WAVETABLE SYNTHESIS

A basic synthesis technique

Building Waveforms

- This is presented more or less as a "formula":

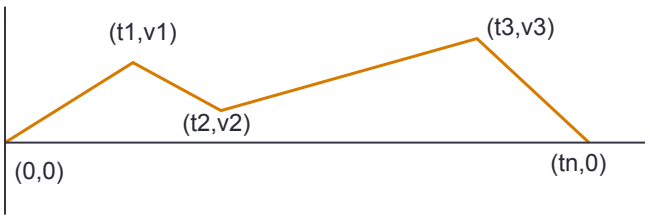
```
define variable *table* =
  sim(0.5 * build-harmonic(1.0, 2048),
      0.25 * build-harmonic(2.0, 2048),
      0.125 * build-harmonic(3.0, 2048),
      0.062 * build-harmonic(4.0, 2048))
set *table* = list(*table*, hz-to-step(1), #t)
```


Using Waveforms

- **`*table*`** is a global – if you set it, **osc** will use it:
  - **`set *table* =`** …
  - **`play osc(c4)`**
- Or, set another global and pass it to **osc**
  - **`set *mytable* =`** …
  - **`play osc(c4, 1.0, *mytable*)`**

Piece-wise Linear Functions: PWL

- Common for control functions.
- By default, produces low, control sample rate.
- **`pwl(`*`t1, v1, t2, v2, …, tn`*`)`**



Variants of PWL

- **`pwlv(v0, t1, v1, t2, v2, …, tn, vn)`**
  - for non-zero starting and ending points
- **`pwe(t1, v1, t2, l2, …, tn)`**
  - exponential interpolation, $v_i > 0$
- **`pwlr(i1, v1, i2, v2, …, in)`**
  - relative intervals rather than absolute times
- See manual for more variants & combinations

Basic Wavetable Synthesis

- Build a wavetable with the harmonics you want
- Use an oscillator (osc) to generate a tone with these harmonics
- Multiply by an envelope (e.g. pwl) to control the amplitude contour.

- Advantages: simple, efficient, direct control
- Disadvantages: spectrum (strength of harmonics) does not change with pitch or time as in most acoustic instruments.

