

Piano String Modeling

MUS420 Supplement Modal Synthesis of a Piano String

Stefan Bilbao and Julius O. Smith III (jos@ccrma.stanford.edu)
Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

February 5, 2019

Outline:

- Piano String Modeling
- Finite Difference Scheme for the Piano String
- Listening to the String
- State Space Formulation
- Diagonalizing to Modal Coordinates
- Adding the Nonlinear Hammer
- Features of Modal State-Space Coordinates

1

Piano String Model, Continued

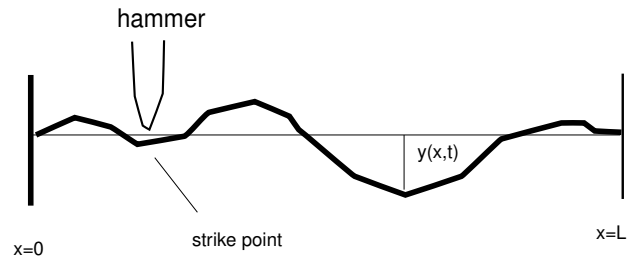
$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2} - \epsilon c^2 L^2 \frac{\partial^4 y}{\partial x^4} - 2b_1 \frac{\partial y}{\partial t} + 2b_3 \frac{\partial^3 y}{\partial t^3} + f(x, y, t)$$

- $y(t, x)$ represents *transverse displacement* along the string at time t seconds and position m meters
- L is the string *length* in meters
- The PDE takes into account
 - *dispersion* (fourth space derivative)
 - frequency-dependent *loss* from 3rd time derivative
 - *hammer excitation* denoted by $f(x, y, t)$
- A derivation of the wave equation for stiff strings is given in Morse 1948 and in many acoustics texts. (Cremer 1984 is especially good.) (See the Course Bibliography¹ for full citations.)

¹<http://ccrma.stanford.edu/~jos/refs420/>

3

Physical picture:



Chaigne and Askenfelt (JASA, Feb. 1995) suggested the following partial differential equation (PDE) to model the piano string:

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2} - \epsilon c^2 L^2 \frac{\partial^4 y}{\partial x^4} - 2b_1 \frac{\partial y}{\partial t} + 2b_3 \frac{\partial^3 y}{\partial t^3} + f(x, y, t)$$

In addition, we must specify

- *boundary conditions*
- *initial conditions* (usually zero position and velocity).

2

Second-Order Finite Difference Schemes

The simplest (and traditional) way of discretizing the 1D wave equation is by replacing first derivatives by first-order differences

$$\left. \frac{\partial y}{\partial t} \right|_{x=mX, t=nT} \simeq \frac{y_m^n - y_m^{n-1}}{T}$$

$$\left. \frac{\partial y}{\partial x} \right|_{x=mX, t=nT} \simeq \frac{y_m^n - y_{m-1}^n}{X}$$

and second derivatives by second-order differences

$$\left. \frac{\partial^2 y}{\partial t^2} \right|_{x=mX, t=nT} \simeq \frac{y_m^{n-1} - 2y_m^n + y_m^{n+1}}{T^2}$$

$$\left. \frac{\partial^2 y}{\partial x^2} \right|_{x=mX, t=nT} \simeq \frac{y_{m-1}^n - 2y_m^n + y_{m+1}^n}{X^2}$$

and so on, where $y_m^n \triangleq y(nT, mX)$ are the *grid variables*. Note that we have *uniformly sampled* the time-space plane, with timestep T (the *temporal sampling interval*) and space step X (the *spatial sampling interval*).

4

FDS for the Ideal String

Consider first the *ideal* (lossless, dispersionless, unforced) string wave equation:

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2}$$

Replacing the second-derivatives by their finite-difference approximations gives

$$y_m^{n-1} - 2y_m^n + y_m^{n+1} = \frac{c^2 T^2}{X^2} (y_{m-1}^n - 2y_m^n + y_{m+1}^n)$$

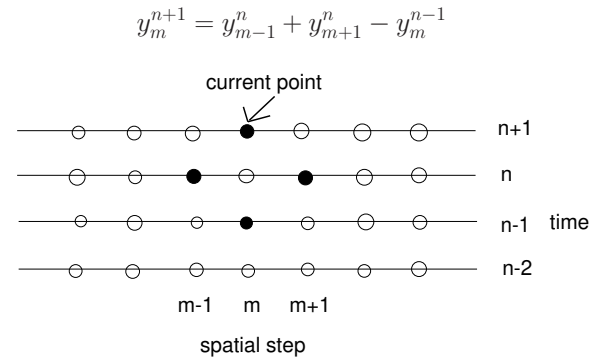
If we choose $X = cT$ (which is most natural physically), the equation reduces further to

$$y_m^{n+1} = y_{m-1}^n + y_{m+1}^n - y_m^{n-1}$$

Let's examine this recursion on the time-space grid, assuming for the moment no boundary conditions:

5

Time-Space Grid for the Second-Order Ideal String FDS

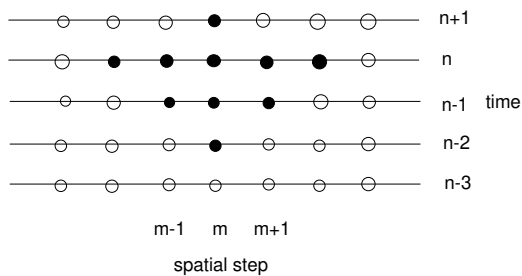


- Grid variable at “current” point depends on value at *two* previous time steps (a *second order* scheme in time). Our initial conditions must therefore specify $y(n, m)$ for all m at times $n = 0$ and $n = 1$.
- Grid variable at “current” point depends on values at adjacent locations on the string (at previous time).
- Difference scheme is *explicit* (thus *parallelizable*); that is, each grid variable at time $n + 1$ depends only on grid variables at previous time instants.

6

Piano String Finite Difference Scheme

The Chaigne and Askenfelt PDE is discretized in space and time to yield a recursive difference scheme with the following space-time “stencil”:



- 3rd order in time, 4th order in space
- *Explicit*, enabling recursive time updates

The FDS can be written in matrix form as

$$\mathbf{y}_n = \mathbf{A}_1 \mathbf{y}_{n-1} + \mathbf{A}_2 \mathbf{y}_{n-2} + \mathbf{A}_3 \mathbf{y}_{n-3} + \mathbf{g} f_n(\mathbf{y}_{n-1}[i_h], y_{n-1}^h)$$

where

- The column vector \mathbf{y}_n contains the displacements of the N nodes at time step n , i.e., $\mathbf{y}_n[i] = y_i^n$
- $\mathbf{A}_1 \mathbf{A}_2$ and \mathbf{A}_3 are N by N matrices

7

Piano String FDS, Continued

We have

$$\mathbf{y}_n = \mathbf{A}_1 \mathbf{y}_{n-1} + \mathbf{A}_2 \mathbf{y}_{n-2} + \mathbf{A}_3 \mathbf{y}_{n-3} + \mathbf{g} f_n(\mathbf{y}_{n-1}[i_h], y_{n-1}^h)$$

The string displacement is updated in time by three matrix-vector products, plus a *forcing term* $\mathbf{g} f_n(\mathbf{y}_{n-1}[i_h], y_{n-1}^h)$:

- The length $N \times 1$ matrix \mathbf{g} represents the *shape of the hammer excitation* over the length of the string. The nonzero elements correspond to the width of the hammer relative to that of the string. For greatest accuracy, it should be time varying, but it may be approximated by a constant shape.
- The element $\mathbf{y}_n[i_h]$ is the string displacement sample closest to the *hammer position*.
- The scalar function y_n^h denotes *hammer position* at time n .
- The scalar function $f_n(\mathbf{y}_n[i_h], y_n^h)$ sets the amplitude of the hammer force distribution across position at time n .

8

- A secondary effect is that the modal frequencies ω_i are slightly shifted.
- Another secondary effect of the third-order time derivative is the introduction of N real rapidly decaying modes, characterized by the ϵ_i . These eigenvalues are typically *negligible*.

We can reduce the size of the system to $2N$, and further change coordinates so that the matrix is made up of N 2 by 2 real “modal” blocks as follows:

$$\mathbf{D}'' = \begin{bmatrix} \begin{bmatrix} 0 & 1 \\ -|\lambda_1|^2 & 2\text{Re}(\lambda_1) \end{bmatrix} & & & \\ & \begin{bmatrix} 0 & 1 \\ -|\lambda_2|^2 & 2\text{Re}(\lambda_2) \end{bmatrix} & & \\ & & \dots & \\ & & & \begin{bmatrix} 0 & 1 \\ -|\lambda_N|^2 & 2\text{Re}(\lambda_N) \end{bmatrix} \end{bmatrix}$$

13

- Extends simply to spatially varying media (not Fourier based)
- Can also be used for *implicit* finite difference schemes
- Relatively efficient when modes are *inharmonic* (bells, gongs, mallet percussion)
- When mode frequencies are nearly *harmonic* (strings, woodwinds, brasses) *digital waveguide models are more efficient* (we’ll take them up later).

15

Features of Modal State-Space Coordinates

- Input and output locations must be known and fixed
- The i th mode of the system is simulated *in isolation*

$$\begin{aligned} \mathbf{x}_n[i] &= \lambda_i \mathbf{x}_{n-1}[i] + b'_i u_n \\ y_n &= \mathbf{C}' \mathbf{x}_n \end{aligned}$$

Notes:

- the input gain b'_i specifies *how the input signal u_n excites mode i*
- the output state-gain-vector \mathbf{C}' specifies *how all the modes add up (weighted) to form the output*
- The mode-input gain vector $\mathbf{B}' = [b'_1, \dots, b'_N]$ *changes if the physical input is moved*
- \mathbf{C}' *changes if the physical output is moved*
- The system modes *do not change when the input or output are moved*
- Dispersion modeling (mode tuning) is much easier than in a finite difference approximation
- Frequency-dependent damping is also much easier
- Extends to higher order models (time and/or space)

14