Fractal Music and Musical Fractals

Aashay Vanarase, Madhavun Candadai, Owoyele Adafemi

# 

# Abstract

***What makes music pleasant to listen to? Is there an inherent rule that composers follow without even knowing about it? Can we find those rules, if they exist, and if we find them can anyone who learns the rules become a great composer? These are some deep questions about music and its nature. We are hoping to scratch the surface here by making an analysis for power law existence in the note recurrence intervals in classical music and estimate its fractal dimension. We have made specific studies with Bach's bwv261 chorale and Mozart's K155. We have found presence of power law distribution in the note recurrence intervals for each note. We have also tried to incorporate the power law nature that we observed in Bach's bwv261 to create our own score programmatically.***

# Introduction and Objective

Music is one of the most natural phenomenon humans have encountered. The history of music goes a long way and it has been studied and developed across numerous civilizations. While music has evolved a lot over centuries there seems to be some inherent property that makes music pleasant to hear. Obviously, musicians do not compose music by following a series of steps, but an insight into the underlying principles common to all music will hopefully help us give a scientific explanation to what makes music pleasant. The perspectives that the field of Complex systems offer have helped us understand and analyze natural and nature related systems better than any other field of science. So we decided to use a complex systems perspective to analyze music.

From a look at all the work related to analysis of music, we understood that music, especially classical music, tends to exhibit properties of complex systems, such as Power law nature and fractal dimensions[1][2][3][5]. What is interesting is that we also found literature describing methods to generate/compose music from fractals [4]. This project aims at analyzing classical music by Bach and Mozart to realize their power law and fractal properties. We also intend to create schemes that can be used to generate/compose music using fractals.

# Standards/Tools/Software

## MusicXML

MusicXML is a standard format for representing western classical music in a computer understandable way. The fact that it is an XML file allows multi-platform support and easy viewing. A large number of software now supports the MusicXML format. This format provides a bridge between sheet music and digital representation of music. The format contains XML tags such as <part/>, <measure/>, <note/>, <pitch/>, <key/> etc. software can read from and playback from the MusicXML format [6].

## Music21

Music21 is a Python library that was built by MIT for computer aided musicology [7]. The motive of this project is to provide a tool to make sophisticated musical and statistical analysis of what we listen based on pitch, note durations etc. This tool takes a symbolic perspective at music by providing notation for scores in terms of numbers. This library provides functions that help attain results with minimal code in a short timeframe. It also provides additional features for visualization and enhancement of the music data using

1. Python imaging library
2. MatPlotLib
3. Pyaudio
4. Pygame

Music21 supports the MusicXML format and allows music to be read from or written into that format. Another important asset of Music21 is that it includes a corpus of music data. The corpus includes symbolic musical scores by Bach, Mozart, Verdi etc. the corpus contains data that with attributes such as notes, note duration, pitch, full name of note, parts, frequency, volume etc.

To analyze musical scores using Music21 we write a python program as follows –

1. Import music21
2. Read from the corpus any specific score. The score would now be available as a list with each of its attributes accessible by name.
3. Read out the required attributes and work with numbers.
4. Data may be shown on a graph or using the MusicXML format explained above.

To create music using Music21 we write a python program as follows –

1. Import music21
2. Use the Note object to create a note by name.
3. Set duration, part and measure
4. Show score or export into the music XML format

## Myriad music plug-in

Myriad is a browser based MusicXml reader and player. With simple html coding this plugin can read any MusicXML file and display it in the form of script on the page. This plug in also has functionalities to play, pause and stop the playback. The player has an indicator to show current level of playback and the user can increase or decrease the tone at which the music is currently playing. There are options to increase tempo, go full screen and save file.

# Strategy and algorithm

## Power Law Analysis

Power law analysis of music has been done over various attributes such as –

1. Frequency – difference in frequency of successive notes are fractal [8]
2. Motivic scaling – repetition of rhythmic patterns [1]
3. Duration scaling – note durations are power law distributed [1]
4. Pitch related scaling – distribution of pitch is power law[1]
5. Structural scaling – dynamics of the piece in terms of recurrences [1]

We have implemented the structural scaling on a couple of pieces of western classical music by Mozart and Bach. We have taken the following steps to analyze structural scaling of music –

Step 1. Pick a particular note.

Step 2. Identify its recurrence intervals in the piece under observation.

Step 3. Sort and rank the sequence.

Step 4. Use the curve fitting toolbox to fit a curve to the sequence.

Step 5. Use the obtained equation to plot the data in log-log scale.

The above steps are repeated for a number of notes.

The following table describes the different environments we used to perform different tasks

|  |  |
| --- | --- |
| **Task** | **Tool / environment** |
| Read music data from corpus | Music21 / python |
| Generate the recurrence matrix for different notes | Python |
| Curve fitting for different notes | Curve fitting toolbox / MATLAB |

Table 1 - Tools/environments used for different tasks

## Creating Music

The first step in creating music from fractals is to identify a representation of fractals that can be used to translate into music. We wanted to extract a number series from a fractal and then use the extracted series to develop music from fractals using the MusicXML format. Then music can then be played using the Myriad plug-in.

The first goal is to generate a series of numbers from a fractal distribution. To do this we reuse the number patterns generated from the previous stage of the project. These number patterns are placed in a long array based on their indices as specified by the number series. This series is then converted to a string of notes and exported into MusicXML format using Music21. This can then be played using the Myriad music plugin on the browser.

The following table describes the different environments we used to perform different tasks

|  |  |
| --- | --- |
| **Task** | **Tool / environment** |
| Generate score from part 1 | MATLAB |
| Convert score to note objects and generate XML | Music21 / python |
| View score/ playback XML | Myriad plug-in / browser |

Table 2 - tools/environments used for different tasks

# 

# Simulations and Results

## Fractal Music

Below are a set of log-log plots for different notes from Bach’s bwv261 chorale. This piece was chosen because it was a long piece that could provide a fairly good number of samples to work with. The note recurrence intervals are found across the piece for each note, they were sorted and ranked and plotted as shown below in log-log scale. For example, the intervals between occurrences of note E3 are 12-34-67-78. These intervals are then ranked and plotted on a log-log scale.

In the plots that follow the term distance simply means the number of notes between consecutive recurrences of a note.

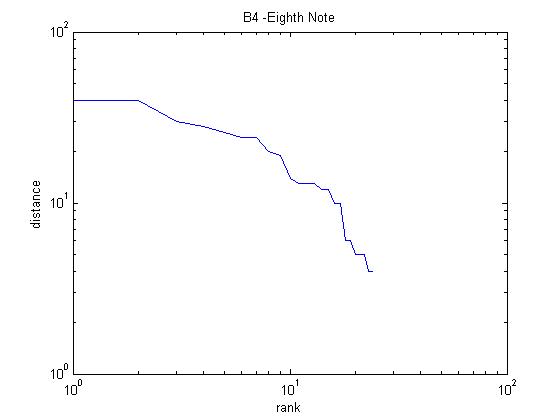


Figure 1 - Log-log plot for recurrence of note - B4

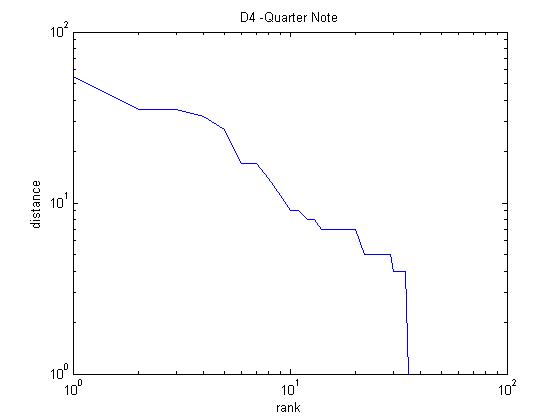


Figure 2 - Log-log plot for recurrence of note - D4

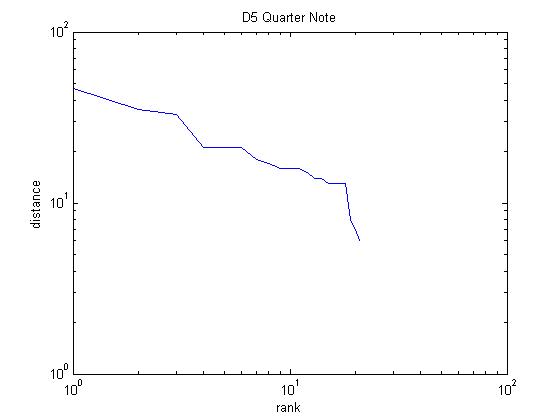


Figure 3 - Log-log plot for recurrence of note - D5

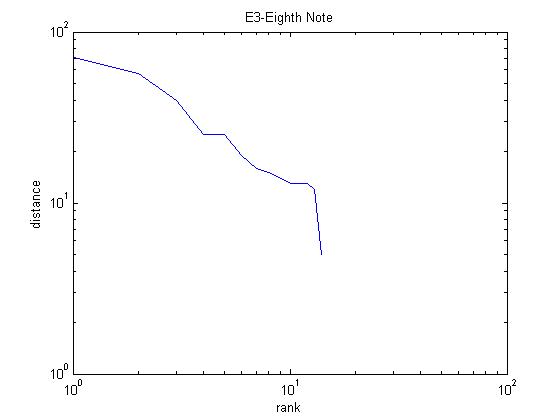


Figure 4 - Log-log plot for recurrence of note - E3

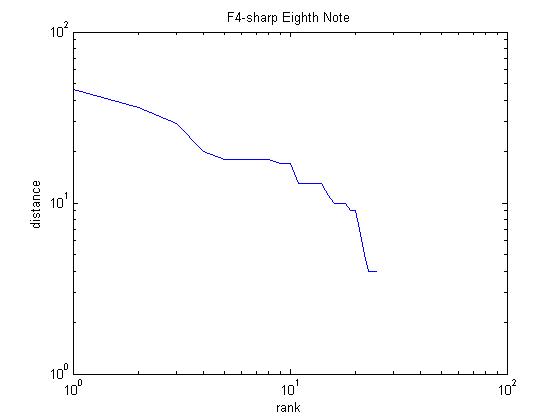


Figure 5 - Log-log plot for recurrence of note - F4

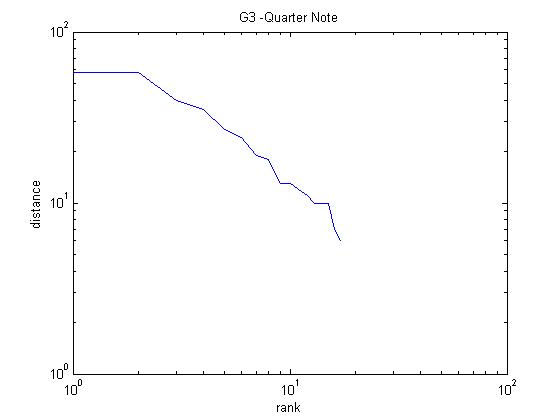


Figure 6 - Log-log plot for recurrence of note - G3

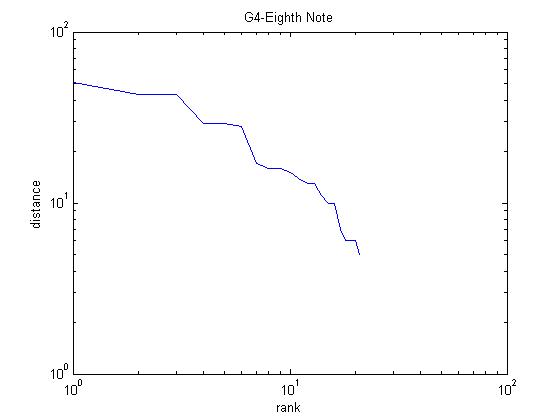


Figure 7 - Log-log plot for recurrence of note - G4

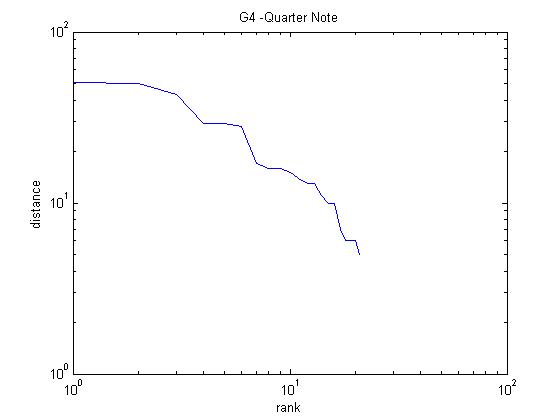


Figure 8 - Log-log plot for recurrence of note - G4

Below is the curve Fitting toolbox view for a sample note and its corresponding power law plot. A goodness-of-fit analysis was done using the same toolbox to compare the fitness of an exponenetial curve as opposed to power law fit. The measure of SSE was taken as the factor of comparison and it was seen that in all cases power law fit was found to be better than an exponential fit. For example in the case of the D4 Quarter note from Bach’s bwv 261 we can see that an exponential fit gave a SSE of 406.9267 whereas the power law fit shown below gave a SSE of 332.6786. We could also visually see that the power law equation fit the data better.

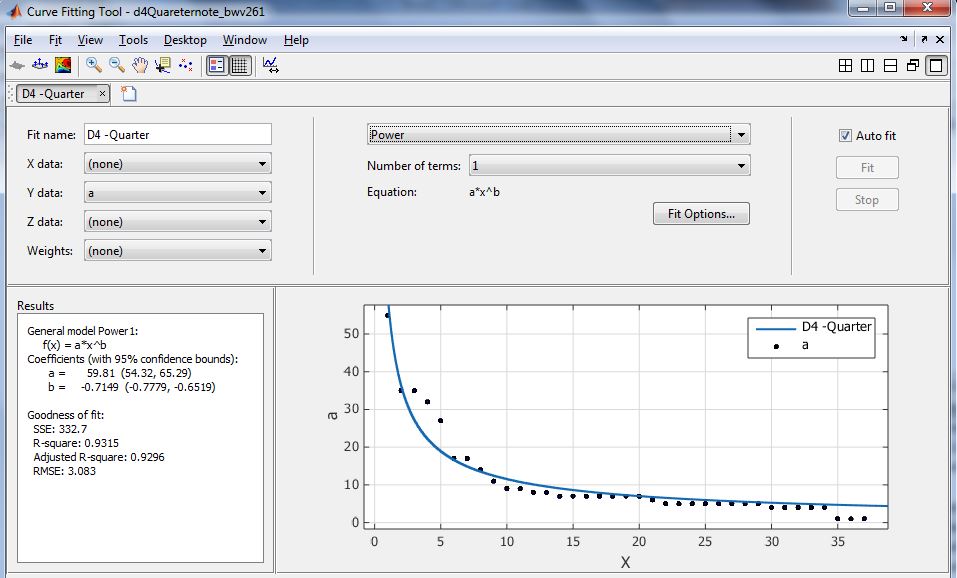
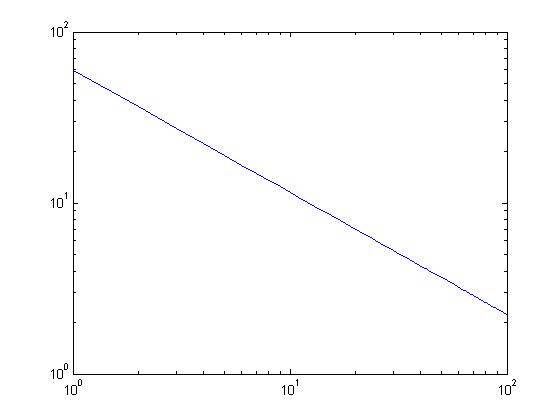


Figure 9 - Curve fitting toolbox for D4 Quarter note



distance

rank

Figure 10 - Log-Log plot for D4 Quarter note from Bach’s bwv261

This above procedure has been repeated for a number of notes and we have determined the fractal dimension to be the slope of the log-log plot. The table below provides a list of notes from Bach’s bwv261 chorale and Mozart’s K155.

|  |  |  |
| --- | --- | --- |
| **PIECE** | **NOTE** | **FRACTAL DIMENSION** |
| Bach/BWV261 | B4 Eighth | 0.5227 |
| D4 -Quarter | 0.7149 |
| E3-Eighth | 0.7272 |
| G4-Eighth Note | 0.5839 |
| F4-sharp Eighth | 0.5598 |
| D5 Quarter | 0.502 |
| G4 -Quarter | 0.5932 |
| G3 -Quarter | 0.6342 |
| C5-Sharp Eighth | 0.8196 |
| Mozart/K155 | A4 Eighth Note | 0.7761 |
| A4 Quartet Note | 0.8082 |
| B4 16th Note | 0.611 |
| D5 16th Note | 0.5817 |
| E5 16th Note | 0.6532 |
| F5-sharp 16th Note | 0.7761 |
| G4 Dotted Eighth Note | 0.6072 |
| F4-sharp 32nd Note | 0.5778 |
| G4 32nd Note | 0.5778 |

Table 3 - Fractal dimension of recurrence interval of different notes in two pieces

## Musical Fractals

Notes chosen from Bach’s bwv261 to be utilized in creating music are mentioned in table 2. The reason for choosing these particular notes is that they are the most recurring set of notes in this piece and they form a well-defined power law curve by themselves.

A long score of 1000 notes are creating after 10 iterations of these notes with a random starting point and with recurrence interval as mentioned by the power law distributed sequence give the time domain note definition as shown in the figure 13.

|  |
| --- |
| G3 sharp Eighth Note |
| C5 sharp Eighth Note |
| D5 Eighth Note |
| F4 sharp Quarter Note |
| G4 Quarter Note |
| E3 Quarter Note |
| D4 Quarter Note |
| B4 Eighth Note |

Table 4 - Notes that were chosen from Bach's bwv261 to create music



Figure 11 - The generated score based on Bach's bwv261

This score is fed into the python code that creates a MusicXML file for the entire score. An excerpt from the XML file looks like this –

<part id="P0c8214ebc2347cb1c31be6f84eff8c0d">

<measure number="0">

<attributes>

<divisions>10080</divisions>

</attributes>

<note>

<pitch>

<step>G</step>

<octave>4</octave>

</pitch>

<duration>5040</duration>

<type>eighth</type>

<notations/>

</note>

</measure>

<measure number="0">

<attributes>

<divisions>10080</divisions>

</attributes>

<note>

<pitch>

<step>B</step>

<octave>4</octave>

</pitch>

<duration>5040</duration>

<type>eighth</type>

<notations/>

</note>

</measure>

Figure 12– An excerpt from MusicXML file of score from figure 11

The following figure gives an idea about how this MusicXML file is interpreted by Myriad plug-in in a browser.

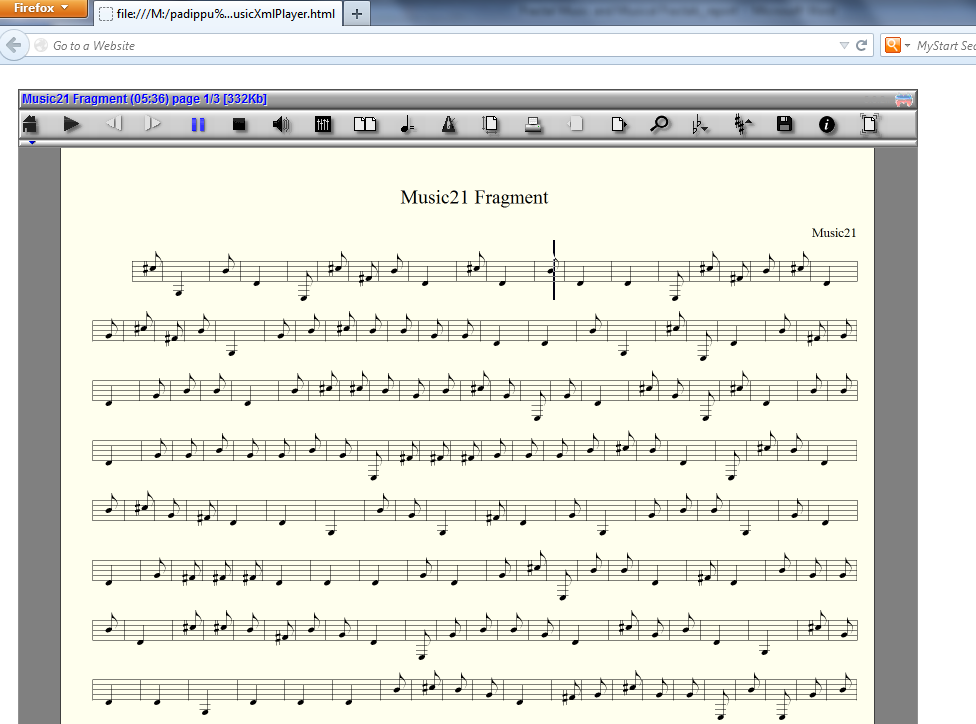


Figure 13 - A view of the music score using the Myriad music plugin

# Discussion

Looking at the graphs generated when conducting a fractal analysis of music, a definite trend towards power law nature was detected. Though this analysis was performed only on a small data set, we believe it holds true not only for other pieces by Bach and Mozart, but for more examples of classical music.

When it comes to the generation of music, while the music we created may not have been a great piece, it had more coherence than random noise. We think that music is characterized by more than just power law distributed note recurrences. It has a lot more to it. Research has revealed power law distribution in various other aspects of music and only a combination of all these would probably enable us to make quality synthetic music.

# Conclusion

From a cursory analysis of music, it can be noted that repetition and variation are two of the cornerstones of music. In nature, when repetition and variation in the way these repetitions occur are put together, fractals arise. Thus, it seems logical that fractals play at least some part in the structure of music.

Though we have only looked into the reoccurrence frequencies of the notes, it is possible to analyze other aspects of music using a complex systems viewpoint. We are of the belief that other aspect too will have some kind of fractal nature. Also, other kinds of music, apart from just classical music can be analyzed to increase the scope of the search. A comprehensive complex systems analysis of music will go a long way in helping us unravel the mysteries of music and enabling computer generated music so our robots may become composers.

# References

[1]. Harlan Brothers, *Structural Scaling in Bach’s cello suite no. 3,* Fractals 2007, 89-95.

[2]. Kenneth Hsu, Andreas Hsu, *Fractal Geometry of music*, Proceedings of National Academy of Science, USA, Vol. 87,February 1990, 938-941.

[3]. Michael Gogins, *Iterated Function Systems Music*, Computer Music Journal, Vol. 15, No. 1 Spring 1991, 40-48.

[4]. Omar López-Ortega, Shani Ioana López-Popa, *Fractals, fuzzy logic and expert systems to assist in the construction of musical pieces*, Expert Systems with Applications, Vol. 39, Issue 15, 1 November 2012, 11911-11923.

[5]. Tomoyuki Higuchi, *Relationship between the fractal dimension and the power law index for a time series: A numerical investigation,* Physica D: Nonlinear Phenomena Vol. 46, Issue 2, November 1990, 254–264.

[6]. Documentation for MusicXML, xsd and dtd files <http://www.musicxml.com/for-developers/>.

[7]. Music21 documentation <http://mit.edu/music21/doc/html/contents.html>.

[8]. Zhi-Yuan Sua, Tzuyin Wu, *Music walk, fractal geometry in music,* Physica D: Nonlinear Phenomena Vol. 221, Issue 2, 15 September 2006, 188–194.