Matrices

Matrices are two-dimensions containers to keep samples (sounds, envelopes, algorithmic patterns, images, etc.) in memory and access them quickly.

NewMatrix

class **NewMatrix**(*width*, *height*, *init=None*) [source]

Create a new matrix ready for recording.

Optionally, the matrix can be filled with the contents of the *init* parameter.

See `MatrixRec` to write samples in the matrix.

Parent: `PyoMatrixObject`

Args: width: int

Desired matrix width in samples.

height: int

Desired matrix height in samples.

init: list of list of floats, optional

Initial matrix. Defaults to None.

See also: `MatrixRec`

```
>>> s = Server().boot()
>>> s.start()
>>> SIZE = 256
>>> mm = NewMatrix(SIZE, SIZE)
>>> mm.genSineTerrain(freq=2, phase=16)
>>> lfw = Sine([.1,.11], 0, .124, .25)
>>> lfh = Sine([.15,.16], 0, .124, .25)
>>> w = Sine(100, 0, lfw, .5)
>>> h = Sine(10.5, 0, lfh, .5)
>>> c = MatrixPointer(mm, w, h, mul=.2).out()
```

replace(*x*) [source]

Replaces the actual matrix.

Args: x: list of list of floats

New matrix. Must be of the same size as the actual matrix.

getRate() [source]

Returns the frequency (cycle per second) to give to an oscillator to read the sound at its original pitch.

genSineTerrain(*freq=1*, *phase=0.0625*) [source]

Generates a modulated sinusoidal terrain.

Args: freq: float

Frequency of sinusoids used to created the terrain. Defaults to 1.

phase: float

Phase deviation between rows of the terrain. Should be in the range 0 -> 1. Defaults to 0.0625.