# Animating Elastic Rods with Sound

ESTON SCHWEICKART, Cornell University
DOUG L. JAMES, Stanford University
STEVE MARSCHNER, Cornell University

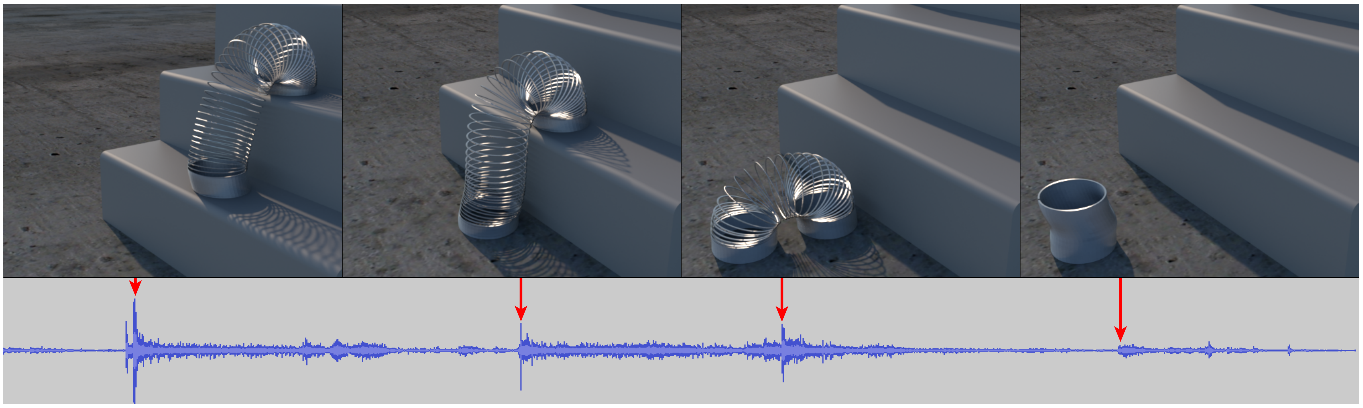

Fig. 1. **Audiovisual simulation of a spring toy slinking down the stairs:** Our method efficiently generates the spatialized sound of this discrete elastic rod (3888 control points) arising from tens of thousands of complex collision events each timestep.

Sound generation methods, such as linear modal synthesis, can sonify a wide range of physics-based animation of solid objects, resolving vibrations and sound radiation from various structures. However, elastic rods are an important computer animation primitive for which prior sound synthesis methods, such as modal synthesis, are ill-suited for several reasons: large displacements, nonlinear vibrations, dispersion effects, and the geometrically singular nature of rods.

In this paper, we present physically based methods for simultaneous generation of animation and sound for deformable rods. We draw on Kirchhoff theory to simplify the representation of rod dynamics and introduce a generalized dipole model to calculate the spatially varying acoustic radiation. In doing so, we drastically decrease the amount of precomputation required (in some cases eliminating it completely), while being able to resolve sound radiation for arbitrary body deformations encountered in computer animation. We present several examples, including challenging scenes involving thousands of highly coupled frictional contacts.

CCS Concepts: • **Computing methodologies → Physical simulation**; Collision detection;

Additional Key Words and Phrases: physically based animation, sound synthesis

**ACM Reference format:**
Eston Schweickart, Doug L. James, and Steve Marschner. 2017. Animating Elastic Rods with Sound. *ACM Trans. Graph.* 36, 4, Article 115 (July 2017), 10 pages.

DOI: http://dx.doi.org/10.1145/3072959.3073680

## 1 INTRODUCTION

Sound is a pervasive part of our world. For virtual environments to be believable, the sounds of virtual objects must be accurately reproduced. In the field of computer graphics, the standard framework to compute the sound generated by solid objects is modal synthesis. These methods involve calculating the frequencies generated by the vibrating surface of the object, and computing the acoustic radiation caused by each frequency band. Modal models are popular because of their computational speed at runtime, their physically based origins, and their plausibility when simulating sound for near-rigid objects.

However, very thin rod-like structures are typically poor candidates for linear modal sound synthesis. Thin objects tend to undergo large displacements and change shape throughout the course of a simulation. This leads to vibration modes that change frequency over time, confounding precomputation, and breaks the assumption of near-rigid motion often used when computing acoustic radiation. For these reasons, simulating acoustic emissions from highly deformable bodies with arbitrary geometry is notoriously expensive. Within the standard linear modal synthesis pipeline, correctly calculating eigenmodes of thin objects requires a fine internal tetrahedral mesh, which is a memory-inefficient representation. Furthermore, dispersion, or vibrational waves traveling through the object at different speeds, can cause chirp-like sounds (like the iconic "blaster" sound) that are characteristic of long, thin objects such as ropes and cables. While it is possible to model this behavior using modal synthesis, it is much more elegantly captured with time-domain methods. Moreover, a good portion of the sound produced by animations of thin objects can be attributed to acceleration noise

0730-0301/2017/7-ART115 $15.00
DOI: http://dx.doi.org/10.1145/3072959.3073680

[Chadwick et al. 2012], which requires additional preprocessing and computation in the modal setting.

All these effects could be computed by a full finite element method simulation that tracks oscillations of the entire body in the time domain using a nonlinear internal force model that allows for large displacements. However, this approach also requires an internal mesh representation, is costly at runtime, and still requires a way to calculate sound radiation from the body; taken together, these could potentially increase memory and computation time requirements by an order of magnitude over modal synthesis.

We propose a method based on Kirchhoff rods to generate physically based sound for thin structures, which is memory efficient and captures changing modal frequencies and dispersion effects. Precomputation is orders of magnitude faster than modal synthesis, and in special cases (such as rods with a circular cross section) can be avoided altogether. Acceleration noise arises naturally from our method and does not require additional computation. Most significantly, our approach efficiently produces plausible sound for highly deformable objects that would otherwise be very costly to simulate.

We achieve these properties by making particular assumptions about the geometry of our object that allow us to simplify both the dynamics and the resulting acoustic radiation. By restricting ourselves to thin structures, we may use a finite-difference approach to simulate the object using relatively few degrees of freedom; we then use a dipole array model to simulate sound radiation due to the object's motion. Our primary contribution is to demonstrate the usefulness of these techniques over a broad range of situations and how they can be integrated into the kinds of 3D environments encountered in many graphics applications.

## 2 RELATED WORK

Simulating the deformations of thin, rod-like structures has been well explored. By assuming the degrees of freedom lie only on the centerline, significant efficiency is gained over more general finite element methods. Throughout computational physics, computer graphics, and computer music, two common models for capturing the dynamics of thin structures are the *Euler-Bernoulli* and *Timoshenko* theories of beams. Both these models express internal forces (such as those due to bending, twisting, and stretching) in terms of deflections from a rest state assuming a continuous, single dimensional representation of the body. The Timoshenko model is more general, as it accounts for the effects of rotational inertia and of shearing along the cross section, which the Euler-Bernoulli model ignores; as such, it is more suitable for rods which are short in comparison to the area of their cross section. However, these models are derived from linear elasticity and are only accurate for small displacements. Generalizations based on geometric nonlinearities exist for each: the *Kirchhoff* and *Cosserat* models generalize the Euler-Bernoulli and Timoshenko models respectively, and allow for greater physical accuracy under large displacements. These models can be discretized using finite difference, finite element, or modal analysis approaches depending on the application. Lang and Arnold [2012] discuss the differences between these theories, whereas a more thorough analysis of rod dynamics is found in [Antman 1973].

Within the field of computer graphics, a number of frameworks exist for simulating rods (most often ropes, strings, and hair strands) using these theories with a focus on large displacements and contact [Bergou et al. 2008; Bertails et al. 2006; Kugelstadt and Schömer 2016; Pai 2002; Spillmann and Teschner 2007; Umetani et al. 2014]. These methods capture the behavior of highly deformable thin structures to varying degrees of physical accuracy, but they focus on visual output and do not address sound synthesis. In this paper, we build upon [Bergou et al. 2010, 2008], though other physics-based rod dynamics frameworks could be used with our method of computing sound radiation. We caution that acceleration-based schemes are more likely to produce realistic sound than position-based schemes, such as [Kugelstadt and Schömer 2016], since our physical model for emitted acoustic pressure uses estimates of the rod's *jerk*, i.e., the third time derivative of position.

In the context of computer music, linear and nonlinear physically based rod models are used, for example, to synthesize sound from stringed instruments, pianos, and pitched percussion; see [Fletcher and Rossing 1998] and [Ducceschi and Bilbao 2016]. Beyond analytical models for simplified systems, more general simulation frameworks include finite difference methods [Bank and Sujbert 2005; Bilbao 2007, 2009], wave digital filters [Bilbao 2004], (banded) waveguide synthesis [Essl et al. 2004; Testa et al. 2004], modal synthesis schemes [Bilbao 2009], and Volterra series [Hélie and Roze 2008], sometimes in combination; for comparative discussions of these techniques see [Smith 2004; Välimäki et al. 2005]. In computer music applications, visual output is unnecessary, which justifies certain simplifications; for example, when modeling pitched percussion, it is often only necessary to track displacements in a single dimension. Additionally, waveguides and similar tools, although more efficient than finite difference methods, often do not lend themselves well to integration into 3D environments. Distributed contact between pairs of vibrating structures has been investigated in this literature, with a focus on preventing numerical instability; see, for example, [Bilbao and Torin 2015; Bilbao et al. 2015; Chatziioannou and van Walstijn 2015]. However, friction is nearly always ignored in these studies for simplicity. Finally, many of these methods assume the sound is radiating from a naturally straight, cylindrical body (or do not model any form of radiation), while we aim to synthesize sound from more general geometries, e.g., a Slinky.

Modal synthesis is the dominant strategy for synthesizing sound in the realm of computer animation. The method has been studied for decades [Takala and Hahn 1992], and been has improved upon many times [James et al. 2006; Li et al. 2015; O'Brien et al. 2002; van den Doel et al. 2001]. To resolve contact coupling and chattering effects, adaptive modal synthesis and contact solvers have been proposed [Zheng and James 2011], but they are still inherently limited to small displacements and linear modal sound.

Accurately and efficiently generating sound from highly deformable bodies remains a difficult challenge. Chadwick et al. [2009] investigated the simulation of nonlinear mode coupling for thin shells, though the overall deformation is still assumed to be small for the purposes of calculating radiation. Sound produced by crumpling of shells and cloth has been approximated using specialized data-driven techniques [An et al. 2012; Schreck et al. 2016], possibly with spatially localized use of piecewise rigid modal models [Cirio et al. 2016]. In contrast, our large-displacement dynamic and radiation rod models can be derived from first physical principles.

O'Brien et al. [2001] presented a method for synthesizing sound from time-domain deformable simulations. Like us, they observe the vibrational signal of elements along the surface of the body in the time domain. A ray-based sound propagation model was used which supports general surface motions, but is not consistent with the acoustic wave equation. In contrast, we are able to derive a consistent radiation model by exploiting acoustic compactness and other properties of slender rods.

Besides what is mentioned above, there is other work investigating sounds generated from slender structures; for instance, Coleman and Dill [1992] and McMillen and Goriely [2003] describe (planar) wave characteristics of Kirchhoff rods; Bilbao [2013] explores modeling reverberation from spring structures; and Dobashi et al. [2003] model sound generated by aerodynamic vortices shed by fast-moving rods. In engineering, Akay et al. [1983] devised methods for simulating the acoustics of beam structures. Our method is similar to theirs in spirit, as they also time-step vibrations and use a dipole model to estimate the resulting acoustic pressure. However, their radiation model emits the dipole pattern with uniform amplitude in response to any motion in the cross section plane, whereas we analyze the cross section to calculate the directionally varying acoustic response in this plane. More generally, we have diverging goals: Akay et al. focused on estimating sound from a single bar impulse in a controlled environment, whereas we address audiovisual simulation of large displacements and complex collisions.

## 3 ROD DYNAMICS MODEL

The core of our simulator is a model for dynamics of rods that is integrated through time to produce motion that drives the acoustic radiation model of §4. The degrees of freedom of our body are the positions of points along the centerline and twists that accumulate between them. Dynamics is then modeled based on Kirchhoff theory, specifically the Discrete Elastic Rods model proposed in [Bergou et al. 2008] and extended in [Bergou et al. 2010]. We have chosen this framework because of its efficiency and widespread use in graphics, and because it is a natural fit for sound synthesis.

### 3.1 Notation

We will follow the notation introduced in [Bergou et al. 2010]. A rod is defined by $n + 2$ control points connected sequentially by $n + 1$ edges; together these define the centerline of the rod. In the following, values associated with control points are indexed using subscripts, and values associated with edges are indexed using superscripts. Let $\boldsymbol{x}_i \in \mathbb{R}^3$ be the position of control point $i$, and let $\theta^j$ denote the twist of edge $j$. Let $\boldsymbol{q} = (\boldsymbol{x}_0^\top, \theta^0, \ldots, \boldsymbol{x}_n^\top, \theta^n, \boldsymbol{x}_{n+1}^\top)^\top \in \mathbb{R}^N$ be the generalized coordinates of the rod, where $N$, the total number of degrees of freedom, is equal to $3(n + 2) + (n + 1)$.

For each edge $\boldsymbol{e}^j = \boldsymbol{x}_{j+1} - \boldsymbol{x}_j$ we define a frame, $[\underline{\boldsymbol{d}}_1^j, \underline{\boldsymbol{d}}_2^j, \underline{\boldsymbol{d}}_3^j] \in SO(3)$ (where $\underline{\boldsymbol{d}}_3^j = \boldsymbol{e}^j/\|\boldsymbol{e}^j\|$), called the reference frame. These frames are updated using parallel transport through time (as described in [Bergou et al. 2010] §3) whenever $\boldsymbol{q}$ is updated. As part of this process, we record the twist $\underline{m}_i$ accumulated between each pair of reference frames $i - 1$ and $i$. We then define a material frame $[\boldsymbol{d}_1^j, \boldsymbol{d}_2^j, \boldsymbol{d}_3^j] \in SO(3)$ at each edge $\boldsymbol{e}^j$ as the rotation of the reference frame around $\underline{\boldsymbol{d}}_3^j$ by the angle $\theta^j$. See Figure 2.

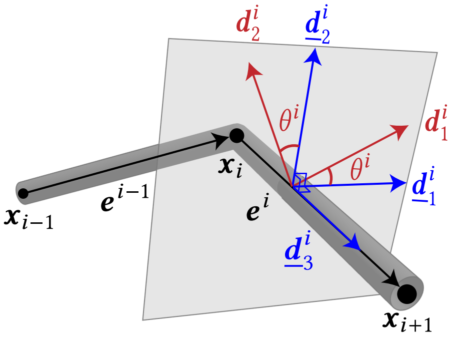

Fig. 2. Notation used within the definition of the rod model.

In addition to these definitions given by prior work, we require an estimate of the material frame at each of the control points in §4. Let $\boldsymbol{R}_i = [\boldsymbol{d}_{i,1},\ \boldsymbol{d}_{i,2},\ \boldsymbol{d}_{i,3}] \in SO(3)$ denote this frame at control point $i$. For the first and last control points, we simply use the material frame of the single neighboring edge; that is, $\boldsymbol{R}_0 = [\boldsymbol{d}_1^0, \boldsymbol{d}_2^0, \boldsymbol{d}_3^0]$ and $\boldsymbol{R}_{n+1} = [\boldsymbol{d}_1^n, \boldsymbol{d}_2^n, \boldsymbol{d}_3^n]$. For internal control points, we set $\boldsymbol{d}_{i,3} = (\boldsymbol{d}_3^{i-1}+\boldsymbol{d}_3^i)/\|\boldsymbol{d}_3^{i-1}+\boldsymbol{d}_3^i\|$. We then parallel transport $\underline{\boldsymbol{d}}_1^{i-1}$ and $\underline{\boldsymbol{d}}_2^{i-1}$ through space to obtain $\underline{\boldsymbol{d}}_{i,1}$ and $\underline{\boldsymbol{d}}_{i,2}$, which represent the approximate reference frame at control point $i$ and are orthogonal to $\boldsymbol{d}_{i,3}$. We then rotate this frame about $\boldsymbol{d}_{i,3}$ by the angle $(\theta^{i-1} + \theta^i + \underline{m}_i)/2$ to obtain $\boldsymbol{d}_{i,1}$ and $\boldsymbol{d}_{i,2}$.

### 3.2 Dynamics

To introduce dynamics, we make $\boldsymbol{q}$ a function of time, i.e., $\boldsymbol{q} = \boldsymbol{q}(t)$. Potential energies due to bending, twisting, and stretching (denoted $E_b$, $E_t$, and $E_s$ respectively) are non-negative scalars defined as functions of these degrees of freedom; see [Bergou et al. 2010] §4 for their definitions. From these values, we derive the internal forces $f_{int}(\boldsymbol{q}) \in \mathbb{R}^N$:

$$f_b(\boldsymbol{q}) = -\frac{\partial E_b}{\partial \boldsymbol{q}} \quad f_t(\boldsymbol{q}) = -\frac{\partial E_t}{\partial \boldsymbol{q}} \quad f_s(\boldsymbol{q}) = -\frac{\partial E_s}{\partial \boldsymbol{q}}$$

$$f_{int}(\boldsymbol{q}) = f_b(\boldsymbol{q}) + f_t(\boldsymbol{q}) + f_s(\boldsymbol{q})$$

From this, we can calculate the stiffness matrix of the structure $\boldsymbol{K}(\boldsymbol{q}) \in \mathbb{R}^{N\times N}$:

$$\boldsymbol{K}(\boldsymbol{q}) = \frac{\partial f_{int}(\boldsymbol{q})}{\partial \boldsymbol{q}}$$

We also use a diagonally lumped mass matrix $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ that remains constant throughout the simulation. Furthermore, we introduce a damping matrix $\boldsymbol{C} \in \mathbb{R}^{N\times N}$ and a velocity-proportional force $-\boldsymbol{C}\dot{\boldsymbol{q}}$ that provides artist-tunable damping to the simulation. As is common in sound synthesis applications, we choose to use a Rayleigh damping matrix [Shabana 2012], i.e., $\boldsymbol{C}(\boldsymbol{q}) = \alpha\boldsymbol{M} + \beta\boldsymbol{K}(\boldsymbol{q})$ for material- and geometry-dependent scalars $\alpha$ and $\beta$. Furthermore, we gather any external forces, such as gravity, into a vector $f_{ext}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^N$. This gives rise to the following equations of motion:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q})\dot{\boldsymbol{q}} - f_{int}(\boldsymbol{q}) = f_{ext}(\boldsymbol{q}, \dot{\boldsymbol{q}})$$

Various techniques can be used to integrate the rod through time. We prefer to use an implicit integration scheme for stability when integrating stiff materials such as metals. Backward Euler, the usual algorithm for implicit integration in graphics applications, introduces artificial damping; while this may be acceptable for purely visual simulations, this leads to unacceptably muted sound in our application. For this reason, we use the Newmark-$\beta$ algorithm with the trapezoidal rule (i.e., $\beta = 1/4$ and $\gamma = 1/2$) [Newmark 1959], which preserves long-term oscillatory behavior. This requires Newton iteration to solve for the generalized accelerations $\ddot{\boldsymbol{q}}(t_{k+1})$ given the state from the previous step $t_k$, from which we can deduce $\dot{\boldsymbol{q}}(t_{k+1})$ and $\boldsymbol{q}(t_{k+1})$. We assume $\boldsymbol{C}$ is constant throughout the timestep, which allows us to elide taking its derivative with respect to $\boldsymbol{q}(t_{k+1})$, which would otherwise require calculating a third-order tensor multiple times per timestep. We have not observed any audible artifacts or convergence problems attributed to this decision.

## 3.3 Caveats

The resulting sound is derived from small-amplitude, high-frequency vibrations arising from the rod model described above, and we can only expect accurate sound if the model adequately simulates the physics of the body. As a result, rods with large or highly eccentric cross sections may not be a good fit for our method, as they may exhibit audible modes and nonlinear effects that may require a full 3D solid or sheet model to properly resolve. In addition, for Kirchhoff rods, the effects of shearing across the cross section must be negligible—or artificial stiffening (higher frequencies) will occur—which implies geometrically that the cross-sectional diameter be small relative to rod length (which is generally true for our examples). More expensive Cosserat rod models could be used to account for shearing, and there is no particular reason that such a model could not be used within our sound radiation framework.

# 4 SOUND RADIATION MODEL

To turn the motions computed using the rod model into sound, we need a model suitable for efficiently estimating the acoustic radiation associated with rapid vibrations due to bending waves, as opposed to typically less audible stretching, twisting, and shearing motions. We now introduce a dipole radiation model associated with a small vibrating rod element, then discuss implementation details and related assumptions of the model.

## 4.1 Dipole Sound Field of a Vibrating Rod Element

*4.1.1 Introduction.* We approximate the sound field radiating from a rod as a linear superposition of contributions due to the motions of small segments along its length. We begin by considering a small rod element located at the origin $\boldsymbol{r} = \boldsymbol{0}$, oriented along the $z$-axis, with interior $\Omega$ and surface $S = \partial\Omega$ (see Figure 3). We assume this segment is moving rigidly, and denote its linear and angular velocities as $\boldsymbol{v}(t)$ and $\boldsymbol{\omega}(t)$ respectively.

Following the theory of linear acoustics, we seek to approximate the acoustic pressure field $p(\boldsymbol{r}, t)$, which satisfies the wave equation outside the body,

$$\frac{\partial^2 p}{\partial t^2}(\boldsymbol{r},t) = c^2 \nabla^2 p(\boldsymbol{r},t), \qquad \boldsymbol{r} \in \mathbb{R}^3 \backslash \Omega, \tag{1}$$

with suitable boundary conditions, where $c$ is the speed of sound in the surrounding medium. While this can be computationally involved for large bodies, we use the derivation in [Howe 2003] to approximate the far-field acoustic pressure from a compact, rigidly vibrating object as a dipole source:

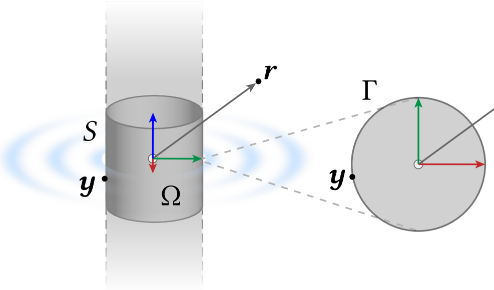

Fig. 3. (Left) A vibrating rod element depicted as part of a longer extruded rod. (Right) Cross-sectional geometry and notation.

$$p(\boldsymbol{r},t) = \frac{\rho}{4\pi c\, r}\,\hat{\boldsymbol{r}}^\top \frac{\partial^2}{\partial^2 t} \oint_S (\boldsymbol{y} - \boldsymbol{\varphi}^*(\boldsymbol{y}))\;\; v_n(\boldsymbol{y},\tau)\;\, dS_{\boldsymbol{y}} \tag{2}$$

where $r = \|\boldsymbol{r}\|$, $\hat{\boldsymbol{r}} = \boldsymbol{r}/r$, and $v_n(\boldsymbol{y}, \tau)$ is the surface normal velocity at point $\boldsymbol{y} \in S$ at retarded time $\tau = t - r/c$. Here $\boldsymbol{\varphi}^*(\boldsymbol{r}) \in \mathbb{R}^3$ is a time-independent harmonic function called the *velocity potential*, that satisfies

$$\nabla^2 \boldsymbol{\varphi}^*(\boldsymbol{r}) = \boldsymbol{0}, \qquad \boldsymbol{r} \in \mathbb{R}^3 \setminus \Omega, \tag{3}$$

$$\boldsymbol{\varphi}^*(\boldsymbol{r}) \to \boldsymbol{0}, \qquad \text{as } \|\boldsymbol{r}\| \to \infty, \tag{4}$$

$$\frac{\partial \boldsymbol{\varphi}^*(\boldsymbol{r})}{\partial \boldsymbol{n}} = \boldsymbol{n}, \qquad \boldsymbol{r} \in S \tag{5}$$

where $\boldsymbol{n}(\boldsymbol{y})$ is the unit normal on $S$.

The surface normal velocity (assuming the origin and the rod element's centroid coincide) is

$$v_n(\boldsymbol{y},\tau) = \boldsymbol{n}(\boldsymbol{y}) \cdot (\boldsymbol{v}(\tau) - \boldsymbol{y} \times \boldsymbol{\omega}(\tau)) \equiv \boldsymbol{N}(\boldsymbol{y})^\top \begin{pmatrix} \boldsymbol{v}(\tau) \\ \boldsymbol{\omega}(\tau) \end{pmatrix} \tag{6}$$

where $\boldsymbol{N}(\boldsymbol{y}) = \left(\begin{smallmatrix} \boldsymbol{n} \\ \boldsymbol{y}\times\boldsymbol{n} \end{smallmatrix}\right) \in \mathbb{R}^6$. Substituting (6) into (2), spatial integration can be performed independent of time:

$$\begin{aligned} p(\boldsymbol{r},t) &= \frac{\rho}{4\pi c\, r}\,\hat{\boldsymbol{r}}^\top \left[ \oint_S (\boldsymbol{y} - \boldsymbol{\varphi}^*(\boldsymbol{y}))\;\; \boldsymbol{N}(\boldsymbol{y})^\top\, dS_{\boldsymbol{y}} \right] \frac{d^2}{dt^2} \begin{pmatrix} \boldsymbol{v}(\tau) \\ \boldsymbol{\omega}(\tau) \end{pmatrix} \\ &= \frac{\rho}{4\pi c\, r}\,\hat{\boldsymbol{r}}^\top\; [\boldsymbol{D}^v\, \boldsymbol{D}^\omega] \begin{pmatrix} \ddot{\boldsymbol{v}}(\tau) \\ \ddot{\boldsymbol{\omega}}(\tau) \end{pmatrix} \end{aligned} \tag{7}$$

where $\boldsymbol{D}^v$ and $\boldsymbol{D}^\omega$ are $3 \times 3$ matrices that depend only on the shape of the element. Note that we have assumed a slow-moving listener, i.e., $\dot{\tau} \approx 1$.

*4.1.2 Symmetry and rotations.* Due to the properties of $\boldsymbol{\varphi}^*(\boldsymbol{y})$, it can be shown that for principal rotations about the centroid of the rod element, all entries of $\boldsymbol{D}^\omega$ vanish for elements with $xyz$ mirror symmetries.[1] Without loss of generality, we will consider rods with symmetric cross sections that only change shape smoothly along the rod, and therefore we can use the simpler pure-translational dipole model [Howe 2003],

$$p(\boldsymbol{r},t) = \frac{\rho}{4\pi c\, r}\,\hat{\boldsymbol{r}}^\top \boldsymbol{D}\, \ddot{\boldsymbol{v}}(\tau) \tag{8}$$

[1] See Appendix A for a proof. A nonzero angular contribution can still arise as the element rotates about the $z$-axis, but it must be considered at quadrupole order, which for most cross sections is small in comparison to dipole radiation from bending.

where $\boldsymbol{D}$ (previously $\boldsymbol{D}^v$) characterizes the dipole radiation fields,

$$\boldsymbol{D} = \oint_S (\boldsymbol{y} - \boldsymbol{\varphi}^*(\boldsymbol{y}))\, \boldsymbol{n}(\boldsymbol{y})^\top dS_{\boldsymbol{y}}. \tag{9}$$

*4.1.3 Reduction to 2D.* If we assume elements are aligned end-to-end along the $z$-axis such that "end cap" contributions cancel with their adjacent neighboring elements, we can ignore the parts of $S$ that are not in contact with the surrounding medium when computing $\boldsymbol{D}$. Furthermore, since we have assumed that the cross-sectional shape varies slowly along the rod, we note that $\boldsymbol{n}(\boldsymbol{y})^\top \hat{\boldsymbol{z}} \approx 0$ on the rest of $S$, where $\hat{\boldsymbol{z}}$ is the unit vector along the $z$-axis. Consequently $\boldsymbol{D}\hat{\boldsymbol{z}} = \boldsymbol{0}$, which is to say that no dipole source is generated by $z$-axis motion for this model. Therefore we can ignore element oscillations in the $z$ direction and consider only the $2 \times 2$ top-left submatrix of $\boldsymbol{D}$. Piecewise constant $z$-integration can be done for a rod element of length $\Delta z$, giving

$$D_{ij} \approx \Delta z \oint_\Gamma \left(y_i - \varphi_i^*(\boldsymbol{y})\right) n_j(\boldsymbol{y}) d\Gamma_{\boldsymbol{y}} \tag{10}$$

for $i, j \in \{1, 2\}$, where $\Gamma$ is the 2D curve defining the element's cross-sectional boundary. Therefore we only need to determine the boundary values of two $\varphi_i^*$ velocity potentials corresponding to 2D exterior Laplace problems in the $xy$-plane. We now explain in more detail how to compute the $\boldsymbol{D}$ surface integral in (10).

## 4.2 Implementation Details

*4.2.1 Boundary Element Method estimation of $\boldsymbol{D}$.* To calculate $\boldsymbol{D}$ for a single element, we must first calculate the $x$ and $y$ components of $\boldsymbol{\varphi}^*$. This requires solving a pair of 2D exterior Laplace problems with Neumann boundary conditions on $\Gamma$ (i.e., a 2D version of (3), (4), and (5)). We discretize $\Gamma$ into a set of line segments $\tilde{\Gamma}$, and then use the 2D Boundary Element Method (BEM) to estimate the $x$ and $y$ components of $\boldsymbol{\varphi}^*$ on $\Gamma$; in our implementation we use the direct BEM collocation method (see [Pechstein 2009] for formulae). Note that this involves solving a dense, nonsymmetric linear system of equations of size $|\tilde{\Gamma}|$. Matrix entries are computed by integrating over each line segment in $\tilde{\Gamma}$ with respect to the midpoint of each other member of $\tilde{\Gamma}$. The integrals have analytical solutions, but their expressions become singular as the line segment and midpoint approach collinearity. To avoid this instability, we simply approximate each integral using Simpson's rule with sufficiently many samples. In practice, we define a maximum length for each of our line segments in $\tilde{\Gamma}$; in our framework, this maximum length is $50\mu$m, which provides a good tradeoff between speed and accuracy. We then calculate the integrand $(\boldsymbol{y} - \boldsymbol{\varphi}^*(\boldsymbol{y}))\, \boldsymbol{n}(\boldsymbol{y})^\top$ piecewise over $\tilde{\Gamma}$ and sum the results, multiplying by $\Delta z$, to determine the $2 \times 2$ top-left submatrix of $\boldsymbol{D}$.

Rods with circular cross sections are common, and in such cases $\boldsymbol{D}$ is known analytically, eliminating the need for BEM calculations:

$$\boldsymbol{D} = 2\pi a^2 \Delta z \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{11}$$

where $a$ is the rod's radius.

*4.2.2 Preprocessing rod elements.* We replicate rod elements for each of our control points along the length of a rod. We define a cross-sectional shape for each control point $i$, from which we precompute $\boldsymbol{D}_i$ as described above. (For extruded rods, i.e., rods with cross sections that are constant along their length, we compute $\boldsymbol{D}$ once and reuse it for every control point.) Because these calculations require an integral along the $z$ dimension of the element (our $\Delta z$ length), we assume that the cross section is nearly constant along the element, and multiply the result of the integral over the 2D cross section by the Voronoi length of control point $i$ to compute $\boldsymbol{D}_i$. Bergou et al. [2010] define the Voronoi length as $\bar{l}_i = (\|\bar{\boldsymbol{e}}^{i-1}\| + \|\bar{\boldsymbol{e}}^i\|)/2$ where overlines denote quantities computed on the undeformed configuration of the rod; we extend this definition to the first and last control points by letting $\bar{l}_0 = \|\bar{\boldsymbol{e}}^0\|$ and $\bar{l}_{n+2} = \|\bar{\boldsymbol{e}}^{n+1}\|$.

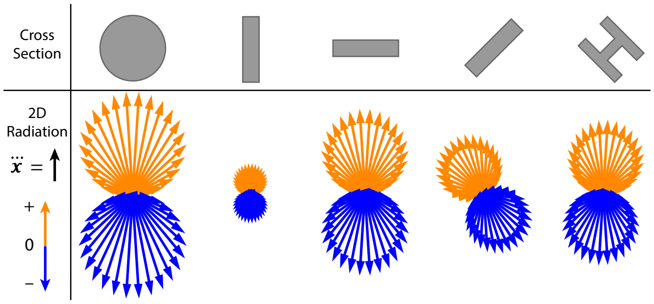

Fig. 4. **2D dipole radiation fields for different cross sections:** Acoustic pressure values are calculated in several directions in the $xy$-plane for a few examples of 2D cross sections in response to jerk motion in the vertical direction. Such effects are encoded in the $\boldsymbol{D}$ dipole matrix. The length of each arrow represents the magnitude of the acoustic pressure; orange represents positive values and blue negative.

*4.2.3 Transforming and accumulating element contributions.* In practice, the rod element is not always aligned with the $z$-axis, and is not located at the origin, both during preprocessing and during runtime animation. Therefore, we must transform the equations above to be used within our framework. Let $\boldsymbol{r}_i(t)$ represent the vector from the $i^{th}$ control point, $\boldsymbol{x}_i(t)$, to the position of a listener at a given instant in time $t$. Let $\boldsymbol{R}_i(t)$ approximate the rod's material frame at control point $i$, as defined in §3.1. Taking this transformation into account, the acoustic pressure due to vibrations of element $i$ at time $t$ contributes sound at the listener's location slightly later:

$$p_i\left(t + \frac{r_i(t)}{c}\right) = \frac{\rho}{4\pi c\, r_i(t)}\, \hat{\boldsymbol{r}}_i(t)^\top \boldsymbol{R}_i(t) \boldsymbol{D}_i \boldsymbol{R}_i(t)^\top \dddot{\boldsymbol{x}}_i(t). \tag{12}$$

Using the value of $\ddot{\boldsymbol{x}}_i$ that is computed at each timestep within our integrator, we approximate $\dddot{\boldsymbol{x}}_i$ at each timestep using a simple finite difference calculation: $\dddot{\boldsymbol{x}}(t_k + \Delta t/2) \approx (\ddot{\boldsymbol{x}}(t_{k+1}) - \ddot{\boldsymbol{x}}(t_k))/\Delta t$, where $\Delta t = t_{k+1} - t_k$ is the length of the timestep. (12) is computed for each rod element at each timestep and summed into a single array of audio samples. Array entries are associated with discrete timestamps, but (12) requires us to record pressure samples at arbitrary times. To accomodate this, acoustic pressure values computed in (12) are linearly interpolated between their two nearest entries in time within the sample array.

Note that we require $\Delta t \leq 1/40000$ (sec) to accurately generate audio for the full audible range (up to 20kHz), though larger timesteps may be used at the price of degrading the audio quality. Unless otherwise specified, we choose a sample rate and timestep size of

44.1kHz for all of our simulations. For simulations with a different timestep, we record audio samples at the timestep rate, and resample the audio to 44.1kHz after the simulation has completed. Resampling is done using FFmpeg [2016], which filters the audio using a Kaiser window with $\beta = 0.9$.

### 4.3 Caveats

Several assumptions were made when deriving the dipole-based sound model (12), which introduce particular limitations:

- **Acoustic compactness** of the rod cross section was assumed by (2) and implies that the rod diameter $d$ is much smaller than the wavelengths of interest, i.e., $d \ll \lambda$. Since $\lambda = c/f$ for frequency $f$ and $c = 340$m/s, for a dominant frequency of 10kHz, this implies that $d \ll 3.4$cm, which is true for our examples. Fatter rods may also require more sophisticated vibrational analyses.
- **Far-field listening positions** were assumed by the dipole radiation model (2), and require that the distance to the listener $r$ satisfy $r \gg \lambda$, i.e., be much larger than wavelengths of interest. For typical low-frequency wavelengths of 100Hz–1kHz, this implies minimum distances of 3.4m–0.34m, which is reasonable for computer animation.
- **Cross-sectional symmetry** was assumed for the simplified pure-translational model in (8) and is true of most of our examples. More exotic cross sections can be accommodated by including additional angular terms, i.e., $\boldsymbol{D}^{\omega}$.
- **Smoothness (slowly varying cross section)** was assumed to ignore radiation from $z$-aligned vibrations. While these contributions are usually negligible, bumpy or more exotic 3D rod shapes may require more detailed analysis to recover these contributions.

## 5 CONTACT

In our framework, we use contact events within the simulation to drive the audible vibrations of the rod. We've found that impulse methods are best suited for our application. Many modern sound synthesis frameworks adopt this solution as well; see, for example, [Zheng and James 2011] and [Li et al. 2015]. A number of contact solvers specific to thin structures exist, such as [Spillmann and Teschner 2008] and [Kaufman et al. 2014]. In our case, we choose the So-Bogus solver [Daviet et al. 2011], as it is designed to be robust and scalable. Because our algorithm requires resolving collision at the audio sample rate (if not faster), quickly resolving multi-contact problems is critical to the efficiency of our framework. The So-Bogus algorithm solves local problems in parallel using Newton iteration, and solves the global problem using Gauss-Seidel iteration. In the case that the global convergence threshold is not reached, So-Bogus generally still produces a reasonable impulse response.

In many sound synthesis frameworks, contact stiffness is typically modeled using springs, as with Hertz contact theory. However, in the case of thin rods and hard surfaces, these springs become exceptionally stiff; for some of our examples, it is necessary to reduce the timestep size by an order of magnitude to maintain stability. Therefore, we use the So-Bogus solver, a discrete impulse-based method, even though it does not provide parameters for tuning contact stiffness.

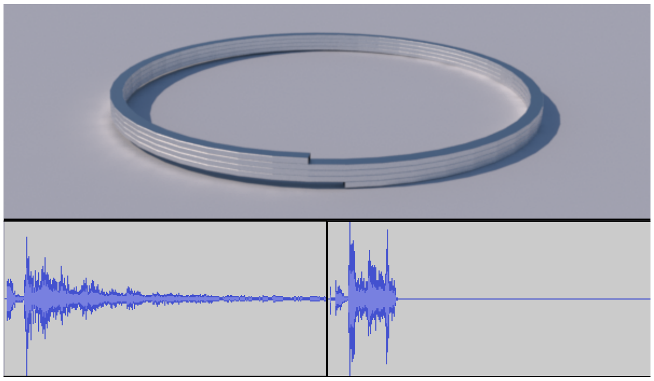

Fig. 5. **Nonrigid damping:** (Top) A 4-ringed Slinky approaching resting contact with the ground plane. (Left) Without damping, we observe buzzing artifacts. (Right) With post-contact nonrigid damping applied, the artifacts are removed, and resting contact is achieved.

Iterative methods such as So-Bogus are prone to buzzing artifacts, as they may not find the exact solution and are not temporally coherent. Such artifacts are perceptible when multiple rods are in resting contact with another. We remove excess energy from bodies coming to rest by applying an impulse counteracting nonrigid motion after contact has been resolved [Müller et al. 2007]. This damping impulse is weighted higher when the norm of the velocity of the rod is low and vanishes when the norm is higher than a user-specified threshold. The impulse is applied only to the spatial degrees of freedom, except in the special case of straight rods, where it is necessary to account for the twist degrees of freedom to avoid a singular matrix solve when calculating the rigid angular velocity.

During collision detection, we treat each edge of the rod as a rigid body. We place each edge's collision detection primitive into a bounding volume hierarchy that is updated each timestep to cull the majority of collision pair candidates. Primitive collision detection is performed using the ODE library [Smith and Developers 2014]. Like [Zheng and James 2011], we require position-level contact response—that is, we generate contacts whenever we detect interpenetrating geometries, ignoring the relative velocity of the two bodies. This is necessary to achieve resting contact without audible jittering.

Another problem we encounter related to contact resolution is integrator instability. Although Newmark-$\beta$ has excellent energy conservation properties, it is known to have stability problems in the presence of hard constraints and contacts. In our examples, this resulted in spurious signals at the Nyquist frequency around contact events, and very occasionally some numerical instability from which Newmark-$\beta$ could not recover. A number of modifications to the method have been proposed that reduce this instability, usually by damping high-frequency oscillations (e.g., [Bathe 2007]). However, we are not aware of an integration method that is approximately as efficient as Newmark-$\beta$ and doesn't introduce audible numerical damping. Instead, we apply a sixth-order Butterworth low-pass filter with a cutoff of 19.8kHz to the resulting audio output to manually remove oscillations at the Nyquist frequency. In the case of unrecoverable instabilities, we simply reduce the timestep by half (which results in approximately the same amount of overall computation as Bathe's integrator using the original timestep). All of our examples are stable when simulated at 88.2kHz.

Finally, we summarize the algorithm in Figure 6.

```
$\boldsymbol{q}(t_{k+1}), \dot{\boldsymbol{q}}(t_{k+1}), \ddot{\boldsymbol{q}}(t_{k+1}) \leftarrow \mathsf{newmark}(\boldsymbol{q}(t_k), \dot{\boldsymbol{q}}(t_k), \ddot{\boldsymbol{q}}(t_k))$
$C \leftarrow \mathsf{detectContacts}(\boldsymbol{q}(t_{k+1}))$
$\boldsymbol{j} \leftarrow \mathsf{calculateContactImpulse}(C, \boldsymbol{q}(t_{k+1}), \dot{\boldsymbol{q}}(t_{k+1}))$
$\boldsymbol{q}(t_{k+1}), \dot{\boldsymbol{q}}(t_{k+1}), \ddot{\boldsymbol{q}}(t_{k+1}) \leftarrow \mathsf{applyImpulse}(\boldsymbol{j})$
if $\|\dot{\boldsymbol{q}}(t_{k+1})\| < \mathsf{dampThreshold}$ then
    $\boldsymbol{j} \leftarrow \mathsf{calculateNonrigidDampingImpulse}(\dot{\boldsymbol{q}}(t_{k+1}))$
    $\boldsymbol{q}(t_{k+1}), \dot{\boldsymbol{q}}(t_{k+1}), \ddot{\boldsymbol{q}}(t_{k+1}) \leftarrow \mathsf{applyImpulse}(\boldsymbol{j})$
$\dddot{\boldsymbol{q}}(t_k + \frac{\Delta t}{2}) \leftarrow (\ddot{\boldsymbol{q}}(t_{k+1}) - \ddot{\boldsymbol{q}}(t_k))/\Delta t$
for all Listener $\ell$ do
    $\mathsf{recordSample}(\ell, \dddot{\boldsymbol{q}}(t_k + \frac{\Delta t}{2}))$
```

Fig. 6. **Timestep loop:** This pseudocode describes one iteration of our simulation, taking in the state at time $t_k$ and advancing to time $t_{k+1}$.

## 6 RESULTS

Our simulator is written in C++. It is multithreaded, and the dynamics of each rod is computed independently. The contact detection step is single-threaded, though we take advantage of OpenMP to solve local contact problems in parallel within the So-Bogus solver. We found that repeated computation of the stiffness matrix becomes the bottleneck in these examples, though contact resolution becomes comparatively expensive in particularly dense cases such as *Slinky Walk*. Precomputation of the $\boldsymbol{D}$ matrix involves at worst taking an *LU* decomposition over a dense matrix with about $1000^2$ entries, and completes in less than a second for all of our examples. See Figure 7 for performance results.

In examples that include a ground plane, we model sound wave reflection against this plane by recording the acoustic pressure for each pair of rods and listeners twice; once normally, and once with the listener's position replaced with its reflection across the ground plane. This second signal may be attenuated to simulate absorption; in all of our examples, we multiply the reflected signal by 0.96.

### 6.1 Examples

We now enumerate a number of examples that demonstrate the range of sounds we can simulate using our method. Please see the accompanying video to view and hear the full results.

*Rectangular Rod:* An aluminum rod 1 foot in length with a rectangular cross section (1 inch by 0.5 inches) is struck on two of its sides, producing two distinct tones resulting from its anisotropic cross section. In our simulation, we model this impulse using a half-sine force pulse applied to a single control point over a short period. We are able to match the lower tone well, but our simulation overestimates the pitch of the higher tone when struck on its thinner side; the fundamental pitch is raised by about 2% compared to the recording. We attribute this observation to the lack of cross section shearing in our dynamics model, which increases the effective bending stiffness, especially in short and thick rods such as this one. Nevertheless, the metallic character of the recording is well captured by our simulation.

We compare the frequencies of the recording and the simulated audio in Figure 8. We also compare these against the solutions to the ideal bar equation, which gives the approximate frequencies of bending modes of a straight rod with uniform cross section. Though we do not match the recorded data perfectly, we approximate it very well for the most perceptually significant frequencies (i.e., between 200Hz and 8000Hz), and it is clear that we predict the modes more accurately than the linear solution given by the ideal bar equation.

*Circular Rod:* An aluminum rod is dropped on a concrete floor. The 2 foot long rod has a circular cross section with a radius of 0.25 inches. The contact stiffness issue mentioned in §5 results in louder high-frequency content than the corresponding recording; however, the frequencies of the bending modes match well. In addition, the rod's chattering behavior is plausible as it comes to rest.

*Ruler:* An oak ruler, 15 inches in length, is dropped on a concrete floor. Its cross section is nearly rectangular, and is 3.3cm by 0.7cm in its principal dimensions. This example demonstrates our ability to simulate different materials and cross sections. Our simulation generates reasonable sound, though there are audible differences between our result and the recording. We attribute these to the contact stiffness limitations as with the previous example, as well as the relatively high aspect ratio of the object that pushes the limits of our Kirchhoff model.

*Strapping:* A strip of stainless steel strapping is dropped on a concrete floor. It measures 3.5 feet in length, with a rectangular cross section of 1 by 1/32 inches. While the sound generated by our method is somewhat plausible, we fail to match a recording of the same object. Due to the high cross section aspect ratio, the body acts more like an elastic sheet than a rod. Although we would need another physics model to accurately reproduce the sound, our radiation model is still effective for this geometry.

*Clock Gong:* A spiral clock gong that is fixed at one end is struck, producing a characteristic sound with a dense frequency spectrum. The gong is steel, has a circular cross section with a radius of 2mm, and is approximately 14cm across its longest dimension. We model the strike as a sum of half-sine pulses. This example shows that our method extends trivially to rods with arbitrary resting shape and with dense modal spectra.

*Rubber Band:* A rubber band is plucked, producing a deep tone. Highly deformable materials such as rubber work well within our framework. At rest, the band has a length of 18.6cm, and it is stretched between two posts 28.3cm apart. Its cross section measures 1.6mm by 0.8mm.

*Guy Wire ("blaster" sound):* A long wire supporting a tower is struck with a quick impulse, displaying the dispersion effects captured by our framework. This approximates the source of the "blaster" sound well-known in the sound design community. Because of its length, this example would be very costly to simulate with a volumetric mesh representation, but our method is able to simulate it efficiently. The steel rope is approximated as a single steel rod with a circular cross section of radius 0.25in. Its resting length is 20m, but

| **Example** | **# CPs** | $E$ (GPa) | $\nu$ | **Computation time/sec** (h:mm:ss) | **Simulation time** | **Collision time** | **Sound time** |
|---|---|---|---|---|---|---|---|
| *Rectangular Rod* | 30 | 68.9 | 0.33 | 0:00:10 | 97.1% | – | 1.8% |
| *Circular Rod* | 50 | 70.7 | 0.33 | 0:00:42 | 67.5% | 30.4% | 1.7% |
| *Ruler* | 36 | 6.3 | 0.3 | 0:00:47 | 47.3% | 50.7% | 1.6% |
| *Strapping* | 100 | 205 | 0.29 | 0:01:31 | 75.0% | 22.9% | 1.8% |
| *Clock Gong* | 400 | 225 | 0.29 | 0:02:24 | 99.2% | – | 0.7% |
| *Rubber Band* | 50 | 0.0003 | 0.49 | 0:00:12 | 96.9% | – | 2.2% |
| *Guy Wire* | 1500 | 205 | 0.29 | 0:16:40 | 99.0% | – | 0.9% |
| *Slinky Dispersion* | 3888 | 205 | 0.29 | 0:51:38 | 99.2% | – | 0.8% |
| *Chain Link Fence* | 15000 | 68.9 | 0.33 | 1:28:16 | 56.8% | 33.3% | 9.9% |
| *Slinky Walk* | 3888 | 205 | 0.29 | 4:35:27 | 33.3% | 65.5% | 1.1% |

Fig. 7. **Performance:** For each example, we state the number of control points present in the simulation, the material parameters used (Young's modulus $E$ and Poisson's ratio $\nu$), the total wall clock time to generate one second of sound (averaged over the entire simulation), as well as the portion of time spent on rod dynamics simulation, collision detection and resolution, and sound computation. All timing data was measured on an 8-core Intel Xeon E5-2637 v3 CPU.

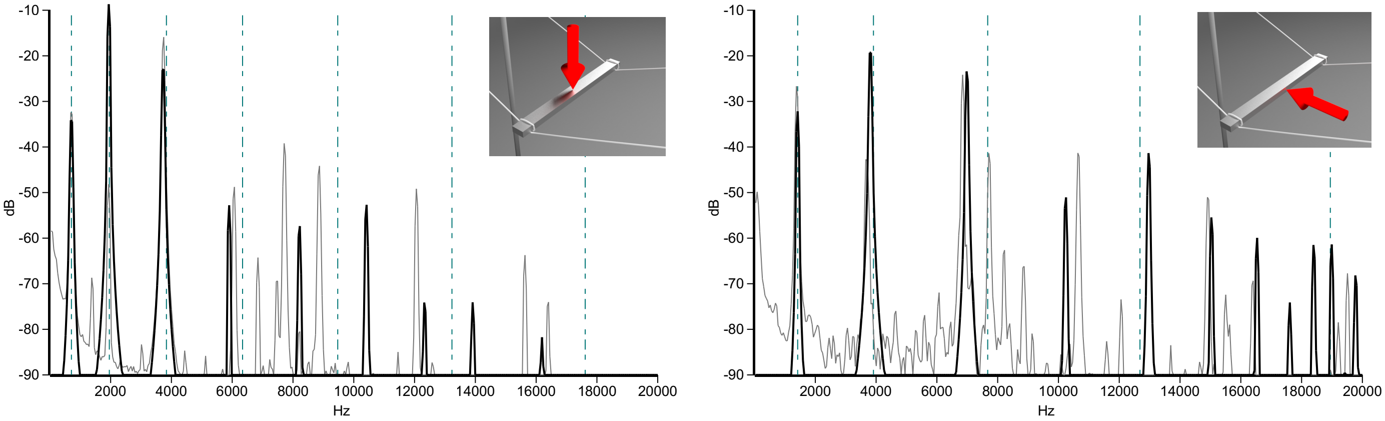

Fig. 8. **Comparison of frequency spectra** for the *Rectangular Rod* example. Comparisons are shown between the simulation output (bold line), the recorded data (thin line), and the solutions to the ideal bar equation (dashed lines) for impulses along the long edge (Left) and short edge (Right) of the rod. Note that the ideal bar equation only gives frequencies corresponding to the bending modes, and not their relative intensities. Audio data is obtained from the tail of a short impulse to isolate the bending frequencies.

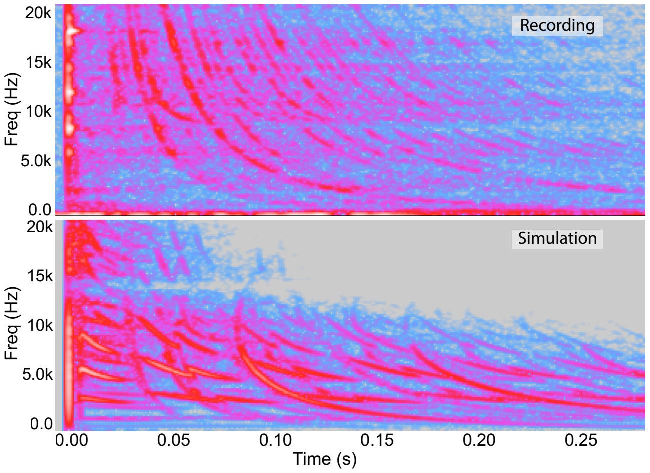

Fig. 9. **Dispersion:** Spectrograms from recorded (Top) and simulated (Bottom) versions of *Slinky Dispersion*. The pitch glide effects due to dispersion are clearly visible in both.

it is stretched to 20.1m between clamped boundary conditions. To accurately simulate bending wave propagation speeds at all audible frequencies, the simulation rate for this example is 88.2kHz.

*Slinky Dispersion:* A Slinky is suspended between a pair of damped springs and is struck with a small metallic bearing. A microphone located close to one end of the Slinky captures the dispersion effects as bending waves travel at different speeds along the Slinky's length; see Figure 9. The steel Slinky consists of 81 loops of a helix 3.25cm in radius. Its cross section is rectangular, measuring 0.95mm by 0.38mm. Again, time integration at 88.2kHz is required to correctly resolve bending wave propagation speeds.

*Chain Link Fence:* A rigid baseball strikes a chain link fence, causing it to rattle. This demonstrates that we can incorporate traditional rigid body dynamics into our framework, and account for large numbers of contacts. The portion of the fence we simulate consists of 35 interlocking aluminum rods, each having a circular cross section of radius 1.45mm. These rods are constrained at the top and bottom. Additionally, we simulate the sound of the posts to the left and right of these links, which are responsible for much of the audible sound; they are aluminum, 1.15m in length, and have a circular hollow cross section that is 1.6cm in radius and 1mm thick. This example is integrated at 88.2kHz to maintain stability, but because the dynamics of each rod are computed independently and many of the contacts are not directly coupled, this example parallelizes very well.

*Slinky Walk:* A Slinky walks down a set of stairs. The parameters of this Slinky are the same as in the *Slinky Dispersion* example. The simulation rate for this example is 88.2kHz for stability reasons. This example demonstrates the resilience of our method; we can handle tens of thousands of highly coupled frictional contact events per timestep, simulate highly deformable materials, and obtain stable resting contact, all while generating compelling audio.

## 6.2 Dipole Model Tests

A number of time-domain sound synthesis frameworks obtain the sound of a vibrating body by sampling the displacement or velocity of a single point (e.g., [Bilbao 2009] and [Hsu and Sosnick-Pérez 2013]). While this is straightforward to implement for small-displacement or static configurations (like those seen in musical acoustics), it is unclear how to extend this to dynamic 3D environments; any rigid motion of the object needs to be filtered from the audio signal, and it is unclear how to reduce displacement in 3D down to a 1D audio signal. Furthermore, selection of the point can introduce bias in the resulting signal, e.g., a point that lies at a node of a particular vibration mode can lead to the mode's vibration frequency being underrepresented in the final signal. We compare our radiation model to the velocity signal of a single control point in the static *Rectangular Rod* example above, and present the results in the accompanying video to demonstrate this bias.

The preceding experiment suggests that a full radiation model, with contributions from all elements of the body, is important to the resulting sound quality. However, naively summing acoustic pressure contributions from all elements can result in mode cancellation if the delay due to the speed of sound is not taken into account. The retarded time from (2) prevents this. We compare the *Rod Drop* example with and without the retarded time in the accompanying video; without the delay, the audio signal is prone to clearly audible interference.

Finally, it may seem appealing to eliminate precomputation by approximating the cross section of the rod as a circle. However, this simplification results in perceptually significant changes to the resulting sound if the cross section is far from circular. We demonstrate this effect by comparing our directionally dependent dipole model with a uniform dipole model for the *Ruler* example. The uniform model artificially exaggerates the high-frequency bending modes that do not radiate well in reality, resulting in increased high-frequency content.

## 7 DISCUSSION AND CONCLUSION

We have presented a method for simultaneously generating sound and physically based animation for thin structures. Our model easily fits into existing 3D simulation environments and is able to generate plausible sound even for highly deformable objects.

The radiation model we introduced is valid for any body with a compact cross section, and is independent of dynamics. This means that other dynamics models may be used in place of our rod model. For instance, using Cosserat dynamics may give results that are more physically accurate for rods with thicker cross sections, or using modal synthesis with our radiation model may provide large gains in efficiency for nearly rigid examples.

For all of our examples above, our method produces plausible sound. However, we are not able to match actual recordings in examples with cross sections with high aspect ratios (*Ruler* and *Strapping*). We believe this is due to limitations in our rod model; using other physical models, such as elastic sheets, may provide sound that more accurately matches the recordings.

### 7.1 Future Work

We have shown that our method is effective at radiating bending waves; these are generally the vibrations that produce the most sound. However, we do not account for sound radiation from other kinds of modes, such as those arising from longitudinal vibrations. We assume that these are generally insignificant, though some important cases fail to produce correct sound because of their absence. For example, a straight rod dropped directly on one of its ends produces no sound in our framework. Torsional waves are also ignored, though these could plausibly contribute noticeably to the overall sound of rods with highly anisotropic cross sections.

As mentioned in §5, one drawback of our method is that there is no way to tune the contact stiffness for collisions between rods and rigid bodies; as a result, many of our examples involving contact exhibit artificially loud high-frequency vibrations excited by very short contact events. We would like to explore ways in which time-domain sound simulations such as ours could take advantage of impulse-based solvers without incurring these artifacts.

## ACKNOWLEDGMENTS

We would like to thank Dae Hyun Kim for early numerical experiments, Rafael Marinheiro for software contributions, and Risa Feng for useful discussion. This work was supported by the National Science Foundation (Projects 0905506, 1011919, 1513967, and 1644523).

## A PROOF OF VANISHING $D^\omega$

Proof. By definition, $\boldsymbol{D}^\omega = \oint_S (\boldsymbol{y} - \boldsymbol{\varphi}^*(\boldsymbol{y}))\quad (\boldsymbol{y} \times \boldsymbol{n}(\boldsymbol{y}))^\top\, dS_{\boldsymbol{y}}$. On a fully symmetric domain, $\boldsymbol{n}(\boldsymbol{y})$ is an odd function, i.e., for all $\boldsymbol{y}$ in $S$, $-\boldsymbol{y}$ is also in $S$ and $\boldsymbol{n}(-\boldsymbol{y}) = -\boldsymbol{n}(\boldsymbol{y})$. $\boldsymbol{\varphi}^*(\boldsymbol{y})$ is a 3D solution to an external Laplace problem with Neumann boundary conditions (see (3), (4), and (5)). The boundary conditions themselves are determined through the odd function $\boldsymbol{n}(\boldsymbol{y})$, and the Laplace equations have no symmetry bias; therefore $\boldsymbol{\varphi}^*$ itself is odd in $\boldsymbol{y}$.

The integrand is a product of three functions that are odd in $\boldsymbol{y}$, meaning the integrand itself is also odd in $\boldsymbol{y}$. Thus, $\boldsymbol{D}^\omega$ is determined by an integral of an odd function over a symmetric domain and therefore vanishes. □

## REFERENCES

A Akay, MT Bengisu, and M Latcha. 1983. Transient Acoustic Radiation from Impacted Beam-like Structures. *Journal of Sound and Vibration* 91, 1 (1983), 135–145.

Steven S An, Doug L James, and Steve Marschner. 2012. Motion-driven Concatenative Synthesis of Cloth Sounds. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 102.

Stuart S Antman. 1973. The Theory of Rods. In *Linear Theories of Elasticity and Thermoelasticity*. Springer, 641–703.

Balázs Bank and László Sujbert. 2005. Generation of Longitudinal Vibrations in Piano Strings: From Physics to Sound Synthesis. *The Journal of the Acoustical Society of America* 117, 4 (2005), 2268–2278.

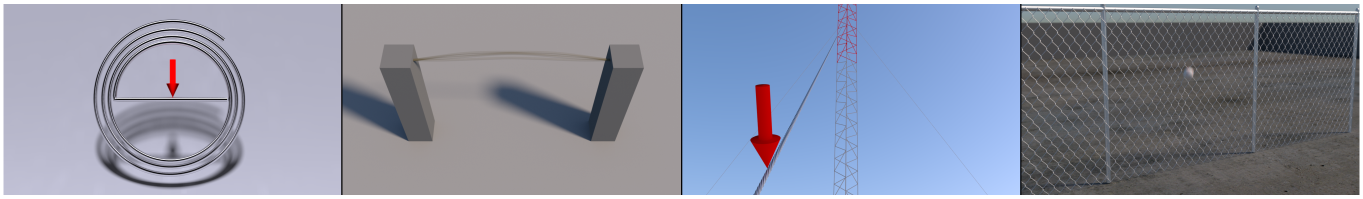

Fig. 10. **Examples:** From left to right: *Clock Gong*, *Rubber Band*, *Guy Wire*, and *Chain Link Fence*.

Klaus-Jürgen Bathe. 2007. Conserving Energy and Momentum in Nonlinear Dynamics: A Simple Implicit Time Integration Scheme. *Computers & Structures* 85, 7 (2007), 437–445.

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete viscous threads. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 116.

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 63.

Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. 2006. Super-helices for Predicting the Dynamics of Natural Hair. In *ACM Transactions on Graphics (TOG)*, Vol. 25. ACM, 1180–1187.

Stefan Bilbao. 2004. *Wave and Scattering Methods for Numerical Simulation.* John Wiley & Sons.

Stefan Bilbao. 2007. Robust Physical Modeling Sound Synthesis for Nonlinear Systems. *IEEE Signal Processing Magazine* 24, 2 (2007), 32–41.

Stefan Bilbao. 2009. *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics.* John Wiley & Sons.

Stefan Bilbao. 2013. Numerical Simulation of Spring Reverberation. In *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)*.

Stefan Bilbao and Alberto Torin. 2015. Numerical Modeling and Sound Synthesis for Articulated String/Fretboard Interactions. *Journal of the Audio Engineering Society* 63, 5 (2015), 336–347.

Stefan Bilbao, Alberto Torin, and Vasileios Chatziioannou. 2015. Numerical Modeling of Collisions in Musical Instruments. *Acta Acustica united with Acustica* 101, 1 (2015), 155–173.

Jeffrey N Chadwick, Steven S An, and Doug L James. 2009. Harmonic Shells: A Practical Nonlinear Sound Model for Near-rigid Thin Shells. In *ACM Transactions on Graphics (TOG)*, Vol. 28. ACM, 119.

Jeffrey N Chadwick, Changxi Zheng, and Doug L James. 2012. Precomputed Acceleration Noise for Improved Rigid-body Sound. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 103.

Vasileios Chatziioannou and Maarten van Walstijn. 2015. Energy Conserving Schemes for the Simulation of Musical Instrument Contact Dynamics. *Journal of Sound and Vibration* 339 (2015), 262–279.

Gabriel Cirio, Dingzeyu Li, Eitan Grinspun, Miguel A Otaduy, and Changxi Zheng. 2016. Crumpling Sound Synthesis. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 181.

Bernard D Coleman and Ellis H Dill. 1992. Flexure Waves in Elastic Rods. *The Journal of the Acoustical Society of America* 91, 5 (1992), 2663–2673.

Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. 2011. A Hybrid Iterative Solver for Robustly Capturing Coulomb Friction in Hair Dynamics. In *ACM Transactions on Graphics (TOG)*, Vol. 30. ACM, 139.

FFmpeg Developers. 2016. FFmpeg. (2016). https://ffmpeg.org/ Version 2.8.6.

Yoshinori Dobashi, Tsuyoshi Yamamoto, and Tomoyuki Nishita. 2003. Real-time Rendering of Aerodynamic Sound Using Sound Textures Based on Computational Fluid Dynamics. *ACM Transactions on Graphics (TOG)* 22, 3 (2003), 732–740.

Michele Ducceschi and Stefan Bilbao. 2016. Linear Stiff String Vibrations in Musical Acoustics: Assessment and Comparison of Models. *The Journal of the Acoustical Society of America* 140, 4 (2016), 2445–2454.

Georg Essl, Stefania Serafin, Perry R Cook, and Julius O Smith. 2004. Theory of Banded Waveguides. *Computer Music Journal* 28, 1 (2004), 37–50.

Neville H Fletcher and Thomas Rossing. 1998. *The Rhysics of Musical Instruments*. Springer Science & Business Media.

Thomas Hélie and David Roze. 2008. Sound Synthesis of a Nonlinear String using Volterra Series. *Journal of Sound and Vibration* 314, 1 (2008), 275–306.

Michael S Howe. 2003. *Theory of vortex sound.* Vol. 33. Cambridge University Press.

Bill Hsu and Marc Sosnick-Pérez. 2013. Real-time GPU audio. *Commun. ACM* 56, 6 (2013), 54–62.

Doug L James, Jernej Barbič, and Dinesh K Pai. 2006. Precomputed Acoustic Transfer: Output-sensitive, Accurate Sound Generation for Geometrically Complex Vibration Sources. In *ACM Transactions on Graphics (TOG)*, Vol. 25. ACM, 987–995.

Danny M Kaufman, Rasmus Tamstorf, Breannan Smith, Jean-Marie Aubry, and Eitan Grinspun. 2014. Adaptive Nonlinearity for Collisions in Complex Rod Assemblies. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 123.

Tassilo Kugelstadt and Elmar Schömer. 2016. Position and Orientation Based Cosserat Rods. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* Eurographics Association, 169–178.

Holger Lang and Martin Arnold. 2012. Numerical Aspects in the Dynamic Simulation of Geometrically Exact Rods. *Applied Numerical Mathematics* 62, 10 (2012), 1411–1427.

Dingzeyu Li, Yun Fei, and Changxi Zheng. 2015. Interactive Acoustic Transfer Approximation for Modal Sound. *ACM Transactions on Graphics (TOG)* 35, 1 (2015), 2.

Tyler McMillen and Alain Goriely. 2003. Whip Waves. *Physica D: Nonlinear Phenomena* 184, 1 (2003), 192–225.

Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position Based Dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118.

Nathan M Newmark. 1959. A Method of Computation for Structural Dynamics. *Journal of the Engineering Mechanics Division* 85, 3 (1959), 67–94.

James F. O'Brien, Perry R. Cook, and Georg Essl. 2001. Synthesizing Sounds from Physically Based Motion. In *Proceedings of ACM SIGGRAPH 2001.* ACM Press, 529–536. DOI: https://doi.org/10.1145/383259.383321

James F O'Brien, Chen Shen, and Christine M Gatchalian. 2002. Synthesizing Sounds from Rigid-body Simulations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* ACM, 175–181.

Dinesh K Pai. 2002. Strands: Interactive Simulation of Thin Solids Using Cosserat Models. In *Computer Graphics Forum*, Vol. 21. Wiley Online Library, 347–352.

Clemens Pechstein. 2009. *Lecture on Boundary Element Methods.* Institute of Computational Mathematics, Johannes Kepler University Linz.

Camille Schreck, Damien Rohmer, Doug James, Stefanie Hahmann, and Marie-Paule Cani. 2016. Real-time Sound Synthesis for Paper Material Based on Geometric Analysis. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2016).*

Ahmed A Shabana. 2012. *Theory of Vibration: Volume II: Discrete and Continuous Systems.* Springer Science & Business Media.

Julius O Smith. 2004. Virtual Acoustic Musical Instruments: Review and Update. *Journal of New Music Research* 33, 3 (2004), 283–304.

Russell Smith and ODE Developers. 2014. Open Dynamics Engine. (2014). http://www.ode.org/ Version 0.13.

Jonas Spillmann and Matthias Teschner. 2007. CORDE: Cosserat Rod Elements for the Dynamic Simulation of One-Dimensional Elastic Objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* Eurographics Association, 63–72.

Jonas Spillmann and Matthias Teschner. 2008. An Adaptive Contact Model for the Robust Simulation of Knots. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 497–506.

Tapio Takala and James Hahn. 1992. Sound rendering. In *Computer Graphics*, Vol. 26. ACM, 211–220.

Italo Testa, Gianpaolo Evangelista, and Sergio Cavaliere. 2004. Physically Inspired Models for the Synthesis of Stiff Strings with Dispersive Waveguides. *EURASIP Journal on Advances in Signal Processing* 2004, 7 (2004), 1–14.

Nobuyuki Umetani, Ryan Schmidt, and Jos Stam. 2014. Position-based Elastic Rods. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation.* Eurographics Association, 21–30.

Vesa Välimäki, Jyri Pakarinen, Cumhur Erkut, and Matti Karjalainen. 2005. Discrete-time Modelling of Musical Instruments. *Reports on Progress in Physics* 69, 1 (2005), 1.

Kees van den Doel, Paul G Kry, and Dinesh K Pai. 2001. FoleyAutomatic: Physically-based Sound Effects for Interactive Simulation and Animation. In *Proceedings of SIGGRAPH '01.* ACM, 537–544.

Changxi Zheng and Doug L James. 2011. Toward High-quality Modal Contact Sound. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 38.