Toward High-Quality Modal Contact Sound

Changxi Zheng   Doug L. James

Cornell University

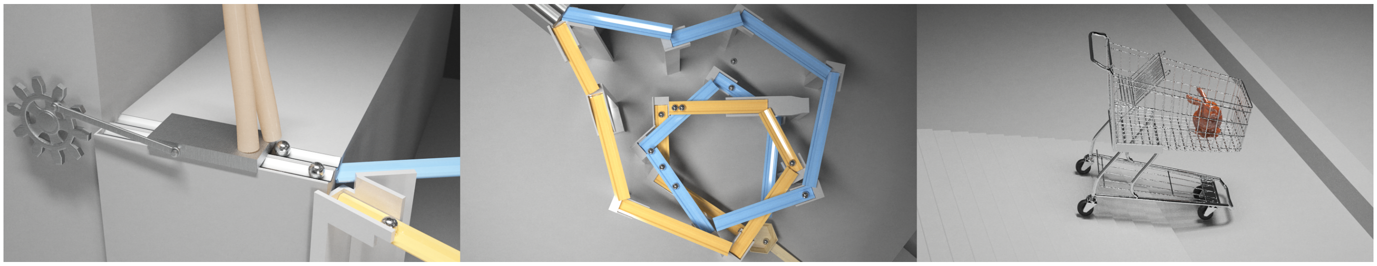


Figure 1: A Rube-Goldberg contraption *that demonstrates many challenging multibody contact sounds. A noisy block feeder (Left) with flexible tubes ejects marbles into a double helix of plastic chutes (Middle), which causes a cup to fill up, lifting a lever that drops a bunny into a runaway shopping cart (Right) producing familiar clattering and clanging sounds due to deformable micro-collisions. Our approach can accurately resolve modal vibrations and contact sounds using an asynchronous, adaptive, frictional contact solver.*

Abstract

Contact sound models based on linear modal analysis are commonly used with rigid body dynamics. Unfortunately, treating vibrating objects as "rigid" during collision and contact processing fundamentally limits the range of sounds that can be computed, and contact solvers for rigid body animation can be ill-suited for modal contact sound synthesis, producing various sound artifacts. In this paper, we resolve modal vibrations in both collision and frictional contact processing stages, thereby enabling non-rigid sound phenomena such as micro-collisions, vibrational energy exchange, and chattering. We propose a frictional multibody contact formulation and modified Staggered Projections solver which is well-suited to sound rendering and avoids noise artifacts associated with spatial and temporal contact-force fluctuations which plague prior methods. To enable practical animation and sound synthesis of numerous bodies with many coupled modes, we propose a novel asynchronous integrator with model-level adaptivity built into the frictional contact solver. Vibrational contact damping is modeled to approximate contact-dependent sound dissipation. Results are provided that demonstrate high-quality contact resolution with sound.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation; H.5.5 [Information Systems]: Information Interfaces and Presentation—Sound and Music Computing

Keywords: Sound synthesis; contact sounds; modal analysis; asynchronous integration; frictional contact

1 Introduction

Sound models based on linear modal vibrations are widely used to efficiently synthesize plausible contact sounds for so-called rigid bodies in computer animation and interactive virtual environments. Unfortunately, there still remain a number of significant contact-related deficiencies that limit the realism of modal contact sounds in practice. To begin with, for speed and simplicity, modal sound models are usually just excited by using contact force impulses from rigid body contact solvers. In reality, there is no such thing as a "rigid" object, and the same small vibrations that produce sound also play an important role in producing rich contact events: micro-collisions, chattering, squeaking, coupled vibrations, contact damping, etc. Ignoring contact-level vibrations is the source of many sound-related deficiencies, as these small vibrations can be visually inconsequential but aurally significant. For example, pounding on a seemingly "rigid" dinner table can shake dishes—and may also upset your friends (see Figure 2). Frictional contact and deformation coupling is also important for sound; for example, slip-stick phenomena is responsible for many familiar squeaking and scraping sounds, e.g., fingernails scraping on a chalkboard. Resolving these vibrational contact effects is challenging due to the need to resolve deformable collisions and contact at high temporal rates.

Even in seemingly rigid scenarios, such as an object resting on a plane, current contact solver implementations can generate temporally incoherent contact impulses which lead to sound artifacts, such as resting objects that strangely humm or buzz when integrated at near-audio rates. These artifacts are a consequence of the fundamental non-uniqueness of rigid body contact forces (e.g., static indeterminacy) which can lead to point-like and nonphysical contact force (traction) distributions. Additionally, rigid-body contact impulses can exhibit nonphysical temporal fluctuations, which lead to noise-related sound artifacts (especially with iterative contact solution techniques) that must be dissipated artificially.

Moreover, the sound of a resting object should also depend on its contact state, and how contacts oppose surface vibrations. For example, a coffee mug exhibits distinctive vibrational damping when placed in different orientations on surfaces (see Figure 4). This contact damping phenomena involves complex vibrational and contact coupling effects, and is ignored in current sound models or handled in *ad hoc* ways, e.g., "increase damping when in contact."

In this paper, we propose the first approach to address all of these concerns and enable richer contact sounds (see Figure 3). We adopt a flexible multibody dynamics formulation, wherein each seemingly rigid object is allowed to deform with linear modal vibrations.

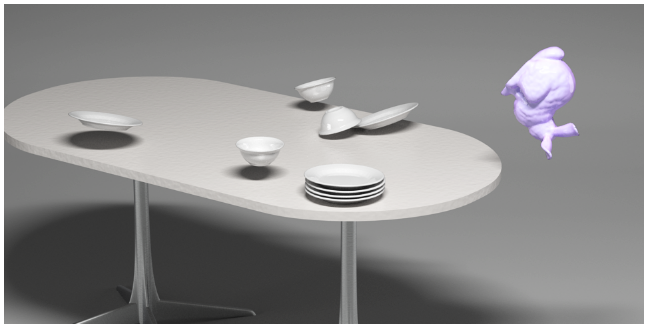


Figure 2: Vibrational coupling makes a racket! *A rigid bunny dropped onto a table causes the table to deform rapidly, which in turn causes the resting dishes to receive contact impulses and go flying—with noticeable sound. In contrast, a traditional rigid body simulation (not shown) bounces the bunny off the perfectly rigid table without disturbing any dishes or causing much sound.*

Deformable collision processing at near audio rates is used to generate frictional contact problems to resolve perceptually important coupling and micro-collisions (see Figures 1 & 2). We use a modified Staggered Projections algorithm to solve the frictional contact problems [Kaufman et al. 2008], with modifications to avoid solver noise due to spatial and temporal incoherence in both the generation of contact points and the computation of contact forces. Vibrational contact damping phenomena are approximated using a time-varying lumped contact damping matrix to attenuate vibrations during sound synthesis.

Since multibody frictional contact solves with thousands of coupled modes at near-audio rates are computationally intractable for sound synthesis, we propose a novel mode-adaptive asynchronous integrator which is capable of identifying and integrating low-frequency vibration modes in the contact solver (see Figure 8). Methods for adding and removing modes from the contact-level simulation are described, as well as noise-free methods for simulating all audible modes during subsequent sound synthesis. Our implementation is able to handle dozens of bodies with thousands of audible modes in a practical manner for high-quality offline sound synthesis.

2 Related Work

Contact Sound Synthesis: Rendering sounds synchronized with animation has a long history in graphics [Takala and Hahn 1992; Gaver 1993]. Modal sound synthesis methods generate plausible rigid body contact sounds efficiently, and were popularized by van den Doel and Pai [1996] who proposed using analytical models of linear modal vibration to produce point-contact sounds in interactive virtual environments. Modal sound models can be conveniently estimated from recorded sounds and measurements [Pai et al. 2001], or estimated using numerical techniques for linear modal analysis [Shabana 1990; O'Brien et al. 2002]. For animation, simply running dynamics simulations at graphics rates can greatly limit the range of sounds achievable. Van den Doel et al. [2001] realized the perceptual importance of resolving micro-collision forces sampled at (near) audio rates, and explicitly distinguished them from the so-called "dynamics force" sampled at graphics rate, but the range of resolvable contact phenomena was limited by a point-like contact model. O'Brien et al. [2002] used tetrahedral finite element simulations to synthesize modal sounds for rigid bodies excited by contact forces from graphics simulations; however the rigid model cannot capture deformable contact events.

Other developments in modal contact sound synthesis include speed-accuracy trade-offs for interactive sound [Doel et al. 2004; Raghuvanshi and Lin 2006; Bonneel et al. 2008; Ren et al. 2010], acoustic transfer models to improve realism of spatialized sounds [James et al. 2006; Chadwick et al. 2009], and techniques for synthesizing brittle fracture sounds [Zheng and James 2010]. Chadwick et al. [2009] considered subspace integration techniques to resolve nonlinear mode-coupling effects in thin shells, but treated objects as rigid for collision processing. In contrast, we address fundamental deficiencies in contact resolution methods which currently limit our ability to animate multibody systems with sounds.

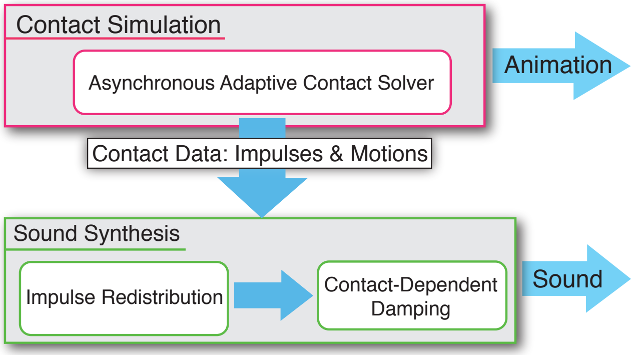

Contact Simulation
Asynchronous Adaptive Contact Solver
Animation
Contact Data: Impulses & Motions
Sound Synthesis
Impulse Redistribution
Contact-Dependent Damping
Sound

Figure 3: Overview: *Multibody simulation is performed using an asynchronous adaptive contact solver (Top). This simulation yields animation data as well as impulses and object motion data which are fed to the sound synthesis algorithm running in a separated pass (Bottom). When synthesizing modal sound, input impulses are redistributed to enhance temporal coherence and eliminate noise, and contact-dependent damping is applied.*

Non-modal sound synthesis for computer animation was explored in [O'Brien et al. 2001] using an explicitly time-stepped large-deformation finite element model to simulate the surface response of an object due to external forces, potentially being able to resolve more detailed contact events. However, explicitly time-stepped detailed FEM models at audio rates is expensive, and the stiff materials, e.g., steel, can result in onerous timestep restrictions. In contrast, we resolve small modal deformations in multibody contact, but only simulate a subset of vibration modes by using an efficient adaptive, asynchronous contact algorithm.

Contact-dependent damping for sound synthesis is not well addressed. Previous works often used a static parametric model for vibration damping [O'Brien et al. 2002; Zheng and James 2010], or apply some additional *ad hoc* damping when objects are in contact, e.g., with the ground [Chadwick et al. 2009]. Zheng and James [2010] introduced additional contact damping for objects at rest to mask noise introduced by the iterative contact solver. In contrast, we resolve frictional contact, and propose a viscous model to approximate spatially and temporally dependent contact damping.

Rigid and Deformable Simulation: Resolving frictional contact with rigid bodies has long been recognized as a challenging problem in graphics and engineering, in part due to the inherent difficulty of contact discontinuities, nonlinear Coulomb friction, and solution non-uniqueness [Brogliato 1999; Stewart 2000]. For flexible multibody simulations, e.g., where modal vibrations are included, additional difficulties arise, especially for high-frequency vibrations [Wasfy and Noor 2003]. Early methods for visual simulation of rigid and deformable objects explored penalty-based methods [Hahn 1988; Moore and Wilhelms 1988]. While penalty contact methods can generate low-noise contact impulses for rigid body dynamics ([James et al. 2006]), these methods can suffer from sta-

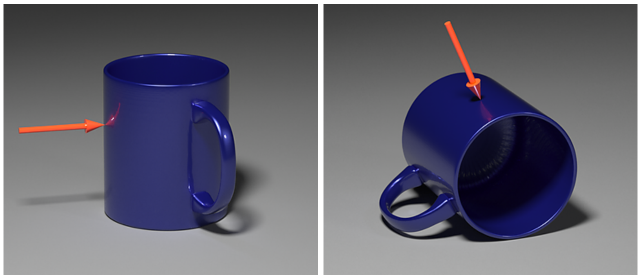


Figure 4: Contact-dependent vibrational damping *is readily apparent by tapping a coffee mug at the same location while in different contact configurations. Our contact sounds differ because contact damping opposes modal vibrations differently.*

bility issues that necessitate extremely small timesteps, and challenges exist for practical modeling of frictional contact and modal chattering. More accurate models of rigid body contact resolution have traditionally motivated Linear Complementarity Programming (LCP) formulations. Unfortunately solving LCP contact problems can be hard computationally, and for certain problems solutions may not exist or be unique when friction is considered [Lötstedt 1982]. Developments in graphics tackled rigid body contact using acceleration-level LCP formulations [Baraff 1990; Baraff 1991], however solution existence could not always be guaranteed [Baraff 1993]. Velocity-level LCP formulations were later proposed which can always find a solution [Stewart and Trinkle 1996; Anitescu and Potra 1997]. More recently the iterative velocity-level LCP methods have seen greater use in practice due to increased performance in exchange for degraded accuracy and stability [Guendelman et al. 2003; Erleben 2007]. Alternate quadratic program (QP) formulations of the LCP are also possible [Moreau 1966] (but are often non-convex and not easy to solve), and have been exploited by some works in graphics [Milenkovic and Schmidl 2001; Kaufman et al. 2008]. Unfortunately, rigid body and flexible multibody contact methods devised for computer graphics and engineering were never designed for physically based sound synthesis in mind, and their straightforward application leads to problems with noise, accuracy, and efficiency (discussed further in §3). While low-noise contact simulation has been demonstrated in certain scenarios, e.g., for continuous single-point contact simulation involving smooth surfaces [van den Doel et al. 2001; Kry and Pai 2003], we address the need for sound-aware contact resolution methods for practical sound synthesis involving rigid and flexible multibody systems. Our method leverages prior work on the Staggered Projections (SP) method [Kaufman et al. 2008]; however, simply running SP at audio rates—despite being prohibitively expensive—is unsatisfactory for sound synthesis since the method suffers from accuracy issues which introduce noise in the final sound. Our proposed method uses different contact generation strategies, and applies a contact filtering algorithm to achieve temporally coherent and spatially well-distributed contact impulses with cleaner sounds.

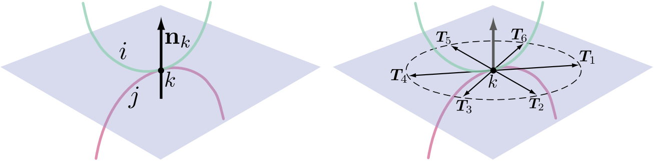
Asychrony and Adaptivity: Only a few methods for simulating rigid and/or deformable objects provide temporal adaptivity or asynchronous integration. Mirtich [2000] introduced Jefferson's timewarp algorithm [Jefferson 1985] into graphics, and enabled asynchronous rigid body simulation whose timestep size is adapted based on accuracy requirements. In comparison, our method can asynchronously evolve both rigid and deformable objects, and can adaptively switch between rigid and deformable simulations. Mode-culling techniques avoid synthesizing sounds for inaudible modes, but are different from adaptive modal contact resolution. Kim and James [2009] adapt the modes of a nonlinear subspace deformation model but do not consider contact. For non-modal deformable simulation, hierarchical multi-resolution methods accelerate simulations using spatial [Grinspun et al. 2002; Capell et al. 2002] and temporal adaptation [Debunne et al. 2001], whereas we exploit the transient nature of high-frequency modes to provide temporal adaptivity. Harmon et al. [2009] consider the more complex case of asynchronous contact mechanics wherein numerous deformable elements are integrated asynchronously with timesteps dependent on element type, proximity, and contact attributes. In contrast we only consider body-level asynchrony wherein each deformable model is synchronized with its contact group, and the timestep size is determined by its highest frequency mode.

3 Low-Noise Contact Resolution

In this section we briefly review multibody contact and the Staggered Projections (SP) solver [Kaufman et al. 2008], then discuss its noise-related limitations for sound synthesis (in §3.2), and propose a modified SP solver which produces low-noise contact impulses efficiently (in §3.3). We refer the reader to [Brogliato 1999; Kaufman et al. 2008] for more details.

3.1 Background on Contact Problems

Following the notation of Kaufman et al. [2008], we consider a multibody contacting system represented by the generalized position coordinate q, generalized velocity $\dot{\mathsf{q}}$, and mass matrix M. The world-frame position of point i on some object is given by the mapping $\boldsymbol{x}_i(\mathsf{q})$ which describes rigid body motion and linear modal deformations. Contacts in the system are described by the set $\mathcal{C}$. Given a contact $k \in \mathcal{C}$ between two points i and j with a normal direction $\boldsymbol{n}_k$ (Figure 5a), the corresponding generalized normal direction is given by $\mathsf{n}_k = \Gamma_k^T \boldsymbol{n}_k$ where Γ_k is the relative velocity Jacobian defined as $\Gamma_k = \nabla \boldsymbol{x}_i - \nabla \boldsymbol{x}_j$. The generalized normal contact impulses can be represented by the vector $\mathsf{c} = \mathsf{N}\alpha$, where $\mathsf{N} = [\mathsf{n}_1 \mathsf{n}_2 \ldots \mathsf{n}_{|\mathcal{C}|}]$ and α is the vector of magnitudes of normal contact impulses.



(a) **(b)**

Figure 5: Contact geometry and friction cone sampling

The isotropic Coulomb friction on the tangential plane can be linearized using $\boldsymbol{f}_k = \boldsymbol{T}_k \beta_k$, where $\boldsymbol{T}_k$ is the matrix whose column vectors uniformly sample the friction disk (Figure 5b), and β_k is the vector of impulse magnitudes along each of the sampled directions. This polyhedral friction cone simplifies the Coulomb friction inequality into

$$e^T \beta_k \le \mu_k \alpha_k, \quad s.t. \quad \beta_k \ge 0, \tag{1}$$

where μ_k is the coefficient of friction, and α_k is the normal contact impulse magnitude at k and $e = [1 \ldots 1]^T$ [Stewart and Trinkle 1996]. The generalized friction impulses on all of the contacts can be written as $\mathsf{f} = \mathsf{D}\beta$, where $\mathsf{D} = [\Gamma_1^T \boldsymbol{T}_1 \ldots \Gamma_{|\mathcal{C}|}^T \boldsymbol{T}_{|\mathcal{C}|}]$ and $\beta = [\beta_1^T \ldots \beta_{|\mathcal{C}|}^T]^T$. Using this notation, we can discretize the Euler-Lagrange equation (with timestep size h) as follows,

$$\mathsf{M}\left(\dot{\mathsf{q}}^{t+1} - \dot{\mathsf{q}}^t\right) = h\mathsf{g}(\mathsf{q}^t, \dot{\mathsf{q}}^t) + h\mathsf{f}_{ext}^t + \mathsf{N}\alpha^{t+1} + \mathsf{D}\beta^{t+1}\ , \tag{2}$$

where g, the *quadratic velocity vector* function, provides the Coriolis and centrifugal forces, and f_{ext} describes the external forces.

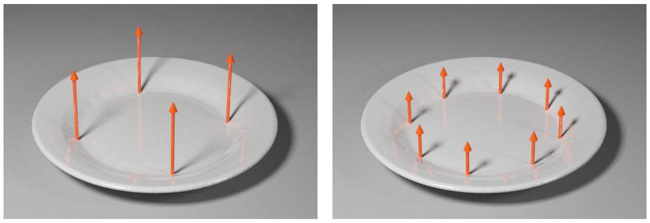


Figure 6: Non-unique contact impulses can produce noise *since fluctuations in resting contact forces, while causing no motions, produce changing modal forces which can produce noise, e.g., resting objects that "hum."*

The SP method solves this equation using a predictor-corrector method. First, it computes a velocity prediction $\dot{\mathsf{q}}^p$ by solving

$$\mathsf{M}\left(\dot{\mathsf{q}}^p - \dot{\mathsf{q}}^t\right) = h\mathsf{g}(\mathsf{q}^t, \dot{\mathsf{q}}^t) + h\mathsf{f}^t_{ext}. \tag{3}$$

Next, it solves

$$\mathsf{M}\left(\dot{\mathsf{q}}^{t+1} - \dot{\mathsf{q}}^p\right) = \mathsf{N}\alpha^{t+1} + \mathsf{D}\beta^{t+1} \tag{4}$$

to correct the velocity, which involves estimating $\dot{\mathsf{q}}^{t+1}$ by simultaneously solving the following two quadratic programming problems iteratively,

$$\dot{\mathsf{q}}^{t+1} = \arg\min_{\mathsf{v}} \left(\frac{1}{2}\mathsf{v}^T\mathsf{M}\mathsf{v} - \mathsf{v}^T(\mathsf{M}\dot{\mathsf{q}}^p + \mathsf{D}\beta^{t+1})\right), \qquad s.t. \quad \mathsf{N}^T\mathsf{v} \geq 0 \tag{5a}$$

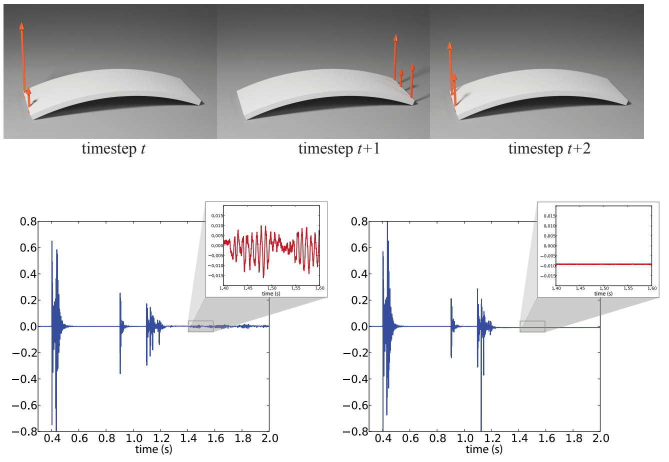
$$\beta^{t+1} = \arg\min_{\beta} \left(\frac{1}{2}\beta^T\mathsf{D}^T\mathsf{M}^{-1}\mathsf{D}\beta + \beta^T\mathsf{D}^T(\dot{\mathsf{q}}^p + \mathsf{M}^{-1}\mathsf{N}\alpha^{t+1})\right), \qquad s.t. \quad E^T\beta < \mathrm{diag}(\mu)\alpha^{t+1},\ \beta \geq 0, \tag{5b}$$

where (5a) is the contact problem for determining the normal impulse, and (5b) is the friction problem that determines the frictional impulse. Here E is the matrix form of all the vectors e in (1) such that each column k has ones in rows corresponding to entries in the subvector $\beta_k \in \beta$ and zeros in all other elements; α^{t+1} in the second problem (5b) is a vector of normal contact impulse magnitudes, which is the Lagrange multipliers of the constraints of the first problem (5a). These QP problems are essentially the dual form of the LCP formulation modeling the contacting systems. We discuss methods for solving these large sparse QP problems later (§6).

3.2 Non-unique Contact Impulses and Noise

There are fundamental difficulties when solving (5) [Brogliato 1999], which are problematic for sound synthesis:

1. First, given a fixed β^{t+1}, it can be proved that equation (5a) has a unique solution, since the mass matrix M is positive definite [1]. However, the Lagrange multipliers of the constraints, i.e., the contact impulses, are not necessarily unique because, according to the Karush-Kuhn-Tucker (KKT) condition of (5a), the Lagrange multipliers should satisfy
$$\mathsf{N}\alpha^{t+1} = \mathsf{M}(\dot{\mathsf{q}}^{t+1} - \dot{\mathsf{q}}^p) - \mathsf{D}\beta^{t+1}. \tag{6}$$
This equation could have non-unique solutions when N is rank deficient, and rank-deficient N is almost inevitable when the number of contacts is large enough. As a result of this non-uniqueness, the contact impulses can be temporally incoherent, and using them to excite the sound model can lead to audible noise artifacts (see Figure 6).



(a) *Velocity-based Contact Sounds* **(b)** *Our Position-based Contact Sounds*

Figure 7: Noisy contact forces on a static curved slab: *(Top) At timestep t, the non-zero contact forces appear at the left end of the slab; at timestep $t + 1$, they are at the right end of the slab; at the next time step, the non-zero forces come back to the left end, and so on so forth. This microscopic contact cycling leads to noise (a) generated rather than pure silence (b) as we expected in this case.*

2. The second difficulty comes from the friction problem (5b), in which the Hessian matrix $\mathsf{D}^T\mathsf{M}^{-1}\mathsf{D}$ tends to be singular when the number of contacts is large—note that the size of the problem is proportional to the number of contacts. Solving such large rank-deficient quadratic programming problems will quickly slow down the simulation.

We therefore seek to estimate a temporally coherent set of active contacts of minimal size.

3.3 Estimating Temporally Coherent Active Contacts

Contact Generation: In standard multi-body simulation, a contact is generated when (i) collision between two objects is detected and (ii) their relative normal velocity at the collision point is negative. These criteria can produce plausible motion effects; however, they tend to also generate noisy contact forces which is problematic for sound synthesis. For example, consider the curved slab resting on the ground with multiple contact points (Figure 7). The traditional contact generation produces non-zero forces which cycle among multiple contact points as the simulation proceeds: contact forces generated to satisfy constraints at the current contact points at the current timestep can cause other contact constraints to be violated at subsequent timesteps.

To avoid contact generation noise, we propose to generate contacts whenever intersections are detected, irrespective of the relative contact velocity. This criteria can generate more contacts than the standard one, leading to more expensive contact problems. In particular, the friction solve (5b) becomes harder since the problem size is proportional to the number of contacts. We therefore propose the following contact filtering scheme to reduce the number of contacts and maintain temporal coherence.

Contact Reduction: We maintain active contacts by applying the *active set method* [Gill et al. 1981] when solving the contact problem (5a). When solutions are found, the active set method also identifies the active constraints for that problem. The resulting con-

[1] Cottle et al. [1992] proved that the LCP form of (5a) has a unique solution if M is a *P-matrix*—a matrix where all principal minors have positive determinants. A positive definite matrix is a P-matrix, however, P-matrices are not necessarily positive definite.

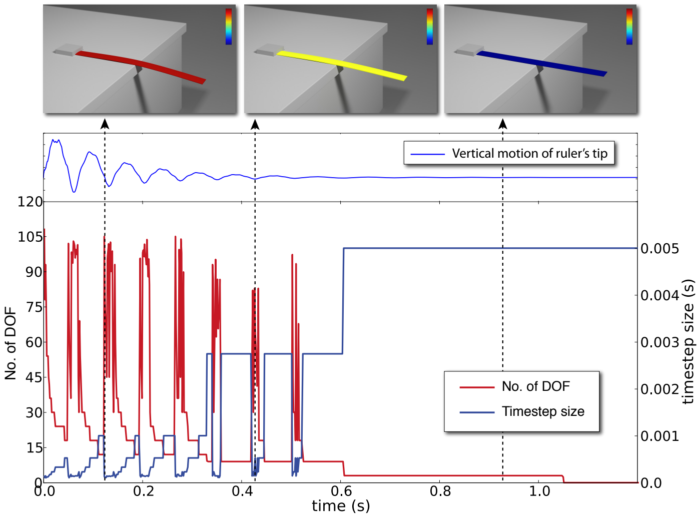


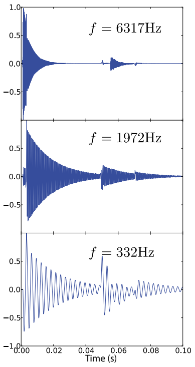
Figure 8: Mode-adaptive contact simulation *(ruler example)*

straint Lagrange multipliers, which correspond to contact forces, become quite sparse. Since zero contact force always leads to zero friction force, we can turn off all inactive contacts to reduce the size of any subsequent friction problem (5b). To ensure temporally coherent active-contact selection (as well as faster SP convergence), we "warm start" the SP solver with the solution from the previous timestep [Kaufman et al. 2008]. To initialize the contact filtration process, we can first solve (5a) using an interior point method, then use that as an initial guess for the active set method. On average, we observe about one order of magnitude speed up over the simulation without contact reduction.

4 Asynchronous Adaptive Contact Solver

By simulating flexible objects we can resolve more interesting dynamic behaviors, such as contact coupling and chattering. These behaviors can enable more realistic visual effects and richer sounds. However, deformable simulation is much more expensive than purely rigid simulation. Furthermore, the simulation timestep size is restricted by the highest modal vibration frequency due to the stability condition—the higher frequency, the smaller the timestep has to be. For high-quality sound synthesis where all audible modes are desired, this simulation cost can be prohibitive.

Fortunately higher frequency vibrations tend to damp more quickly, and are often perceptually important for only a short time (see inset time-series). Afterwards, simulating only lower frequency modes (or even rigid objects) can suffice to capture perceptually important dynamics. We propose to exploit this transient nature of higher frequency vibrations by adaptively selecting the simulated modes and asynchronously integrating the system using the largest timestep possible (see Figure 8). To avoid diminishing returns, we can also limit the maximum mode frequency used in contact simulation, leaving higher frequency modes to be considered only during a subsequent sound synthesis phase (§5). An overview of our asynchronous adaptive integration scheme is outlined in Algorithm 1. Next we will describe each of its steps.



Algorithm 1: **Overview of asynchronous adaptive integration**

```
begin
    while simulation is not over do
        ao ← queue.pop()
        foreach obj ≠ ao do
            interpolate_state(obj, ao.timestamp)
        end
        detect_collision()
        identify_contact_groups()
        foreach contact group cg do
            advance_to_next_step(cg); // Algorithm 2
        end
        foreach contacting object obj do
            increase_timestamp(obj)
            reschedule(obj)
        end
    end
end
```

Asynchronous Integration: First, consider a simulation without contacting bodies. To evolve the objects independently, each of them has a private timestamp indicating its local simulated time. This timestamp is similar to the concept of *local virtual time* in the rigid timewarp method [Mirtich 2000]; it is used to schedule the objects in a priority queue. At each simulation step, the object at the top of the queue has the earliest timestamp and is popped to advance to its next timestep. If it is not in contact, we simply reschedule it into the priority queue after advancing its state and increasing its timestamp. Otherwise, we need to resolve collisions before rescheduling it.

Collision Detection: In this section, we refer the object currently popped from the priority queue the *active object*. To detect collisions, all the other objects need to synchronize to it. These objects are all advanced no less than the active object, since they are deeper in the priority queue. We therefore synchronize them by linearly interpolating their states at the time of the active object. While requiring a little extra memory since each object now needs to maintain two states from last two consecutive timesteps, linear interpolation introduces almost no performance overhead in the simulation. Next the standard discrete-time collision detection is performed using oriented bounding box (OBB) hierarchies for rigid objects [Gottschalk et al. 1996]; for modal deformation, we use Bounded Deformation Tree ideas to quickly update the OBB hierarchy (with fixed bounding-box orientation) [James and Pai 2004].

Contact Groups: Based on the detected intersections, objects are grouped into *contact groups* [Mirtich 2000]; the component objects of a contact group intersect with each other and are separated from other contact groups. Therefore they must be integrated as a unit. We identify these contact groups by detecting independent connected sets in the contact graph [Guendelman et al. 2003].

State Rollback: For any object that is more advanced than the active object, if it is not in contact at all its interpolated state is simply discarded, and it remains at its current advancement; otherwise, it must belong to some contact group and has to be integrated within that group. Therefore, we roll back its current state by replacing it with its interpolated state (see Figure 9 for an illustration). Then each contact group is integrated independently using an adapted timestep size according to the algorithm described in the rest of the section (See Algorithm 2).

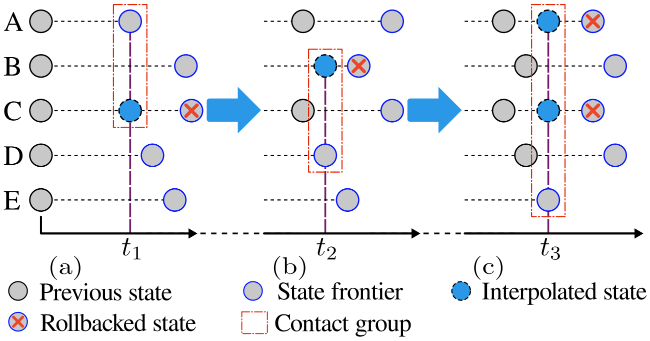


Figure 9: Asynchronous Advancement of Objects: *(a) At time t_1, object A is selected as the current active object. It is in contact with a more advanced object C. Then C synchronizes with A using its interpolated state at t_1, and advance from there together with A; (b) Next D is at the top of the queue, and is in contact with B at t_2. B rolls its state back to t_2, and advances its state with D; (c) Then E is processed, which is in contact with A and C. Therefore E, A and C advance together to the next timestep.*

4.1 Contact Group Advancement

We solve the contact-friction problem described in section 3.1 to advance a contact group. In particular, for velocity prediction (Equation 3), we use explicit forward Euler method and the Newmark integrator [Wriggers 2006] to integrate rigid motion and modal deformation respectively; velocities are then corrected using the SP iterations. The simulated vibration modes, i.e. the simulated degree of freedom (DOF), are determined based on the current modal vibrating state–a mode that is largely damped can be safely ignored from simulation. However, the current vibrating state of a mode cannot be determined without solving the contract-friction problem. Therefore, we first estimate the current normal contact impulses (line 2 of Algorithm 2), then use the estimated impulses to determine the simulated DOF (line 3), and finally solve the adapted problem (line 5-6).

Normal Contact Impulse Estimation: The normal contact impulses are estimated by solving a single QP problem (5a), wherein we use the friction impulses from the last timestep assuming temporal coherence. This leads to one extra QP solve at each timestep. Fortunately, this QP is well-posed and is much cheaper to solve compared to the full SP algorithm. The timestep size used in the solve is determined as follows.

Timestep Determination: Due to the integration stability condition, the timestep size of integrating a contact group is restricted by the highest simulated vibration frequency $\mathtt{f}_h$ of its component objects. In practice, we found that a timestep size $\Delta t = 1/(6.5\mathtt{f}_h)$ produces stable modal deformation. In addition, a default timestep size Δt_{max} is used for rigid bodies, giving us the timestep size of a contact group as $\Delta t = \min(1/(6.5\mathtt{f}_h), \Delta t_{max})$. For contact impulse estimation mentioned above, we use the $\mathtt{f}_h$ value from last timestep (line 1). To finally advance the contact group, $\mathtt{f}_h$ is updated (line 4) after the simulated DOF is adapted.

Adaptation (DOF Increase): In the following description, we denote the maximum number of simulated modes of an object as N_m. Assume these modes are labeled from 1 to N_m, ordered from lower to higher frequencies; and let s denote the current number of simulated modes (modes $1 \ldots s$). To adapt the simulated DOF of an object, we first check if s should be increased. Large external impulses can excite high-frequency modes, thus activating them in the simulation. We measure the excitement of modes by considering the time derivative of the modal vibration equation, i.e.

Algorithm 2: **Adaptive timestep for contact group** C : advance_to_next_step(contact$_\mathrm{g}$roup)

```
begin
1   ts ← determine_timestep_size(C)
2   $f_c$ ← contact_solve(C, ts)
3   adapt_dof(C, $f_c$)
4   ts ← determine_timestep_size(C)
    if cull_friction_solve(C) then
5       frictionless_contact_solve(C, ts)
    else
6       staggered_projections(C, ts)
    end
end
```

$$\ddot{\mathsf{v}} + \mathsf{C}\dot{\mathsf{v}} + \mathsf{K}\mathsf{v} = \mathsf{U}^T \dot{f} \ . \tag{7}$$

The impulse response of this equation measures the velocity excitement of modes due to external impulses, and has the magnitude proportional to $\mathsf{U}^T \dot{f}$. The time derivative $\dot{f}$ is estimated using $(f_c - f^{n-1})/\Delta t$, where f_c is the estimation of normal contact impulse described above, and f^{n-1} is the normal contact impulse from last timestep. Then we approximate the velocity excitement vector using

$$\hat{\boldsymbol{v}} = \mathsf{U}^T \frac{f_c - f^{n-1}}{\Delta t}, \tag{8}$$

and find the highest mode j such that $\hat{\boldsymbol{v}}_j \geq \delta_u$, where δ_u is a parameter to control the upgrade criterion. In practice, we use $\delta_u = 1E - 5$ for all the adaptive simulations. We increase the simulated modes up to j if $j > s$.

Adaptation (DOF Decrease): If s is not increased, we further check if it could be decreased to enable faster simulation. A mode could be deactivated if its modal vibration energy has been largely dissipated. The modal energy of mode i is computed using

$$\mathsf{E}_i = \frac{1}{2}(\dot{\mathsf{q}}_i^2 + \mathsf{k}_i \mathsf{q}_i^2) \ , \tag{9}$$

where q and $\dot{\mathsf{q}}$ are its modal vibrational displacement and velocity, and k_i is its mass-normalized modal stiffness. We then find the highest mode j such that $\mathsf{E}_j \geq \delta_d$, where δ_d is a parameter to control the downgrade criterion. $\delta_d = 1E - 7$ is used in our examples. We decrease the simulated modes down to j if $j < s$. If none of the modes has modal energy larger than δ_d, then the modal deformation is deactivated, and only rigid motion is considered.

Culling Friction Solves: We note that the contact-friction problem gets largely simplified for frictionless contacts. Not only is this because the friction problem (5b) is fundamentally harder to solve than the contact problem (5a) but also no iteration is needed at all in this case. When objects are resting without any static friction, ignoring friction introduces no error. In simulations, this tends to happen frequently, since objects always tend to become static due to energy dissipation. We detect such frictionless contact groups by applying the following heuristics: given a contact group possessing a set of contacts denoted by $\mathcal{C}$ and a set of objects denoted by $\mathcal{O}$, its friction solve can be turned off if it simultaneously satisfies two conditions:

$$\sum_{i\in\mathcal{C}} |\boldsymbol{v}_i^{(t)}| < \delta_t, \tag{10}$$

where $\boldsymbol{v}_i^{(t)}$ is the predicted relative tangential velocity (after integrating using (3)) at contact i, and δ_t is a positive value close to zero (1E-8 in practice); and

$$\sum_{j \in \mathcal{O}} \|\mathsf{f}_j^{(n-1)}\| < \delta_f, \tag{11}$$

where $\mathsf{f}_j^{(n-1)}$ is the generalized friction impulse of object j from last timestep, and δ_f is also a small value (1E-10 in practice). Note that these heuristics are conservative: (10) guarantees no relative movement in the contact group, while (11) ensures no friction force applied in the last timestep. In our experiments, we observe 6%-24% additional performance improvement.

5 Sound Synthesis

In this section, we describe how the contact simulation fits into our two-pass sound synthesis pipeline (§5.1), and how impulses generated during contact resolution are spatially redistributed prior to exciting all-frequency modal sounds (§5.2), then used to parameterize our modal contact damping model (§5.3).

5.1 Sound Synthesis Pipeline

Aside from the sound-aware contact resolution approach, our sound synthesis pipeline is similar to the approach used in [Zheng and James 2010]. Specifically, solid objects are represented using tetrahedral meshes; modal vibration models are precomputed using the finite element method; the modal sound model is excited by contact impulses obtained from dynamic simulation; sound radiation is approximated using Helmholtz acoustic transfer models represented via precomputed multipole expansions, and used with *head related transfer functions* (HRTF) for sound rendering.

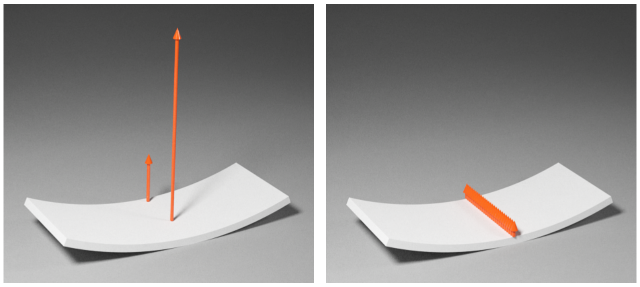
While our method could generate sound as the simulation advances, in practice we use a *two-pass implementation*. The first pass simulates the dynamics while recording the time series of contact impulses to disk. The second pass synthesizes sound by integrating the modal vibration equations of excited objects. Each object's recorded contact impulses are first spatially redistributed (§5.2) before determining modal excitations, and then used to determine each mode's contact damping (§5.3). To support integration of numerous modes, mode-level parallelism is exploited. Overall the second pass is much faster than the first dynamics pass (See Table 2). Once the expensive dynamics is computed, we can quickly resynthesize the sound, e.g., to efficiently tune sound-related damping parameters, such as α, β and γ in (19) which are perceptually important [Klatzky et al. 2000].

5.2 Impulse Redistribution

The algorithms of §3 and §4 described how to generate temporally coherent low-noise force impulses; however, for efficiency, these impulses are applied only at a minimal number of active contacts selected by the active set method (see Figure 10a). Applying these nonphysically sparse impulses to the modal sound model leads to increased noise in the synthesized sound which we would like to avoid. Fortunately we can again exploit the non-uniqueness of the impulse solution to redistribute contact impulses after each simulation timestep to obtain a more uniform spatial distribution (see Figure 10). We emphasize that these redistributed impulses are only used for sound synthesis, and do not affect the simulated motion–which uses sparse contact impulses for speed.

At the end of each timestep, let $\bar{\alpha}$ and $\bar{\beta}$ denote the normal contact and friction impulse magnitudes, respectively, as a result of solving (5); the associated contact and friction impulses are

$$\mathsf{c} = \mathsf{N}\bar{\alpha} \;\text{ and }\; \mathsf{f} = \mathsf{D}\bar{\beta}. \tag{12}$$



(a) *Without refinement* **(b)** *With refinement*

Figure 10: Impulse Refinement: *A curved slab is sitting on the ground. The orange arrow bars indicate contact force distribution with their length proportional to the magnitude of the forces. (a) The quadratic programming solve somewhat arbitrarily selects only a few contacts to be active; (b) our impulse refinement algorithm produces more uniformly distributed forces among all the contacts.*

Our adjusted impulse distribution can be written as

$$\alpha = \bar{\alpha} + \boldsymbol{N}_{\mathsf{N}} k_\alpha \;\text{ and }\; \beta = \bar{\beta} + \boldsymbol{N}_{\mathsf{D}} k_\beta \tag{13}$$

where $\boldsymbol{N}_{\mathsf{N}}$ and $\boldsymbol{N}_{\mathsf{D}}$ are the orthonormal matrices spanning the null space of N and D, respectively; we obtain these matrices using the LAPACK [Anderson et al. 1999] routine geqp3. Since both $\mathsf{N}\boldsymbol{N}_{\mathsf{N}} = 0$ and $\mathsf{D}\boldsymbol{N}_{\mathsf{D}} = 0$, the impulse redistribution does not change the total impulses c and f, and thus does not affect the simulated dynamics. To make these more uniformly distributed, we solve a QP problem for k_α and k_β:

$$\begin{aligned} \arg\min_{k_\alpha, k_\beta} \frac{1}{2} \left(\|\bar{\alpha} + \boldsymbol{N}_{\mathsf{N}} k_\alpha - \alpha^{t-1}\|^2 + \|\bar{\beta} + \boldsymbol{N}_{\mathsf{D}} k_\beta - \beta^{t-1}\|^2 \right) \\ \bar{\alpha} + \boldsymbol{N}_{\mathsf{N}} k_\alpha \geq 0 \\ s.t. \qquad \bar{\beta} + \boldsymbol{N}_{\mathsf{D}} k_\beta \geq 0 \\ E\left(\bar{\beta} + \boldsymbol{N}_{\mathsf{D}} k_\beta\right) \leq \mathrm{diag}(\mu)\left(\bar{\alpha} + \boldsymbol{N}_{\mathsf{N}} k_\alpha\right) \end{aligned} \tag{14}$$

where α^{t-1} and β^{t-1} are the contact and friction impulse magnitudes from the previous timestep. Note that this QP problem is defined for a single object, and therefore each can be solved independently. The problem size is much smaller than in (5), and these solves add only a little overhead to the overall simulation, e.g., only 2%-10% extra cost is observed. However, it results in computed sounds with much less noise (see Figure 11).

5.3 Contact-dependent Modal Damping

Given the distributed contact forces for sound synthesis (from §5.2), we now proceed to efficiently approximate contact-dependent damping forces (recall Figure 4). Prior sound synthesis methods typically use a velocity-proportional Rayleigh damping model where the damping matrix is a linear combination of the mass and stiffness matrices, $\alpha\boldsymbol{M} + \beta\boldsymbol{K}$, and α and β are material-dependent positive scalars [Shabana 1990]. Our contact damping model begins by considering simple viscous dampers at each contact point (see Figure 12) where the viscous coefficient is proportional to the contact force magnitude estimated from dynamic simulation. Specifically, given point contact k with normal direction $\boldsymbol{n}_k$ and contact force magnitude c_k, its vibrational velocity along both normal and tangential directions are

$$\boldsymbol{v}_k^{(n)} = \boldsymbol{n}_k \boldsymbol{n}_k^T \mathsf{U}_k \dot{\mathsf{q}}; \quad \boldsymbol{v}_k^{(t)} = (\boldsymbol{I} - \boldsymbol{n}_k \boldsymbol{n}_k^T)\mathsf{U}_k \dot{\mathsf{q}} \tag{15}$$

respectively, where $\mathsf{U}_k \in \mathbb{R}^{3\times r}$ is the r-mode displacement submatrix corresponding to point k. The viscous contact damping force

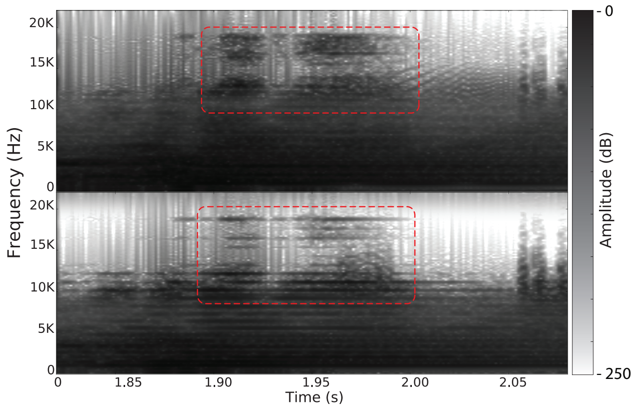


Figure 11: Clearer sounds with impulse refinement: *Contact-impulse noise can result in muddier spectral responses (Top), whereas the spectrogram of sound generated using impulse refinement (Bottom) exhibits less noise, e.g., compare modal frequency lines in highlighted region (See smooth rocking example from Figure 10).*

can be modeled as

$$\boldsymbol{d}_k = \gamma c_k(\boldsymbol{v}_k^{(n)} + \mu \boldsymbol{v}_k^{(t)}) = \gamma c_k \left(\mu \boldsymbol{I} + (1-\mu)\boldsymbol{n}_k \boldsymbol{n}_k^T \right) \mathsf{U}_k \dot{\mathsf{q}} \ , \tag{16}$$

where μ is the coefficient of friction, and the positive scalar γ is a material-dependent parameter controlling the strength of damping forces. With this model, the generalized damping force for contact k can be written as

$$\mathsf{U}_k^T \boldsymbol{d}_k = \gamma c_k \mathsf{U}_k^T \left(\mu \boldsymbol{I} + (1-\mu)\boldsymbol{n}_k \boldsymbol{n}_k^T \right) \mathsf{U}_k \dot{\mathsf{q}} \ \in \mathbb{R}^r. \tag{17}$$

To express in matrix form, we define

$$\mathsf{G} \equiv \sum_{k \in \mathcal{C}} c_k \mathsf{U}_k^T \left(\mu \boldsymbol{I} + (1-\mu)\boldsymbol{n}_k \boldsymbol{n}_k^T \right) \mathsf{U}_k \ . \tag{18}$$

where $\mathcal{C}$ denotes the set of point contacts at the current simulation step. Then the total effective modal damping of the vibration is

$$\mathsf{C} = \mathsf{U}^T(\alpha \boldsymbol{M} + \beta \boldsymbol{K})\mathsf{U} + \gamma \mathsf{G}. \tag{19}$$

Here α, β and γ are all material dependent constants (see Table 1 for values used in the examples). Note that the contact damping term above is also symmetric positive definite, which guarantees it will dissipate energy.

Unfortunately, due to G this damping matrix is non-diagonal, and couples all vibration modes together thereby leading to vastly more expensive modal dynamics integration costs for sound synthesis, e.g., $O(r^2)$ versus $O(r)$ costs for r modes. Fortunately, we observe that we can still obtain phenomenologically similar results by simply ignoring off-diagonal elements of G, and thus preserve a linear-time modal sound synthesis phase. Therefore the contact damping of mode m is simply $\gamma G_{mm} \dot{q}_m$, where the diagonal matrix coefficient is

$$G_{mm} = \gamma \sum_{k \in \mathcal{C}} c_k \mathsf{U}_{k,m}^T \left(\mu \boldsymbol{I} + (1-\mu)\boldsymbol{n}_k \boldsymbol{n}_k^T \right) \mathsf{U}_{k,m}, \tag{20}$$

where $\mathsf{U}_{k,m} \in \mathbb{R}^3$ is the displacement at contact k of mode m. Note that this damping term depends on the current contact force c_k and modal displacement $\mathsf{U}_{k,m}$, and therefore captures spatial and temporal dependencies. In practice, we compute the contact damping coefficient G_{mm} at each simulation timestep, and cubically interpolate them for use in an IIR filter for modal sound synthesis.

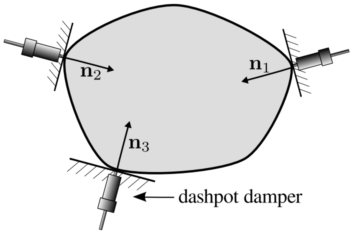


Figure 12: Modified viscous contact dampers *are used to approximate mode-dependent contact-damping phenomenon.*

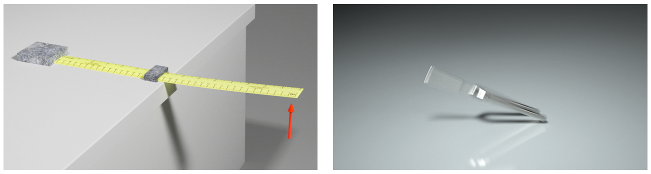


Figure 13: Ruler and tuning fork examples

6 Results

We list all material related parameters used in our examples in Table 1. Simulation statistics are given in Table 2. All reported timings are based on our implementation on a Linux system with an 8-core Intel Xeon X5570 CPU. Please see our video for all sound and animation results, and related comparisons.

Material	Density (kg/m^3)	Young's modulus(GPa)	Poisson's ratio	Damping α	β	γ
Ceramic	2700	7.4E+10	0.19	6	1E-7	3E-2
Polystyrene	1050	3.5E+9	0.34	30	8E-7	4E-4
Steel	7850	2E+11	0.29	5	3E-8	3E-1
MDF	615	4E+9	0.32	35	5E-6	9E-3
Wood	750	1.1E+10	0.25	60	2E-6	5E-4

Table 1: Material Parameters: *The table was made with medium density fiberboard (MDF).*

Example	Surface Vertices	tetrahedra	DOF	Time (s)	Dynamics Cost (hr)	Sound Cost (s)	Sound Modes
Ruler	5760	18860	108	2.0	0.08	7.2	305
Table	362550	2836215	132	2.2	4.8	22.2	1005
Mug	29718	161543	6	3.0	0.12	5.3	64
Tuning Fork	6601	44076	6	2.0	0.08	4.3	30
Pipes	185912	910765	138	12.2	6.3	15.7	842
Marble Tracks	188864	310844	426	35	9.6	32.4	3822
Shopping Cart	215150	992355	186	4.0	4.7	14.2	2060

Table 2: Example Statistics: *Maximum degrees of freedom (DOF) for rigid and modal dynamics. Adaptive dynamics simulations used a highest simulated vibration frequency of $\mathtt{f}_h = 5000$ Hz. Consequently the minimum timestep size is around 3.07E-5s, except for the two purely rigid examples (mug and tuning fork) which used a timestep size of 0.001s. Sound synthesis uses all modes below a 20 kHz cutoff, except for the shopping cart where 12 kHz was used.*

Ruler Twang: We simulated a comedic ruler "twang" sound. We fixed one end of a ruler near the edge of a table (see Figure 13). When excited by a single impulse, the ruler starts chattering against the table, producing a distinctive "twang" sound. Next we move the ruler as it vibrates, changing its contact positions against the table. The changing contact configuration and movement cause varying contact coupling between the modes which leads to a characteristic (and funny) rising-pitch ruler twang. This example highlights the strength of our simulation, since these sound effects could not be synthesized without resolving micro-collision events accurately.

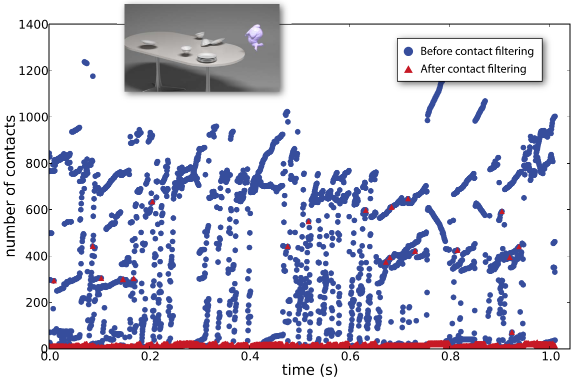


Figure 14: Contact filtering for fast low-noise contact sound: *The stack of plates and other objects generate a large number of contacts at each timestep. Our contact filtering method (§3.3) can effectively cull most contacts, producing fewer and more temporally coherent contacts.*

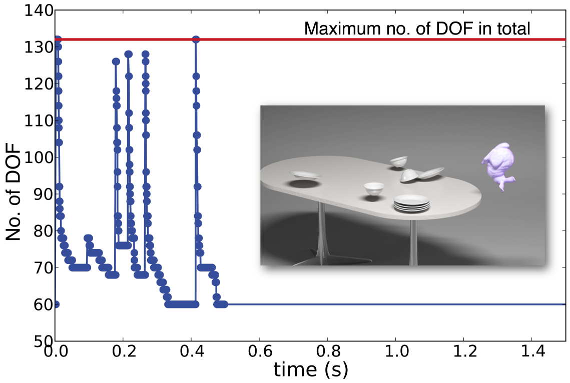


Figure 15: Mode-adaptive contact simulation *of a table (72 modes max) and 10 rigid objects (dishes and bunny) greatly reduces the number of active table modes (max 72) except for impact events.*

Despite the short timescale and rapid collisions, the method is still able to make use of mode adaptation (see Figure 8).

Table: We dropped a heavy rigid body bunny onto a table (fixed to the ground) which had a stack of dinner plates and bowls on it. This example demonstrates the perceptual importance of resolving vibrational coupling for both visual and sound realism (see Figure 2). Using a traditional rigid-body simulation, the table does not deform and thus the kitchenware stays static since no elastic energy can be transmitted by the rigid-body model. In our simulation, however, the stiff table deforms a tiny amount, and consequently all the kitchenware on the table is excited, and even bounced into the air when a heavy bunny is dropped on the table. The audible difference is clear and dramatic. Despite our asynchronous adaptive simulator often only simulating a small number of total modes, it can still resolve these rich vibrational coupling effects (see Figure 15); comparing to our non-adaptive simulation, we observed about 2.6X speedup. This example also illustrates the low-noise sound generation of our method and contact filtering (see Figure 14)

Coffee mug: To demonstrate our contact damping model, we simulated a coffee mug being tapped at a fixed position with different contact orientations (see Figure 4). For validation, we compared our synthesized sounds with recordings of a real mug experiment to demonstrate qualitatively similar damping effects.

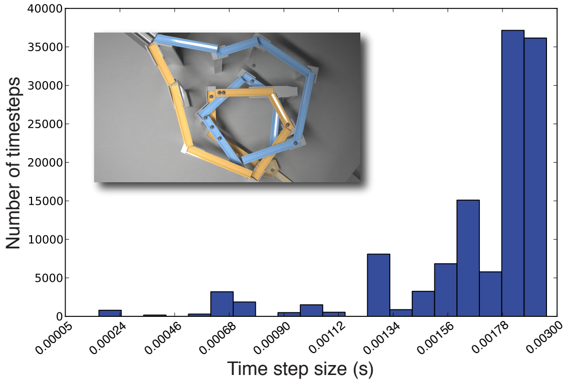


Figure 16: Histogram of timestep size *("Rube-Goldberg" example) demonstrates that very small timesteps are used rarely, e.g., to resolve transient high-frequency modes.*

Tuning fork: We also demonstrate contact damping using a tuning fork dropped on the ground (see Figure 13). The tuning fork can ring for a long time when excited by an impulse and not in contact. However, the ringing rapidly dissipates when the tuning fork is in contact with the ground.

Rube-Goldberg Contraption: This example (see Figure 1) demonstrates the ability of our simulator to handle more complicated animations. It consists of three parts: (1) a block feeder with flexible marble-loaded tubes that eject marbles; (2) a myriad of plastic marble tracks that guide marbles to fall into a wood cup, which is attached to a raised lever, which will eventually lift a bunny and let it fall into a shopping cart; (3) the shopping cart then descends a bumpy incline and hits a curb. To aid in timing and construction, we simulated the three parts separately, using ending conditions from the previous stage as initial conditions for the next stage.

The flexible tubes barely touch the block feeder. Because of the friction forces, the tubes are deformed as the feeder moves. When the tubes deform sufficiently, the friction forces cannot maintain the deformation, and interesting squeaking sounds are produced. To model the friction sounds of the block feeder against the table, we perturb the normal directions on the bottom of the feeder using Gaussian noise, analogous to the normal maps used in [Ren et al. 2010]. The double-helix marble runs illustrate the efficiency of our adaptive simulation: we observed a 6.8× speedup over non-adaptive simulation due to the fewer active modes (see Figure 16 and 17). The small metal balls have lowest modal frequencies well above 20 kHz, so all ball impact sounds ("clicks") were approximated using a recorded metal-ball sound.

Solving large-scale sparse QP problems: We use third-party QP solvers in our implementation. Unlike [Kaufman et al. 2008], we can not use the robust QP solver referred to as "QL" [Schittkowski 2005] since it only supports *dense* QP which become impractical for larger problems involving numerous modes. Unfortunately, in our experience, none of the large-scale *sparse* QP solvers are robust enough to successfully solve all of the problems generated by our contact simulations: they can fail to find optimal solutions on feasible problems. Since their failure cases are often rare and different, our practical solution was to use two solvers: the *GALAHAD* package's *QPC* routine (`galahad.rl.ac.uk`), and *KNITRO* (`www.ziena.com/knitro.htm`). Both of them implement the active set method and the interior point method. In our implementation, we first try to solve using the QPC routine, then, if

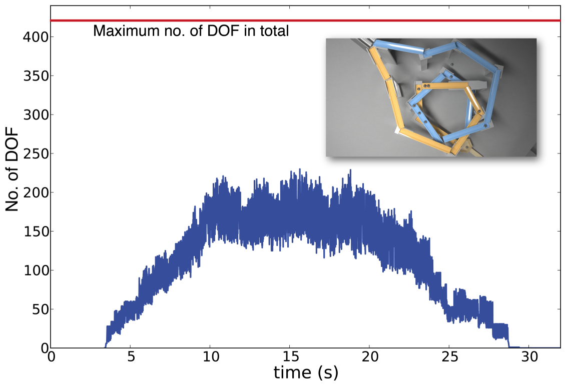


Figure 17: Mode-adaptive contact simulation

it fails, we switch to the active set method in KNITRO. In very rare cases, the second try also fails, then we change to the interior point method implemented in the *QPB* routine in GALAHAD. Using this scheme, we could find optimal (or occasionally only near optimal) solutions to all contact problems.

7 Limitations and Future Work

We have proposed a detailed approach for contact resolution which is able to synthesize a new range of challenging sound phenomena, such as vibration-induced chattering and contact damping, as well as to avoid deficiencies in current practice, such as contact-solver-induced noise. We have also proposed an adaptive asynchronous contact solver which makes it possible to efficiently simulate coupling of rigid and modal objects via frictional contact. As a result, we have enabled high-quality sound synthesis for a range of previously unexplored sound phenomena.

Our approach and implementation are not without limitations, and there are many opportunities for future work. While it is clear that speed-accuracy trade-offs can always be made, we are interested in faster methods which retain accuracy. Robustness of the frictional contact solver is dependent on our ability to solve large-scale sparse quadratic programs robustly, and current QP solvers need improvement[2]. Our post-dynamics impulse refinement step generates friction impulses for the sound phase which are not guaranteed to satisfy the maximum dissipation optimality conditions. Our method involves several user-specific parameters, such as for adaptivity, which although straightforward to set in our examples, may be difficult to tune in more complex cases. Further studies are needed into the simulation of squeaking and other high-frequency coupling phenomena. Modal analysis provides only a partial theory for contact sounds, since very small objects, such as marbles, have vibration frequencies which can far exceed our hearing limits, yet they still produce audible contact "clicks." Generalizing contact, sound and radiation models to support larger deformation scenarios is a major unsolved problem for modal sound synthesis. Our current implementation exhibits triangle faceting artifacts, which can be a problem for rolling phenomena [van den Doel et al. 2001]. Finally there is more to computing multibody contact sounds than just simulating vibrations: multibody radiation effects can also play a significant role in the resulting sound. For example, the sound of a spoon falling into a mug is strongly affected by the mug's cavity resonance, but current single-body acoustic transfer models [James et al. 2006; Chadwick et al. 2009; Zheng and James 2010] and large-scale room acoustics techniques [Raghuvanshi et al. 2010] do not entirely suffice.

Acknowledgements: We thank the anonymous reviewers for feedback, Danny Kaufman and Dinesh Pai for discussions on Staggered Projections, and Jon Kaldor and Cem Yuksel for proofreading. This work was supported in part by the National Science Foundation (HCC-0905506), funding and hardware from Intel (ISTC-VC), fellowships from the Alfred P. Sloan Foundation and the John Simon Guggenheim Memorial Foundation, Pixar, and software donations from Autodesk and Advanced CAE Research (FastBEM). 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

ANDERSON, E., BAI, Z., BISCHOF, C., BLACKFORD, S., DEMMEL, J., DONGARRA, J., DU CROZ, J., GREENBAUM, A., HAMMARLING, S., MCKENNEY, A., AND SORENSEN, D. 1999. *LAPACK Users' Guide*, third ed. Society for Industrial and Applied Mathematics, Philadelphia, PA.

ANITESCU, M., AND POTRA, F. 1997. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dynamics 14*, 231–247.

BARAFF, D. 1990. Curved surfaces and coherence for non-penetrating rigid body simulation. In *Computer Graphics (Proceedings of SIGGRAPH 90)*, 19–28.

BARAFF, D. 1991. Coping with friction for non-penetrating rigid body simulation. In *Computer Graphics (Proceedings of SIGGRAPH 91)*, 31–40.

BARAFF, D. 1993. Issues in computing contact forces for non-penetrating rigid bodies. *Algorithmica 10*, 2-4, 292–352.

BONNEEL, N., DRETTAKIS, G., TSINGOS, N., VIAUD-DELMON, I., AND JAMES, D. 2008. Fast modal sounds with scalable frequency-domain synthesis. *ACM Transactions on Graphics 27*, 3 (Aug.), 24:1–24:9.

BROGLIATO, B. 1999. *Nonsmooth Mechanics*, second ed. Springer.

CAPELL, S., GREEN, S., CURLESS, B., DUCHAMP, T., AND POPOVIĆ, Z. 2002. A multiresolution framework for dynamic deformations. In *ACM SIGGRAPH Symposium on Computer Animation*, 41–48.

CHADWICK, J. N., AN, S. S., AND JAMES, D. L. 2009. Harmonic Shells: A practical nonlinear sound model for near-rigid thin shells. *ACM Trans. Graph. 28*, 5, 1–10.

COTTLE, R., PANG, J., AND STONE, R. 1992. *The linear complementarity problem*. Academic Press.

DEBUNNE, G., DESBRUN, M., CANI, M.-P., AND BARR, A. H. 2001. Dynamic real-time deformations using space & time adaptive sampling. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 31–36.

DOEL, K., KNOTT, D., AND PAI, D. 2004. Interactive simulation of complex audiovisual scenes. *Presence: Teleoperators & Virtual Environments 13*, 1, 99–111.

[2] In order to spur algorithmic developments in the QP solver community, a range of challenging QP test problems arising from our simulations will be made available on our website.

ERLEBEN, K. 2007. Velocity-based shock propagation for multibody dynamics animation. *ACM Transactions on Graphics 26*, 2 (June), 12:1–12:20.

GAVER, W. 1993. Synthesizing auditory icons. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, ACM, 228–235.

GILL, P., MURRAY, W., AND WRIGHT, M. 1981. *Practical Optimization*. Academic Press, London, UK.

GOTTSCHALK, S., LIN, M., AND MANOCHA, D. 1996. OBBTree: A Hierarchical Structure for Rapid Interference Detection. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, 171–180.

GRINSPUN, E., KRYSL, P., AND SCHRÖDER, P. 2002. CHARMS: A Simple Framework for Adaptive Simulation. *ACM Transactions on Graphics 21*, 3 (July), 281–290.

GUENDELMAN, E., BRIDSON, R., AND FEDKIW, R. P. 2003. Nonconvex rigid bodies with stacking. vol. 22, 871–878.

HAHN, J. K. 1988. Realistic animation of rigid bodies. In *Computer Graphics (Proceedings of SIGGRAPH 88)*, 299–308.

HARMON, D., VOUGA, E., SMITH, B., TAMSTORF, R., AND GRINSPUN, E. 2009. Asynchronous contact mechanics. *ACM Transactions on Graphics 28*, 3 (July), 87:1–87:12.

JAMES, D. L., AND PAI, D. K. 2004. BD-Tree: Output-sensitive collision detection for reduced deformable models. *ACM Transactions on Graphics 23*, 3 (Aug.), 393–398.

JAMES, D. L., BARBIC, J., AND PAI, D. K. 2006. Precomputed Acoustic Transfer: Output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics 25*, 3 (July), 987–995.

JEFFERSON, D. 1985. Virtual time. *ACM Transaction on Programming Languages and Systems 7*, 3 (July), 404–425.

KAUFMAN, D. M., SUEDA, S., JAMES, D. L., AND PAI, D. K. 2008. Staggered projections for frictional contact in multibody systems. *ACM Transactions on Graphics 27*, 5 (Dec.), 164:1–164:11.

KIM, T., AND JAMES, D. L. 2009. Skipping steps in deformable simulation with online model reduction. *ACM Transactions on Graphics 28*, 5 (Dec.), 123:1–123:9.

KLATZKY, R., PAI, D., AND KROTKOV, E. 2000. Perception of material from contact sounds. *Presence: Teleoperators & Virtual Environments 9*, 4, 399–410.

KRY, P. G., AND PAI, D. K. 2003. Continuous contact simulation for smooth surfaces. *ACM Trans. Graph. 22*, 1, 106–129.

LÖTSTEDT, P. 1982. Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM J. of Appl. Math. 42*, 2, 281–296.

MILENKOVIC, V. J., AND SCHMIDL, H. 2001. Optimization-based animation. In *Proc. of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 37–46.

MIRTICH, B. 2000. Timewarp rigid body simulation. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, 193–200.

MOORE, M., AND WILHELMS, J. 1988. Collision detection and response for computer animation. In *Computer Graphics (Proceedings of SIGGRAPH 88)*, 289–298.

MOREAU, J. 1966. Quadratic programming in mechanics: One-sided constraints. *J. of SIAM Control 4*, 1, 153–158.

O'BRIEN, J. F., COOK, P. R., AND ESSL, G. 2001. Synthesizing sounds from physically based motion. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 529–536.

O'BRIEN, J. F., SHEN, C., AND GATCHALIAN, C. M. 2002. Synthesizing sounds from rigid-body simulations. In *2002 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 175–182.

PAI, D. K., VAN DEN DOEL, K., JAMES, D. L., LANG, J., LLOYD, J. E., RICHMOND, J. L., AND YAU, S. H. 2001. Scanning physical interaction behavior of 3d objects. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 87–96.

RAGHUVANSHI, N., AND LIN, M. C. 2006. Interactive Sound Synthesis for Large Scale Environments. In *SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, ACM Press, New York, NY, USA, 101–108.

RAGHUVANSHI, N., SNYDER, J., MEHRA, R., LIN, M., AND GOVINDARAJU, N. 2010. Precomputed wave simulation for real-time sound propagation of dynamic sources in complex scenes. *ACM Transactions on Graphics 29*, 4 (July), 68:1–68:11.

REN, Z., YEH, H., AND LIN, M. 2010. Synthesizing contact sounds between textured objects. In *IEEE Virtual Reality*.

SCHITTKOWSKI, K. 2005. QL: A Fortran code for convex quadratic programming-users guide, version 2.11. *Research Report, Department of Mathematics, University of Bayreuth*.

SHABANA, A. A. 1990. *Theory of Vibration, Volume II: Discrete and Continuous Systems*, first ed. Springer-Verlag, New York, NY.

STEWART, D., AND TRINKLE, J. 1996. An implicit time-stepping scheme for rigid-body dynamics with inelastic collisions and coulomb friction. *Inter. J. for Numerical Methods in Engineering 39*, 2673–2691.

STEWART, D. 2000. Rigid-body dynamics with friction and impact. *SIAM Review 42*, 1, 3–39.

TAKALA, T., AND HAHN, J. 1992. Sound rendering. In *Computer Graphics (Proceedings of SIGGRAPH 92)*, 211–220.

VAN DEN DOEL, K., AND PAI, D. 1996. Synthesis of shape dependent sounds with physical modeling. In *Proc. of the International Conference on Auditory Display*.

VAN DEN DOEL, K., KRY, P. G., AND PAI, D. K. 2001. FoleyAutomatic: Physically-Based Sound Effects for Interactive Simulation and Animation. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 537–544.

WASFY, T., AND NOOR, A. 2003. Computational strategies for flexible multibody systems. *Appl. Mech. Rev. 56*, 6.

WRIGGERS, P. 2006. *Computational Contact Mechanics*, second ed. Springer Berlin Heidelberg.

ZHENG, C., AND JAMES, D. L. 2010. Rigid-body fracture sound with precomputed soundbanks. *ACM Transactions on Graphics 29*, 4 (July), 69:1–69:13.