Higher-Order Euclidean Sets || SMT || November 2024

Over the last thirty years, a considerable corpus of research has been built around the notion of "maximally-even sets" and "Euclidean sets". One early milestone was Clough and Douthett's 1991 article, which established the mathematical properties of maximally even sets. A second landmark was Godfried Toussaint's 2005 paper entitled "The Euclidean Algorithm Generates Traditional Musical Rhythms", which proposed geometrical analogues to Clough and Douthett's method. Justin London's concept of "non-isochronous meters" and Jay Rahn's idea of "diatonic rhythms" represent other fruitful directions in which this research has progressed. One of the most appealing aspects of this research is that it has allowed us to see and hear connections between Western tonal music, popular music, and other music traditions around the globe.

My paper presents an algorithm I've developed that generates what I call "higher-order Euclidean sets", or "hypereuclidean sets". Inspired by the concluding paragraphs of Clough and Douthett's article, my approach produces a very diverse and complex array of structures by applying the maximally-even algorithm recursively. Today, I'll describe methods that create, analyze, and find higher-order Euclidean sets and trace their derivation or ancestry. I will also present musical examples that could point the way for composers wishing to use my methods. All my material is available for free at my web site. It can be found by scanning the QR code on your handout, or on the title screen of the presentation.

Familiar to many, the classical Euclidean set is beautifully elegant in its simplicity and impressive in its ability to model a surprising variety of salient musical structures around the world. Clough and Douthett's related formula for generating *maximally even sets* can be written as shown in **[example 1]**. Here, capital *E* sub-alpha denotes what I will call a "first-order" set, where k_1 represents the desired number of onsets, and n_1 is the set's span – or, total number of *potential* onsets. *i* is simply an iterator, and the square brackets indicate a floor function – to "round down". I show a simple example for those of you who, like me, cannot just visualize this in your heads. There are only two constants (*k* and *n*), and one iterator (*i*) to keep track of. By setting k_1 to 7 and n_1 to 12, the method generates a maximally-even set [0 1 3 5 6 8 10].

Toussaint proposed another way to create this structure, using an algorithm related to Euclid's method for finding the greatest common divisor between two integers. We first line up 7 attack points (represented by "x"es) followed by five rests (represented by periods). Then we take the five rests and move them below the attack points, continuing in this manner until we have no remainder, or the remainder is 1. Note that there is one little span left over in step 4, indicating that the numbers 7 and 12 are relatively prime – meaning they have no common divisors other than 1. Read vertically, the resulting set is [0 2 3 5 7 8 10], which is exactly the same as the maximally-even set [0 1 3 5 6 8 10] if you subtract 2 from each element, modulo 12. While it is fair to say that Clough and Douthett were somewhat more inclined towards the pitch domain, Toussaint was more oriented towards rhythm – but, the method can produce structures in either.

Ableton Live's built-in Euclidean set generator can make these sets, as shown in **example 2**. Here I produced the both instances of the set we just generated. They are shown here arranged in a canon, of interval 2. Here is what this sounds like, played four times in a row. **[play musical example, 0:13 long].**

The canon reminds us that by rotating all the onsets clockwise by two positions, we create a new pattern of onsets. Toussaint called the family of all these rotations a "necklace": in his words, "a rhythm *necklace* is the inter-onset duration interval pattern that disregards the starting point in the cycle". Both rhythmic cycles in the Ableton example are members of the same necklace. For this particular necklace $E_1(7)$, 12, Toussaint cites many different rhythmic cycles or timelines used throughout the world.

Music theorists have used this technique to model rhythmic structures in music, particularly Brad Osborn, who has applied it to songs by the band Radiohead. Osborn aptly describes such rhythms as "metric-ish". He identifies at least dozen Euclidean and maximally even rhythms in Radiohead's music, from tracks such as "Pyramid Song", "Codex" and others. Osborn positions Euclidean rhythms as an appealing middle-ground between regular, repeating pulse patterns and unpredictable, indeterminant rhythmic structures.

For all their utility, maximally-even sets and Euclidean sets can only produce structures that are very "smooth", by which I mean sets that differ by at most 1 in their inter-onset intervals. Indeed, this is part of the very definition of a maximally-even set. But when applied recursively, the algorithm can produce much more variety. At the end of their paper, Clough and Douthett briefly investigated what they called "second-order" or "high-order" maximally even sets. However, they only wrote about those which had a place within the classical tonal system: that is, diatonic scales, seventh chords and triads. They also only considered only sets of depth 2 and 3, meaning they did not recursively apply their algorithm more than three times. Could we study *all* possible sets their methods create? What about producing sets of depth 4, 5 or more? The answers to both questions is _yes_.

I will outline my method by walking through an example. Consider the clave timeline, a ubiquitous rhythmic cycle found in a wide variety of music from sub-Saharan Africa, Brazil, and Cuba, and found in many genres such as salsa, conga, and Afro-Cuban jazz. This cycle is shown in the blue box in **example 3**. It has onsets [0 3 6 10 12] and an IOI of < 3 3 4 2 4 >. Because it has **three** different inter-onset intervals, it cannot be Euclidean or maximally even. However, it is easily derived from an intermediate set. First, consider the Euclidean set $E_1(14^0)$, 16. I will label this the *alpha* set. This structure has onsets at every time-point except at 7 and 15. The exponent indicates how many time-points the set is rotated around the circle, clockwise – in this case, zero. The blue dot indicates the first element of the set, which corresponds here to time-point zero because the rotation is 0. Now, take these fourteen **onsets** and create another Euclidean set $E_1(5^9)$, 14. This is the *beta* set. Notice that the first *k* value in the *alpha* set – 14 – has now become *n* in the *beta* set. Rotate the *beta* set by 9 as shown in the bottom yellow box to yield [0 3 6 9 11]. Finally, use this as an *index set*, taking these exact elements in the *alpha* set of span 16 and discarding the rest. This produces a set of [0 3 6 10 12], one of the most common representations of the clave timeline, shown in the blue box. Because we rotated first, **then** thinned, I call this the "rotate-first method". For the sake of convenience, I label this $E_2(14^0, 5^9)$, 16 where again, the exponents give the rotations of each k_n. The depth – what I call *p* – is 2, indicated as a subscript to capital E.

This is a straightforward example of a second-order Euclidean set, inspired by the examples Clough and Douthett outlined at the end of their paper. In *classical* first-order Euclidean sets, *k* indicates the number of onsets desired. In my method I designate k_1 as the number of onsets in the first-order set, and k_n as the number of onsets in the final, desired set. Since this is a second-order set, we have only k_1 and k_2. How many sets can be produced using the parameters that generated the clave timeline? When *n* equals 16, k_2 equals 5, and k_1 varies from between 15 and 6 inclusive, we produce exactly 2560 second-order sets. We will see why this is in a moment, but for now it is sufficient to notice that only 14 of these sets make the clave timeline. These are shown in **example 4**.

Let us derive the clave timeline another way. This time, we create a first-order Euclidean set $E_1(13^0)$, 16 as shown in the top yellow box of **example 5**. Rotate it by 7 to create the *alpha* set. Create a second Euclidean set $E_1(5^0)$, 13 shown in the yellow box on the bottom. This is the *beta* set. Use this set as an *index set* on the *alpha* set. Finally *rotate* this derived set by 3, as shown in the bottom-right blue box. Notice that this time, we thinned the *alpha* set by the beta set ***first***, and ***then*** performed the rotation. I call this the subtract-first method. It produces the clave timeline as $E_2(13^7, 5^3)$, 16. Like the rotate-first method, the subtract-first method yields exactly 2560 second-order sets, but now only 10 of them have the desired onset pattern as shown in **example 6**.

How does this algorithm work? First, it creates what I call a *slate* of all *n* and k_n values. **Example 7** shows the *slate* that produces clave timelines. It is the same for both rotate-first and subtract-first methods. S*lates* simply iterate k_{n+1} values such that all combinations are produced between *n* and k_n. Slates of third- and higher-order Euclidean sets are a little more complicated than the one shown in **example 7**. Next, the algorithm produces all rotations of the sets involved, which means that unfortunately, complexity increases exponentially as depth – or "*p*" – grows linearly. Then, the algorithm goes about calculating the many derived sets. We can now see why the method generates 2560 sets for the clave timeline parameters, because each row produces *n* times *n*, or 16 times 16 sets (=256). The sum of all the rows is 256 times 10, or 2560.

At this point my paper makes a methodological swerve towards perception and cognition. A recent insightful article on microtiming in *Music Theory Spectrum* by Anne Danielsen, Mats Johansson and Chris Stover argues that performed rhythm often involves an:

> ...***interaction*** between actual sounding events and “virtual” structuring mechanisms, such as meter, pulse, subdivision, or stylistic figures (the latter encompassing both style-specific rhythmic patterns and melodic-rhythmic formulas), which the perceiver projects onto sounding events.

This suggests that rotating an onset-pulse pattern may not produce an equivalent performance, because “points” on the circle might be more like little “clouds”. A related critique comes from Mark Gotham, who notes that the visual representation of timelines on a circle can “lead to some quintessentially spatial relations that do not necessarily analogize well to temporal patterns”. The ease of rotating attack points around circle diagrams belies the fact that different rotations of the same onset pattern can be perceived as very different rhythmic identities. Yet another cognitive issue one might raise, is that in higher-order Euclidean sets, the “parent” set can stand at quite a distance from the “child”, “grandchild,” or “great-grandchild”. What perceptual relevance – if any – do the intermediate sets have?

To gain perspective on this question, consider the timeline Steve Reich made famous in his “Clapping Music”. This is an interesting case of a second-order Euclidean rhythm that only includes IOIs of 1 and 2. It can be derived in two different ways using the subtract-first method, and three ways using the rotate-first method. We see one of these derivations in **example 8**. It would appear that the first-order set from which the Clapping Music timeline is derived, only offers the ability to *remove* one onset from the final second-order pattern. Consequently, the cognitive relevance of this first-order set $E_1(11^7)$, 12 might seem remote and perceptually unrelated to the derived Clapping Music timeline. This suggests that while higher-order Euclidean rhythms model a very impressive variety of structures – including all six of Toussaint’s “good” timelines as second- and third-order structures – we might want to exercise caution when using them as analytical mechanisms.

Now, let me suggest another use of the method which I personally find appealing: namely, algorithmic composition. If we generate child, grandchild and great-grandchild sets using the Clapping Music timeline as parent, we can arguably produce some compelling rhythmic structures. In the following etude, I made a family of structures using the subtract-first method, shown in **example 9**. By successively rotating the three higher order sets in canon, I produced a short piece with an interesting musical property, shown in **example 10**. About two-thirds of the way through, the top part becomes symmetrical, having an IOI of < 4 4 4 >. This is the only configuration of parent and child sets that can produce this symmetry. The symmetrical set is in a red box on the score. We hear the entire piece, without repeats now. **[musical example is 0:50 long]**

Higher-order Euclidean algorithms can also generate extremely intricate, interrelated rhythmic polyphony at very high *n* values. Let’s look at **example 11**, which shows another algorithmically generated composition using sets up to depth 5. All of the *k* values are relatively prime to their preceding value, *i.e.*, 105 is relatively prime to 128, 71 is relatively prime to 105, etc. The rotational values were chosen somewhat *ad hoc*. The piece exhibits a quality that Dimitri Tymoczko describes as feeling that there is structure, without necessarily knowing what that structure is. Perhaps it is the mind's continual search for periodicity that enlivens the texture of this exercise. An amusing "gap" occurs about two-thirds of the way through, when no part has an onset. **[musical example is 1:07 long.]**

So, in conclusion: Although higher-order Euclidean sets can easily model many complex static timelines and pitch structures, there are potential cognitive issues that suggest we might approach their use as an analytical tool with caution. Their future may lie in algorithmic composition, because the method can produce potentially compelling musical structures. One of the main roadblocks preventing more comprehensive study has been the lack of an algorithm that produces all the possible sets to a particular depth. It is not currently known whether an algorithm exists that runs in less than exponential time. [$O(n^p$]. With such an approach now in place, we may begin to explore these sets more methodically.