Using Mathematics to Analyze, Manipulate, and Create Music

Dirk Schlingmann, Bradley Biera, Sarah Neary, David Tran, Vivian Tran
Honors Program
USC Upstate
800 University Way, Spartanburg, SC 29303
dschlingmann@uscupstate.edu

Abstract *— In this paper, we will demonstrate how we use mathematics and the technical computing software Mathematica to analyze, manipulate, and create music. Musical pieces like Beethoven's famous Pathetique Sonata 2nd Mvt, Adagio, are available as Musical Instrument Digital Interface (MIDI) files. We will show how to view the raw digital data in these music files and how to perform statistical analysis on the data. We will demonstrate how to manipulate the data to create different variations of the music, for example playing the Adagio backwards and reflected. We will also explain how to create music that is entirely based on computer code using mathematical integer sequences.*

Keywords —Music, Mathematics, *Mathematica*, MIDI Programming, Statistical Analysis

Introduction

Mathematics is an invaluable tool to quantify and study the intrinsic physical structure of sound and music. With computers and innovative technical computing software like *Mathematica*, it is now possible to digitize, analyze, manipulate, and create music or sounds. Once physical phenomena like sound and music are digitized or coded in the abstract virtual world of numbers and digits, it is possible to analyze and manipulate these numbers and their abstract structures and even create new abstract musical structures within this virtual world. These creations can then be transformed back or decoded into the physical world of sound and music. Digitized music is available in different file formats. We will focus on the Musical Instrument Digital Interface (MIDI) file format. In the MIDI format, musical elements like pitches and instruments are represented by integers. For example, the integer 57 represents the musical pitch "A3" or the musical instrument Trumpet. We will show how Beethoven's *Pathetique Sonata 2nd Movement*, Adagio is coded as a MIDI file within *Mathematica*, and how a statistical analysis can be performed on the data. We will demonstrate how we manipulate the Adagio data to create transformations or variations of the Adagio that are based on geometrical operations. Furthermore, we will explain how we can create music that is completely based on computer code using mathematical integer sequences.

Methodology

We wrote *Mathematica* programs to extract useful data, which we analyzed statistically and manipulated. Finally, we created our own data based on mathematical integer sequences to produce original music.

Statistical Analysis of MIDI files

The MIDI file of Beethoven's Adagio ("Beethoven-Pathetique Sonata 2nd Mvt (Adagio) midi file for Piano (midi)," n.d.) can be imported into *Mathematica*. Here is a sample of the Adagio data:

```
Sound[{SoundNote["G#3", {0., 0.25}, "Piano", SoundVolume -> 0.42], SoundNote["D#3", {0.25, 0.5}, "Piano", SoundVolume -> 0.45], SoundNote["G#3", {0.5, 0.75}, "Piano", SoundVolume -> 0.49], ...1618..., SoundNote["G#1", {145., 146.}, "Piano", SoundVolume -> 0.35], SoundNote["G#3", {145., 146.}, "Piano", SoundVolume -> 0.33], SoundNote["C3", {145., 146.}, "Piano", SoundVolume -> 0.33]}]
```

Sound has a list of 1624 SoundNotes as input. SoundNote["G#3", {0., 0.25}, "Piano", SoundVolume -> 0.42], turns on the pitch G#3 at time 0 sec and off at time 0.25 sec using the piano with volume 0.42 (1-max volume, 0-no volume). We wrote *Mathematica* programs to extract the pitch data {G#3, D#3, G#3, … , G#1, G#3, C3}. Then we analyzed the data statistically. Figure 1 shows how many times a pitch is played. In Figure 1, we used MIDI codes 32 (G#1), 48 (C3), 51 (D#3), 56 (G#3), 60 (C4), 61 (C#4), 62 (D4). For example, 60 (C4) is played 56 times in the Adagio. The statistical analysis on the pitch data gave the following results. The median is 56 (G#3), the mean is 56.83, the standard deviation is 9.29, the mode is 51 (D#3), and the range is 60. In a follow-up paper, we will compare this information with the data of other Beethoven piano pieces to find patterns or anomalies. We will also include comparisons with the musical data of other composers.

Manipulation of MIDI files

The Adagio data can now be manipulated to create variations or transformations. For example, we wrote *Mathematica* code that played the Adagio backwards and reflected every pitch with respect to the median G#3 (56). Thus, the reflection of C3 (48) is “E4” (64), and the reflection of G#3 (56) is G#3 (56). The reflection of D#3 (51) is C#4 (61), and so on. Here is a sample of the data of the Adagio played backwards and reflected

Sound[{SoundNote["E4", {0., 1.}, "Piano", SoundVolume -> 0.33], SoundNote["G#3", {0., 1.}, "Piano", SoundVolume -> 0.33], SoundNote["G#5", {0., 1.}, "Piano", SoundVolume -> 0.35], ...1618..., SoundNote["G#3", {145.25, 145.5}, "Piano", SoundVolume -> 0.49], SoundNote["C#4", {145.5, 145.75}, "Piano", SoundVolume -> 0.45], SoundNote["G#3", {145.75, 146.}, "Piano", SoundVolume -> 0.42]}]

Creation of MIDI files

We wrote computer programs that create music based only on integer sequences originating from mathematical formulas. For example, s[n]=n^2 creates the following integer sequence:

1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, ...

The MIDI standard only allows a maximum of 128 different pitches, and a grand piano uses only 88 keys that range from 21 (A0) through 108 (C8). To convert the sequence s[n] to appropriate piano pitches, we use the modulus function. For example, mods[n]=Mod[s[n], 88]+21, creates the following sequence:

22, 25, 30, 37, 46, 57, 70, 85, 102, 33, 54, 77, 102, 41, 70, ...

The *Mathematica* code:

Sound[{SoundNote[#[[1]], #[[2]], #[[3]], SoundVolume -> #[[4]]] & /@ Table[{Mod[s[i], 88] + 21, 1/10, 1, 1}, {i, 1, 100}]}]

Creates the following MIDI song:

Sound[{SoundNote["A#0", {0., 0.1}, "Piano", SoundVolume -> 1.], SoundNote["C#1", {0.1, 0.2}, "Piano", SoundVolume 1.], SoundNote["F#1", {0.2, 0.3}, "Piano", SoundVolume -> 1.], SoundNote["C#2", {0.3, 0.4}, "Piano", SoundVolume -> 1.], ...94... , SoundNote["A1", {9.7, 9.8}, "Piano", SoundVolume -> 1.], SoundNote["F#3", {9.9, 9.9}, "Piano", SoundVolume -> 1.], SoundNote["F5", {9.9, 10.}, "Piano", SoundVolume -> 1.]}]

We will present more sophisticated music that has been completely created through *Mathematica* programming.

Conclusion

This paper presents our innovative work where we demonstrate how recently developed software tools can enhance the analysis, manipulation, and creation of music. During our oral presentation, we will play MIDI audio files that we created by writing programs using *Mathematica*.

References

Beethoven-Pathetique Sonata 2nd Mvt (Adagio) midi file for Piano (midi). (n.d.). Retrieved from http://www.8notes.com/scores/3062.asp?ftype=midi

Figures

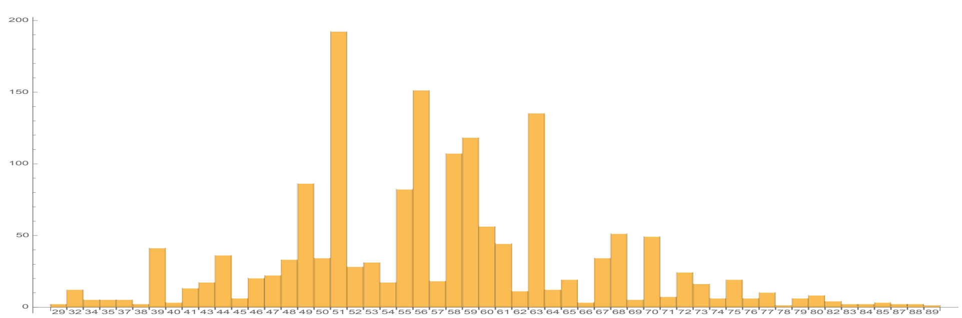

Figure 1. Number of occurrences of MIDI pitches in Beethoven’s Adagio