Physics-based Audio: Models, Computation and Parameter Spaces

Stefan Bilbao
Acoustics and Audio Group
University of Edinburgh

ISMIR
21 September 2025

Overview

Physics-based Audio vs. Data-centred Approaches

A distinguishing feature of physics-based audio: it is algorithmic, so no samples are used.

- "pure" PM codes are tiny (kilobytes usually)…no "data"
- computational/memory costs: highly dependent on target system
- parameter sets: small. Sometimes hard to obtain!
- can be extended to include samples (e.g., key clicks)
- always operate as recursions at an audio sample rate---feedback/IIR systems

Distantly related to "procedural audio," also algorithmic…

$$\ddot{u} = -\omega^2 u$$

"Equation based"

$$u(t) \ = A cos(\omega t + \varphi)$$

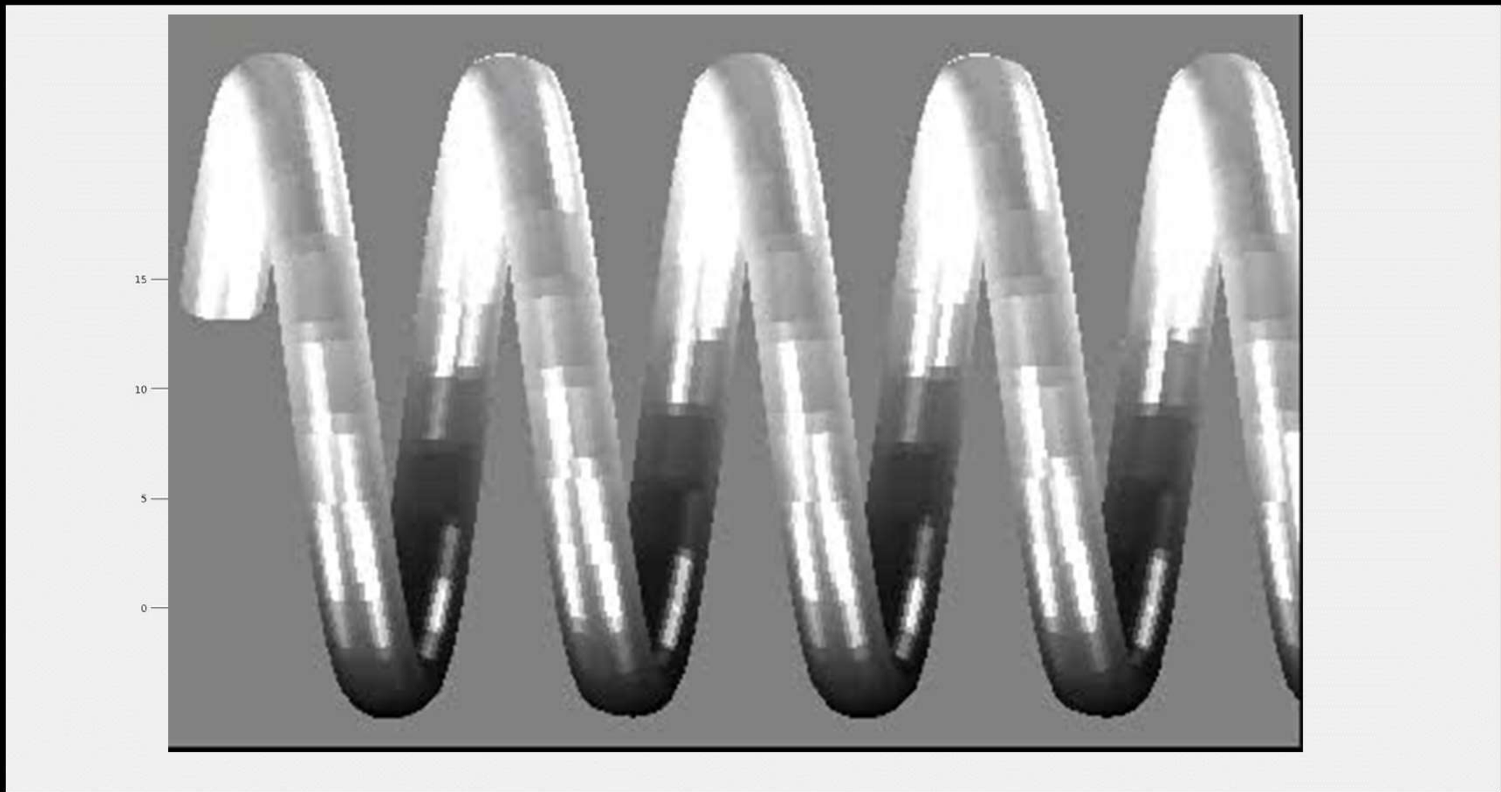
"Solution based"

No training: the deterministic mathematical model (and not the sound it generates) is the "ground truth"…

Physics-based Audio

Idea: obtain high-quality “natural” acoustic sound through simulation. Main system families:

Virtual Acoustics
Synthesis
Effects



System Overview

Virtual acoustics and mechanical effects:

Classic synthesis models:

ODE system

PDE system

Generally: coupled systems of differential equations $\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u})$

Initial conditions: always zero in practice. Instead: physical models activated using external driving functions

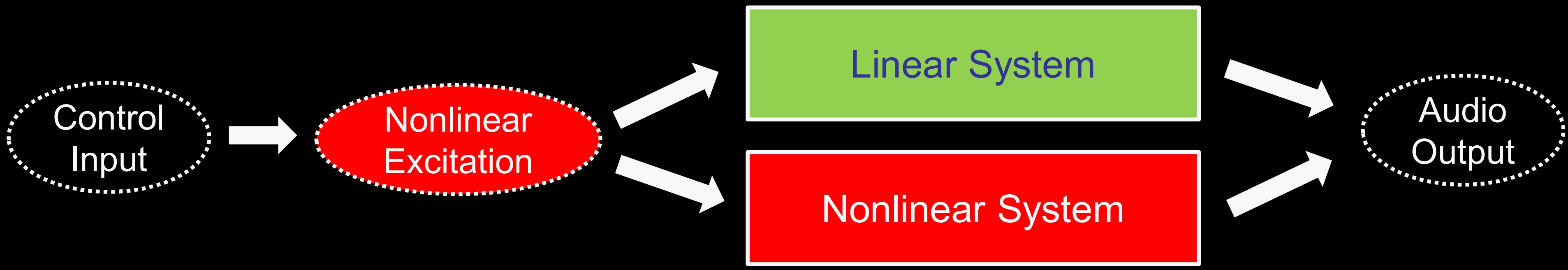
$$\cancel{\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}) \qquad \boldsymbol{u}(0) = \boldsymbol{u}_0}$$

$$\dot{\boldsymbol{u}} = \boldsymbol{f}(\boldsymbol{u}) + \boldsymbol{g}(t) \qquad \boldsymbol{u}(0) = \boldsymbol{0}$$

Loss: generally very low, but of high perceptual significance! “High-Q” oscillatory systems…

State Space Forms and Nonlinearities

Usually: breakdown of system into a significant linear part and an additional nonlinearity



Nonlinear quasi-state space form: $\dot{\boldsymbol{u}} = \boldsymbol{A}\boldsymbol{u} + \boldsymbol{f}_{NL}(\boldsymbol{u}) + \boldsymbol{b}x$ $\qquad y = \boldsymbol{g}(\boldsymbol{u})$

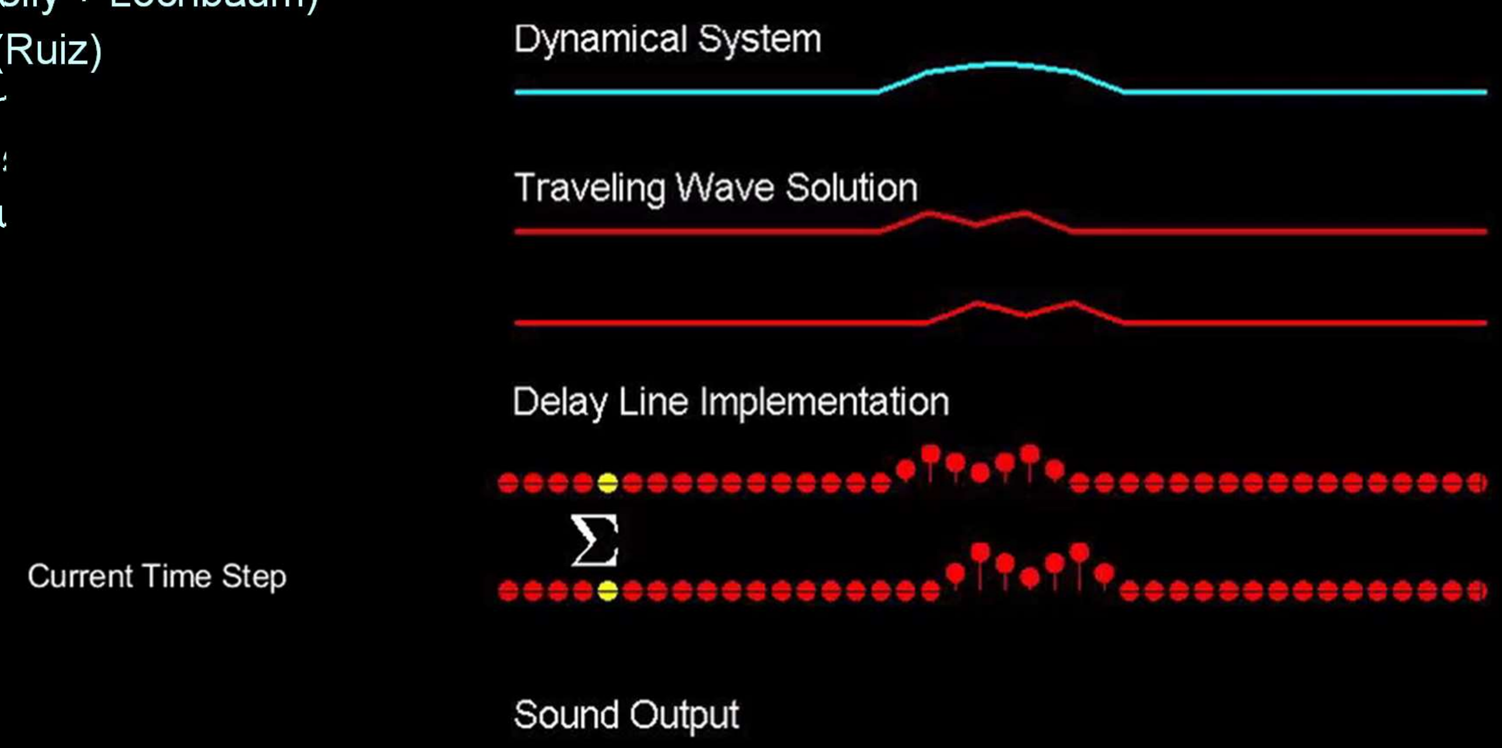
Types of nonlinearity:

- polynomial (usu. cubic): geometric nonlinearity in strings, membranes, plates
- one-sided power laws: collisions (frets, snares, strikes, plucks, reed beating)
- square root (Bernoulli): wind instruments
- signum (Coulomb friction): bowed string instruments

Sometimes differentiable…sometimes not continuously differentiable, or even continuous.

Classic Numerical Simulation Approaches

1962: vocal tract (Kelly + Lochbaum)
1970: FDTD string (Ruiz)
1979: mass spring r
1985: modal synthe
1986: digital wavegu



Limited computational power→ simplified physics…

Advances: Large-scale Instrument Models

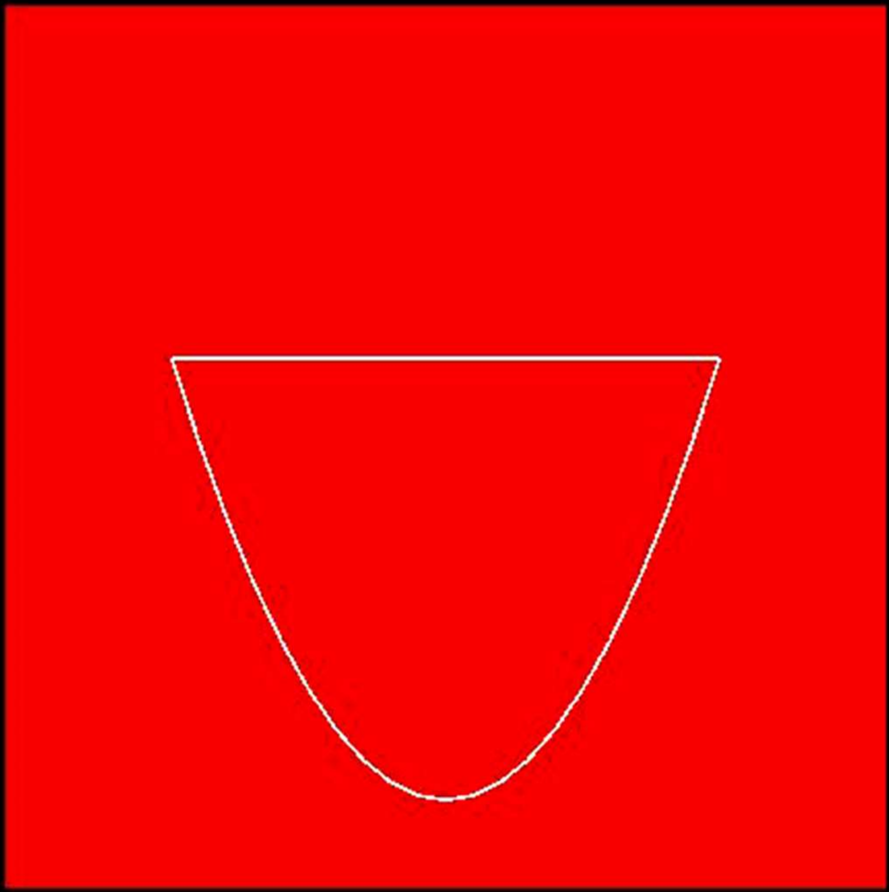
1990s: musical acoustics investigations of complex instruments:

Timpani drum: Rhaouti, Chaigne + Joly, JASA, 1999

Mainstream time-domain methods---
finite difference time domain.

3D acoustic field modelling!

Very large computation…offline!



System Types

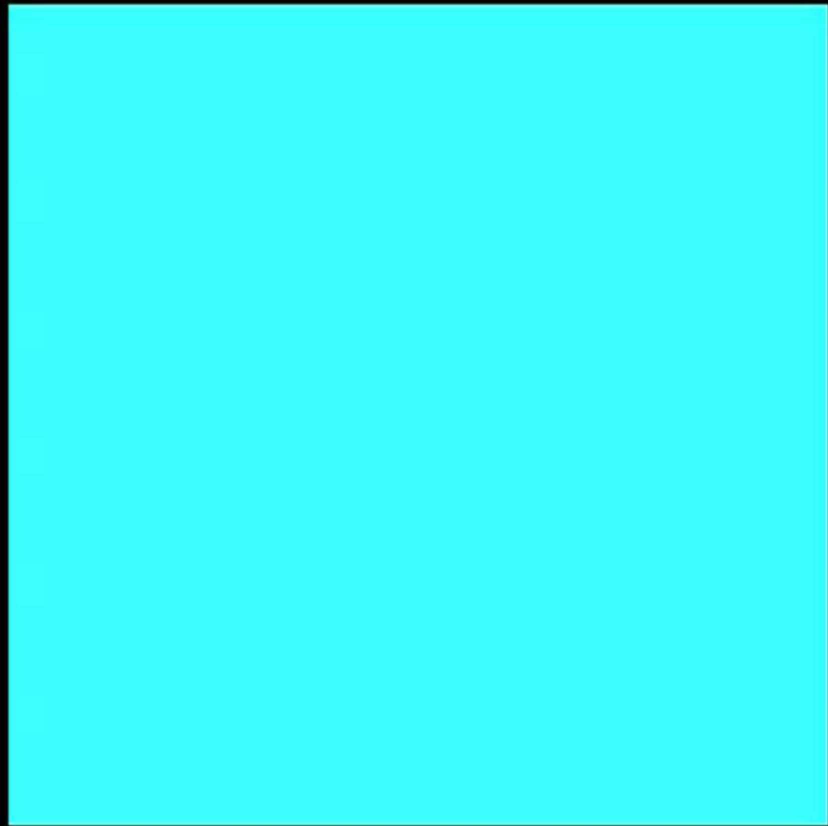
In sound synthesis and virtual acoustics, different manifestations:

Classic: Excitation/resonator interaction

Coupled Systems: Modular instrument construction

Multiphysics: Heterogeneous coupled systems

Complex Systems: Single problems with distinct timescales, or strong nonlinearities



Gallery

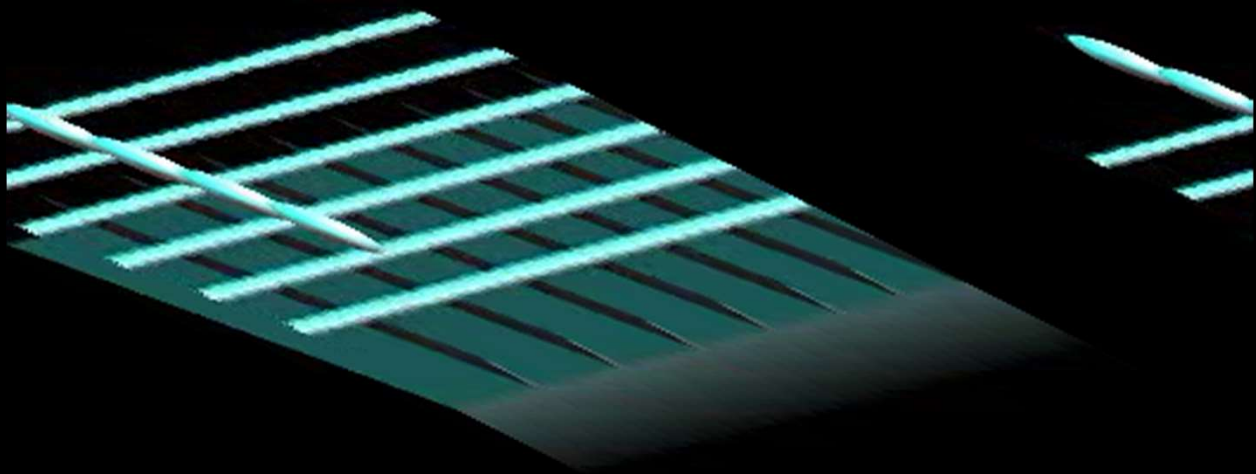
Wind Instruments

Coupled components: Main bore + bypass tubes (valves) or toneholes + excitation mechanism

Easily real time. Typical sounds…

(T. Mudd, “Brass Cultures”, 2020)

Guitars



Coupled components:
strings + frets + fingers
(+ body/acoustic field)

Real time (without body/field)

(G. Sassoon, “Multiverse”, 2021)

Room Modelling

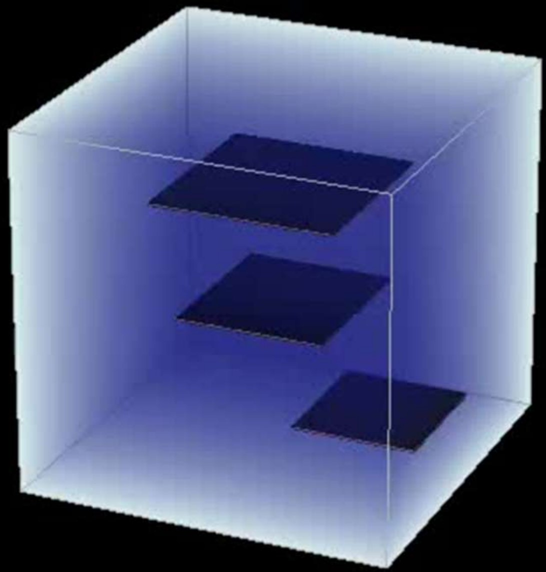
Industry standard: geometrical acoustics (e.g. ray tracing)---neglects diffraction!

Full wave-based solution is complete (audio rate).

At audio rates---a very large computational problem (offline on GPU only)

3D Percussion Instruments: Multiphysics

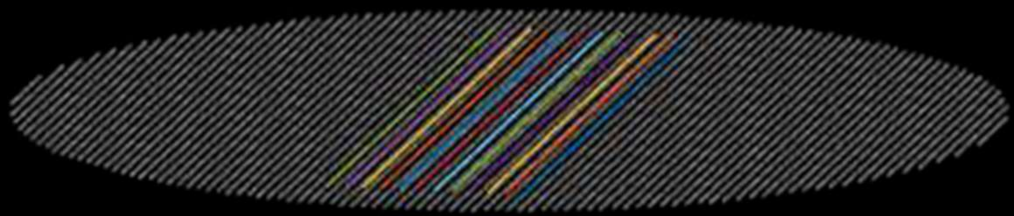
Multiple instruments “embedded” in acoustic field; multichannel (spatialized) output possible



Gongs:

Timpani:

Building a Sound: the Snare Drum



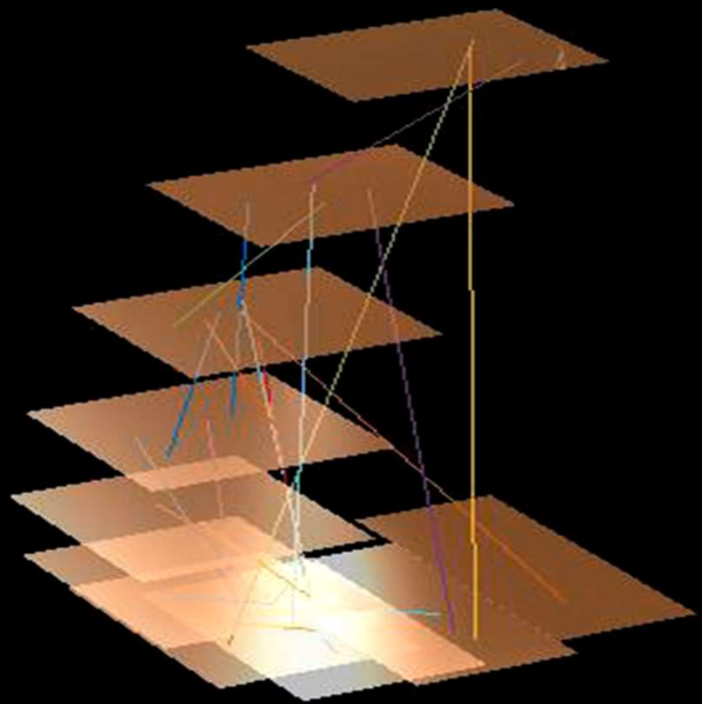
Membrane only

Two snare drums in a large room

Membrane with snares

With drum cavity

Modular Environments

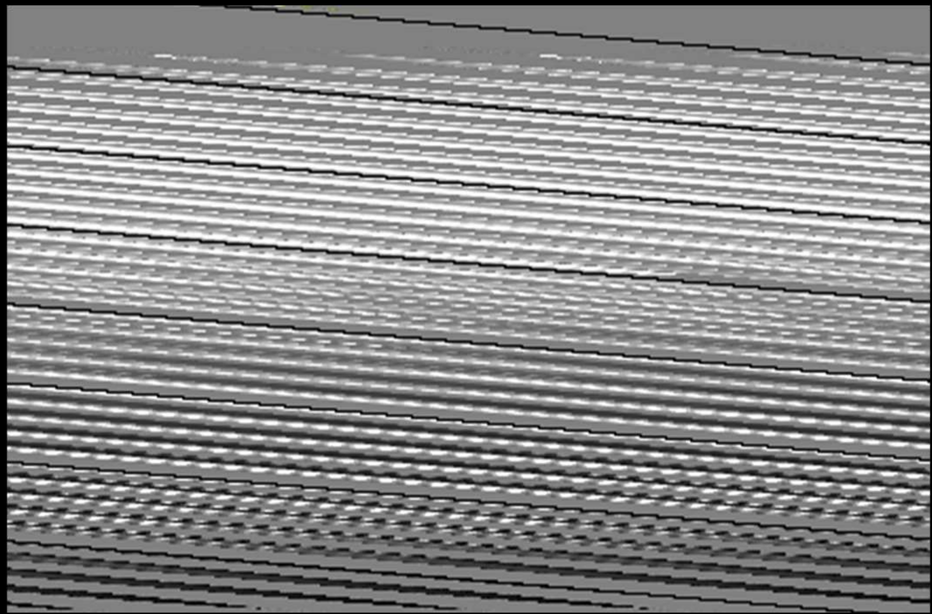


Toolboxes for modular construction of new instruments…

(G. Delap, “Orbit”, 2009)

Electromechanical Reverbs

Classic electromechanical reverbs---such as the spring:



Case Study: String Vibration

String Models: Model parameter space

The simplest nontrivial physical model. Under lossless, linear, unforced conditions, system described by four “physical” parameters:

ρ: mass density   T: tension   r: radius   L: length

Partial differential equation model: describes transverse displacement u(x,t) in time t, and spatial coordinate x:

$$\frac{\partial^2 u}{\partial t^2} = \frac{T}{\rho\pi r^2}\frac{\partial^2 u}{\partial x^2} \qquad x \in [0, L]$$

But: this space is redundant…in fact, need only one “perceptual” parameter:

$$\frac{\partial^2 u}{\partial t^2} = \gamma^2\frac{\partial^2 u}{\partial x^2} \qquad x \in [0,1] \qquad \gamma = \sqrt{\frac{T}{\rho\pi r^2 L^2}}$$

Some “preparatory” work necessary in terms of scaling/nondimensionalization in order to get to the minimal parameter set. Esp. important if using ML methods for fitting parameters!

String Models: Control parameter space

The control aspect also requires a parameterization---much harder!

$$\frac{\partial^2 u}{\partial t^2} = \gamma^2 \frac{\partial^2 u}{\partial x^2} + \delta(x - x_i) f$$

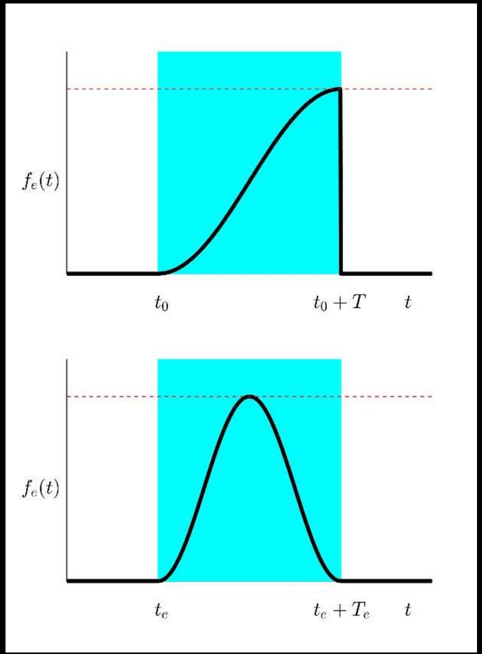
x_i: excitation location
f(t): excitation function

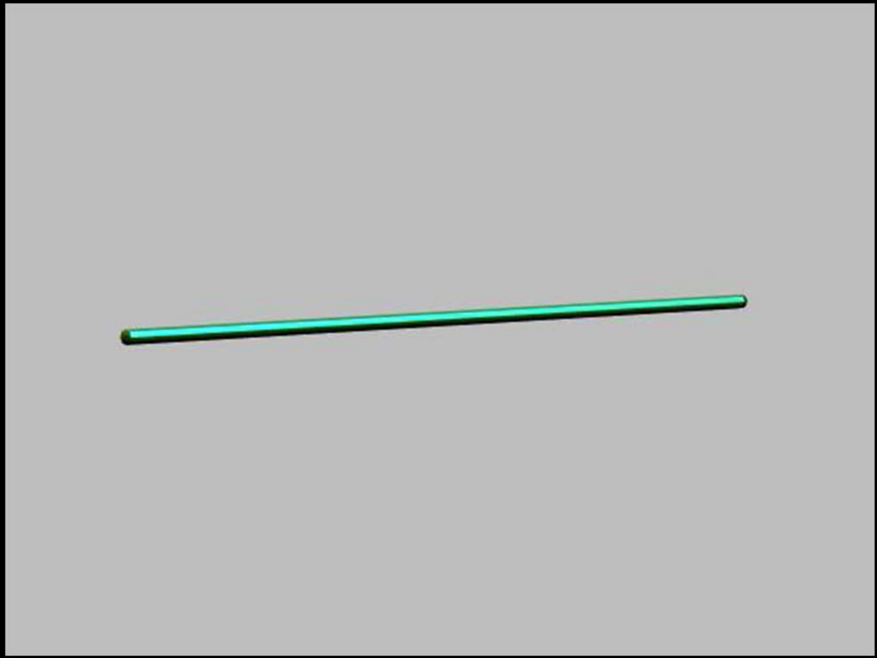
Now: an entire function is required.

Simple parameterized forms:

Pluck:

Strike:

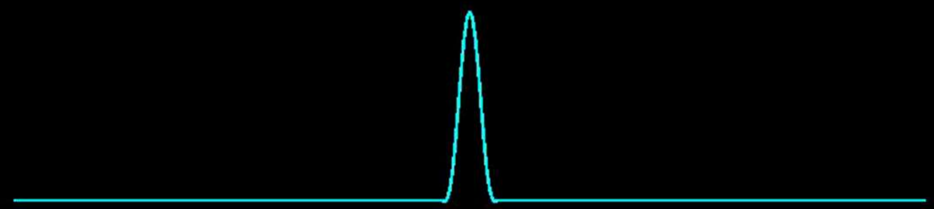




Adding Stiffness

$$\frac{\partial^2 u}{\partial t^2} = \gamma^2 \frac{\partial^2 u}{\partial x^2}$$

β: stiffness



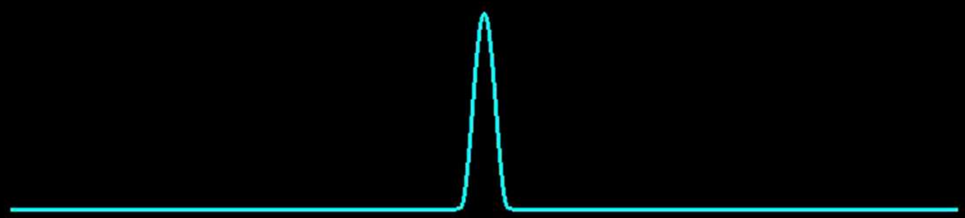
pitch and timbre both change…

Losses

$$\frac{\partial^2 u}{\partial t^2} = \gamma^2 \frac{\partial^2 u}{\partial x^2} - \beta^2 \frac{\partial^4 u}{\partial x^4}$$

σ_0: loss parameter

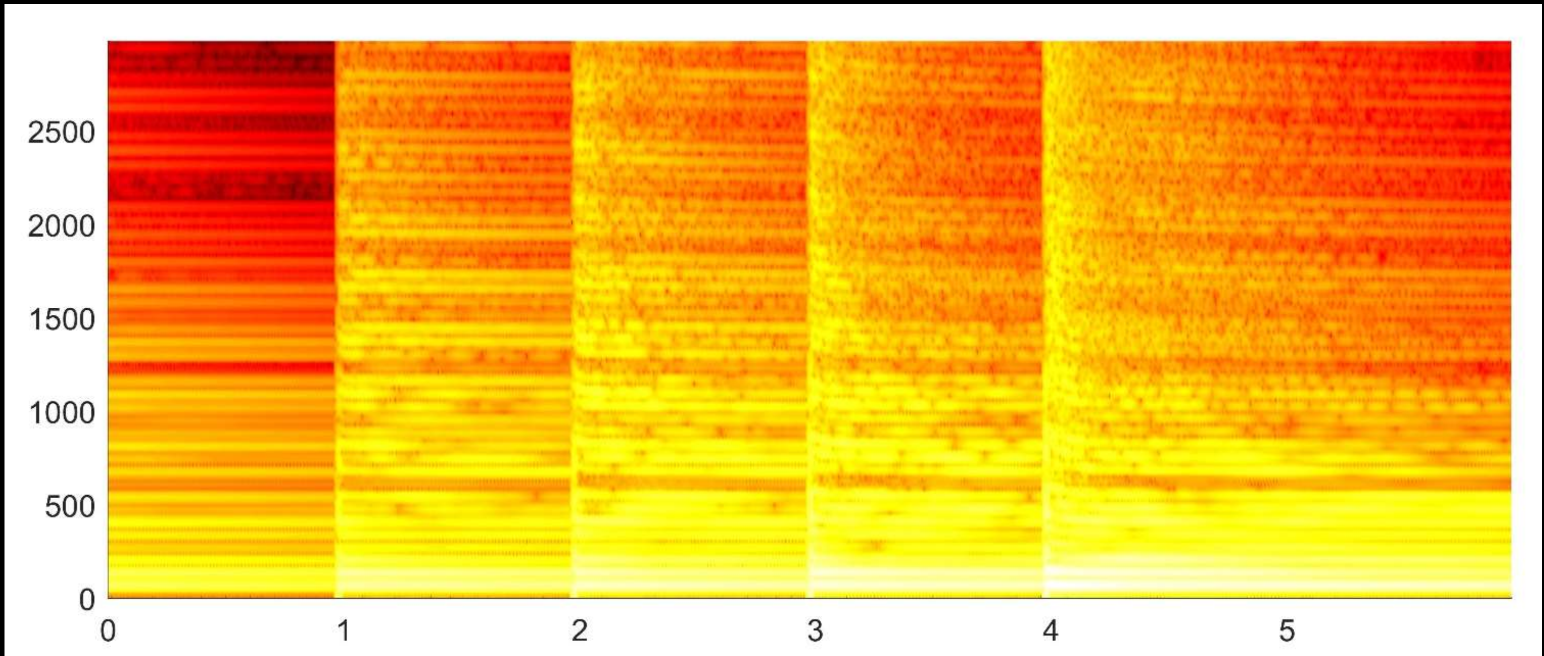
σ_1: frequency-dependent loss parameter



Example: nonlinear string vibration

$$\frac{\partial^2 u}{\partial t^2} = \gamma^2 \frac{\partial^2 u}{\partial x^2} - \beta^2 \frac{\partial^4 u}{\partial x^4} - \sigma_0 \frac{\partial u}{\partial t} + \sigma_1 \frac{\partial^3 u}{\partial t \partial x^2} + K^2 \left(\frac{\partial u}{\partial x}\right)^2 \frac{\partial^2 u}{\partial x^2}$$

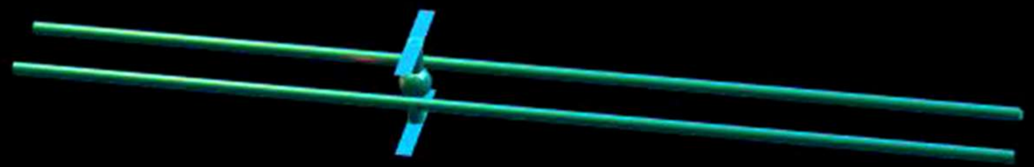
amplitude-dependent
nonlinearity



linear

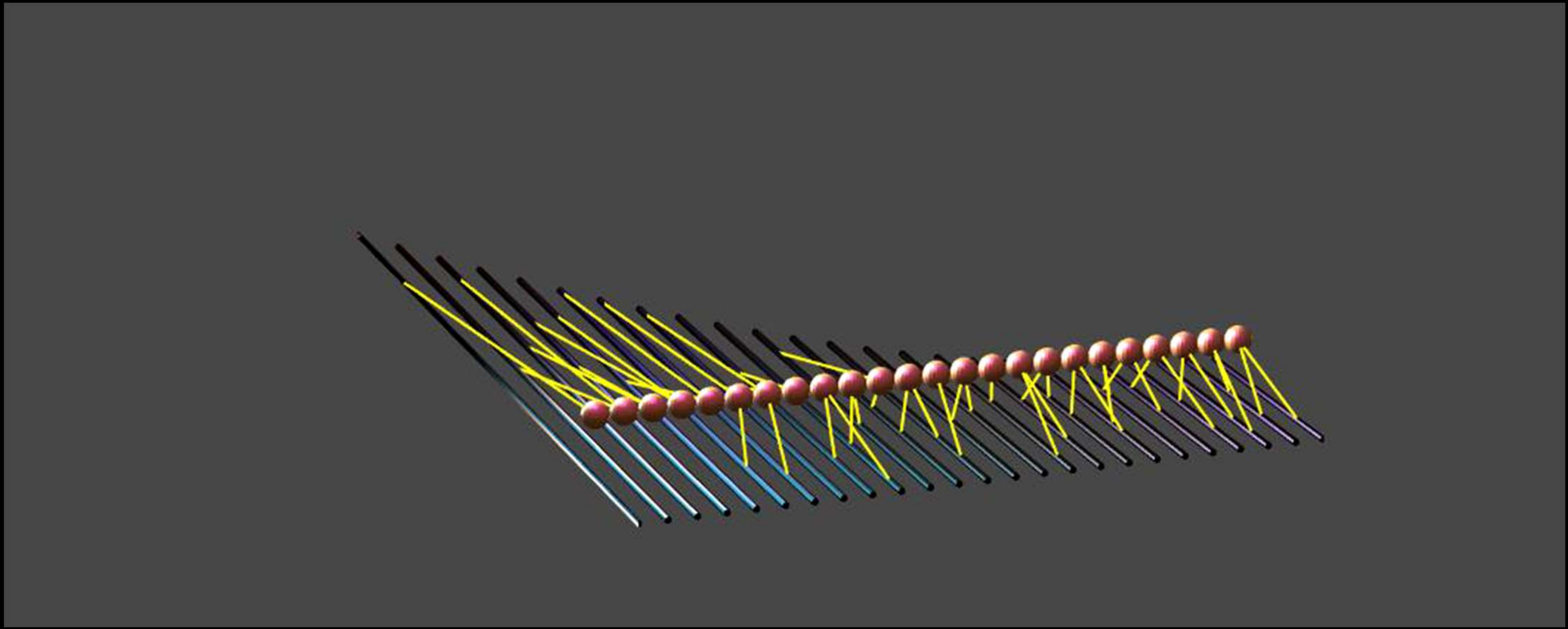
nonlinear

“With just one musician, you can really do an unlimited number of things on the inside of the piano if you have at your disposal an exploded keyboard.” (John Cage)



2-3 additional parameters required here…

Complete String Instruments



Parameter space: >200

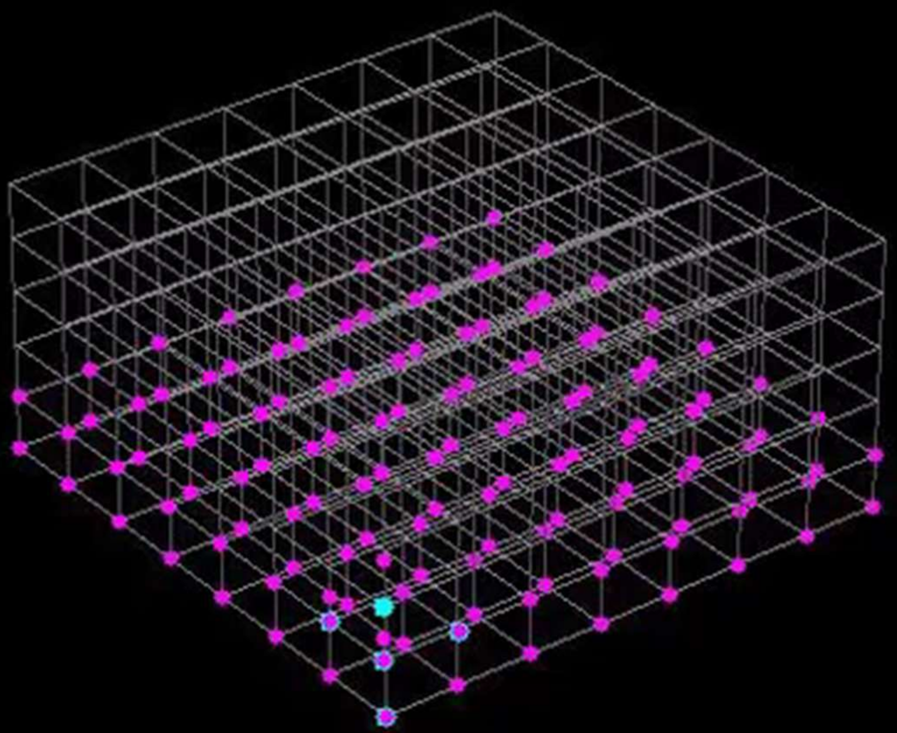

IN CONTROL
1
2
3
4
5
6
7
8
Hold
novation
LAUNCHKEY 25
41
TRACK
MIDI Channel
OCTAVE
Transpose
Device
Pan
Send A
Send B
Scene
PITCH
MODULATION

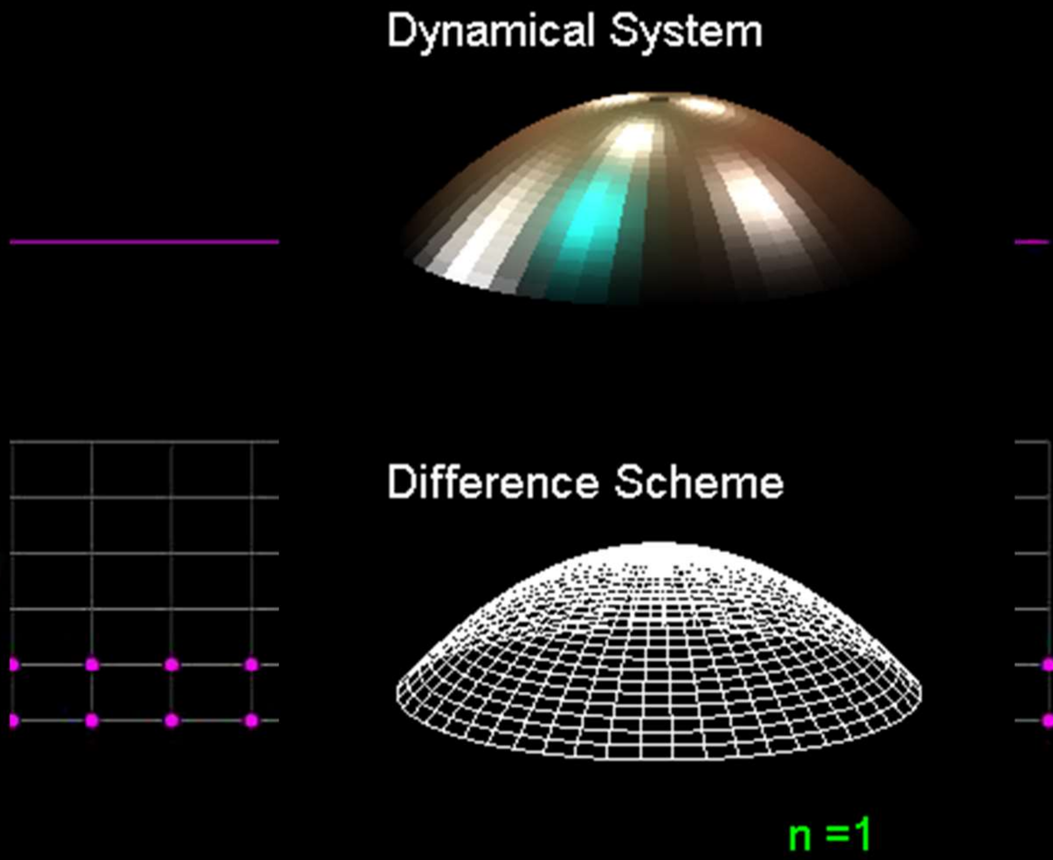
Computing

Time-stepping Methods

Time stepping methods: the classic way of proceeding.

Many different approaches in different contexts:





Run-time Loop

Suppose we have a linear system (very unrealistic!). Any physical modeling synthesis method will look like an audio rate recursion (state space):

```
for n=1:final_time_step
      x = A*x_last+b*u;
      x = f(x_last,u);
      f(x,x_last,u)=0;
      y = c*x;

      x_last = x;
end
```

x: the current "state."
u: input
A: the "system."
b: selects input location
c: selects output location

A key parameter: state size or # degrees of freedom N…determines computational cost!

A, b, c generally extremely sparse (comes from "local" nature of physical laws)

Nonlinear case: update line above replaced by nonlinear algebraic equations to solve…

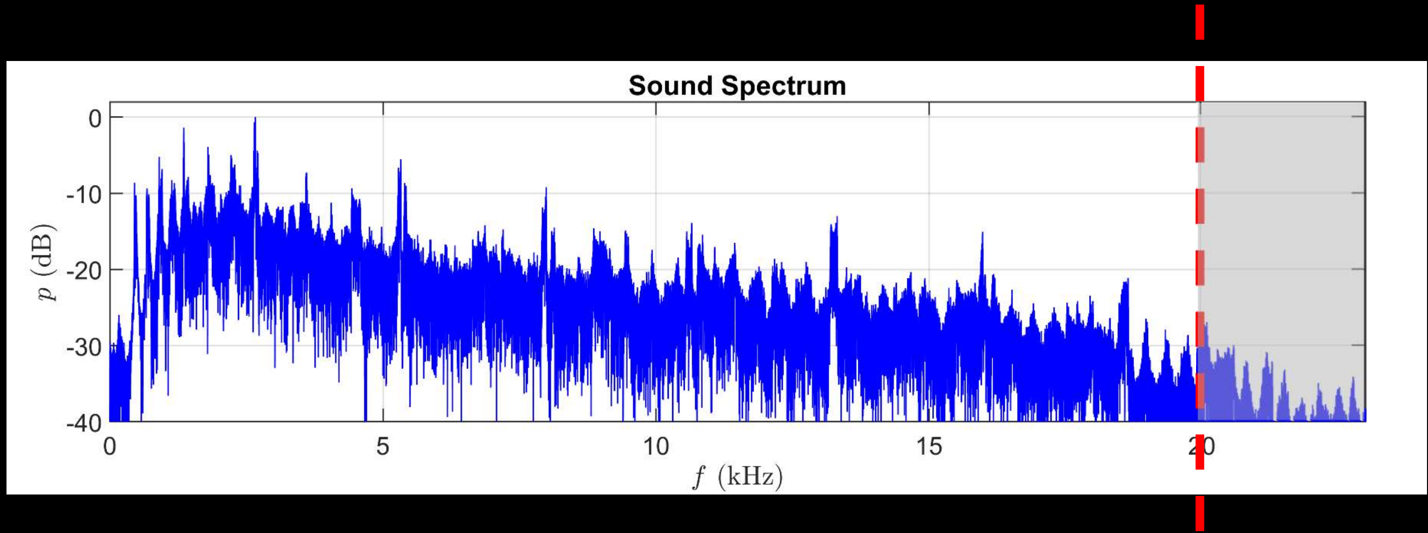
Sometimes---can arrive at an "explicit" nonlinear update; can interpret as a large, possibly nonlinear IIR filtering operation…with major stability considerations!

Audio Rate Simulation

Need to capture all information up to f_c = 20 kHz

Bound on time step T:

$$T < 1/2f_c = 25\ \mu s$$



f_c = 20 kHz

But: computational cost generally scales as $(1/T)^p$ for some integer p = 2,3,4…

Thus, ideally:

$$T \approx 1/2f_c$$

Some standard notions in simulation need to be viewed in this light…more later!

Computational Requirements and Factors

Many different factors at play when grappling with the question of computational efficiency:

State memory
Additional memory (coefficients, connectivity graphs)

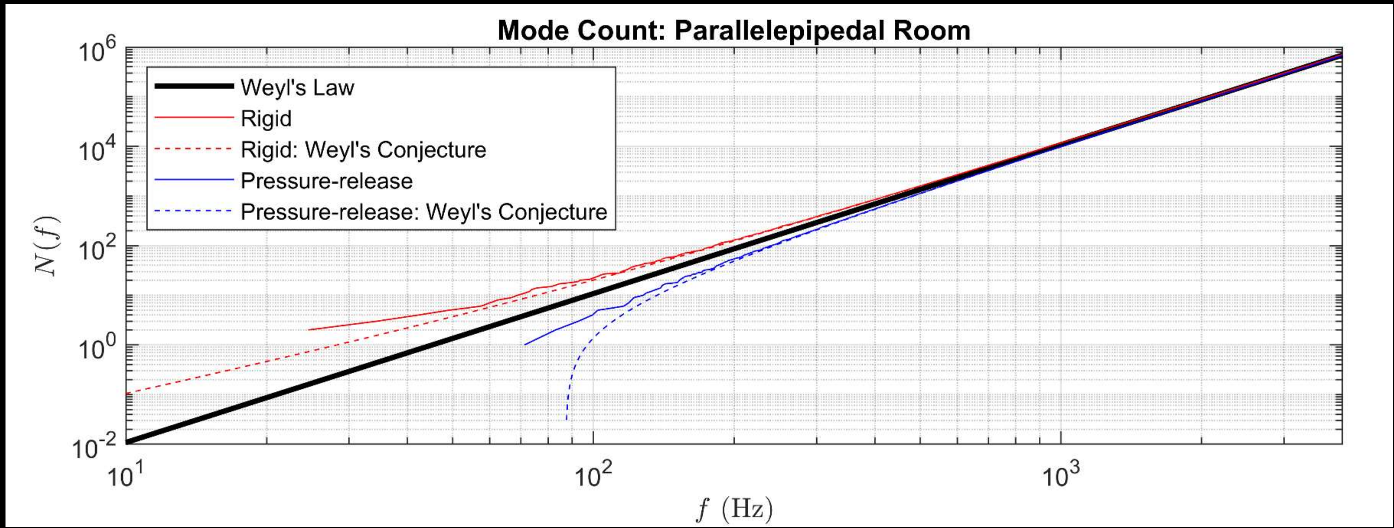
Raw operation count
Linear systems: Sparsity? Other structure to be exploited?
Nonlinear systems: iterative methods: stopping criterion + bit depth for audio?
Parallelizability
Single vs. double precision?

Weyl's Law (1911)

A means of counting the number of degrees of freedom (twice number of "modes") for a given system.

E.g., room, volume V m^3, wave speed c m/s.

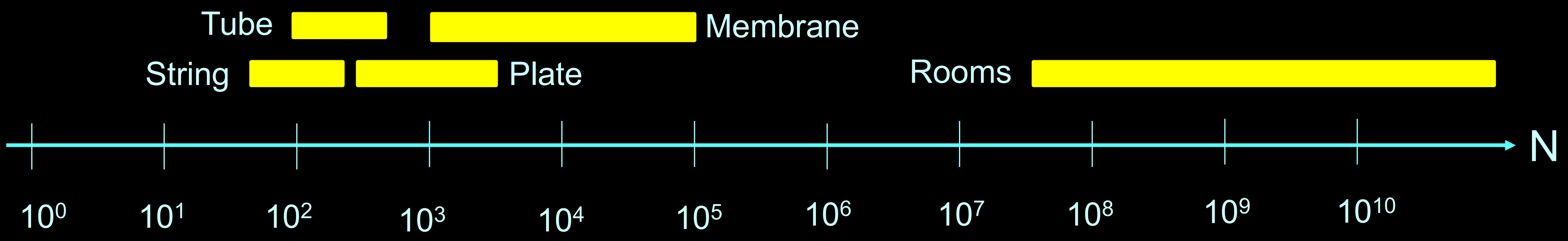
Number of modes N of frequency < f_c: $$N(f_c) = \frac{4\pi V f_c^3}{3c^3}$$ **Geometry independent**



BCs: lead to higher order effects (dep. on bounding area A): $$N(f_c) = \frac{4\pi V f_c^3}{3c^3} \pm \frac{\pi A f_c^2}{4c^2}$$

Audio-rate Simulation: Problem Sizes

Weyl's Law: tells us # DOF N required, and thus minimal state memory requirement. Highly system/dimension dependent! To get to audio rate, f_c = 20 kHz



Implications for neural audio rate synthesis…

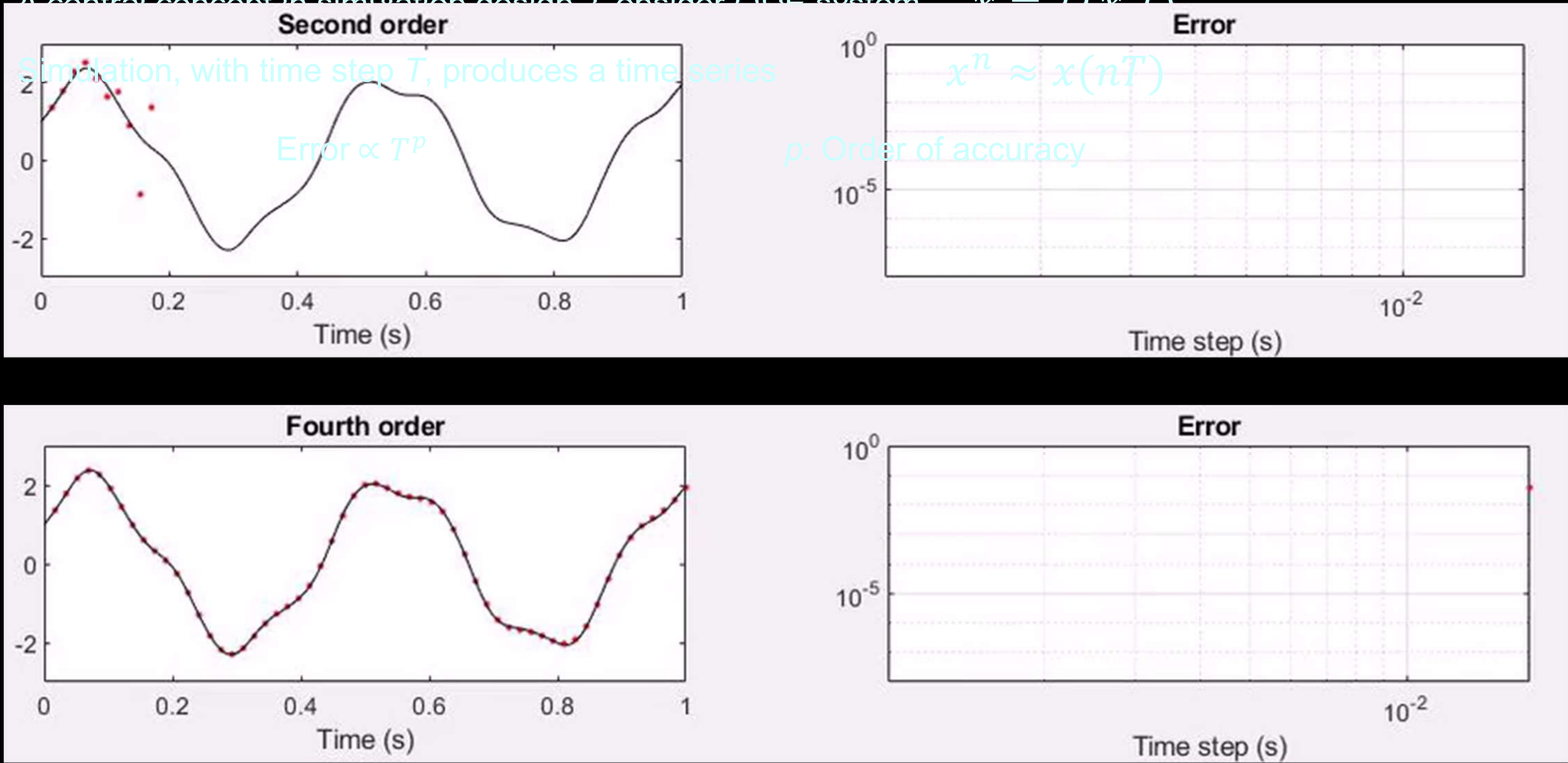
Order of Accuracy

A central concept in simulation design. Consider ODE system $\dot{x} = f(x, t)$

Simulation, with time step T, produces a time series $x^n \approx x(nT)$

Error $\propto T^p$

p: Order of accuracy



Example: Simple Harmonic Oscillator

The most basic oscillatory system: the Simple Harmonic Oscillator (SHO): $\dot{x} = 2\pi f_0 \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x$

Solutions: pure sinusoids at frequency f_0.

Audio rate simulations with f_s = 1/T = 44.1 kHz:

	Exact	2^{nd} order accurate method (Verlet)	4^{th} order accurate method (classic Runge Kutta)
f_0 = 1 kHz:			
f_0 = 4 kHz:			

Major implications for low-loss systems (most musical instruments and rooms…)

Numerical Instability

A major problem in audio-rate acoustic simulation…low loss, strong nonlinearities, long duration simulations.

Linear systems: spurious exponential solution growth:

Nonlinear systems: consider simple cubic nonlinear oscillator

$$\ddot{u} = -u^3 \xrightarrow[\text{Verlet}]{} u^{n+1} = 2u^n - u^{n-1} - T^2(u^n)^3$$

Need a suitable robust design strategy…

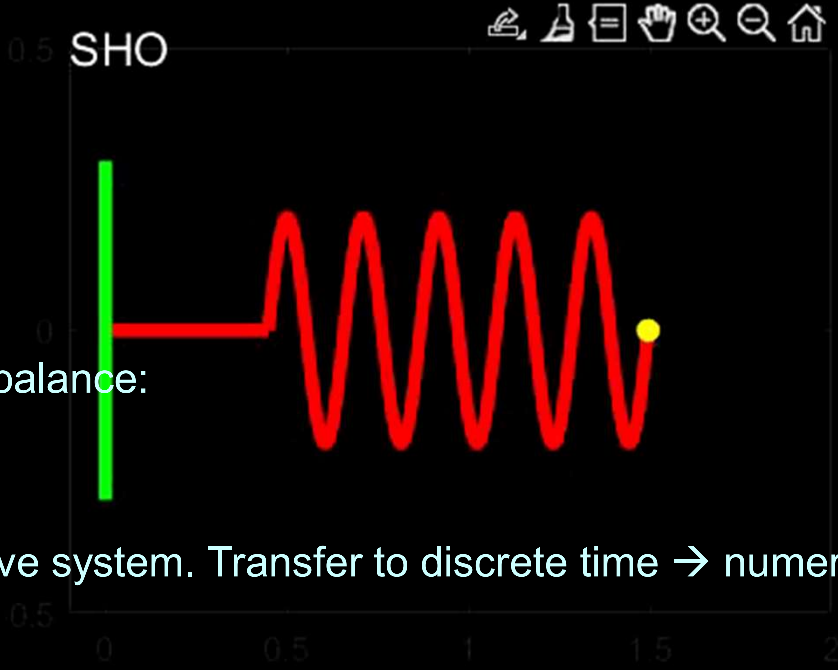
Geometrical Numerical Integration

Structure-preserving numerical methods (symplectic, energy-conserving, etc.)

Numerical invariants are included by construction. Usually employed for strictly conservative systems

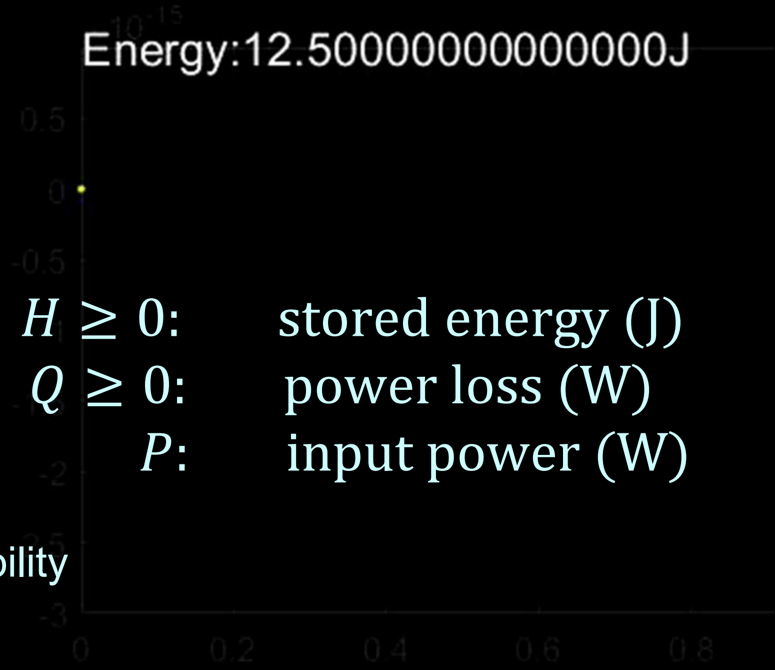
Example: SHO $$\dot{x} = 2\pi f_0 \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x \quad \rightarrow \quad H = \frac{1}{2}||x||^2 = \text{constant}$$

Can build this conservation property into a numerical method:



Can go

Power balance:

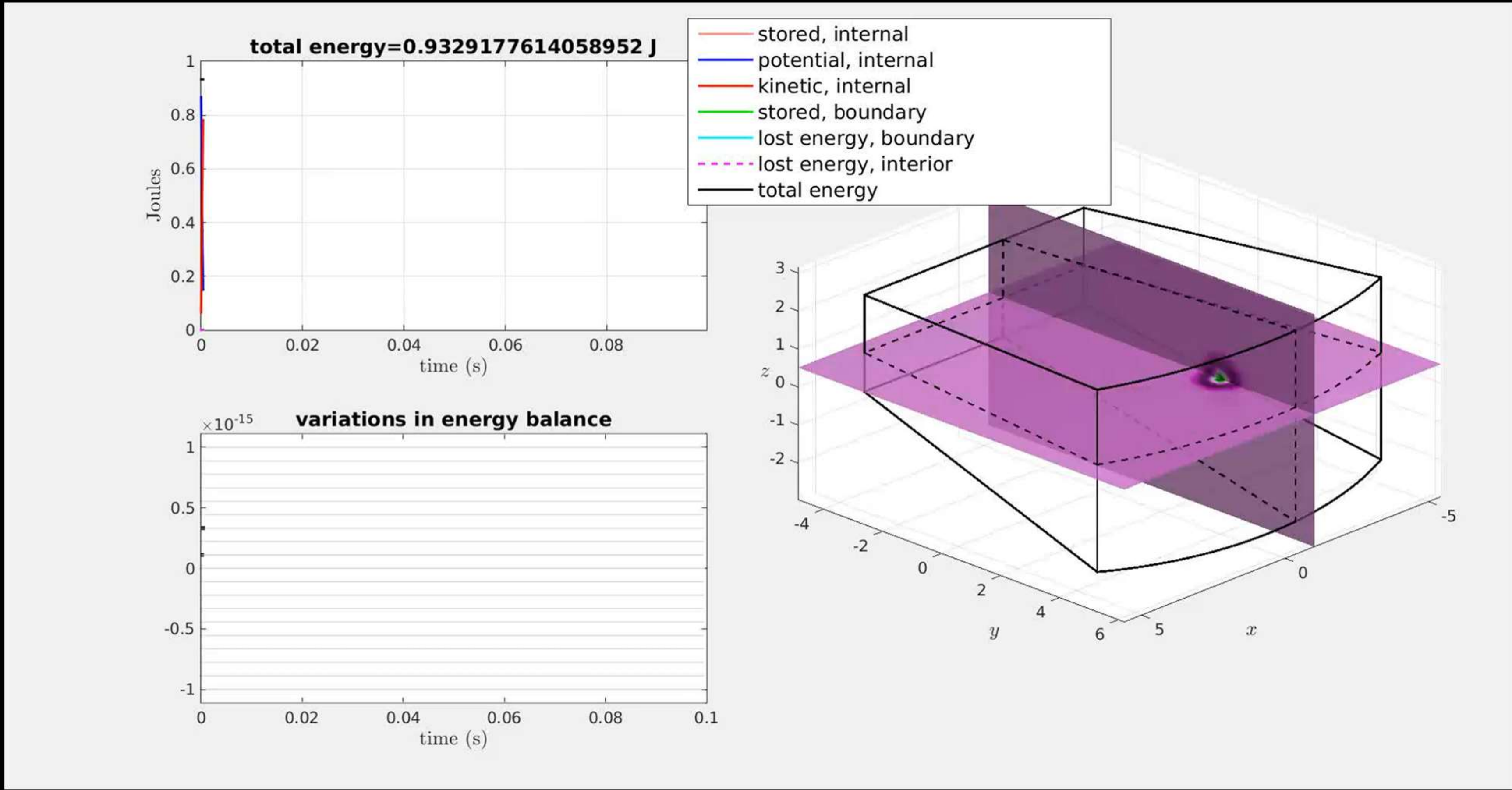


$$\begin{aligned} H \geq 0 &: \quad \text{stored energy (J)} \\ Q \geq 0 &: \quad \text{power loss (W)} \\ P &: \quad \text{input power (W)} \end{aligned}$$

A passive system. Transfer to discrete time → numerical stability

Energy Conservation and Numerical Stability

A very robust solution: numerical energy conservation to machine accuracy allows stable behaviour for a wide range of complex systems. Schemes are “structurally dissipative”. For room acoustics, e.g.,

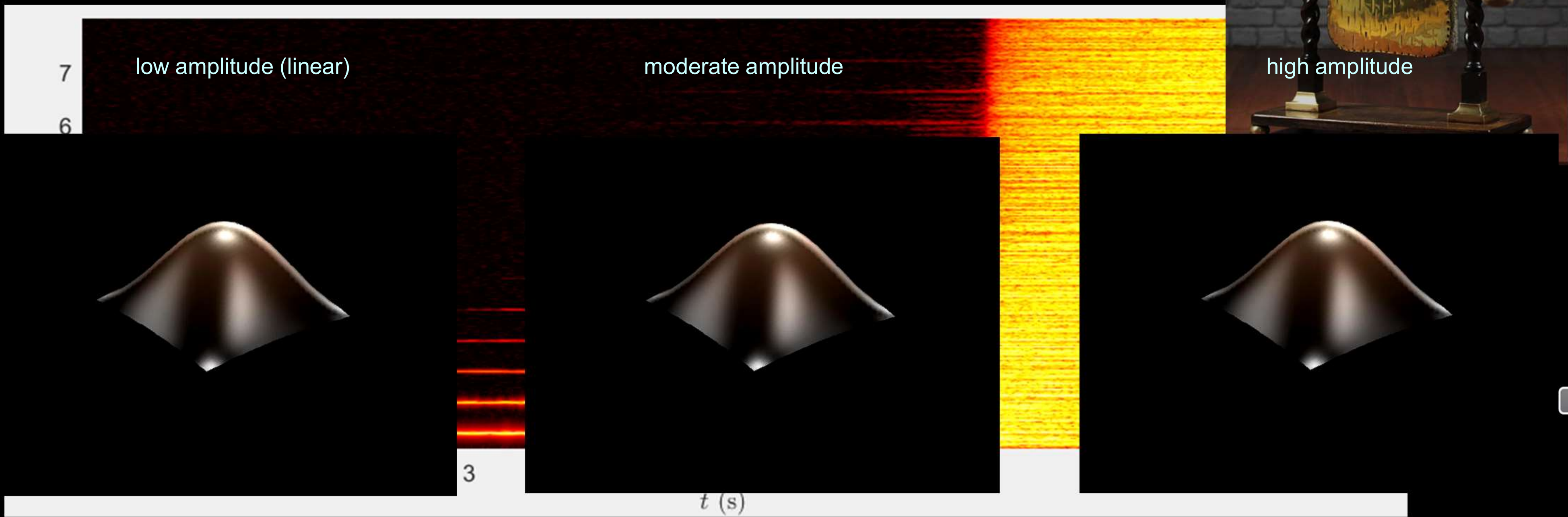


Nonlinear systems: Plate Vibration

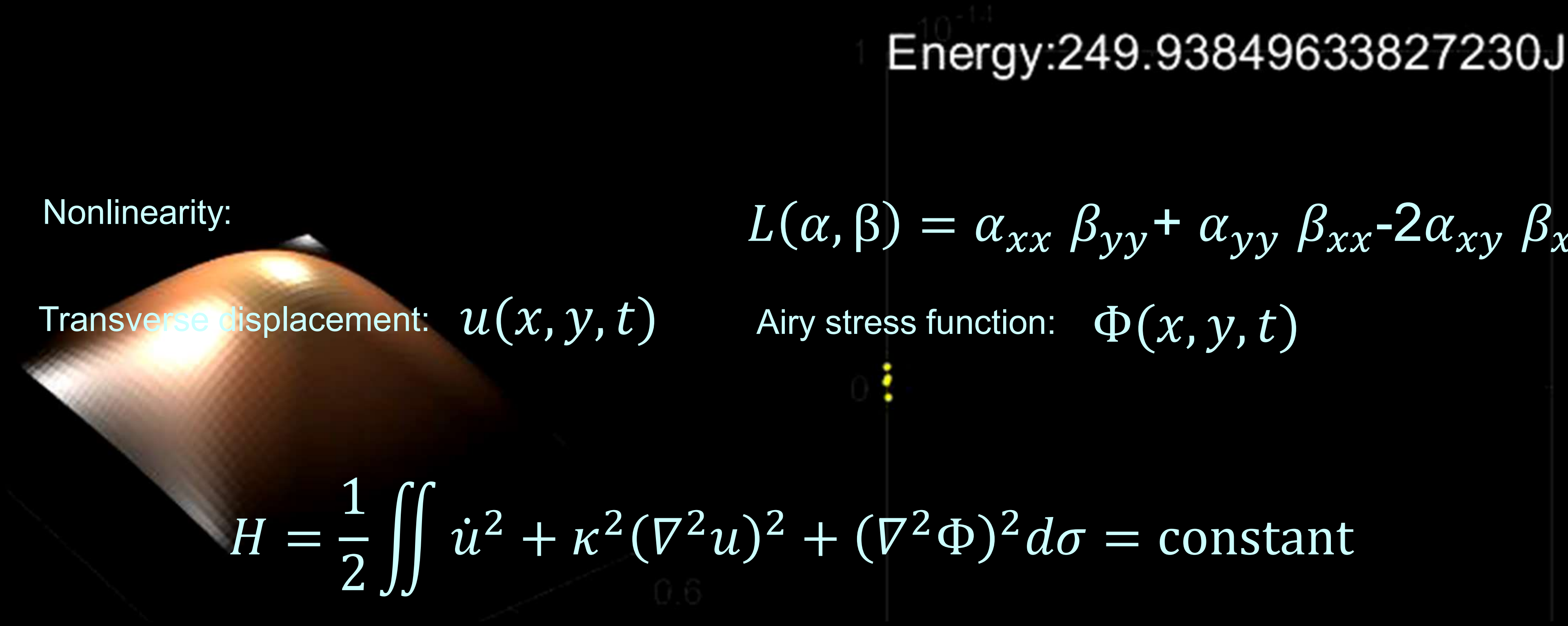
Thin metallic structures: the basis for many percussion instruments (gongs, cymbals, tamtams, etc.)

At high vibration amplitudes: very strong nonlinear effects (crashes etc.)

Nonlinearity very different from in the case of electronic circuits…



The Föppl-von Kármán Equations



Nonlinearity: $L(\alpha, \beta) = \alpha_{xx}\, \beta_{yy} + \alpha_{yy}\, \beta_{xx} - 2\alpha_{xy}\, \beta_{xy}$

Transverse displacement: $u(x, y, t)$

Airy stress function: $\Phi(x, y, t)$

$$H = \frac{1}{2} \iint \dot{u}^2 + \kappa^2 (\nabla^2 u)^2 + (\nabla^2 \Phi)^2 d\sigma = \text{constant}$$

Can use our energy conservation framework → numerically stable method

Perspectives

Physical modelling synthesis and audio effects---possible to run large models in real time now.

Specialised designs (passive) necessary to cope with strong nonlinear effects

For future “black box” modelling, some useful constraints appear: state size, as well as “passive” nature of recursive update.