

# Resonant ultrasound spectroscopy: The essential toolbox



Cite as: Rev. Sci. Instrum. 90, 121401 (2019); doi: 10.1063/1.5123165

Submitted: 3 August 2019 • Accepted: 12 November 2019 •

Published Online: 17 December 2019



View Online



Export Citation



CrossMark

Fedor F. Balakirev, Susan M. Ennaceur, Robert J. Migliori, Boris Maiorov, and Albert Migliori<sup>b)</sup>

## AFFILIATIONS

Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

<sup>a)</sup>Also at: Alamo Creek Engineering, Santa Fe, New Mexico 87506, USA.

<sup>b)</sup>Author to whom correspondence should be addressed: migliori@lanl.gov

## ABSTRACT

Resonant Ultrasound Spectroscopy (RUS) is an ultrasound-based minimal-effort high-accuracy elastic modulus measurement technique. RUS as described here uses the mechanical resonances (normal modes of vibration or just modes) of rectangular parallelepiped or cylindrical specimens with a dimension of from a fraction of a millimeter to as large as will fit into the apparatus. Provided here is all that is needed so that the reader can construct and use a state-of-the-art RUS system. Included are links to open-source circuit diagrams, links to download Los Alamos National Laboratory open-source data acquisition software, links to request free analysis software, procedures for acquiring measurements, considerations on building transducers, 3-D printed stage designs, and a full mathematical explanation of how the analysis software extracts elastic moduli from resonances.

© 2019 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/1.5123165>

## I. INTRODUCTION TO RUS AND ITS APPLICATIONS IN SCIENCE AND ENGINEERING

Resonant ultrasound spectroscopy (RUS) is a process whereby one measures the mechanical resonances or normal modes of a solid object and uses them to compute all the components of the elastic modulus tensor. The mechanical resonances of a solid object are uniquely determined by its shape, elastic moduli, symmetry, and density. One can reasonably expect that the correspondence between measured resonances and those computed is of the order of 1 part in  $10^4$ . The difference between computed and measured is caused by deviations in shape, homogeneity, and symmetry direction from those assumed in the computation. The computation of moduli from resonances is complicated—the two computational approaches are finite-element based and Lagrangian minimization based. We focus here only on cylindrical or rectangular parallelepiped (RPR) specimens with symmetry of orthorhombic or higher and with symmetry axes aligned with the geometric axes of the specimen. There are two reasons for this. One is that it is easiest to make, measure the dimensions of, and align the symmetry axes with those of the specimen for cylinders or rectangular parallelepiped resonators (RPRs). The other is that the computation of moduli from resonances can be achieved without the need for a finite-element code by using

Lagrangian minimization codes which are available free and are very fast.

To whet your appetite, we show a bit (10 out of 30 measured resonances) of the final fit of computed to measured frequencies and a RUS scan of a cylinder of polycrystalline gadolinium (Fig. 1) using the hardware, data acquisition codes, and analysis codes described here. We will explain all the features later (Fig. 2).

We describe here in sufficient detail the most common approach to acquiring resonances and computing moduli from them so that the reader can implement RUS on RPRs and cylinders with no other information required but from this article.

RUS can also be exceptionally effective by observing the change in resonance frequency of a single mode vs temperature. Because the (adiabatic) elastic moduli are second derivatives of the internal energy with respect to the strains, phase transitions always leave a signature (or “feature”) in the moduli vs temperature. For example, even in a second order phase transition (e.g., superconductivity), the bulk modulus is step discontinuous, while the shear moduli exhibit a break in slope. Thus, a plot of resonance frequency vs temperature can be used to map a phase boundary. Figure 3 shows such a plot for the ferromagnetic phase transition in the Gd specimen used for the example in Fig. 3. A clear break in slope at about 292 K is apparent as are differing temperature

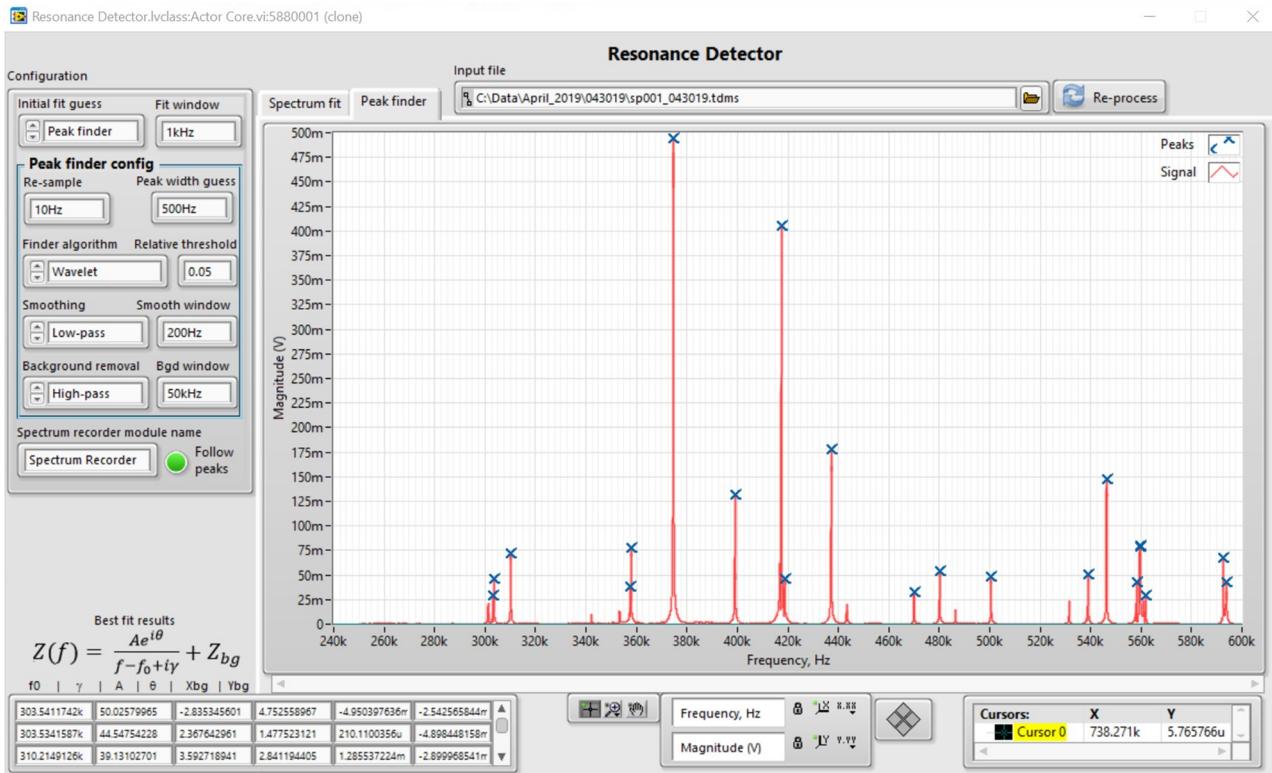


FIG. 1. An RUS scan of a Gd cylinder. The crosses show the peaks detected by the data acquisition code.

dependencies between the high temperature paramagnetic state and the low temperature ferromagnetic state. This particular mode has components 23% compressional and 77% shear as seen in line 2 of the fit shown above/under the d ln f/d ln c11 and d ln f/d ln c44 columns which are the fractional dependencies of the frequency vs elastic moduli.

Because the lowest RUS mode is often pure shear in an isotropic specimen, one can measure several resonances, assume that the

lowest mode is shear, and then subtract its normalized temperature dependence from that of another mode. If that second mode is not pure shear, the remainder will be the temperature dependence of the bulk modulus.

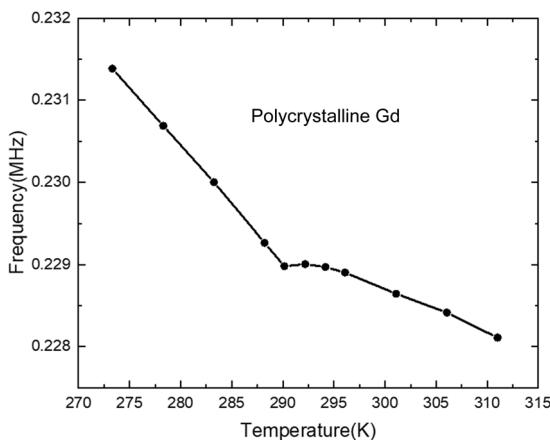
We note that there are three primary stumbling blocks to success in a RUS measurement. They are inaccurate sample geometry (including crystallographic alignment), high dissipation, and missing modes. Accurate geometry is under the control of the user in

```
LANL Cylcode Ver. 10.0 units cm, grams, GPa, MHz
Polycrystal Gd 305.15K .3300 .3300 .43
using 12 order polynomials mass= 0.2910 gm rho= 7.868 gm/cc
```

n	fexp(MHz)	fcalc(MHz)	%err	wt	k	i
1	0.1963020	0.1957102	-0.30	1.00	4	1
2	0.2284422	0.2278866	-0.24	1.00	1	2
3	0.0000000	0.2278866	0.00	0.00	7	2
4	0.3002769	0.3003283	0.02	1.00	5	1
5	0.3094602	0.3090473	-0.13	1.00	8	1
6	0.0000000	0.3090473	0.00	0.00	2	1
7	0.3381673	0.3382347	0.02	1.00	6	2
8	0.3387496	0.3382348	-0.15	1.00	4	2
9	0.3611121	0.3613038	0.05	1.00	5	2
10	0.3619525	0.3613038	-0.18	1.00	3	1
...						

```
2 x logarithmic derivates with respect to free moduli
dlnf/dlnc11 dlnf/dlnc44
0.00000 1.00000
0.22962 0.77038
0.22962 0.77038
0.13699 0.86301
0.05501 0.94499
0.05501 0.94499
0.00607 0.99393
0.00607 0.99393
0.06445 0.93555
0.06445 0.93555
```

FIG. 2. Actual code output of modulus fit to scan in Fig. 1.



**FIG. 3.** An RUS scan of the temperature dependence of a mode of polycrystalline Gd. The ferromagnetic phase transition is clearly detected as a break in slope.

the preparation of a cylinder or RPR, dissipation is controlled by the material, but missing modes, that is, modes that for whatever reason go undetected, feed a frequency set to the analysis code that cannot correspond to any real object. This last is one where we have taken great pains to maximize the dynamic range of the electronics so that weak resonances can be extracted in the presence of strong ones and made provisions in the analysis codes to indicate a missed mode. This last provision takes some practice to use. We will describe how best to deal with missing modes below.

## II. REQUIREMENTS OF THE SPECIMEN TO BE MEASURED

It is necessary that the specimen be homogeneous. Internal flaws such as voids or cracks, surfaces different from the interior, variations in composition, and variations in texture are examples of inhomogeneities. The specimen does not need to be isotropic. In fact, one of the great strengths of RUS is the ability to determine anisotropic elastic moduli in a single measurement. It is also necessary that the elastic moduli be well defined. Moduli are not well defined if dissipation is high, for example. Practically, this constrains the quality factor  $Q$  to be above several hundred.  $Q$  is defined to be the energy stored on resonance divided by  $2\pi$  times energy lost per cycle. It is computed from the resonance frequency divided by the full width at half maximum of the resonance *power*. If the data are the response voltage from a transducer, then the resonance power is proportional to the square of the response voltage. Thus,  $Q$  is equivalently the full width of the in-phase component of the voltage peak at points  $\sqrt{2}$  of the maximum. A steel cylinder of about  $\varnothing 6$  mm by 5 mm long has a  $Q$  of about  $2 \times 10^4$  for its lowest 30 modes.

The symmetry of the specimen is important. For example, an isotropic solid has no symmetry axis, but for the other symmetries handled by our Lagrangian minimization code (cubic, hexagonal, tetragonal, orthorhombic, and for textured material, fiber texture), the symmetry axes must be aligned with the sample

geometry. Errors in symmetry direction introduce cosine errors in the result. That is, small errors will have less than proportional effects on results.

There are two types of well-defined elastic moduli. They are adiabatic and isothermal. At low frequencies, the thermal penetration depth is less than an ultrasonic wavelength. Thus, heat cannot propagate from a compressional peak to trough in one period of the excitation and the wave propagates adiabatically. At very high frequencies, the thermal penetration depth is much greater than the wavelength, and so, heat transfers across an ultrasonic wavelength in times much less than the period and so the wave propagates isothermally. At zero frequency (mechanical testing), moduli are also isothermal. There can be a crossover between adiabatic and isothermal. In that frequency region, dissipation is high. RUS as described here measures adiabatic moduli.

## III. HARDWARE

### A. Less frequently used methods of acquiring resonances

For a resonating solid, the frequency at which a Lorentzian resonance is in phase with the drive, the maximum amplitude of vibration, and the frequency measured during ring-down of an impulse excited measurement all differ by of order  $1/Q^2$ . The classes of excitation used for RUS fall into three general categories. They are swept-sine (that we will describe here), impulse, and broadband noise excitation. Laser heating, electromagnetic coupling, tiny hammers, and electronically generated white noise have all been used, but we will not discuss such excitation methods.

Methods of detecting and recording resonances also vary. They include commercial so-called lock-in amplifiers, various heterodyne-based systems, and precision rectifier based systems and more. Some systems acquire only the amplitude of the resonance signal. This is a substantial disadvantage when attempting to find and analyze overlapping or weak resonances. By recording the in-phase and quadrature signals from the transducers, no information is lost, providing unambiguous input to software routines that fit Lorentzians to the acquired data. In implementing a full phase-sensitive RUS system, there are substantial advantages to an all-digital detection and recording scheme<sup>1</sup> and that is what we describe here.

### B. The Los Alamos piezoelectric-transducer-based RUS system hardware overview

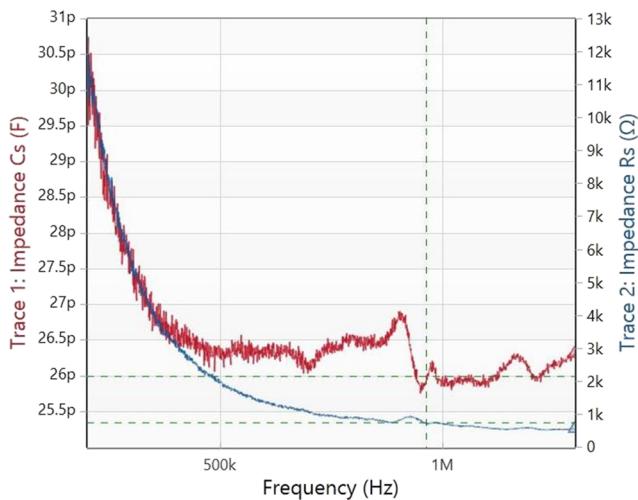
We describe the transducers and a simple stage for them, the conditioning electronics, and the microcomputer-based digitizing and signal processing. Features include a successful transducer design suitable from cryogenic temperatures to about 370 K, a preamplifier design that is immune to cable length effects, a drive amplifier that provides a robust 24 V peak-to-peak transducer drive, and the microcomputer-based high-speed digitizer, its signal processing, and conditioning. In Secs. V A and V B, we describe the data acquisition software and the analysis codes that compute elastic moduli from resonances. The circuit designs, data acquisition software, and analysis codes are all available upon request as open source.

### 1. Piezoelectric transducer construction

There are several design considerations. One is that the electronics must detect vibrations of an object making essentially point contact to the transducers with very little force applied. Weak contact between transducers and the sample ensures that the sample is as close to a free-body resonator as possible, a requirement for the analysis codes to provide accurate results. However, weak contact requires sensitive electronics. A consequence is that it is essential that there be no electrical contact between the sample and transducer electrodes, else with a conducting sample, the drive signal can leak, via ground return voltages generated in the shield cable resistance, into the detect signal path. The alumina cup we discuss below ensures electrical isolation. It is also important that the transducers do not generate much in the way of their own resonances. It is impractical to make transducers with no resonances at all. Note also that any technique for damping the intrinsic resonances of the transducer also generates Johnson noise just like a resistor in an electrically resonant circuit. Finally, we note that application of a voltage to the transducer element generates a stress. The resulting strain depends on the mass and moduli of the transducer material, and the frequency. The higher the frequency, the greater the inertial effects, and the smaller the resulting strain, and it is the strain that drives the specimen resonances. The transducers described here do have resonances, but they are weak and broad, hence easily separated from those of the sample. We show the electrical response of the transducers described here in Fig. 4.

### 2. Construction details

Note that although we use “piezoelectric” to describe the transducers, in fact those described here are not piezoelectrics but are lead-zirconate-titanate ferroelectrics, used because they have higher dielectric constants and stronger response and are cheaper than piezoelectric crystals. There are three steps to assemble the

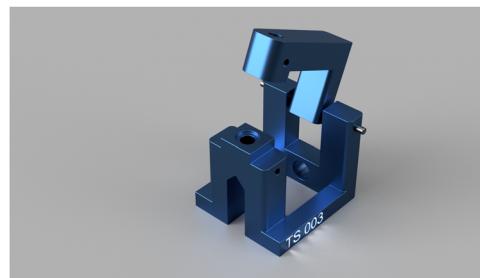


**FIG. 4.** The complex electrical impedance of a transducer of the type described here. Trace 1 is the noisier trace.

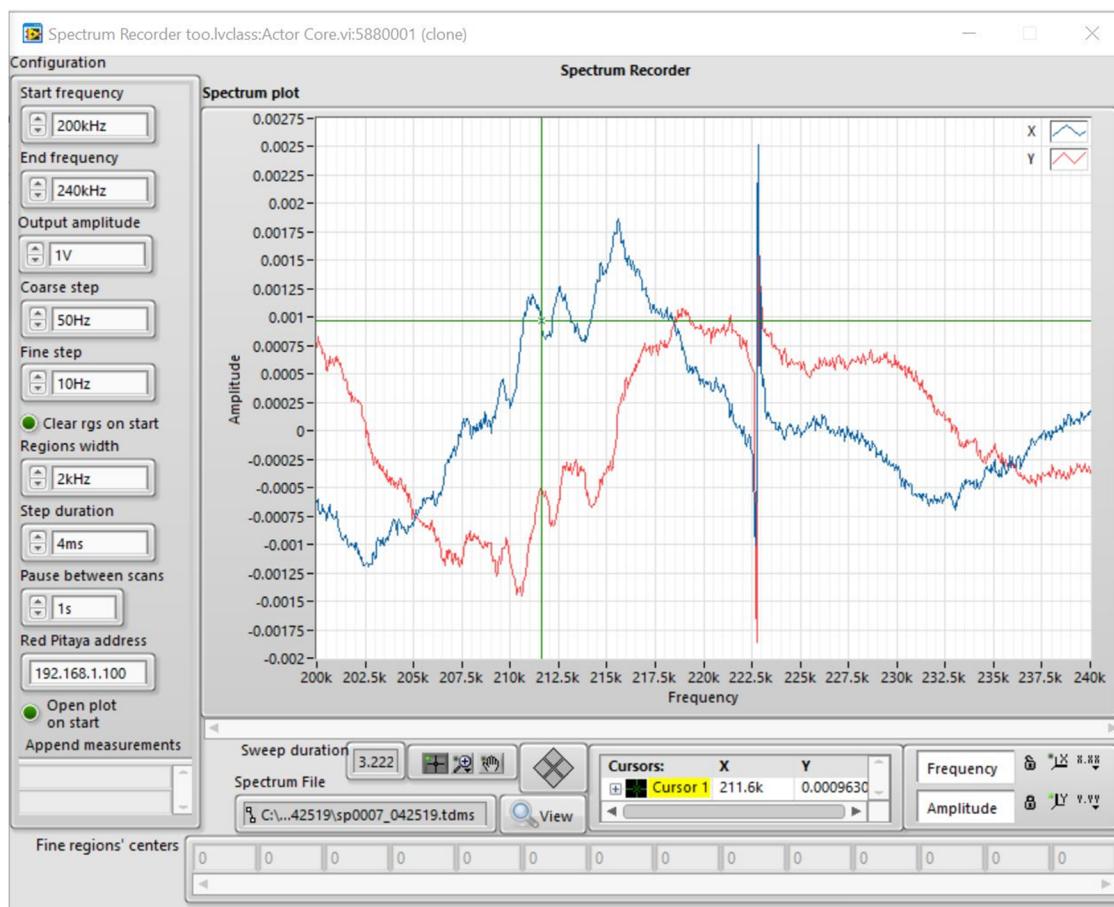


**FIG. 5.** The transducers for drive and detect are identical. They have four components: an SMA female barrel (left), an aluminum tube 8 mm  $\varnothing$  with 1/4-36 female threads at one end and a counterbore to fit the alumina cup at the other, a  $\varnothing$  5 mm piezoelectric transducer<sup>3</sup> with any of several types of wrap-around electrodes, and an alumina cup that is 7.2 mm  $\varnothing$ , 6 mm bore, 5 mm long, 4 mm deep.

transducer (Fig. 5). Each step requires a 24 h cure at 50 C. All use a marine-grade very-slow-cure epoxy.<sup>2</sup> The first is to epoxy the transducer element<sup>3</sup> to the inside of the alumina cup.<sup>4</sup> Very little epoxy is needed and light pressure (a wood closepin and 6 mm airsoft plastic BBs work fine). Let cure. Then, 36 AWG (American wire gauge) thermally strippable 3.5-cm-long Cu magnet wire is soldered to the two transducer electrodes with ordinary electrical solder. Identify the lead attached to the transducer face in contact with the cup as this will be the ground electrode. Next, fill the cup to about 3/4 full with epoxy and slowly dribble in 200 grit alumina until the epoxy surface is reached. This takes some patience. The alumina controls thermal expansion stresses for cryogenic use. Let cure. Insert the transducer leads through the tube and solder to the SMA (SubMiniature version A) female barrel, taking care not to get solder on the threads and to connect transducer ground to the SMA barrel shell. Screw the barrel into the tube, leaving the transducer assembly loose. Pour some epoxy into the tube. Do not fill more than half way as then air pressure will build when the transducer assembly is inserted. Be sure to get some epoxy on the female threads of the aluminum tube. Insert the transducer assembly into the tube. Apply some weight to the assembly to keep it from pushing the cup out as it cures. Check that there is no electrical short across the SMA connector. Let cure. For a rough test, connect the transducer via a 30 cm SMA cable to the input of an oscilloscope. The cable capacitance loads the transducer output, so longer cables will yield smaller peaks in this test. For a good transducer, tapping the alumina wear face with the handle of a metal tweezers should give single pulses of about 5 V or more.



**FIG. 6.** This 3D printed stage requires some final assembly. Use a #20 drill for the arm and #19 drill for the base; drill and tap arm and base for 1/4-20 NC threads; drill out transducer holes with a P drill. Use 10-32 set screws to center the arm pivot on the base. Use a 4 mm  $\varnothing$  steel dowel pin for the pivot. Use nylon 1/4-20 NC screws to hold transducers in place.



**FIG. 7.** Shown is a very weak specimen resonance in the presence of the acoustic background associated with transmission through air and the weak remnants of the transducer resonances. The specimen resonance at 222.5 kHz is easily distinguished from background. The ratio of largest to smallest specimen resonance that can be assigned reliably is about  $10^5$ .

### C. 3D-print design for room temperature stage

A simple stage for bench-top measurements is shown in Fig. 6. This is 3D printed from ABS with a 20% fill. The design and material result in an acoustically dead stage, reducing direct acoustic transmission from one transducer to the other (Fig. 7). The STL files for 3D printing can be obtained free from Ref. 4.

## IV. ELECTRONICS

### A. Overview of strategy for acquisition

The resonances of the solid objects usefully measured using RUS are approximately Lorentzian in shape. The approximation is better for higher- $Q$  materials.<sup>5</sup> For a  $Q$  greater than about 100, the approximation is good enough that a Lorentzian fit is within the noise. The electronics described here acquires resonances with the maximum possible speed, independent of cable length for cables less than about 2 m, and delivers the information via an Ethernet port to a computer running Microsoft<sup>TM</sup> Windows 10. The

maximum possible speed is determined by the width in frequency of the resonance. For example, an  $f = 1$  MHz resonance  $w = 100$  Hz wide ( $Q$  is 10 000) takes about  $1/w = 10$  ms ( $Q/f$ ) for the resonance to respond. Thus, the total time spent sweeping frequency through the resonance must be much longer than  $1/w$ . The time to sweep through the resonance is determined in the data acquisition software by the spacing between frequency steps (frequency is scanned in discrete steps by the system described here) and the duration of each step. The dynamic range from noise floor to maximum signal for the electronics described here with no transducers connected is of order  $10^6$ . There are three components of the electronics. A preamplifier (1) buffers the signal from one transducer to input 1 of the Red Pitaya microcomputer 14-bit A/D converter (2). Output 1 of the microcomputer drives a boost amplifier (3) to produce up to 24 V P-P transducer drive. The microcomputer has a Field Programmable Gate Array (FPGA) programmed specifically for this application by Los Alamos; its image is freely distributed.<sup>6</sup> The microcomputer firmware uses sophisticated signal processing to reduce noise in ways not possible with any purely analog design.

## B. Charge amplifier

One challenge in using PZT transducers is that they have an output impedance that is nearly purely capacitive (the ones described here of order 30 pF, Fig. 4). At 500 kHz, the impedance is about 10 k $\Omega$ . For a 1 mV resonance, this produces about 100 nA. Thus, it is not the voltage noise of the first bit of electronics that the receive transducer signal encounters but the current noise. This forces the use of CMOS op-amps in the first stage. Another challenge is to maintain gain over a very broad frequency range. Los Alamos makes the schematic and printed circuit board (PCB) layout for the solution to these problems available.<sup>6</sup> The PC design software used is freely available<sup>7</sup> from Sunstone Circuits™. The preamplifier is powered via a micro-USB (universal serial bus) standard cable with current draw low enough that any USB port will power it.

The design is most easily understood if we distill the circuit diagram into the most basic of its functions, as in Fig. 8. Noting that op-amps have extremely high differential gain, the negative feedback from C keeps the - terminal at the same voltage as the + terminal which is fixed at a level defined by the setting of  $R_{trim}$ . Op-amps also have very high input impedance, so no current flows into the + or - terminal. Thus, (ignoring noise) any current into - generated by the transducer  $C_t$  is extracted by the current into C. The result is that the output voltage is  $-C/C_t$  times the transducer open-circuit voltage. That is, the gain is independent of frequency and the output of the transducer is held at a fixed voltage. Thus, the transducer short-circuit current is what is amplified. In addition to frequency-independent gain, because the transducer output is held at fixed voltage by the op-amp, the voltage between the center conductor and the shield of the coax is unchanging. Therefore, no current flows into the cable capacitance, and so, the system has gain independent of both frequency and cable length. However, op-amps produce tiny amounts of current and voltage even when there is no differential input voltage. The result is that C is quickly charged to the power supply voltage and the preamplifier quits working. The fix is to add

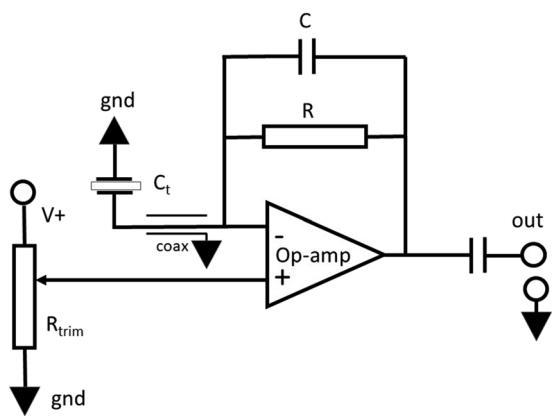
R to prevent saturation. However, now the output before the final capacitor is some random DC voltage. Adjusting  $R_{trim}$  makes the DC go to half (1.65 V) the supply voltage (3.3 V from the onboard regulator, not shown) for maximum possible peak-to-peak output. As long as frequencies are well above 1/RC (about 17 kHz), the circuit works as described. Below this, the gain is frequency dependent, but worse, the Johnson noise of R (about 200 nV/ $\sqrt{Hz}$ ) or about 200  $\mu$ V at 1 MHz bandwidth is no longer shunted out by C. In the actual Los Alamos implementation,<sup>6</sup> there are several features including an onboard voltage regulator to block noise from the USB power source, a second gain stage, and other details. The op-amp is chosen for low current noise at the expense of voltage noise because it is current noise into C that dominates. Useful gain ranges from about 30 kHz to 4 MHz for the design freely available.

## C. Boost amplifier

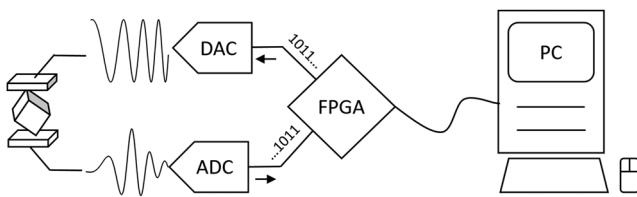
The boost amplifier<sup>6</sup> is a very simple op-amp-based inverting amplifier with gain adjustable using the onboard potentiometer. It is powered via a micro-USB standard cable with current draw low enough that any USB port will power it. To obtain the 24 V peak-to-peak output, an on-board DC-DC converter provides  $\pm 15$  V to the bipolar op-amp. The op-amp is chosen to provide sufficient drive to 2 m coaxial cables. It can produce full output more or less from a few kilohertz to 4 MHz. However, above about 6 MHz, the current required to drive 2M of coax is such that the op-amp gets quite warm. Heating is lower with shorter cables. Thus, a stick-on heat sink on the op-amp is used. The setup of the circuit board is done by setting the microcomputer to scan between 100 kHz and 110 kHz using the RUS data acquisition software set to 1 V amplitude, and adjusting the gain until maximum output without clipping is observed. This will be about 24 V peak-to-peak.

## V. HOW TO MAKE A MEASUREMENT USING LANL OPEN-SOURCE DATA ACQ SOFTWARE

The advent of digital signal processing (DSP) technologies, and Field Programmable Gate Arrays (FPGAs), in particular, opened many avenues for improvement in megahertz signal generation and detection, such as flexible configuration, high precision, compact implementation, low cost, and noise reduction and filtering strategies impossible with purely analog systems. Because resonant ultrasound spectroscopy can be applied to very small specimens (millimeter-sized specimens of diamond have useful modes above 8 MHz), Digital-to-Analog (DAC) and Analog-to-Digital (ADC) converters with sampling rates above 100 MSPS are required to provide the bandwidth necessary for direct digitization of RUS signals. The RUS data acquisition process for the system described here is implemented in a multiplatform approach (Fig. 9). A low-noise FPGA-based Red Pitaya-based detector system<sup>8</sup> is programmed both to generate high frequency excitation and to record the response using direct high-speed digitization. The detector system architecture is derived from Red Pitaya Notes open source code made available by Pavel Demin.<sup>9</sup> The Resonance Spectrometer PC-client user interface, available for free download from Los Alamos National Laboratory (LANL),<sup>6</sup> is implemented as a stand-alone Windows 10™ executable designed using National Instruments LabVIEW™ on the



**FIG. 8.** Distilled schematic of the transducer preamplifier. It is powered by a single supply. The transducer has capacitance  $C_t = 30$  pF, R is 2 M $\Omega$ , and C is described in the text as are the DC trim  $R_{trim}$  and op-amp (triangle). The output capacitor is any large value such as 2  $\mu$ F to block DC. The transducer is connected to the preamplifier with coaxial cable (coax) with a grounded shield.



**FIG. 9.** Block diagram of the Red-Pitaya FPGA-based resonance detector. A computer readies the FPGA to output a sequence of user-defined frequencies which are synthesized using the Red Pitaya DAC. The voltage response is simultaneously detected by the Red-Pitaya FPGA code via ADC and transmitted to the computer.

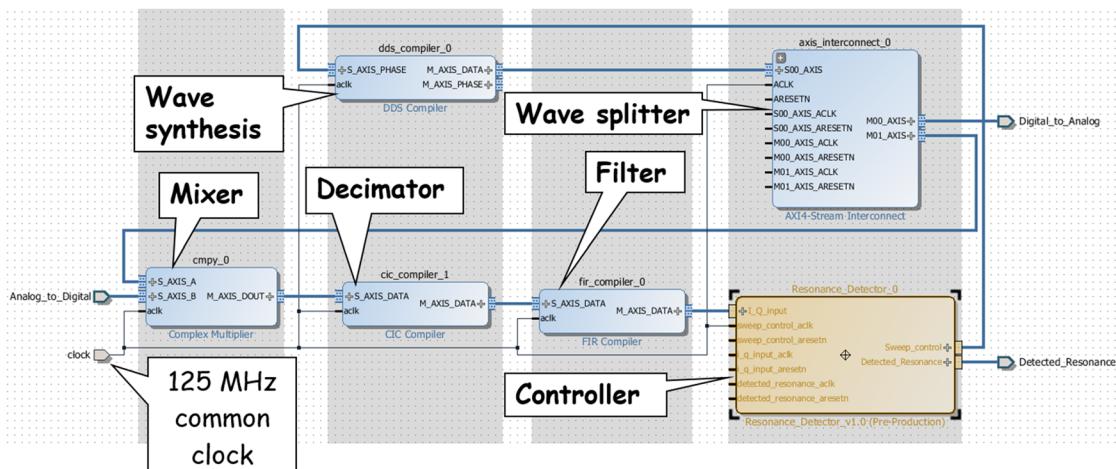
personal computer (PC). The user interface application communicates with the detector over an Ethernet connection.

The Red Pitaya DAC is programmed to sweep a 125 MSPS sinusoidal excitation. Sweep parameters, including start frequency, end frequency, step duration, and frequency step, can be set by the user in the “spectrum recorder” module of the data acquisition code. Simultaneously, the Red Pitaya ADC records the voltage of the transducer preamp output synchronously with the sine wave excitation generated by the Red Pitaya DAC. A secondary bandwidth and noise reduction occurs here because the preamp is bandwidth limited to about 10 MHz. The recorded waveform, is a set of numbers the result of 125 MSPS digitization. Each number is multiplied by the complex exponential function  $e^{j\omega t}$  in the software-implemented mixer (a mixer multiplies instantaneously two AC signals). The result of the multiplication is then averaged for a time set by the step duration and decimated to obtain in-phase and quadrature signal components referenced to the original excitation waveform. The value for  $t$  is taken to be the mean time over which each digitization occurs. This is the primary bandwidth-limiting noise reduction process. It is purely digital and has no analog. It is far superior to the time constant of a lock-in amplifier while being faster settling, more stable,

and with fewer artifacts. It is these results that are delivered to the PC. Frequency is then advanced to the next step, and the process is repeated. This front-end stage of the RUS DSP is implemented in the Red Pitaya FPGA fabric, which makes it possible to synthesize, detect, and process complex waveforms in real time at the native data converter rate of 125 MSPS (Fig. 10) while delivering only what is needed to the PC, greatly reducing bandwidth requirement of the Ethernet interface.

The in-phase and quadrature (decimated) data points together with the mean excitation frequency at each reported data point are streamed to the PC client, which delivers it to the frequency spectrum file. On completion of each frequency sweep, the spectrum file is analyzed by the “resonance detector” module of the PC client software. The resonance detector can be configured to find the approximate position of the resonance peaks (“peak detector” function) or, alternatively, look for peaks at last known locations (“follow peaks” function). The resonance detector then attempts to fit individual and overlapping resonances to a complex Lorentzian. The results of the fit for each resonance, such as its center frequency, width, amplitude, and relative phase, are recorded in the spectrum file as well as a measurement log file. Additional and optional complementary measurements, e.g., temperature and magnetic field, can be recorded with the resonance data.

To reduce measurement time and improve throughput, the frequency sweep can be configured to focus on the narrow frequency bands of interest around known resonances. This is accomplished by coarse frequency steps through regions known to contain no resonances and fine steps through regions with resonances. The software can adjust the positions of the focus bands after each frequency sweep to follow the detected center frequency of each resonance. This feature is particularly useful in determining temperature dependence or the presence of phase transitions. There are many other features of the PC client that are documented on the LANL<sup>6</sup> GITHUB site.



**FIG. 10.** Block diagram of the detector programmable logic generated in the Vivado<sup>10</sup> Design Suite. The excitation waveform is synthesized by a direct digital synthesizer and streamed out via the DAC. The ADC records the response signal, which is then mixed with the excitation waveform, decimated, and filtered to detect in-phase and quadrature signal components. The controller module configures the logic and transmits the results.

## A. Acquisition of resonances and finding missing modes

A manual for use of the Los Alamos data acquisition code, as well as the code itself, is regularly updated.<sup>6</sup> In the manual are reasonably detailed instructions for use of many features. We present here overall strategic guidance on what is important for acquiring the data needed by the analysis codes, described in Sec. VI B. Begin by ensuring that the RUS system is connected. Using the Windows 10 command prompt and pinging the IP address (192.168.1.100) of the spectrometer will test the Ethernet connection. Open the Resonance Spectrometer executable and click start. A scan should begin that, with no sample in place, will be random responses as sound is conducted through air and direct transducer-transducer contact (functional test would measure the resonance spectrum of a standard material such as a hardened and ground steel cylinder). Position the specimen of material to be measured between the transducers taking no care in its placement at this point. Steel has a quality factor Q of about  $2 \times 10^4$  for its lowest 30 modes and will produce resonance peaks that can be clearly identified and fit to determine how the system is operating. For either a test object or an actual specimen, it is useful to generate a customized configuration file with start and end frequencies, coarse and fine frequency steps, and drive amplitude. The best way to get the start and stop frequencies is to run a single pass of the analysis codes with specimen dimensions, mass, and a guess at elastic moduli so that approximate values are found. Using the main window, “write” the resulting configuration to save it. Note that as the data acquisition code is updated, older configuration files may be invalid. It is extremely important to ensure that the frequency range covered contains the lowest frequency as this is often a pure shear mode which both anchors the data set and is the required input for the autoguess feature, if used, of the analysis codes. Take a scan and then adjust drive amplitude so that no peaks are above 0.9 V to prevent distortion. Reduce drive as needed.

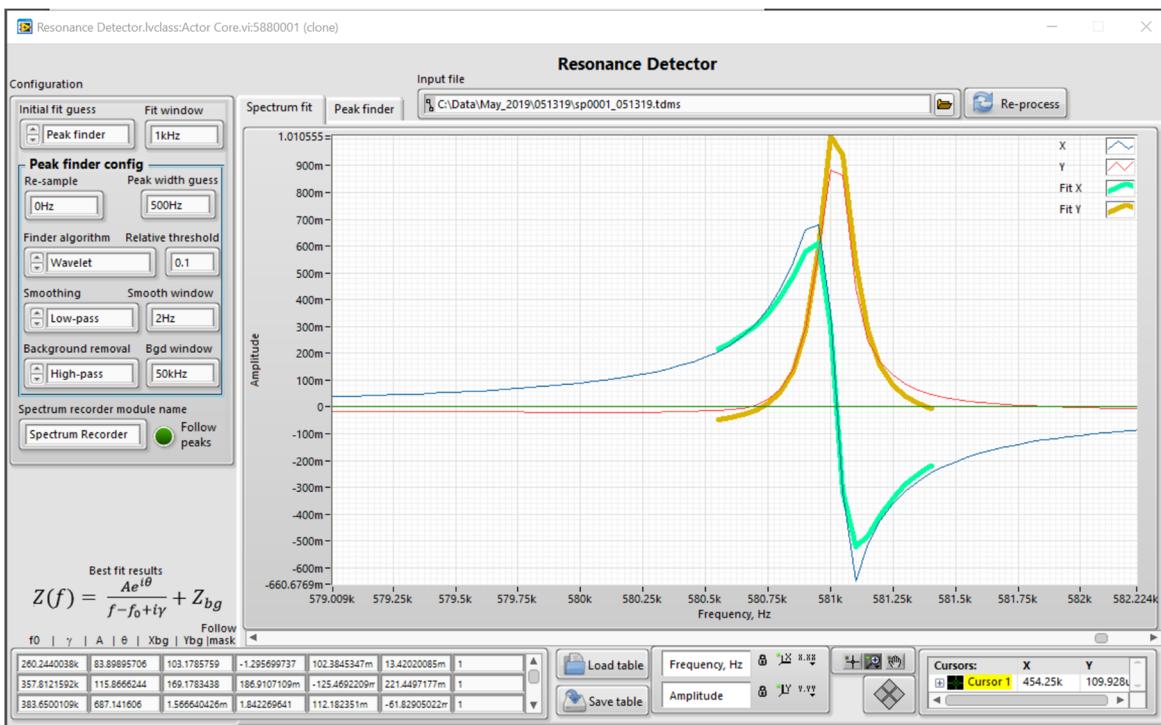
With the system set up for the specimen to be measured, note that the resonance amplitude is dependent on specimen placement on the transducers. Near the transducer center, drive is perpendicular, and near the edge, it is parallel to the transducer surface. Each mode has a different combination of parallel and perpendicular motion. Removing and replacing the specimen differently will make modes increase or decrease in amplitude, sometimes by very large amounts, and is the primary method for finding (missing) modes. Start by positioning the specimen with the minimum of material touching the transducers, for example, at the corners of a cube or the edges of a cylinder. This provides a close approximation to a free resonator, required by the analysis codes. Rotate the specimen to produce the most well-identified resonances as can be readily achieved. The specimen can also be positioned with a flat face or side of a cylinder on the transducer face with a negligible effect on accuracy while often exposing missing modes while suppressing others. Note that peaks can overlap, for example, from symmetry-required doublets or they may be hidden under a larger peak. The transducers will talk to each other through air and the mounts, so the user should be aware that some broad small peaks may not be relevant to the specimen measurement. However, the user should note that it is not necessary to determine all the modes to achieve a good result.

On completion of the scan, the “Resonance Detector” window displays peaks found and fit. The peak detector is a full Lorentzian-fit complex peak finder with the equation shown at lower left in Fig. 1. The relative threshold determines the sensitivity of the peak detector. If this is set too low, multiple hits will occur on the same peak and noise will be detected as peaks. Set so that no spurious peaks are found (Fig. 11) and later manually locate missing peaks (Fig. 12) using the magnifier and cursor tools or a built-in local fitting process described in the manual. The signal-to-noise ratio for the 500  $\mu\text{V}$  peak is so high that it is easy to see that 10  $\mu\text{V}$  peaks are detectable. Note that the “Smooth” window must be set less than the width of the peaks. The peak widths are given in the second column from right, lower left table in Fig. 1.

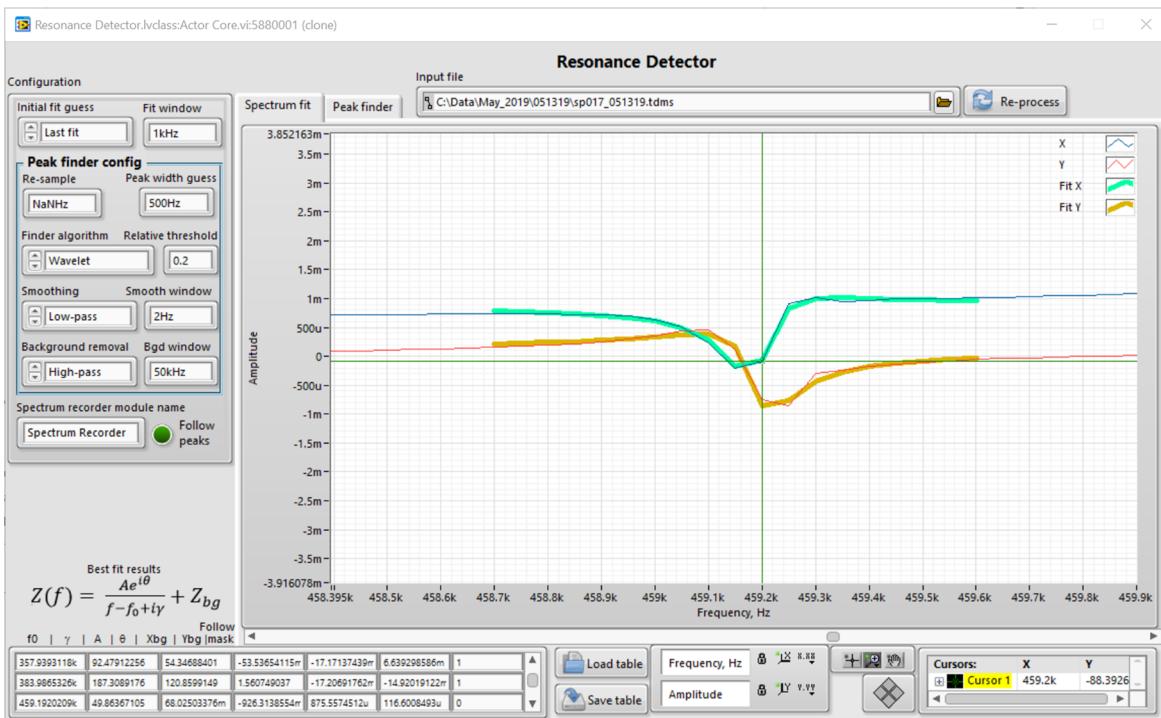
Once only actual resonances are identified, open the “Format Resonances” window from the main window and click on the folder icon on the input file top bar and navigate to C:\Data\month\date to find the stored scans that were just made. Reprocess the date-stamped spXXXX.tdms file to generate a rusin.dat file for use with the analysis codes in the directory C:\Data\month\date. Open the rusin.dat file (in Notepad) and now manually enter any peaks not found by the peak finder (Fig. 12).

Choose either the RPR or cylinder codes from the analysis directory. Files for cylinders are identified with cyl, and those for rectangle parallelepipeds are identified with rpr. Copy and paste the frequency list from the rusin.dat file into the appropriate \*\*\*.in.dat file where \*\*\* is either cyl or rpr. Enter the number of moduli to fit (2,3,5,6,9 for isotropic, cubic, hexagonal, tetragonal, orthorhombic), order of polynomials to approximate the mode shapes (12 for cyl and 14 for RPR are usually adequate, although shapes with aspect ratios far from unity will require higher order polynomials), number of modes to compute, mass of sample, convergence parameter (set it to 1; we do not document this feature), number of iterations, number of modes to include in fit, guess for Poisson’s ratio if autoguess is to be used, moduli guesses, and sample dimensions. Detailed instructions for the analysis codes are included in the code download. Save and close the \*\*\*.in.dat file. Double click on the appropriate \*\*\*.exe file and enter the desired number of iterations to start the fit, the result of which will be output in the \*\*\*.out.dat file. The number of iterations should be entered as 2 if a nonzero value for Poisson’s ratio was entered in the \*\*\*.in.dat file, activating the autoguess feature, but if no guess (0.0) was entered, then 1 iteration will simply compute mode frequencies based on the initial values. This first fit pass will generate a list of calculated resonances that are compared by the user line by line to the list of measured resonances. A percentage error between measured and calculated resonances is generated for each resonance line entry. This is a measure of the mismatch between the calculated and measured frequencies. Large values will indicate where missing modes have occurred.

The user should reposition the sample and rescan over the frequency range where the modes were missed if desired. Very often these are a component of a doublet, where one component may be weak and hidden in the tail of the stronger component. Missing modes can be found using the procedure of Sec. V B. Run the codes again with a greater number of iterations to allow convergence to a minimum. The \*\*\*.out.dat file will give the calculated values for the specimen’s bulk modulus, shear modulus, Young’s modulus, and Poisson’s ratio as well as compressional and shear wave speeds.



**FIG. 11.** A 1 V resonance fit for the same scan as the 500  $\mu$ V resonance of Fig. 12. The peak detector threshold captured this one.



**FIG. 12.** A 500  $\mu$ V resonance fit for the same scan as the 1 V resonance of Fig. 11. The peak detector threshold was set too high to capture this one. Using a code option, the approximate location was entered, forcing the Lorentzian peak fit routine to fit this one.

The quality of the fit and thus the confidence in the elastic moduli values are *not* assessed by the size of the rms error. The actual error determination is discussed below. The rms error is more a measure of accurate geometry, symmetry, and homogeneity.

### B. Using the open source cylinder and rectangular parallelepiped analysis codes to find missing modes and obtain moduli

There are two codes for analysis of RUS measurements. One is for cylinders and the other is for RPRs. Both<sup>4</sup> are freely available upon request in the executable form to run on Windows 10<sup>TM</sup>. There are several points to keep in mind when using these codes.

The most important is to be sure that the set of input frequencies is complete. It is very common for the acquisition code to miss resonances. This is because the combination of transducer surface motion and the specimen node pattern can combine to produce weak or absent response at the transducer or because two resonances are degenerate or so close to each other that they cannot be separated. Note that multiple detect transducers connected to the same input only make things worse. This is because there are many more places where one transducer can cancel the other than there are nodes. The electronics has a very large dynamic range so that it is possible in a single scan to record resonances whose amplitude ranges from so large that the microcomputer A/D saturates (about 1 V) to submillivolt signals. The phase sensitive electronics provides the best possible information to the data acquisition fitting

algorithms to minimize near-overlap misses, but modes still can be missed. If a mode is missed and not accounted for in the input file, then a nonphysical set is fed to the analysis codes. That is, the set of resonances does not correspond to any possible set of elastic moduli. To fix this, the code input files have provision to enter a “zero” as a missing mode placeholder. In the section of the input file in the introduction, which is part of the input file to the cylinder code for a well-prepared Gd specimen, after a bit of guided trial and error, of the 40 modes used to obtain moduli to about 0.1% accuracy, modes 3 and 6 (and some others) were missing and had to be entered by the user. Part of the problem here is that this was a very round, very homogeneous specimen. Therefore, modes 2 and 3, and 5 and 6 were doublets (degenerate) and so were expected to be missed. That is, there are two peaks almost perfectly overlapping at these frequencies. Another important point is that the codes will always converge and that there are multiple minima, but the values will not be correct if a nonoptimal local minimum is found. A useful data set with accurate results with a reasonably well prepared specimen that is reasonably homogeneous will produce an rms error [the square root of the average of the square of the differences between computed and measured resonances, modified by the number of degrees of freedom (the number of elastic moduli)] better than 1%. An exceptional specimen such as an oriented single crystal of Si will produce rms errors of order 0.03% or better.

To explain the code workings, we examine the Gd output file (Fig. 13). Then, we discuss the input file and how to actually run the code. Note that the RPR and cylinder codes use the same format

```
www.rusalamocreek.com ver. 11.0 units cm, grams, GPa, MHz
Polycrystal Gd 305.15K
using 12 order polynomials mass= 0.2910 gm rho= 7.868 gm/cc
  n   fexp(MHz)   fcalc(MHz) %err wt   k   i      dlnf/dlnc11 dlnf/dlnc44
  1   0.1963020   0.1957102 -0.30 1.00  4   1      0.00000   1.00000
  2   0.2284422   0.2278866 -0.24 1.00  1   2      0.22962   0.77038
  3   0.0000000   0.2278866  0.00 0.00  7   2      0.22962   0.77038
...
  35  0.5791929   0.5799109  0.12 1.00  4   4      0.04637   0.95363
  36  0.5856058   0.5869318  0.23 1.00  5   6      0.08368   0.91632
  37  0.5867641   0.5869318  0.03 1.00  3   4      0.08368   0.91632
  38  0.0000000   0.5871306  0.00 0.00  4   5      0.00000   1.00000
  39  0.5887240   0.5893978  0.11 1.00  7   7      0.04556   0.95444
  40  0.5901877   0.5893978 -0.13 1.00  1   7      0.04556   0.95444
  Bulk Modulus(GPa)= 38.4707
Youngs modulus(GPa) = 56.5709 Shear modulus= 22.5397 Poissons ratio = 0.255
This is an isotropic system
c11(GPa)= 68.52360
c44(GPa)= 22.53969
  d1(cm)   d2(cm)   d3(cm)
  0.33000  0.33000  0.43240
loop# 2 rms error= 0.1451 %, changed by -.0000490 %
rms error increased 2% by the following % changes in
  c11   c44
  0.19  0.05
V(compressional)(km/s)= 2.95104
V(shear)(km/s)= 1.69250
```

FIG. 13. The final fit to the Gd cylinder of Fig. 1 showing the material properties computed by the code.

1	0.1963020	0.1957194	-0.30	1.00	4	1	0.00000	1.00000
2	0.2284422	0.2278913	-0.24	1.00	1	2	0.22976	0.77024
3	0.3002769	0.2278913	-24.11	1.00	7	2	0.22976	0.77024
4	0.3094602	0.3003376	-2.95	1.00	5	1	0.13709	0.86291
5	0.3381673	0.3090599	-8.61	1.00	8	1	0.05505	0.94495
6	0.3387496	0.3090599	-8.76	1.00	2	1	0.05505	0.94495
7	0.3611121	0.3382503	-6.33	1.00	6	2	0.00607	0.99393
8	0.3619525	0.3382504	-6.55	1.00	4	2	0.00607	0.99393

FIG. 14. Code output snippet of data from Fig. 1 with two missed modes after a single pass of the fitting code.

for input and output files, so we only discuss the cylinder code. The primary task at this point is to find missing modes. Figure 13 is the output of a good fit.

Let us go through the missing mode problem in some detail as this is the primary cause of failure for RUS. Finding missing modes involves running the acquisition code and the analysis code iteratively, so keep both easily accessible. Figure 13 is our goal. As we proceed, we show snippets of the input and output files without and with missed modes identified.

In Fig. 14 is a section of output with two missed modes after a single iteration with good initial guesses. In Fig. 15 is the (snippet) of the cylin.dat file that produced it.

Note the error of 24.11% for the third mode. However, it is easy to see that from Fig. 14 if the actual third mode was missed, the measured frequency of 0.300 276 9 MHz would be moved down in the list and sit nicely next to the computed frequency of 0.300 337 6 MHz. Now, we insert a missing mode into the input file, so it becomes Fig. 16 (again, just a snippet) where we entered 0 0 0 (note that no delimiter or decimal point is needed).

However, there is another problem now revealed at mode 6 (Fig. 17). Moving mode 6 down looks like it will match up with the computed mode there, so we add a second missed mode (Fig. 18).

Now, we have very small errors between measured and computed modes (Fig. 19). As we attempt to indicate, it is useful to keep running single iterations of the analysis codes and adding missed modes one at a time until no more obvious missed modes are apparent. At this point, run the code with several iterations while watching the on-screen rms error behavior. For these data, the code converges to an rms error of 0.1451% and suggests missed modes very near 0.227 891 3 MHz and 0.309 059 9 MHz.

If the fit is good, one can quit, but it is often revealing to find the missed modes. To attempt to find them, reposition the sample and rescan over the frequency range where the mode was missed. Very

often missed modes are one of a doublet, where one mode may be weak and hidden in the tail of the stronger resonance. Rotating and repositioning can change the relative amplitudes of the components of a doublet. Missing modes can be entered manually into \*\*\*.in.dat via Notepad after using the “Spectrum recorder” window and cursor to read the missed frequencies or using the “follow peaks” feature and manually entering an approximate guess (Fig. 12). Again, delimiters are not needed, but the third number (the weight used in the fit) should now be 1. If the initial guesses are far from the actual moduli, this process will be more difficult, with a bit more trial and error involved, including changing the moduli to get the first mode computed close to the first mode measured. The autoguess feature does this automatically if the first mode is pure shear, as it often is, and is in this case as seen by the 1.0000 at the top of the far right column in Fig. 13. The numbers in the rightmost columns are the computed sensitivity of the mode to a modulus. For example, the fractional dependence of the 3rd mode on c11 is 0.229 76 and, on c44, 0.770 24. Now, run the code again with a sufficient number of iterations so that it converges and exits. The \*\*\*.out.dat file will contain the best-fit values for the bulk modulus, shear modulus, Young’s modulus, and Poisson’s ratio as well as compressional and shear wave speeds, with some quantities only defined and therefore computed for isotropic materials. The quality of the specimen and thus the confidence in the elastic moduli values are assessed by the size of the rms error. However, the actual error determination is more complex.

The best fit is the bottom of an n-dimensional bowl where n is the number of elastic moduli fitted. The principle axes of the bowl are not along the elastic moduli axes. To determine the errors, we have measured many specimens of identical isotropic composition and for many types of materials where we had determined the dimensions to of order 0.01%. Using the measured scatter in c11,

0.1963020	0.1957193	1.00
0.2284422	0.2278911	1.00
0.3002769	0.3003374	1.00
0.3094602	0.3090597	1.00
0.3381673	0.3382501	1.00
0.3387496	0.3382502	1.00

FIG. 15. Code input snippet of data from Fig. 1 with two missed modes.

0.1963020	0.1957193	1.00
0.2284422	0.2278911	1.00
0 0 0		
0.3002769	0.3003374	1.00
0.3094602	0.3090597	1.00
0.3381673	0.3382501	1.00
0.3387496	0.3382502	1.00

FIG. 16. Code input snippet of data from Fig. 1 with the first missed mode replaced with 0 0 0.

1	0.1963020	0.1957194	-0.30	1.00	4	1	0.00000	1.00000
2	0.2284422	0.2278913	-0.24	1.00	1	2	0.22976	0.77024
3	0.0000000	0.2278913	0.00	0.00	7	2	0.22976	0.77024
4	0.3002769	0.3003376	0.02	1.00	5	1	0.13709	0.86291
5	0.3094602	0.3090599	-0.13	1.00	8	1	0.05505	0.94495
6	0.3381673	0.3090599	-8.61	1.00	2	1	0.05505	0.94495
7	0.3387496	0.3382503	-0.15	1.00	6	2	0.00607	0.99393
8	0.3611121	0.3382504	-6.33	1.00	4	2	0.00607	0.99393

FIG. 17. Code output snippet of data from Fig. 1 with first missed mode replaced with 0 0 0.

we determined, using the curvature of each principal axis, how far up the bowl we had to move the best-fit c11 to get the measured spread. Because the preponderance of the lowest modes (even the 40 modes we measured here for the Gd sample) is more strongly dependent on c44 than c11, the precision (or the curvature of the c44 bowl axis) approaches the accuracy with which we can determine geometry. Therefore, we based the error estimate instead on c11 and then used the same criteria on the other moduli. That is, we found that if we moved up the bowl along the c11 axis by some certain amount to get a change in rms error by 2%, we got about the independently determined scatter in c11. We then used that same 2% change in rms error for the other moduli to determine how much they changed. The result is shown in Fig. 20.

We see that the error we estimate for c11 is 0.19%, quite a bit larger than the error of 0.05% for c44. Neither error is the same as the rms error. Thus, it is very important to use the fit-curvature-determined errors in reporting results.

## VI. COMPUTATION OF MODULI FROM RESONANCES AND THE PHYSICS OF THE AUTOGUESS FUNCTION

### A. The math needed to understand how to compute resonances from moduli (the “forward computation”)

The following is a description of how to compute the resonance frequencies of a free-surface rigid solid body with no dissipation (in practice, this means  $Q >$  several hundred) from its elastic tensor, geometry, and density, based on the process detailed by Visscher.<sup>11,12</sup>

0.1963020	0.1957193	1.00
0.2284422	0.2278911	1.00
0 0 0		
0.3002769	0.3003374	1.00
0.3094602	0.3090597	1.00
0 0 0		
0.3381673	0.3382501	1.00
0.3387496	0.3382502	1.00
0.3611121	0.3613178	1.00
0.3619525	0.3613179	1.00

FIG. 18. Code input snippet of data from Fig. 1 with both missed modes replaced with 0 0 0.

This is known as the “forward computation,” which was originally introduced by Demarest.<sup>13,14</sup>

The linear elastic Lagrangian as a function of the displacement field  $\mathbf{u}(\mathbf{r}, t) = \sum_i^3 u_i(\mathbf{r}, t)\hat{\mathbf{e}}_i$ , where we are looking at the motion of a tiny element in the solid, is

$$L = T - V = \frac{1}{2} \int_V \left( \sum_i \rho \dot{u}_i^2 - \sum_{ijkl} C_{ijkl} \frac{\partial u_i}{\partial x_j} \frac{\partial u_k}{\partial x_l} \right) dV, \quad (1)$$

which, provided the harmonic displacement field

$$u_m(\mathbf{r}, t) = u_m(\mathbf{r}) e^{-i\omega t} \quad (2)$$

may be expressed as

$$L = \frac{1}{2} \int_V \left( \sum_i \rho \omega^2 u_i^2 - \sum_{ijkl} C_{ijkl} \frac{\partial u_i}{\partial x_j} \frac{\partial u_k}{\partial x_l} \right) dV \quad (3)$$

with  $u_i \equiv u_i(\mathbf{r})$ .  $u_i(\mathbf{r})$  is then expanded in the Visscher basis,

$$u_i(\mathbf{r}) = \sum_{\lambda}^N \alpha_{i\lambda} \phi_{\lambda}(\mathbf{r}), \quad (4)$$

where  $a_{i\lambda}$  are scalar constants and  $\phi_{\lambda}(\mathbf{r})$  is

$$\phi_{\lambda}(\mathbf{r}) \triangleq x^l y^m z^n \equiv \phi_{lmn}. \quad (5)$$

Expressed in the Visscher basis, the kinetic and potential energy terms in Eq. (3) become

$$\sum_i \rho \omega^2 u_i^2 = \sum_{ii'\lambda\lambda'} \delta_{ii'} \rho \omega^2 a_{i\lambda} a_{i'\lambda'} \phi_{\lambda} \phi_{\lambda'}, \quad (6)$$

and

$$\sum_{ijkl} C_{ijkl} \frac{\partial u_i}{\partial x_j} \frac{\partial u_k}{\partial x_l} \stackrel{k \rightarrow i'}{=} \sum_{jj'} C_{ijj'j'} a_{i\lambda} a_{i'\lambda'} \frac{\partial \phi_{\lambda}}{\partial x_j} \frac{\partial \phi_{\lambda'}}{\partial x_{j'}}, \quad (7)$$

respectively, and Eq. (3) becomes

$$L = \frac{1}{2} \omega^2 \sum_{ii'\lambda\lambda'} a_{i\lambda} \left( \delta_{ii'} \int_V \phi_{\lambda} \rho \phi_{\lambda'} dV \right) a_{i'\lambda'} - \frac{1}{2} \sum_{ii'\lambda\lambda'} a_{i\lambda} \left( \sum_{jj'} C_{ijj'j'} \int_V \frac{\partial \phi_{\lambda}}{\partial x_j} \frac{\partial \phi_{\lambda'}}{\partial x_{j'}} dV \right) a_{i'\lambda'}. \quad (8)$$

It is typical at this point to make the following definitions:

$$E_{\lambda i\lambda' i'} \triangleq \delta_{ii'} \int_V \phi_{\lambda} \rho \phi_{\lambda'} dV, \quad (9)$$

1	0.1963020	0.1957194	-0.30	1.00	4	1	0.00000	1.00000
2	0.2284422	0.2278913	-0.24	1.00	1	2	0.22976	0.77024
3	0.0000000	0.2278913	0.00	0.00	7	2	0.22976	0.77024
4	0.3002769	0.3003376	0.02	1.00	5	1	0.13709	0.86291
5	0.3094602	0.3090599	-0.13	1.00	8	1	0.05505	0.94495
6	0.0000000	0.3090599	0.00	0.00	2	1	0.05505	0.94495
7	0.3381673	0.3382503	0.02	1.00	6	2	0.00607	0.99393
8	0.3387496	0.3382504	-0.15	1.00	4	2	0.00607	0.99393
9	0.3611121	0.3613180	0.06	1.00	5	2	0.06450	0.93550

FIG. 19. Code output snippet with both missed modes replaced with 0 0 0 to produce the final fit of Fig. 13.

$$\Gamma_{\lambda i \lambda' i'} \triangleq \sum_{jj'} C_{ijj'i'} \int_V \frac{\partial \phi_\lambda}{\partial x_j} \frac{\partial \phi_{\lambda'}}{\partial x_{j'}} dV, \quad (10)$$

to get

$$L = \frac{1}{2} \omega^2 \sum_{ii' \lambda \lambda'} a_{i\lambda} E_{\lambda i \lambda' i'} a_{i' \lambda'} - \frac{1}{2} \sum_{ii' \lambda \lambda'} a_{i\lambda} \Gamma_{\lambda i \lambda' i'} a_{i' \lambda'}. \quad (11)$$

Equation (11) is stationary when

$$\frac{\partial L}{\partial a_{i\lambda}} = 0 \quad (12)$$

as demonstrated by Visscher.<sup>5</sup> It can be shown that application of this condition on Eq. (11) produces the following:

$$\begin{aligned} \frac{\partial L}{\partial a_{i'' \lambda''}} &= \sum_{i'' \lambda'} a_{i'' \lambda'} \left( \frac{1}{2} \omega^2 E_{\lambda'' i'' \lambda' i'} \right) + \sum_{i\lambda} \alpha_{i\lambda} \left( \frac{1}{2} \omega^2 E_{\lambda i \lambda'' i''} \right) \\ &\quad - \sum_{i'' \lambda'} a_{i'' \lambda'} \left( \frac{1}{2} \Gamma_{\lambda'' i'' \lambda' i'} \right) - \sum_{i\lambda} \alpha_{i\lambda} \left( \frac{1}{2} \Gamma_{\lambda i \lambda'' i''} \right), \end{aligned} \quad (13)$$

which can be simplified into

$$\omega^2 \sum_{i\lambda} \alpha_{i\lambda} E_{\lambda i \lambda'' i''} = \sum_{i\lambda} \alpha_{i\lambda} \Gamma_{\lambda i \lambda'' i''} \quad (14)$$

by exploiting the symmetries  $E_{\lambda i \lambda''} = E_{\lambda'' i \lambda}$  and  $\Gamma_{\lambda i \lambda''} = \Gamma_{\lambda'' i \lambda}$ .

Equation (14) is the root relationship between elastic moduli and resonance frequencies upon which RUS is founded, enabling one to compute resonances from elastic properties by solving this eigenvalue problem.

Typically, Eq. (14) is solved in a lower-dimensional framework, with all rank-4 and rank-2 quantities expressed as rank-2 (matrix) and rank-1 (vector) quantities, respectively. This is accomplished by mapping positions of elements in the higher-dimensional quantities to new positions in the lower dimensional ones.

A general map for Eq. (14) that achieves the desired results (noting that the symmetry  $\Lambda_{\lambda'' i'' \lambda i} = \Lambda_{\lambda i \lambda'' i''}$  has been assumed because the system is a physical, time-reversal invariant positive-definite one)

$$(a_{\beta(i,\lambda)} \in \mathbf{a}) = (a_{i\lambda} \in \mathbf{a}), \quad (15)$$

```
loop# 2 rms error= 0.1451 %, changed by -.0000490 %
rms error increased 2% by the following % changes in
  c11   c44
  0.19  0.05
```

FIG. 20. Code output snippet showing error listing for the final fit of Fig. 13.

$$(\Lambda_{\alpha(\lambda'', i'')} \beta(\lambda, i) \in \underline{\Lambda}) = (\Lambda_{\lambda'' i'' \lambda i} \in \tilde{\Lambda}), \quad (16)$$

can be defined by the functions

$$\alpha = (i'' - 1)N + \lambda'' \forall i'' \in (1, 2, 3), \lambda'' \in (1, 2, 3, \dots, N), \quad (17)$$

$$\beta = (i - 1)N + \lambda \forall i \in (1, 2, 3), \lambda \in (1, 2, 3, \dots, N), \quad (18)$$

to write Eq. (14) as

$$\omega^2 \sum_{\beta=1}^{3N} E_{\alpha\beta} a_{\beta} = \sum_{\beta=1}^{3N} \Gamma_{\alpha\beta} a_{\beta}, \quad (19)$$

where  $N$  is the number of terms out to which the displacement field  $u_i(\mathbf{r})$  is expanded in the Visscher basis (4). Equation (19) can be expressed as

$$\omega^2 \underline{\mathbf{E}} \mathbf{a} = \underline{\mathbf{\Gamma}} \mathbf{a}, \quad (20)$$

which is the familiar RUS eigenvalue problem that enables computing resonances from an elastic modulus tensor, the forward computation.

## B. Process to compute moduli from resonances (the “inverse computation”)

Within the context of RUS, the process by which a solid’s elastic properties are computed from its measured mechanical resonances always involves an inversion of the forward computation and is as such generally referred to as the “inverse computation.”

Because the forward computation has no analytic inverse, it must be inverted numerically. An iterative approach is taken, which consists of repeated applications of the forward computation on a varying elastic tensor until a set of resonances are computed that are a best fit to the corresponding measured set to within some predefined threshold. The elastic tensor that achieves this result is taken as that of the solid from which the resonance frequencies were measured.

This process is known as *fitting* the elastic tensor. It requires an objective function—some meaningful measure of similarity between two sets of numbers (specifically, the computed and measured resonance frequencies) and some means of varying the elastic tensor to minimize the objective function (so that the computed resonances tend toward the measured ones as the computation proceeds).

The objective function and means of varying the elastic tensor are arbitrary. We use as the objective function a computed sum of

weighted residuals, and a common approach to varying the elastic tensor is by the Levenberg–Marquardt method of steepest descent,<sup>15</sup> but genetic algorithms<sup>16</sup> have also been used to the same effect with the added benefit of determining missing mode placeholders but at the expense of massive computational time. We do not discuss this approach.

### C. Autoguess strategy

The “autoguess strategy” refers to a routine that does a one-time modification of the initial input estimated elastic properties, based on the assumption that the first measured resonance frequency is a pure shear mode, to provide the inverse computation with a starting point from which it is more likely to converge.

The physical basis for the autoguess assumption that the first resonance is pure shear (i.e., dependent entirely on  $C_{44}$ ) is derived from the tendency of  $C_{44}$  to be the softest of any of a material’s stiffnesses, generally so because shear manipulation involves no volume change (does not change all bond lengths). Each of a material’s resonance frequencies is dependent on a mix of all its elastic moduli, so because frequency increases with stiffness, it is plausible that the lowest resonance is more dependent on the softest stiffness (modulus) than any other.

By assuming a functional relationship between only the first measured resonance and the shear modulus  $C_{44}$ , the autoguess routine is capable of updating a whole set of input estimated elastic moduli through orthorhombic materials for the inverse computation to act on. This function is implemented, and the method for using it is described in the analysis code<sup>4</sup> distribution.

### D. More on missing modes

A consequence of inverting the forward computation numerically is the loss of physical context. There exists no mechanism within the inverse computation that checks its inputs or outputs for correspondence with a real, physical system; it simply acts blindly on nominally valid data.

This is a problem because it is not unusual for the set of resonances found to be physically invalid, meaning no real material with quality and dimensions as required by RUS could have produced it. The inverse computation applied to such a spectrum will produce correspondingly meaningless elastic moduli.

If there are no issues with specimen quality or dimensions, the way in which a measured spectrum can be nonphysical is by being incomplete; that is, not all modes that should be present within the frequency range over which the spectrum was acquired were captured. For example, suppose a measured spectrum captures the mode  $f_i$ , followed by  $f_{i+1}$ , but the spectrum that physically corresponds to the specimen from which these modes were extracted contains some unknown mode  $f_q$ , *missed* during measurement, that exists between  $f_i$  and  $f_{i+1}$ .

Such an incomplete spectrum must be prepared appropriately for fitting by the inverse computation. This requires locating the positions (identified by and referred to as zeros) of the missing

modes in the sequence of measured frequencies—knowing their values are not necessary. Generally, the user performs this task manually, a process that takes experience and time and is prone to error. Recently though, an algorithm for automatically locating missing modes in incomplete spectra was tested in a code for performing the inverse computation and showed promise for eliminating this job from the user’s responsibilities.

This automated solution to the missing mode problem takes a brute-force approach to locating missed resonances in measured spectra. Based on the rms error between *every* arrangement of  $n$  or fewer zeros in the measured spectrum and the spectrum computed for that iteration of the inverse computation, the positions of the missing modes are taken from that configuration which yields the smallest error. These positions are, of course, subject to change with each iteration of the inverse computation. This feature will be included in future updates to the analysis codes and made freely available.<sup>4</sup>

### ACKNOWLEDGMENTS

Work at Los Alamos National Laboratory (LANL) was supported by National Science Foundation Grant No. DMR-1644779, by the U.S. Department of Energy Los Alamos National Laboratory Science Campaigns 1 and 2, and by the State of Florida.

### REFERENCES

- <sup>1</sup>A. Migliori, *Acoust. Today* **4**, 17 (2008).
- <sup>2</sup>WEST SYSTEM®, <https://www.westsystem.com/>, #105 resin, #209 hardener.
- <sup>3</sup>Physik Instruments®, <https://www.physikinstrumente.com/en/products/piezoelectric-transducers-actuators/>, PZT part # disc c255 o5 t2 wsCuNi.
- <sup>4</sup>Alamo Creek Engineering, 13 Alamo Creek Drive, Santa Fe, NM 87506, <http://www.rusalamocreek.com/>.
- <sup>5</sup>A. Migliori, J. Sarrao, W. M. Visscher, T. Bell, M. Lei, Z. Fisk, and R. Leisure, *Physica B* **183**, 1 (1993).
- <sup>6</sup>F. Balakirev, LANL RUS data acq package, <https://github.com/ffb-LANL/resonant-ultrasound/releases>, 2019.
- <sup>7</sup>PCB123®, <https://www.sunstone.com/pcb-resources/cad-tools/pcb123>, Sunstone Circuits.
- <sup>8</sup>Red Pitaya, <https://www.redpitaya.com/>, 2018.
- <sup>9</sup>P. Demin, Red Pitaya Notes, <https://pavel-demin.github.io/red-pitaya-notes>, 2018.
- <sup>10</sup>Vivado™, <https://www.xilinx.com/products/design-tools/vivado.html>, XILINX™.
- <sup>11</sup>W. M. Visscher, A. Migliori, T. M. Bell, and R. A. Reinert, *J. Acoust. Soc. Am.* **90**, 2154 (1991).
- <sup>12</sup>A. Migliori and J. L. Sarrao, *Resonant Ultrasound Spectroscopy: Applications to Physics, Materials Measurements, and Nondestructive Evaluation* (Wiley-Interscience, 1997).
- <sup>13</sup>H. H. Demarest, “The cube resonance method of elastic constant determination,” Ph.D. thesis, Reed College, 1969.
- <sup>14</sup>H. H. Demarest, Jr., *J. Acoust. Soc. Am.* **49**, 768 (1971).
- <sup>15</sup>K. Levenberg, *Q. Appl. Math.* **2**, 164 (1944).
- <sup>16</sup>B. Ramshaw, A. Shekhter, R. D. McDonald, J. B. Betts, J. Mitchell, P. Tobash, C. Mielke, E. Bauer, and A. Migliori, *Proc. Natl. Acad. Sci. U. S. A.* **112**, 3285 (2015).