Computing Approaches to Music Generation

Kylee Hartman-Caballero

April 8, 2021

1 Introduction

Computing processes and the fields in which we are able to apply them to have broadened and become more complex between the innovation of the twentieth century and the present. Typically, one thinks of these applications as being in fields related to STEM, but may overlook the applications to humanities-related fields like the arts. A discipline that has been capitalizing off of STEM thought since its inception is music. Music, at its core, is humans harnessing the properties of sound waves into subjective experience.

Before exploring the classical and quantum computing applications in music, we should take care to differentiate between music production and music generation. "Music production" has evolved in meaning since recorded music became mainstream, but in current use, we think of music production as being the act of a human sitting behind a computer screen adding instruments, fluctuating tempo, distorting a singer's voice, etc. through an interface like Apple's Garage Band. However, "music generation," as we will define here, is the actual composition of musical lines (melodies and harmonies). Thus, we will be exploring a couple techniques from both the classical and quantum computing perspectives of how music can be generated.

2 History of Music Generation

Music generation by computing methods can be traced back to the 1940s. The Australian Council for Scientific and Industrial Research (CSIR), which spent most of its resources during that time researching for World War II, did invest in non-wartime research. Notable outcomes of their research is using sound to track the progress of running a program on an early computer. A short time later in 1951, Geoff Hill, a researcher at CSIR with perfect pitch (that is, the ability of a person to recognize a pitch of any note or produce any note) programmed a computer to playback a tune. A menial task for a computer today, in 1951 the limits of contemporary computers were already being tested. In the same decade halfway across the world, researchers at the University of Illinois at Urbana-Champaign produced the Illiac Suite, a computer generated, four-movement suite composed by a computer using a different technique and experiment for each movement. Finally worth mentioning is Max Mathews developing MUSIC, the first widely used program for sound generation, on an IBM 7094 in 1957. The technology available for common use has continued to expand and innovate into the advanced techniques we have today.

3 Classical Methods

While there are many ways that one can approach music generation through a classical computing lens: we will consider two here. First, let us set up our environment. We have a scale (the scale can be in any key - that is, center of pitch - but we will consider C4 Major here):

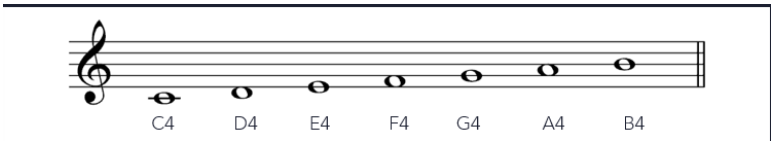

Figure 1: The C4 Major scale on the treble staff

It is important to note that Figure 1 does not show the top note of the scale, C5, which would be on the next space above B4. A widely used approach to programming classical computers to compose music is setting up a system where, given a scale like the one above, notes are randomly generated based on a probability model. Such a model that is useful is a Gaussian function.

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

This type of function, when used represent the probability distribution of a random variable is given by the function above with expected value μ, and variance σ^2. This model can bias the system into picking notes that are closer to the center of the scale (F4 and G4 in the case of the major scale in Figure 1). This can be useful when trying to generate voice parts for alto and tenor ranges. The reason this is useful is because by the rules of Western classical voice leading, which is very much the golden standard for how music is written today, the inner voices like the alto and tenor are not allowed to move more than an interval of a fifth in one beat. Biasing the computer to choose notes in the middle of the scale is helpful then because these notes are the root and/or dominant (the first note and the fifth note, respectively) of the most commonly used chords in the relevant key. What this means is that the inner lines will, by consequence of the rules of the Western writing style, rarely leave these notes. However, this method does not lend itself to writing a piece with melodic variation. The lines generated by using a Gaussian function may not be the most interesting to listen to.

To counter this, we can use another method called a Markov chain to make a more sonorous melodic line. A Markov chain is a stochastic process where the probability of an event depends only on the event before it. Using the C Major scale, let us develop such a chain using a list of rules.

Rule 1: If C4, then E4, G4, or A4
Rule 2: If D4, then C4 or F4
Rule 3: If E4, then G4
Rule 4: If F4, then A4 or D4
Rule 5: If G4, then C4 or B4
Rule 6: If A4, then B4 or D4
Rule 7: If B4, then C5
Rule 8: If C5, then F4 or E4

The method to use these rules is that a user would provide a starting note, say C4, for example, and the computer would then start at *Rule 1* where there would be a 33% chance of any of the possible notes being chosen at random. Say that E4 is randomly chosen, then the next rule followed would be *Rule 3*, which only provides the possibility of choosing G4, thus leading to *Rule 5*, and so on. A piece composed by this method looks like this:

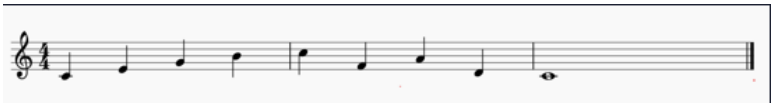

Figure 2: Piece composed by following a Markov chain

We could also consider a situation where we rewrite the rules so that each current note only leads to an adjacent note, either above or below. So, for example, *Rule 4* would become *"If F4, then E4 or G4."* This process is called a random walk, and will be the framework for our quantum consideration.

4 A Quanutum Algorithm

Moving into the quantum approach, we must first acknowledge some problems that arise. Since quantum states cannot be observed without collapsing and because a quantum system cannot be replicated per the no-cloning principle, we cannot replicate a system from previous steps in the algorithm. However, in the upcoming example, there is some information we can store in classical memory. In a classical approach, we are on one note only at any given time. In a quantum walk, though, we can be on all possible notes at a given time because of superposition. So in *Rule 1* of the Markov chain, we could be on C4, E4, G4, and A4 at the same time. Researchers at the University of Plymouth have set up a system that takes advantage of this mechanism. The system contains a server and a client that is accessed through an SSH. The server runs the algorithm and sends measurements to the client. The client turns those measurements into musical notes and encodes them as a MIDI file, which is a file type that allows visualization of the music with a third-party application. The quantum walk algorithm can be summarized like this: Imagine a cube like the one below.

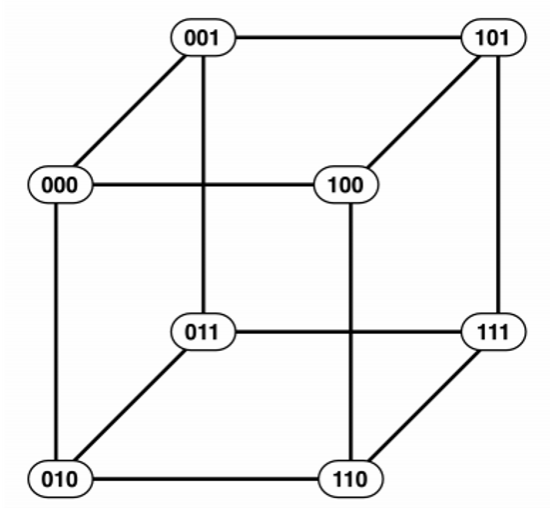

Figure 3: A cube demonstrating a quantum setup. Credit to Eduardo Reck Miranda at the University of Plymouth.

Each of the vertices is represented by a three bit long binary number. Three qubits are used to represent the vertices (call these q0, q1, and q2 - "input qubits") and two qubits are used to represent possible routes that can be taken by the metaphorical "walker" following the algorithm (call these q3 and q4 - "die qubits"). Each vertex is connected to three other vertices that vary by one bit from the given bit. The algorithm is as follows:

The input qubits are set up to represent the "nodes of departure," meaning the possible routes from from one vertex to another, and the die qubits are set up in superposition. The input qubits are measured and the measurement is stored in classical memory. The values of the die qubits determine how the input qubits are inverted. The measurement result sets up the next execution. Note that in this algorithm, a starting note and duration must be provided, which is done by how the input qubits are set up before the first measurement. To turn these results into music, each pitch and duration has a three-bit binary number assigned to it. The circuit runs twice every time; once to choose a pitch and once to choose a duration. If one of the four "pause" binary numbers is chosen, then the pitch that was measured is discarded and the duration outcome specifies how long the ensuing rest will be.

It should be noted that this can also be done by a classical computer. The trade-off for an easier and more readily available implementation is more computation time and more necessary memory. When this computation was done by the University of Plymouth researchers, they used the quantum computer located just outside of Los Angeles, California, and accessed it via the cloud.

5 Practicality and Conclusions

From a music analysis perspective, these two methods create very different compositions. In our example of the Markov chain, we used a major scale, the most tonal and standard scale in use in Western pedagogy and practice. Because of this implementation, the pieces created will always have a sense of "home." A listener will feel like there is a note that should be returned to and

that sounds complete (we call this the "tonic"). The quantum algorithm will tend to produce pieces that sound more atonal and chromatic a la Arnold Schoenberg. The style of music the quantum algorithm tends to produce is characteristic of 1920s European expressionism, and is not utilized much today. Despite this, an equally complex and atonal piece could be produced by the Markov chain, although it would take an intricate rule set up by the person preparing the algorithm. What we can conclude from this is that the Markov chain is heavily influenced by user input and can be bent in many different ways to cater to the style of music desired by the user. The quantum algorithm, however, only takes one input note and duration from the user, so the computer really is doing a vast majority of the work.

It is also worth noting the cultural implications of what each algorithm tends to produce. From a Western perspective, the atonal product of the quantum algorithm is "avant-garde," and some might even say "odd." However, we must be careful when making these judgments. Traditional music from other cultures, such as East and Southeast Asian cultures, do not generally use the same sense of a "home" feeling that we get from Western classical practice. These styles of music are not inferior, simply different from what we may be used to. So when considering what one wants to get out of using either the classical or quantum algorithms, it is not always a matter of the quality of the piece produced.

Research on the intersection of, specifically, quantum computing and music application is still slim, as much of the available funding is put towards application is medicine, engineering, or other STEM fields. Another hurdle to overcome is the lack of research labs actually studying the subject because it is so young and the availability of quantum computers, of which there are only 15 in the world. Perhaps as we begin to understand quantum computing and computing in general even more than we already do, then the field will be much more open to abstract and artistic applications.

References

[1] Eduardo Miranda Alexis Kirke. *Guide to Unconventional Computing for Music*, chapter 5. Springer International Publishing, 2017. Chapter titled: Experiments in Sound and Music Quantum Computing.

[2] Eduardo Reck Miranda. Quantum computer: Hello, music! Technical report, Interdisciplinary Centre for Computer Music Research, University of Plymouth, 2021.

[3] Researchers to investigate quantum computing for the music industry. Web, December 2020. From the Department of Computer Science at the University of Oxford.

[4] Cade Metz. Quantum computers don't make sense. but this one makes music. Web, July 2016. From "Wired.".