



HAL
open science

Multi-Scale Simulation of Nonlinear Thin-Shell Sound with Wave Turbulence

Gabriel Cirio, Ante Qu, George Drettakis, Eitan Grinspun, Changxi Zheng

► **To cite this version:**

Gabriel Cirio, Ante Qu, George Drettakis, Eitan Grinspun, Changxi Zheng. Multi-Scale Simulation of Nonlinear Thin-Shell Sound with Wave Turbulence. *ACM Transactions on Graphics*, 2018, 37 (4), pp.14. 10.1145/3197517.3201361 . hal-01797920

HAL Id: hal-01797920

<https://inria.hal.science/hal-01797920>

Submitted on 22 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Scale Simulation of Nonlinear Thin-Shell Sound with Wave Turbulence

GABRIEL CIRIO, Inria, Université Côte d’Azur and Columbia University

ANTE QU, Stanford University

GEORGE DRETTAKIS, Inria and Université Côte d’Azur

EITAN GRINSPUN and CHANGXI ZHENG, Columbia University

Thin shells — solids that are thin in one dimension compared to the other two — often emit rich nonlinear sounds when struck. Strong excitations can even cause chaotic thin-shell vibrations, producing sounds whose energy spectrum diffuses from low to high frequencies over time — a phenomenon known as wave turbulence. It is all these nonlinearities that grant shells such as cymbals and gongs their characteristic “glinting” sound. Yet, simulation models that efficiently capture these sound effects remain elusive.

We propose a physically based, multi-scale reduced simulation method to synthesize nonlinear thin-shell sounds. We first split nonlinear vibrations into two scales, with a small low-frequency part simulated in a fully nonlinear way, and a high-frequency part containing many more modes approximated through time-varying linearization. This allows us to capture interesting nonlinearities in the shells’ deformation, tens of times faster than previous approaches. Furthermore, we propose a method that enriches simulated sounds with wave turbulent sound details through a phenomenological diffusion model in the frequency domain, and thereby sidestep the expensive simulation of chaotic high-frequency dynamics. We show several examples of our simulations, illustrating the efficiency and realism of our model.

CCS Concepts: • **Computing methodologies** → **Physical simulation**; • **Applied computing** → Sound and music computing;

Additional Key Words and Phrases: Sound synthesis, thin shells, reduced simulation, wave turbulence

ACM Reference Format:

Gabriel Cirio, Ante Qu, George Drettakis, Eitan Grinspun, and Changxi Zheng. 2018. Multi-Scale Simulation of Nonlinear Thin-Shell Sound with Wave Turbulence. *ACM Trans. Graph.* 37, 4, Article 110 (August 2018), 14 pages. <https://doi.org/10.1145/3197517.3201361>

1 INTRODUCTION

Thin shells produce rich and complex sounds: from containers such as trash cans and plastic bottles, to musical instruments like cymbals or gongs. Once excited, vibrational displacements as small as the thickness trigger highly nonlinear dynamics, such as time-varying amplitudes of the spectrum harmonics, energy exchange between vibrational modes, and ultimately chaos and turbulence [Chaigne et al. 2005]. As a result, thin-shell sound has time-varying details across a wide frequency spectrum. In this paper, we present a method

Authors’ addresses: Gabriel Cirio, Inria, Université Côte d’Azur and Columbia University; Ante Qu, Stanford University; George Drettakis, Inria and Université Côte d’Azur; Eitan Grinspun; Changxi Zheng, Columbia University.

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2018/8-ART110 \$15.00 <https://doi.org/10.1145/3197517.3201361>

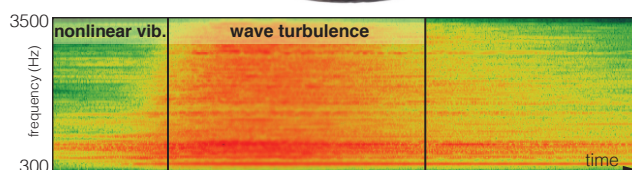
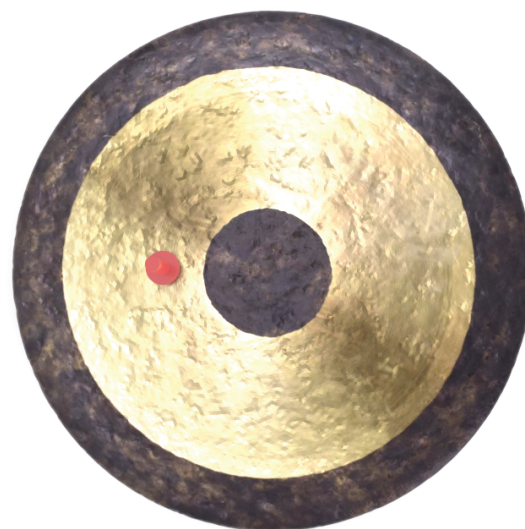


Fig. 1. **Splashy Tamtam.** A Tamtam gong produces rich nonlinear sounds. When it is struck hard (e.g., at the position indicated by the red dot), its vibrational energy quickly cascades from low to high frequencies — a phenomenon known as *wave turbulence*, and produces rich sound details in a wide spectrum. Our method is able to efficiently simulate nonlinear thin-shell sounds, and enrich them with a fast phenomenological model of wave turbulence. The nonlinear sound and the emergence of wave turbulence are shown in the spectrogram on the bottom and the supplemental video.

that efficiently captures these challenging nonlinear sound effects, including wave turbulence in particular.

Recent advances in physics-based sound models have enabled the synthesis of nonlinear sounds from thin shells, often at a prohibitive simulation cost [O’Brien et al. 2001]. The Harmonic Shells model [Chadwick et al. 2009] significantly reduces the runtime simulation cost, but requires a heavy precomputation process that learns a thin-shell cubature scheme from full simulation data. While the resulting sounds capture certain nonlinearities such as pitch glide, other important nonlinear effects such as the turbulence-like energy cascade are still missing.

We propose a multi-scale reduced simulation method to synthesize nonlinear thin-shell sounds. In comparison to previous approaches, the advantages of our method are threefold: *i)* Our pre-computation does not require any full-space simulation nor any training step, and is therefore significantly faster than classic cubature schemes. *ii)* At runtime, our simulation is tens of times faster than the state-of-the-art approach for qualitatively similar sounds. *iii)* More notably, our method is able to capture a wide spectrum of nonlinearities and greatly enrich the resulting sound details.

A remarkable nonlinearity that our method captures is *wave turbulence*, a phenomenon that emerges when the thin shell is struck hard [Touzé et al. 2012; Zakharov et al. 1992]. Here the term “turbulence” indicates an analogy with fluid turbulence wherein the energy of the system is transferred through scales. With a strong excitation, thin-shell vibration becomes chaotic, and the vibrational energy cascades from low to high frequencies. Eventually, the vibrational energy — and the emitted sound — is spread over a wide frequency spectrum. The perceptual importance of wave turbulence has long been recognized by musicians: they use a large metallic “thunder sheet” to produce dramatic shimmering sounds, and describe thin-shell instruments like cymbals and gongs as being “splashy” and as having “lots of spread” (see Figure 1). However, a computational model of wave turbulence sound remains elusive in existing thin-shell simulation methods.

To capture the nonlinear inter-modal energy transfers and complex turbulent behavior proper to thin-shell sounds, one could simulate the fully nonlinear equations of motion with sufficiently fine spatial and temporal discretization. However, such simulations are intractable. The simulation remains expensive even in a reduced subspace, due to the high number of modes and their nonlinear forcing involved in wave turbulence.

To address this computational bottleneck, the cornerstone of our method is a multi-scale view of a thin-shell’s nonlinear vibration. We model the vibration at three scales. First, we split the nonlinear vibration into two parts: a low-frequency part contributing to large-scale vibration and a high-frequency part usually in small scale. The low-frequency vibration consists of a small number of vibrational modes simulated in a fully nonlinear fashion. While the high-frequency vibration contains many more modes, they can be approximated through time-varying linearization. Both parts are coupled together by a frequency-splitting scheme in a two-way manner. Meanwhile, by carefully applying an isometry approximation to thin-shell bending, we reformulate the nonlinear internal force as a cubic polynomial whose terms are dynamically pruned at runtime. In consequence, the simulation cost is largely reduced.

Furthermore, we model wave turbulence across low and high frequencies, as the turbulence-like energy cascade develops at low frequencies and evolves toward high frequencies. Instead of painstakingly simulating wave turbulence, we generate wave turbulence sounds procedurally. By augmenting a recently developed phenomenological model [Humbert et al. 2016], we capture the cascading behavior of the energy spectrum in turbulent regimes. Through analyzing system chaos using the theory of dynamical systems, we couple the phenomenological model with simulated thin-shell dynamics. This approach yields a user-controlled tool that enriches the simulated sounds with turbulent details at a minimal cost.

In summary, our major technical contributions include:

- A frequency-splitting scheme to simulate nonlinear modal vibration of many modes.
- A reformulation of the thin-shell bending force as a cubic polynomial. Based on this formulation, we further develop a pruning algorithm that drastically reduces the cost of the internal force evaluation.
- A fast, user-controlled tool that enriches the simulated sound with a physically inspired wave turbulence sound texture.

2 RELATED WORK

2.1 Thin-shell dynamics

Thin plates and shells are ubiquitous in computer animation. Simulation methods have been used early on to animate large-deformation soft materials such as cloth [Baraff and Witkin 1998; Terzopoulos et al. 1987], small-deformation stiff materials [Chadwick et al. 2009], and elastoplastic materials such as crumpling paper [Narain et al. 2013].

While various thin-shell models have been developed in mechanics [Chapelle and Bathe 2010], hinge-based bending methods are often preferred for graphics applications for their simplicity and efficiency [Bridson et al. 2003; Grinspun et al. 2003]. Under the assumption of isometric deformations, hinge-based energies can be simplified into low-order polynomials in thin plates [Bergou et al. 2006] and shells [Garg et al. 2007]. We also harness the isometry assumption, but apply it to the formulation of forces rather than potential energy. Our approach yields a cubic polynomial form of bending forces and avoids sound artifacts that would be produced by previous polynomial models.

For membrane energies, Gingold et al. [2004] proposed a model that penalizes the deviation of triangle edges from their rest lengths. Volino et al. [2009] later used the St. Venant-Kirchhoff (StVK) model, arguably the simplest material model able to capture geometric nonlinearity. The StVK energy is quartic in the deformation degrees of freedom, and the forces are therefore cubic. Barbic and James [2005] have already exploited this polynomial form in reduced models for 3D volumetric meshes. Here we also use this property to compute membrane forces in a reduced space of 2D shells.

2.2 Nonlinear vibrations

There is a rich literature on nonlinear vibrations of thin structures [Nayfeh and Mook 1995]. We refer to [Moussaoui and Benamar 2002] for a comprehensive literature review. Most of these works simulate nonlinear vibrational motion in a limited frequency range. Very few aimed for simulating the emitted thin-shell sounds [Cremer and Heckl 2013].

To synthesize sound, one way is to explicitly timestep a finite element model at audio rates [O’Brien et al. 2001]. This idea was further developed by Bilbao [2008], who designed an energy conserving finite difference discretization and time-stepping scheme to synthesize the sound of nonlinear plates. Bilbao [2010] later showed that using specifically constructed discretizations for special domains like rectangles and spherical caps, one could capture nonlinear and turbulent effects at a very coarse resolution of a few hundred elements.

Pierce and Van Duyne [1997] used all-pass nonlinear passive filters to imitate nonlinear mode-coupling effects. Linear eigenmodes have been used for nonlinear subspace integration and resolve coupling among modes [Bathe 2007]. Our work uses a similar approach to construct vibrational modes. More recently, Cirio et al. [2016] proposed a time-varying linear sound model accelerated by the component mode synthesis technique. Their method focuses on synthesizing sounds from crumpling shells but not nonlinear phenomena such as wave turbulence.

Ducceschi and Touzé [2015] recently presented a method that synthesizes impressive sounds of *planar* gongs and cymbals with up to a thousand modes. Their method exploits the analytical expressions of vibrational modes for regular planar plates, and employs *ad hoc* rules to limit the number of couplings for a given mode. However, their approach is limited to flat sheets, and it is unclear what these rules are for arbitrary shells and how to apply them.

Most related to ours is the work of Chadwick et al. [2009]. It uses precomputed cubature schemes to compute nonlinear forces in reduced space, and thereby greatly reduces the runtime computational cost, producing high quality nonlinear sounds. This comes at the cost of an expensive cubature optimization step, as well as the need to precompute a training set using full-space simulations. In contrast, our method does not require full-space simulation nor a training step, and our runtime simulation appears more than an order of magnitude faster for qualitatively similar sound quality. In addition, we propose a procedural wave turbulence model that further enriches simulated sounds.

2.3 Wave turbulence

Wave turbulence is a chaotic wave phenomenon, wherein the energy of the system is transferred through scales, often referred as an energy cascade [Zakharov et al. 1992]. Therefore, the term “turbulence” is an analogy with hydrodynamic turbulence [Ducceschi et al. 2014].

Wave turbulence is a universal phenomenon, not just in nonlinear shells. The Kolmogorov–Zakharov spectrum emerges from these phenomena, as analytically shown in [Newell et al. 2001]. Similar energy spectra have been derived for turbulent phenomena in different media: water waves [Falcon et al. 2007; Zakharov and Filonenko 1967], plasmas [Musher et al. 1995], nonlinear optics [Dyachenko et al. 1992] and quantum liquids [Vinen 2000]. In computer graphics, the Kolmogorov spectrum has been used to enrich the details of animated fluids [Kim et al. 2008; Rasmussen et al. 2003].

We aim to capture the wave turbulence in thin shells, especially its manifestation in sound. The resulting sound exhibits a wide-band spectrum where linear modal vibrational frequencies can no longer be easily distinguished [Chaigne et al. 2005]. Thin-plate wave turbulence has been extensively studied using the Föppl-von Kármán equation for large deflections [Ducceschi et al. 2014; Düring et al. 2006; Touzé et al. 2012]. As the vibrational energy is transferred from the input energy scales to the high-frequency scales, the Kolmogorov-Zakharov spectrum emerges again.

The connection of wave turbulence and Kolmogorov spectrum also motivates the development of phenomenological models used in many different contexts [Falkovich and Shafarenko 1991; Josseland

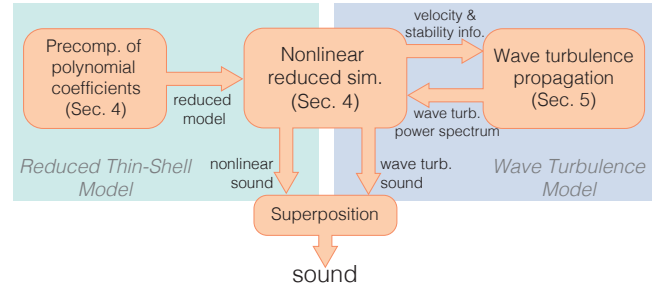


Fig. 2. **Method overview.** (Left) Our method precomputes a polynomial form for runtime evaluation of thin-shell internal forces. Our runtime simulation is built on a new frequency-splitting scheme in reduced modal space, and is further accelerated by dynamically pruning polynomial terms. (Right) Our reduced simulation not only produces nonlinear sounds but also captures chaos in low-frequency modes, which drive a phenomenological model to produce a wave turbulence sound that enriches the simulated sound.

et al. 2006; Leith 1967]. In particular, the recent work of Humbert et al. [2016] uses a nonlinear diffusion equation in the frequency domain to reproduce the wave turbulence spectrum that agrees with experiments. We augment this phenomenological approach with a damping term and a source term tailored for diffusing the modal energies produced by our reduced simulation. This model enables us to procedurally generate wave turbulence at a minimal cost.

3 METHOD OVERVIEW

This section provides an overview of our multi-scale treatment of thin-shell sound synthesis (see Figure 2). We discretize the geometry of a thin shell with a triangle mesh. The dynamics of the discretized shell are governed by Newton’s second law of motion,

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{D}\dot{\mathbf{u}} + \mathbf{f}_{\text{int}}(\mathbf{u}) = \mathbf{f}_{\text{ext}}, \quad (1)$$

where \mathbf{u} is the vector of nodal displacements of the mesh, \mathbf{M} and \mathbf{D} are respectively the mass and damping matrices, \mathbf{f}_{ext} is the vector of external nodal forces such as gravity and contact, and $\mathbf{f}_{\text{int}}(\mathbf{u})$ represent the internal forces, including membrane and bending forces (§4). Crucially, the forces \mathbf{f}_{int} are nonlinear with respect to the configuration degrees of freedom.

We accelerate computation via model reduction [Krysl et al. 2001]. Let \mathbf{U} denote the basis of a reduced displacement space and \mathbf{q} describe thin-shell displacement in the reduced space. Substituting $\mathbf{u} = \mathbf{U}\mathbf{q}$ into (1) and premultiplying by \mathbf{U}^T , we obtain the reduced governing equation for thin-shell vibration,

$$\ddot{\mathbf{q}} + \tilde{\mathbf{D}}\dot{\mathbf{q}} + \tilde{\mathbf{f}}_{\text{int}}(\mathbf{q}) = \mathbf{U}^T \mathbf{f}_{\text{ext}} \quad (2)$$

where $\tilde{\mathbf{f}}_{\text{int}}(\mathbf{q}) = \mathbf{U}^T \mathbf{f}_{\text{int}}(\mathbf{U}\mathbf{q})$, and $\tilde{\mathbf{D}} = \mathbf{U}^T (\alpha\mathbf{M} + \beta\mathbf{K}_{\mathbf{u}})\mathbf{U}$ is the reduced Rayleigh damping matrix, with $\mathbf{K}_{\mathbf{u}}$ being the stiffness matrix, $\mathbf{K}_{\mathbf{u}} = \frac{\partial}{\partial \mathbf{u}} \mathbf{f}_{\text{int}}(\mathbf{u})$. Following Chadwick et al. [2009], we construct \mathbf{U} through linear modal analysis that solves a generalized eigenvalue problem $\mathbf{K}_0\mathbf{U} = \mathbf{M}\mathbf{U}\mathbf{S}$ [Shabana 2012], where $\mathbf{K}_0 = \mathbf{K}_{\mathbf{u}}|_{\mathbf{u}=\mathbf{0}}$ and \mathbf{S} is the resulting diagonal matrix of squared modal frequencies.

3.1 Frequency Splitting

The column vectors of \mathbf{U} describe distinct vibrational modes that have different vibrational frequencies. Because of the nonlinear internal force \mathbf{f}_{int} , these frequencies may vary, depending on the

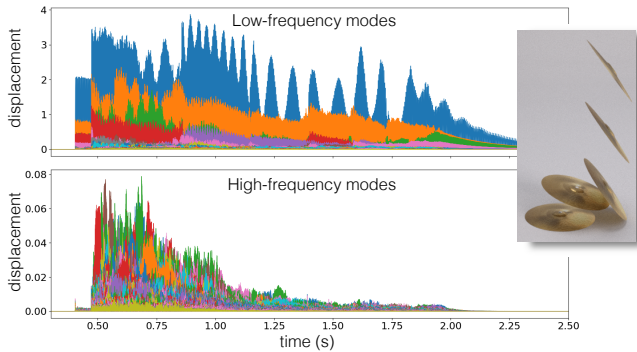


Fig. 3. **Modal displacements** of low- and high-frequency modes. We take the example of a falling cymbal (right and also see video), and plot the modal displacements of individual modes in low ($<400\text{Hz}$) and high ($>4000\text{Hz}$) frequencies with respect to time. It is evident that the modal displacements of low-frequency modes are generally orders of magnitude larger than those of high-frequency modes (please note the different scales of the Y-axes in both panels). This observation motivates our frequency-splitting scheme.

deformation configuration \mathbf{u} . We observe that, regardless of the variation, the frequency values at the rest configuration $\mathbf{u} = \mathbf{0}$ (i.e., the square root of $S_{i,i}$, the diagonal values of S) reliably indicate the frequency scales of individual modes. In other words, low-frequency modes remain in a low-frequency band throughout, and likewise for high-frequency modes. More importantly, we notice that higher-frequency modes have small modal displacements (Figure 3). This is because to reach the same vibrational amplitude, high-frequency modes consume more kinetic energy than their low-frequency counterparts.

These observations underpin our proposed frequency splitting scheme. We split the modal basis \mathbf{U} into low-frequency and high-frequency modes, \mathbf{U}_L and \mathbf{U}_H , based on a user-controlled frequency threshold for the rest-state frequencies. We then express the displacement as

$$\mathbf{u} = \mathbf{u}_L + \mathbf{u}_H = \begin{bmatrix} \mathbf{U}_L & \mathbf{U}_H \end{bmatrix} \begin{bmatrix} \mathbf{q}_L \\ \mathbf{q}_H \end{bmatrix}. \quad (3)$$

Here the high-frequency displacement \mathbf{u}_H is small compared to low-frequency displacement \mathbf{u}_L . This decomposition allows us to linearize the internal force \mathbf{f}_{int} around \mathbf{u}_L through a first-order Taylor expansion,

$$\mathbf{f}_{\text{int}}(\mathbf{u}) \approx \mathbf{f}_{\text{int}}(\mathbf{u}_L) + \frac{\partial \mathbf{f}_{\text{int}}}{\partial \mathbf{u}_L}(\mathbf{u} - \mathbf{u}_L). \quad (4)$$

Since $\mathbf{u}_H = \mathbf{u} - \mathbf{u}_L$, we express $\mathbf{f}_{\text{int}}(\mathbf{u})$ using \mathbf{u}_L and \mathbf{u}_H as

$$\mathbf{f}_{\text{int}}(\mathbf{u}) \approx \mathbf{f}_{\text{int}}(\mathbf{u}_L) + \mathbf{K}_L \mathbf{u}_H, \quad (5)$$

where \mathbf{K}_L is the stiffness matrix at the displacement \mathbf{u}_L (i.e. without any contribution from high-frequency displacements). Introducing (5) into the reduced equations of motion (2) yields a new form of the equation, one that we will numerically integrate in this work:

$$\ddot{\mathbf{q}} + \tilde{\mathbf{D}}\dot{\mathbf{q}} + \tilde{\mathbf{f}}_{\text{int}}(\mathbf{q}_L) + \tilde{\mathbf{K}}_L \mathbf{q}_H = \mathbf{U}^T \mathbf{f}_{\text{ext}}. \quad (6)$$

The internal force term now has two parts, $\tilde{\mathbf{f}}_{\text{int}}(\mathbf{q}_L) = \mathbf{U}^T \mathbf{f}_{\text{int}}(\mathbf{U}_L \mathbf{q}_L)$ and $\tilde{\mathbf{K}}_L \mathbf{q}_H = \mathbf{U}^T \mathbf{K}_L \mathbf{U}_H \mathbf{q}_H$.

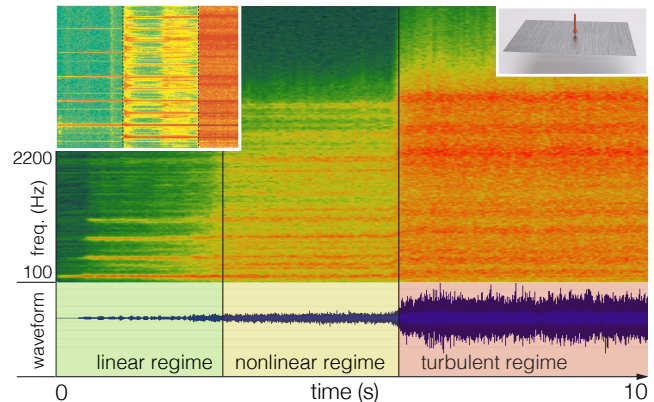


Fig. 4. **Thin-shell bifurcation.** We excite a thin plate with increasing forces (the red arrow in the top-right inset) and simulate its dynamical responses (Eq. (6)). As the force increases, its vibration bifurcates, changing from linear vibration (left) to nonlinear (middle), and finally moving into a turbulent regime (right). This spectrogram is generated without any wave turbulence enrichment, indicating that Eq. (6) is able to capture chaos, albeit in low frequencies. We note that this spectrogram is qualitatively close to spectrograms from physical experiments, as shown in Figure 1 of [Touzé et al. 2012] (also shown in the top-left inset, **Image courtesy of Cyril Touzé**).

In this form, fully nonlinear dynamics are captured by $\tilde{\mathbf{f}}_{\text{int}}(\mathbf{q}_L)$ at low frequencies. At high frequencies, while the force is linear with respect to \mathbf{q}_H , the stiffness matrix $\tilde{\mathbf{K}}_L$ varies with respect to \mathbf{u}_L , thereby coupling together low- and high-frequency displacements. This coupling is two-way: low-frequency displacement $\mathbf{u}_L = \mathbf{U}_L \mathbf{q}_L$ affects $\tilde{\mathbf{K}}_L$ and in turn drives the high-frequency force $\tilde{\mathbf{K}}_L \mathbf{q}_H$. Meanwhile, the force vector $\tilde{\mathbf{K}}_L \mathbf{q}_H$ spans the entire space of \mathbf{q} , not just the subspace that \mathbf{q}_H resides in, thus influencing low-frequency displacement as well.

By splitting in the frequency domain, we can apply different, appropriate models to the two frequency bands. There are often many vibrational modes (i.e., a long vector \mathbf{q}) within the human hearing range. Directly integrating (2) with a long vector \mathbf{q} is expensive, as noted in [Chadwick et al. 2009]. Our scheme is able to sidestep this hurdle by keeping \mathbf{q}_L short (in practice less than 50) while ensuring \mathbf{q}_H is long enough to cover sufficient sound spectrum. Furthermore, this idea enables us to compute both low- and high-frequency parts of the internal forces efficiently, by exploiting proper membrane and bending models. Indeed, as will be discussed in §4.2, we are able to reduce $\tilde{\mathbf{f}}_{\text{int}}(\mathbf{q}_L)$ into a cubic polynomial, with an $O(m_L^3)$ evaluation complexity, where m_L is the size of \mathbf{q}_L ; the computation of $\tilde{\mathbf{K}}_L \mathbf{q}_H$ is comparably fast. As a result, our runtime simulation is much faster than the cubature scheme used in [Chadwick et al. 2009] (see §7.3). The two later sections (§4 and §5) are dedicated to describing our fast computation of the two internal force terms.

3.2 Wave Turbulence Enrichment

Explicitly simulating wave turbulence requires us to solve the Föppl-von Kármán equations, a system of fourth-order nonlinear PDEs [Chia 1980]. This is rather expensive: when the vibrational energy is cascaded to high frequencies, one has to solve the PDE with a high-resolution mesh to capture high-frequency wave turbulence.

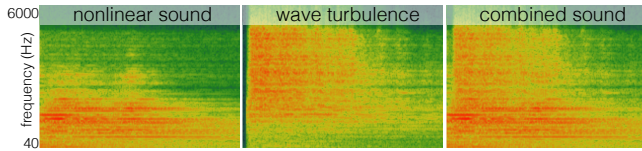


Fig. 5. **Wave turbulence enrichment.** We simulate the sound from a thunder sheet (in Figure 13): (Left) the sound spectrogram resulting from nonlinear shell simulation, (Middle) the wave turbulence spectrogram generated by diffusing vibrational energies, and (Right) the final sound spectrogram that mixes the simulated sound with turbulent details.

Inspired by the work of adding synthetic turbulence for fluids [Kim et al. 2008], we propose an algorithm that procedurally generates a sound “texture” of wave turbulence in a wide frequency spectrum. The resulting wave turbulence texture is in turn added to the simulated thin-shell sound to enrich its details.

Our wave turbulence sound is generated in analysis-synthesis fashion. First, at each timestep during thin-shell simulation, we measure the chaos in low-frequency modes using a well-defined quantity, called the *Lyapunov exponent*, from the theory of dynamical systems [Wiggins 2003]. We are able to capture low-frequency chaos, because the low-frequency force \tilde{f}_{int} in (6) is fully nonlinear, and a strong excitation can render the low-frequency vibrations chaotic (see Figure 4). We use the measured Lyapunov exponent at every timestep to estimate how much wave turbulence should be generated. Consequently, a system that is not chaotic will not produce wave turbulence sound.

After the shell simulation, we synthesize a time-varying power spectrum (i.e., power spectrogram) of the wave turbulence sound. Without explicitly simulating the energy cascade, we generate the power spectrogram using a phenomenological model that diffuses the vibrational power spectrum over time. The amount of energy cascaded at each timestep is related to the modal vibrational energy as well as system chaos measured by the Lyapunov exponents. From the power spectrogram, we construct the wave turbulence sound and merge it with the simulated thin-shell sound (Figure 5).

4 LOW-FREQUENCY FORCE

We now describe how to efficiently compute the low-frequency part of the internal force, $\tilde{f}_{\text{int}}(\mathbf{q}_L)$, to integrate the reduced equation of motion (6). A classic approach of evaluating nonlinear forces in reduced space is to use a cubature scheme [An et al. 2008], which requires preparation of training data using full-space simulations. In addition, cubature schemes for thin shells exhibits slower convergence rates than volumetric models [Chadwick et al. 2009].

Instead, we found that we could efficiently compute internal forces directly in the subspace when approximating $\tilde{f}_{\text{int}}(\mathbf{q}_L)$ using a cubic polynomial of \mathbf{q}_L . We first approximate the full-space force $\tilde{f}_{\text{int}}(\mathbf{u})$ as a cubic polynomial of \mathbf{u} . Then, because of the linear relationship $\mathbf{u} = \mathbf{U}\mathbf{q}$, the reduced force $\tilde{f}_{\text{int}}(\mathbf{q}) = \mathbf{U}^T \tilde{f}_{\text{int}}(\mathbf{U}\mathbf{q})$ can be

expressed as

$$\tilde{f}_{\text{int}}(\mathbf{q}) = \sum_{i=1}^m L_i q_i + \sum_{i=1}^m \sum_{j \geq i}^m Q_{ij} q_i q_j + \sum_{i=1}^m \sum_{j \geq i}^m \sum_{k \geq j}^m C_{ijk} q_i q_j q_k, \quad (7)$$

where m is the length of \mathbf{q} , and q_i is the i -th element of \mathbf{q} . The coefficients, L_i , Q_{ij} , and C_{ijk} , are all vectors having the same length as \mathbf{q} . Then, the low-frequency force $\tilde{f}_{\text{int}}(\mathbf{q}_L)$ can be expressed by using \mathbf{q}_L in (7) (and $\mathbf{q}_H = \mathbf{0}$). This results in a cubic polynomial with many fewer terms:

$$\tilde{f}_{\text{int}}(\mathbf{q}_L) = \sum_{i=1}^{m_L} L_i q_i + \sum_{i=1}^{m_L} \sum_{j \geq i}^{m_L} Q_{ij} q_i q_j + \sum_{i=1}^{m_L} \sum_{j \geq i}^{m_L} \sum_{k \geq j}^{m_L} C_{ijk} q_i q_j q_k,$$

where m_L is the length of \mathbf{q}_L , and $m_L \ll m$. Precomputation of the coefficients does not require any full-space simulation or training step, making it faster than using cubature schemes. Moreover, since the vector \mathbf{q}_L is short, runtime evaluation of the polynomial is fast, with a complexity of $O(m_L^3)$. In the next section, we will see that this polynomial form is also beneficial to the fast evaluation of high-frequency forces. In consequence, our runtime simulation is significantly sped up.

Our formulation is derived from the elastic shell model. The elastic deformation energy, $E(\mathbf{u})$, consists of two parts, $E(\mathbf{u}) = E_m(\mathbf{u}) + E_b(\mathbf{u})$, where $E_m(\mathbf{u})$ and $E_b(\mathbf{u})$ are the membrane strain energy and bending energy of the shell, respectively. The internal force in full space is defined as $\tilde{f}_{\text{int}}(\mathbf{u}) = -\nabla_{\mathbf{u}} E(\mathbf{u})$. Next, we start from a particular choice of membrane and bending models, and show how to derive the cubic polynomial force from these models.

4.1 Membrane Energy

The membrane energy model is straightforward to choose. Since our shells are almost inextensible, strains will remain small and we can therefore use a linear strain-stress relationship with a nonlinear strain measure. Inspired by previous work [Barbič and James 2005], we use the St. Venant-Kirchhoff (StVK) model in its cubic polynomial form for reduced-order force computation, applying it to the context of our thin shell triangular discretization.

Starting from the work of [Volino et al. 2009] which discretizes the StVK model for thin-shell meshes, we reformulate the StVK full-space forces into a cubic polynomial with respect to \mathbf{u} , and expand the resulting polynomial coefficients into their corresponding linear, quadratic, or cubic reduced polynomial vector. For more details, we refer the reader to the supplementary document.

4.2 Bending Energy

Caution is needed when computing the bending force. At first glance, a natural choice is the hinge-based bending model [Garg et al. 2007], wherein the bending energy associated to an edge i of the shell mesh is defined as:

$$E_b(\mathbf{u})_i = \frac{3|\bar{\mathbf{e}}_i|^2}{A_i} (1 - \cos(\theta_i - \bar{\theta}_i)). \quad (8)$$

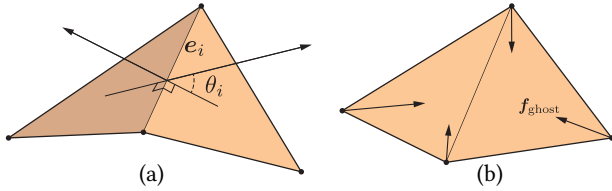


Fig. 6. (a) Notation used in the bending model (8). (b) Directly applying a cubic polynomial energy [Garg et al. 2007] yields nonzero ghost forces even at the rest configuration (see video for the sound artifacts).

The notation here is shown in Figure 6-a: \bar{A}_i denotes the total area of the two triangles incident to the edge \bar{e}_i , and θ_i is the dihedral angle of the edge on the deformed mesh. A bar (e.g., $\bar{\theta}_i$) denotes the quantity evaluated on the undeformed mesh. For most thin shells producing interesting sounds (e.g., shells with metallic or plastic material), the membrane stiffness is high, and therefore it is reasonable to assume that the shell bending undergoes isometric deformations. Garg et al. [2007] applied this assumption to the bending energy (8) and reduced it to a cubic polynomial of \mathbf{u} .

This seems to suggest that, under the isometry assumption, we can benefit from a bending force in quadratic polynomial form with respect to \mathbf{u} . However, perhaps surprisingly, we have found that this formulation from [Garg et al. 2007] leads to nonzero bending forces at the *undeformed* shell state. The ghost bending force at $\mathbf{u} = \mathbf{0}$ is due to the negative bending energy caused by applying the isometry assumption, and is always within the shell membrane (i.e., in the plane of one of the two triangles in Figure 6-b). In Appendix A, we provide an explanation why the ghost bending forces occur.

These ghost bending forces might not produce noticeable artifacts visually: since they are always within the shell membrane, they can be quickly absorbed and balanced by the stiff membrane forces before producing any noticeable motion. Yet, they ruin the sound synthesis process. They produce membrane vibrations, perceived as high-pitch ringing, even before any external force is applied (see video). They also introduce local stiffening and affect bending modes. To make matters even worse, because the ghost bending forces are tangent to their respective triangles, they rotate when the shell undergoes rigid rotation. As a result, rotational rigid-body modes become mixed up with deformation modes, which completely spoils the modal analysis step in §3.

We therefore seek a new way of expressing the hinge-based bending model (8) in polynomial form. We start by directly taking the gradient of (8) with respect to \mathbf{u} to obtain an expression of the bending force. This yields a rational polynomial whose order is much higher than three, as detailed in the supplemental document.

This rational polynomial, although complex, expresses the bending force precisely. Next, we simplify the bending force expression by applying the isometry assumption. This means that all the edge lengths and in-plane angles of all triangles are preserved. Thus, in all the terms of the rational polynomial, we replace norms of all the edges and all the dot products of vectors within individual triangles by their values at the rest state, as shown in the supplemental document. We note that this simplification does not change the bending force evaluation at the rest state; no ghost bending force would be generated at rest.

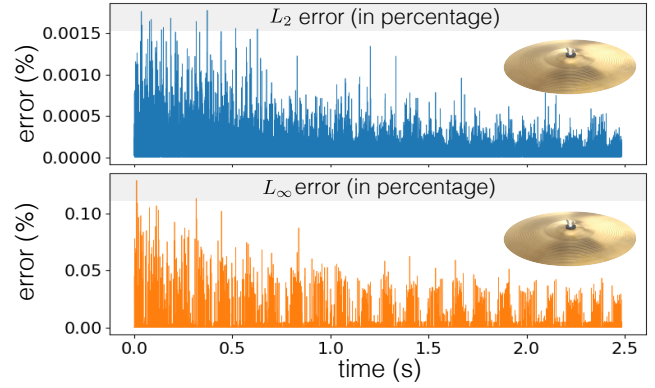


Fig. 7. **Quartic terms.** Using the falling cymbal as an example, we compute the L_2 (top) and L_∞ errors introduced by dropping quartic terms in the force computation at each timestep. The L_2 error is computed as $\frac{\|f_c - f_q\|_2}{\|f_q\|_2}$, where f_q and f_c are the internal forces evaluated with and without quartic terms. The L_∞ error is defined in a similar way.

Remarkably, this simplification yields a quartic polynomial with respect to \mathbf{u} , which we call the *isometric bending force*. We use symbolic computation to turn the simplified force expressions into polynomial form.

Furthermore, we notice that all quartic terms in the expression of the isometric bending force have the form $\mathbf{e}_a \mathbf{e}_b^T (\mathbf{e}_c \times \mathbf{e}_d)$, where \mathbf{e}_i is the vector of a triangle edge. We observe that, under the isometry assumption, the contribution of these quartic terms to the overall displacement is negligible even for moderately large deformations, as shown in Figure 7. In addition, we discovered that in the absence of local rigid rotation these quartic terms simply vanish. Appendix C provides an explanation of this interesting discovery. Consequently, we expand each of these quartic terms until we can isolate individual quartic products among degrees of freedom, and drop them. The resulting isometric bending force is a cubic polynomial of \mathbf{u} , producing no ghost bending force at the rest state. With this cubic polynomial, we substitute \mathbf{u} with $\mathbf{U}\mathbf{q}$, and obtain a cubic polynomial of \mathbf{q} as expressed in (7) (see details in Appendix B).

5 HIGH-FREQUENCY FORCE

We now present our fast algorithm for evaluating the high-frequency force term, $\bar{\mathbf{K}}_L \mathbf{q}_H$, appearing in (6). Recall that $\bar{\mathbf{K}}_L = \mathbf{U}^T \mathbf{K}_L \mathbf{U}_H$, which can be rewritten as

$$\begin{aligned} \bar{\mathbf{K}}_L &= \mathbf{U}^T \frac{\partial}{\partial \mathbf{q}_H} f_{\text{int}}(\mathbf{U}_L \mathbf{q}_L + \mathbf{U}_H \mathbf{q}_H) \Big|_{\mathbf{q}_L, \mathbf{q}_H = \mathbf{0}} \\ &= \frac{\partial}{\partial \mathbf{q}_H} \tilde{f}_{\text{int}}(\mathbf{q}) \Big|_{\mathbf{q}_L, \mathbf{q}_H = \mathbf{0}}. \end{aligned} \quad (9)$$

$\tilde{\mathbf{K}}_L$ is therefore the derivative of the reduced internal forces $\tilde{f}_{\text{int}}(\mathbf{q})$ with respect to the reduced coordinates \mathbf{q}_H . As shown in (7), $\tilde{f}_{\text{int}}(\mathbf{q})$ is approximated by a cubic polynomial of \mathbf{q} . After taking its partial derivative with respect to \mathbf{q}_H , many terms in the polynomial vanish because we are evaluating the derivative at $\mathbf{q}_H = \mathbf{0}$. First, any term that involves variables only in \mathbf{q}_L but not in \mathbf{q}_H vanishes in the derivative. Second, if a term involves at least two variables in \mathbf{q}_H , it also disappears in the partial derivative, because taking the

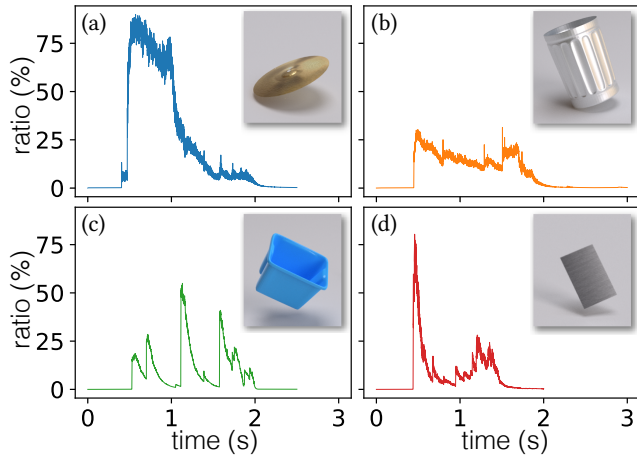


Fig. 8. **Pruning efficiency.** In four examples (a-d) that we will show in §7, we collect the ratio between the number of evaluated polynomial terms (after pruning) and the number of total terms at each timestep. Many terms are needed for accurate force evaluation only when the shell is heavily impacted. Most of the time, only a small number of terms are used. This pruning algorithm yields a total speedup of up to two orders of magnitude.

derivative eliminates only one of the q_H variables in that term. After discarding the vanishing terms, the high-frequency force can be written as the following polynomial:

$$\begin{aligned} \bar{\mathbf{K}}_L \mathbf{q}_H = & \sum_{i=m_L+1}^m L_i q_i + \sum_{i=1}^{m_L} \sum_{j=m_L+1}^m Q_{ij} q_i q_j + \\ & \sum_{i=1}^{m_L} \sum_{j \geq i}^{m_L} \sum_{k=m_L+1}^m C_{ijk} q_i q_j q_k, \end{aligned} \quad (10)$$

where m_L and m are the length of \mathbf{q}_L and \mathbf{q} , respectively. The coefficient vectors, L_i , Q_{ij} , and C_{ijk} , are precomputed as described in §4. Runtime evaluation of (10) has therefore a tractable complexity of $O(m_L^2 m)$. Next, we propose a simple but extremely efficient pruning technique to further reduce the evaluation cost.

5.1 Runtime Vector Pruning

We observed that many terms in (10) can become small during the simulation. Without loss of generality, consider a cubic term, $C_{ijk} q_i q_j q_k$. Summing up this term in the polynomial costs $4m$ operations (three multiplications and one addition for each of the m elements in vector C_{ijk}). On the other hand, the contribution of this term has an upper bound of $|C_{ijk}|_\infty |q_i q_j q_k|$. Here the infinity norm of C_{ijk} can be precomputed and stored. At runtime, computing this upper bound costs three multiplications. If the upper bound is sufficiently small (below a user set threshold), we can skip this term, saving $4m$ operations.

Because the modal displacement \mathbf{q} oscillates, the estimated upper bound $|C_{ijk}|_\infty |q_i q_j q_k|$ may sometimes oscillate around the pruning threshold, resulting in (subtle) temporal incoherence. To avoid this artifact, we use the value $\max(|q_i|, |\hat{q}_i|)$ as $|q_i|$ when computing the upper bound. Here $|\hat{q}_i|$ is the most recent peak value of $|q_i|$ right before the current timestep.

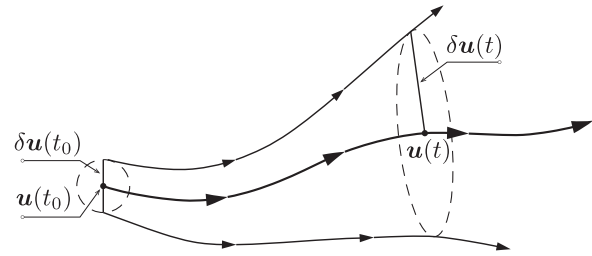


Fig. 9. **Interpretation of Lyapunov Exponents.** A solution $\mathbf{u}(t)$ of a dynamical system is a trajectory starting at an initial point $\mathbf{u}(t_0)$. A small perturbation leads to a different trajectory. The Lyapunov exponent measures how quickly an initial perturbation $\delta \mathbf{u}(t_0)$ at $\mathbf{u}(t_0)$ grows as the system evolves. In a chaotic system, this perturbation grows exponentially.

In practice, we apply the same pruning technique to all linear, quadratic, and cubic terms in both low- and high-frequency forces, while the evaluation of (10) is the major performance bottleneck of our runtime simulation. Consequently, the pruning threshold gives full control over the graceful degradation of quality, allowing the user to adjust the tradeoff between speed and quality. In all our final examples, we use the same threshold value (i.e., 10^{-3}), and observe up to $14\times$ speedups over the non-pruning evaluation, without noticeable degradation in sound quality.

6 SYNTHETIC WAVE TURBULENCE

We now show how to enrich simulated thin-shell sounds with synthetic wave turbulence. Here we are not aiming to simulate wave turbulence on triangle meshes, as this is fundamentally challenging, especially at high frequencies. Instead, our goal is to provide a user-controllable tool to add turbulent “textures” to the simulated sounds, akin in spirit to Perlin Noise [Perlin 1985] and fluid turbulence textures [Kim et al. 2008] in computer graphics.

6.1 Measuring Chaos

Wave turbulence is a manifestation of thin shells vibrating in a chaotic regime [Cadot et al. 2016]. Mathematically, the chaos of a dynamical system can be quantified through *Lyapunov exponent*, so we start by introducing this concept.

6.1.1 Lyapunov exponents. Consider a dynamical system such as that described in (1). The chaos of this system is characterized by its sensitivity to initial conditions: two initial conditions that differ by an infinitesimal perturbation $\delta \mathbf{u}(t_0)$ can yield diverging trajectories after a finite amount of time [Chaigne et al. 2005] (see Figure 9). When a system becomes chaotic, this divergence is asymptotically exponential (as $t \rightarrow \infty$), expressed by

$$|\delta \mathbf{u}(t)| \approx e^{\lambda t} |\delta \mathbf{u}(t_0)|, \quad (11)$$

where the rate $\lambda > 0$ is called the Lyapunov exponent [Wiggins 2003]. The value of λ can be different for different directions of initial perturbation $\delta \mathbf{u}(t_0)$. Among all λ values, the largest one is called the *Maximal Lyapunov exponent* (MLE).

6.1.2 Numerical computation. To numerically evaluate the MLE of our thin-shell system, we follow the algorithm described in [Rugonyi and Bathe 2003]. Consider a normalized random perturbation

$\delta \mathbf{q}_{L0}$ of (6) at time t_0 (i.e., $|\delta \mathbf{q}_{L0}|_2 = 1$) and zero velocity perturbation. In a single timestep, the evolution of $\delta \mathbf{q}_{L0}$ satisfies the differential equation,

$$\delta \ddot{\mathbf{q}}_{\mathbf{L}} + \tilde{\mathbf{K}}_{\mathbf{L}} \delta \mathbf{q}_{\mathbf{L}} = \mathbf{0}. \quad (12)$$

Here we neglect the damping term for simplicity and linearize the force $\tilde{\mathbf{f}}_{\text{int}}$ at $\mathbf{q}_{\mathbf{L}}$ because we consider only a single timestep. Through numerical integration, we evolve the perturbation over a timestep and obtain $\delta \mathbf{q}_{L1}$. Let $d_1 = |\delta \mathbf{q}_{L1}|_2$ denote its norm. Next, we use d_1 to normalize both $\delta \mathbf{q}_{L1}$ and its current velocity perturbation $\delta \dot{\mathbf{q}}_{L1}$, and use the normalized quantities as the initial condition of (12) to continue the numerical integration of the perturbation vector. These steps repeat at each timestep, and we obtain a time series of Lyapunov exponent values according to (11):

$$\lambda(t_k) = \frac{1}{t_k} \sum_{i=1}^k \ln d_i. \quad (13)$$

At each timestep, normalization is needed to avoid numerical overflow, since the perturbation can otherwise grow exponentially due to chaos. Also, at each timestep before integrating the perturbation, we update $\tilde{\mathbf{K}}_{\mathbf{L}}$, setting it as the stiffness matrix at the current low-frequency displacement $\mathbf{u}_{\mathbf{L}}$. In fact, $\tilde{\mathbf{K}}_{\mathbf{L}}$ has been computed at every timestep for integration of the thin-shell vibration in (6) and can be reused here. So the overhead of computing $\lambda(t_k)$ is minimal.

6.1.3 Local stability indicator. It can be proved that the time series of $\lambda(t_k)$ converges to the MLE of the system regardless of the choice of the initial perturbation $\delta \mathbf{q}_{L0}$ [Wolf et al. 1985]. A positive MLE is usually considered as an indication that the system becomes chaotic as $t \rightarrow \infty$, but it contains no information about the *local* dynamic stability, one that indicates how likely the system changes from stable to unstable (chaos) at a specific instant. Fortunately, as pointed out in [Rugonyi and Bathe 2003], the local dynamic stability can be measured by the average *slope* of the curve $t\lambda(t)$. The average is required since $\lambda(t)$ slightly oscillates as $\tilde{\mathbf{K}}_{\mathbf{L}}$ changes over time. By definition, the slope is always non-negative. The higher the slope is, the more chaos the system is currently experiencing. In our case, although $\lambda(t_k)$ is computed using low-frequency modes only, it is sufficient to indicate wave turbulence (see Figure 10), because wave turbulence, while evolving towards high frequencies, usually develops at low frequencies [Cadot et al. 2016].

In summary, we compute $t_k \lambda(t_k)$ as given in (13) at each timestep (see Algorithm 1). We then smooth it out and use finite differences to compute its slope $s(t_k)$ at each timestep. Afterwards, we normalize all $s(t_k)$ into the range $[0, 1]$. $s(t_k)$ is our measure of how much wave turbulence sound needs to be generated at each timestep, as described next.

6.2 Synthesizing Wave Turbulence

With the local stability indicator computed at each timestep, we now synthesize wave turbulence sound by leveraging a phenomenological model of cascading energy [Humbert et al. 2016]. This phenomenological model treats the vibrational energy spectrum in frequency domain. Let $E(f, t)$ denote the shell's vibrational energy in spectral band f at time t , which is the quantity we will estimate. The cascading energy is modeled as a diffusion process in the

ALGORITHM 1: Numerical computation of $\lambda(t_k)t_k$

Input: Random initial modal perturbation vector $\delta \mathbf{q}_{L0}$.

Output: The time series of $\lambda(t_k)t_k$.

Set $\delta \dot{\mathbf{q}}_{L0} = \mathbf{0}$;

foreach timestep i **do**

$d_i \leftarrow \delta \mathbf{q}_{Li}$;

 compute $\lambda(t_i)t_i$ using (13);

$\delta \mathbf{q}_{Li} \leftarrow \delta \mathbf{q}_{Li}/d_i$, $\delta \dot{\mathbf{q}}_{Li} \leftarrow \delta \dot{\mathbf{q}}_{Li}/d_i$;

$\delta \mathbf{q}_{L(i+1)}$, $\delta \dot{\mathbf{q}}_{L(i+1)} \leftarrow$ timestep (12) using $\delta \mathbf{q}_{L(i)}$ and $\delta \dot{\mathbf{q}}_{L(i)}$;

end

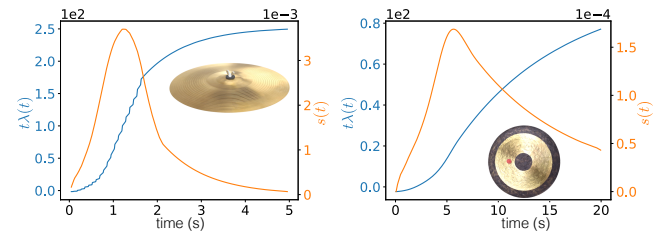


Fig. 10. **Stability indicator.** We plot $t\lambda(t)$ and its slope from simulations wherein a cymbal (left) and gong (right) are struck with a sequence of impacts (see more discussion in §7.2). $t\lambda(t)$ increases as the vibrational energy builds up until the system turns chaotic. Afterwards, as the energy dissipates, the increase of $t\lambda(t)$ slows down.

frequency domain, described by

$$\frac{\partial E}{\partial t} = \frac{\partial}{\partial f} \left(\phi f E^2 \frac{\partial E}{\partial f} \right) - \gamma(f, E) + \kappa(f, t), \quad (14)$$

where ϕ is a user-defined scaling factor that controls how quickly the energy diffuses over time, and $\gamma(f, E)$ is the dissipation rate of E due to damping. We introduce $\kappa(f, t)$ as the source term describing how much energy is injected at each time instant. This term does not appear in the model of [Humbert et al. 2016]. We include it to model the energy transferred from modal vibrations, as will be elaborated shortly. Humbert et al. [2016] showed that the stationary solution of (14), without damping and source terms, agrees with the Kolmogorov spectrum emerging in many turbulent phenomena [Düring et al. 2006; Kim et al. 2008]; and the time-varying solution of (14) describes how the wave turbulence cascades energies *dynamically* to reach the stationary Kolmogorov spectrum.

6.2.1 Damping term $\gamma(f, E)$. We derive a simple damping term $\gamma(f, E)$ based on the Rayleigh damping model that we use in shell simulation (6). Given a vibrational mode at frequency f , the Rayleigh damping force is linear with respect to the modal velocity \dot{q}_f , written as $d_f \dot{q}_f$, where d_f is the damping coefficient from the matrix $\tilde{\mathbf{D}}$ in (6). Over an infinitesimal time period Δt , the velocity change due to damping alone is $\Delta \dot{q}_f = -\Delta t d_f \dot{q}_f$, and the corresponding change of vibrational energy at frequency f is

$$\Delta E_f = (\dot{q}_f - \Delta t d_f \dot{q}_f)^2 - \dot{q}_f^2. \quad (15)$$

Taking the derivative of (15) with respect to time and neglecting the second order term (as $\Delta t d_f \ll 1$), we get the damping term,

$$\gamma(f, E) = -2d_f E_f. \quad (16)$$

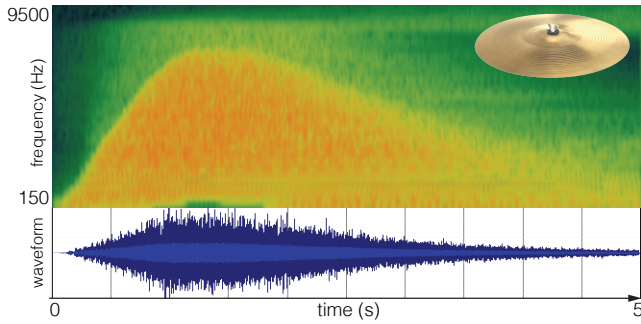


Fig. 11. **Energy diffusion.** On the top is the wave turbulence spectrum resulted from solving (14) when a cymbal is strongly hit. Sound energies are cascaded from low to high frequencies, and then dissipated due to damping. The synthesized turbulence sound waveform is shown below. We note that this spectrum, while generated phenomenologically, agrees qualitatively with the prediction by nonlinear finite difference simulation, as shown in Figure 16 of [Ducceschi et al. 2014].

6.2.2 Source term $\kappa(f, t)$. We model the source term directly in a discrete setting. At each timestep t_k , we would like to use the change of the shell’s modal vibrational energy to drive the diffusion of wave turbulence. Also, the injected wave turbulence energy should be proportional to the local stability indicated by $s(t_k)$ from the previous step: if the system is stable at t_k , then no additional wave turbulence should be generated at t_k . Thus, we propose a source term defined at timestep t_k as

$$\kappa(f, t_k) = \sigma s(t_k)[E_v(f, t_k) - E_v(f, t_{k-1})], \quad (17)$$

where σ is a user-defined scalar to control the strength of the wave turbulence (see video), and $E_v(f, t_k)$ is the power spectrum of the modal vibration at t_k . $E_v(f, t_k)$ is computed using modal vibrational energies of all modes (i.e., \dot{q}_f^2) in our nonlinear simulation, and then interpolated over the entire spectrum. It is possible that $\kappa(f, t_k)$ becomes negative at some point (i.e., a sink), reflecting the dissipation of modal vibrational energies.

6.2.3 Implementation. Numerically solving (14) in frequency-time domain is very fast. In practice, we use Matlab’s `pdedpde` function to solve it. We discretize the frequency domain with 10Hz steps and use a time discretization of $200\Delta t$, where $\Delta t = 1/44100\text{sec}$ is the integration timestep. The computational overhead introduced by this step is negligible, occupying only about 1% to 3% of the total time cost of the pipeline (see Figure 14). An example is illustrated in Figure 11.

6.2.4 Sound synthesis. Lastly, we generate wave turbulence sound signals from the diffused spectrogram $E(f, t)$, which determines the sound amplitude at each timestep but not the phase. It is worth noting that this problem is similar to the classic phase retrieval problem in optical and acoustic imaging [Fienup 1982]. Here we turn back to our nonlinear simulation (6) and use the diffused spectrogram $E(f, t)$ instead of external forces to drive the vibrations. Throughout the simulation, we inject additional modal velocity into every mode to match $E(f, t)$. The velocity to be injected for each mode is computed as the change in velocity in the spectrogram at the corresponding modal frequency. Since vibrational velocities oscillate,

Table 1. **Material parameters** used in all examples.

Examples	Young mod.(Pa)	Poisson ratio	Density (kg/m^3)	Thick. (m)	Damping	
					α	β
CYMBAL	124e9	0.33	8400	0.7e-3	6.25e-9	1
RECYCLING BIN	2.4e9	0.4	916	2.5e-3	300e-9	4
TRASH CAN	190e9	0.30	7850	1e-3	75e-9	0.5
WATER BOTTLE	2.4e9	0.37	1200	1.5e-3	400e-9	0.5
METAL SHEET	70e9	0.35	2700	2.5e-4	400e-9	0.5
GONG	110e9	0.33	8400	3e-3	6.25e-9	0.1
THUNDER SHEET	190e9	0.30	7850	1.6e-3	75e-9	0.5

we inject velocity to a given mode right after its velocity oscillation peak. This way, we weakly constrain the velocities to target the diffused spectrogram but let the nonlinear coupling between modes dictate the overall turbulent behavior.

7 RESULTS

All our results were computed on a 3.6 GHz Quad-core Intel Xeon E5-1620 CPU (2012) with 32GB of memory, using symplectic Euler integration timestepped at audio rates (44100 Hz). Our precomputation step is trivial to parallelize, and we implemented it on an Nvidia GTX 970 GPU using CUDA. Simulation parameters and representative timings are summarized in Table 2. Material parameters used in these examples are listed in Table 1. All examples include modal sound propagation, following the standard far-field acoustic transfer approximation [Cirio et al. 2016; Zheng and James 2010] that uses the fast multipole method to solve the Helmholtz equation for every vibrational mode [Shen and Liu 2006]. We refer the reader to our accompanying video for all our animation and audio results.

7.1 Nonlinear Shells

First, we synthesized the sound produced by thin-shell objects hitting a rigid ground. The shells bounce a few times before coming to rest. Similar to [Chadwick et al. 2009], we treat the thin shells as rigid bodies to compute contact forces using the method [Smith et al. 2012], and timestep the simulation at audio rates. Contact forces were used during sound synthesis to excite the subspace nonlinear model. Forces were always convolved with a raised sine function to approximate their short temporal distribution [Ducceschi and Touzé 2015]. To capture additional damping due to contact with the ground, we increased the damping values whenever the object was under contact [Zheng and James 2011].

We compared the sound generated by our approach to the sound synthesized by a purely linear model [O’Brien et al. 2002], as well as the “reference” sound produced by directly solving (2) – namely, computing nonlinear forces in full space and then projecting them onto the subspace with the same number of modes. All our sounds are perceptually indistinguishable from the reference sounds (see video), but our simulations, including the precomputation, are tens of times faster (see Table 2).

Cymbal. We simulated a bronze cymbal with a 50cm diameter dropped to the ground. One can clearly hear the rich metallic crash of the cymbal due to the large initial impact. The linear sound, on the other hand, is more reminiscent of a bell and thus unrealistic. As the impacts become softer, the nonlinear approach produces sounds



Fig. 12. **Falling shells.** A set of thin shells are dropped on the ground to produce various nonlinear sounds, including (from left to right) a cymbal, a plastic recycling bin, a galvanized steel trash can, a large plastic bottle, and a small piece of metallic sheet. The one-meter ruler serves as a reference to provide a sense of the physical sizes of the shells. Please refer to the video sound track.



Fig. 13. **Turbulent shells.** We simulated three shells, a cymbal (left), a gong (see Figure 1), and a large thunder sheet, to demonstrate their very noticeable wave turbulence sound effects.

with different pitch, amplitude and structure, while the linear sounds get only proportionally softer. Our approach is 77× faster compared to the reference without any qualitative difference.

7.1.1 Recycling bin. A 40×38×55cm polyethylene (plastic) recycling bin hits the ground and bounces around, producing a characteristic wobbly sound when impacts are large. The nonlinear sound is richer and more complex than its linear counterpart. Our simulation is 34× faster than the reference approach.

7.1.2 Trash can. A 75cm high and 53cm wide galvanized steel trash can produces a metallic crash when it hits the ground. When the impacts soften, the trash can vibrates in the linear regime. Our approach is 52× faster than the reference.

7.1.3 Water bottle. A large water cooler polycarbonate (plastic) bottle (53cm high and 31cm wide) bounces and rolls on the ground. The nonlinear low-frequency fluttering is clearly audible, producing a hollow effect, while the linear counterpart remains dry. In this case, our sound synthesis is 20× times faster than the reference.

7.1.4 Metal sheet. A small 18×12cm metallic sheet hits the ground, exciting the easily “bent” vibrational frequencies and giving rise to a sequence of wobbly impacts. The linear sound, on the other hand, sounds like a thick object and fails to convey the characteristics of a thin sheet. Our simulation is 12× faster than the reference.

7.2 Turbulent Shells

We used a cymbal (described above), a gong (112cm in diameter), and a thunder sheet (1.2m×0.5m) to illustrate the synthesis of wave turbulence sound. Specifically, we used a Tamtam gong, well known for its long ringing time and easily triggered turbulent regime. The

thunder sheet is a large and thin sheet of metal, sometimes bent, used in musical or theatrical performances to produce loud and shimmering noises.

The turbulent sounds are produced in two different ways: a single large strike, and a “roll” motion. Both are standard in the repertoire of these instruments. The roll motion is a sequence of soft impacts to build up energy in the shell until the turbulent regime is reached.

Figure 10 shows the plot of the local stability indicator and its gradient (as described in §6.1) for the cymbal and the gong when excited by a roll with strikes of constant magnitude. We can observe how chaos builds up as the roll unfolds. The build up in the case of the gong is slower since impacts are softer: energy is fed into the system at a slower rate compared to the cymbal.

We compared different turbulent sounds generated by varying σ in Eq. (17), the strength of the wave turbulence. We used σ values of 0.3, 1, and 3 for the cymbal, and 1, 3, and 9 for the thunder sheet. We observe that the resulting sounds are not only increasingly louder, but they also deliver a higher pitch. This is because as higher amounts of energy are input into the diffusion process, the diffused energy can cover a larger frequency spectrum, yielding a higher overall pitch. This parameter provides the user the ability to control the strength of wave turbulence enrichment.

We also compared qualitatively with recorded sounds of a cymbal, a gong, and a thunder sheet. These sounds are downloaded online, without knowing their precise geometry and recording setup. Our wave turbulence enrichment renders the synthesized sounds with similar characteristics.

The performance breakdown of these examples is shown in Figure 14. We note that the costs percentages for nonlinear and wave turbulence simulation are the costs for the entire simulation, not the cost per second of sound (unlike Table 2), so it is proportional to the simulated time length, while the precomputation cost is independent from the simulation time length. The turbulence simulation cost includes the cost of solving (14) and final sound synthesis for which we reuse our nonlinear simulation (recall §6.2).

7.3 Experiments

7.3.1 Vibratory regimes. We reproduce the physical experiment of Touzé et al. [2012], which subjected a 40×60cm rectangular plate to a sinusoidal forced excitation. The resulting sound spans all three regimes, as shown in the spectrogram of Figure 4. Our results are qualitatively similar to their experimental results (e.g., as shown in Figure 1 of [Touzé et al. 2012]). When designing a numerical experiment to reproduce the experimentally observed vibrational

Table 2. **Simulation parameters and timings** for all examples. Please note that the synthesis and reference costs are per second of sound, while the precomputation costs are the total time spent on the entire precomputation stage.

Examples	# DoFs	# modes		Freq. range (Hz)		Length (s)	Precomp. cost (s)	Simulation cost (s)	Reference cost (s)	Speedup
		Low freq.	High freq.	Low freq.	High freq.					
CYMBAL	93312	30	100	142-3903	3939-9933	2.5	960	43	3331	77×
RECYCLING BIN	82632	50	240	99-730	746-2272	2.5	1840	119	4012	34×
TRASH CAN	75195	40	120	55-2594	2596-6591	3	1790	57	2990	52×
WATER BOTTLE	43254	30	240	85-949	951-3331	2	930	90	1764	20×
METAL SHEET	29637	50	150	38-1387	1405-5836	2	870	108	1304	12×
GONG	71742	30	260	38-626	646-3364	20	1615	97	3296	34×
THUNDER SHEET	45768	30	160	23-715	781-5132	4	620	83	1851	22×

regimes, Touzé et al. relied on an prohibitively expensive finite difference simulation of the von Kármán equations. While their solution took several days to compute, ours took only 324 seconds for 10 seconds of sound (although likely on different hardware).

We also simulated the same experiment in full space to perform a direct qualitative comparison (see audio in the supplementary material). To mitigate the resulting massive computational burden and prohibitive timestep constraints, we coarsened the mesh down to a few thousand triangles and used an implicit Euler integration scheme. For the reduced simulation, we used the same coarsened mesh and added synthetic turbulence. Results show qualitatively similar sounds, as well as the three vibratory regimes (recall Figure 4). A clear monotone sound due to the sinusoidal excitation can be heard before the turbulent regime kicks in. This sound becomes quickly blurred in the reduced simulation due to synthetic turbulence. In addition, the reduced simulation has an overall higher-frequency pitch, showing that our approach captures the textural characteristics, but not the exact behavior, of the turbulent regime.

Using the same finite difference approach as [Touzé et al. 2012], Ducchieschi et al. [2014] studied the influence of damping in a plate under forced excitations. In particular, they successfully captured the buildup and decay of wave turbulence in the power spectrum. Since our approach includes damping and arbitrary energy injection using our proposed $\gamma(f, E)$ and $\kappa(f, t)$ terms in (14), we could also model and capture this interesting behavior at a minimal computational cost. For example, in our case, the cymbal as shown in Figure 11 shows similar energy diffusion and damping as captured by Figure 16 of their work.

7.3.2 Computation time. We compared our approach with the Harmonic Shells (HS) model [Chadwick et al. 2009] for the cymbal and water bottle scenarios in terms of computation time during both precomputation and runtime synthesis phases. For this comparison, we run HS simulator (shared by the authors) and ours on the same hardware. We note that HS model uses the bending model [Gingold et al. 2004] different from ours. As a result, even with the same material parameters, the number of vibrational modes and their frequencies are different. In general, our model appears to have fewer modes in the low-frequency range.

First, we compared the simulation cost for generating their presented sounds and ours. In this case, the number of modes and the frequency range are different. Using the HS model, precomputation

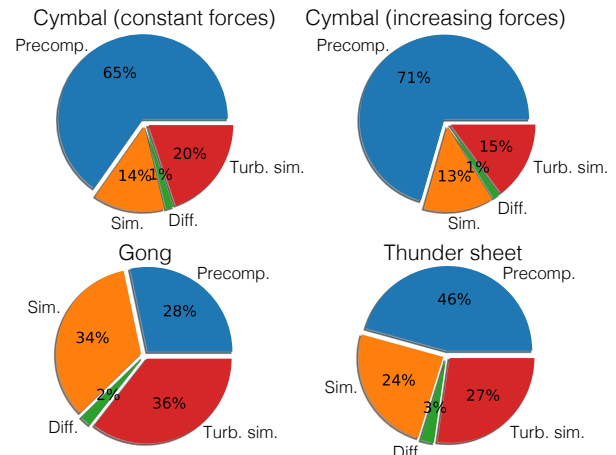


Fig. 14. **Performance breakdown.** The percentages of performance cost are for precomputation (blue), nonlinear simulation (orange), spectrogram diffusion (green), and wave turbulence propagation and sound synthesis (red).

time took 156 minutes for the cymbal and 51 minutes for the water bottle, while our model was 10× and 3.3× faster respectively. At runtime, we are 51× faster than HS for the cymbal and 8.7× for the water bottle, mainly thanks to our runtime vector pruning.

Next, we *artificially* modified our bending stiffness in order to match the number of modes resulted from HS (500 for the cymbal example) as well as the frequency range (61~9940Hz). In this case, we are still 2.6× faster at the precomputation stage and 20× faster at synthesis. We include the resulting sounds as supplementary material, but remind the reader that for the latter comparison, our degraded sound quality is due to the artificial change in stiffness.

7.3.3 Tradeoff between quality and cost. There are two ways to gain further speedups, although at the cost of sound quality. Reducing the number of low frequency modes has a direct impact on precomputation times, because precomputation has a cubic complexity on the number of low frequency modes. We run our different scenarios using different numbers of low frequency modes while keeping the total number of modes unchanged. For the cymbal, precomputation took 715s for 20 low-frequency modes, 489s for 10 low-frequency modes, and 469s for 5 low-frequency modes. For the trash can, we measured 1102s for 30 low-frequency modes, 594s for 20 low-frequency modes and 385s for 10-low frequency modes. As

the number of low-frequency modes is reduced, the sounds gracefully diverge from the reference, but remain plausible.

Another potential speedup lies in the threshold used for runtime vector pruning. For the four falling shell examples, Figure 8 shows the percentage of polynomial vectors that are active at a given time, using a pruning threshold of 10^{-3} . The speedups over the complete polynomial evaluation are 5.6×, 13.2×, 13.7×, and 13.5×, respectively. In addition, we run our synthesis step by increasing the pruning threshold by one order of magnitude each time (10^{-2} and 10^{-1}). The synthesis of the falling cymbal sound took 32s and 18s respectively, while the trash can took 35s and 20s respectively. Meanwhile, the quality loss is evident as well: artifacts and glitches start to appear in the sound as the pruning threshold becomes too large.

8 CONCLUSION

We have presented a physically based reduced simulation method to synthesize the sound of nonlinear and turbulent thin shells. We proposed a new frequency-splitting scheme in reduced space to lower precomputation and runtime cost. We timestep low-frequency modes in a fully nonlinear way, and drive high-frequency modes through a time varying linearization of the subspace dynamics. We used a polynomial form for runtime evaluation of thin-shell internal forces directly in the subspace, and further accelerated the force evaluation by dynamically pruning polynomial terms. Our reduced simulation not only produces fast nonlinear sounds, but is also able to capture chaos in low-frequency modes. We use this information to drive a phenomenological model of wave turbulence to enrich sounds with turbulent details at minimal cost.

Our approach and implementation are not without limitations, and there are many opportunities for future work. By constructing our reduced basis through linear modal analysis, we are limited to moderately small deformations. While nonlinear effects are dramatic under this construction, modal locking may emerge if excitations are too large. For the very same reason, compliant materials remain an open challenge even under small excitations. It would be interesting to investigate the use of a different subspace basis, perhaps using Principal Component Analysis on a subset of deformations, or exploring the use of modal derivatives to enrich the linear basis.

In this work, we exploit the mathematical structure in the hinge-based model to reduce the computational complexity, but the frequency-splitting method is independent from the choice of bending model. It remains a future work to investigate how to accelerate other bending models using the frequency splitting method.

Large thin shells often exhibit a low frequency spectrum, and a large number of modes is required to cover even a small subset of the audible frequency range. Our approach significantly reduces the complexity of precomputation and synthesis steps by splitting the frequency range, but a model with several thousands of modes would remain very expensive to compute. Previous approaches [Ducceschi and Touzé 2015] have employed ad-hoc rules to limit the number of coupling terms among modes in thin plates only. It would be interesting to explore similar ways to reduce the precomputation burden when the modal basis is too large.

Finally, our modeling of turbulent phenomena relies on a purely phenomenological model of wave turbulence. While this is attractive

for the Computer Graphics field where computation speed is often a critical factor — and similar strategies have been adopted (e.g., in [Kim et al. 2008; Perlin 1985]), we cannot claim that our turbulent sounds are accurate from a mechanical point of view. Finding more principled ways to address challenging chaotic dynamics at tractable computation times remains an open problem.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their feedback and Jeffrey N. Chadwick for early software assistance. Ante Qu contributed to this project when he was a summer intern at Columbia University. This project was partially funded by the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 706708, PhySound. This material is based upon work supported in part by the National Science Foundation under Grant Nos. CAREER-1453101, 1717268, 1409286, 1717178, and DGE-1656518. We are grateful for generous support from SoftBank Group, Adobe, Autodesk, Pixar, and SideFX. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or others.

REFERENCES

- Steven S. An, Theodore Kim, and Doug L. James. 2008. Optimizing Cubature for Efficient Integration of Subspace Deformations. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2008) (SIGGRAPH Asia '08)*. ACM, New York, NY, USA, 165:1–165:10. <https://doi.org/10.1145/1457515.1409118>
- David Baraff and Andrew Witkin. 1998. Large steps in cloth simulation. In *Proceedings of SIGGRAPH (SIGGRAPH '98)*. ACM, New York, NY, USA, 43–54. <https://doi.org/10.1145/280814.280821>
- Jernej Barbic and Doug L. James. 2005. Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005)* 24, 3 (2005), 982–990.
- K. J. Bathe. 2007. *Finite Element Procedures*. Prentice-Hall, Boston, Mass.
- Miklos Bergou, Max Wardetzky, David Harmon, Denis Zorin, and Eitan Grinspun. 2006. A Quadratic Bending Model for Inextensible Surfaces. In *Proceedings of the Eurographics Symposium on Geometry Processing (SGP '06)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 227–230. <http://dl.acm.org/citation.cfm?id=1281957.1281987>
- Stefan Bilbao. 2008. A family of conservative finite difference schemes for the dynamical von Karman plate equations. *Numerical Methods for Partial Differential Equations* 24, 1 (Jan. 2008), 193–216. <https://doi.org/10.1002/num.20260>
- S. Bilbao. 2010. Percussion Synthesis Based on Models of Nonlinear Shell Vibration. *IEEE Transactions on Audio, Speech, and Language Processing* 18, 4 (May 2010), 872–880. <https://doi.org/10.1109/TASL.2009.2029710>
- R. Bridson, S. Marino, and R. Fedkiw. 2003. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA '03)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 28–36. <http://dl.acm.org/citation.cfm?id=846276.846281>
- Olivier Cadot, M Ducceschi, T Humbert, Benjamin Miquel, N Mordant, C Josserand, and Cyril Touzé. 2016. Wave turbulence in vibrating plates. In *Handbook of applications of chaos theory*, Charilaos Skiadas Christos H. Skiadas (Ed.). Chapman and Hall/CRC.
- Jeffrey N. Chadwick, Steven S. An, and Doug L. James. 2009. Harmonic shells: a practical nonlinear sound model for near-rigid thin shells. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009)* 28, 5 (2009).
- Antoine Chaigne, Cyril Touzé, and Olivier Thomas. 2005. Nonlinear vibrations and chaos in gongs and cymbals. *Acoustical Science and Technology* 26, 5 (2005), 403–409.
- Dominique Chapelle and Klaus-Jürgen Bathe. 2010. *The Finite Element Analysis of Shells - Fundamentals* (2nd ed. 2011 edition ed.). Springer, Berlin ; New York.
- Chuen-Yuan Chia. 1980. *Nonlinear analysis of plates*. McGraw-Hill International Book Company.
- Gabriel Cirio, Dingzeyu Li, Eitan Grinspun, Miguel A. Otaduy, and Changxi Zheng. 2016. Crumpling Sound Synthesis. *ACM Trans. Graph.* 35, 6 (Nov. 2016), 181:1–181:11.
- Lothar Cremer and Manfred Heckl. 2013. *Structure-borne sound: structural vibrations and sound radiation at audio frequencies*. Springer Science & Business Media.
- Michele Ducceschi, Olivier Cadot, Cyril Touzé, and Stefan Bilbao. 2014. Dynamics of the wave turbulence spectrum in vibrating plates: A numerical investigation using

- a conservative finite difference scheme. *Physica D: Nonlinear Phenomena* 280-281, Supplement C (July 2014), 73–85.
- M. Ducceschi and C. Touzé. 2015. Modal approach for nonlinear vibrations of damped impacted plates: Application to sound synthesis of gongs and cymbals. *Journal of Sound and Vibration* 344, Supplement C (May 2015), 313–331. <https://doi.org/10.1016/j.jsv.2015.01.029>
- S. Dyachenko, A. C. Newell, A. Pushkarev, and V. E. Zakharov. 1992. Optical turbulence: weak turbulence, condensates and collapsing filaments in the nonlinear Schrödinger equation. *Physica D: Nonlinear Phenomena* 57, 1 (June 1992), 96–160. [https://doi.org/10.1016/0167-2789\(92\)90090-A](https://doi.org/10.1016/0167-2789(92)90090-A)
- Gustavo Düring, Christophe Josserand, and Sergio Rica. 2006. Weak turbulence for a vibrating plate: can one hear a Kolmogorov spectrum? *Physical Review Letters* 97, 2 (July 2006), 025503. <https://doi.org/10.1103/PhysRevLett.97.025503>
- Éric Falcon, Claude Laroche, and Stéphane Fauve. 2007. Observation of Gravity-Capillary Wave Turbulence. *Physical Review Letters* 98, 9 (March 2007), 094503. <https://doi.org/10.1103/PhysRevLett.98.094503>
- G. E. Falkovich and A. V. Shafarenko. 1991. Nonstationary wave turbulence. *Journal of Nonlinear Science* 1, 4 (Dec. 1991), 457–480. <https://doi.org/10.1007/BF02429849>
- J. R. Fienup. 1982. Phase retrieval algorithms: a comparison. *Applied Optics* 21, 15 (Aug. 1982), 2758–2769. <https://doi.org/10.1364/AO.21.002758>
- Akash Garg, Eitan Grinspun, Max Wardetzky, and Denis Zorin. 2007. Cubic Shells. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 91–98.
- Yotam Gingold, Adrian Secord, Jefferson Y. Han, Eitan Grinspun, and Denis Zorin. 2004. *A Discrete Model for Inelastic Deformation of Thin Shells*. Technical Report. Courant Institute of Mathematical Sciences, New York University.
- Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete Shells. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 62–67.
- T. Humbert, C. Josserand, C. Touzé, and O. Cadot. 2016. Phenomenological model for predicting stationary and non-stationary spectra of wave turbulence in vibrating plates. *Physica D: Nonlinear Phenomena* 316, Supplement C (Feb. 2016), 34–42.
- Christophe Josserand, Yves Pomeau, and Sergio Rica. 2006. Self-similar Singularities in the Kinetics of Condensation. *Journal of Low Temperature Physics* 145, 1-4 (Nov. 2006), 231–265. <https://doi.org/10.1007/s10909-006-9232-6>
- Theodore Kim, Nils Thürey, Doug James, and Markus Gross. 2008. Wavelet Turbulence for Fluid Simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)* (2008), 50:1–50:6.
- P. Krysl, S. Lall, and J. E. Marsden. 2001. Dimensional model reduction in non-linear finite element dynamics of solids and structures. *Internat. J. Numer. Methods Engrg.* 51, 4 (June 2001), 479–504. <https://doi.org/10.1002/nme.167>
- C. E. Leith. 1967. Diffusion Approximation to Inertial Energy Transfer in Isotropic Turbulence. *The Physics of Fluids* 10, 7 (July 1967), 1409–1416. <https://doi.org/10.1063/1.1762300>
- F. Moussaoui and R. Benamar. 2002. Nonlinear vibrations of shell type structures: A review with bibliography. *Journal of Sound and Vibration* 255, 1 (Aug. 2002), 161–184. <https://doi.org/10.1006/jsvi.2001.4146>
- Sam L. Musher, Alexander M. Rubenchik, and Vladimir E. Zakharov. 1995. Weak Langmuir turbulence. *Physics Reports* 252, 4 (Feb. 1995), 177–274. [https://doi.org/10.1016/0370-1573\(94\)00071-A](https://doi.org/10.1016/0370-1573(94)00071-A)
- Rahul Narain, Tobias Pfaff, and James F. O'Brien. 2013. Folding and Crumpling Adaptive Sheets. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2013)* 32, 4 (2013), 51:1–51:8. <https://doi.org/10.1145/2461912.2462010>
- Ali H. Nayfeh and Dean T. Mook. 1995. *Nonlinear Oscillations*. Wiley-VCH, New York.
- Alan C. Newell, Sergey Nazarenko, and Laura Biven. 2001. Wave turbulence and intermittency. *Physica D: Nonlinear Phenomena* 152-153 (May 2001), 520–550. [https://doi.org/10.1016/S0167-2789\(01\)00192-0](https://doi.org/10.1016/S0167-2789(01)00192-0)
- James F. O'Brien, Perry R. Cook, and Georg Essl. 2001. Synthesizing sounds from physically based motion. In *Proceedings of SIGGRAPH (SIGGRAPH '01)*. ACM, New York, NY, USA, 529–536. <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 *The ACM SIGGRAPH 2002 Symposium on Computer Animation*. ACM Press, 175–181. <https://doi.org/10.1145/545261.545290>
- Ken Perlin. 1985. An Image Synthesizer. In *Proceedings of SIGGRAPH (SIGGRAPH '85)*. ACM, New York, NY, USA, 287–296. <https://doi.org/10.1145/325334.325247>
- John R. Pierce and Scott A. Van Duyne. 1997. A passive nonlinear digital filter design which facilitates physics-based sound synthesis of highly nonlinear musical instruments. *The Journal of the Acoustical Society of America* 101, 2 (Feb. 1997), 1120–1126. <https://doi.org/10.1121/1.418017>
- Nick Rasmussen, Duc Quang Nguyen, Willi Geiger, and Ronald Fedkiw. 2003. Smoke Simulation for Large Scale Phenomena. *ACM Trans. Graph.* 22, 3 (July 2003), 703–707. <https://doi.org/10.1145/882262.882335>
- Sandra Rugonyi and Klaus-Jürgen Bathé. 2003. An evaluation of the Lyapunov characteristic exponent of chaotic continuous systems. *Internat. J. Numer. Methods Engrg.* 56, 1 (2003), 145–163.

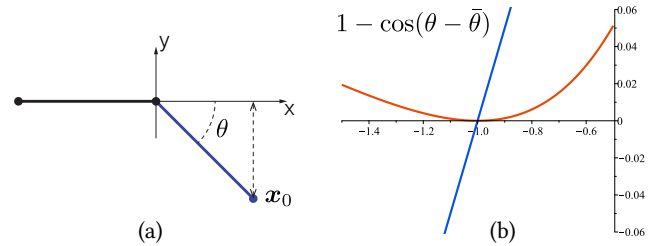


Fig. 15. Occurrence of ghost bending forces.

- Ahmed A Shabana. 2012. *Theory of vibration: Volume II: discrete and continuous systems*. Springer Science & Business Media.
- L. Shen and Y. J. Liu. 2006. An adaptive fast multipole boundary element method for three-dimensional acoustic wave problems based on the Burton–Miller formulation. *Computational Mechanics* 40, 3 (Oct. 2006), 461–472. <https://doi.org/10.1007/s00466-006-0121-2>
- Breannan Smith, Danny M. Kaufman, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. 2012. Reflections on Simultaneous Impact. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012)* 31, 4 (July 2012), 106:1–106:12. <https://doi.org/10.1145/2185520.2185602>
- Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically deformable models. In *Proceedings of SIGGRAPH*. ACM, 205–214. <https://doi.org/10.1145/37401.37427>
- C. Touzé, S. Bilbao, and O. Cadot. 2012. Transition scenario to turbulence in thin vibrating plates. *Journal of Sound and Vibration* 331, 2 (Jan. 2012), 412–433.
- W. F. Vinen. 2000. Classical character of turbulence in a quantum liquid. *Physical Review B* 61, 2 (Jan. 2000), 1410–1420. <https://doi.org/10.1103/PhysRevB.61.1410>
- Pascal Volino, Nadia Magnenat-Thalmann, and Francois Faure. 2009. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2009)* 28, 4 (2009), 105:1–105:16.
- Stephen Wiggins. 2003. *Introduction to Applied Nonlinear Dynamical Systems and Chaos* (2nd edition ed.). Springer, New York.
- Alan Wolf, Jack B. Swift, Harry L. Swinney, and John A. Vastano. 1985. Determining Lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena* 16, 3 (July 1985), 285–317. [https://doi.org/10.1016/0167-2789\(85\)90011-9](https://doi.org/10.1016/0167-2789(85)90011-9)
- V. E. Zakharov and N. N. Filonenko. 1967. Energy Spectrum for Stochastic Oscillations of the Surface of a Liquid. *Soviet Physics Doklady* 11 (April 1967), 881. <http://adsabs.harvard.edu/abs/1967SPD...11..881Z>
- Vladimir E. Zakharov, Victor S. L'vov, and Gregory Falkovich. 1992. *Kolmogorov Spectra of Turbulence I: Wave Turbulence*. Springer.
- Changxi Zheng and Doug L. James. 2010. Rigid-body Fracture Sound with Precomputed Soundbanks. *ACM Trans. Graph.* 29, 4 (July 2010).
- Changxi Zheng and Doug L. James. 2011. Toward High-Quality Modal Contact Sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)* 30, 4 (Aug. 2011).

A OCCURRENCE OF GHOST BENDING FORCES

In their bending model, Garg et al. [2007] approximate the bending energy (8) under the isometry assumption. First, Equation (8) is written as

$$E_b(\mathbf{u})_i = \frac{3|\bar{\mathbf{e}}_i|^2}{\bar{A}_i} (1 - \cos \theta_i \cos \bar{\theta}_i - \sin \theta_i \sin \bar{\theta}_i). \quad (18)$$

By applying the isometry assumption, $\cos \theta_i$ and $\sin \theta_i$ are approximated using cubic polynomials with respect to vertex positions. However, this approximation breaks a key property of the energy function: namely, $E_b(\mathbf{u})_i$ must be semi-positive definite.

To illustrate why this property cannot be satisfied under isometry assumption, consider a 2D toy example shown in Figure 15-a, where a 2D hinge consists of two edges, one horizontal along X-axis and another pointing toward the 45° direction (i.e., $\bar{\theta} = 45^\circ$). The hinge is centered at origin. Suppose the rest position of \mathbf{x}_0 is at (1, -1), and its deformed position is denoted by \mathbf{x} . In this case, we have

$$\cos \theta = \frac{\mathbf{x} \cdot (1 \ 0)}{|\mathbf{x}|_2} \quad \text{and} \quad \sin \theta = \frac{\mathbf{x} \times (1 \ 0)}{|\mathbf{x}|_2}. \quad (19)$$

The red curve in Figure 15-b shows the bending energy (up to a constant scale) with respect to the Y-coordinate of \mathbf{x} (i.e., $E_b(y) = 1 - \cos \theta(y) \cos \bar{\theta} - \sin \theta(y) \sin \bar{\theta}$). As expected, $E_b(y)$ is always non-negative, reaching zero when $\mathbf{x} = \mathbf{x}_0$ (i.e., $y = -1$).

Under the isometry assumption, the edge length is assumed constant. In other words, $\|\mathbf{x}\|_2$ in (19) is replaced by a constant $\|\mathbf{x}_0\|_2$. Then, as the Y-coordinate of \mathbf{x} changes, the estimated $\cos \theta$ and $\sin \theta$ are $\frac{\|\mathbf{x}\|_2}{\|\mathbf{x}_0\|_2}$ times of their true values. Thus, the estimated $E_b(y)$ value might become negative, as shown by the blue curve in Figure 15-b. At the rest position $\mathbf{x} = \mathbf{x}_0$, the isometry assumption introduces no error, so the energy function remains zero. However, because a small perturbation of \mathbf{x} from \mathbf{x}_0 can lead to a negative energy value, the bending force, which is the negated derivative of $E_b(y)$ with respect to the position, becomes non-zero. That non-zero force is the unwanted ghost bending force.

B COMPUTATION OF REDUCED FORCE POLYNOMIAL

Here we describe how to compute the reduced force polynomial, starting from the full space force polynomial formulation. We focus on the derivation of cubic polynomial terms. Quadratic and linear terms follow the same procedure.

The cubic polynomial force applied on the degree of freedom d can be written in full space as:

$$f_d(\mathbf{u}) = \sum_{r=1}^n \sum_{s=1}^n \sum_{t=1}^n a_d^{r,s,t} u_r u_s u_t, \quad (20)$$

where \mathbf{u} is the displacement vector of size n and $a_d^{r,s,t}$ is the polynomial coefficient. Without loss of generality, let's consider the cubic term $F_d^{r,s,t} = a_d^{r,s,t} u_r u_s u_t$, applied to degree of freedom d and involving displacements r , s and t .

For convenience, let's call V_i row i of the modal matrix U . We expand $F_d^{r,s,t}$ and reformulate it in terms of reduced displacements \mathbf{q} of size m :

$$\begin{aligned} F_d^{r,s,t} &= a_d^{r,s,t} u_r u_s u_t \\ &= a_d^{r,s,t} (\mathbf{V}_r^T \mathbf{q}) (\mathbf{V}_s^T \mathbf{q}) (\mathbf{V}_t^T \mathbf{q}) \\ &= a_d^{r,s,t} \left(\sum_{i=1}^m V_r(i) q_i \right) \left(\sum_{j=1}^m V_s(j) q_j \right) \left(\sum_{k=1}^m V_t(k) q_k \right) \\ &= a_d^{r,s,t} \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m V_r(i) q_i V_s(j) q_j V_t(k) q_k. \end{aligned} \quad (21)$$

To compute the contribution of $F_d^{r,s,t}$ to $\tilde{\mathbf{f}}_{\text{int}}$, we need to premultiply $F_d^{r,s,t}$ by V_d , since $\tilde{\mathbf{f}}_{\text{int}}(\mathbf{q}) = \mathbf{U}^T \mathbf{f}_{\text{int}}(\mathbf{U}\mathbf{q})$. Replacing $F_d^{r,s,t}$ by its expanded version we get:

$$V_d F_d^{r,s,t} = V_d a_d^{r,s,t} \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m V_r(i) V_s(j) V_t(k) q_i q_j q_k. \quad (22)$$

The highlighted term is constant, can be precomputed, and is cumulated in the coefficient vector C_{ijk} of Eq. (6).

C QUARTIC TERMS IN ISOMETRIC BENDING FORCES

We recall that all quartic terms in the expression of the isometric bending force have the form $\mathbf{e}_a \mathbf{e}_b^T (\mathbf{e}_c \times \mathbf{e}_d)$, where \mathbf{e}_i is the vector

of a triangle edge. Specifically, $c = 4$ and $d = 2$, where \mathbf{e}_4 and \mathbf{e}_2 are the two flap edges belonging to one triangle of the hinge.

Since we are interested in the displacement degrees of freedom, we can rewrite all edges \mathbf{e}_i as the sum of the (constant) rest state edge $\bar{\mathbf{e}}_i$ and its time varying displacement \mathbf{e}_i^* . Since all operations in the term are distributive, we can isolate the quartic terms in the displacements to the term $\mathbf{e}_a^* (\mathbf{e}_b^*)^T (\mathbf{e}_c^* \times \mathbf{e}_d^*)$. When the hinge does not undergo rigid rotations, the term $\mathbf{e}_4^* \times \mathbf{e}_2^*$ vanishes since $\mathbf{e}_4^* \times \mathbf{e}_2^* = (\mathbf{u}_2 - \mathbf{u}_0) \times (\mathbf{u}_2 - \mathbf{u}_1)$ and $\mathbf{u}_0 = \mathbf{u}_1 = \mathbf{0}$.