Physically Based Sound for Computer Animation and Virtual Environments
Modal Vibration

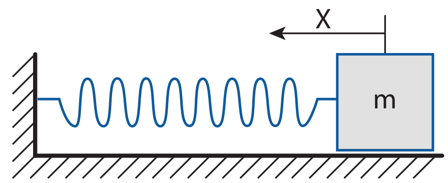
Changxi Zheng
Columbia University

Modal Vibration

Changxi Zheng
Columbia University

TOPICS TO BE COVERED: *Simple harmonic oscillator, mass-spring systems; modal vibration of 3D solids; mass and stiffness matrices; meshing and discretization; generalized eigenvalue problem; eigenfrequencies and eigenmodes; damping models; time-stepping modal vibrations; integration with rigid-body dynamics engines.*

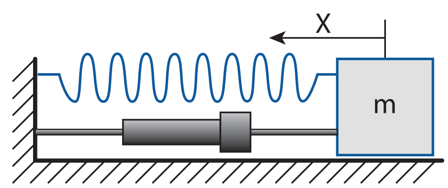
1 Basic Vibration Model



We start by reviewing the basics of vibration theory and refer to the textbook [8] for more details. A spring-mass system exemplifies the basic vibration model, with a single degree of freedom (DoF). Without friction, the position x of the mass is described by the equation

$$m\ddot{x} + kx = 0, \tag{1}$$

where k is the stiffness of the spring and m is the mass. If the spring is initially stretched by a distance of A, it then starts to vibrate, following a sinusoid function $x(t) = A\cos(\omega t)$. Here $\omega = \sqrt{k/m}$ is the angular frequency.



Damped vibration. The spring-mass system (1) has no energy loss and will keep vibrating. To model energy dissipation, one can add a "viscous" damper that generates a damping force proportional to the velocity of the mass and thereby attempts to suppress the current velocity. Let c denote the strength of the damper, then this damped system follows the damped vibration equation,

$$m\ddot{x} + c\dot{x} + kx = f(t), \tag{2}$$

where $f(t)$ describes the (possibly) time-varying external force. This is a second-order ordinary differential equation (ODE). When the external force vanishes, the vibration behavior depends on the damping strength c: if c is larger than the critical damping, $c_c = 2\sqrt{km}$, the system is *overdamped*, with no vibration produced; if $c < c_c$, the system is *underdamped*; it will vibrate but eventually stop.

Impulse response. In order to solve the vibration equation analytically, it is instructional to look at the impulse response of Eq. (2). Assume that the forcing function on the right hand side of (2) is an impulse (i.e., a Dirac delta function $\delta(t)$). If we assume the system starts from rest at $t = 0$, then the vibration that satisfies (2) can be written as

$$x(t) = \frac{1}{m\omega_d} e^{-\xi\omega t} \sin\omega_d t. \tag{3}$$

This is the impulse response function of (2). Here ω is the *undamped natural frequency* of vibration,

$$\omega = \sqrt{\frac{k}{m}},$$

the frequency value when there is no damping (i.e., $c = 0$). ξ is the dimensionless *modal damping factor*,

$$\xi = \frac{c}{2m\omega}.$$

Lastly, ω_d is the *damped natural frequency*,

$$\omega_d = \omega\sqrt{1-\xi^2}.$$

The damped natural frequency is meaningful only when $\xi < 1$, which amounts to requiring the damping coefficient c less than the critical damping c_c.

There are many ways of deriving the impulse response (3). We sketch a simple derivation here using Laplace transform. One can apply the Laplace transform on both sides of (2), and transform it into an algebraic equation,

$$(ms^2 + ds + k)\tilde{x}(s) = 1,$$

where $\tilde{x}(s)$ is the Laplace transform of $x(t)$, and the solution is

$$\tilde{x}(s) = \frac{1}{ms^2 + ds + k}.$$

Applying the inverse Laplace transform on $\tilde{x}(s)$ yields the expression of impulse response (3).

Provided an arbitrary forcing function $f(t)$ on the right hand side of (2), we can now express the solution as a time convolution with the impulse response,

$$x(t) = \int_0^t \frac{f(\tau)}{m\omega_d} e^{-\xi\omega(t-\tau)} \sin\omega_d(t-\tau)\mathrm{d}\tau. \tag{4}$$

Numerical integration. Directly computing the time convolution (4) is expensive and unstable. However, there are many different numerical integration schemes, such as the explicit Euler, implicit Euler, and Runge-Kutta method. One simple approach is by using a small IIR digital filter [5, 9, 10]. Concretely, if the timestep size if h, then the x value at the timestep k is computed as

$$x_k = 2\varepsilon\cos\theta x_{k-1} - \varepsilon^2 x_{k-2} + \frac{2f_{k-1}[\varepsilon\cos(\theta+\gamma) - \varepsilon^2\cos(2\theta+\gamma)]}{3m\omega\omega_d},$$

where $\varepsilon = e^{-\xi\omega h}$, $\theta = \omega_d h$ is the phase shift across a timestep, $\gamma = \arcsin\xi$, and f_{k-1} is the forcing value at the timestep $k-1$.

2 Elastic Vibration

In this section, we introduce the differential equation describing the elastic vibration of a solid body, starting by representing the solid body volume with a finite element mesh. Typical representations of the volume include tetrahedral mesh and hexahedral mesh. Computer graphics community has developed many geometry processing tools to convert a closed surface mesh into a tetrahedral mesh. In the supplementary material of this course, we provide the source code of an implementation of the method of Labelle and Shewchuk [6] for this purpose.

Suppose a tetrahedral mesh consists of n nodes, whose rest positions are $\boldsymbol{X}_i, i = 1\cdots n$. Since the elastic vibration displaces the nodal positions, we denote their time-varying displaced positions as $\boldsymbol{x}_i(t)$ and the nodal displacement as $\boldsymbol{u}_i(t) = \boldsymbol{x}_i(t) - \boldsymbol{X}_i$. Our goal here is to derive a system of equations to describe the solid body's elastic vibration, in a form of

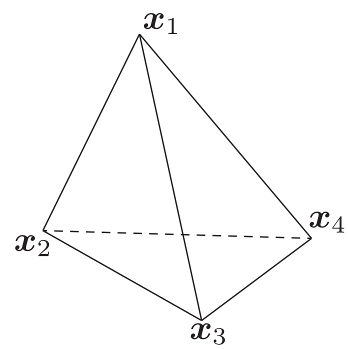
$$\mathsf{M}\ddot{\boldsymbol{u}} + \mathsf{D}\dot{\boldsymbol{u}} + \mathsf{K}\boldsymbol{u} = \boldsymbol{f}(t). \tag{5}$$

This is the high-dimensional equivalent of the vibration equation (2). Here $\boldsymbol{u}$ is a $3n \times 1$ vector stacking the displacement vectors $\boldsymbol{u}_i$ of all nodes. M, K and D are respectively the mass, stiffness, and damping matrices with a size $3n \times 3n$.

The equation (5) arises from discretizing the dynamical elastic equation of continuum body using the standard finite element scheme. We therefore refer the reader to the textbooks [4, 2] for a thorough introduction of continuum mechanics and its finite element simulation. Here we focus on outlining the computational routine that constructs the mass, stiffness, and damping matrices.

Stiffness matrix. Consider a single tetrahedron of the finite element mesh. If it is small, we can approximate the deformation gradient as a constant value inside of the tetrahedron, namely,

$$\mathsf{F} = [\boldsymbol{x}_2 - \boldsymbol{x}_1 \;\; \boldsymbol{x}_3 - \boldsymbol{x}_1 \;\; \boldsymbol{x}_4 - \boldsymbol{x}_1][\boldsymbol{X}_2 - \boldsymbol{X}_1 \;\; \boldsymbol{X}_3 - \boldsymbol{X}_1 \;\; \boldsymbol{X}_4 - \boldsymbol{X}_1]^{-1}.$$



The deformation gradient allows us to further estimate the strain also as a constant tensor in the tetrahedron. In general, the strain tenor is nonlinear with respect to the deformation gradient. Here, since the surface vibration of a rigid body that produces sounds is often very tiny, we consider the linearized strain tensor for small deformations E expressed as

$$\mathsf{E} = \frac{1}{2}(\mathsf{F} + \mathsf{F}^T),$$

and further compute the stress tensor S using the linear constitutive law,

$$\mathsf{S} = \mathsf{C} : \mathsf{E},$$

where C is the symmetric fourth-order tensor, known as the *stiffness tensor*. Depending on specific material parameters, the stiffness tensor is symmetric with $C_{ijkl} = C_{klij} = C_{jikl} = C_{ijlk}$. For example, for isotropic material, the stiffness tensor has the form,

$$C_{ijkl} = K\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk} - \frac{2}{3}\delta_{ij}\delta_{kl}),$$

where δ_{ij} is the Kronecker delta function.

With the stress tensor S computed, we can now estimate the internal elastic force applied on each of the four tetrahedral nodes. Given a stress tensor and a direction $\boldsymbol{n}$, the elastic force pointing along that direction is $\mathsf{S}\boldsymbol{n}$. Thus, for each tetrahedral node, we need to estimate an effective normal direction, which we use the average of the normal directions of the three triangles adjacent to that node. For example, the effective normal of node-1 is the *normalized* vector of

$$\boldsymbol{n}_1 = (\boldsymbol{X}_3 - \boldsymbol{X}_1) \times (\boldsymbol{X}_4 - \boldsymbol{X}_1) + (\boldsymbol{X}_4 - \boldsymbol{X}_1) \times (\boldsymbol{X}_2 - \boldsymbol{X}_1) + (\boldsymbol{X}_2 - \boldsymbol{X}_1) \times (\boldsymbol{X}_3 - \boldsymbol{X}_1).$$

At this point, we have obtained an expression of the internal force introduced by the deformation of a single tetrahedron. Let S_i be the piece-wise constant stiffness tenor of the tetrahedron i and $\boldsymbol{n}_{ji}$ be the effective normal direction of a node j in the tetrahedron i. The total internal force at the node j is

$$\boldsymbol{t}_j = \sum_{i \in \mathcal{T}_j} \mathsf{S}_i \boldsymbol{n}_{ji}, \tag{6}$$

where $\mathcal{T}_j$ indicate the set of tetrahedra incident to the node j. In this expression, $\boldsymbol{n}_{ji}$ is a constant direction. $\boldsymbol{S}_i$ is linearly related to the nodal positions $\boldsymbol{x}_i$. Thus, the internal force $\boldsymbol{t}_j$ linearly depends on the nodal displacement vector $\boldsymbol{u}$, and we express this dependence as

$$\boldsymbol{t} = \mathsf{K}\boldsymbol{u},$$

where $\boldsymbol{t}$ stacks the internal force vectors of all finite element nodes, and K is the stiffness matrix, which can be constructed by rewriting S_i in (6) as a linear function with respect to the displacement vectors $\boldsymbol{u}_j$.

We also note that K is rather sparse. This is because the internal force $\boldsymbol{t}_j$ of a node j depends on only a small set of nodal displacement vectors, as shown above that $\mathcal{T}_j$ consists of only a small number of tetrahedra.

Mass matrix. The mass matrix can be derived by taking the second partial derivative of the kinetic energy with respect to the nodal velocities. If we consider a single tetrahedron again, it contributes a 12×12 submatrix to the final mass matrix. It can be shown that this 12×12 submatrix is a 4×4 block matrix, where each block $B_{ij}, i, j = 1...4$, is a 3×3 diagonal matrix, in particular, $B_{ij} = \frac{1}{20}\rho V(1 + \delta_{ij})\mathsf{I}_{3\times 3}$. Once the 12×12 submatrix is computed, it is added into the $3n \times 3n$ matrix M according to its column and row indices.

Damping matrix. With the stiffness and mass matrices computed, Rayleigh damping model is often used to compute the damping matrix. It defines D as a linear combination of M and K, that is, $\mathsf{D} = \alpha\mathsf{M} + \beta\mathsf{K}$, where both α and β are user-specified parameters. We note that the Rayleigh damping model is not physically principled. In fact, a general and physically based elastic damping model remains unclear. Rayleigh damping has been widely used in many graphics and engineering simulations, because of the numerical convenience it enables: it allows us to decouple the linear vibration system into individual modal vibrations, as introduced in the next section.

3 Linear Modal Analysis

We now numerically solve the system (5) to obtain the object's elastic vibration. We will decouple the system into a set of independent one-dimensional vibration equations (2), each of which can be solved as described in Section 1.

The key idea is to exploit the algebraic properties of generalized eigenvalue decomposition of (M, K). In particular, we perform the generalized eigenvalue decomposition,

$$\mathsf{KU} = \mathsf{MUS},$$

which yields a modal shape matrix U and a diagonal eigenvalue matrix S. Since U is full rank, we can use it as the linear basis to express the displacement vector $\boldsymbol{u}$ as $\boldsymbol{u} = \mathsf{U}\boldsymbol{q}$. We then substitute it into (5) and pre-multiply both sides by U^T, obtaining

$$\mathsf{U}^T\mathsf{MU}\ddot{\boldsymbol{q}} + \mathsf{U}^T\mathsf{DU}\dot{\boldsymbol{q}} + \mathsf{U}^T\mathsf{KU}\boldsymbol{q} = \mathsf{U}^T\boldsymbol{f}(t).$$

Two algebraic properties of the generalized eigenvalue decomposition help to simplify this equation: from the generalized eigenvalue decomposition, we have $\mathsf{U}^T\mathsf{MU} = \mathsf{I}$ and $\mathsf{U}^T\mathsf{KU} = \mathsf{S}$. Meanwhile, with the

Rayleigh damping model, $\mathsf{U}^T\mathsf{D}\mathsf{U} = \alpha\mathsf{I} + \beta\mathsf{S}$, also a diagonal matrix. As a result, the vibration system (5) is transformed into

$$\ddot{\boldsymbol{q}} + (\alpha\mathsf{I} + \beta\mathsf{S})\dot{\boldsymbol{q}} + \mathsf{S}\boldsymbol{q} = \mathsf{U}^T\boldsymbol{f}(t). \tag{7}$$

On the left hand side, all pre-multiplied matrices are diagonal. Thus, equations about every element $\boldsymbol{q}_i$ of $\boldsymbol{q}$ are independent from each other. In other words, instead of solving a set of coupled equations, we can now solve n damped vibration equations independently. For the purpose of sound synthesis, we are interested in the vibrational frequencies in our hearing range (20Hz to 20kHz). Thus, we only need to consider the rows of (7) whose vibrational frequencies are in this range.

4 Simulation Pipeline

In summary, the linear modal vibration of a solid object can be simulated using the following steps.

1. Convert a surface mesh into a volumetric mesh (e.g., a tetrahedral mesh).
2. Provided the physical material parameters, construct the mass, stiffness, and damping matrices.
3. Compute the generalized eigenvalue decomposition $\mathsf{K}\mathsf{U} = \mathsf{M}\mathsf{U}\mathsf{S}$.
4. Decompose the vibration system into individual modal vibrations.
5. Numerically integrate individual modal vibrations.

5 Discussion

Linear modal analysis eases the simulation of linear elastic vibration by decoupling the vibration into independent modes. Yet, this approach is limited. While linear elasticity very often suffices for modeling surface vibration of rigid objects that produce sounds, there exist many sound phenomena related to nonlinear dynamics. For example, thin sheets vibrate nonlinearly when they are struck strongly. This is why many percussion instruments, such as gong and cymbal, are able to produce rich and intriguing sounds. Simulating sound generation from nonlinear surface vibration can leverage many nonlinear simulation techniques developed in engineering and computer graphics (e.g., see [7, 3]). Generally speaking, nonlinear simulation requires to use a very small timestep size (e.g., 1/22050sec) to capture audible frequency content (from 20Hz to 20kHz), and it is hard to decouple the nonlinear vibration into individual vibrational modes—their vibrational modes can be coupled as shown in [3]. Thus, those simulations are computationally much more expensive.

The Rayleigh damping model is just a numerical model, in which the use of linear combination of stiffness and mass matrices as the damping matrix enables to fully decouple the vibrational system into independent modes. However, this damping model is not physical. A major reason is that in general it is unclear which state variables are relevant to determine the damping forces [1]. So far, the damping parameters α and β still require manual tweaking.

References

[1] Sondipon Adhikari. *Damping models for structural vibration*. PhD thesis, University of Cambridge, 2001.

[2] Javier Bonet and Richard D Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

[3] Jeffrey N. Chadwick, Steven S. An, and Doug L. James. Harmonic shells: A practical nonlinear sound model for near-rigid thin shells. *ACM Transactions on Graphics*, 28(5):119:1–119:10, December 2009.

[4] Morton E Gurtin. *An introduction to continuum mechanics*, volume 158. Academic press, 1982.

[5] Doug L. James and Dinesh K. Pai. Dyrt: Dynamic response textures for real time deformation simulation with graphics hardware. *ACM Trans. Graph.*, 21(3):582–585, July 2002.

[6] François Labelle and Jonathan Richard Shewchuk. Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles. *ACM Transactions on Graphics*, 26(3):57.1–57.10, July 2007. Special issue on Proceedings of SIGGRAPH 2007.

[7] James F. O'Brien, Perry R. Cook, and Georg Essl. Synthesizing sounds from physically based motion. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 529–536, New York, NY, USA, 2001.

[8] Ahmed A Shabana. *Theory of vibration: Volume II: discrete and continuous systems*. Springer Science & Business Media, 2012.

[9] Ken Steiglitz. *A Digital Signal Processing Primer, with Applications to Digital Audio and Computer Music*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1996.

[10] Kees van den Doel, Paul G. Kry, and Dinesh K. Pai. Foleyautomatic: Physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 537–544, New York, NY, USA, 2001. ACM.