ONE CAN HEAR THE COMPOSITION OF A STRING: EXPERIMENTS WITH AN INVERSE EIGENVALUE PROBLEM*

STEVEN J. COX†, MARK EMBREE†, AND JEFFREY M. HOKANSON†

28 June 2010

Abstract. To what extent do the vibrations of a mechanical system reveal its composition? Despite innumerable applications and mathematical elegance, this question often slips through those cracks that separate courses in mechanics, differential equations, and linear algebra. We address this omission by detailing a classical finite dimensional example: the use of frequencies of vibration to recover positions and masses of beads vibrating on a string. First we derive the equations of motion, then compare the eigenvalues of the resulting linearized model against vibration data measured from our laboratory's monochord. More challenging is the recovery of masses and positions of the beads from spectral data, a problem for which a variety of elegant algorithms exist. After presenting one such method based on orthogonal polynomials in a manner suitable for advanced undergraduates, we confirm its efficacy through physical experiment. We encourage readers to conduct their own explorations using the numerous data sets we provide.

Key words. beaded string, inverse eigenvalue problem, vibration, continued fractions

AMS subject classifications. 34C10, 47B36, 65F18, 70J10

1. Introduction. The 18th century witnessed revolutionary progress in the mathematical description of fundamental problems in mechanics, thanks to the collective efforts of natural philosophers such as Leonhard Euler, the Bernoulli family, d'Alembert, Lagrange, and others [5, 24]. These old masters developed predictive models: given the material properties of a system, along with its position and velocity at some initial time, determine the system's state at all future times. While such *forward models* give great insight, modern applications often present the problem backwards: We can measure how a system responds to some stimulus, and from such experiments seek to discover the system's composition. In many instances the response is an acoustic signature and so one is led to pose the backward, or *inverse*, problem in the form of the question, "Can one hear?" For example, Gopinath and Sondhi [13] ask "Can one hear the shape of your throat?", Kac [15] asks "Can one hear the shape of a drum?", Sekii and Shibahashi [20] ask "Can one hear into the sun?", Lin [18] asks "Can one hear a crack in a beam?", and Gutkin and Smilansky [14] ask "Can one hear the shape of a graph?" Each of these investigations seeks to echo the success of Borg [2], Levinson [17], and Gelfand and Levitan [11] in their various proofs that one can hear a "potential," and Krein's demonstration that one can hear the mass distribution of a nonuniform string [16].

Krein's argument, as developed by Dym and McKean [9], proceeds from the beaded string (a massless thread supporting a finite number of point masses) to the general nonhomogeneous distribution of mass. With Gantmacher [10], Krein returned to the beaded case, resurrected the lovely work of Stieltjes [21] on continued fractions, and carefully developed the requisite matrix analysis and complex function theory. This finite dimensional case has since been systematized, and numerous related algorithms now exist for its resolution [6]. Although these methods supply constructive

*Supported by National Science Foundation grants DMS-0240058 (VIGRE), DMS-0505893, and DMS-CAREER-0449973.

†Department of Computational and Applied Mathematics, Rice University, 6100 Main Street—MS 134, Houston, Texas 77005–1892 (cox@rice.edu, embree@rice.edu, jeffreyh@rice.edu).

means for determining masses and lengths from spectral data, this theory has, to our knowledge, remained untested. We here provide experimental confirmation on data taken from our own laboratory. Our larger aim however is to give an entry to inverse spectral theory, numerics and experiment that is accessible to students possessing a solid undergraduate background in linear algebra. Indeed, beaded string experiments form the culminating exercises in an optional one-credit physical laboratory that accompanies our junior-level Matrix Analysis course at Rice University.

The remainder of our tour is organized as follows. In §2 we derive a system of differential equations for the displacement of a plucked beaded string and show how to express the solution in terms of eigenvalues and eigenvectors of underlying mass and stiffness matrices. In §3 we introduce our experimental setup and explain how we measure the eigenvalues of a beaded string. In §4 we combine algorithms of de Boor and Golub [8] and Gladwell [12, §4.4] to determine each bead's location and mass from two sets of eigenvalues, and in §5 we confirm, using real data, that one set of eigenvalues suffices to reveal the location and masses of symmetrically placed beads.

2. The Forward Problem. We thread a massless string through n beads, apply a known tension τ, and clamp its ends at a known distance, ℓ, apart. With reference to Figure 2.1 we denote the mass of the jth bead by m_j, and let ℓ_j denote the length between mass m_j and m_{j+1} (with ℓ_0 and ℓ_n denoting the length between the beads at each end and the clamped support). Following a small vertical pluck we presume that the jth bead suffers the planar displacement (x_j, y_j) and that the jth segment makes the angle ϕ_j with the horizontal.

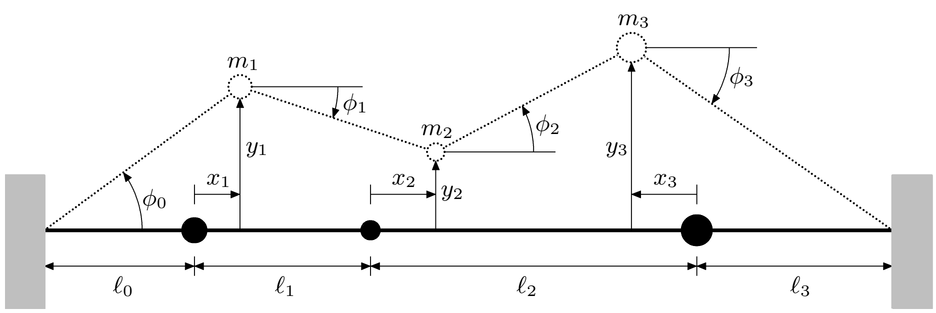


FIG. 2.1. *A string with $n = 3$ beads at rest (solid black) and in a deformed state (dotted).*

In this state the horizontal and vertical components of the string restoring forces at the jth mass are

$$\tau(\cos\phi_j - \cos\phi_{j-1}) \quad \text{and} \quad \tau(\sin\phi_j - \sin\phi_{j-1})$$

respectively. The angles ϕ_{j-1} and ϕ_j can be determined from the horizontal and vertical displacements. As evident from Figure 2.1,

$$\cos\phi_j = \frac{\ell_j + (x_{j+1} - x_j)}{\sqrt{(\ell_j + (x_{j+1} - x_j))^2 + (y_{j+1} - y_j)^2}}$$

$$\sin\phi_j = \frac{y_{j+1} - y_j}{\sqrt{(\ell_j + (x_{j+1} - x_j))^2 + (y_{j+1} - y_j)^2}},$$

and similarly for ϕ_{j-1}. If the pluck is small and the string is taut we may make the (customary) assumptions that $|y_{j+1} - y_j| \ll \ell_j$ and $|x_{j+1} - x_j| \ll \ell_j$ (for a careful alternative, see [1]), and so arrive at the approximations

$$\cos\phi_j \approx 1, \qquad \sin\phi_j \approx \frac{y_{j+1}-y_j}{\ell_j},$$
$$\cos\phi_{j-1} \approx 1, \quad \sin\phi_{j-1} \approx \frac{y_j - y_{j-1}}{\ell_{j-1}}.$$

With these approximations the horizontal forces balance, so it remains only to balance the vertical restoring forces with their associated inertial terms ("ma = F"):

$$m_j y_j''(t) = \tau\Big(\frac{y_{j+1}(t)-y_j(t)}{\ell_j} - \frac{y_j(t)-y_{j-1}(t)}{\ell_{j-1}}\Big), \qquad j = 1,\ldots,n.$$

For this equation to hold at the first and last mass ($j=1$ and $j=n$), we define $y_0 = y_{n+1} = 0$, thus describing the fixed ends of the string. As a consequence of our first-order approximations to the sines and cosines, these n coupled equations are *linear*, and can be most conveniently organized into the matrix form

$$\mathbf{M}\mathbf{y}''(t) = -\mathbf{K}\mathbf{y}(t), \tag{2.1}$$

with *state vector* $\mathbf{y}(t)$, *mass matrix* $\mathbf{M}$,

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_n(t) \end{bmatrix}, \qquad \mathbf{M} = \begin{bmatrix} m_1 & & & \\ & m_2 & & \\ & & \ddots & \\ & & & m_n \end{bmatrix},$$

and *stiffness matrix*

$$\mathbf{K} = \tau\begin{bmatrix} \ell_0^{-1}+\ell_1^{-1} & -\ell_1^{-1} & & \\ -\ell_1^{-1} & \ell_1^{-1}+\ell_2^{-1} & \ddots & \\ & \ddots & \ddots & -\ell_{n-1}^{-1} \\ & & -\ell_{n-1}^{-1} & \ell_{n-1}^{-1}+\ell_n^{-1} \end{bmatrix}, \tag{2.2}$$

with all unspecified entries equal to zero. The mass and stiffness matrices enjoy two lovely properties: they are *symmetric*, $\mathbf{M} = \mathbf{M}^T$ and $\mathbf{K} = \mathbf{K}^T$, and *positive definite*, meaning that

$$\mathbf{y}^T\mathbf{M}\mathbf{y} = \sum_{j=1}^{n} m_j y_j^2 \quad \text{and} \quad \mathbf{y}^T\mathbf{K}\mathbf{y} = \frac{y_1^2}{\ell_0} + \frac{y_n^2}{\ell_n} + \sum_{j=1}^{n-1}\frac{(y_j-y_{j+1})^2}{\ell_j}$$

are both positive for every nonzero vector $\mathbf{y} \in \mathbb{R}^n$.*

*If the masses and lengths are uniform, say $m_j = 1/(n+1)$ and $\ell_j = 1/(n+1)$, and $\tau = 1$, then $\mathbf{M} = \mathbf{I}$ and $\mathbf{K}$ becomes the familiar tridiagonal matrix that arises from the second-order finite difference discretization of the second spatial derivative; see, e.g., [23]. This reflects one way in which the wave equation $u_{tt} = u_{xx}$, can be derived as the limit of lumped masses.

We are interested in the motion induced by an initial pluck of the string, whereby the masses are vertically displaced by the components of the vector $\mathbf{y}_0$, then released. Thus we presume that

$$\mathbf{y}(0) = \mathbf{y}_0, \qquad \mathbf{y}'(0) = \mathbf{0}.$$

Equation (2.1) is a typical second-order, constant-coefficient homogeneous system of equations, a problem routinely tackled with the help of some linear algebra.

2.1. Solving the Differential Equation. By analogy with the harmonic motion experienced by a single tethered mass we put forward the educated guess that the solution of equation (2.1) takes the form

$$\mathbf{y}(t) = e^{i\omega t}\,\mathbf{v},$$

where the scalar ω is the frequency and the constant vector $\mathbf{v}$ somehow accounts for the interplay between the masses. On substituting our guess into equation (2.1) we find that ω and $\mathbf{v}$ must obey the *generalized eigenproblem*

$$\mathbf{K}\mathbf{v} = \omega^2\mathbf{M}\mathbf{v}, \tag{2.3}$$

that is, ω^2 is an eigenvalue with associated eigenvector $\mathbf{v}$ for the pair $(\mathbf{K}, \mathbf{M})$. We now argue that this pair has n positive distinct eigenvalues and n linearly independent real eigenvectors.

We begin by noting that $\mathbf{K}\mathbf{v} = \lambda\mathbf{M}\mathbf{v}$ can be transformed to the standard eigenproblem, $\mathbf{A}\mathbf{u} = \lambda\mathbf{u}$, via the substitutions

$$\mathbf{u} = \mathbf{M}^{1/2}\mathbf{v} \quad \text{and} \quad \mathbf{A} = \mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2},$$

where $\mathbf{M}^{1/2}$ is the element-wise square root of $\mathbf{M}$ (since $\mathbf{M}$ is a diagonal matrix). We next note that, like $\mathbf{M}$ and $\mathbf{K}$, the matrix $\mathbf{A}$ is symmetric and positive definite. The Spectral Theorem (see, e.g., [19, Ch. 7] or [22, p. 61]) guarantees that $\mathbf{A}$ has n positive real eigenvalues, $\{\lambda_j\}_{j=1}^n$, and an orthonormal base of n real eigenvectors $\{\mathbf{u}_j\}_{j=1}^n$. It follows that $\{\mathbf{v}_j \equiv \mathbf{M}^{-1/2}\mathbf{u}_j\}_{j=1}^n$ is a basis of eigenvectors of the generalized problem (2.3), and we identify the frequencies as $\omega_j^2 = \lambda_j$. The orthogonality of the eigenvectors $\{\mathbf{u}_j\}$ of $\mathbf{A}$ ensures the eigenvectors of $(\mathbf{K}, \mathbf{M})$ are $\mathbf{M}$-*orthogonal*:

$$\mathbf{v}_j^T\mathbf{M}\mathbf{v}_k = 0 \text{ if } j \neq k, \qquad \mathbf{v}_j^T\mathbf{M}\mathbf{v}_j \neq 0. \tag{2.4}$$

It remains to demonstrate that the n eigenvalues are in fact distinct, or, in other words, that no eigenvalue may be associated with more than one eigendirection. If we express the jth row of $\mathbf{K}\mathbf{v} = \lambda\mathbf{M}\mathbf{v}$ in components with eigenvector $\mathbf{v} = [v_1, \ldots, v_n]^T$, we find

$$\Big(-\frac{\tau}{\ell_{j-1}}\Big)v_{j-1} + \Big(\frac{\tau}{\ell_{j-1}} + \frac{\tau}{\ell_j}\Big)v_j + \Big(-\frac{\tau}{\ell_j}\Big)v_{j+1} = \lambda m_j v_j. \tag{2.5}$$

where, by convention, $v_0 = v_{n+1} = 0$. Please note that if $v_1 = 0$ then the above implies that $v_2 = 0$ and so on. Hence, true eigenvectors obey $v_1 \neq 0$. Now if $\mathbf{w}$ also obeys $\mathbf{K}\mathbf{w} = \lambda\mathbf{M}\mathbf{w}$ for this same value of λ, then its components also satisfy (2.5), and so any linear combination of $\mathbf{v}$ and $\mathbf{w}$ will satisfy this equation. As $\mathbf{z} \equiv \mathbf{w} - (w_1/v_1)\mathbf{v}$ obeys $z_1 = 0$, it follows that $\mathbf{z} = \mathbf{0}$ and hence $\mathbf{w} = (w_1/v_1)\mathbf{v}$, i.e., our “new” eigenvector is simply a multiple of the original.

Since the eigenvectors $\{\mathbf{v}_k\}$ form a basis for n-dimensional space, we can write the solution to (2.1) as a time-varying linear combination:

$$\mathbf{y}(t) = \sum_{k=1}^{n} \gamma_k(t)\mathbf{v}_k.$$

Substituting this expansion into the differential equation (2.1) gives

$$\sum_{k=1}^{n} \gamma_k''(t)\mathbf{M}\mathbf{v}_k = -\sum_{k=1}^{n} \gamma_k(t)\mathbf{K}\mathbf{v}_k = -\sum_{k=1}^{n} \omega_k^2\gamma_k(t)\mathbf{M}\mathbf{v}_k.$$

To decompose into n independent equations, we premultiply this last equation by $\mathbf{v}_j^T$ and appeal to (2.4), giving the familiar scalar equation

$$\gamma_j''(t) = -\omega_j^2\gamma_j(t), \qquad j = 1, \ldots, n$$

with solution

$$\gamma_j(t) = c_j \cos(\omega_j t) + s_j \sin(\omega_j t).$$

These c_j and s_j coefficients are determined by the pluck: $\mathbf{y}'(0) = \mathbf{0}$ implies each $s_j = 0$, while $\mathbf{y}(0) = \mathbf{y}_0$ implies that the c_j are the expansion coefficients of $\mathbf{y}_0$ in the eigenvector basis, which can be found by solving the linear system

$$\mathbf{y}_0 = \sum_{j=1}^{n} c_j\mathbf{v}_j.$$

3. Experimental Apparatus and Model Verification. How well does the model (2.1) just derived predict what really happens when a beaded string is plucked?

We investigate this question by conducting experiments on a high-precision monochord constructed by students in our laboratory at Rice University, shown in Figure 3.1. For the “massless string” we use a length of 0.015 inch diameter nickel-plated steel musical wire donated by the Mapes Piano Wire Company. Tension is measured with a *force transducer* placed at the end of the string. The string then passes through a *collet*, which itself is mounted in a *collet vise*. The string proceeds through a photodetector that measures the vibrations at one point on the string. Brass beads are threaded onto the string, which continues through a second collet. (These beads have been carefully machined so as to snugly fit onto our wire.) Finally, the string is wound upon a spindle, which applies tension to the string. The experimenter winds the spindle until the string achieves a desired tension, then tightens the collet vises to fix the string at both ends (enforcing $y_0 = y_{n+1} = 0$).

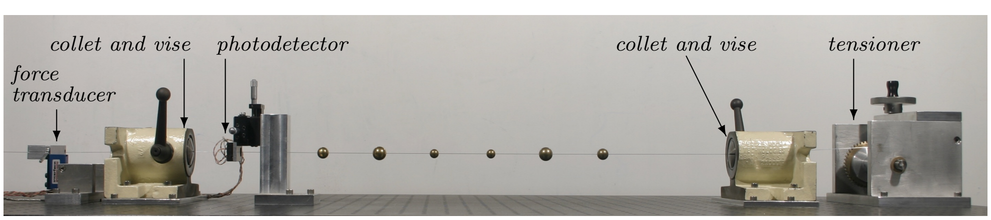


FIG. 3.1. *The monochord loaded with beads.*

A photodetector measures the displacement η_k at a single point along the string (not at a bead) at times $t_k = kh$ for some fixed time-step h. (The model only describes the motion of the beads, but the string itself must vibrate in concert: since we assume the string is perfectly elastic and the detector is placed between the fixed end and the first bead, these measurements are proportional to the first bead's displacement.) Consider a string loaded with five beads, as specified in Figure 3.2. We measure displacements for 10 sec. with $h = 1/50000$ sec., producing the samples $\{\eta_k\}$ shown on the left of Figure 3.3. (The magnitude of the displacements decay over the course of this ten second sample, reflecting some mild damping not captured by our simple physical model.) By analogy with the model (2.1), we expect that

$$\eta_k = \sum_{j=1}^{n} c_j \cos(t_k\sqrt{\lambda_j}) + \text{noise} \tag{3.1}$$

for some constants $c_1, \ldots, c_n$ that depend on the initial pluck. The "noise" term captures errors both in our mathematical description of physical reality and in our ability to accurately measure that reality, as discussed in more detail in §7.

To assess the accuracy of the model, we shall investigate whether the series of measurements $\{\eta_k\}$ for the five-beaded string in Figure 3.2 indeed oscillate at the frequencies predicted by the analysis in §2. To do so, we compute the discrete Fourier transform (DFT) of the data. A detailed discussion of the DFT is beyond our scope, but excellent expositions can be found in [4, 22], and the operation can be implemented in just a few lines of MATLAB.

```
freq = 2*pi*[0:N-1]/N*sample_rate; % set up vector of frequencies
semilogy(freq,abs(fft(eta)))       % plot magnitude of Fourier coefs
xlim([0 700])                      % set axis to relevant frequencies
```

These operations produce a plot that shows the component of the signal over a range of frequencies as shown on the right in Figure 3.3. A signal behaving like $t \mapsto \cos(\omega t)$ should produce a peak in the DFT at ω sec^{-1}. By equation (3.1), we expect our signal to be dominated by combinations of $\cos(t\sqrt{\lambda_j})$ terms, and so we should find peaks precisely at $\sqrt{\lambda_j}$, where λ_j is an eigenvalue of $(\mathbf{K}, \mathbf{M})$. As the beads are not point masses, their finite diameters restrict the string's ability to vibrate freely; this could effectively shorten the total length of the string. In Figure 3.3 we predict a range for each eigenvalue, with the lower end determined by the actual length of the string, and upper derived from the shorter string with the bead diameters removed.

4. Determining Mass and Position from Vibrations. Having seen the predictive ability of the forward model (2.1), we now address a more interesting—and challenging—problem: Given knowledge of eigenvalues (e.g., as discerned from the

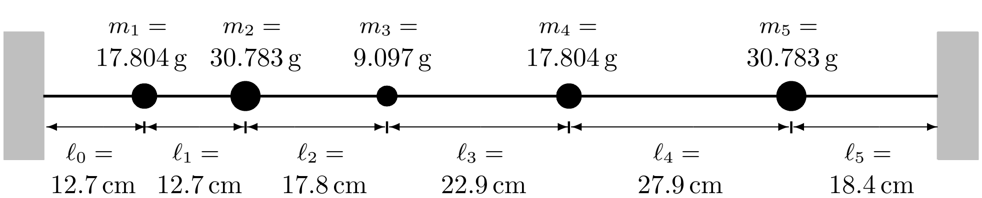


FIG. 3.2. *Configuration of the five-bead experiment described in §3. The string has total length of 112.4 cm and is drawn to a tension of* 1.706×10^7 *dyn. The beads have diameter of 5/8 in. (beads 1 and 4), 3/4 in. (beads 2 and 5), and 1/2 in. (bead 3). (Bead widths are exaggerated relative to the string length in our plots.)*

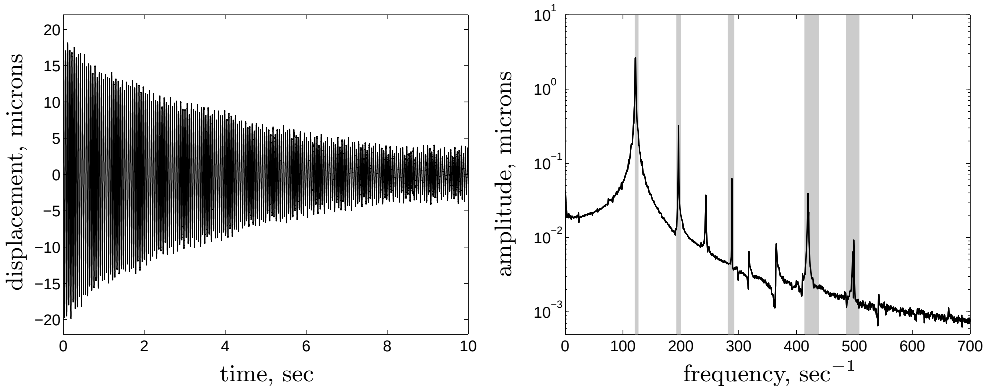


FIG. 3.3. *Displacement of the five-bead string in Figure 3.2 in the time domain (left) and frequency domain (right). Notice the five prominent peaks on the right, each corresponding to an eigenvalue of the pair* $(\mathbf{K}, \mathbf{M})$*: The gray shaded regions denote the predicted location of the peaks from the mathematical model,* $\omega_j = \sqrt{\lambda_j}$*, based on a string of full length (left boundaries) and shortened by the bead diameters (right boundaries).*

peaks in Figure 3.3), can we "hear the beads on the string"? Can we determine the bead positions and masses? Numerous elegant algorithms solve this problem. Here we present a general approach that first recovers the tridiagonal matrix $\mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2}$ using an orthogonal polynomial algorithm of de Boor and Golub [8], followed by extraction of bead lengths and masses using a technique of Gladwell [12, §4.4]. Some students may prefer Krein's continued fraction approach (see Supplement II of [10]), which is more elementary but specialized. (We provide a detailed guide to this technique in Appendix A, which can readily substitute for the remainder of this Section.)

4.1. Orthogonal polynomials with discrete inner products. We begin the inversion process with an excursion into the beautiful, important subject of orthogonal polynomials. Let $\mathcal{P}_k$ denote the set of all polynomials of degree k or less (with real coefficients). A k-degree polynomial p is *monic* if its leading coefficient is one, i.e., $p(z) = z^k + r(z)$ for some $r \in \mathcal{P}_{k-1}$. The set $\mathcal{P}_{n-1}$ can be viewed as a n dimensional vector space, and just as we have the dot product on n dimensional Euclidean space, we can also specify an *inner product* on $\mathcal{P}_n$. Toward this end, choose real numbers for

$$\begin{aligned} \textit{nodes}: &\quad \xi_1 < \xi_2 < \cdots < \xi_n \\ \text{and } \textit{weights}: &\quad w_1, w_2, \ldots, w_n > 0, \end{aligned}$$

and define the *inner product* of polynomials p and q to be

$$\langle p, q \rangle = \sum_{j=1}^{n} w_j \, p(\xi_j) q(\xi_j). \tag{4.1}$$

One can readily verify that this function obeys the axioms required for an inner product on $\mathcal{P}_{n-1}$; see, e.g., [19, p. 286]. We say p and q are *orthogonal* when $\langle p, q \rangle = 0$, and use the inner product to define a norm: $\|p\|^2 = \langle p, p \rangle \geq 0$.

The inner product is central to our development, which closely follows the important early work of de Boor and Golub [8]. The first step involves the construction of a sequence of monic polynomials of increasing degree that are all orthogonal to one another. The monic degree-0 polynomial must be

$$p_0(z) = 1.$$

To build the monic degree-1 polynomial, multiply $p_0(z)$ by z (to get a monic polynomial of degree 1), then subtract out the projection of zp_0 onto the span of p_0. This amounts to a single step of the Gram–Schmidt orthogonalization process [19, p. 307]:

$$p_1(z) = zp_0(z) - \frac{\langle zp_0, p_0\rangle}{\langle p_0, p_0\rangle} p_0(z) =: zp_0(z) - a_1 p_0(z).$$

Step $k+1$ follows similarly, via Gram–Schmidt:

$$p_{k+1}(z) = zp_k(z) - \sum_{j=1}^{k} \frac{\langle zp_k, p_j\rangle}{\langle p_j, p_j\rangle} p_j(z).$$

Now we make an essential simplifying observation: p_k must be orthogonal to *any* lower degree polynomial (which must be a linear combination of $p_0, \ldots, p_{k-1}$, as these orthogonal polynomials form a basis for $\mathcal{P}_{k-1}$). Since $zp_j \in \mathcal{P}_{j+1}$, notice that

$$\langle zp_k, p_j\rangle = \langle p_k, zp_j\rangle = 0, \qquad j < k-1.$$

The formula for p_{k+1} thus reduces to a *three-term recurrence relation*:

$$p_{k+1}(z) = zp_k(z) - \frac{\langle zp_k, p_k\rangle}{\langle p_k, p_k\rangle} p_k(z) - \frac{\langle zp_k, p_{k-1}\rangle}{\langle p_{k-1}, p_{k-1}\rangle} p_{k-1}(z).$$

The coefficient of p_{k-1} is always negative: since $zp_{k-1} = p_k + r$ for some $r \in \mathcal{P}_{k-1}$,

$$\frac{\langle zp_k, p_{k-1}\rangle}{\langle p_{k-1}, p_{k-1}\rangle} = \frac{\langle p_k, zp_{k-1}\rangle}{\langle p_{k-1}, p_{k-1}\rangle} = \frac{\langle p_k, p_k + r\rangle}{\langle p_{k-1}, p_{k-1}\rangle} = \frac{\langle p_k, p_k\rangle}{\langle p_{k-1}, p_{k-1}\rangle} = \frac{\|p_k\|^2}{\|p_{k-1}\|^2} \geq 0.$$

Thus we write

$$\begin{aligned} p_{k+1}(z) &= zp_k(z) - \frac{\langle zp_k, p_k\rangle}{\langle p_k, p_k\rangle} p_k(z) - \frac{\|p_k\|^2}{\|p_{k-1}\|^2} p_{k-1}(z) \\ &=: zp_k(z) - a_{k+1} p_k(z) - b_k^2 p_{k-1}(z); \end{aligned} \tag{4.2}$$

the label "b_k^2" emphasizes that the constant $b_k^2 := \|p_k\|^2/\|p_{k-1}\|^2$ is positive. We see that orthogonal polynomials intimately link to the constants $a_1, \ldots, a_n$ and $b_1, \ldots, b_{n-1}$. File this fact away for the moment, and return to the matrices at hand.

4.2. Eigenvalues of symmetric tridiagonal matrices. The matrices modeling beaded string vibrations can be written in the form

$$\mathbf{A}_n = \mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2} = \begin{bmatrix} a_1 & b_1 & & \\ b_1 & a_2 & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n \end{bmatrix},$$

where $a_k = \tau(1/\ell_{k-1} + 1/\ell_k)/m_k$ and $b_k = -\tau/(\ell_k\sqrt{m_k m_{k+1}}) < 0$; see eq. (2.5). The matrix $\mathbf{A}_n$ (more precisely, $-\mathbf{A}_n$) is an example of a *Jacobi matrix*, a widely-studied family having many fascinating properties. To analyze the eigenvalues, consider the characteristic polynomial $q_n(z) := \det(z\mathbf{I} - \mathbf{A}_n)$. Expand this determinant along the last row and column of $z\mathbf{I} - \mathbf{A}_n$ to arrive at the formula

$$\det(z\mathbf{I} - \mathbf{A}_n) = (z - a_n)\det(z\mathbf{I} - \mathbf{A}_{n-1}) - b_{n-1}^2 \det(z\mathbf{I} - \mathbf{A}_{n-2}).$$

In other words,

$$q_n(z) = (z - a_n)q_{n-1}(z) - b_{n-1}^2 q_{n-2}(z), \tag{4.3}$$

which is exactly the recurrence (4.2) for the orthogonal polynomial p_n. Jacobi matrices and orthogonal polynomials thus enjoy a rich intertwined history, with applications spanning from numerical analysis to mathematical physics.

4.3. Custom-tailoring an inner product. Suppose we know the eigenvalues of $\mathbf{A}_n$ and its upper-left $(n-1) \times (n-1)$ block, $\mathbf{A}_{n-1}$;

$$\text{eigenvalues of } \mathbf{A}_n\text{:} \quad \lambda_1 < \lambda_2 < \cdots < \lambda_n$$

$$\text{eigenvalues of } \mathbf{A}_{n-1}\text{:} \quad \mu_1 < \mu_2 < \cdots < \mu_{n-1}$$

but we do not know the entries of $\mathbf{A}_n$. Is there some way to build the matrix from these eigenvalues?

First note that the eigenvalues of $\mathbf{A}_n$ and $\mathbf{A}_{n-1}$ never coincide, and in fact weave between each other,

$$\lambda_1 < \mu_1 < \lambda_2 < \cdots < \lambda_{n-1} < \mu_{n-1} < \lambda_n, \tag{4.4}$$

an instance of Cauchy's Interlacing Theorem; see, e.g., [19, p. 552].

Given the parallel between the entries in $\mathbf{A}_n$ and the coefficients of the orthogonal polynomial recurrence relation (4.2), one naturally wonders: Does there exists some inner product (defined by nodes $\xi_1 < \cdots < \xi_n$ and (positive) weights $w_1, \ldots, w_n$) in which the characteristic polynomials

$$q_n(z) = \det(z\mathbf{I} - \mathbf{A}_n) = \prod_{j=1}^{n} (z - \lambda_j) \tag{4.5}$$

$$q_{n-1}(z) = \det(z\mathbf{I} - \mathbf{A}_{n-1}) = \prod_{j=1}^{n-1} (z - \mu_j) \tag{4.6}$$

are monic orthogonal polynomials? Given such an inner product, we could identify

$$a_n = \frac{\langle z q_{n-1}, q_{n-1}\rangle}{\langle q_{n-1}, q_{n-1}\rangle},$$

then run recurrence (4.2) *backward* to compute

$$b_{n-1}^2 q_{n-2}(z) = (z - a_n)q_{n-1}(z) - q_n(z). \tag{4.7}$$

To split b_{n-1}^2 from q_{n-2}, use the fact that q_{n-2} must be monic.

To discover such an inner product (i.e., suitable nodes and weights), we will explore some properties this inner product would need to obey. For one thing, q_n should be orthogonal to any lower degree polynomial in $\mathcal{P}_{n-1}$. We can always build a polynomial $r \in \mathcal{P}_{n-1}$ that interpolates $q_n(z)$ at the n nodes $z = \xi_1, \ldots, \xi_n$:

$$r(\xi_j) = q_n(\xi_j), \quad j = 1, \ldots, n.$$

(This is an extension of the fact that one can always construct a linear polynomial through any two points, a quadratic through any three points, etc.) Since $r \in \mathcal{P}_{n-1}$,

$$0 = \langle q_n, r\rangle = \sum_{j=1}^{n} w_j q_n(\xi_j) r(\xi_j) = \sum_{j=1}^{n} w_j q_n(\xi_j)^2.$$

As the weights must be positive, we conclude that

$$q_n(\xi_j) = 0, \qquad j = 1, \ldots, n.$$

That is, the n nodes $\xi_1, \ldots, \xi_n$ must coincide with the n roots of q_n:

$$\boxed{\xi_j = \lambda_j, \qquad j = 1, \ldots, n.} \tag{4.8}$$

To finish determining the inner product, we must specify the weights $w_1, \ldots, w_n$. In this endeavor we are aided by the *Lagrange interpolating polynomials*, objects that arise in a basic numerical analysis class. Define $\delta_k \in \mathcal{P}_{n-1}$ by the following n conditions: δ_k should pass through zero at the nodes ξ_j for $j \neq k$, while taking the value one at ξ_k. You can quickly verify that such polynomials have the form

$$\delta_k(z) = \prod_{\substack{j=1\\ j\neq k}}^{n} \frac{z - \xi_j}{\xi_k - \xi_j} \in \mathcal{P}_{n-1}. \tag{4.9}$$

Since $q_n(\xi_k) = 0$ for all k,

$$\begin{aligned} q_n'(\xi_k) &= \lim_{z\to\xi_k} \frac{q_n(z) - q_n(\xi_k)}{z - \xi_k} \\ &= \lim_{z\to\xi_k} \frac{(z-\xi_k) \displaystyle\prod_{j=1; j\neq k}^{n} (z - \xi_j)}{z - \xi_k} = \prod_{\substack{j=1\\ j\neq k}}^{n} (\xi_k - \xi_j), \end{aligned} \tag{4.10}$$

giving the simple expression

$$\delta_k(z) = \frac{q_{n-1}(z)}{q_n'(\xi_k)} + r_k(z)$$

for some $r_k \in \mathcal{P}_{n-2}$, since q_{n-1} is monic. Take the inner product of the Lagrange polynomial with q_{n-1} to find

$$\langle q_{n-1}, \delta_k \rangle = \sum_{j=1}^{n} w_j q_{n-1}(\xi_j)\delta_k(\xi_j) = w_k q_{n-1}(\xi_k),$$

where $\langle q_{n-1}, r_k \rangle = 0$ since $r_k \in \mathcal{P}_{n-2}$. Now solve for the weights:

$$w_k = \frac{\langle q_{n-1}, \delta_k \rangle}{q_{n-1}(\xi_k)} = \frac{\langle q_{n-1}, q_{n-1} \rangle}{q_n'(\xi_k) q_{n-1}(\xi_k)} = \frac{\|q_{n-1}\|^2}{q_n'(\xi_k) q_{n-1}(\xi_k)}, \qquad k = 1, \ldots, n.$$

Do you notice a subtle drawback of this formula? The expression for w_k involves $\|q_{n-1}\|$, which we can only compute once all the weights are known. There is an easy dodge: the term $\|q_{n-1}\|^2$ is independent of k, so it affects all the weights the same way. Orthogonality—the property we care most about—is independent of the collective scaling of the weights, so we can select any convenient scaling, such as

$$\boxed{w_k = \frac{1}{q_n'(\xi_k) q_{n-1}(\xi_k)}, \qquad k = 1, \ldots, n.} \tag{4.11}$$

There can be no division by zero in this formula, since the nodes $\xi_k = \lambda_k$ never coincide with the roots $\mu_1, \ldots, \mu_{n-1}$ of q_{n-1} by the interlacing property (4.4); however, computational subtleties can emerge when computing these weights, as they can vary substantially in magnitude with the index k, especially when n is large.

4.4. Extraction of eigenvalue data. In our string laboratory we experimentally determine the eigenvalues $\lambda_1 < \ldots < \lambda_n$ of the matrix $\mathbf{A}_n = \mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2}$ having the form

$$\begin{bmatrix} \frac{\tau}{\ell_0 m_1} + \frac{\tau}{\ell_1 m_1} & -\frac{\tau}{\ell_1\sqrt{m_1 m_2}} & & \\ -\frac{\tau}{\ell_1\sqrt{m_1 m_2}} & \frac{\tau}{\ell_1 m_2} + \frac{\tau}{\ell_2 m_2} & \ddots & \\ & \ddots & \ddots & -\frac{\tau}{\ell_{n-1}\sqrt{m_n m_{n-1}}} \\ & & -\frac{\tau}{\ell_{n-1}\sqrt{m_n m_{n-1}}} & \frac{\tau}{\ell_{n-1} m_n} + \frac{\tau}{\ell_n m_n} \end{bmatrix}.$$

We can measure the total length of the string, $\ell = \sum_{k=0}^{n} \ell_k$ and the tension, τ. How can we *experimentally* obtain $\mu_1 < \cdots < \mu_{n-1}$, the second set of eigenvalues coming from $\mathbf{A}_{n-1}$? This would amount to fixing the nth mass, thus making the string one bead shorter. One could imagine building an apparatus to implement this condition, but it would require knowledge of the internal composition of the string: namely, the *location* of the nth mass, hence the quantity ℓ_n. We prefer a different strategy that will avoid such an intrusion into the interior of the string. We have laboratory data for the string fixed at both ends (the "fixed–fixed" string), giving measured eigenvalues $\lambda_1 < \cdots < \lambda_n$. Now suppose we leave the string configuration the same, except now the right end is attached by a ring to a frictionless vertical pole, so that end remains level with nth bead (the "fixed–flat" string): measure the eigenvalues $\widehat{\lambda}_1 < \cdots < \widehat{\lambda}_n$ of this modified string. (We will later find a more palatable route to this data using only our regular fixed–fixed experiment.)

The beads of the fixed–flat string obey the same equations of motion (2.1) derived for the fixed–fixed string save for the last bead, which is governed by

$$m_n y_n''(t) = -\tau\Big(\frac{y_n(t) - y_{n-1}(t)}{\ell_{j-1}}\Big);$$

that is, $y_{n+1} = y_n$ replaces the fixed condition $y_{n+1} = 0$. This change in the equation for the nth bead gives a fixed–flat matrix $\widehat{\mathbf{A}}_n$ that differs from the fixed–fixed matrix only in the last entry:

$$\widehat{\mathbf{A}}_n = \mathbf{A}_n - \frac{\tau}{\ell_n m_n}\mathbf{e}_n\mathbf{e}_n^{\mathrm{T}}. \tag{4.12}$$

Here $\mathbf{e}_k$ denotes the kth column of the identity matrix, whose dimension is clear from the context. Label the (n, n) entry of $\widehat{\mathbf{A}}_n$ as $\widehat{a}_n$ (to distinguish it from the (n, n) entry a_n of $\mathbf{A}_n$), Now partition the two Jacobi matrices as

$$\mathbf{A}_n = \begin{bmatrix} \mathbf{A}_{n-1} & b_{n-1}\mathbf{e}_{n-1} \\ b_{n-1}\mathbf{e}_{n-1}^{\mathrm{T}} & a_n \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{n-2} & b_{n-2}\mathbf{e}_{n-2} & \mathbf{0} \\ b_{n-2}\mathbf{e}_{n-2}^{\mathrm{T}} & a_{n-1} & b_{n-1} \\ \mathbf{0} & b_{n-1} & a_n \end{bmatrix}$$

and

$$\widehat{\mathbf{A}}_n = \begin{bmatrix} \mathbf{A}_{n-1} & b_{n-1}\mathbf{e}_{n-1} \\ b_{n-1}\mathbf{e}_{n-1}^{\mathrm{T}} & \widehat{a}_n \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{n-2} & b_{n-2}\mathbf{e}_{n-2} & \mathbf{0} \\ b_{n-2}\mathbf{e}_{n-2}^{\mathrm{T}} & a_{n-1} & b_{n-1} \\ \mathbf{0} & b_{n-1} & \widehat{a}_n \end{bmatrix}.$$

With these forms, the characteristic polynomials for $\mathbf{A}_n$ and $\widehat{\mathbf{A}}_n$ are

$$q_n(z) = \det(z\mathbf{I} - \mathbf{A}_n) = (z - a_n)\det(z\mathbf{I} - \mathbf{A}_{n-1}) + b_{n-1}\det\left(\begin{bmatrix} z\mathbf{I} - \mathbf{A}_{n-2} & \mathbf{0} \\ -b_{n-2}\mathbf{e}_{n-2}^{\mathrm{T}} & -b_{n-1} \end{bmatrix}\right)$$

$$\widehat{q}_n(z) = \det(z\mathbf{I} - \widehat{\mathbf{A}}_n) = (z - \widehat{a}_n)\det(z\mathbf{I} - \mathbf{A}_{n-1}) + b_{n-1}\det\left(\begin{bmatrix} z\mathbf{I} - \mathbf{A}_{n-2} & \mathbf{0} \\ -b_{n-2}\mathbf{e}_{n-2}^{\mathrm{T}} & -b_{n-1} \end{bmatrix}\right),$$

whose difference is

$$\begin{aligned} \widehat{q}_n(z) - q_n(z) &= (a_n - \widehat{a}_n)\det(z\mathbf{I} - \mathbf{A}_{n-1}) \\ &= (a_n - \widehat{a}_n)q_{n-1}(z). \end{aligned} \tag{4.13}$$

Hence the eigenvalues $\mu_1 < \cdots < \mu_{n-1}$ of the matrix $\mathbf{A}_{n-1}$, which helped determine the inner product, must match the roots of $\widehat{q}_n - q_n$. As de Boor and Golub [8] observe, we can obtain a formula for the weights without explicitly computing the eigenvalues of $\mathbf{A}_{n-1}$ (which would require that we find the roots of a polynomial $\widehat{q}_n - q_n$ whose coefficients are subject to errors, a numerically unappealing prospect). In particular, the formula (4.11) for the weights of the inner product only requires the values $q_{n-1}(\lambda_k)$. We can access these quantities by recalling that the eigenvalues $\lambda_1, \ldots, \lambda_n$ are the roots of q_n, hence from (4.13),

$$\widehat{q}_n(\lambda_k) = (a_n - \widehat{a}_n)q_{n-1}(\lambda_k), \qquad k = 1, \ldots, n. \tag{4.14}$$

This equation specifies $q_{n-1}(\lambda_k) = q_{n-1}(\xi_k)$ up to the constant $\widehat{a}_n - a_n$, which can be omitted from the weights (like $\|q_{n-1}\|^2$ earlier), as we are unconcerned about scalings of the inner product. Thus use the scaled weights

$$\boxed{w_k = \frac{1}{q_n'(\xi_k)\widehat{q}_n(\xi_k)}, \qquad k = 1, \ldots, n.} \tag{4.15}$$

To obtain q_{n-1}, use (4.13) along with the fact that q_{n-1} must be monic, i.e., $a_n - \widehat{a}_n$ is the leading coefficient of the degree $n-1$ polynomial $\widehat{q}_n - q_n$. Since

$$q_n(z) = \prod_{j=1}^{n}(z - \lambda_j) = z^n - \left(\sum_{j=1}^{n}\lambda_j\right)z^{n-1} + \cdots$$

$$\widehat{q}_n(z) = \prod_{j=1}^{n}(z - \widehat{\lambda}_j) = z^n - \left(\sum_{j=1}^{n}\widehat{\lambda}_j\right)z^{n-1} + \cdots,$$

we can compute

$$a_n - \widehat{a}_n = \sum_{j=1}^{n}(\lambda_j - \widehat{\lambda}_j).$$

and thus

$$q_{n-1}(z) = \frac{\widehat{q}_n(z) - q_n(z)}{\sum_{j=1}^n (\lambda_j - \widehat{\lambda}_j)}. \tag{4.16}$$

4.5. An algorithm for recovering the Jacobi matrix. We are now equipped to build $\mathbf{A}_n$ from the measured eigenvalues $\lambda_1 < \cdots < \lambda_n$ and $\widehat{\lambda}_1 < \cdots < \widehat{\lambda}_n$ from the fixed–fixed and fixed–flat string. We will represent the orthogonal polynomials q_k by their values at the nodes $\xi_1, \ldots, \xi_n$, rather than by their coefficients. As such, equation (4.7) will not suffice to reveal b_k^2. Instead, take the inner product of (4.7) (with k replacing $n-1$) with $zq_k(z)$ to obtain

$$b_k^2 = \frac{\langle zq_k, zq_k\rangle - a_{k+1}\langle q_k, zq_k\rangle - \langle q_{k+1}, zq_k\rangle}{\langle q_{k-1}, zq_k\rangle}. \tag{4.17}$$

The numerator appears troubling, as q_{k-1} will not have been determined at this stage. There is a slick work-around:

$$\langle q_{k-1}, zq_k\rangle = \langle zq_{k-1}, q_k\rangle = \langle q_k + r, q_k\rangle = \langle q_k, q_k\rangle$$

for some $r \in \mathcal{P}_{k-1}$. Hence we can use the formula for b_k^2 given in step 5(a) in Figure 4.1. Since $b_k = -\tau/(\ell_k\sqrt{m_k m_{k+1}}) < 0$, we define $b_k := -\sqrt{b_k^2}$.

4.6. Relating the Jacobi matrix approach to beaded strings. We can use the procedure detailed in Figure 4.1 to recover the entries of $\mathbf{A}_n$, but how then do we access the masses and lengths from which these entries derive? Gladwell describes a procedure for mass–spring networks that we can adapt for our setting [12, p. 77ff].

Let $\mathbf{e}$ denote the vector with one in each entry, $\mathbf{e} = [1, 1, \ldots, 1]^{\mathrm{T}}$, while $\mathbf{e}_1$ and $\mathbf{e}_n$ denote the first and last columns of the identity matrix. Observe from (2.2) that

$$\mathbf{K}\mathbf{e} = \frac{\tau}{\ell_0}\mathbf{e}_1 + \frac{\tau}{\ell_n}\mathbf{e}_n, \tag{4.18}$$

so knowledge of $\mathbf{K}$ would reveal ℓ_0 and ℓ_n via (4.18), from which we could determine the other lengths. However at this stage we only know $\mathbf{A}_n$, not $\mathbf{K}$. Using $\mathbf{K} = \mathbf{M}^{1/2}\mathbf{A}_n\mathbf{M}^{1/2}$, write

$$\mathbf{M}^{1/2}\mathbf{A}_n\mathbf{M}^{1/2}\mathbf{e} = \frac{\tau}{\ell_0}\mathbf{e}_1 + \frac{\tau}{\ell_n}\mathbf{e}_n.$$

Premultiplying by $\mathbf{M}^{-1/2}$ and labeling $\mathbf{d} := \mathbf{M}^{1/2}\mathbf{e} = [\sqrt{m_1}, \ldots, \sqrt{m_n}]^{\mathrm{T}}$ gives

$$\mathbf{A}_n\mathbf{d} = \frac{\tau}{\ell_0\sqrt{m_1}}\mathbf{e}_1 + \frac{\tau}{\ell_n\sqrt{m_n}}\mathbf{e}_n. \tag{4.19}$$

The vector $\mathbf{d}$ that solves this linear system will thus reveal the bead masses. We could solve the system (4.19) for $\mathbf{d}$ using Gaussian elimination, except we lack a formula for the right hand side. We can obtain the right hand side up to a scaling factor through the following strategy. Solve the two linear systems

$$\mathbf{A}_n\mathbf{x} = \mathbf{e}_1, \qquad \mathbf{A}_n\mathbf{y} = \mathbf{e}_n$$

for the vectors $\mathbf{x}$ and $\mathbf{y}$.† Now

$$\mathbf{A}_n\Big(\frac{\tau}{\ell_0\sqrt{m_1}}\mathbf{x} + \frac{\tau}{\ell_n\sqrt{m_n}}\mathbf{y}\Big) = \frac{\tau}{\ell_0\sqrt{m_1}}\mathbf{e}_1 + \frac{\tau}{\ell_n\sqrt{m_n}}\mathbf{e}_n = \mathbf{A}_n\mathbf{d}.$$

†Alternatively, the entries in $\mathbf{x}$ and $\mathbf{y}$ can be obtained using explicit formulas involving minors of $\mathbf{A}_n$; see eq. (4.4.12) in Gladwell [12].

1. Use (4.8) to determine the nodes of the inner product:
$$\xi_k := \lambda_k.$$
2. Use (4.15) to construct the weights, with $q_n'(\xi_k)$ given by (4.10):
$$w_k := \frac{1}{q_n'(\xi_k)\widehat{q}_n(\xi_k)} = \frac{1}{\Big(\prod_{\substack{j=1\\ j\neq k}}^n(\xi_k-\xi_j)\Big)\Big(\prod_{j=1}^n(\xi_k-\widehat{\lambda}_j)\Big)}.$$
3. Determine the values of q_n and q_{n-1} at the nodes; see (4.16):
$$q_n(\xi_k) := 0$$
$$q_{n-1}(\xi_k) := \frac{\widehat{q}_n(\xi_k)-q_n(\xi_k)}{\sum_{j=1}^n(\lambda_j-\widehat{\lambda}_j)} = \frac{\prod_{j=1}^n(\xi_k-\widehat{\lambda}_j)}{\sum_{j=1}^n(\lambda_j-\widehat{\lambda}_j)}.$$
4. Compute, via the inner product (4.1)
$$a_n := \frac{\langle zq_{n-1},q_{n-1}\rangle}{\langle q_{n-1},q_{n-1}\rangle} = \frac{\sum_{j=1}^n w_j\xi_j q_{n-1}(\xi_j)^2}{\sum_{j=1}^n w_j q_{n-1}(\xi_j)^2}.$$
5. For $k = n-1, n-2, \ldots, 1$
   (a) Compute $b_k = -\sqrt{b_k^2}$ via the approach described after (4.17):
   $$b_k^2 := \frac{\langle zq_k,zq_k\rangle - a_{k+1}\langle q_k,zq_k\rangle - \langle q_{k+1},zq_k\rangle}{\langle q_k,q_k\rangle}.$$
   (b) Compute q_{k-1} at the nodes:
   $$q_{k-1}(\xi_j) := \frac{(\xi_j-a_{k+1})q_k(\xi_j)-q_{k+1}(\xi_j)}{b_k^2}.$$
   (c) Define
   $$a_k := \frac{\langle zq_{k-1},q_{k-1}\rangle}{\langle q_{k-1},q_{k-1}\rangle}.$$

FIG. 4.1. *Algorithm for recovering the main diagonal $(a_1,\ldots,a_n)$ and off-diagonal $(b_1,\ldots,b_{n-1})$ of the Jacobi matrix $\mathbf{A}_n$ from the fixed–fixed and fixed–flat eigenvalues $\lambda_1<\cdots<\lambda_n$ and $\widehat{\lambda}_1<\cdots<\widehat{\lambda}_n$. Adapted from de Boor and Golub [8].*

Since $\mathbf{A}_n$ is nonsingular (it is positive definite, as described in §2), we can access $\mathbf{d}$ as

$$\mathbf{d} = \frac{\tau}{\ell_0\sqrt{m_1}}\mathbf{x} + \frac{\tau}{\ell_n\sqrt{m_n}}\mathbf{y}. \tag{4.20}$$

One obstacle remains: What are the coefficients $\tau/(\ell_0\sqrt{m_1})$ and $\tau/(\ell_n\sqrt{m_n})$? We can resolve this issue using our experimental data. Since the trace of a matrix (sum of the diagonal elements) is the sum of the eigenvalues [19, p. 494], the experimentally-measured eigenvalues reveal, by way of (4.12),

$$\frac{\tau}{\ell_n m_n} = \mathrm{tr}(\mathbf{A}_n) - \mathrm{tr}(\widehat{\mathbf{A}}_n) = \sum_{k=1}^n(\lambda_k-\widehat{\lambda}_k). \tag{4.21}$$

Thus the second coefficient in (4.20) is thus given by

$$\frac{\tau}{\ell_n\sqrt{m_n}} = \frac{\tau\sqrt{m_n}}{\ell_n m_n} = \sqrt{m_n}\sum_{k=1}^{n}(\lambda_k - \widehat{\lambda}_k). \tag{4.22}$$

To obtain the first coefficient in (4.20), note that the nth row of (4.20) gives

$$\sqrt{m_n} = \frac{\tau}{\ell_0\sqrt{m_1}}x_n + \frac{\tau}{\ell_n\sqrt{m_n}}y_n,$$

which can be rearranged as

$$\frac{\tau}{\ell_0\sqrt{m_1}} = \frac{1}{x_n}\Big(\sqrt{m_n} - \frac{\tau}{\ell_n\sqrt{m_n}}y_n\Big) = \frac{\sqrt{m_n}}{x_n}\Big(1 - y_n\sum_{k=1}^{n}(\lambda_k - \widehat{\lambda}_k)\Big). \tag{4.23}$$

Given coefficients (4.22) and (4.23), equation (4.20) gives $\mathbf{d}$ up to a factor of $\sqrt{m_n}$:

$$\widetilde{\mathbf{d}} := \frac{\mathbf{d}}{\sqrt{m_n}} = \frac{1}{x_n}\Big(1 - y_n\sum_{k=1}^{n}(\lambda_k - \widehat{\lambda}_k)\Big)\mathbf{x} + \Big(\sum_{k=1}^{n}(\lambda_k - \widehat{\lambda}_k)\Big)\mathbf{y}, \tag{4.24}$$

and so

$$\mathbf{K}/m_n = (\mathbf{M}^{1/2}\mathbf{A}_n\mathbf{M}^{1/2})/m_n = \mathrm{diag}(\widetilde{\mathbf{d}})\mathbf{A}_n\mathrm{diag}(\widetilde{\mathbf{d}}). \tag{4.25}$$

Recalling (2.2), the $(j, j+1)$ entries of (4.25) give $-\tau/(\ell_j m_n) = b_j\widetilde{d}_j\widetilde{d}_{j+1}$, i.e.,

$$\ell_j m_n = -\frac{\tau}{b_j\widetilde{d}_j\widetilde{d}_{j+1}}, \qquad j = 1, \ldots, n-1. \tag{4.26}$$

We can then find the end lengths from the $(1, 1)$ and (n, n) entries of (4.25):

$$\ell_0 m_n = \frac{\tau}{a_1\widetilde{d}_1^2 + \tau/(\ell_1 m_n)}, \qquad \ell_n m_n = \frac{\tau}{a_n\widetilde{d}_n^2 + \tau/(\ell_{n-1}m_n)}. \tag{4.27}$$

1. Solve $\mathbf{A}_n\mathbf{x} = \mathbf{e}_1$ and $\mathbf{A}_n\mathbf{y} = \mathbf{e}_n$ for $\mathbf{x}$ and $\mathbf{y}$.
2. $\gamma_2 := \sum_{j=1}^{n}(\lambda_j - \widehat{\lambda}_j)$ and $\gamma_1 := (1 - y_n\gamma_2)/x_n$.
3. $\widetilde{\mathbf{d}} := \mathbf{d}/\sqrt{m_n} = \gamma_1\mathbf{x} + \gamma_2\mathbf{y}$.
4. $\ell_j m_n := -\tau/(b_j\widetilde{d}_j\widetilde{d}_{j+1})$ for $j = 1, \ldots, n-1$.
   $\ell_0 m_n := \tau/(a_1\widetilde{d}_1^2 - \tau/(\ell_1 m_n))$.
   $\ell_n m_n := \tau/(a_n\widetilde{d}_n^2 - \tau/(\ell_{n-1}m_n))$.
5. $m_n := \Big(\sum_{j=0}^{n}\ell_j m_n\Big)/\ell$.

FIG. 4.2. *Algorithm for extracting bead masses and locations from a Jacobi matrix $\mathbf{A}_n$ (main diagonal entries $a_1, \ldots, a_n$ and off-diagonal entries $b_1, \ldots, b_{n-1}$), given knowledge of the tension τ, total length of the string, ℓ, and the fixed–fixed and fixed–flat eigenvalues. (For the symmetric strings in §5, be sure to use $\ell = L/2$.)*

We now have expressions for all the masses via (4.24) and lengths via (4.26)–(4.27), up to the scaling factor m_n. We do not know m_n (as it is an interior property of the string), but we do know the total string length, ℓ, and from this we can compute $m_n = \left(\sum_{j=0}^{n} \ell_j m_n\right)/\ell$. This overall recovery process is summarized in Figure 4.2.

4.7. The special case of $n = 1$ bead. The case of $n = 1$ bead, in which case $\mathbf{A}_n = a_1$ is a 1×1 matrix, requires special attention. Recovery of $\mathbf{A}_n$ from λ_1 and $\widehat{\lambda}_1$ becomes trivial: $\mathbf{A}_n = \lambda_1$. The reader can then verify, starting from (4.21), that

$$\ell_0 m_1 = \frac{\tau}{\widehat{\lambda}_1}, \qquad \ell_1 m_1 = \frac{\tau}{\lambda_1 - \widehat{\lambda}_1}, \qquad m_1 = \frac{\tau \lambda_1}{\ell \widehat{\lambda}_1 (\lambda_1 - \widehat{\lambda}_1)}.$$

5. Symmetrically Loaded Strings. Having seen how to determine the bead locations and masses from two sets of eigenvalues, we must now address the practical issue of experimentally measuring this data. When the string is fixed at both ends, we can approximate the eigenvalues using the reliable experiments performed in §3. The fixed–flat string poses a greater challenge. Fortunately, in one interesting special case we can determine these eigenvalues without any modification to the experimental apparatus. (Boyko and Pivovarchik recently proposed an alternative: clamping the string at an interior point and measuring the spectra on each sub-string [3].)

Suppose that the number of beads, N, is even, and the masses of the beads and lengths between them are symmetric about the midpoint of the string. To distinguish this case from the general scenario considered earlier, we shall denote these masses by $M_1, \ldots, M_N$, and the lengths by $L_0, \ldots, L_N$, with the total length $L = \sum_{j=0}^{N} L_j$. Thus, the symmetric arrangement requires that

$$\begin{aligned} M_1 &= M_N & L_0 &= L_N \\ M_2 &= M_{N-1} & L_1 &= L_{N-1} \\ &\vdots & &\vdots \\ M_{N/2} &= M_{N/2+1}, & L_{N/2-1} &= L_{N/2+1} \end{aligned}$$

as illustrated for $N = 6$ in Figure 5.1. We can install such a symmetric arrangement on our laboratory's monochord (as pictured in Figure 3.1), then experimentally measure the eigenvalues when both ends of the string are fixed; let us label these eigenvalues $\Lambda_1 < \Lambda_2 < \cdots < \Lambda_N$. In Figure 5.2 we show the eigenvectors for the configuration shown in Figure 5.1. This illustration reveals a remarkable property: eigenvectors corresponding to the odd eigenvalues Λ_1, Λ_3, and Λ_5 are all *symmetric* about the middle of the string: If we cut the string in half, these eigenvectors would be *fixed* on the left end and be *flat* at the right. On the other hand, eigenvectors corresponding to the even eigenvalues Λ_2, Λ_4, and Λ_6 are all *antisymmetric* about the midpoint: If we cut the string in half, these eigenvectors would be *fixed at zero at both ends.* These

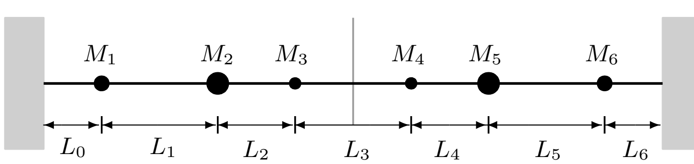


FIG. 5.1. *A string with six beads with lengths and masses arranged symmetrically about the middle of the string, denoted by the vertical gray line.*

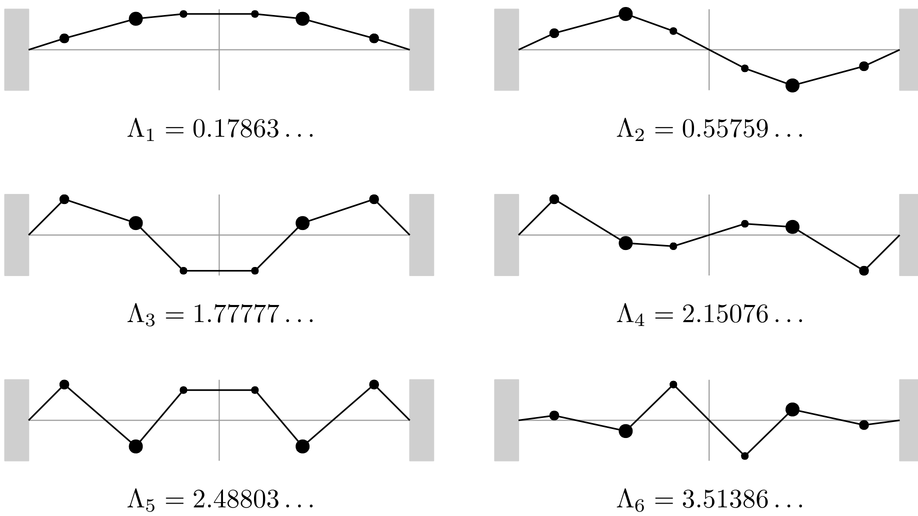


FIG. 5.2. *Eigenvectors for the string shown in Figure 5.1. In each plot, the vertical displacement of the kth mass indicates the kth entry of the corresponding eigenvector* $\mathbf{v}_j$ *of* $\mathbf{M}^{-1}\mathbf{K}$.

observations hold for all symmetric string configurations, and they hint at a key fact:

> *The N eigenvalues of a symmetric beaded string fixed at both ends exactly match the $N/2$ fixed–fixed and $N/2$ fixed–flat eigenvalues associated with half of the string.*

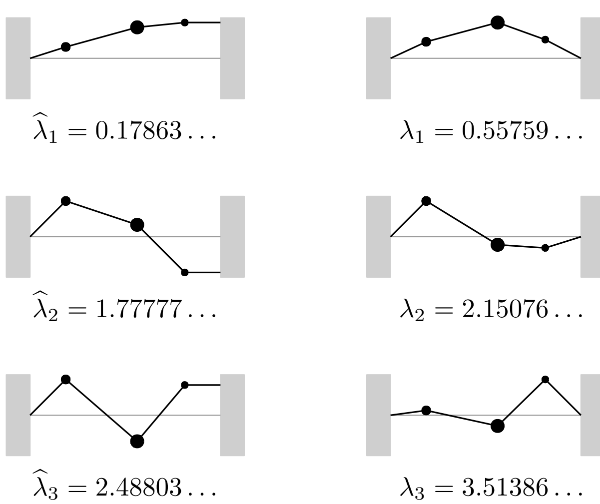


FIG. 5.3. *Eigenvectors for the half-string corresponding to the symmetric configuration in Figure 5.1. The eigenvectors on the left satisfy the fixed–flat conditions; those on the right satisfy fixed–fixed conditions. Compare to the eigenvectors for the symmetric string in Figure 5.2.*

More precisely, consider a length $\ell = L/2$ string having $n = N/2$ beads with masses

$$m_j = M_j, \quad j = 1, \ldots, n$$

separated by lengths

$$\begin{aligned} \ell_j &= L_j, \quad j = 0, \ldots, n-1 \\ \ell_n &= L_n/2. \end{aligned}$$

The odd eigenvalues for the symmetric string match the eigenvalues for the fixed–flat half-string,

$$\widehat{\lambda}_j = \Lambda_{2j-1}, \quad j = 1, \ldots, n$$

while the even eigenvalues for the symmetric string match the eigenvalues for the fixed–fixed half string,

$$\lambda_j = \Lambda_{2j}, \quad j = 1, \ldots, n.$$

Figure 5.3 shows eigenvectors for the half-string corresponding to the symmetric string in Figure 5.1. The fixed–flat eigenvectors, shown in the left column, correspond to the left halves of the eigenvectors shown on the left of Figure 5.2, and the eigenvalues match Λ_1, Λ_3, and Λ_5 exactly. Similarly, the fixed–fixed eigenvectors on the half-string, shown on the right of Figure 5.3, equal the left halves of the eigenvectors on the right of Figure 5.2, and the eigenvalues perfectly match Λ_2, Λ_4, and Λ_6.

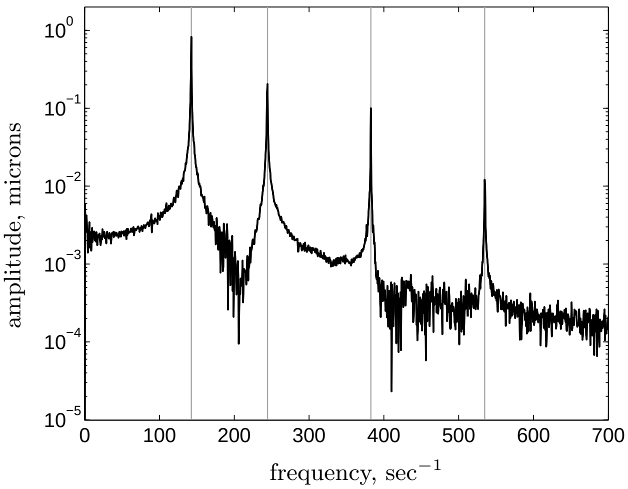
We thus have a method for obtaining the data required for the inversion procedure described in the last section, *provided our string has symmetrically arranged beads.*

1. Pluck the symmetric string and record the $N = 2n$ eigenvalues $\Lambda_1, \ldots, \Lambda_{2n}$.
2. Relabel these eigenvalues for the corresponding half-string: $\lambda_j = \Lambda_{2j}$ and $\widehat{\lambda}_j = \Lambda_{2j-1}$ for $j = 1, \ldots, n$.
3. Apply the inversion algorithms (Figures 4.1 and 4.2) to the half-string eigenvalues to obtain $\ell_0, \ldots, \ell_n$ and $m_1, \ldots, m_n$.
4. Recover the parameters for the original symmetric string:

$$L_0 = L_N = \ell_0, \quad \ldots, \quad L_{n-1} = L_{n+1} = \ell_{n-1}, \quad L_n = 2\ell_n;$$

$$M_1 = M_N = m_1, \quad \ldots, \quad M_n = M_{n+1} = m_n.$$

6. Experimental Results for the Inverse Problem. How well does this algorithm perform when applied to real data collected from a string with symmetrically configured beads? Figures 6.1 and 6.2 provide data for strings with four and six beads. For each string we collect displacement data for ten seconds at 50000 samples per second. As described in §3, the discrete Fourier transform (DFT) of this data reveals peaks that should correspond to $\sqrt{\Lambda_j}$ for $j = 1, \ldots, N$. Deriving from these peaks estimates of the eigenvalues Λ_j, we sort the eigenvalues into fixed–flat and fixed–fixed eigenvalues as described in the last section, then feed them to the inversion algorithms in Figure 4.1 and 4.2. The results are illustrated in Figures 6.1 and 6.2. We see quite satisfactory agreement for the four bead case, with all quantities recovered to a relative error less than 3.5%. Some challenges begin to emerge with six beads. For one, the frequency plot in Figure 6.2 reveals a number of secondary peaks. Moreover, the forward experiment illustrated in Figure 3.3 hints that the finite width of the beads may introduce some uncertainty. For this data, the recovered length L_0 suffers from an 18% relative error; the other lengths and masses are a bit more accurate.



$L = 112.4$ cm

$\tau = 1.542 \times 10^7$ dyn

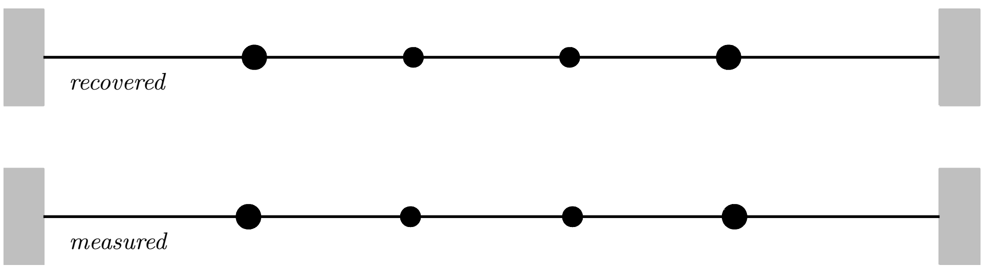
experimental eigenvalues:

$\Lambda_1 = 2.03 \times 10^4\ \text{sec}^{-2}$

$\Lambda_2 = 5.97 \times 10^4\ \text{sec}^{-2}$

$\Lambda_3 = 1.46 \times 10^5\ \text{sec}^{-2}$

$\Lambda_4 = 2.86 \times 10^5\ \text{sec}^{-2}$



	M_1 (g)	M_2 (g)	L_0 (cm)	L_1 (cm)	L_2 (cm)
recovered	17.2	8.8	26.5	19.9	19.6
measured	17.8	9.1	25.7	20.3	20.3

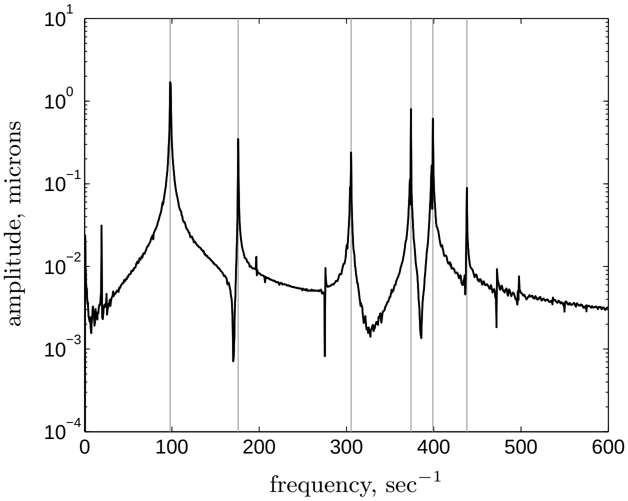
FIG. 6.1. *Experimental results illustrating the recovery of positions and masses for a string with four symmetrically-placed beads. The top plot shows the DFT of the displacement data; the four peaks give values for $\sqrt{\Lambda_j}$, shown as gray vertical lines. The middle illustrations compare the positions and masses recovered from these experimental eigenvalues to the "true" positions and masses measured directly, with the corresponding data presented in the table.*

7. Further explorations. We encourage readers to conduct explorations of both the forward and inverse problem using data sets we provide on the website

`http://www.caam.rice.edu/~beads`

which includes time series data and peak locations for numerous two, four, and six beads systems.

As you will observe in your own experiments, the inverse procedure we describe is subjected to a variety of errors. To begin with, our measurements introduce imprecision: we measure lengths accurate to ± 2 mm, tension to $\pm 10^4$ dyn, masses to ± 0.01 g, and frequencies to $\pm.7\,\text{sec}^{-1}$. Our mathematical model is only an approximation of the true system; we account for neither nonlinear effects, nor the damping that causes our string to eventually stop vibrating. The string itself is neither perfectly flexible nor massless, as the model assumes, nor are the beads point-masses. (Figure 3.3 hints at one possible effect of the bead widths on the eigenvalues.) The model describes



$L = 112.4$ cm

$\tau = 1.660 \times 10^7$ dyn

experimental eigenvalues:

$\Lambda_1 = 9.61 \times 10^3$ sec^{-2}

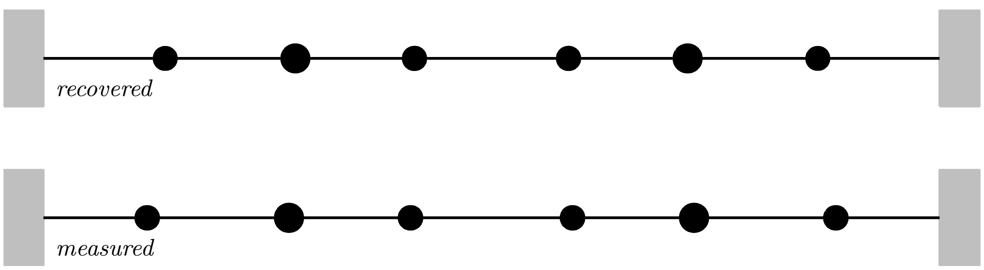
$\Lambda_2 = 3.10 \times 10^4$ sec^{-2}

$\Lambda_3 = 9.32 \times 10^4$ sec^{-2}

$\Lambda_4 = 1.40 \times 10^5$ sec^{-2}

$\Lambda_5 = 1.59 \times 10^5$ sec^{-2}

$\Lambda_6 = 1.92 \times 10^5$ sec^{-2}



	M_1 (g)	M_2 (g)	M_3 (g)	L_0 (cm)	L_1 (cm)	L_2 (cm)	L_3 (cm)
recovered	16.6	30.4	17.1	15.3	16.3	14.9	19.3
measured	17.8	30.8	17.8	13.0	17.8	15.2	20.3

FIG. 6.2. *Repetition of Figure 6.1, but now for a string with six beads.*

asymptotically small vibrations, but the amplitude of our vibrations must be large enough to be measured by the photodetector. Given this litany of errors, you might be surprised at the accuracy achieved in the experiments described in §4!

There is one further source of error that merits consideration, particularly when we consider applying this experiment to a string with many beads. The numerical algorithm detailed in §4 will incur rounding errors when implemented in floating point computer arithmetic. Roughly speaking, the algorithm in 4.1 can exhibit significant errors when $n > 50$, in MATLAB's default double-precision arithmetic. (One should also scale the units appropriately, as physical values such as $\tau = 10^7$ dyn can lead to overflow as n get large). Readers can explore this instability by computing eigenvalues for hypothetical symmetric strings using the $\mathbf{K}$ and $\mathbf{M}$ matrices from §2 (use `eig(K,M)` in MATLAB), then feeding this "exact" data to the inversion algorithm. How accurately do you recover the lengths and masses that you started with? How does this accuracy depend on the number of beads, as n gets very large? Interested readers can study alternative algorithms such as the continued fractions approach (see [7]) and the Lanczos method (see [6, §4.2]).

Acknowledgments. We thank Sean Hardesty for much initial design and development of our laboratory's monochord, Fan Tao (J. d'Addario & Co.) and our colleague Stan Dodds for important experimental advice and encouragement, and the Mapes Piano Wire Company for their generous donation of supplies. We appreciate the comments, corrections, and good humor of students Anthony Austin, Sharmaine Jennings, Claire Krebs, Charlie Laubach, Jon Stanley, John Vogelgesang, and Will Vogelgesang, as well as the helpful suggestions of the referees, which significantly shaped this work.

REFERENCES

[1] S. S. Antman, *The simple pendulum is not so simple*, SIAM Review, 40 (1998), pp. 927–930.
[2] G. Borg, *Eine Umkehrung der Sturm–Liouvilleschen Eigenwertaufgabe*, Acta Math., 78 (1946), pp. 1–96.
[3] O. Boyko and V. Pivovarchik, *The inverse three-spectral problem for a Stieltjes string and the inverse problem with one-dimensional damping*, Inverse Problems, 24 (2008). 13 pp.
[4] W. L. Briggs and V. E. Henson, *The DFT: An Owner's Manual for the Discrete Fourier Transform*, SIAM, Philadelphia, 1995.
[5] J. T. Cannon and S. Dostrovsky, *The Evolution of Dynamics: Vibration Theory from 1687 to 1782*, Springer-Verlag, New York, 1981.
[6] M. T. Chu and G. H. Golub, *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications*, Oxford University Press, Oxford, 2005.
[7] S. J. Cox, M. Embree, and J. M. Hokanson, *Inverse eigenvalue experiments for beaded strings*, 2010. Web site: `http://www.comlab.ox.ac.uk/~beads`.
[8] C. de Boor and G. H. Golub, *The numerically stable reconstruction of a Jacobi matrix from spectral data*, Linear Algebra Appl., 21 (1978), pp. 245–260.
[9] H. Dym and H. P. McKean, *Gaussian Processes, Function Theory, and the Inverse Spectral Problem*, Academic Press, New York, 1976.
[10] F. R. Gantmacher and M. G. Krein, *Oscillation Matrices and Kernels and Small Vibrations of Mechanical Systems*, AMS Chelsea, Providence, RI, revised ed., 2002.
[11] I. M. Gelfand and B. M. Levitan, *On determination of a differential equation from its spectral function*, Izv. Acad. Nauk SSSR Ser. Mat., 15 (1951), pp. 309–360.
[12] G. M. L. Gladwell, *Inverse Problems in Vibration*, Kluwer, Dordrecht, second ed., 2004.
[13] B. Gopinath and M. M. Sondhi, *Determination of the shape of the human vocal tract by acoustic measurements*, Bell Sys. Tech. J., 49 (1970), pp. 1195–1214.
[14] B. Gutkin and U. Smilansky, *Can one hear the shape of a graph?*, J. Phys. A: Math. Gen., 34 (2001), pp. 6061–6068.
[15] M. Kac, *Can one hear the shape of a drum?*, Amer. Math. Monthly, 73 (1966), pp. 1–23.
[16] M. G. Krein, *On some cases of effective determination of the density of an inhomogeneous cord from its spectral function*, Doklady Akad. Nauk SSR (N.S.), 93 (1953), pp. 617–620.
[17] N. Levinson, *The inverse Sturm–Liouville problem*, Math. Tidsskr. B, (1949), pp. 25–30.
[18] H.-P. Lin, *Direct and inverse methods on free vibration analysis of simply supported beams with a crack*, Eng. Structures, 26 (2004), pp. 427–436.
[19] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.
[20] T. Sekii and H. Shibahashi, *Inverse problem of solar oscillations*, in Stellar Pulsation, no. 274 in Lecture Notes in Physics, Springer, New York, 1987, pp. 322–325.
[21] T.-J. Stieltjes, *Recherches sur les fractions continues*, Ann. Fac. Sci. Toulouse, Series 1, 8 (1894), pp. J1–J122. 9 (1895), pp. A5–A47. English translation in Thomas Jan Stieltjes, *Collected Papers*, vol. II, Gerrit van Dijk, ed., pages 609–745, Springer-Verlag, Berlin, 1993.
[22] G. Strang, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Wellesley, MA, 1986.
[23] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, SIAM, Philadelphia, second ed., 2004.
[24] C. Truesdell, *The Rational Mechanics of Flexible or Elastic Bodies, 1638–1788*, Leonhardi Euleri Opera Omnia, Introduction to Volumes X and XI, Second Series, Orell Füssli, Zürich, 1960.

Appendix A. Determining Mass and Position from Vibrations using Continued Fractions. As an alternative to the orthogonal polynomial method described in Section 4, we here derive the lengths and masses of a beaded string using Krein's continued fractions approach. This algorithm is a bit more direct than the orthogonal polynomial method presented earlier; for example, we do not explicitly reconstruct the entries of the matrix $\mathbf{A}_n$, but rather go straight for the lengths and masses of interest. Our presentation explicates the method described in Supplement II of the recently revised English edition of Gantmacher and Krein's classic text on *Oscillation Matrices* [10].

A.1. Shooting functions. In §2 we saw that bead vibrations were governed by the eigenvalues and eigenvectors of the pair $(\mathbf{K},\mathbf{M})$. Here we begin by first building two important sets of polynomials that will reveal how the eigenvalues relate to the bead masses and locations. Now notice that we can rearrange equation (2.5) into the form

$$v_{j+1} = \Big(-\frac{\ell_j}{\ell_{j-1}}\Big)v_{j-1} + \Big(1+\frac{\ell_j}{\ell_{j-1}} - \frac{\lambda \ell_j m_j}{\tau}\Big)v_j. \tag{A.1}$$

We can use this equation (with $j=1$ and $v_0=0$) to produce a formula for v_2 in terms of v_1, then v_3 in terms of v_1, and so on. As any desired nonzero value for v_1 merely scales the entire eigenvector, we will set $v_1=1$. Notice that we have created a mechanism to construct eigenvectors one entry at a time, *provided the value of* λ *used in* (A.1) *is an eigenvalue.*

What goes wrong if λ is *not* an eigenvalue, but we still use the recurrence (A.1) to generate an "eigenvector"? The process proceeds uneventfully until we compute v_{n+1}. The fact that our string is fixed at the right end requires that $v_{n+1}=0$, but for arbitrary values of λ, (A.1) will generally produce a nonzero value for v_{n+1}, indicating that λ is not a true eigenvalue and $\mathbf{v}=[v_1,\ldots,v_n]^T$ is not a true eigenvector of $(\mathbf{K},\mathbf{M})$. If we compute $v_{n+1}=0$ from (A.1), then the value of λ we used *must* be an eigenvalue.

This suggests a procedure for computing eigenvalues known as the *shooting method*: assume initial values $v_0=0$ and $v_1=1$, and adjust λ until (A.1) produces $v_{n+1}=0$. (The name is inspired by the act of progressively adjusting the angle of a gun barrel to zero a projectile in on a target.) It would be tedious to hunt for eigenvalues by guessing values of λ; a more systematic procedure follows from writing the eigenvector entries in terms of polynomials. When $j=1$, the formula (A.1) implies

$$\begin{aligned} v_2 &= \Big(-\frac{\ell_1}{\ell_0}\Big)v_0 + \Big(1+\frac{\ell_1}{\ell_0}-\frac{\lambda\ell_1 m_1}{\tau}\Big)v_1 \\ &= \Big(1+\frac{\ell_1}{\ell_0}-\frac{\lambda\ell_1 m_1}{\tau}\Big)v_1 \\ &= p_1(\lambda), \end{aligned}$$

where

$$p_1(\lambda) := \Big(1+\frac{\ell_1}{\ell_0}\Big) - \lambda\Big(\frac{\ell_1 m_1}{\tau}\Big)$$

is a *linear* polynomial in λ. Similarly, we have

$$v_3 = \Big(-\frac{\ell_2}{\ell_1}\Big)v_1 + \Big(1+\frac{\ell_2}{\ell_1}-\frac{\lambda\ell_2 m_2}{\tau}\Big)v_2$$

$$= \Big(-\frac{\ell_2}{\ell_1}\Big)v_1 + \Big(1 + \frac{\ell_2}{\ell_1} - \frac{\lambda \ell_2 m_2}{\tau}\Big)p_1(\lambda)v_1$$
$$= p_2(\lambda),$$

where

$$p_2(\lambda) := \Big(-\frac{\ell_2}{\ell_1}\Big) + \Big[\Big(1 + \frac{\ell_2}{\ell_1}\Big) - \lambda\Big(\frac{\ell_2 m_2}{\tau}\Big)\Big]p_1(\lambda),$$

which is a *quadratic* polynomial.

We can continue this process for the subsequent entries in the eigenvector:

$$\begin{aligned} v_{j+1} &= \Big(-\frac{\ell_j}{\ell_{j-1}}\Big)v_{j-1} + \Big(1 + \frac{\ell_j}{\ell_{j-1}} - \frac{\lambda \ell_j m_j}{\tau}\Big)v_j \\ &= \Big(-\frac{\ell_j}{\ell_{j-1}}\Big)p_{j-2}(\lambda)v_1 + \Big(1 + \frac{\ell_j}{\ell_{j-1}} - \frac{\lambda \ell_j m_j}{\tau}\Big)p_{j-1}(\lambda)v_1 \\ &= p_j(\lambda), \end{aligned}$$

where

$$p_j(\lambda) := \Big(-\frac{\ell_j}{\ell_{j-1}}\Big)p_{j-2}(\lambda) + \Big[\Big(1 + \frac{\ell_j}{\ell_{j-1}}\Big) - \lambda\Big(\frac{\ell_j m_j}{\tau}\Big)\Big]p_{j-1}(\lambda). \tag{A.2}$$

(Note that p_j is a degree-j polynomial, and that this recurrence works also for $j = 2$ if we define $p_0(\lambda) = 1$ for all λ.) Continuing this process, we eventually arrive at

$$v_{n+1} = p_n(\lambda),$$

where p_n is a polynomial of degree n called the *shooting function*. We see that

$$v_{n+1} = 0 \qquad \text{if and only if} \qquad p_n(\lambda) = 0.$$

In other words

$$\lambda \text{ is an eigenvalue of } (\mathbf{K}, \mathbf{M}) \qquad \text{if and only if} \qquad p_n(\lambda) = 0.$$

Since p_n is a polynomial of degree n, it has precisely n roots: up to a scaling, p_n is the characteristic polynomial $\det(\mathbf{K} - \lambda\mathbf{M})$. However, if we want to build p_n using (A.2), *we must know the lengths* $\{\ell_j\}_{j=0}^n$ *and masses* $\{m_j\}_{j=1}^n$.

By contrast, the inverse problem presents us with known (experimentally measured) eigenvalues; we need another way to build p_n without *a priori* knowledge of lengths and masses. Here is the key: If we know eigenvalues $\lambda_1, \ldots, \lambda_n$, we can build p_n from its roots, up to some constant factor, σ:

$$p_n(\lambda) = \sigma \prod_{j=1}^{n} (\lambda - \lambda_j).$$

We can see from (A.2) that the coefficients of p_n must be rich in information about the lengths and masses. Can we use these coefficients, which we build from our measured eigenvalues, to obtain formulas for those embedded lengths and masses?

In general, the answer is *no*—which should not be surprising after a moment's reflection: we are trying to extract $2n+1$ independent pieces of information (n masses and $n+1$ lengths) from $n+1$ pieces of data (n eigenvalues and the total length, ℓ). To make the problem well determined, we need n more pieces of data.

What extra data might we practically obtain? It shall ultimately prove most convenient for us to use another set of eigenvalues from a closely related problem: keep the masses and lengths the same and the left end fixed, but now allow the right end to move in such a way that the string has *zero slope* there, i.e., $v_{n+1} = v_n$. You might naturally wonder how a condition could be reliably implemented in the lab (e.g., by attaching the right end of the string to a ring mounted, with no friction, to a vertical pole...), but suspend disbelief for the moment. (We will eventually find a clever way to get this data, at least for one class of bead configurations.) To differentiate this new scheme, we call it the *fixed–flat* string, to contrast with the usual *fixed–fixed* string.

From the "zero slope" condition on the last segment of the string, we will derive a polynomial recurrence akin to (A.1) for the slopes of the string between successive beads, and the roots of the ultimate polynomial will correspond to those values of λ for which the eigenvector $\mathbf{v}$ satisfies the zero slope condition on the right end.

To start, rearrange (A.1) to obtain a relation between consecutive slopes:

$$\frac{v_{j+1}-v_j}{\ell_j} = \frac{v_j - v_{j-1}}{\ell_{j-1}} - \Big(\frac{\lambda m_j}{\tau}\Big) v_j. \tag{A.3}$$

We can use the polynomials $\{p_j\}$ to write the left hand side of (A.3) as

$$\frac{v_{j+1}-v_j}{\ell_j} = \frac{p_j(\lambda) - p_{j-1}(\lambda)}{\ell_j}.$$

The polynomial on the right must be of degree at most j, since both p_j and p_{j-1} are both of degree-j or less. Define this polynomial to be

$$q_j(\lambda) := \frac{1}{\ell_j}\big(p_j(\lambda) - p_{j-1}(\lambda)\big), \tag{A.4}$$

for $j = 1, \ldots, n$, so that $q_j(\lambda)$ denotes the slope of the segment of string between beads j and $j+1$. The right hand side of (A.3) can then be written as

$$\frac{v_j - v_{j-1}}{\ell_{j-1}} - \Big(\frac{\lambda m_j}{\tau}\Big) v_j = q_{j-1}(\lambda) - \Big(\frac{\lambda m_j}{\tau}\Big) p_{j-1}(\lambda).$$

Equating our expressions from the left and right sides of (A.3), we obtain

$$q_j(\lambda) = q_{j-1}(\lambda) - \Big(\frac{\lambda m_j}{\tau}\Big) p_{j-1}(\lambda). \tag{A.5}$$

A similar relation follows from rearranging the definition of the slope polynomials, (A.4):

$$p_j(\lambda) = \ell_j q_j(\lambda) + p_{j-1}(\lambda). \tag{A.6}$$

These two equations, (A.5) and (A.6), form the fundamental tools we use to solve the inverse problem. For the fixed–flat string, we must have

$$q_n(\lambda) = 0,$$

and hence λ must be a root of the polynomial q_n. Thus, measurements of the n eigenvalues of the fixed–flat string allow us to construct q_n up to a scaling factor.

A.2. Continued Fractions. We are now prepared to determine the material properties of the beaded string from the polynomials p_n and q_n. Using the recurrences for the displacement and slope polynomials, we have

$$\begin{aligned}
\frac{p_n(\lambda)}{q_n(\lambda)} &= \frac{\ell_n q_n(\lambda) + p_{n-1}(\lambda)}{q_n(\lambda)} && \text{[by (A.6)]} \\
&= \ell_n + \frac{p_{n-1}(\lambda)}{q_n(\lambda)} \\
&= \ell_n + \cfrac{1}{\cfrac{q_n(\lambda)}{p_{n-1}(\lambda)}} \\
&= \ell_n + \cfrac{1}{\cfrac{-(m_n/\tau)\lambda p_{n-1}(\lambda) + q_{n-1}(\lambda)}{p_{n-1}(\lambda)}} && \text{[by (A.5)]} \\
&= \ell_n + \cfrac{1}{-(m_n/\tau)\lambda + \cfrac{1}{\cfrac{p_{n-1}(\lambda)}{q_{n-1}(\lambda)}}} && \text{(A.7)} \\
&= \ell_n + \cfrac{1}{-(m_n/\tau)\lambda + \cfrac{1}{\cfrac{\ell_{n-1} q_{n-1}(\lambda) + p_{n-2}(\lambda)}{q_{n-1}(\lambda)}}} && \text{[by (A.6)]} \\
&\vdots \\
&= \ell_n + \cfrac{1}{-(m_n/\tau)\lambda + \cfrac{1}{\ell_{n-1} + \cfrac{1}{-(m_{n-1}/\tau)\lambda + \cdots + \cfrac{1}{\ell_1 + \cfrac{1}{-(m_1/\tau)\lambda + \cfrac{1}{\ell_0}}}}}}, && \text{(A.8)}
\end{aligned}$$

a *continued fraction* for the rational function p_n/q_n. From this beautiful decomposition we can simply read off the masses and string lengths.

To summarize: the eigenvalues of the original (fixed–fixed) system give us the polynomial p_n up to a scaling factor; the eigenvalues of fixed–flat system provide q_n up to a scaling factor. To construct the continued fraction decomposition (A.8), we used knowledge of the subordinate polynomials p_{n-1}, q_{n-1}, ...: unfortunately, as these polynomials were built from knowledge of the masses and lengths, we cannot immediately produce them directly from the two sets of eigenvalues. Hence we seek some method for computing the continued fraction decomposition (A.8) that does not require knowledge of p_{n-1}, q_{n-1}, etc. If we had some other way to construct this decomposition, we could simply read off the values of ℓ_j and m_j/τ. These would be systematically incorrect by a scaling factor inherited from p_n and q_n, but that can be resolved from our knowledge of the total length ℓ: we must have $\sum_{j=0}^{n} \ell_j = \ell$. Our next goal is to determine an algorithm that will deliver the continued fraction decomposition.

A.3. A Recipe for Recovering Material Parameters from Polynomials. Suppose that we have measured the eigenvalues of the string with both ends fixed,

$$\lambda_1 < \lambda_2 < \cdots < \lambda_n, \qquad \textit{fixed–fixed}$$

and the eigenvalues of the string with fixed left end and a zero-slope on the right,

$$\widehat{\lambda}_1 < \widehat{\lambda}_2 < \cdots < \widehat{\lambda}_n, \qquad \textit{fixed–flat.}$$

With these eigenvalues, we can construct a pair of degree-n polynomials,

$$a_n(\lambda) = \prod_{j=1}^{n} (\lambda - \lambda_j) = \lambda^n + \alpha_1 \lambda^{n-1} + \alpha_2 \lambda^{n-2} + \cdots + \alpha_{n-1}\lambda + \alpha_n \tag{A.9}$$

$$b_n(\lambda) = \prod_{j=1}^{n} (\lambda - \widehat{\lambda}_j) = \lambda^n + \beta_1 \lambda^{n-1} + \beta_2 \lambda^{n-2} + \cdots + \beta_{n-1}\lambda + \beta_n. \tag{A.10}$$

Because a_n and b_n have a coefficient of one multiplying the leading term λ^n, they are called *monic polynomials*. These polynomials equal p_n and q_n up to constants, and so the ratio a_n/b_n differs from p_n/q_n only by a constant, which we shall call σ_n:

$$\sigma_n \frac{p_n(\lambda)}{q_n(\lambda)} = \frac{a_n(\lambda)}{b_n(\lambda)}.$$

Our goal is to determine a continued fraction decomposition of a_n/b_n that will expose the material properties of the beaded string. In particular, we seek to write the ratio of the degree-n monic polynomials a_n and b_n as continued fractions involving degree $n-1$ monic polynomials a_{n-1} and b_{n-1}. To do so, we seek ξ_n and σ_{n-1} such that

$$\sigma_n \frac{p_n(\lambda)}{q_n(\lambda)} = \frac{a_n(\lambda)}{b_n(\lambda)} = 1 + \cfrac{1}{-\xi_n \lambda + \cfrac{1}{\sigma_{n-1}^{-1} \dfrac{a_{n-1}(\lambda)}{b_{n-1}(\lambda)}}}.$$

Once we know how to compute this decomposition, we can apply the same ideas to the ratio a_{n-1}/b_{n-1}, and so on, until we arrive at degree-0 *monic* polynomials a_0 and b_0, which means that $a_0(\lambda) = 1$ and $b_0(\lambda) = 1$. At this stage, we will have a decomposition that matches (A.8) up to a constant. An algorithm for computing this factorization is given in Figure A.1.

A.4. Derivation of the Continued Fraction Decomposition of a_n/b_n. We shall now explain the origins of the algorithm in Figure A.1, following [10]. Inspired by the formula (A.7), we would like to write

$$\frac{a_n(\lambda)}{b_n(\lambda)} = \nu_n + \cfrac{1}{-\xi_n \lambda + \cfrac{1}{\dfrac{\widehat{a}_{n-1}(\lambda)}{\widehat{b}_{n-1}(\lambda)}}}$$

for some scalars ξ_n and ν_n, and polynomials $\widehat{a}_{n-1}$ and $\widehat{b}_{n-1}$, each of degree $n-1$. Toward this end, algebraically manipulate the right hand side to obtain the equivalent expression

$$\frac{a_n(\lambda)}{b_n(\lambda)} = \nu_n + \frac{\widehat{a}_{n-1}(\lambda)}{\widehat{b}_{n-1}(\lambda) - \xi_n \lambda \widehat{a}_{n-1}(\lambda)}. \tag{A.11}$$

A note about notation: $\mathbf{a}_1$ refers to the first element of the vector $\mathbf{a}$, $\mathbf{a}_{2:k}$ refers to vector consisting of the second through kth elements of $\mathbf{a}$, and similarly for $\mathbf{b}_1$ and $\mathbf{b}_{2:k}$.

Construct vectors $\mathbf{a} := [\alpha_1, \alpha_2, \ldots, \alpha_n]$ and $\mathbf{b} := [\beta_1, \beta_2, \ldots, \beta_n]$ from eigenvalue data via equations (A.9) and (A.10).

for $k := n, n-1, \ldots, 1$
    $\xi_k := 1/(\mathbf{b}_1 - \mathbf{a}_1)$
    $\mathbf{a} := \mathbf{a} - \mathbf{b}$
    $\mathbf{b} := \mathbf{b} + \xi_k[\mathbf{a}_{2:k}\ 0]$
    $\sigma_{k-1} := \mathbf{b}_1/\mathbf{a}_1$
    $\mathbf{a} := \mathbf{a}_{2:k}/\mathbf{a}_1$
    $\mathbf{b} := \mathbf{b}_{2:k}/\mathbf{b}_1$
end

$$\sigma_n := \frac{1}{\ell}\Big(1 + \frac{1}{\sigma_{n-1}} + \frac{1}{\sigma_{n-2}\sigma_{n-1}} + \cdots + \frac{1}{\sigma_0\sigma_1\cdots\sigma_{n-1}}\Big)$$

$\ell_n := 1/\sigma_n$
for $k := n, n-1, \ldots, 1$
    $\ell_{k-1} := \ell_k/\sigma_{k-1}$
    $m_k := \tau\xi_k/\ell_k$
end

FIG. A.1. *Algorithm for the recovery of the masses and locations of n beads from fixed–fixed and fixed–flat eigenvalues.*

Our first goal is to determine ν_n. Since a_n and b_n are both monic, degree-n polynomials, they each behave like λ^n when $|\lambda|$ is very large. Thus the left hand side of (A.11) must tend to 1 as $\lambda \to \infty$, so the same must be true of the right hand side. Given that $\widehat{a}_{n-1}$ and $\widehat{b}_{n-1}$ are of degree $n-1$, the second term on the right of (A.11) goes to zero as $\lambda \to \infty$. We thus conclude

$$\nu_n = 1.$$

With $\nu_n = 1$, we can manipulate (A.11) into the form

$$\frac{a_n(\lambda) - b_n(\lambda)}{b_n(\lambda)} = \frac{\widehat{a}_{n-1}(\lambda)}{\widehat{b}_{n-1}(\lambda) - \xi_n\lambda\widehat{a}_{n-1}(\lambda)}.$$

Equating the numerators and denominators, this formula suggests that we *define*

$$\widehat{a}_{n-1}(\lambda) := a_n(\lambda) - b_n(\lambda)$$

$$\widehat{b}_{n-1}(\lambda) := b_n(\lambda) + \xi_n\lambda\widehat{a}_{n-1}(\lambda).$$

Do these definitions satisfy our demand that both $\widehat{a}_{n-1}$ and $\widehat{b}_{n-1}$ be degree $n-1$ polynomials? Since a_n and b_n have λ^n as a common leading term, the definition of $\widehat{a}_{n-1}$ ensures that its λ^n coefficient is zero, and hence $\widehat{a}_{n-1}$ has degree $n-1$. Now we check $\widehat{b}_{n-1}$:

$$\widehat{b}_{n-1}(\lambda) \;=\; b_n(\lambda) + \xi_n\lambda\left(a_n(\lambda) - b_n(\lambda)\right)$$

$$
\begin{aligned}
&= \big(\lambda^n + \beta_1\lambda^{n-1} + \cdots + \beta_n\big) \\
&\qquad\qquad + \xi_n\lambda\Big((\alpha_1-\beta_1)\lambda^{n-1} + (\alpha_2-\beta_2)\lambda^{n-2} + \cdots + (\alpha_n-\beta_n)\Big) \\
&= \Big(1+\xi_n(\alpha_1-\beta_1)\Big)\,\lambda^n + \Big(\beta_1+\xi_n(\alpha_2-\beta_2)\Big)\,\lambda^{n-1} \\
&\qquad\qquad + \cdots + \Big(\beta_{n-1}+\xi_n(\alpha_n-\beta_n)\Big)\,\lambda + \beta_n.
\end{aligned}
$$

This polynomial will have degree $n-1$ provided the coefficient of λ^n is zero, which we can ensure by assigning the only remaining free parameter to be

$$\xi_n := \frac{1}{\beta_1-\alpha_1}.$$

Hence we have a recipe for finding scalar ξ_n and degree $n-1$ polynomials $\widehat{a}_{n-1}$ and $\widehat{b}_{n-1}$ so that

$$\sigma_n\frac{p_n(\lambda)}{q_n(\lambda)} \;=\; \frac{a_n(\lambda)}{b_n(\lambda)} \;=\; 1 + \cfrac{1}{-\xi_n\lambda + \cfrac{1}{\cfrac{\widehat{a}_{n-1}(\lambda)}{\widehat{b}_{n-1}(\lambda)}}}.$$

To get the full continued fraction that reveals all the lengths and masses as in (A.8), we would like to apply this procedure recursively to $\widehat{a}_{n-1}/\widehat{b}_{n-1}$, but one small adjustment is needed. Recall that we started with polynomials a_n and b_n that had leading term λ^n; in general, $\widehat{a}_{n-1}$ and $\widehat{b}_{n-1}$ will have nontrivial coefficients multiplying λ^{n-1}:

$$
\begin{aligned}
\widehat{a}_{n-1}(\lambda) &= \Big(\alpha_1-\beta_1\Big)\,\lambda^{n-1} + \cdots + \Big(\alpha_{n-1}-\beta_{n-1}\Big)\,\lambda + \Big(\alpha_n-\beta_n\Big) \\
\widehat{b}_{n-1}(\lambda) &= \Big(\beta_1+\xi_n(\alpha_2-\beta_2)\Big)\,\lambda^{n-1} + \cdots + \Big(\beta_{n-1}+\xi_n(\alpha_n-\beta_n)\Big)\,\lambda + \beta_n,
\end{aligned}
$$

and these leading coefficients must be scaled out. Define a constant to capture the ratio of these coefficients,

$$\sigma_{n-1} := \frac{\beta_1+\xi_n(\alpha_2-\beta_2)}{\alpha_1-\beta_1},$$

so that if we define

$$
\begin{aligned}
a_{n-1}(\lambda) &:= \frac{\widehat{a}_{n-1}(\lambda)}{\alpha_1-\beta_1} \\
&= \lambda^{n-1} + \frac{\alpha_2-\beta_2}{\alpha_1-\beta_1}\lambda^{n-2} + \cdots + \frac{\alpha_n-\beta_n}{\alpha_1-\beta_1}
\end{aligned}
\tag{A.12}
$$

$$
\begin{aligned}
b_{n-1}(\lambda) &:= \frac{\widehat{b}_{n-1}(\lambda)}{\beta_1+\xi_n(\alpha_2-\beta_2)} \\
&= \lambda^{n-1} + \frac{\beta_2+\xi_n(\alpha_3-\beta_3)}{\beta_1+\xi_n(\alpha_2-\beta_2)}\,\lambda^{n-2} + \cdots \\
&\qquad \cdots + \frac{\beta_{n-1}+\xi_n(\alpha_n-\beta_n)}{\beta_1+\xi_n(\alpha_2-\beta_2)}\,\lambda + \frac{\beta_n}{\beta_1+\xi_n(\alpha_2-\beta_2)},
\end{aligned}
\tag{A.13}
$$

then

$$\sigma_{n-1}\frac{\widehat{a}_{n-1}(\lambda)}{\widehat{b}_{n-1}} = \frac{a_{n-1}(\lambda)}{b_{n-1}(\lambda)}.$$

Using this expression, our evolving continued fraction takes the form

$$\sigma_n\frac{p_n(\lambda)}{q_n(\lambda)} = \frac{a_n(\lambda)}{b_n(\lambda)} = 1+\cfrac{1}{-\xi_n\lambda+\cfrac{1}{\sigma_{n-1}^{-1}\cfrac{a_{n-1}(\lambda)}{b_{n-1}(\lambda)}}}.$$

Note that the coefficients of a_{n-1} and b_{n-1} can be extracted directly from those of a_n and b_n by way of (A.12) and (A.13).

Now apply the procedure we have just described to a_{n-1}/b_{n-1} to obtain the decomposition

$$\frac{a_{n-1}(\lambda)}{b_{n-1}(\lambda)} = 1+\cfrac{1}{-\xi_{n-1}\lambda+\cfrac{1}{\sigma_{n-2}^{-1}\cfrac{a_{n-2}(\lambda)}{b_{n-2}(\lambda)}}},$$

and continue this process until one finally arrives at $a_0(\lambda)/b_0(\lambda)$: this term must be trivial, since a_0 and b_0 are both monic and constant, i.e., $a_0(\lambda)=1$ and $b_0(\lambda)=1$ for all λ.

Before we proceed to a concrete example, we should address several fine points that might have crossed your mind. (1) Can we be certain that this procedure does not *break down*? That is, can we be sure that we never encounter $\sigma_j=0$, which would cause division by zero? (2) Are the choices for ξ_j and σ_j unique, or are there other equivalent continued fraction decompositions of p_n/q_n with different constants? For our beaded strings, these issues will never arise: the procedure never breaks down, and the decomposition is unique. The details are beyond our scope here, but interested readers can pursue the issue in the book of Gantmacher and Krein [10] (Supplement II, especially the lemma on page 284). You might also wonder how stable this procedure in the presence of errors; we address this important point in §7.

A.5. Special case: $n=2$. In the case of a string with two beads, we twice apply the procedure outlined above to obtain

$$\sigma_2\frac{p_2(\lambda)}{q_2(\lambda)} = \frac{a_2(\lambda)}{b_2(\lambda)} = 1+\cfrac{1}{-\xi_2\lambda+\cfrac{1}{\sigma_1^{-1}+\cfrac{1}{-\xi_1\sigma_1\lambda+\cfrac{1}{\sigma_0^{-1}\sigma_1^{-1}}}}}.$$

Dividing through by the as yet *unknown* constant σ_2, we obtain

$$\frac{p_2(\lambda)}{q_2(\lambda)} = \sigma_2^{-1}+\cfrac{1}{-\xi_2\sigma_2\lambda+\cfrac{1}{\sigma_1^{-1}\sigma_2^{-1}+\cfrac{1}{-\xi_1\sigma_1\sigma_2\lambda+\cfrac{1}{\sigma_0^{-1}\sigma_1^{-1}\sigma_2^{-1}}}}}. \tag{A.14}$$

Comparing this term-by-term with the formula (A.8), we identify

$$\ell_2 = \frac{1}{\sigma_2}, \qquad \ell_1 = \frac{1}{\sigma_1\sigma_2}, \qquad \ell_0 = \frac{1}{\sigma_0\sigma_1\sigma_2},$$

$$m_2 = \tau\xi_2\sigma_2, \qquad m_1 = \tau\xi_1\sigma_1\sigma_2.$$

Even this simple case provides enough clues for you to guess general formulas for $n > 2$.

Given the eigenvalues $\lambda_1, \lambda_2, \widehat{\lambda}_1$, and $\widehat{\lambda}_2$, we can determine the constants ξ_2, σ_1, ξ_1, and σ_0 through the procedure outlined above. Furthermore, we presume that we can measure the tension τ in the laboratory, and that we also know the total length of the string, $\ell = \ell_0 + \ell_1 + \ell_2$. This last expression finally gives a formula for the last remaining unknown, σ_2: since

$$\ell = \frac{1}{\sigma_2} + \frac{1}{\sigma_1\sigma_2} + \frac{1}{\sigma_0\sigma_1\sigma_2},$$

we conclude that

$$\sigma_2 = \frac{1}{\ell}\left(1 + \frac{1}{\sigma_1} + \frac{1}{\sigma_0\sigma_1}\right).$$

With σ_2 in hand, we can compute all the lengths and masses.