notch Cut. Here the measured frequencies were plotted against those predicted by the model. The line has a slope of .997, the maximum relative error is 1.9 % and the RMS value is .59%. This sample, too, has many more modes under 240 kHz than the cylindrical sample did. This fit is even better than the fit for the Plane Cuts.

Mode Tracking

Once the initial fitting procedure was completed, the next step was to see how well the model did at tracking individual modes. The modes that were tracked as a function of sample shape include the first and second bending, torsional and barrel modes. These modes were chosen because they are on the lower end of the frequency spectrum and they are the lowest overtones of their respective types. These factors make these modes easily detectable with the equipment available for the experiment.

In order to figure out how well the change in frequency of these modes was actually predicted, a way was devised to verify which mode was which in the case where the frequencies are close together. This is to avoid mistaking one mode for another. For example, the lowest bending mode for the cylinder is a degenerate mode, which means that there are actually two modes that share the same frequency. This is possible because of the geometric symmetry of the sample. This is illustrated below in Figure 5.6.

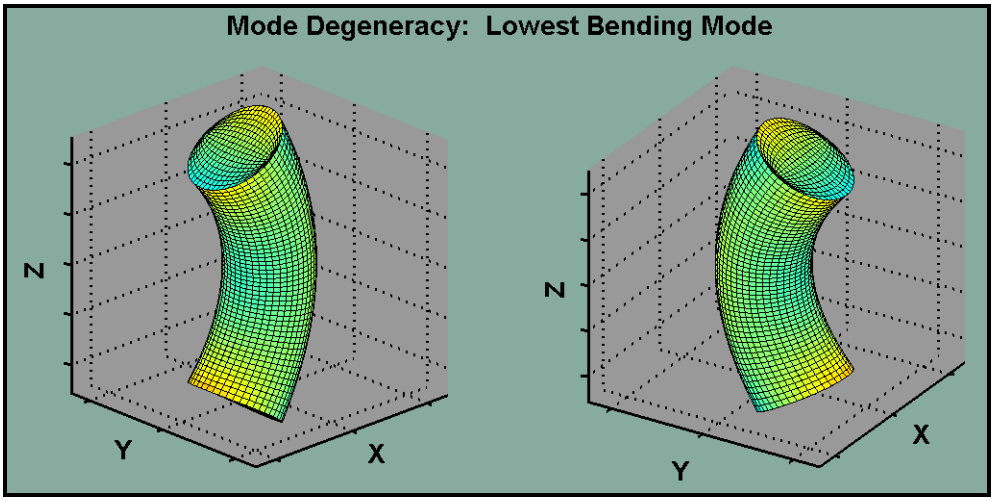

Figure 5.6 Bending Mode Degeneracy. Because of symmetry, there is no physical difference between the positive x and y directions. If the symmetry were destroyed, these modes would exist at different frequencies.

The method used to distinguish between modes was to try and place the transducer on an anti-node of the mode to generate the strongest possible signal of that mode, while generating a low strength signal for the other mode, then repeating for the other mode so that there is some certainty that the correct mode is being followed. This method is rather crude, but seemed to give good results. The first bending mode was tracked as a function of sample shape, this is plotted below.

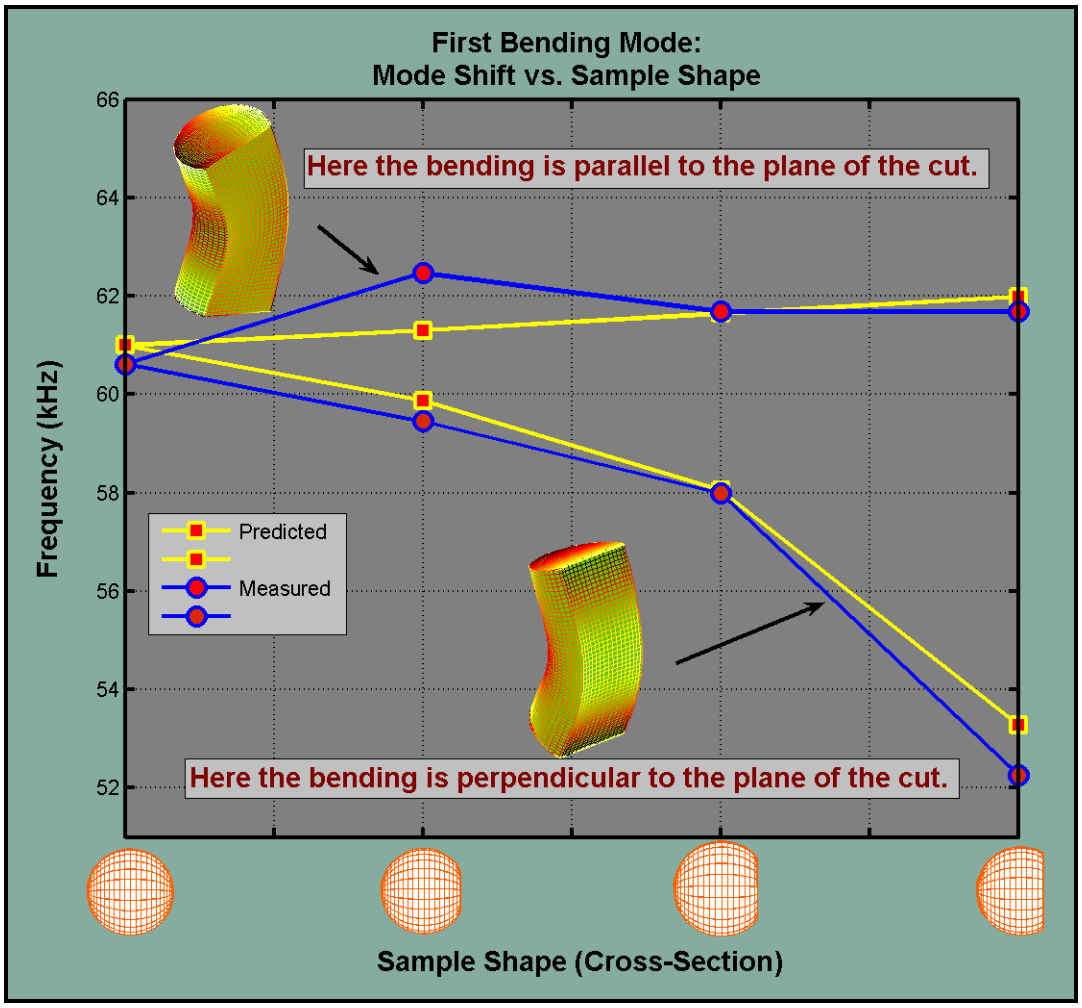

Figure 5.7 First Bending Mode: Mode Shift vs. Sample Shape. The lowest bending mode was tracked as a function of sample shape. Notice that the left hand side corresponds to the cylinder. This mode was degenerate with one mode increasing in frequency and the other decreasing in frequency as the plane cut depth grew deeper.

As Figure 5.7 shows, both the predicted and measured frequencies for the bending mode show the same trend. Once this degenerate mode splits, one mode went up slightly in frequency and the other went down in frequency as a function of sample shape. Thus it appears that for the mode where the bending is parallel to the plane of the cut, the flat plane acts as a modest stiffener, increasing the frequency of vibration. To distinguish these modes from one another, the transducers where placed at the anti-nodes on the endpoints for both modes in a separate setup [1]. The output signals for this are shown below in Figures 5.8 and 5.9.

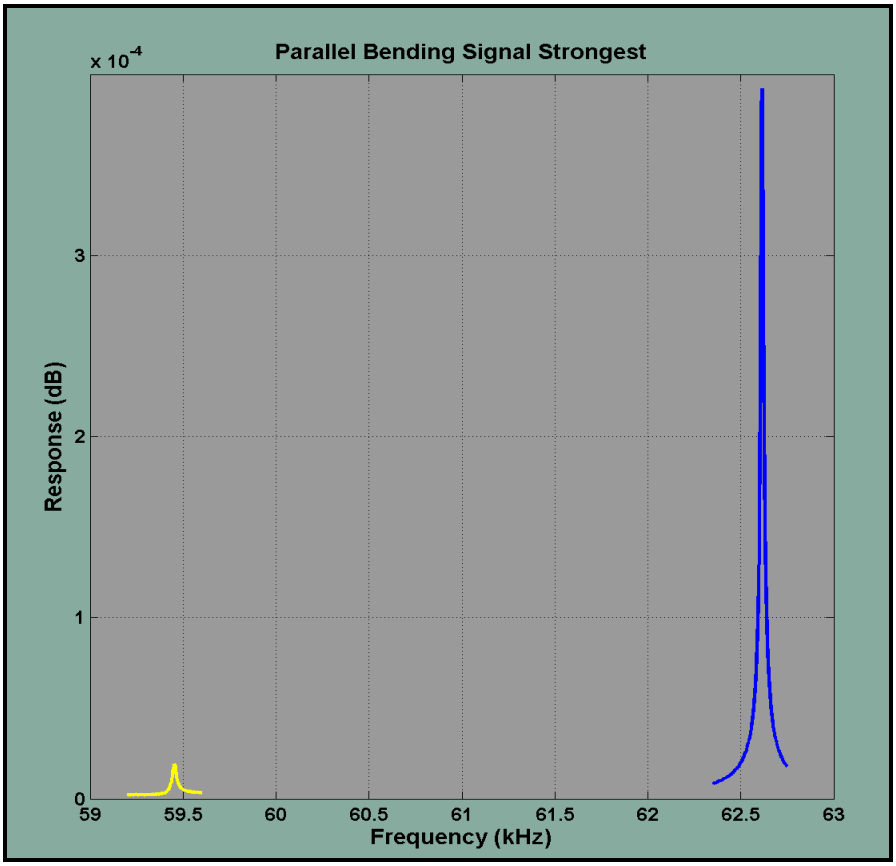

Figure 5.8 Parallel Bending Signal Strongest. Here the transducers were placed at the anti-nodes that corresponds to the bending parallel to the plane for Plane Cut 1. Notice how much stronger the signal is for the anti-node of the parallel bending mode than the signal from what is essentially a node for the perpendicular bending mode.

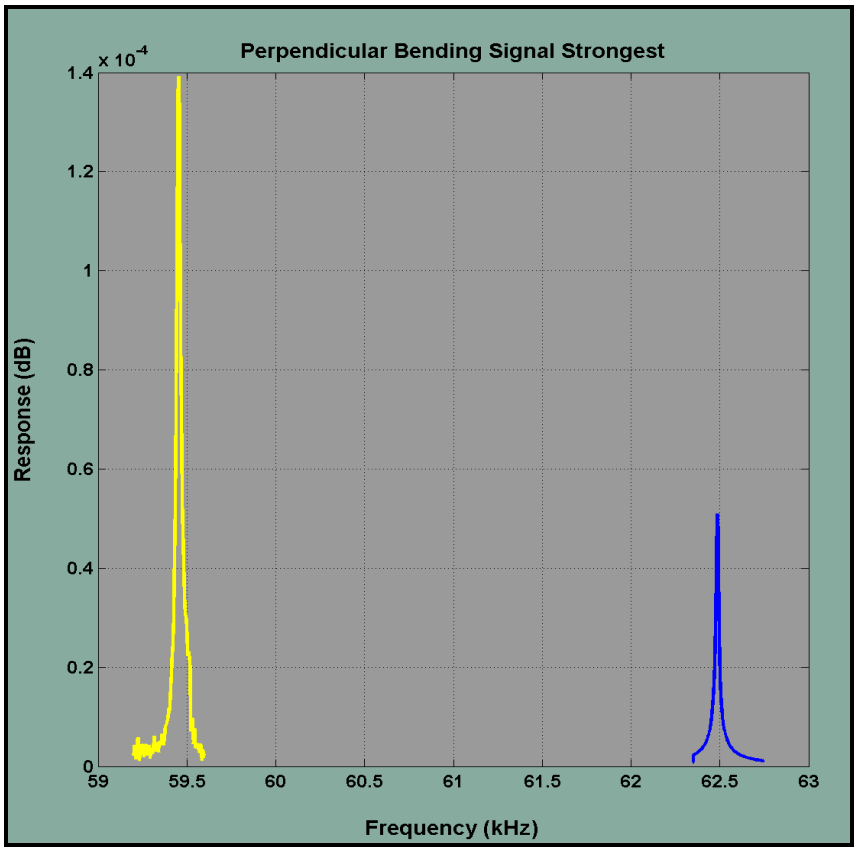

Figure 5.9 Perpendicular Bending Signal Strongest. Here the transducers were placed at the anti-nodes of the lowest bending mode corresponding to the bending perpendicular to the plane for Plane Cut 1. Notice how much stronger the signal is for the anti-node here than the signal from what is essentially a node for the parallel bending mode.

This type of comparison wasn't always possible and the accuracy of this approach is not quantified, but it was made for the second bending modes just as above. The first torsional mode was also tracked as a function of sample shape. This mode, being torsional along the axis of the cylinder, has no degeneracy. More specifically, this mode is not degenerate because the cylinder twists back and forth with the angular momentum vector along the axis of the cylinder. Thus there is no dependence on the distinction between the x and y coordinate directions. Thus even when the plane cut is introduced, the symmetry of the z direction is preserved. The frequency plotted as a function of

sample shape is shown in Figure 5.10

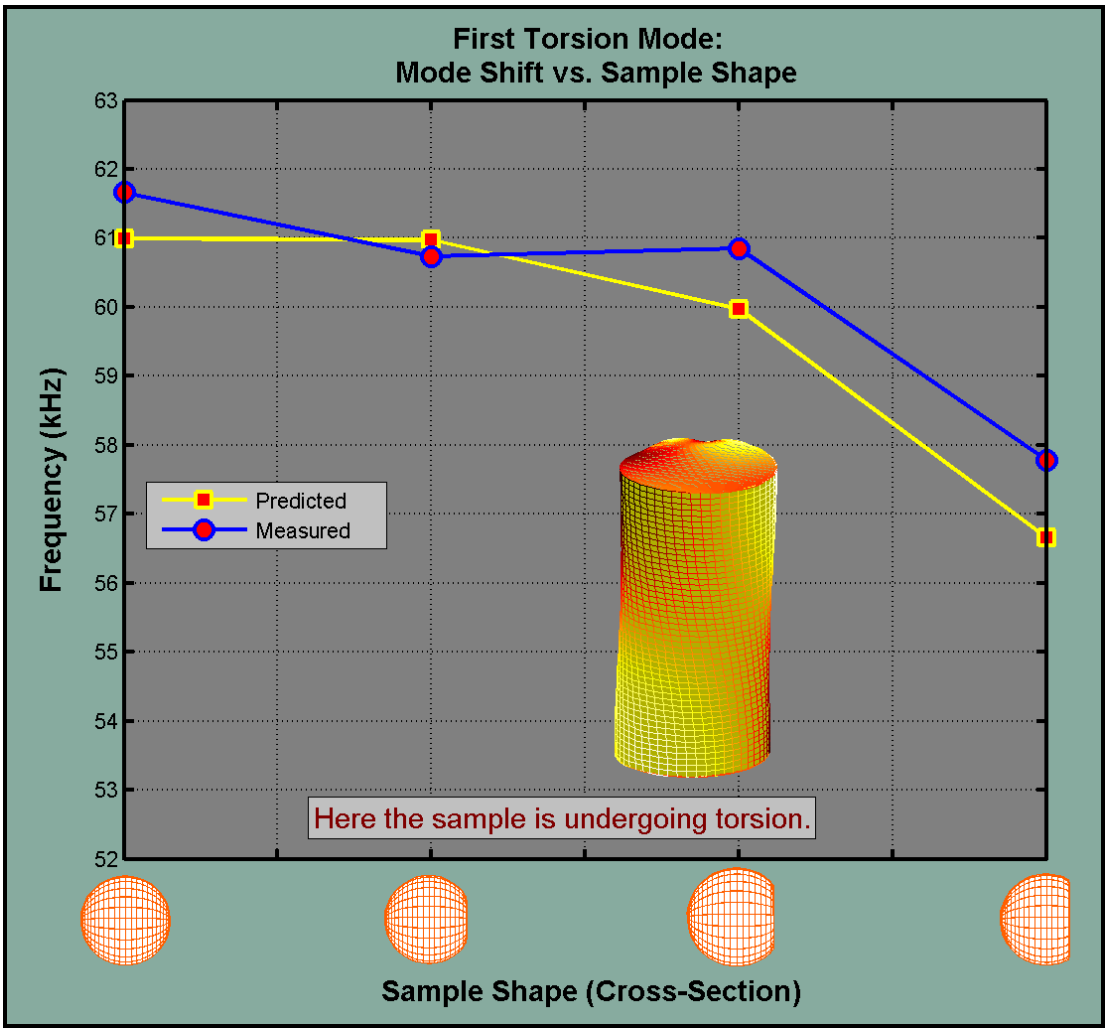

Figure 5.10 First Torsion Mode: Mode Shift vs. Sample Shape. The lowest torsional mode was tracked as a function of sample shape. Notice that the left hand side corresponds to the cylinder. The measured line exhibits the same trend as the predicted line, that the mode frequency would decrease as the plane cut grew deeper.

In Figure 5.10 it is apparent that both the measured and predicted modes decrease in frequency as the plane cut gets deeper. Also by comparing Figure 5.10 with 5.9 it is apparent that the path of the curves in Figure 5.10 lie in between the two degenerate modes in Figure 5.9. Notice that a similar phenomena is observed for the second torsional mode, shown below in Figure 5.13. The second torsional mode drops off a lot

faster than the first mode, though the reason why this happens is not known .

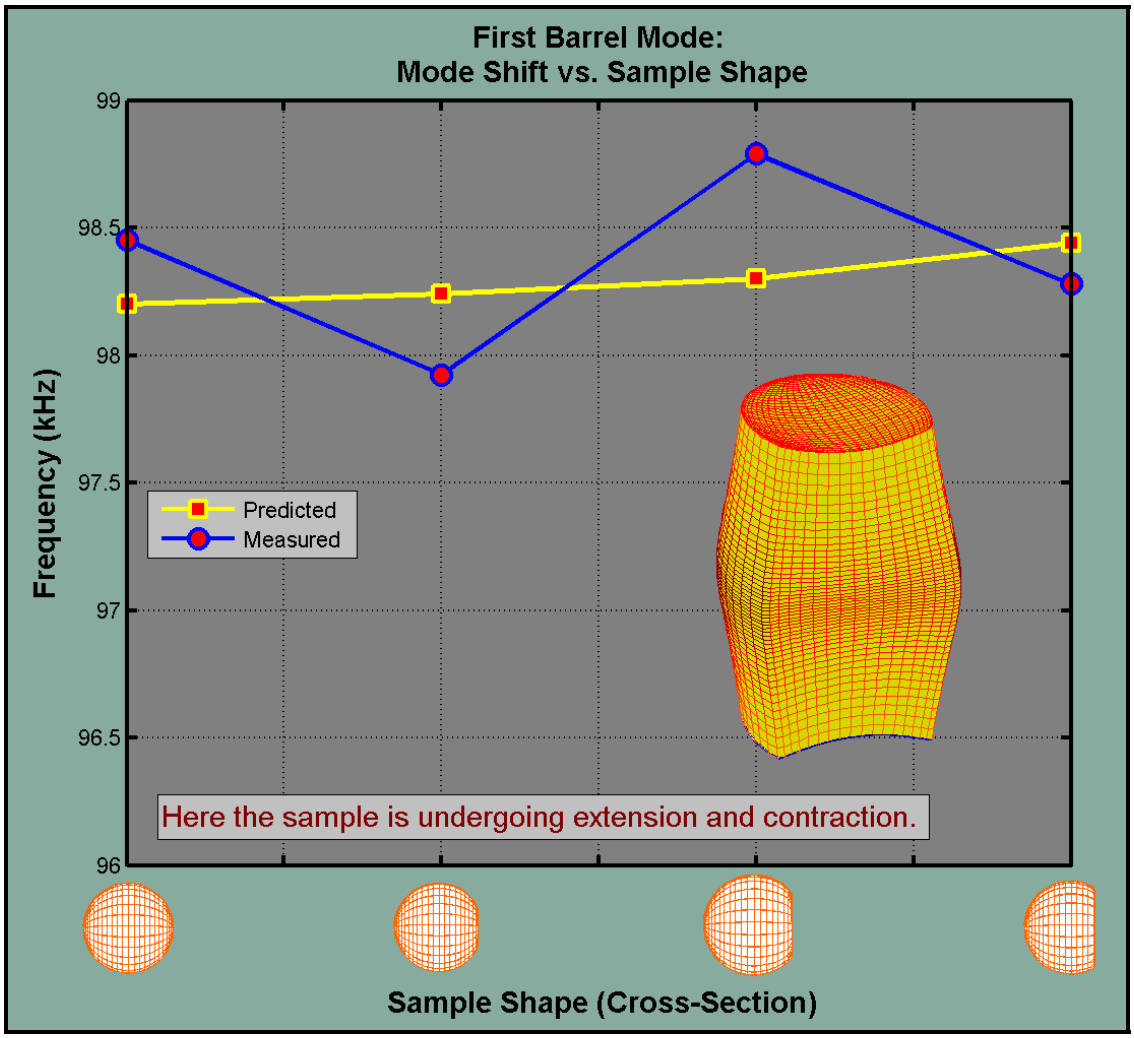

Figure 5.11 First Barrel Mode: Mode Shift vs. Sample Shape. The lowest Barrel mode was tracked as a function of sample shape. Notice that the left hand side corresponds to the cylinder. The measured line exhibits the same trend as the predicted line, that the mode frequency would increase as the plane cut grew deeper.

In Figure 5.11 the first barrel mode is tracked as a function of sample shape. Again, here the measured and predicted modes both exhibited the same trend, although the measured mode seems to oscillate around the predicted mode as the samples change. Here, as in the bending mode where the bending is parallel to the plane cut, the presence of the plane cut seems to act as a stiffener, thus effectively increasing the strength of the

elastic restoring force as the mode goes through a cycle of vibration.

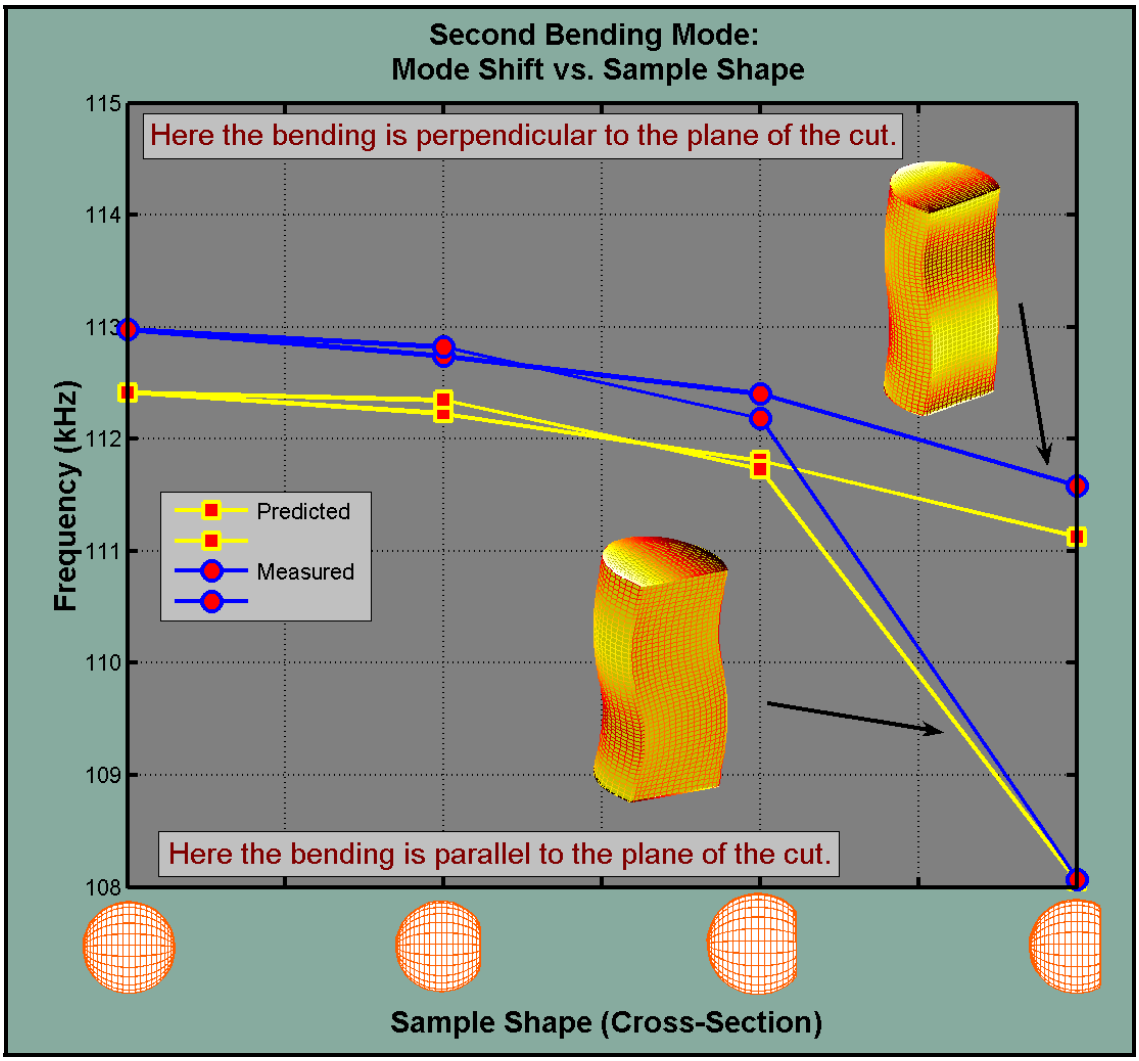

Figure 5.12 Second Bending Mode: Mode Shift vs. Sample Shape. The second bending mode was tracked as a function of sample shape. Notice that the left hand side corresponds to the cylinder. The measured lines exhibits the same trend as the predicted lines, both show a dramatic drop-off of frequency with the last sample shape.

In Figure 5.12 the second bending mode is tracked as a function of sample shape. Notice that this mode, like the first bending mode, is a degenerate mode. This is reflected in both the measured and the predicted plots. Here the presence of the plane cut doesn't seem to have the stiffening effect that it did for the lower bending mode. In fact this

bending direction drops in frequency faster than the other direction.

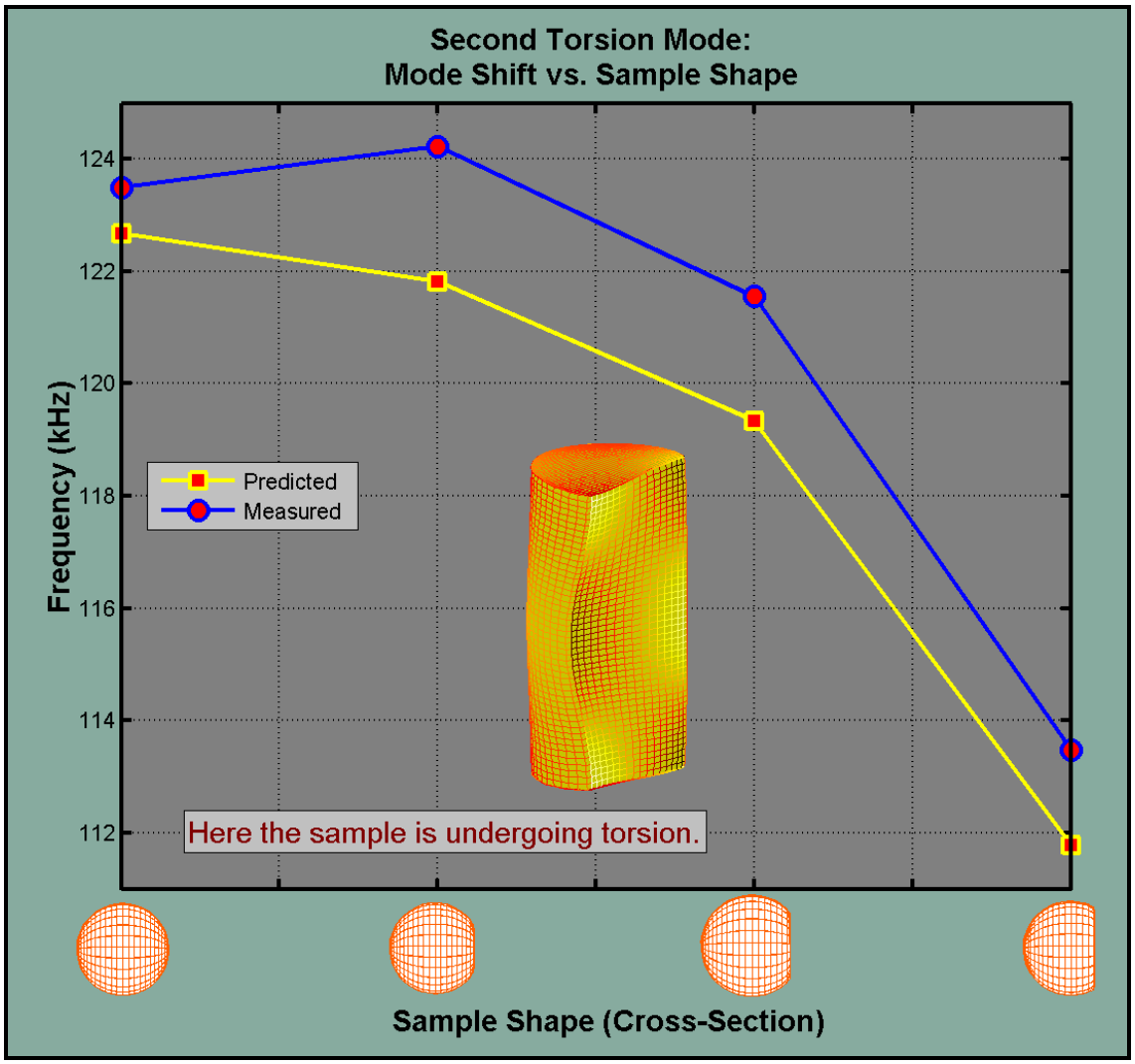

Figure 5.13 Second Torsion Mode: Mode Shift vs. Sample Shape. The second torsional mode was tracked as a function of sample shape. Notice that the left hand side corresponds to the cylinder. The measured lines exhibits the same trend as the predicted lines, although the measured line has a starting point that appears too low for the trend to be parallel to the predicted trend.

In Figure 5.13 the second torsional mode is tracked as a function of sample shape. This mode had one of the largest actual errors of any measured mode, especially for the Plane Cut 1, as can be seen in Figure 5.2. The reason for this discrepancy is unknown.

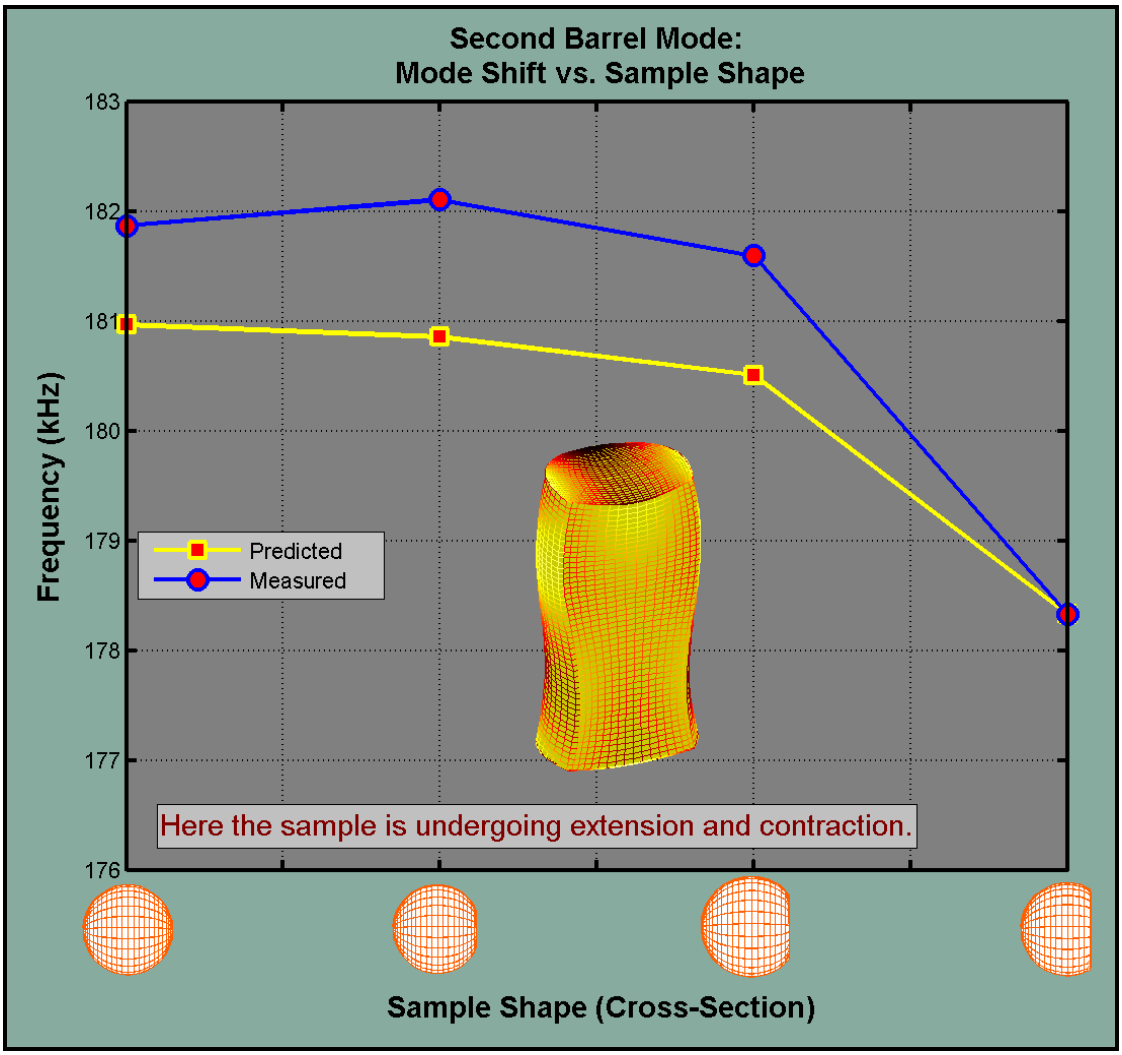

Figure 5.14 Second Barrel Mode: Mode Shift vs. Sample Shape. The second Barrel mode was tracked as a function of sample shape. Notice that the left hand side corresponds to the cylinder. The measured line exhibits the same trend as the predicted line.

In Figure 5.14, Plane Cut 1 sample seems to buck the trend, increasing when it was predicted to decrease, though the amount of increase is within the worst-case RMS error, calculated later. The general trend of the predicted line and the measured line is the same. For all of the tracked modes, the agreement in trend and the closeness in number for the frequencies indicates that the model can successfully predict mode shift.

ANALYSIS

Expected Uncertainty

Here the worst-case error is sought to provide an estimate of the upper bound for what the largest RMS error could be. Thus by comparing this with the measured RMS error, an idea of how good the fit is can be developed. Since the actual functional relationship between the individual frequencies and the parameters of the exercise is not known, a brute force method must be used [21]. This method is to simply take the square root of the sum of the squares of the variation of the frequencies with the variation of each parameter separately. For the cylinder, this is given by

$$\delta f = \sqrt{\left(\delta f_{mass}\right)^2 + \left(\delta f_{radius}\right)^2 + \left(\delta f_{length}\right)^2 + \left(\delta f_{measurement}\right)^2} \qquad (6.1)$$

where δf_{mass} is the relative change in frequency f predicted by the model when the mass is varied by the uncertainty in the mass measurement, and the other terms are similar. The $\delta f_{measurement}$ term is the observed randomness in the measurement of each frequency due to random variations in temperature, position, electrical fluctuations in the equipment and the like. These factors are the variables that make the experiments not exactly repeatable. The observed change in frequency from one measurement to another was greater for some frequencies than others, but the range is from 300 Hz for the low-lying frequencies to 900 Hz for the higher frequency modes. Once the variation of each frequency is found for each sample tested, the RMS value of the variation for the spectrum can be found and compared with the standard RMS found earlier. Note that for the other sample shapes

there will be at least one other term, that of the variation due to the other dimension(s). This calculation has been done and the results are summarized below in Table 6.1.

Table 6.1 Summary of Upper Bound RMS Errors. Here the upper bound for the RMS error of the frequency spectrum is shown. All of the measured frequency spectrums have an RMS error less than the worst-case RMS error.

	Cylinder	Plane Cut 1	Plane Cut 2	Plane Cut 3	Notch Cut
Worst-Case RMS (%)	.6	.71	.71	.72	.84
Actual RMS (%)	.47	.62	.59	.61	.59

Comparison with Published Results

The most common shape used for RUS experiments is the RPPD, thus few examples of the application of this technique to cylinders exists. However, one such investigator did use highly precise cylindrical steel samples in an RUS experiment [22]. The RMS error attained in this experiment was .3% for the lowest 9 modes. Another investigator used isotropic glass cylinders in a RUS experiment and achieved an RMS error of .5% for 5 chosen frequencies [5]. This investigator also used anisotropic cylinders of granite, which the RUS method should be able to handle. The RMS error for this experiment was 1.3% for 21 chosen frequencies. Thus the work done here corresponds well with this previous work, although the inclusion of many more frequencies may skew a direct comparison.

If one includes the work done with RPPDs and Spheres, the reported RMS error ranges from .1% to .4%. Thus the RMS error found here is within the expected range. The reason for the error here being near the upper end of the range of reported errors probably has to do with the relatively low level of perfection the prepared samples

exhibit. These samples weren't as highly precise as those usually used when investigating material properties as that was not the goal of this research. Here it was hoped that the tools of RUS could be applied to sample geometry that is more complex, not just in shape but in imperfections, and thus that RUS could be useful outside the strict confines of the laboratory. Another reason that the error reported here is higher than the usual error for these types of experiments, is that often when looking for the best fit between model and measurement, an investigator will freely choose to throw out one or two of the worst fit frequencies [1,23]. The offending modes will either be completely ignored or, conveniently, the weight of the modes will be sharply reduced in the final sum. Had this been done here, the errors would have been considerably less. This wasn't done, however, because of a lack of experience in determining when a mode qualifies to be thrown out or by how much to artificially reduce its impact on the final RMS fit.

Finally it must be noted that there were two modes that were predicted but not found. One of these was predicted for Plane Cut 2 and the other was for the Notch Cut. The problem of so called 'missed modes' is not unheard of in the literature, several examples of this for RPPDs and Spheres have been found[1,6]. Both of these modes were fairly isolated, meaning sufficiently far in frequency from neighboring modes that they probably weren't missing because of interference. After an extensive search, made by altering the sample position and repeated measurement, it was decided that the equipment was not able to excite these modes. Thus they were eliminated from the final RMS analysis as there is no way to calculate the discrepancy without a measurement.

Comparison of RUS to FEM

Recall that the fundamental mathematical approach taken here is the Rayleigh-Ritz (R.R.) variational method applied to the body as a whole. An alternative approach is to solve the problem with the finite element method (F.E.M.). The finite element method involves breaking the body into discrete elements and solving the governing physical equations for each element separately while imposing continuity across all element boundaries. The F.E.M. effectively uses a scheme such as the R.R. method or more commonly, the Galerkin residual method, in a piecewise manner. Thus the R.R. approach can be viewed as a one-element form of the F.E.M. The comparison between these two approaches here will include a brief literature survey and the results of an actual test using FEMLAB software.

Others have compared these two approaches while investigating RPPDs [2]. The results of the comparison indicate that both methods are able to yield accurate results. However, the main difference was that the F.E.M. was much slower. The investigator cited above found that the F.E.M. calculation for a particular shape (an RPPD) took 15 minutes, whereas the R.R. calculation took only a few seconds on the same machine. This clearly shows that the R.R. method has a cost advantage in terms of machine time, at least for simple geometry.

When the sample geometry is complex, the outcome is different. The eigenvalue problem was solved for Plane Cut 2 with the FEMLAB software, which is a generic F.E.M. application capable of solving many different physical problems. The first 30

resonant frequencies were found to the same degree of accuracy as that found with the R.R. method in much less time. Unfortunately, the FEMLAB software was not run on the same P.C. as the R.R. method, but the two computers were nominally similar in terms of processor speed and available memory. The frequencies that took 3 hours to obtain with the R.R. method took less than 15 minutes using the FEMLAB software. This represents a huge savings in computational time in the case of complex sample geometry.

It is believed that the speed differences mentioned above are rooted in the way these methods work. The finite element method uses quadratic or cubic elements and Gaussian (or other similar [26]) quadrature methods to evaluate the stiffness integrals, thus only a few sampling points are needed. The R.R. method used for this project, on the other hand, required 22 Gauss points to ensure convergence with the complex geometry. The fact that the routines written for the current project may not be as fully optimized as the code found in commercial packages could also have had an impact in terms of computational speed differences, though this is hard to quantify.

For simple sample geometry the situation is different. In this case the R.R. method was found to be faster. This is likely due to the fact that the integrals have analytic solutions, so that there is effectively only one “sampling point” needed for each integral, and the solutions are exact.

Thus the two methods yield the same results, and either method is a viable choice when finding resonant frequencies. However, the choice of which method to use will depend on the geometry of the sample under investigation. Clearly for simple geometry the R.R. method is preferred, while the opposite is true for more complex shapes.

CONCLUSIONS, RECOMMENDATIONS AND FUTURE WORK

The goal of this thesis was to explore the applicability of RUS methods to complex sample geometry. Toward this end, several samples were prepared with increasingly complex geometry. A computer algorithm was developed for use in predicting the frequency spectrum of several of these samples from the sample's geometry, density and elasticity matrix. After finding appropriate elastic constants with the standard RUS iteration method, the frequency spectrum of all other samples was made. After the predictions were made, measurements were taken to determine how well the model was able to predict the actual frequency spectrum. From the results of these experiments, several conclusions and recommendations can be made.

Conclusions

- The RMS error of the initial fitting closely correlates with published results. This was expected as the geometry of the cylindrical sample has been explored by other investigators using RUS methods. Another indication that the fitting was successful is that the elastic constants that correspond to the minimum RMS relative error are well aligned with standard values for aluminum, and the further away from these values the input to the computer code became, the worse the RMS relative error became.
- Several low-lying modes were tracked as a function of sample shape, with good results. Of course this is just more verification that the model was successful for

the individual sample shapes, but the ability to show a trend adds to the knowledge of how the resonant spectrum behaves for a particular class of sample shape.

- The RMS relative error for the most complex sample geometry, that of the Notch cut, also was low, matching the best of the Plane Cuts. This was surprising, as it was expected that as the geometric complexity increased, the error would grow correspondingly simply because there are more geometric parameters and their associated uncertainties. The size of the RMS error for even such a complicated geometry as the Notch Cut suggests that the RUS method is applicable for use with at least some geometries more complicated than the RPPD, cylinder and sphere.
- The Gaussian quadrature method of integration proved its utility here by outperforming other methods of integration. This form of integration allows (1.36) and (1.37) to be accurately evaluated for a wide variety of sample shapes. Thus the frequency spectrum of virtually any analytically defined sample geometry can be predicted with this method, though for really complex geometry, the number of Gauss points needed might have to be dramatically increased.

Recommendations

- Based on the results discussed above, a couple recommendations can be made. These experiments reveal the importance of highly precise sample geometry if the goal is highly precise elastic constants. Thus when investigating a new material,

the better approach is probably to rely on simple sample geometry in order to reduce errors from geometric imperfections. In addition, when sample geometry is more complex than that usually used for RUS, the finite element method can be used to obtain the same results in considerably less time.

- Throughout this thesis, the term 'complex geometry' was used. This term may have different meaning to different researchers. As such, it is necessary to define how complex the geometry of a sample can be before the limits of this technique are reached. Though the limits of sample complexity don't appear to have been reached with the particular samples used here, it is intuitive that the use of polynomials to represent the physical displacements puts a limit on the types of geometry admissible. The technique of reducing the problem to integrals of the form given in (1.36) and (1.37) should make this method applicable to any sample with an analytically definable surface. That is, as long as the integrals are definable, they can be evaluated in one way or another. Even with very high order polynomials, however, representing sharp corners exactly is out of reach [26]. This is because polynomials are smooth functions and thus cannot achieve the infinite curvature of a corner. These observations imply that the term 'complex geometry' be used cautiously, and that more experiments are needed to quantify exactly what this means. These experiments could explore aspects of geometry beyond those possessed by the non-symmetric shapes explored here. These could include things like sharp points in the geometry or very thin foil-like samples, etc.

Future Work

Future work in extending the RUS method may involve looking at the development of a quantitative description of changes in the microstructure of metals as a function of changes in observed resonant frequency. This idea is based on the fact that metals are made up of grains, and as such it may be possible to build a model of a piece of metal that is made up of individual crystals so that bulk resonant behavior can be determined by accounting for the boundary conditions between grains. This would require a complete reformulation of the problem, starting with the Lagrangian for the system as a conglomerate of smaller systems, including various boundary conditions. To the author's knowledge, this has not been done, but may prove an interesting course of research in extending this useful method.

LITERATURE CITED

[1] A. Migliori and J. L. Sarrao, Resonant Ultrasound Spectroscopy New York: Wiley, (1997).

[2] J. Plesek, Radek Kolman et al, *Journal of the Acoustical Society of America* **116** (1) 282-287 (2004)

[3] D. Litwiller, *Resonant Ultrasound Spectroscopy and the Elastic Properties of Several Selected Materials* Iowa State University, (2000)

[4] R. P. Hallion (ed.) Designers and Test Pilots *The Epic of Flight* Alexandria Virginia: Time-Life (1983)

[5] B. Zadler, J. H. L. Le Rousseau et al, *Geophysics Journal International* **156** 154-169 (2003)

[6] Francis Birch, *Journal of Geophysical Research* **80** (5) 756-764 (1975)

[7] R. Holland, *Journal of the Acoustical Society of America* **43** (5) (1967)

[8] H. H. Demarest, *Journal of the Acoustical Society of America* **3** (2) (1969)

[9] W. M. Visscher, A. Migliori et al, *Journal of the Acoustical Society of America* **90** (4) (1991)

[10] M. Marcus, Introduction to Linear Algebra New York: Macmillan, (1965)

[11] I. S. Gradshteyn, I. M. Ryzhik, Alan Jeffrey (ed.) Table of Integrals, Series and Products New York: Academic Press, (1994)

[12] J. Stewart, Calculus New York: Brooks/Cole Publishing Company (1999)

[13] J. Lund, K. L. Bowers, Sinc Methods for Quadrature and Differential Equations New York: Soc. for Industrial & Applied Math (1992)

[14] Nicodameous Alonso, PhD candidate in Mathematics, private communication.

[15] E. Wiesstein, *Mathworld* (2005) <http://mathworld.wolfram.com/FundamentalTheoremofGaussianQuadrature.html>

[16] A. Ezziranni, A. Guessab, *Mathematics of Computation* **68** (225) 217-248 (1999)

[17] Dynamic Resonance Systems, product brochure for M3 Modulus II system (2000)

[18] A. Yoneda, *Earth, Planets, and Space* **54** 763-770 (2002)

[19] Automation Creators (ed.), *Matweb* (2005) <http://www.matweb.com/search/SpecificMaterial.asp?bassnum=MA6061T6>

[20] R. G. Leisure, F. A. Willis, *Journal of Physics of Condensed Matter* **9** 6001-6029 (1997)

[21] J. Jacobs, *Errors and the Treatment of Data* University of Montana, 2001

[22] P. Heyliger, A. Jilani, *Journal of the Acoustical Society of America* **94** (3) (1993)

[23] A. Stekel, *Journal of the Acoustical Society of America* **92** (2) 1992

[24] B. A. Auld, Acoustic Fields and Waves in Solids 2nd ed. Malabar, Florida: Krieger, (1990).

[25] W. M. Lai, D. Rubin, E. Krempl, Introduction to Continuum Mechanics 3rd ed. Burlington, MA: Butterworth-Heinemann (1999)

[26] D. S. Burnett Finite Element Analysis Whippany, NJ: Addison Wesley (1988)

[27] S. C. Chapra, Applied Numerical Methods with Matlab 2nd ed. New York, NY: McGraw-Hill (2003)

[28] W. H. Press et al, Numerical Recipes in C 2nd ed. New York, NY: Cambridge University Press (2002)

APPENDICES

APPENDIX A

PLANE CUT 2
16 MODE SHAPES

The following pages contain illustrations of the mode shapes for Plane Cut 2 (Appendix A) and the Notch Cut (Appendix B). These illustrations were made by solving the eigenvalue problem (1.46) to find the $\{a\}$ values, which are the coefficients in the expansion of the displacement values as in (1.25). Note that the first 6 modes are not included in these appendices. This is because the first 6 modes correspond to the 6 translational and rotational modes that involve no local displacements. The mode shapes for Plane Cut 1 and 3 are similar to those for Plane Cut 2. These images were made with Matlab.

The color mapping in the illustrations also carries information. Recall that what is plotted here is the (highly exaggerated) surface normal displacement. As such, the color ranges from blue to red, with blue representing a displacement opposite to the surface normal (an indentation) and red representing a displacement in the direction of the surface normal (a bulge).

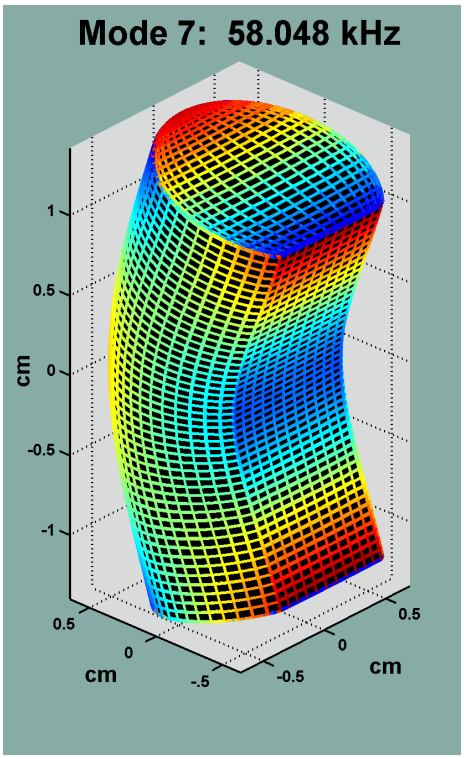

Figure A.1 Mode 7, Plane Cut 2

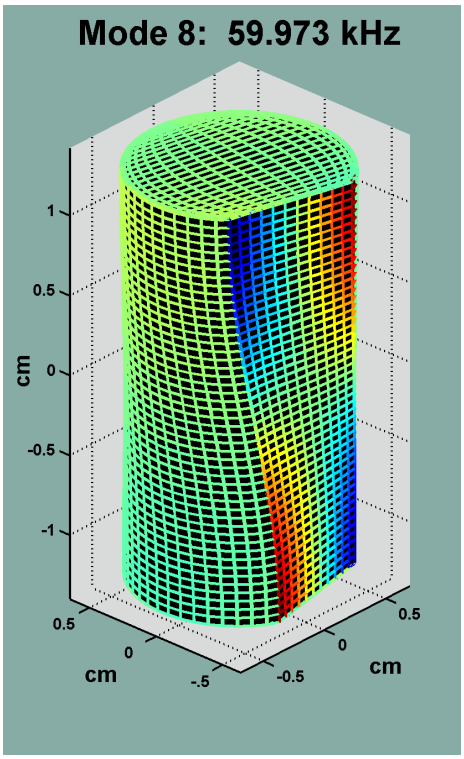

Figure A.2 Mode 8, Plane Cut 2

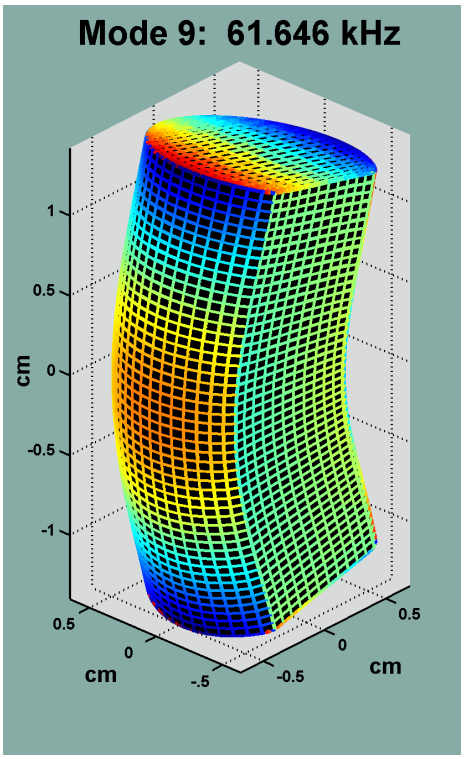

Figure A.3 Mode 9, Plane Cut 2

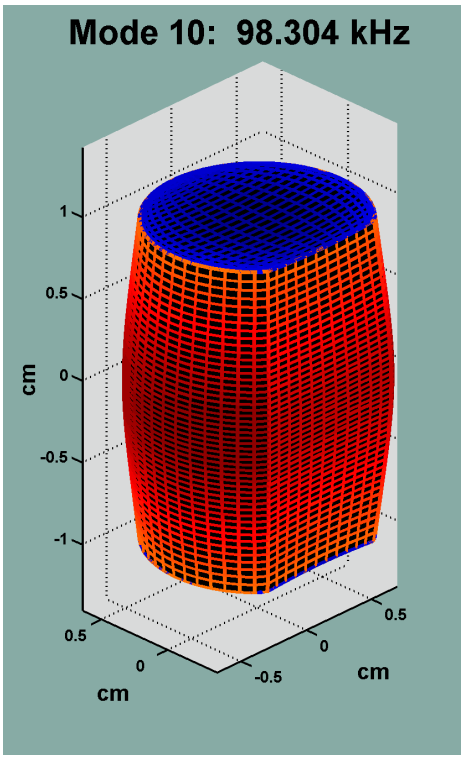

Figure A.4 Mode 10, Plane Cut 2

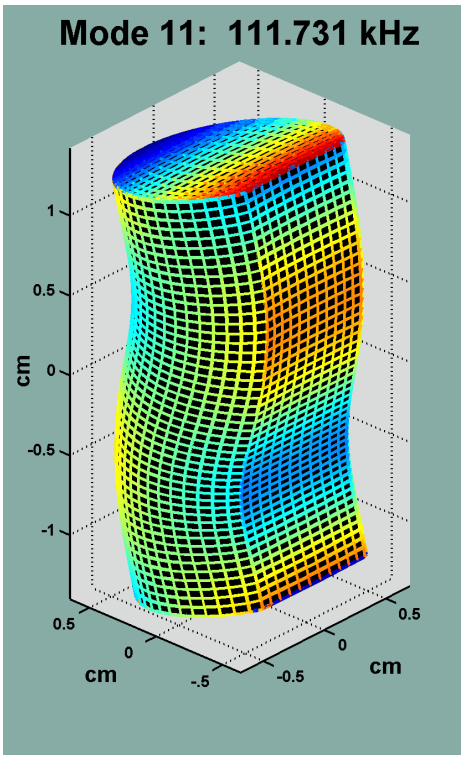

Figure A.5 Mode 11, Plane Cut 2

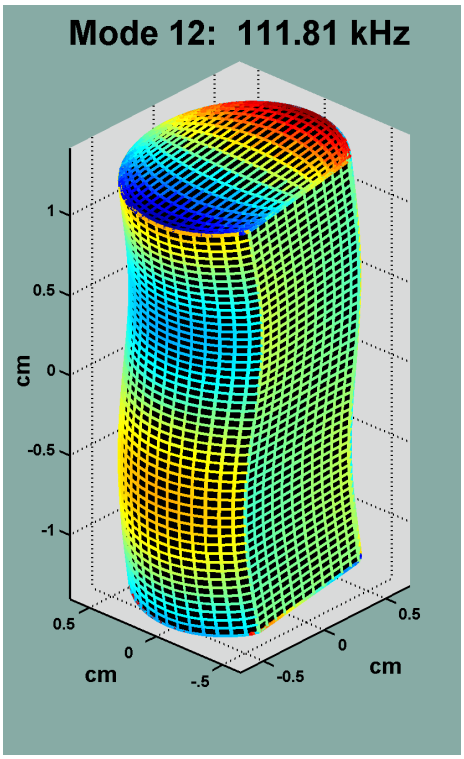

Figure A.6 Mode 12, Plane Cut 2

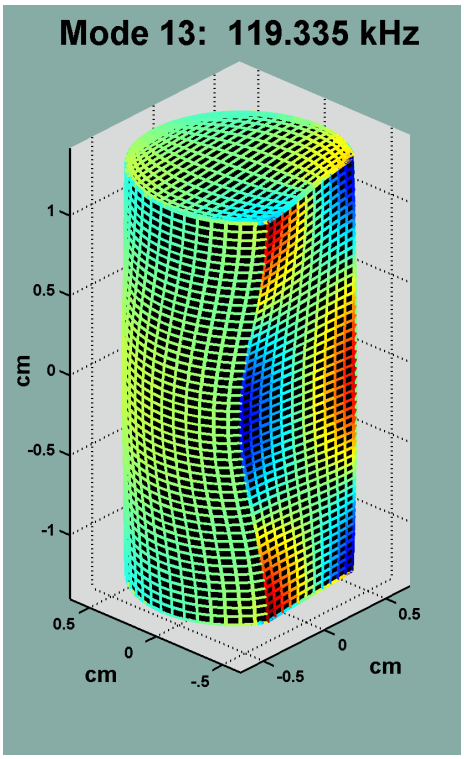

Figure A.7 Mode 13, Plane Cut 2

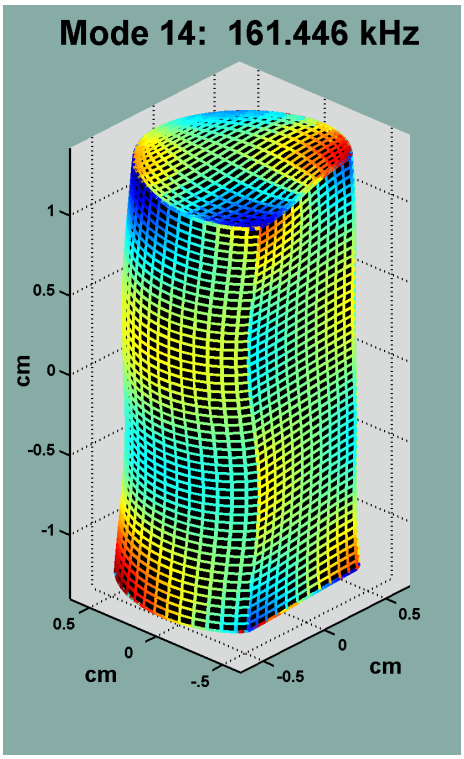

Figure A.8 Mode 14, Plane Cut 2

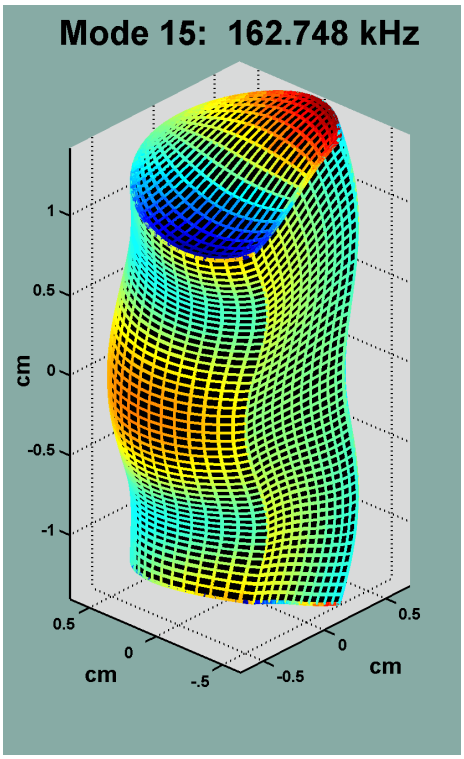

Figure A.9 Mode 15, Plane Cut 2

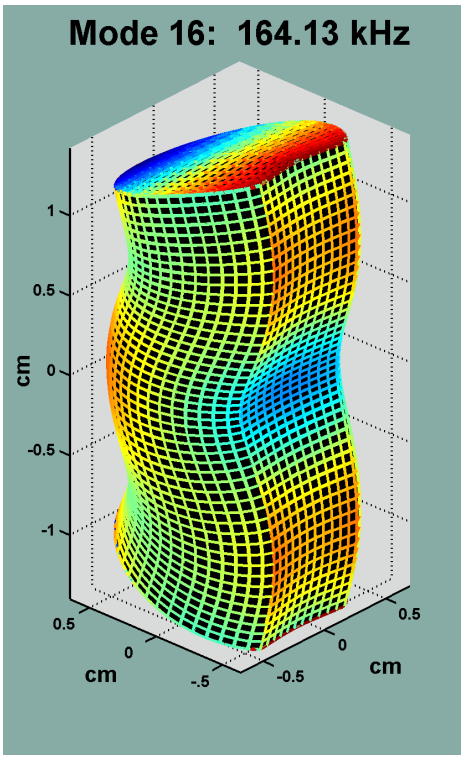

Figure A.10 Mode 16, Plane Cut2

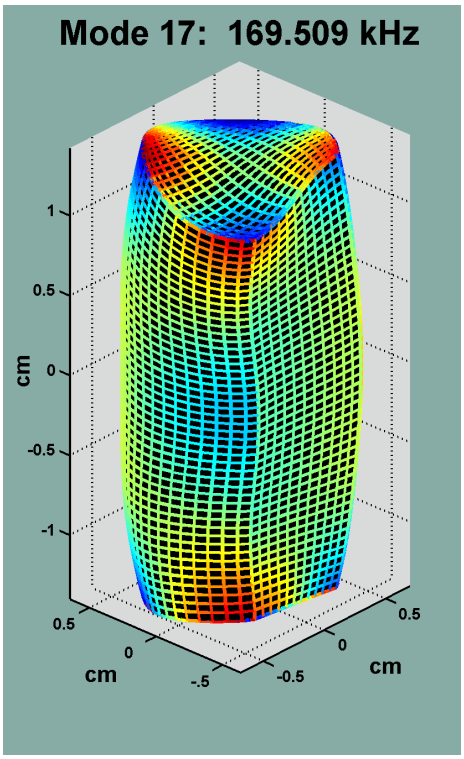

Figure A.11 Mode 17, Plane Cut 2

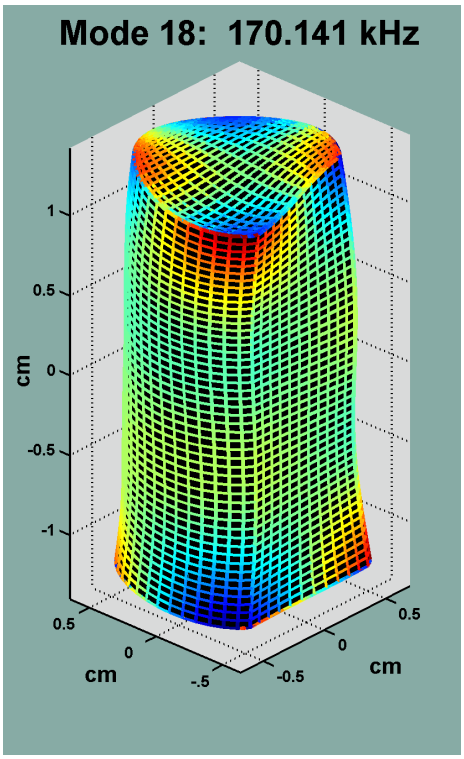

Figure A.12 Mode 18. Plane Cut 2

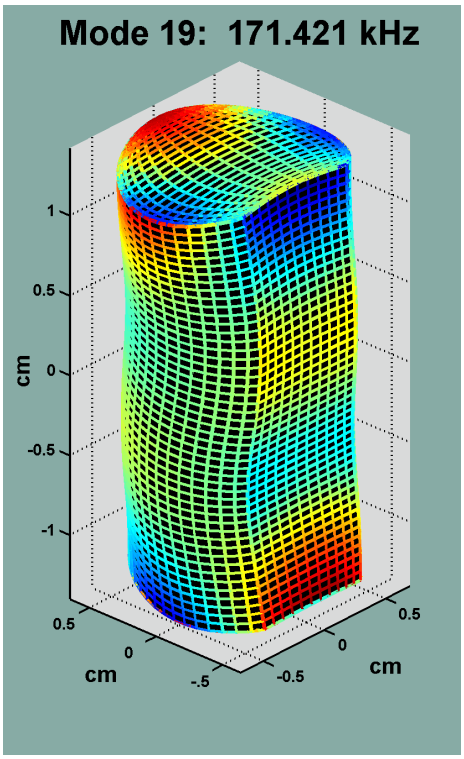

Figure A.13 Mode 19, Plane Cut 2

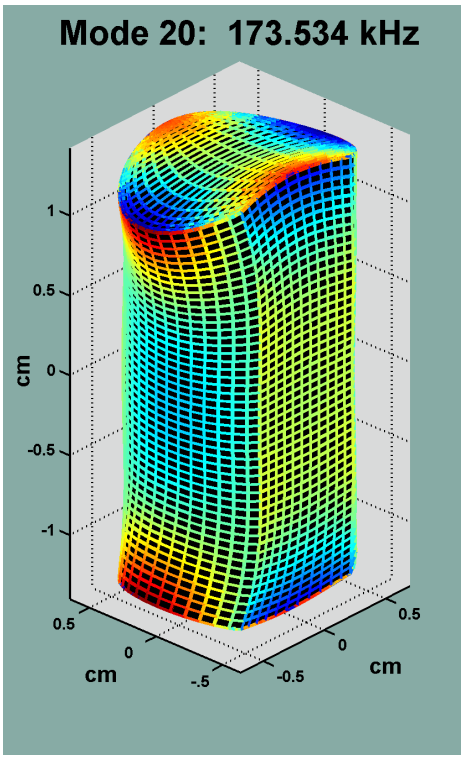

Figure A.14 Mode 20, Plane Cut 2

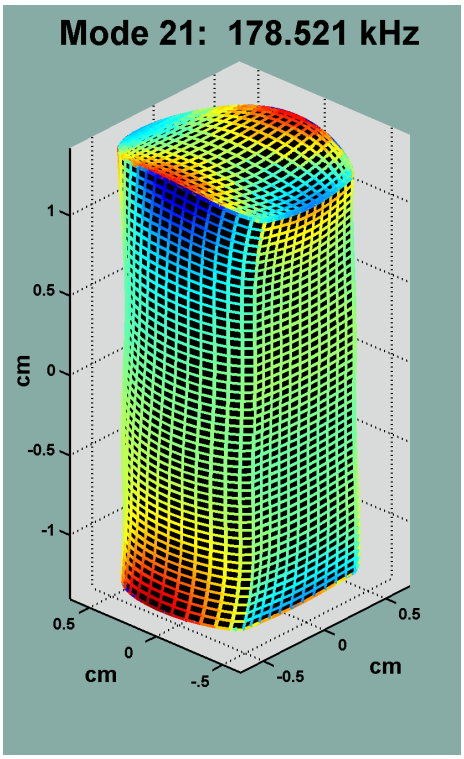

Figure A.15 Mode 21, Plane Cut 2

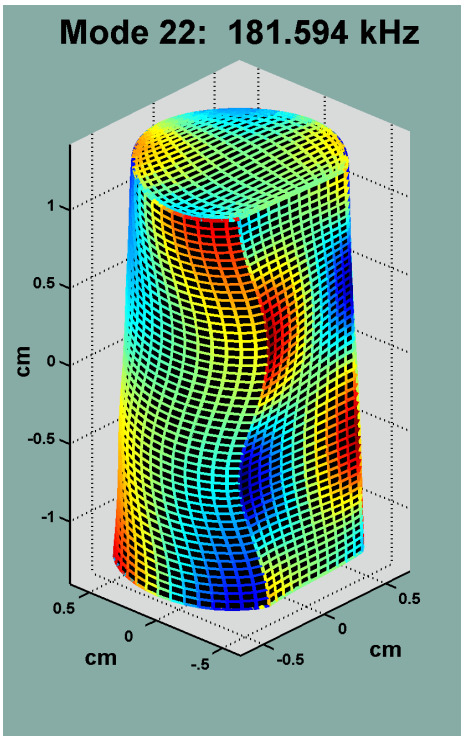

Figure A.16 Mode 22, Plane Cut 2

APPENDIX B

NOTCH CUT
16 MODE SHAPES

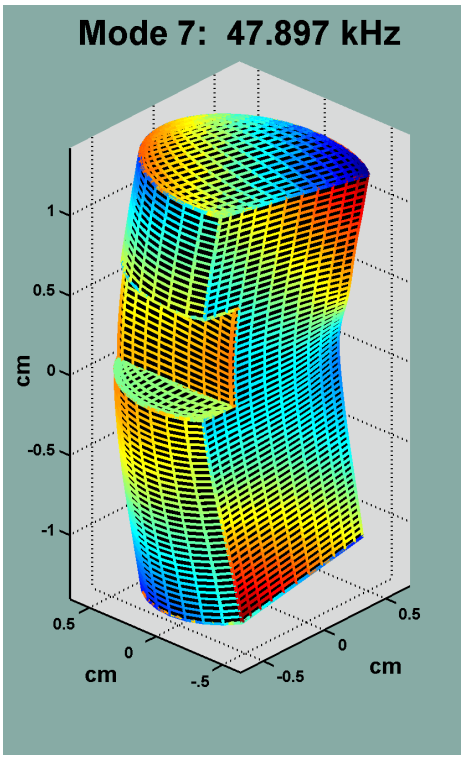

Figure B.1 Mode 7, Notch Cut

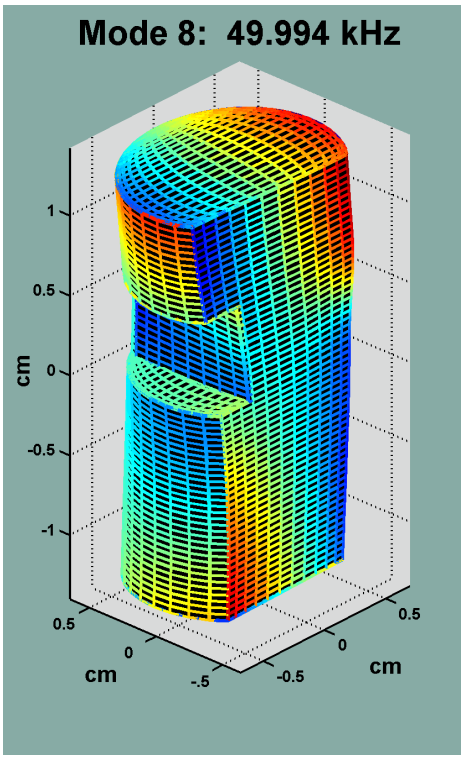

Figure B.2 Mode 8, Notch Cut

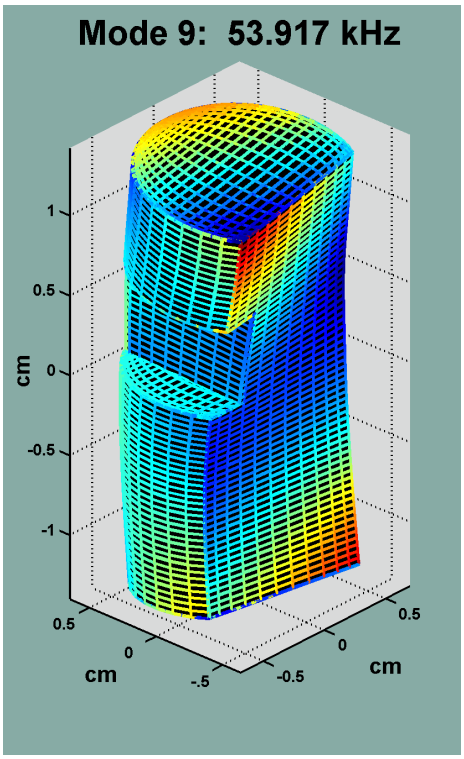

Figure B.3 Mode 9, Notch Cut

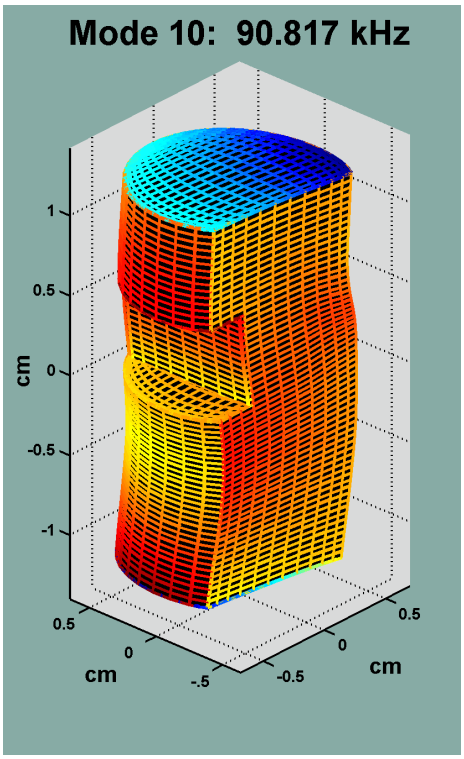

Figure B.4 Mode 10, Notch Cut

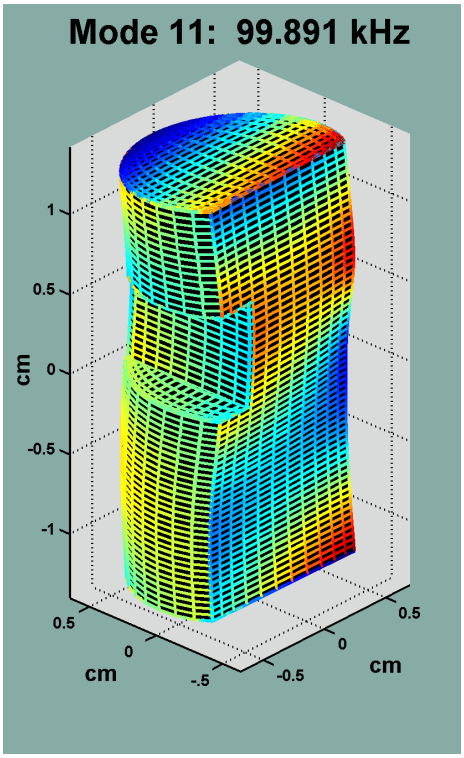

Figure B.5 Mode 11, Notch Cut

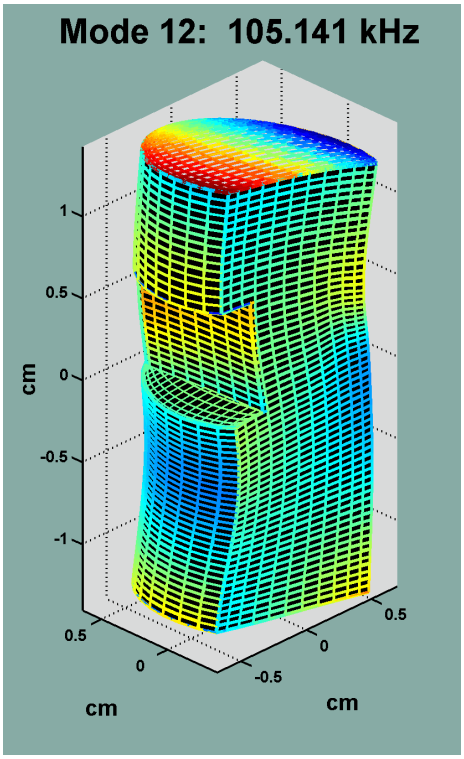

Figure B.6 Mode 12, Notch Cut

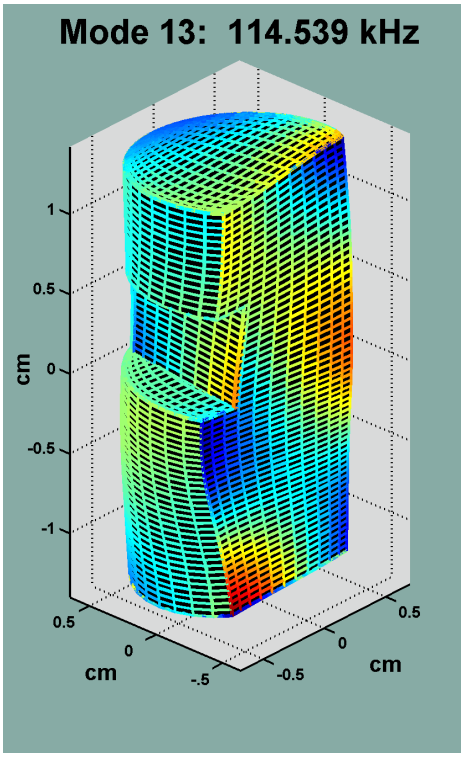

Figure B.7 Mode 13, Notch Cut

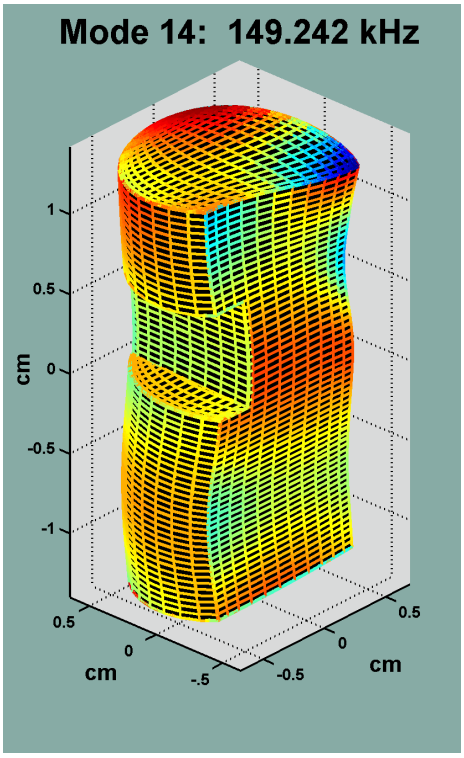

Figure B.8 Mode 14, Notch Cut

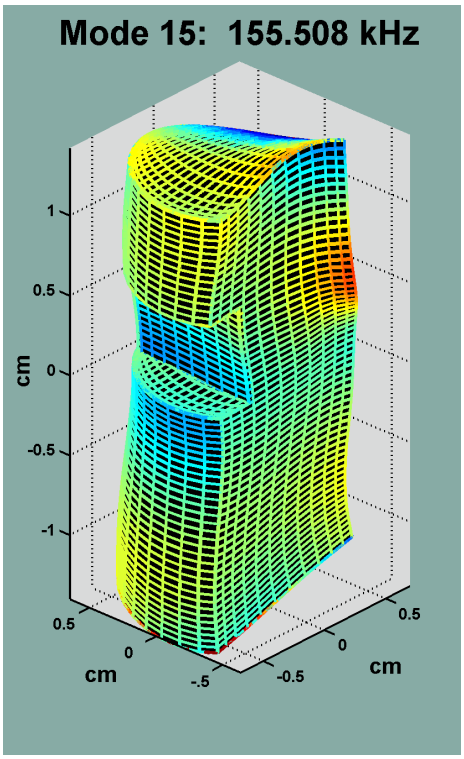

Figure B.9 Mode 15, Notch Cut

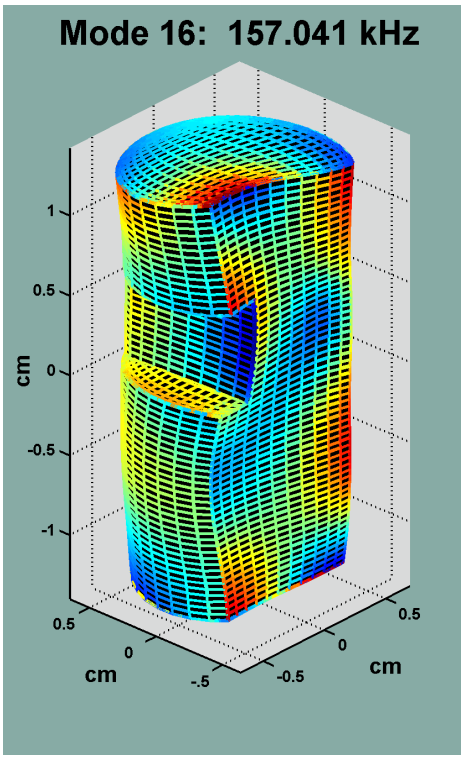

Figure B.10 Mode 16, Notch Cut

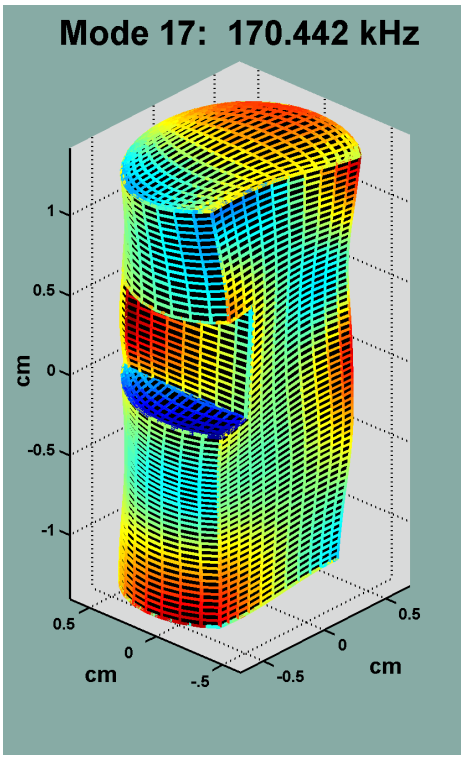

Figure B.11 Mode 17, Notch Cut

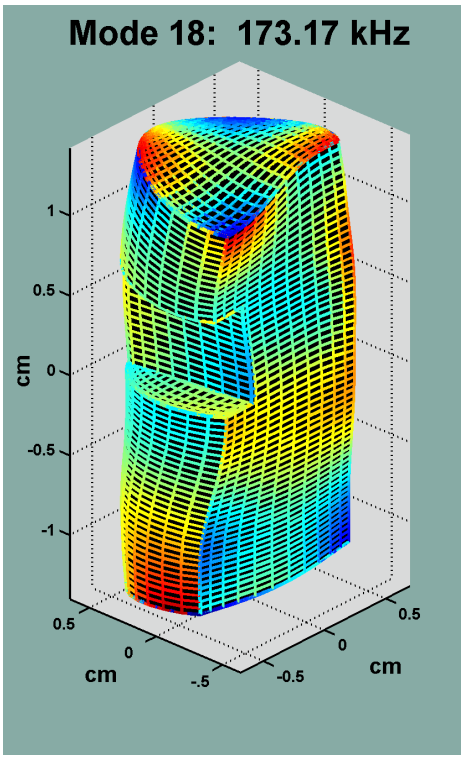

Figure B.12 Mode 18, Notch Cut

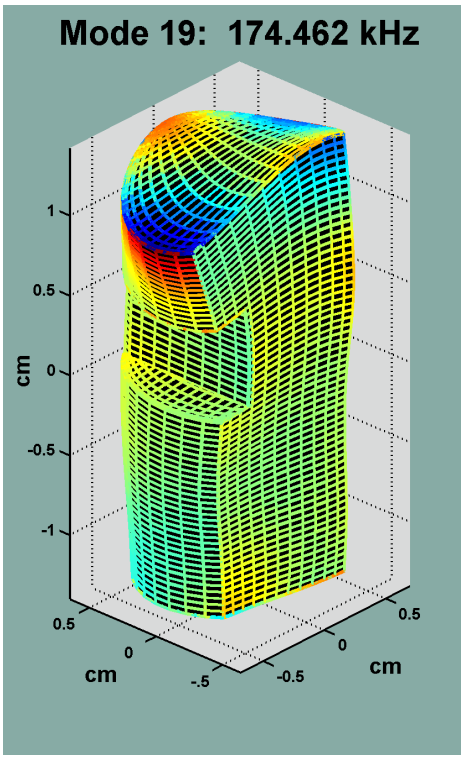

Figure B.13 Mode 19, Notch Cut

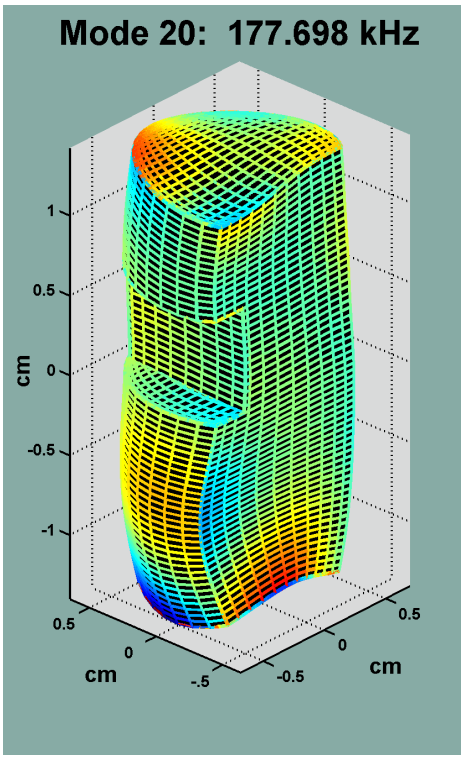

Figure B.14 Mode 20, Notch Cut

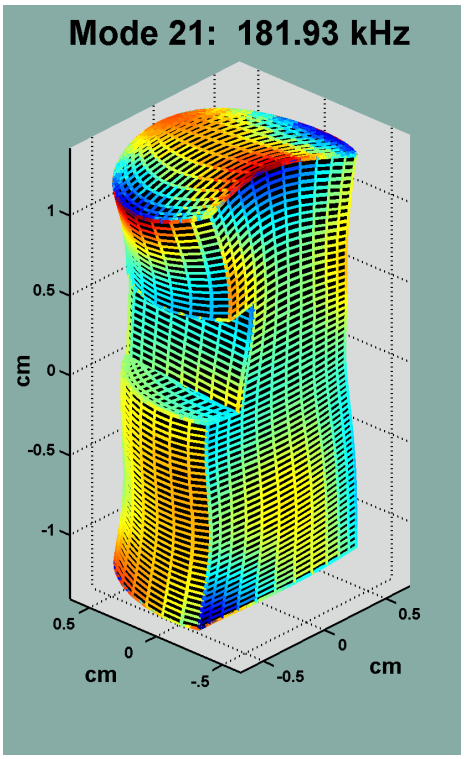

Figure B.15 Mode 21, Notch Cut

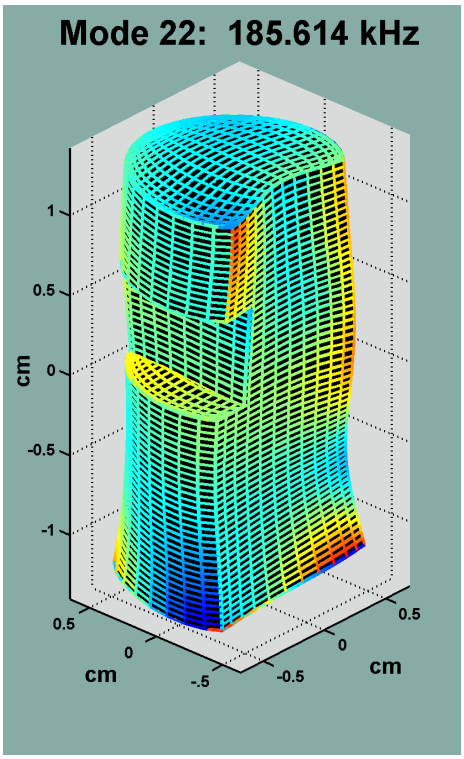

Figure B.16 Mode 22, Notch Cut

APPENDIX C

FORTRAN CODE

This appendix lists the Fortran code that was used to calculate the geometrical component of the $[E]$ and $[\Gamma]$ matrices found in (1.36) and (1.37) for the Plane Cut 2. The program outputs a text file that Matlab can then read and use to solve the eigenvalue equations. The Matlab code used to do this is listed separately in Appendix D.

```
PROGRAM Calcgeomcuttwo
************************************************************
*     The size of the matrices is sufficient for N = 14.                  *
*     To increase the N, E should be of rank = (N+1)*(N+2)*(N+3)/2        *
*     and the length of Geom should be 3*((rank)^2+rank).  LB,MB,NB       *
*     all will have to be changed to length rank as well.                 *
************************************************************
      DOUBLEPRECISION quad3d,k1,k2,Geom(12490920),E(2040,2040)
      INTEGER N,count,LB(2040),MB(2040),NB(2040),IC(2040),rank,sizer

      PRINT *, 'Enter the Visscher number : '
      READ *, N
      rank=(N+1)*(N+2)*(N+3)/2
      sizer=3*(rank**2+rank)
      k1=mclock()
      igg=0

      do 11 ii=1,3
      do 10 ll=1,N+1
      do 10 mm=1,N+1
      do 10 nn=1,N+1
      if(ll+mm+nn.GT.N+3) GOTO 10
      igg=igg+1
      IC(igg)=ii
      LB(igg)=ll-1
      MB(igg)=mm-1
      NB(igg)=nn-1
10     continue
11     continue

      count=0
      do 13 ig=1,igg
      do 12 jg=ig,igg
      I=IC(ig)
      J=IC(jg)
      LS=LB(ig)+LB(jg)
      MS=MB(ig)+MB(jg)
      NS=NB(ig)+NB(jg)
      count=count+1

      Geom((1+6*(count-1)))=quad3d((LS-2),MS,NS)
      Geom((2+6*(count-1)))=quad3d(LS,(MS-2),NS)
      Geom((3+6*(count-1)))=quad3d(LS,MS,(NS-2))
```

```
      Geom((4+6*(count-1)))=quad3d((LS-1),(MS-1),NS)
      Geom((5+6*(count-1)))=quad3d((LS-1),MS,(NS-1))
      Geom((6+6*(count-1)))=quad3d(LS,(MS-1),(NS-1))
      if(I.eq.J) E(ig,jg)=quad3d(LS,MS,NS)
      E(jg,ig)= E(ig,jg)
12    continue
13    continue

       k2=mclock()
       PRINT*,(k2-k1)/1000

       do 14 vv=1,19
       write(*,990) E(1,vv)
990   format (' Here:  ', D29.19)
14     continue
      PRINT*,' '
      PRINT*,E(1,500)
      PRINT*,' '
      do 15 vv=1,19
      write(*,991) Geom(vv)
991    format (' Here:  ', D29.19)
15     continue

*****************************************************
      open(UNIT=20,FILE='Geommcuttwo',STATUS='new')

      do 16 kk=1,sizer
      write(20,'(1X D29.19)') Geom(kk)
16     continue
      close(unit=20)

*****************************************************
      open(UNIT=21,FILE='EEcuttwo',STATUS='new')

      do 18 jj=1,rank
      do 17 bb=1,rank
      write(21, '(1X, D29.19)') E(jj,bb)
17     continue
18    continue
       end

  *****************************************************
      DOUBLEPRECISION FUNCTION quad3d(p,q,r)
```

```
DOUBLEPRECISION h,qgausx
INTEGER p,q,r
EXTERNAL h
quad3d=qgausx(h,p,q,r)
RETURN
END

DOUBLEPRECISION FUNCTION f(zz,p,q,r)
DOUBLEPRECISION zz,func,x,y,z
INTEGER p,q,r
COMMON /xyz/ x,y,z
z=zz
f=func(x,y,z,p,q,r)
RETURN
end

DOUBLEPRECISION FUNCTION g(yy,p,q,r)
DOUBLEPRECISION yy,f,z1,z2,x,y,z,qgausz
INTEGER p,q,r
EXTERNAL f
COMMON /xyz/ x,y,z
y=yy
g= qgausz(f,z1(x,y),z2(x,y),p,q,r)
RETURN
END

DOUBLEPRECISION FUNCTION h(xx,p,q,r)
DOUBLEPRECISION xx,g,y1,y2,x,y,z,qgausy
INTEGER p,q,r
EXTERNAL g
COMMON /xyz/ x,y,z
x=xx
h=qgausy(g,y1(x),y2(x),p,q,r)
RETURN
END

DOUBLEPRECISION FUNCTION qgausx(func,p,q,r)
DOUBLEPRECISION xm,dx,xr,x,w,AA,BB,func,ss


INTEGER p,q,r
DIMENSION x(11),w(11)
DATA w/0.13925187285563d0,0.13654149834601d0,0.13117350478706d0,
```

```
*       0.12325237681051d0,0.11293229608054d0,0.10041414444288d0,
*       0.08594160621707d0,0.06979646842452d0,0.05229333515268d0,
*       0.03377490158481d0,0.01462799529827d0/
      DATA x/0.06973927331972d0,0.20786042668822d0,0.34193582089208d0,
*       0.46935583798676d0,0.58764040350691d0,0.69448726318668d0,
*       0.78781680597921d0,0.86581257772030d0,0.92695677218717d0,
*       0.97006049783543d0,0.99429458548240d0/
      xm=0.5d0*(BB()+AA())
      xr=0.5d0*(BB()-AA())
      ss=0d0
      do 12 j=1,11
        dx=xr*x(j)
        ss=ss+w(j)*(func(xm+dx,p,q,r)+func(xm-dx,p,q,r))
12    continue
      qgausx=xr*ss
      RETURN
      END

      DOUBLEPRECISION FUNCTION qgausy(func,a,b,p,q,r)
      DOUBLEPRECISION xm,dx,xr,x,w,a,b,ss,func
      INTEGER p,q,r
      DIMENSION x(11),w(11)
      DATA w/0.13925187285563d0,0.13654149834601d0,0.13117350478706d0,
*       0.12325237681051d0,0.11293229608054d0,0.10041414444288d0,
*       0.08594160621707d0,0.06979646842452d0,0.05229333515268d0,
*       0.03377490158481d0,0.01462799529827d0/
      DATA x/0.06973927331972d0,0.20786042668822d0,0.34193582089208d0,
*       0.46935583798676d0,0.58764040350691d0,0.69448726318668d0,
*       0.78781680597921d0,0.86581257772030d0,0.92695677218717d0,
*       0.97006049783543d0,0.99429458548240d0/
      xm=0.5d0*(b+a)
      xr=0.5d0*(b-a)
      ss=0d0
      do 12 j=1,11
        dx=xr*x(j)
        ss=ss+w(j)*(func(xm+dx,p,q,r)+func(xm-dx,p,q,r))
12    continue
      qgausy=xr*ss
      RETURN
      END

      DOUBLEPRECISION FUNCTION qgausz(func,a,b,p,q,r)
      DOUBLEPRECISION xm,dx,xr,x,w,a,b,ss,func
```

```
      INTEGER p,q,r
      DIMENSION x(11),w(11)
      DATA w/0.13925187285563d0,0.13654149834601d0,0.13117350478706d0,
     *       0.12325237681051d0,0.11293229608054d0,0.10041414444288d0,
     *       0.08594160621707d0,0.06979646842452d0,0.05229333515268d0,
     *       0.03377490158481d0,0.01462799529827d0/
      DATA x/0.06973927331972d0,0.20786042668822d0,0.34193582089208d0,
     *       0.46935583798676d0,0.58764040350691d0,0.69448726318668d0,
     *       0.78781680597921d0,0.86581257772030d0,0.92695677218717d0,
     *       0.97006049783543d0,0.99429458548240d0/
      xm=0.5d0*(b+a)
      xr=0.5d0*(b-a)
      ss=0d0
      do 12 j=1,11
        dx=xr*x(j)
        ss=ss+w(j)*(func(xm+dx,p,q,r)+func(xm-dx,p,q,r))
12    continue
      qgausz=xr*ss
      RETURN
      END
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCC
C          Below the integral definitions are made.

      DOUBLEPRECISION FUNCTION func(x,y,z,p,q,r)     !CCCCCCCCCCCCCCCCC
      DOUBLEPRECISION x,y,z                          !CCCCCCCCCCCCCCCCC
      INTEGER p,q,r                                  !CCCCCCCCCCCCCCCCC
      func=(x**r)*(y**q)*(z**p)                      !CCCCCCCCCCCCCCCCC
      END                                            !CCCCCCCCCCCCCCCCC
                                                     !CCCCCCCCCCCCCCCCC
      DOUBLEPRECISION FUNCTION AA()                  !CCCCCCCCCCCCCCCCC
      AA=-1.27                                       !CCCCCCCCCCCCCCCCC
      END                                            !CCCCCCCCCCCCCCCCC
                                                     !CCCCCCCCCCCCCCCCC
      DOUBLEPRECISION FUNCTION BB()                  !CCCCCCCCCCCCCCCCC
      BB=1.27                                        !CCCCCCCCCCCCCCCCC
      END                                            !CCCCCCCCCCCCCCCCC
                                                     !CCCCCCCCCCCCCCCCC
      DOUBLEPRECISION FUNCTION y1(x)                 !CCCCCCCCCCCCCCCCC
      DOUBLEPRECISION x                              !CCCCCCCCCCCCCCCCC
      y1=-.47625                                     !CCCCCCCCCCCCCCCCC
```

```
      END                                                !CCCCCCCCCCCCCCCCCC
                                                         !CCCCCCCCCCCCCCCCCC
      DOUBLEPRECISION FUNCTION y2(x)                     !CCCCCCCCCCCCCCCCCC
      DOUBLEPRECISION x                                  !CCCCCCCCCCCCCCCCCC
      y2=.635                                            !CCCCCCCCCCCCCCCCCC
      END                                                !CCCCCCCCCCCCCCCCCC
                                                         !CCCCCCCCCCCCCCCCCC
      DOUBLEPRECISION FUNCTION z1(x,y)                   !CCCCCCCCCCCCCCCCCC
      DOUBLEPRECISION x,y                                !CCCCCCCCCCCCCCCCCC
      z1=-sqrt(.635**2-y**2)                             !CCCCCCCCCCCCCCCCCC
      END                                                !CCCCCCCCCCCCCCCCCC
                                                         !CCCCCCCCCCCCCCCCCC
      DOUBLEPRECISION FUNCTION z2(x,y)                   !CCCCCCCCCCCCCCCCCC
      DOUBLEPRECISION x,y                                !CCCCCCCCCCCCCCCCCC
      z2= sqrt(.635**2-y**2)                             !CCCCCCCCCCCCCCCCCC
      END                                                !CCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

APPENDIX D

MATLAB CODE

This Appendix lists the Matlab code necessary to solve for the resonant frequencies of a shape whose geometry was defined in the Fortran code. This code reads the text file which is the output of the Fortran code an uses the data therein to build the matrices needed for the eigenvalue equation. With these matrices, this code then solves for the eigenvalues and eigenvectors, then builds a structure to store the information needed to plot the displacements.

```
function [freqs, vects]=practgeomfortran (NN,file1,file2,ycut)
% [freqs, vects]=practgeom(NN,file1,file2,ycut)
% NN is the Visscher number, file1 has the geometric part of the E
% matrix and file2 has the geometric part of the Gamma matrix.
% Calculates the resonant frequencies from Fortran output and saves
% the system information in a structure called RUSdata.
rank=.5*(NN+1)*(NN+2)*(NN+3);
LB=zeros(1,rank);  MB=LB;  NB=LB;  IC=LB;   C=zeros(3,3,3,3);   IGG=0;
warning off MATLAB:divideByZero
lambda=.507812;   mu=.261597;   Rho=2.688;
tic
for ii=1:3
    for jj=1:3
        for kk=1:3
            for LL=1:3
                C(ii,jj,kk,LL)=lambda*del(ii,jj)*del(kk,LL)...
                               +mu*(del(ii,kk)*del(jj,LL)...
                               +del(ii,LL)*del(jj,kk));
            end
        end
    end
end
IC(1:rank/3)=1;   IC(rank/3+1:2*rank/3)=2;   IC(2*rank/3+1:rank)=3;

for I=1:3
    for L=1:NN+1
        for M=1:NN+1
            for N=1:NN+1
                if (L+M+N>NN+3),break,end
                IGG=IGG+1;
                LB(IGG)=L-1;  MB(IGG)=M-1;  NB(IGG)=N-1;
            end
        end
    end
end
RUSdata.Basis=[LB',MB',NB'];
RUSdata.rank=rank;
GAMMA=zeros(rank,rank);  NR=IGG;
Geom=textread(file2);


count=0;
for IG=1:NR
    for JG=IG:NR
        I=IC(IG);
        J=IC(JG);
        LS=LB(IG)+LB(JG);
        MS=MB(IG)+MB(JG);
        NS=NB(IG)+NB(JG);
        count=count+1;
        GAMMA(IG,JG)=C(I,1,J,1)*LB(IG)*LB(JG)*Geom(1+6*(count-1))+...
                     C(I,2,J,2)*MB(IG)*MB(JG)*Geom(2+6*(count-1))+...
```

```
                        C(I,3,J,3)*NB(IG)*NB(JG)*Geom(3+6*(count-1))+...
                        (C(I,1,J,2)*LB(IG)*MB(JG)+...
                        C(I,2,J,1)*MB(IG)*LB(JG))*Geom(4+6*(count-1))+...
                        (C(I,1,J,3)*LB(IG)*NB(JG)+...
                        C(I,3,J,1)*NB(IG)*LB(JG))*Geom(5+6*(count-1))+...
                        (C(I,2,J,3)*MB(IG)*NB(JG)+...
                        C(I,3,J,2)*NB(IG)*MB(JG))*Geom(6+6*(count-1));
            GAMMA(JG,IG)=GAMMA(IG,JG);
        end
    end
end
clear Geom
E=textread(file1,'%29f');
E=reshape(E,rank,rank);
[vects vals]=eig(E\GAMMA);
clear('E','GAMMA')

newvals=real(sqrt(diag(vals)/Rho))/(2*pi);
% Sort the vects and vals in the correct order.
[freqs,indx]=sort(newvals);
vects=vects(:,indx);
% Build structure to save data.
RUSdata.runtime=toc;
RUSdata.eigvct=vects;
RUSdata.rank=rank;
RUSdata.xdim=.635;
RUSdata.ydim=.635;
RUSdata.zdim=1.27;
RUSdata.ycut=ycut;
RUSdata.eigvl=diag(vals);
RUSdata.freqs=round(freqs*10^6);
save RUSdata  RUSdata
clear('vects','vals','freqs')

function out=del(a,b)
if a==b
    out=1;
else
    out=0;
end
```