Version 0.96beta

Scanned Generators

Intro

Scanned synthesis involves a slow dynamic system whose frequencies of vibration are below about 15 hz. The system is directly manipulated by motions of the performer. The vibrations of the system are a function of the initial conditions, the forces applied by the performer, and the dynamics of the system. Examples include slowly vibrating strings, two dimensional surfaces obeying the wave equation, and a waterbed. We have simulated the string and surface models on a computer. Our waterbed model is purely conceptual.

****

The Scanned Generators for Blue consists of 4 main parts:
1. Scanned Matrix Generator
2. Scanned Matrix Plot
3. Scanned Trajectory Generator
4. Scanned Trajectory Plot

The idea is to generate patterns for use with the Scanned Synthesis opcodes scanu/scanu2 and a trajectory for the scans opcode of Csound.

Blue provides a way to run formulas in Csound code using a GUI: buttons and sliders, creating Matrices and Trajectories. In addition, 2 Python scripts allow creating a visual representation of these matrices and trajectories.

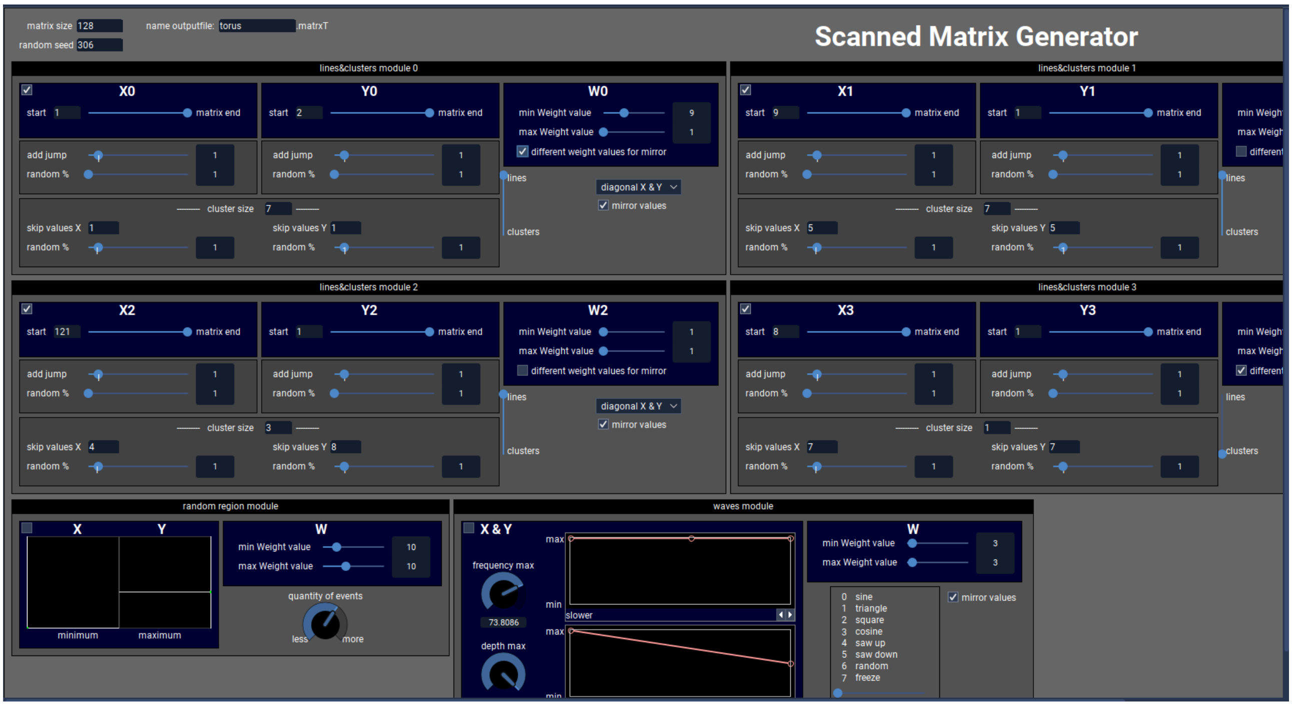
1. Scanned Matrix Generator

- Used to generate patterns for the Matrix.
- Adds the .matrxT extension to the outputfile
- in the format as needed by GEN44: *f33 0 16384 -44 "cylinder_128.matrxT"*

The Generator starts counting at 1, but the real start value is up to the user. It ends generating values for Lines or Clusterts until matrix size or lower by setting the ‘matrix size’ slider.
In the example below the matrix size is set to 128.

Both the start value and end value can be set . Start may be >1, end values < matrix values; in that case, a cut section of that Pattern will be generated.

Matrix size value is printed to the name of the output file, for example: string_128.matrxT.
This provides the user the information so that he/she can use this to set the correct value to the other tables that are needed: initial position, damping, masses, centering and displace (=velocity).



The Scanned Matrix Generator provides 3 different ways for generating the X, Y and W coordinates. X and Y are the coordinates, W is the weight of that node (see GEN44 information):

1. **Lines & Clusters**: Lines are uninterrupted lines (although the lines can contain value jumps in between the nodes). Clusters are interrupted, or several 'sections' of short lines.
2. **Random Pattern**: a defined section in the Matrix can be populated with an adjustable number of nodes.
3. **Waves**: sines and other waveforms can be used to generate a continuous flow of nodes but in a waveform.

All these generator modules can be mixed to get really complicated matrices.

The 'traditional' Matrix with 128 node points (the .matrxB format) can be found as presets in the Preset list of the Instrument: string, stringcircular, grid, cylinder and torus. See tutorials: https://www.csounds.com/scanned.
They are converted 1:1 to the text format .matrxT.

2. Scanned Matrix Plot

This Python script uses the Matlabplot Python library to plot the mass nodes of a Scanned Synthesis Matrix of type ‘text’.

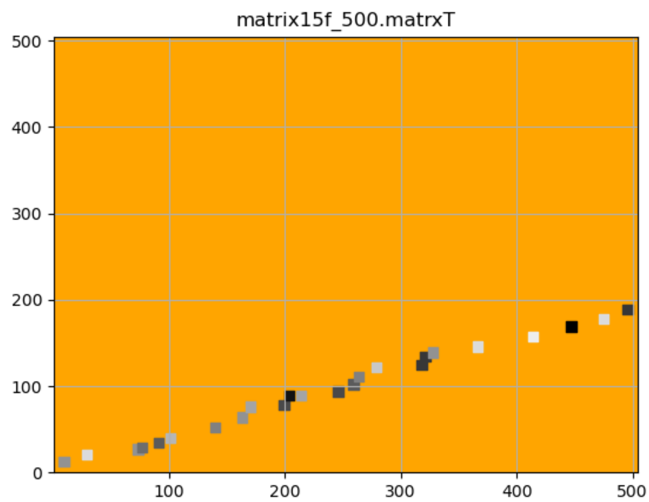
Background and Markers (the values) can be adjusted to your taste.

The presentation is one comparable to the Matrix Editor from Steven Yi, a Java Matrix editor. This Scanned Matrix Editor is a part of the program Blue (https://blue.kunstmusik.com/).

But there are differences with the Scanned Matrix Generator:

- Steven's Matrix presentation of a Matrix starts at the top left and reads downwards. The Scanned Matrix Plot starts at the bottom left and moves upwards.
- Steven's Matrix Editor loads and edits binary matrix files (.matrxB) only, the Scanned Matrix MatLabplot reads text matrix files (.matrxT) only.
- The Scanned Matrix Plot can **not** be edited but is meant as a visual presentation only.

The Blue file has all Presets matrices, like cylinder, string and torus. These .matrxT files should sound exactly like their older brothers that are in .matrxB format.



In this example, looking at the name of the file: matrix15f_500.matrxT, you can see that a Matrix, named matrix15f, contains 500 points. The dots show their coordinates. The dots have a color range from white to black, representing the lowest and the highest weight of that mass.

How to interpret the Matrix plot?

I can not tell you exactly. But it has to be read from bottom left to top up.

I find it difficult to interpret Matrices. A string is easy as it is one connected line of masses. But it is getting less transparent if you see the representation of the Matrix surface and connections when the string is attached from the end to the rear, making it a ‘necklace’.
Once the ‘necklace’ is moving, you can understand that this motion will continue once it has reached the end of the Matrix and starts moving the first mass…

When you compare a few matrices as generated by the Scanned Matrix Generator, I do hear differences in how the sound propagates through these matrices. The wave speed may be the same, but the ‘waves’ speed and timbre vary.

After that, it gets blurry…

It is, hopefully, by experimenting that cues can be learned.

3. Scanned Trajectory Generator

For the scans opcode, the table of a Trajectory does not need to have the same size as the other ftables used in scanu/scanu2. Size can be higher, equal or smaller. There can even be empty values (spaces).
In short: you do not have to scan all the masses in a Matrix, but you can decide what portions you want to scan and how many times you want to scan that portion- or make jumps to another section: a smaller size ftable makes a 'slice' of the masses to scan.

Example of the 'old' Trajectory text format in action, read by GEN23:

ifntraj ftgen   7, 0, 0, -23, "128-spiral-8,16,128,2,1over2.traj"

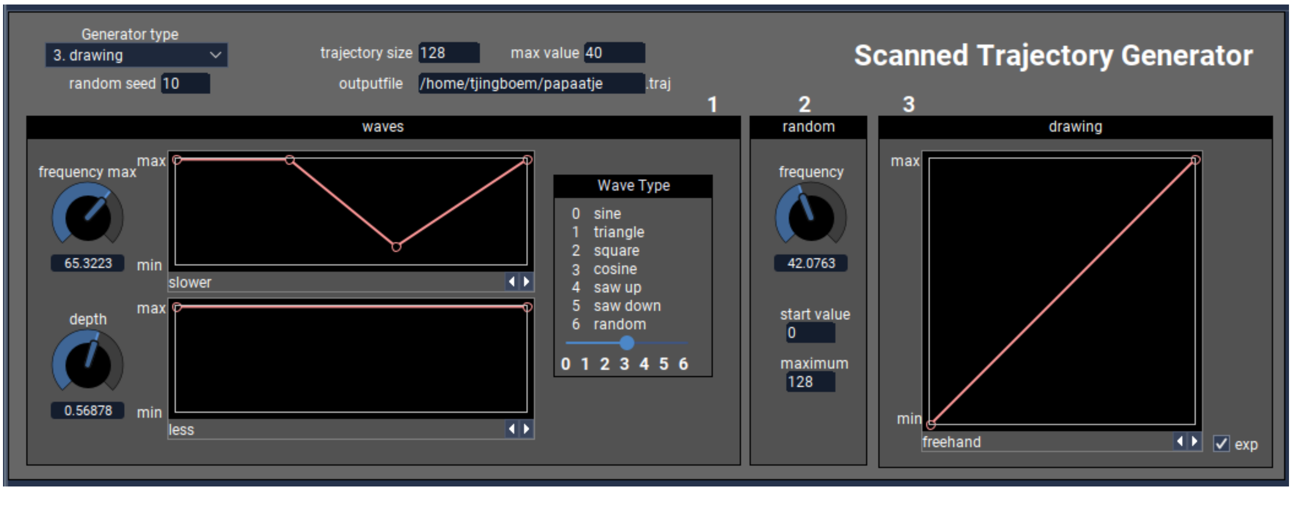
(Let Csound determine the size of the elements by using '0'. The negative value for GEN23 (-23) is to avoid normalization.)

The format of the Scanned Trajectory Generator is: **NAME_maxVALUE_NUMBER.traj**
where:
1. NAME : the name to identify this trajectory. Can be anything that makes sense to you.
2. _maxNUMBER : the maximum value of the mass node you want to include
3. _NUMBER   : length of the trajectory. May be higher, equal or lower than the number of mass nodes in the Matrix.

Example of the format as generated by the Scanned Trajectory Generator :

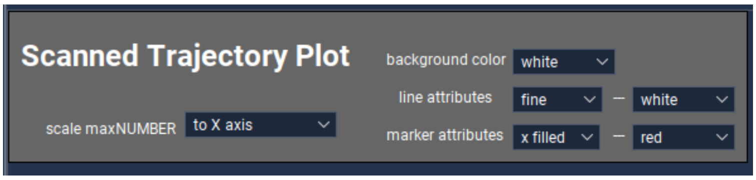
ifntraj ftgen   7, 0, 0, -23, "spiral_max128_128.traj"

The value must be between 0 and the max or equal amount of nodes in the Matrix.



4. Scanned Trajectory Plot

This Python script uses the Matlabplot Python library to plot a Scanned Synthesis Trajectory. This is a trajectory connecting the value points, showing how the mass nodes of the scanned synthesis Matrix are going to be read.



The format of the Scanned Trajectory Generator is: **NAME_maxVALUE_NUMBER.traj**
where:
1. NAME : the name to identify this trajectory. Can be anything that makes sense to you.
2. _maxNUMBER : the maximum value of the mass node you want to include
3. _NUMBER : length of the trajectory. May be higher, equal or lower than the number of mass nodes in the Matrix.

Optionally, the maxNUMBER value can be scaled so that the Y axis zoom in a bit, relative to the X axis, and depending on the values of the Y axis. Result is that only the Trajectory is shown and not the empty spots. Background color, the Line and Marker attributes can be changed to your taste.

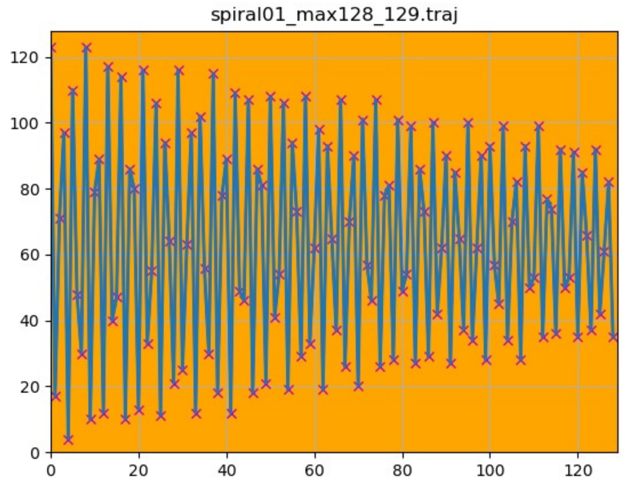
Important: Trajectories that are NOT in this format can not be recognized and shown.

The Scanned Trajectory Generator is used to generate the Trajectory by the Waves, Random points or Drawing modules manually;
the Scanned Trajectory Plot shows the generated routes reading through the Matrix. It is a visual presentation of this route.

The Trajectory has a great influence on the sonic result.
When the trajectory would be linear, like going from beginning to end, all mass nodes would be scanned in sequence like a straight line, scanning the Matrix from zero to end. When in Drawing Mode, there is an option to generate the 'straight' line and convert this to an exponential line.

How to interpret the Trajectory plot?

Example 1: *ifntraj1 ftgen 7, 0, 0, -23, "spiral01_max128_129.traj"*

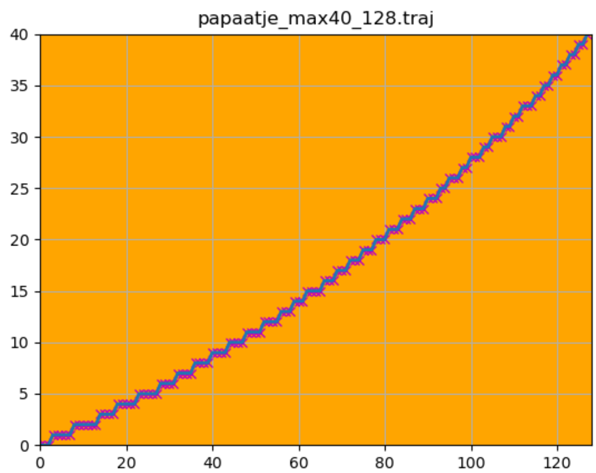


The trajectory path reads from left to right. The Trajectory above is reading at high speed, several times over some of the 128 mass nodes; it is indeed spiraling- the great advantage of actually have a visual presentation showing this. Originally, the old name is *spiral-8,16,128,2,1over2*. The name ‘spiral’ makes sense immediately, but I have no idea what these numbers stand for.

So, not all of the masses of this Matrix (mostly between from about minimum 10 to maximum 122, moving to the values between 35 and about 90) are part of the Trajectory. Quite a few masses, if they are present in the particular Matrix, are not read at all!
But it moves at high frequency, probably responsible for the FM-like sounding result.

It is quite feasible and plausible, that by using another Trajectory for reading the same Matrix, you get an **entirely** different sounding result; it is all about which portion you are going to scan and which masses you want to ignore.

Another example, a simple one:
Example 2: *ifntraj2 ftgen 7, 0, 0, -23, "papaatje_max40_128.traj”*



If you were to add a second scans opcode to your code and let it use the Trajectory from Example 2 on the same Matrix (there is no one stopping you), you would get a **very** different sounding result; only the first 40 masses of the Matrix are going to be scanned, and that in an exponential way.

A few thoughts

2 things are needed to set an existing table in motion: the matrix and the trajectory. Using different matrices results the table to sound in **different harmonics and different waves in speed and timbre**. Where different trajectories (reading back the table) result in a **filter** effect on these harmonics: when a Trajectory does not scan the matrix from 1 to matrix size completely, and not scan all matrix nodes, some harmonics (that are in movement) will not all be read or are less strong. And a fast scan trajectory like the one you see above, adds a FM like distortion due to its fast changing frequency.

Preliminary findings: there is a HUGE difference in how the scanu sounds and reacts to k-rate changes, compared to scanu2. Although all i-rate and k-rate settings and order are the same, these settings from one to the other do NOT map at all!

The sounding result from the ‘old’ scanu is:
1. a ‘thinner’ sound, most of the times less bass
2. scanu blows up easily and that makes it harder to actually play with the k-rate options and keep things going.
3. scanu produces great results with the audio injection.

Stiffness and Masses have a great impact, the other k-rate options do not or have a more limited range before things starts getting crazy and blow up.
And once the system blows up, there is no recovering.

This does not mean that scanu is not usable- it sounds a bit rough but that is a value on its own.
Also, it reacts quite nicely to the audio injection.
It is, however, much more difficult to keep it happy and going.

This is a case where both tomatoes and tomatoes are different :-)