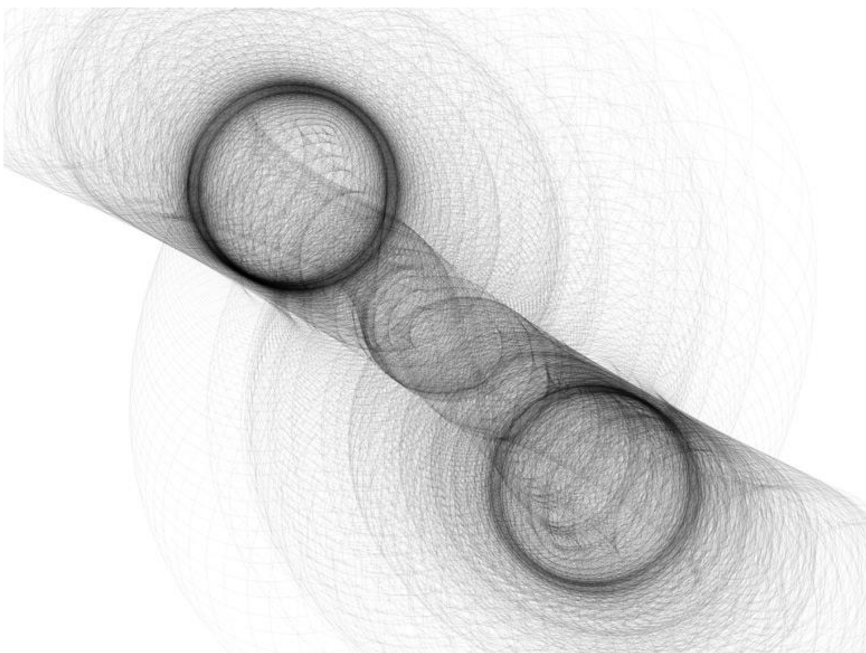
Chaotic Soundscapes: The Aural Dimension of Strange Attractors



CPSC432: Sound Representation and Synthesis
Final Project

Arsalan Sufi and Kevin Jiang

Due: 4/30/14

Introduction ---

For our final project, we were interested in further exploring the intersection between mathematical models and musical signals. Throughout CPSC432, we covered both the theoretical and practical foundations of computer-generated music by working through the basics of signal modification, sound synthesis, and the representation of musical elements in Haskell. What we found especially interesting and novel about using Haskell as an interface for generating computer music was the ease with which we were able to convert pure mathematical equations into musical outputs. In our project, we examined the sound outputs of strange attractors, which are sets of differential equations that exhibit chaotic behavior. Strange attractors were of particular interest to us due to their repetitive, yet non-repeating nature. These unique characteristics allowed us to explore sounds with timbres and harmonies that would have been difficult to generate through normal human composition.

Previous research has been done concerning the sounds of strange attractors in other audio programming languages such as Csound. These programs generally use an iterative programming approach to simulate the sets of differential equations governing the systems. Using Haskell, we approached the problem of modelling these attractors from a functional programming point of view--by using the concept of signal functions, we were able to come closer to mimicking the continuous nature of these equations. Specifically, we converted the differential equations into integral equations so that we could easily access the outputs of each equation. That said, as we did have to modify the clock rates of these signal functions, we were ultimately still working in the discrete realm.

Below, we have discussed the results from three of the strange attractors that we examined and modelled: the Lorentz, Rossler, and Chua attractors. All these attractors govern different phenomena in the natural world. The Lorentz equations were originally derived from a model describing fluid circulation in shallow layers of fluid heated from below and cooled from above known as Rayleigh-Bénard convection. The Rossler attractor has been useful in modelling equilibria in chemical reactions and the Chua attractor is a mathematical representation of a simple electronic circuit known to exhibit chaotic behavior. In our project, we examined a different dimension of these strange attractors by looking at their sound characteristics. By experimenting with the input parameters to these sets of differential equations, the signal characteristics (i.e. frequency, amplitude, and spatial aspects) that each equation governed, and the musical output of these signals, we were able to create a varied array of soundscapes that each represent a unique aspect of the aural behavior of these strange attractors.

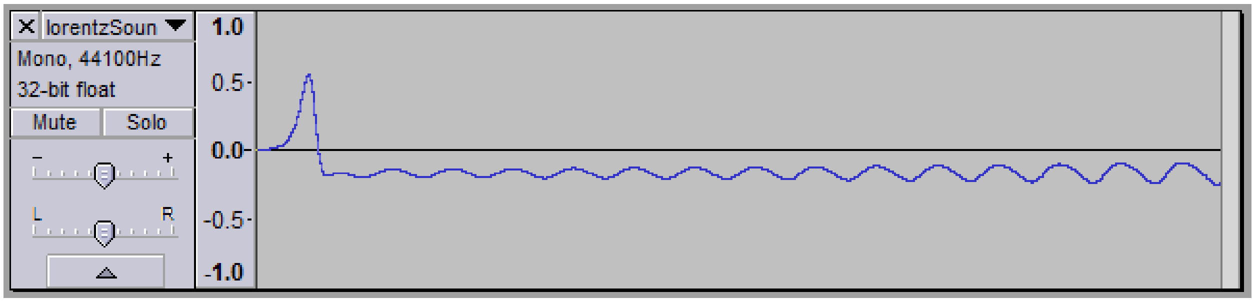
Lorentz Attractor ---

$dx/dt = s(y - x)$
$dy/dt = -y - xz + rx$
$dz/dt = xy - bz$

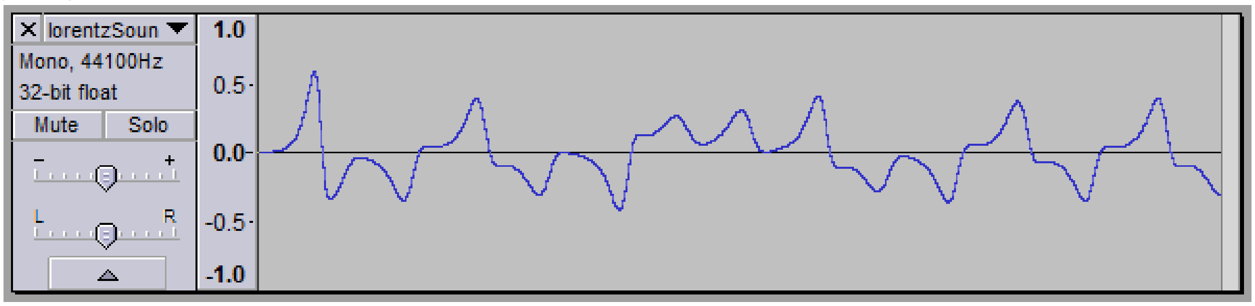
Initial Approach

We first tried use the outputs of the Lorentz attractor directly as sound waves. However, the frequencies of the outputs were well below the audible range. To overcome this issue, we tried changing the clock rate of our lorentz signal function, reducing it from the standard 44100 Hz to 100 Hz. While this change did produce a different output, the frequencies were still below the audible range. Below are two plots illustrating the change.

y-output of our lorentz function at a 44100-Hz clock rate



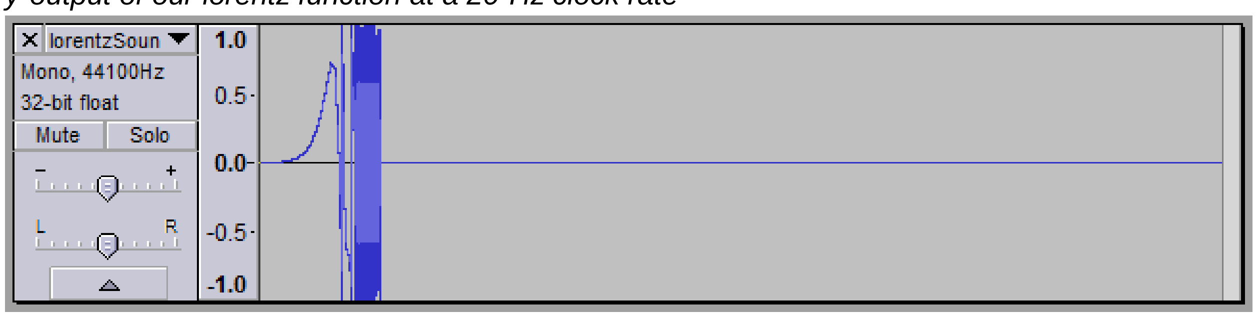
y-output of our lorentz function at a 100-Hz clock rate



Sine-Wave Modulation

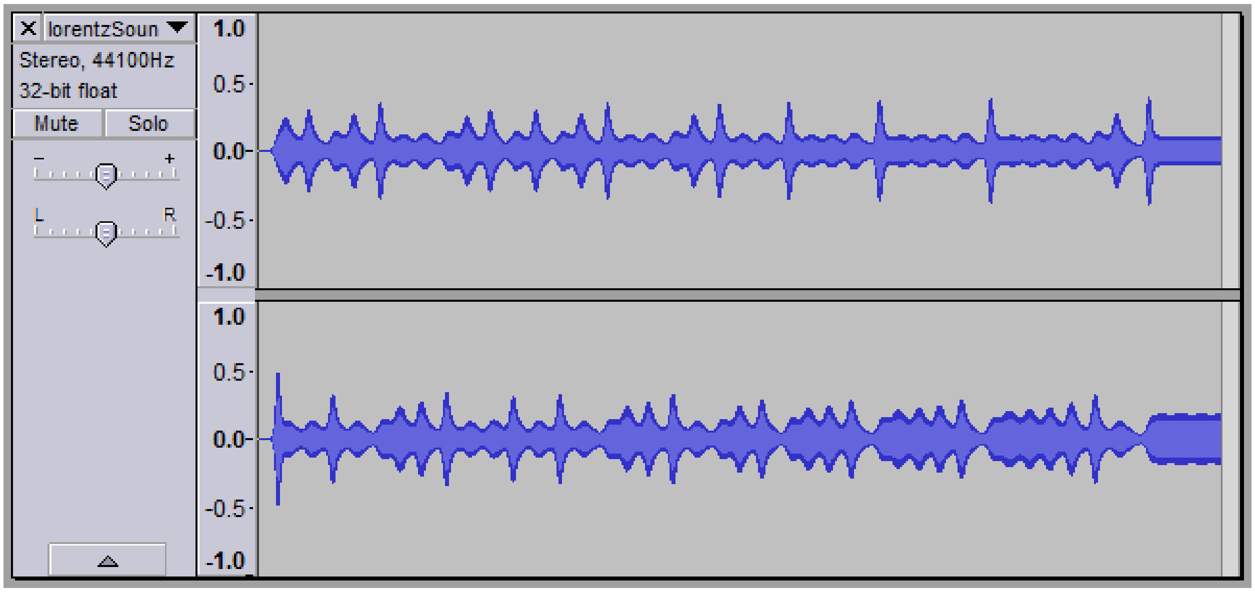
From here, we decided to use the three outputs of our lorentz function (x, y, and z) to modulate a sine wave. We still maintained the 100-Hz clock rate as it produced more interesting waves than the 44100-Hz clock rate (the above images being a testament to this). We attempted to lower the clock rate down even further to 20 Hz to see if this introduced even more variance, but this proved not to be the case.

y-output of our lorentz function at a 20-Hz clock rate

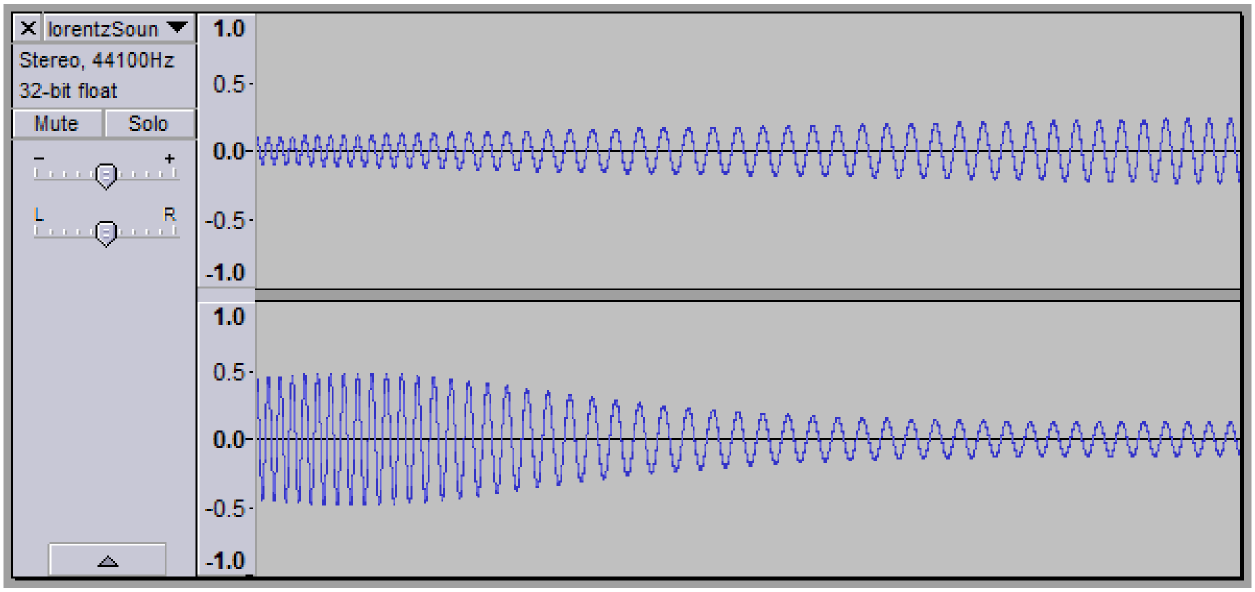


We ultimately decided to use the y-output to modulate frequency, the z-output to modulate amplitude, and the x-output to modulate the balance between the left- and right-channels. This pairing was suggested by Michael Winters in his paper “Musical Mapping of Chaotic Attractors.” The paper suggested that the most interesting output be used to modulate frequency, the next-most interesting output be used to modulate amplitude, and the least interesting output be used to modulate left-right balance. After finding the rate scaling factors for each of these outputs, a process which involved quite a bit of trial-and-error, we produced the following wave.

lorentz-modulated sine wave with the following parameters:
s = 10
r = 28
b = 2.667
and the following initial value for the x-, y-, and z-outputs:
i = 0.6

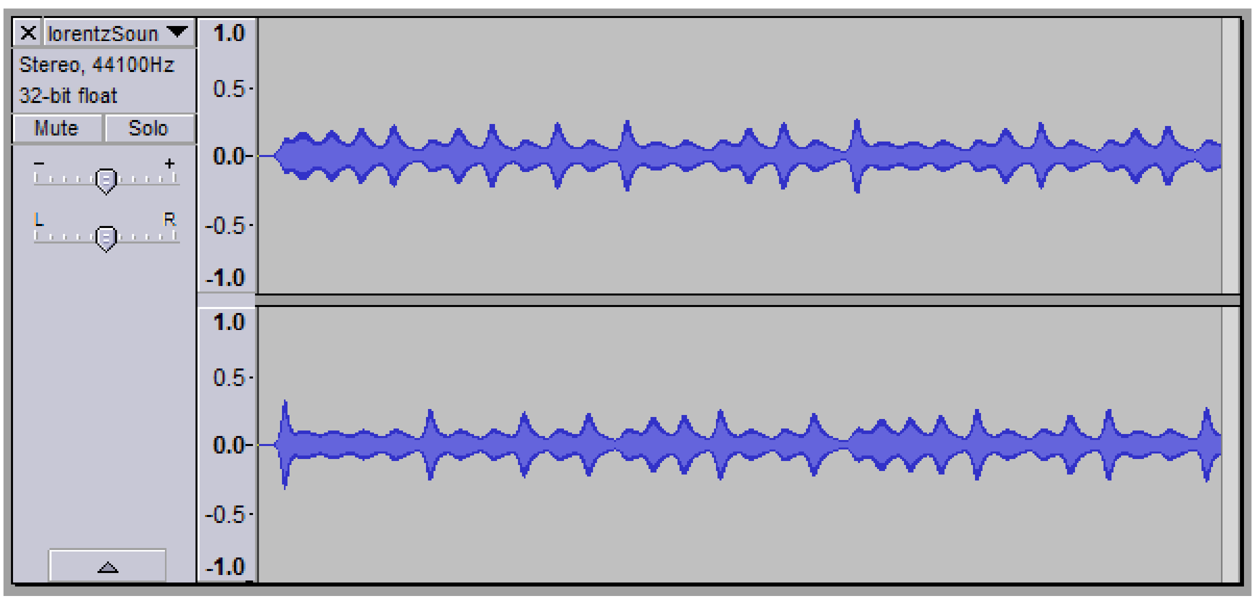


close-up of the same wave



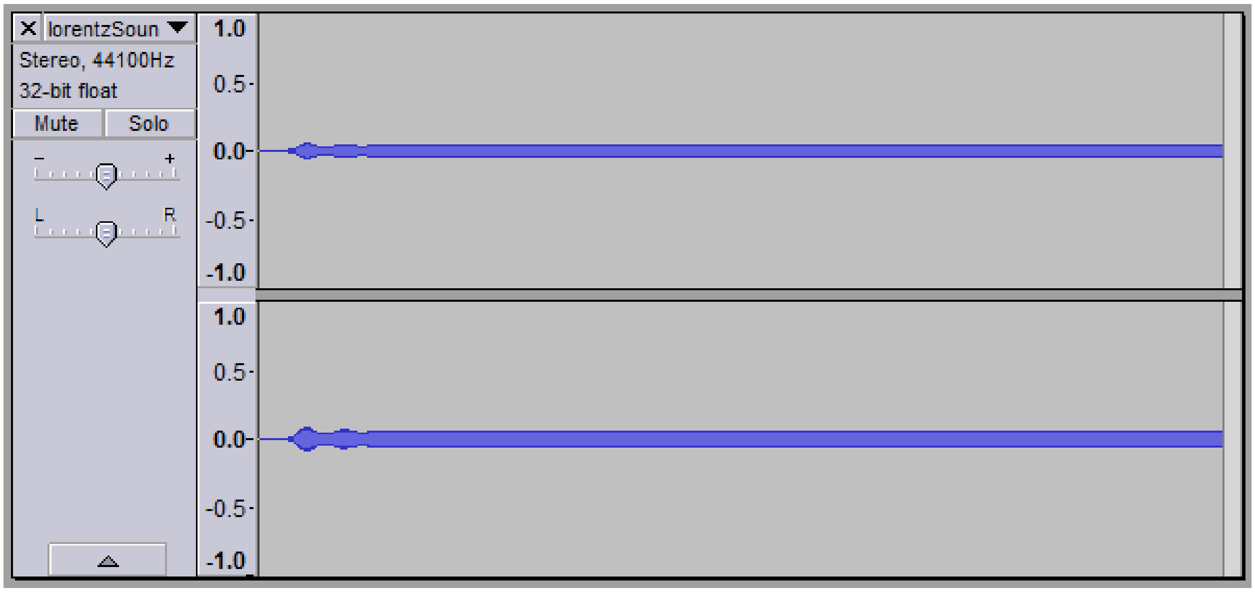
The parameters used above were suggested in many research papers. As a proof-of-concept, we tried varying these parameters to experiment with the chaotic nature of the Lorentz attractor.

lorentz-modulated sine wave with the following parameters:
s = 5
r = 23
b = 2
and the following initial value for the x-, y-, and z-outputs:
i = 0.6



The above sound is similar in nature the first sound that we outputted but clearly distinct. We tried one additional set of parameters after this.

lorentz-modulated sine wave with the following parameters:
s = 2.718
r = 10
b = 3.14
and the following initial value for the x-, y-, and z-outputs:
i = 0.5

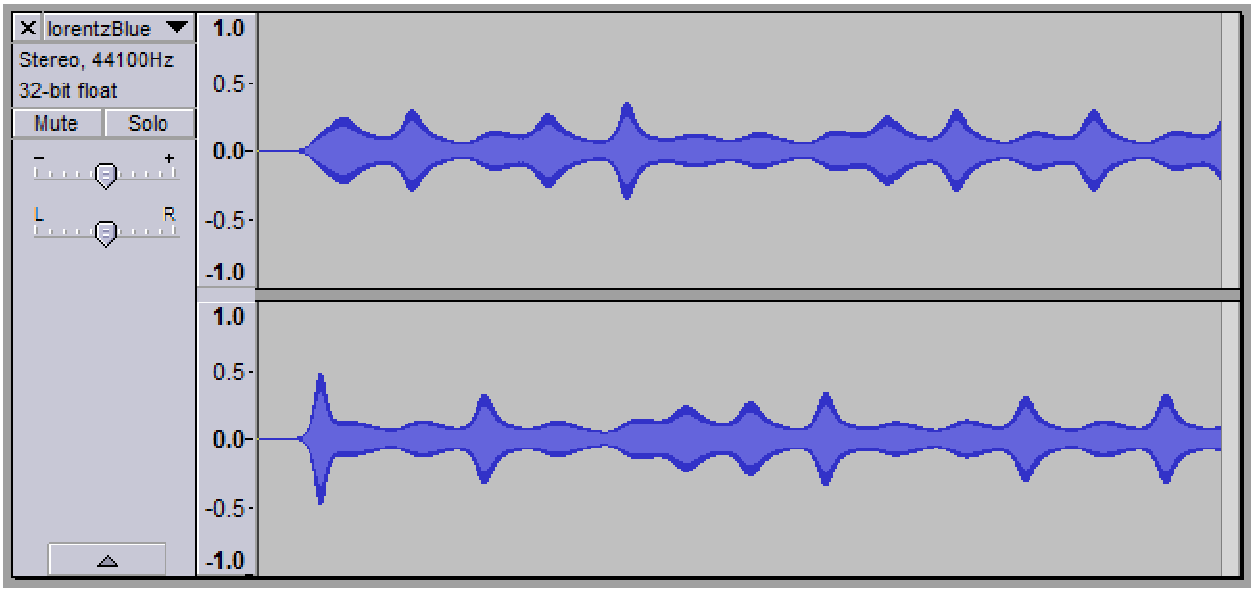


This final wave was much less chaotic than the previous waves.

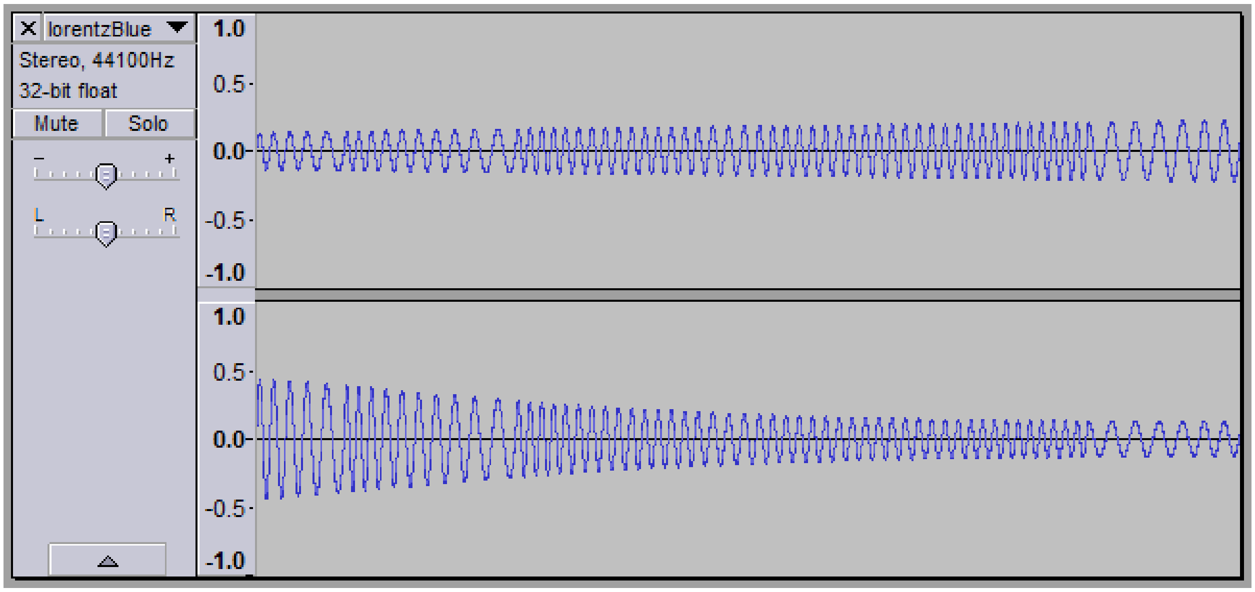
Alternative Methods of Modulation

In addition to simple modulation of a sine wave, we decided to try some alternative methods of modulation. Rather than scaling the y-output to bring it into the range of audible frequencies, we originally tried setting up “buckets” such that, whenever the y-output of our Lorentz function fell in a certain range, a specific frequency was outputted. However, because the y-output tends to oscillate around a central region, it was hard to set up the buckets such that the selected frequencies played with equal probability. To overcome this issue, we took the mod of the y-output to randomly select a pitch. The pitches were restricted to the blues scale rooted at C4. Below is the result.

lorentzBlues



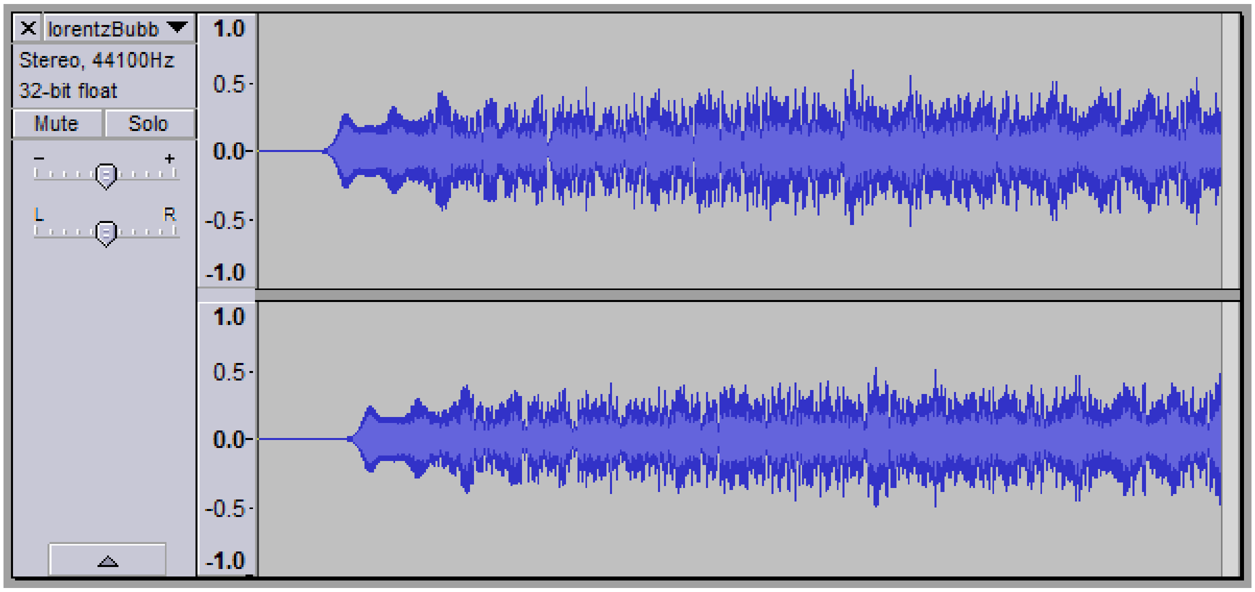
close-up of the same wave



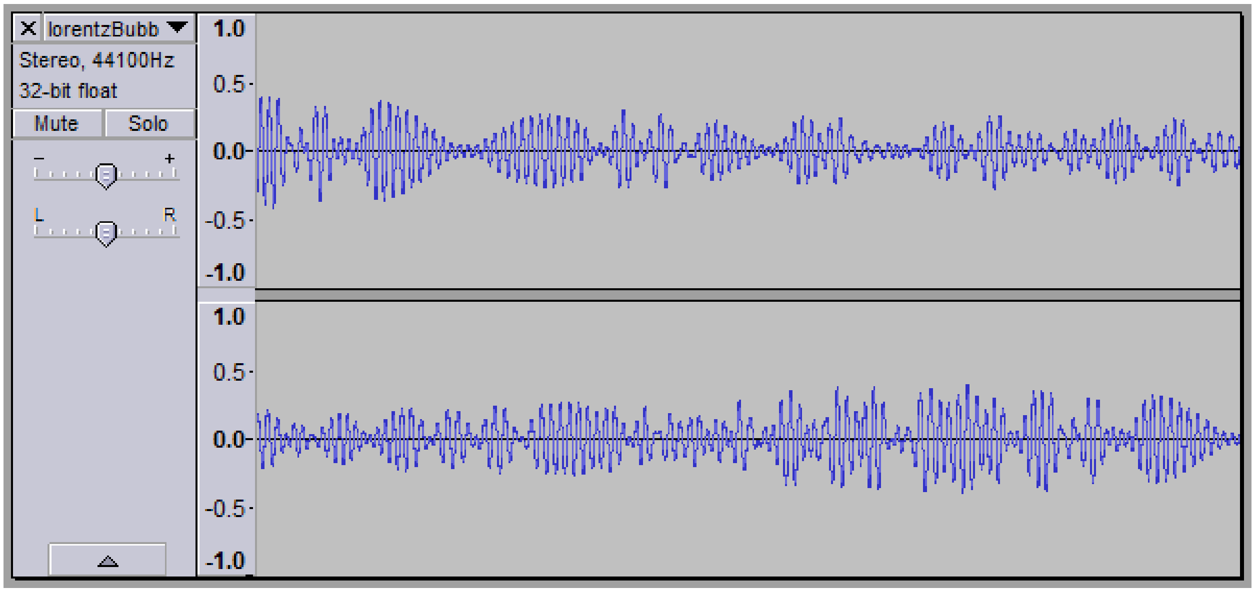
Because we didn't scale the y-output down (without any scaling, the y-output tends to vary from -50 to 50) and took a mod 7, the frequencies changed very rapidly, something which is apparent in the above plot. Accordingly, the wave sounded more like a cheesy, science fiction spaceship than a blues piece.

Our final approach was similar to the above approach except that the pitches were limited to C4, E4, F4, and G4. In addition, the pitch bursts were sent through the echo bounce function that we wrote for HW9. (Note that the x-output was no longer used to modulate the left-right balance as this modulation was now taken care of by the echo bounce function.) The first wave that we produced sounded somewhat like a bubbling pot.

lorentzBubbles1

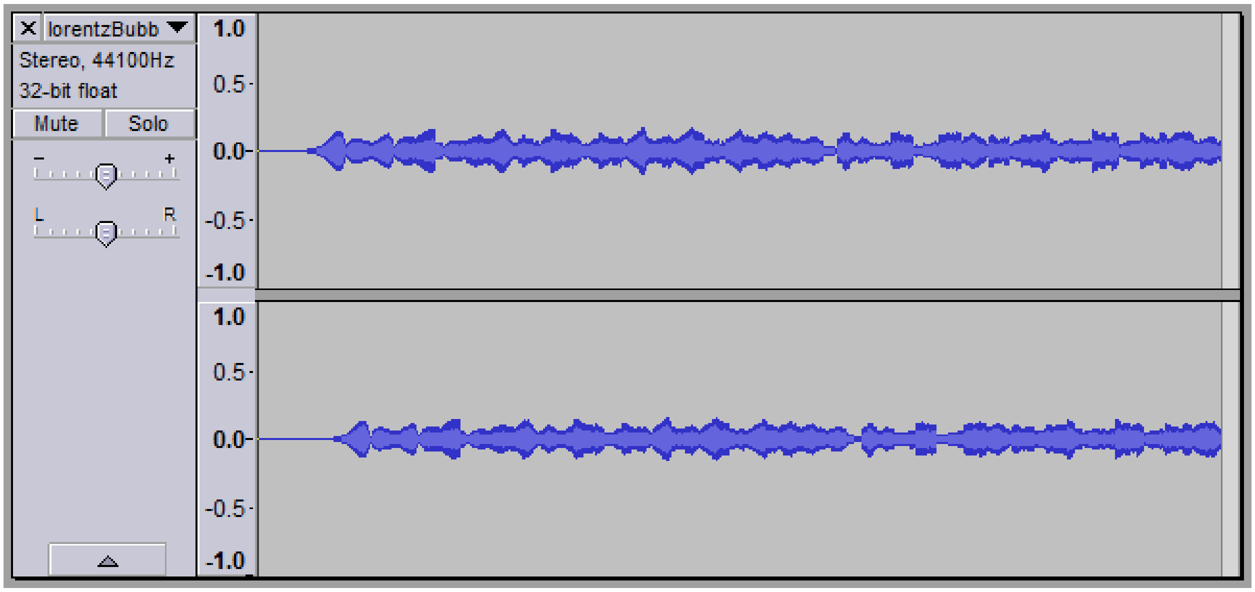


close-up of the same wave

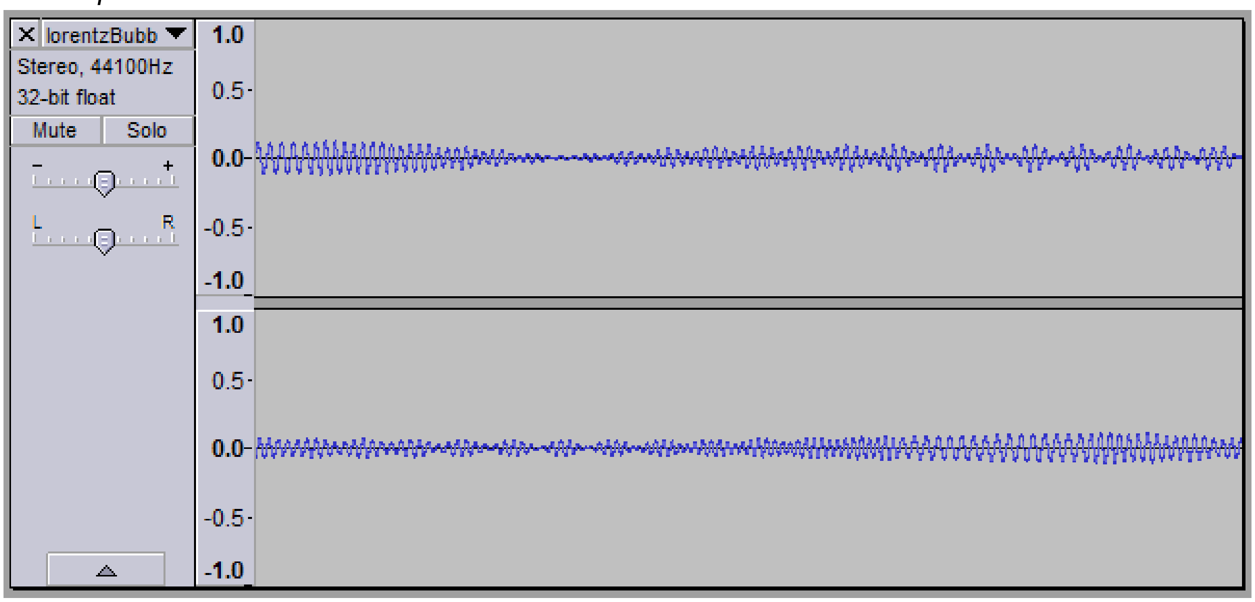


The final wave that we produced is identical to the one above except that the pitch bursts happen less frequently. In order to accomplish this, we simply scaled down the output used to modulate frequency. In addition, the roles of the y- and z-outputs were swapped. The y-output was used to modulate amplitude, and the z-output was used to modulate frequency (this was actually an accident). The result was easily the most pleasing sound that we produced over the course of this project! It sounds somewhat aquatic.

lorentzBubbles2



close-up of the same wave

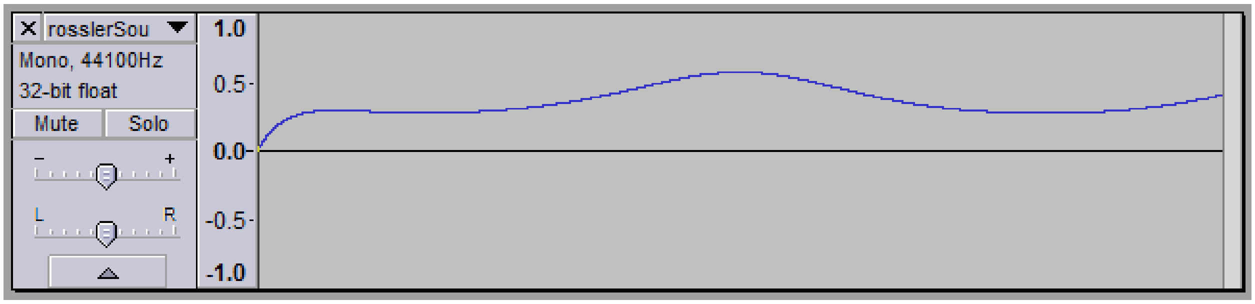


Rossler Attractor **---**

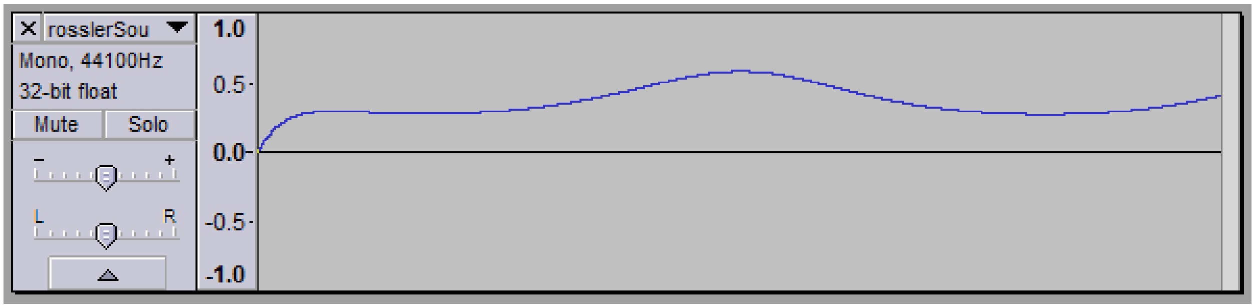
dx/dt = -y - z
dy/dt = x + uy
dz/dt = v + xz - wz

With the Rossler attractor, we simply used the sine-wave modulation approach. Nonetheless, we still outputted the z-output directly at two different clock rates to determine whether or not reducing the clock produced a more interesting output (as was the case for the Lorentz attractor).

z-output of our rossler function at a 44100-Hz clock rate



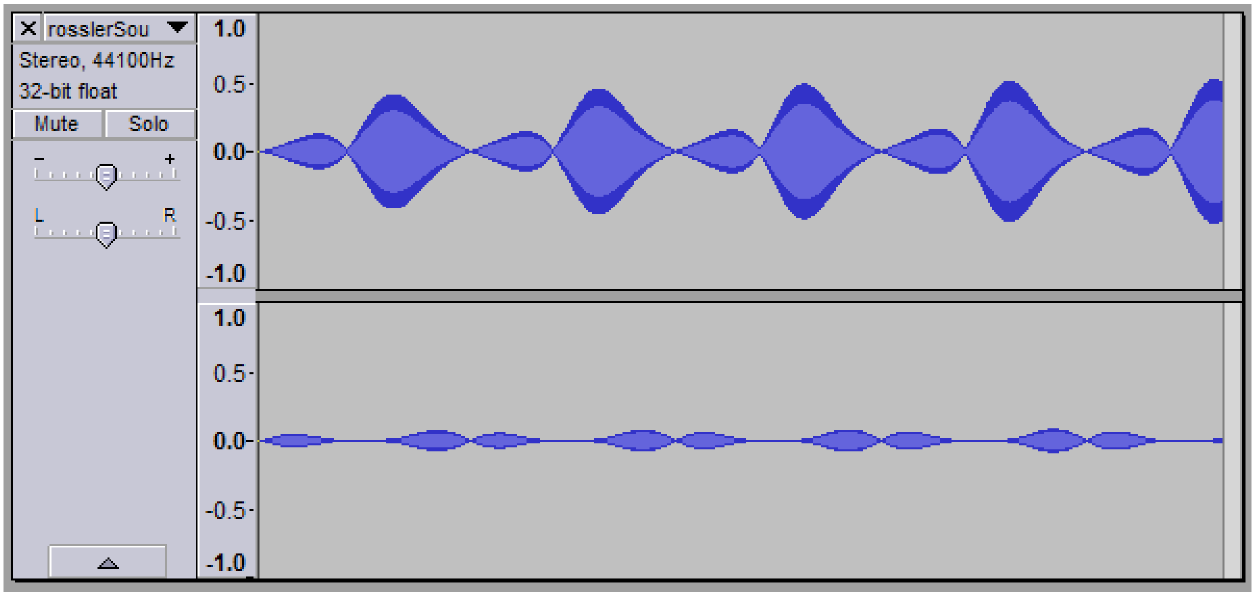
z-output of our rossler function at a 100-Hz clock rate



Interestingly, with our rossler function, changing the clock rate didn't produce a noticeable change. Accordingly, to maximize efficiency, we selected the 100-Hz clock rate. Applying the same pairing principle described for our lorentz function, using the most interesting output to modulate frequency, the next-most interesting output to modulate amplitude, and the least interesting output to modulate left-right balance, we generated the following modulation scheme:
x → amplitude
y → left-right balance
z → frequency
Note that, for the Rossler attractor, the x-, y-, and z-outputs need not be initialized to a value other than 0.

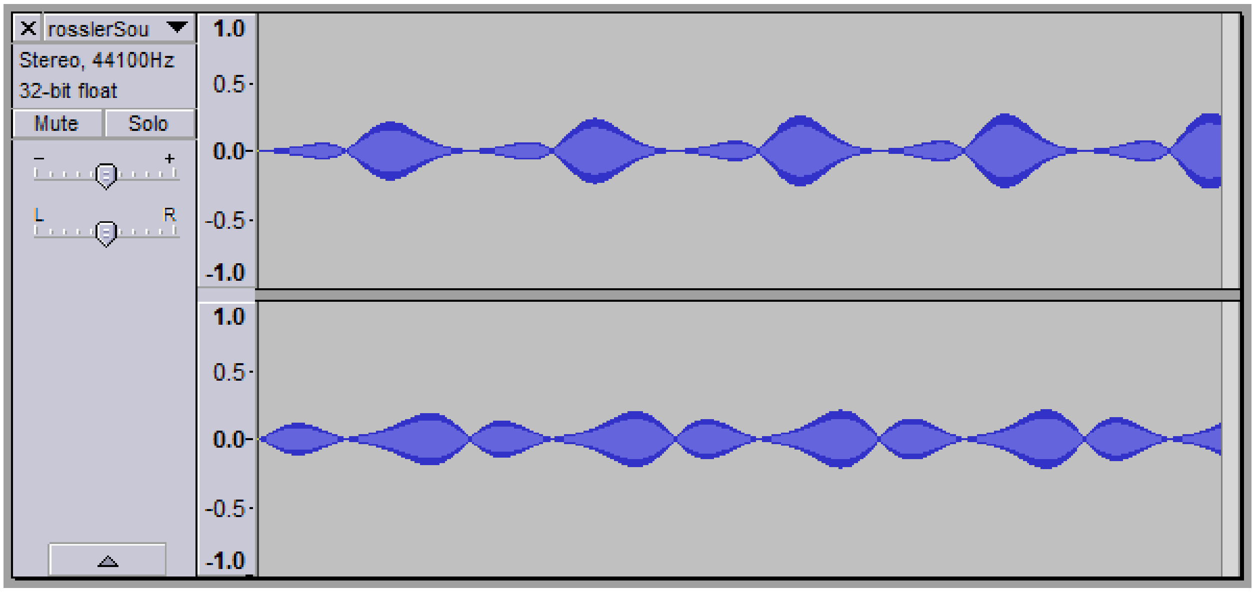
Once again, it's important to mention that finding the right scaling factors / expressions for each of the outputs required a process of trial-and-error. To illustrate this, we've included both our final output for our rossler function and an intermediate output where the spatial (left-right balance) modulation was still being adjusted.

intermediate-stage unbalanced rossler-modulated sine wave with the following parameters:
u = 0.375
v = 4
w = 4



Note how the amplitude balance is heavily-biased toward the left channel. With this output, the following expressions were used to scale the left and right channels:
left channel → (y / 3 - 1)^2 / 4
right channel → (y / 3 - 1)^2 / 4
To fix this issue, we simply added a constant prior to squaring:
left channel → ((y / 3 + 0.5) - 1)^2 / 4
right channel → ((y / 3 + 0.5) - 1)^2 / 4
Below is the resulting wave.

final balanced rossler-modulated sine wave with the following parameters:
u = 0.375
v = 4
w = 4

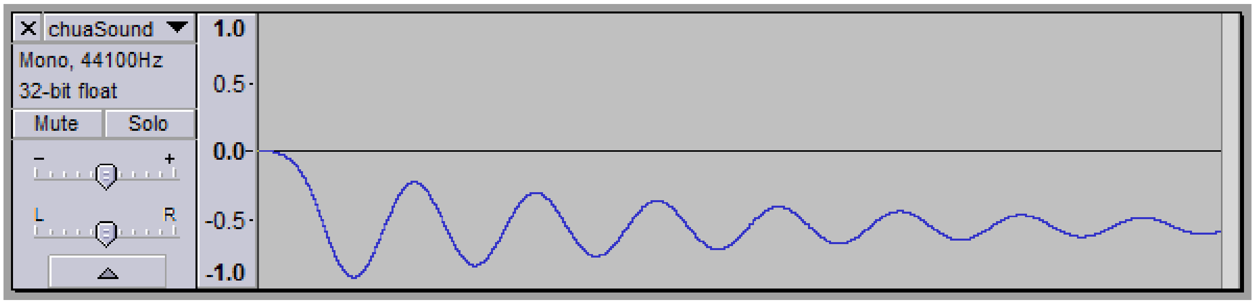


Chua Attractor --

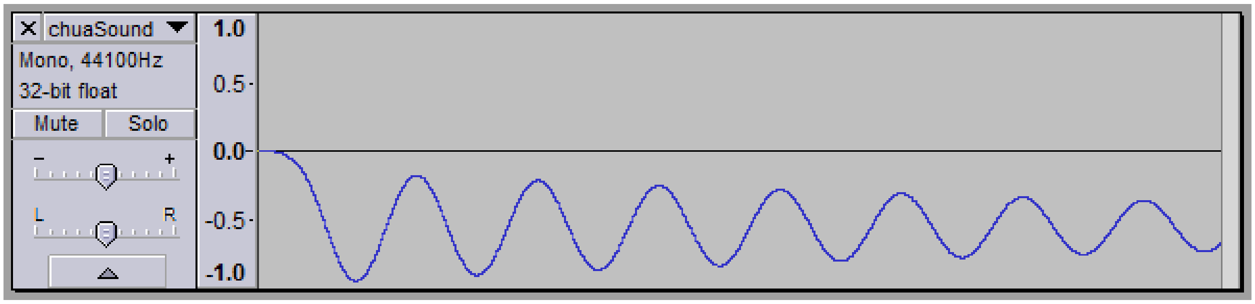
$dx/dt = c1(y - x - (m1x + ½(m0 - m1)(|x + 1| - |x - 1|)))$
$dy/dt = c2(x - y + z)$
$dz/dt = -c3y$

With the Chua attractor, we once again used the sine-wave modulation approach. As with the Lorentz and Rossler attractors, we first tested different clock rates.

z-output of our chua function at a 44100-Hz clock rate



z-output of our chua function at a 100-Hz clock rate

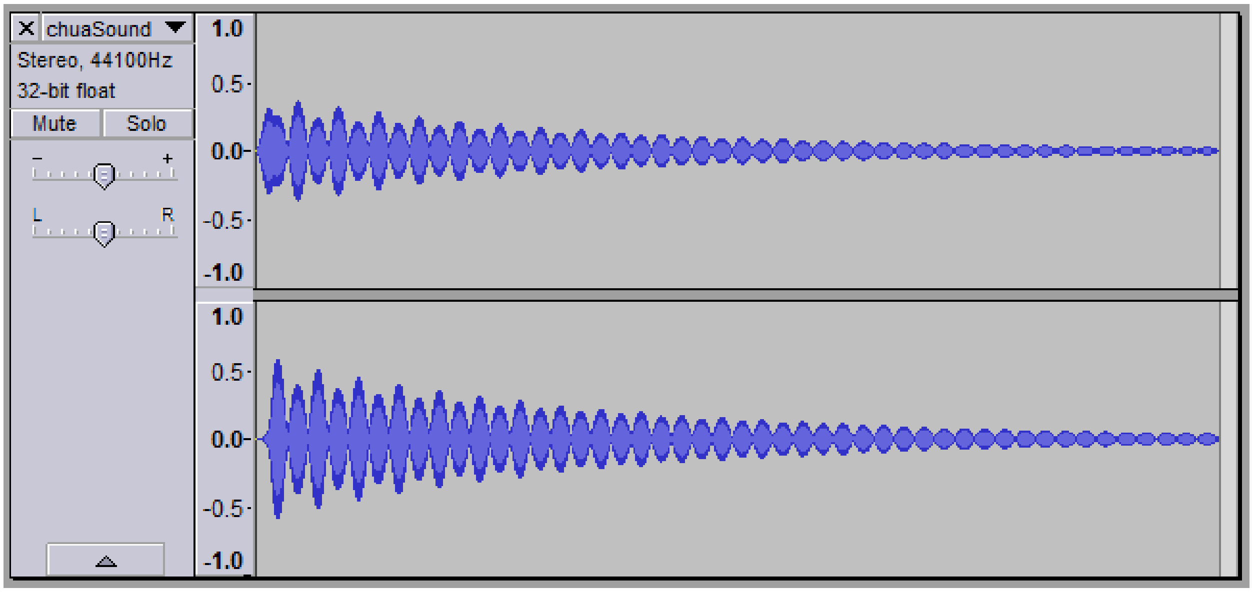


Although the change wasn't nearly as drastic as it was for our lorentz function, it was definitely noticeable. The 100-Hz output decayed more slowly than the 44100-Hz output. Once again, we selected the 100-Hz clock rate.

We used the following modulation scheme:
x ⟶ left-right balance
y ⟶ amplitude
z ⟶ frequency

chua-modulated sine wave with the following parameters:
c1 = 15.6
c2 = 1
c3 = 25.58
m0 = -1.143
m1 = 0.714
and the following initial value for the x-output:

i = 0.7



Final Remarks --

Through experimentation with the Lorentz, Rossler, and Chua attractors, we were able to generate a variety of soundscapes aurally congruent with the mathematical equations governing each attractor's behavior. However, we faced some challenges in the process of converting these chaotic equations into signal outputs. In particular, we had some issues getting our sound signals to correctly interpolate when we were upsampling our outputs. We would like to thank Professor Hudak and Daniel Winograd-Cort for helping us rectify this issue. Another challenge that we faced during this project was figuring out the best way to represent the outputs of these strange attractors musically. At first, we attempted to directly output the x, y, and z variables from the attractors as a sound by itself--however, we found that this did not create a very satisfying soundscape as we were limited to using only a portion of the potential outputs in generating music (i.e. only the x variable would be used to generate sound, while the y and z variables would be ignored). Additionally, these sounds often resulted in frequencies that were outside the human audible range. We later decided on using the x, y, and z outputs of each of these strange attractors to drive either the frequency, amplitude, or spatial organization of our sound output. This not only allowed us to incorporate all of the equation outputs into our sound, it also created sounds that were much more interesting and varied than the alternative. In picking the appropriate outputs to drive each musical characteristic, we generally tried to use the most "interesting" variable to drive the frequency and the "least interesting" variable to drive the spatial organization.

There are a variety of areas regarding the sound representations of strange attractors that we would like to explore in the future. While we experimented lightly with alternative methods to sine-wave modulation (particularly when examining the Lorentz attractor), we would have liked to further explore the potential of using these strange attractors as tools for generating sound effects and perhaps even making algorithmic compositions. Additionally, it would have been interesting to apply some other signal effects (i.e. tremelo, vibrato) that we examined during the course to the sounds that we created to generate a more varied soundscape. Finally, there are many bounded, non-repetitive sets of equations that could be converted into musical sounds using the techniques described in our project above. It would be interesting in the future to examine other mathematical models that describe the physical, chemical, and mathematical nature of the world around us and observe their aural dimension in Haskell.

Citations ---

We found the below resources very useful for our project.

1) http://mymbs.mobeard.org/~paul.fisher/FOV2-0010016C/FOV2-0010016E/FOV2-001001A3/book/chapters/19mikelson/
2) http://physics.wooster.edu/JrIS/Files/Winters_Web_article.pdf

A final note, throughout this write-up as well as our code, we've mistakenly referred to the Lorenz attractor as the Lorentz attractor.