ChordDyn: Generation of Novel Chord Progressions via a Musically-Inspired Chaotic Mapping

CSCI 5446, Spring 2023 — Zach Atkins

May 4, 2023

1 Introduction

We propose *ChordDyn*, a novel method for generating chord progressions which does not depend on existing compositions, complex rules, or human input. *ChordDyn* utilizes a chaotic mapping onto a set of symbols to generate chord progressions. The symbols and their arrangement are prescribed by the *Tonnetz*, a representation of musical pitches as vertices on a doubly-periodic simplectic mesh; the triangular faces of this mesh are triads, or collections of three pitches. *ChordDyn* uses the nonlinear double pendulum as the chaotic attractor of interest, as it has two loosely-coupled periodic degrees of freedom, and maps a given trajectory onto the *Tonnetz*. The sequence in which each triangle on the mesh is visited is used to generate the chord progressions. The resulting chord progressions show strong voice leading and low perceptual distance under metrics which correlate with positive listening experiences.

1.1 Related Work

Computer-aided music composition has a rich history, from the generation of phrases using strict rules, such as generative grammars in [7], to approaches using machine learning techniques, such as genetic algorithms in [8] or long short-term memory (LSTM) networks in [3], to synthesize existing compositions. Many of these techniques are limited by the need for a large corpus of existing compositions to train on, which raises a number of practical and ethical issues, or by the need for a human composer to provide a starting point and feedback. Other approaches, such as ChordAIS [11] and ConChord [2], utilize a combination of rule-based and statistical methods centered around Bernardes' Tonal Interval Space, which examines the relationships between the Fourier coefficients of *chroma vectors* [1]. These approaches are of particular interest to our work, as the Tonal Interval Space can be viewed as a continuous extension of the chord space we utilize, the *Tonnetz*.

Few attempts have been made to utilize chaos in computer-aided music composition. Most notably, Dabby utilized existing compositions to define a chaotic mapping to generate musical variations, leveraging the attractor's sensitivity to initial conditions to reorder pitches, chords, and rhythmic events according to the difference between two trajectories on a Lorenz attractor [5].

2 Methods

2.1 Pitch Class Set and Tonal Interval Vector Representations of Chords

In order to represent music computationally, we consider notes as pitch classes, a mathematical abstraction of pitches which ignores octave information. Each pitch class is represented as an integer modulo 12. Conventionally, the pitch class of a note is the number of semitones above the note C. Chords are represented as pitch class sets (pc-sets), a collection of pitches represented as a set of integers $S \subseteq \mathbb{Z}_{12}$. See Figure 1 for an example of the pc-set representations of several chords. Note, the pitch classes 10 and 11 are represented with the symbols χ and ϵ, respectively.

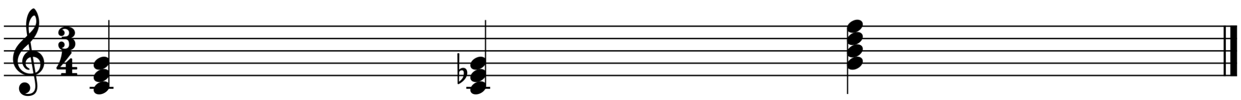

Figure 1: Example pc-set representations of several chords: C major (C), consisting of C, E, and G (left); C minor (Cm), consisting of C, E flat, and G (center); and G major seventh (G7), consisting of G, B, D, and F (left).

While the pc-set representation of chords is useful for our chaotic mapping, it does not capture the harmonic relationships between chords. To that end, we will utilize the Tonal Interval Space (TIS) as proposed by Bernardes in [1]. The TIS represents each pc-set by a *tonal interval vector* (TIV), the weighted discrete Fourier transform of its *chroma vector*. The chroma vector of a pc-set is a 12-dimensional binary vector, where each entry represents the inclusion of the corresponding pitch class in the collection. Figure 2 shows the *chroma* vectors of the chords in Figure 1. The

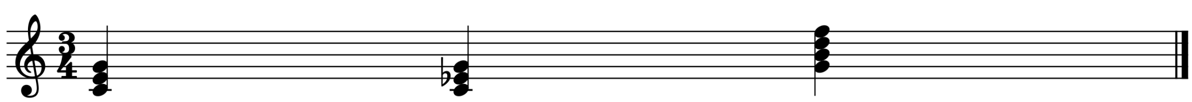

Figure 2: The chroma vectors of the C major chord (left), C minor chord (center), and G7 chord (right). By representing each included pitch as a separate dimension, the chroma vector can be interpreted as a discrete signal. Using the DFT on these signals yields a *Tonal Interval Vector* in frequency space, which encodes many harmonic relationships.

chroma vector can be viewed as a discrete signal, motivating the use of the weighted DFT to define the corresponding TIV as

$$T(k) = \frac{w(k)}{\bar{c}} \sum_{n=0}^{11} c(n) e^{-\frac{2\pi i \cdot nk}{12}}, \quad k \in \mathbb{Z} \tag{1}$$

where $\bar{c} = \sum_{n=0}^{11} c(n)$ is the energy of the chroma vector [1]. While $T(k) \in \mathbb{C}$ is defined for all $k \in \mathbb{Z}$, we will only consider $k \in \{1, \dots, 6\}$, as $T(0)$ is the sum of the signal, which is normalized to 1, and the DFT of a real-valued signal is symmetric about the midpoint of the length of the signal. Each of the entries in the TIV encode different intervals between the notes in the chord, as shown in Figure 3. The weights w scale each interval according to its harmonic importance, with $w = [2, 11, 17, 16, 19, 7]$ as proposed in [1] according to empirical listening tests.

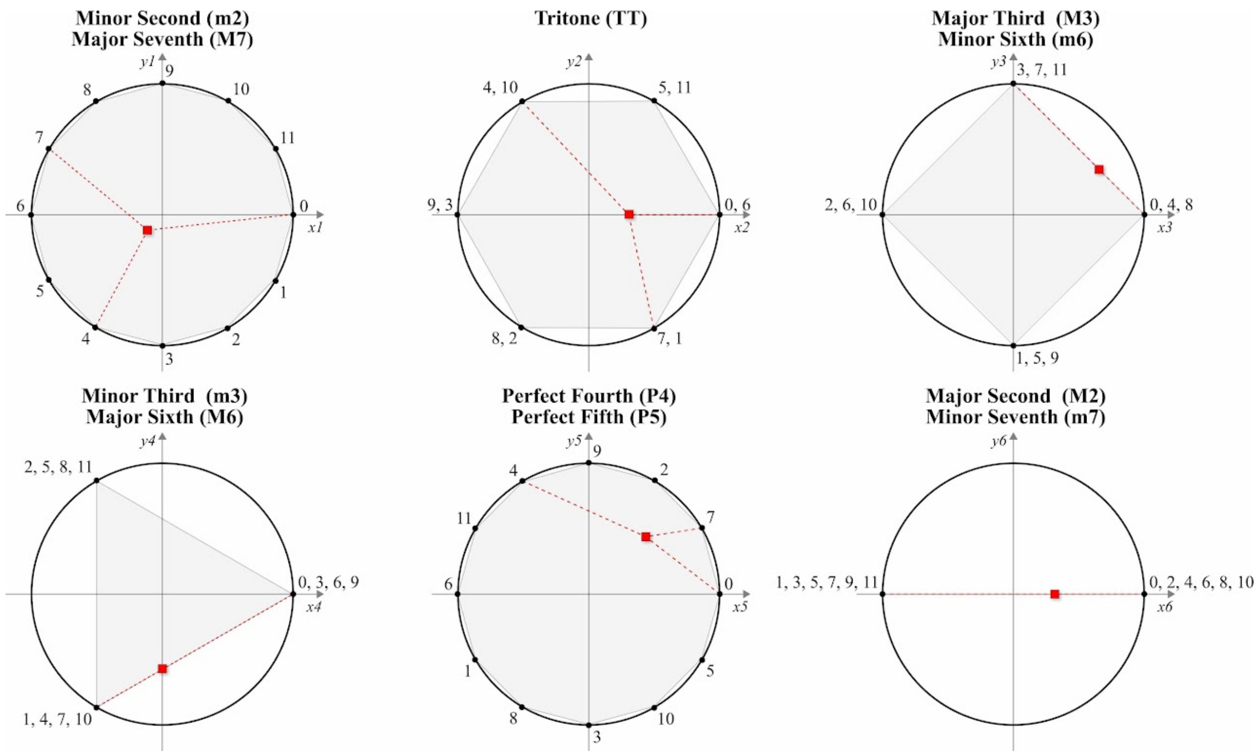

Figure 3: Illustration from [1] of the $T(k)$ values and corresponding harmonic intervals for the C major chord (left-most chord in Fig. 1 and 2). The real and imaginary parts of $T(k)$, $k = 1, \ldots, 6$, are plotted on the x- and y-axes, respectively, as a red square in left-to-right, top-to-bottom order. The shaded regions display the range of possible values, bounded by regular polygons with vertices given by the TIVs of the single-note chroma vectors. Not to scale.

We consider three metrics for comparing TIVs which have corresponding musical interpretations. The L_2 norm of a TIV represents the *consonance* of the chord [2]:

$$||T|| = \sqrt{\sum_{k=1}^{6} \left|T(k)\right|^2}. \tag{2}$$

While useful for evaluating general chords, the consonance $||T||$ of major and minor chords are identical; since all chords on the *Tonnetz* are major or minor chords, we will not consider this in our evaluation. The Euclidean distance between two TIVs represents their *perceptual distance* [2]:

$$d(T_1, T_2) = \sqrt{\sum_{k=1}^{6} \left|T_1(k) - T_2(k)\right|^2}. \tag{3}$$

The distance d is small for chords which are parsimonious, or share the majority of their notes. This is useful for comparing two TIVs with the same energy, or number of notes. The angle between two TIVs measures how well two pitch profiles are aligned [13]:

$$\theta(T_1, T_2) = \cos^{-1} \frac{\Re(T_1 \cdot T_2)}{||T_1|| ||T_2||}. \tag{4}$$

The angle θ is useful for evaluating the harmonic function of a chord in a key [2]. Using these metrics, we can express a number of musically relevant relationships between chords, which we discuss further in Section 3.

2.2 The *Tonnetz*

Toward the goal of producing tonally coherent chord progressions, we will utilize the *Tonnetz*, a representation of musical pitches as vertices on a simplicial mesh [16], originating from Leonhard Euler and later made popular by Hugo Riemann in the 19th century as a component of Riemannian music theory. The *Tonnetz* represents each of the 12 pitch classes as vertices on a simplicial lattice, wherein each of the three simplicial axes orders the notes according to a given interval [16]: the minor third (3), major third (4), and perfect fifth (7). Then, each triangle in the lattice represents a chord, given as the pc-set of its three corners. This construction results in edge-adjacent triads in the *Tonnetz* sharing two notes and vertex-adjacent triads are sharing one note; this property ensures that chord progressions resulting from a continuous trajectory on the *Tonnetz* should exhibit strong parsimonious voice-leading and have small Euclidean distances in the TIV space. Figure 4 shows the *Tonnetz* with the pitch classes labeled. Since we consider pitch classes to be periodic, the *Tonnetz* is a doubly-periodic domain. We specifically seek a flat, rectangular, and doubly-periodic domain, so our representation of the *Tonnetz* has a large number of repeated vertices, which can be interpreted as different inversions, or orders, of the same chord. This fact will be important later when we discuss the mapping between the chaotic trajectory and the *Tonnetz*. In general, however, the *Tonnetz* can be formulated as the twisted product of a circle and a higher dimensional sphere, in our case another circle, with no repeated vertices [16]. One important limitation of the *Tonnetz* is that it is restricted to modern Western tonal harmony, and particularly, 12-tone equal temperament [16]. It may be possible to define a similar space for other tuning systems, but we leave this for future work.

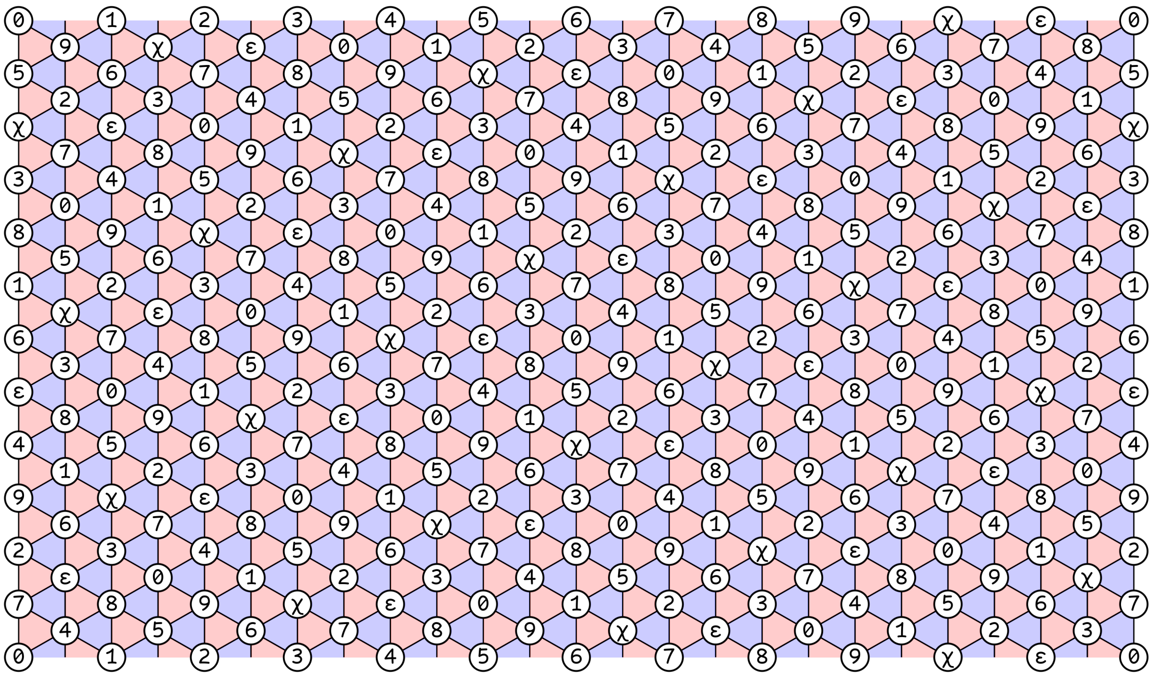

Figure 4: The *Tonnetz* with the pitch classes labeled. The major and minor triads are shown in red and blue, respectively. Both the top and right side contain duplicated phantom notes for visual clarity, as the edges are both periodic.

2.3 The Double Pendulum Chaotic Attractor

Our method requires an appropriate chaotic attractor as well as a mapping to generate chord progressions. We use the double pendulum as our chaotic attractor, as it has two loosely-coupled periodic degrees of freedom to be mapped onto *Tonnetz*. The double pendulum system consists of two masses, m_1 and m_2, ordered by their proximity to the origin, connected to the origin and each other by rigid rods of length ℓ_1 and ℓ_2. The angles θ_1 and θ_2 are measured counter-clockwise from the negative y-axis, and the angular velocities ω_1 and ω_2 are shown as red arrows with dashed lines in Figure 5. Mathematically, the double pendulum kinematics are described by the following

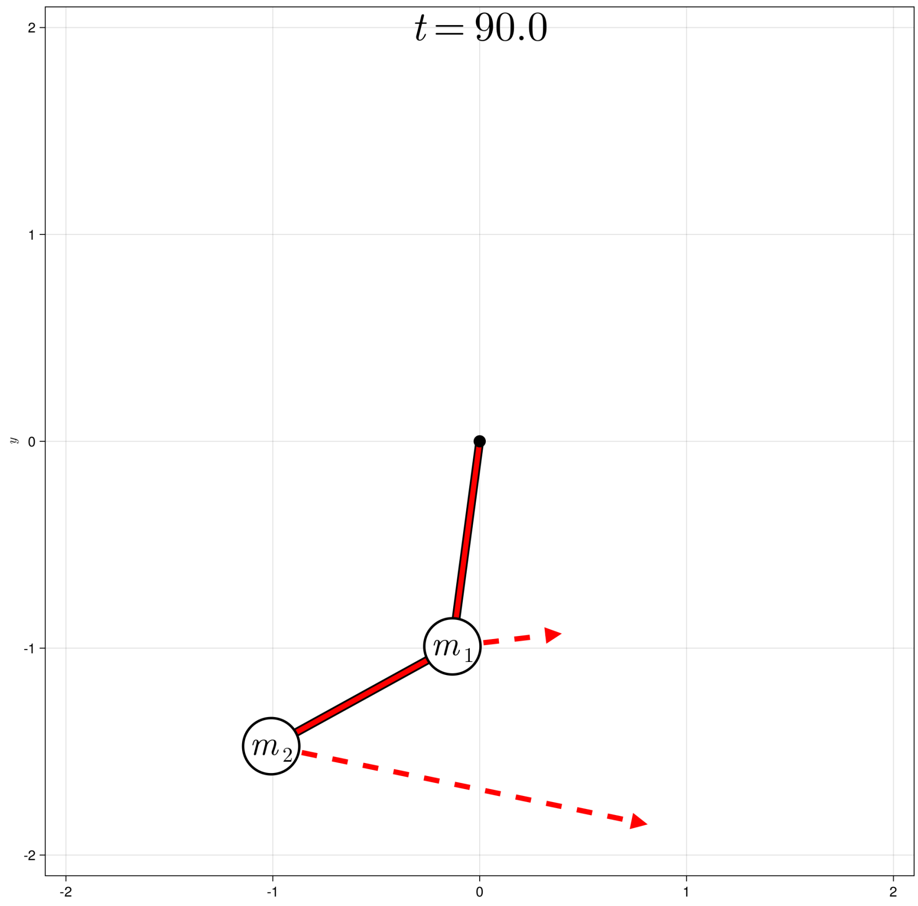

Figure 5: Visual representation of the double pendulum system at $t = 90$s with $m_1 = m_2 = 1$ and $\ell_1 = \ell_2 = 1$.

system of first-order differential equations [12]:

$$\dot{\theta}_1 = \omega_1 \tag{5}$$

$$\dot{\theta}_2 = \omega_2 \tag{6}$$

$$\dot{\omega}_1 = \frac{-g(2m_1 + m_2)\sin\theta_1 - m_2 g \sin(\theta_1 - 2\theta_2) - 2\sin(\theta_1 - \theta_2) m_2(\omega_2^2 \ell_2 + \omega_1^2 \ell_1 \cos(\theta_1 - \theta_2))}{\ell_1(2m_1 + m_2 - m_2\cos(2(\theta_1 - \theta_2)))} \tag{7}$$

$$\dot{\omega}_2 = \frac{2\sin(\theta_1 - \theta_2)(\omega_1^2 \ell_1(m_1 + m_2) + g(m_1 + m_2)\cos\theta_1 + \omega_2^2 \ell_2 m_2 \cos(\theta_1 - \theta_2))}{\ell_2(2m_1 + m_2 - m_2\cos(2(\theta_1 - \theta_2)))}, \tag{8}$$

where $g = 9.81$ is the acceleration due to gravity. The double pendulum exhibits chaotic behavior for sufficiently large initial angles, regardless of pendulum parameters [15].

We map the angles θ_1 and θ_2 of the double pendulum onto the *Tonnetz* by the following:

$$\theta_1 \mapsto \theta_1 \mod 16\pi \tag{9}$$

$$\theta_2 \mapsto \theta_2 \mod 12\pi. \tag{10}$$

This mapping is chosen so that the periods of θ_1 and θ_2 are the same as the period of the major third and minor third axes of the *Tonnetz*. Functionally, this means that the double pendulum trajectory will make one full rotation around the x-axis of the *Tonnetz* after eight full rotations of the inner pendulum and one full rotation around the y-axis of the *Tonnetz* after six full rotations of the outer pendulum. Our symbolic dynamics considers only the sequence in which each triangle on the mesh is visited, discarding any notion of time.

2.4 Software

The double pendulum, *Tonnetz*, and RK4 solver were all implemented with entirely custom Julia code. All plots of the pendulum, trajectories, and the *Tonnetz* were generated using the `Makie.jl` package [6]. Synthesis and analysis of the chord progressions was done using the `music21` library for Python [4], which has built-in support for analysis of progressions of pc-sets. Implementation of tonal interval space metrics and related plots for evaluating the quality of the chord progressions additionally use the `TIV.lib` [13] and `matplotlib` [9] Python packages.

3 Results

3.1 Double Pendulum Trajectory and Chord Progressions

In the following, we consider the double pendulum system with $m_1 = m_2 = 1$ and $\ell_1 = \ell_2 = 1$. Additional simulations with different parameters were performed, but the results were qualitatively similar to those presented here. We use the initial conditions $\theta_1(0) = \pi, \theta_2(0) = \pi - 0.3$ and $\omega_1(0) = \omega_2(0) = 0$, which are close to the unstable equilibrium point of the double pendulum system, ensuring the trajectory exhibits chaotic behavior. The state-space trajectory modulo 2π of the double pendulum is shown in Figure 6.

When mapped onto the *Tonnetz*, the trajectory of the double pendulum produces a sequence of chords, as shown in Figure 7. For visualization purposes, a temporal section of the trajectory with period 0.125s is shown in Figure 8. We include an excerpt of the total generated chord progression

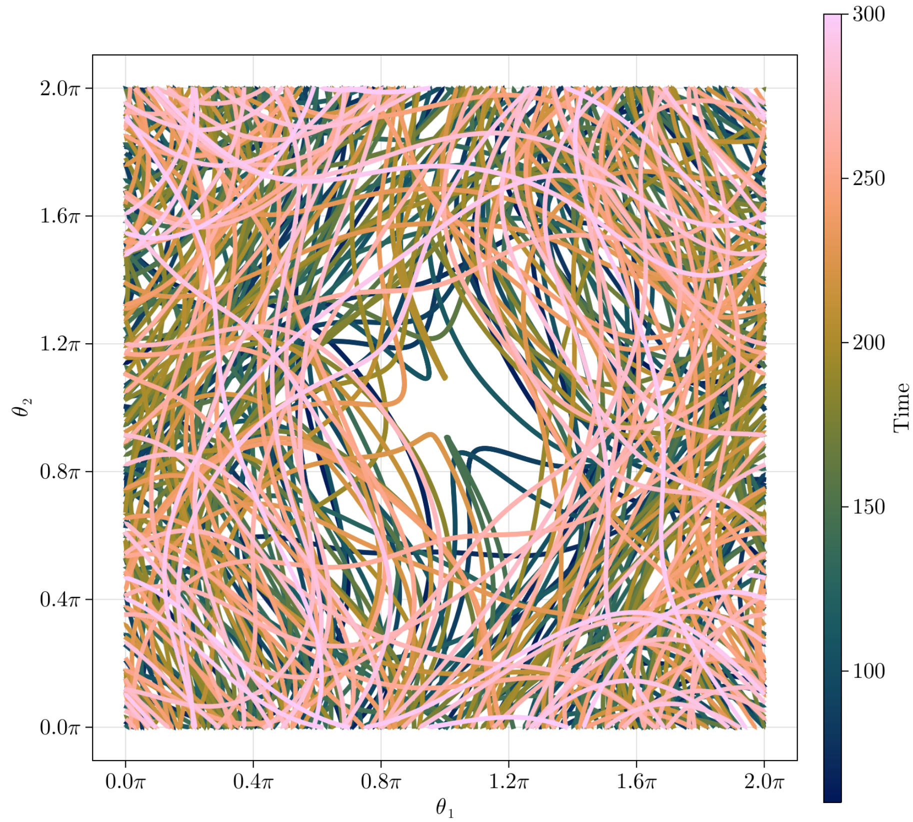

Figure 6: Angular positions of the double pendulum modulo 2π from time t = 60s to t = 300s. The color of the trajectory represents the time, with cooler colors representing earlier points in the trajectory and warmer colors representing later ones. The trajectory is chaotic with a strange attractor near $\theta_1 = \theta_2 = 0$.

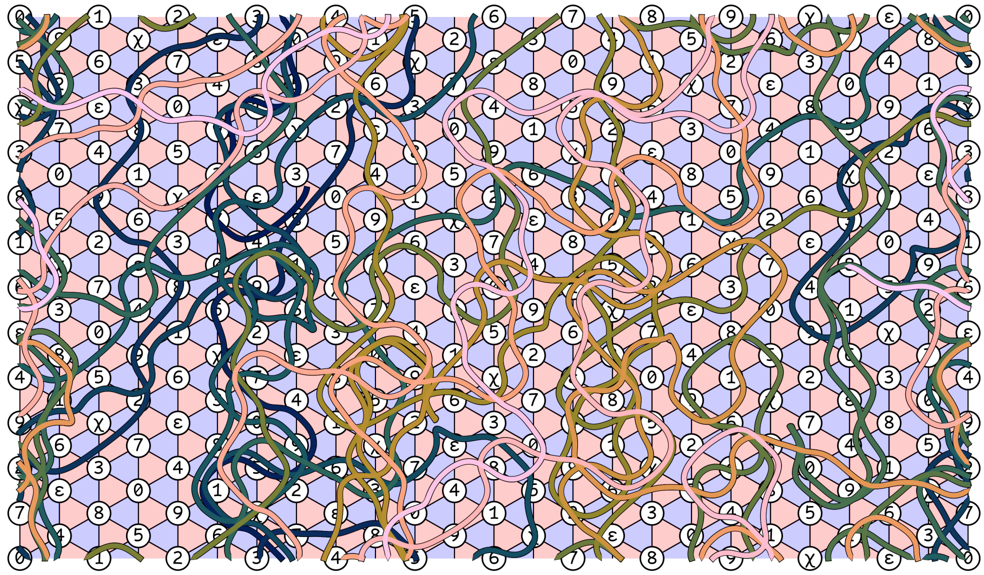

Figure 7: The trajectory of the double pendulum mapped onto the *Tonnetz*. The color of the trajectory represents the time, with cooler colors representing earlier points in the trajectory and warmer colors representing later ones. The x-axis is $(\theta_1 \mod 16\pi)$ and the y-axis is $(\theta_2 \mod 12\pi)$. This visual provides some validation for our choice of mapping, as parts of the trajectory cross through large portions of the *Tonnetz* before reversing direction.

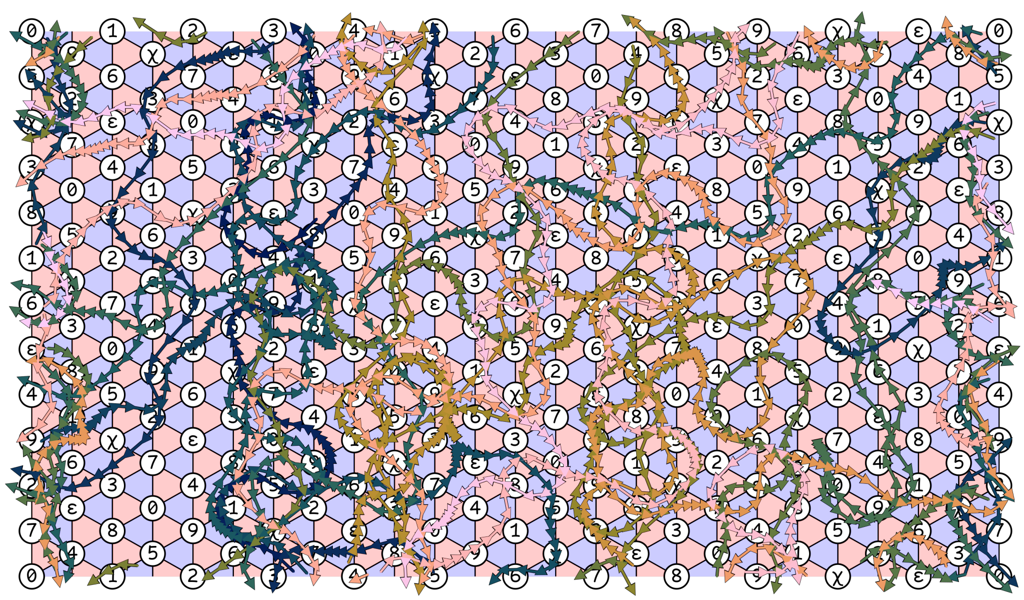

Figure 8: Temporal section of the trajectory in Figure 7 with period 0.125s. The color of the trajectory represents the time. The relative angular velocities of the double pendulum are far more apparent than in Figure 7, with regions of high velocity marked by longer arrows.

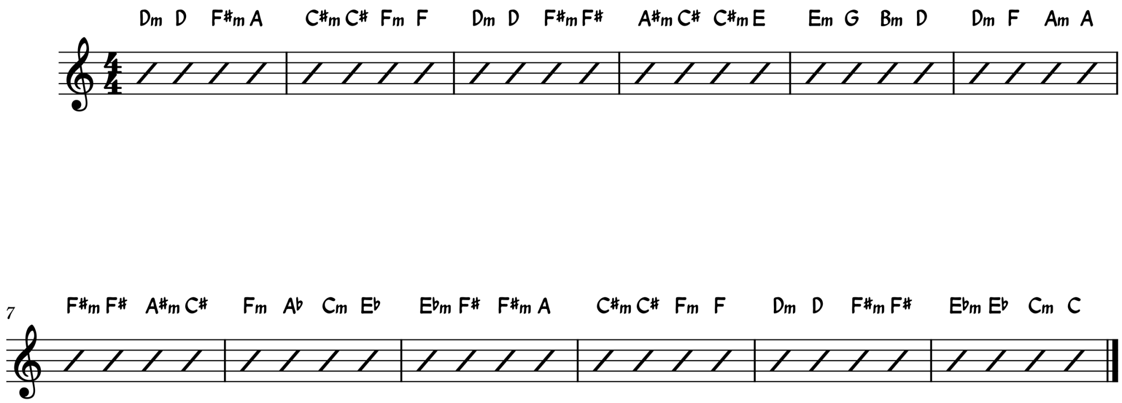

Figure 9: Excerpt of the chord progression generated by the double pendulum trajectory mapped onto the *Tonnetz*. The pc-sets are converted to chords using the `music21` library and typeset using MuseScore 4[1]

in Figure 9, as the full progression is too long to include here.

3.2 Chord Progression Analysis

Since we aim to produce chord progressions which are tonally coherent, it is important to choose a metric to evaluate the quality of the progressions. In [10], the authors propose a metric for evaluating what they call *tonal fitness*, consisting of tonal pitch distance, surface dissonance, voice leading, and hierarchical structure. While all of these are important for a larger composition, we will focus on the tonal pitch distance, divided into the perceptual distance between subsequent chords and how well a chord fits within the key of the progression, and voice leading distance, which measures the melodic attraction of two consecutive chords, as these metrics are relevant for short chord progressions [10]. Since adjacent triangles on the *Tonnetz* share one or two notes, we expect that the chord progressions generated will have high voice leading. Each component of the tonal fitness metric is calculated in the TIS.

We consider two components of the tonal pitch distance. First, the perceptual distance $\hat{d}(T_i, T_{i-1})$ between subsequent chords quantifies their musical similarity, with smaller distances indicating more related chords. The perceptual distance is given by the normalized Euclidean distance between their TIVs. Second, the key unrelatedness quantifies how well a chord is aligned with the key of the composition, with lower values indicating a better fit with the key. Denote the TIV of the diatonic pitches of the key as T_{key}. We consider only the major and minor modes, as they have the same consonance [2]. Then, the key unrelatedness is given by

$$k(T_i, T_{key}) = \hat{\theta}\left(T_i, T_{key}\right), \tag{11}$$

where $\hat{\theta}$ is the angular distance normalized to $[0, 1]$ for all pairs of keys and triads. Figure 10 shows the key unrelatedness of all triads from the key of C, colored according to the number of pitches in the triad which are part of the key.

[1]MuseScore 4 is a free, open-source (GPL-3 licensed) music composition program. More information is available at `musescore.org`.

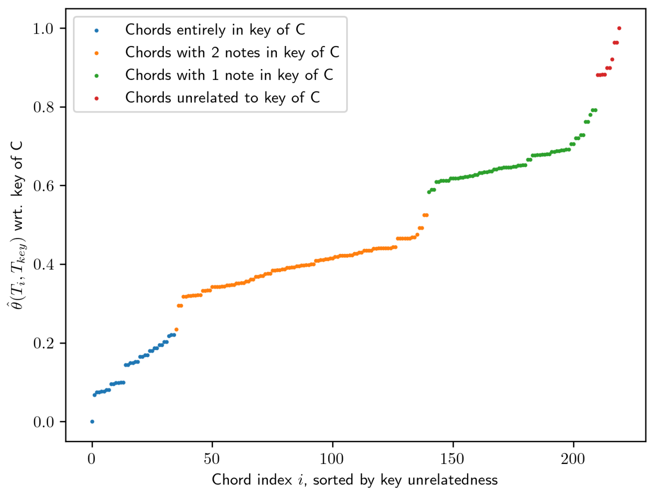

Figure 10: Key unrelatedness, or normalized angular distance, of all triads from the key of C, colored according to the number of pitches in the triad which are part of the key. The chords are also ordered according to their key unrelatedness values. This metric has high categorical accuracy, with each set of triads occupying a distinct range of metric values. Additionally, the boundaries between each class show a noticeable jump in metric value, further aiding its classification potential.

We also consider the voice leading distance, which measures the melodic attraction of two consecutive chords. The voice leading distance d_v between two TIVs T_i and T_{i-1} is given by

$$d_v\left(T_i, T_{i-1}\right) = \frac{\hat{d}(T_i, T_{i-1})}{3} \sum_{\ell=1}^{3} v\left(n_{\ell_{i-1}}, n_{\ell_i}\right), \tag{12}$$

where n_{ℓ_i} is the note of the ℓth voice of the ith chord and v is the per-voice voice leading distance. The per-voice distance is defined as

$$v\left(n_{\ell_{i-1}}, n_{\ell_i}\right) = 0.5\left(1 - \cos(\theta(T_{n_{\ell_i}}, T_{key}))\right) e^{0.05(s-5)}, \tag{13}$$

where s is the number of semitone steps between n_{ℓ_i} and $n_{\ell_{i-1}}$. The exponential term penalizes large jumps between notes in a single voice and is scaled such that the maximal average distance of five semitones results in a scaling value of one. Since the *Tonnetz* naturally results in parsimonious voice leading, we compute the voices by taking the three notes of the chord which are closest to the notes of the previous chord. The $\hat{\theta}$ term is the normalized angular distance between the TIV of the note and the key of the chord progression, which penalizes voices for which the new note is outside of the key. This metric is inspired by the voice leading distance metric used in [10], which differs in that they use the Euclidean distance of the note from the key, rather than the angle, and they divide d_v by the Euclidean distance between the two chords. We found that using the angle results in greater stratification of the metric values, which is useful for classification. Moreover, dividing by the Euclidean distance results in musically incoherent voice leading distances[2]. Instead, we

[2]Specifically, dividing by the Euclidean distance causes parsimonious progressions, or progressions where two of the

multiply by the perceptual distance between the two notes; this leads to parsimonious progressions having minimal voice leading distances.

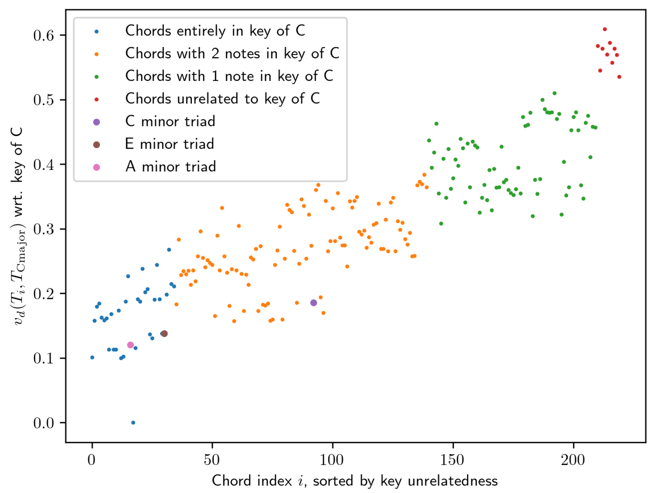

Figure 11: Voice leading distance from the C major triad (pc-set $\langle 0,4,7\rangle$) to all triads with respect to the key of C. The values are colored according to the number of pitches in the triad which are part of the key. For comparison, the chords are ordered the same as in Figure 10. There is a clear region containing all three parsimonious progressions, which are labeled individually on the plot. Unlike in Figure 10, the voice leading distance has more overlap across the key boundary.

We divide the generated progression into sub-progressions of length $n = 3,4,6$, common lengths of chord progressions in contemporary music. The key of each sub-progression is estimated using the Krumhansl algorithm as implemented by `music21` [4]. Other methods for key estimation directly within the TIS are considered in [1], but we found that the `music21` method was sufficient for our purposes. Figure 12 shows the mean and standard deviations of the metric values within the sub-progressions for each of the chord progression lengths. As expected, the progressions show exceptional and consistent voice leading, with the voice leading distance being near 0.1 for all progressions. The perceptual distance is also consistently low across progressions. The key unrelatedness shows far more inconsistency across progression length, which is to be expected, as movement across the *Tonnetz* can result in rapid key changes. Despite this, the three-chord progressions show impressive consistency in key unrelatedness.

4 Discussion and Future Work

ChordDyn is a method for generating chord progressions using a chaotic double pendulum trajectory mapped onto a musically-inspired symbolic dynamics. The resulting chord progressions generated by this method show high voice leading and are tonally coherent. Future work could include a more thorough analysis of the chord progressions generated by this method, including

pitches are shared between chords, to be given substantially larger voice leading distances. Incidentally, parsimonious progressions are the primary kind of progression we expect from continuous functions on the *Tonnetz*

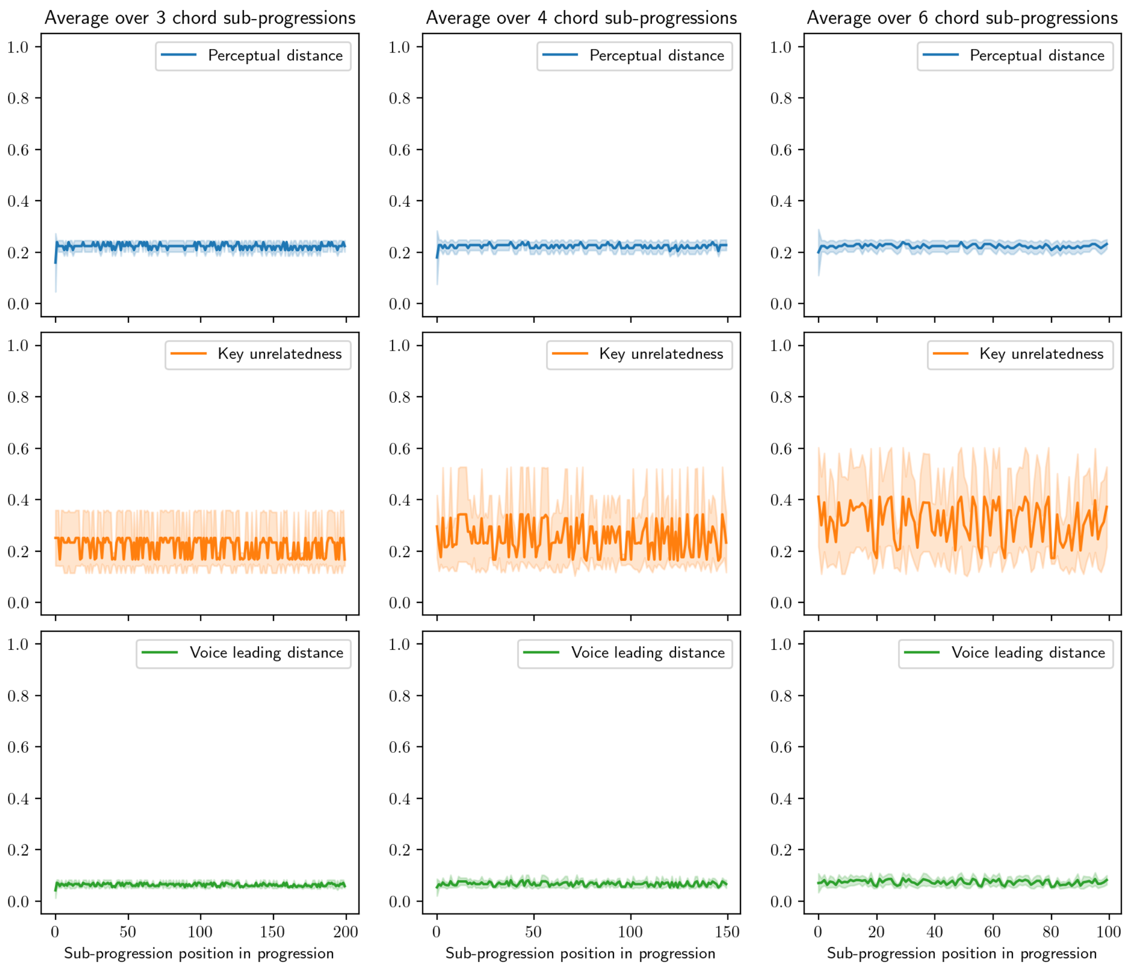

Figure 12: Metric values of chord progressions generated using the double pendulum, averaged over all chord progressions of length $n = 3, 4, 6$.

integration of hierarchical tension and harmonic function, and a comparison of the chord progressions generated by the double pendulum to those generated by other chaotic systems. The method presented here could be extended generate progressions including diminished and augmented triads, seventh chords, and other chord types using existing extensions to the *Tonnetz*, such as adding additional axes. Such changes would require a different chaotic attractor, although one trivial option would be to use a triple (or quadruple, etc.) pendulum. Additionally, it may be possible to define different chord spaces for tuning systems beyond Western 12-tone equal temperament, such as Carnatic Indian classical music, which views pitches as regions rather than points [14]; the resulting chord spaces may then be more suited to continuous chaotic mappings rather than symbolic ones. Finally, the method presented here could be extended to generate other musical structures, such as melodies or rhythms, by mapping the trajectory onto other musically-inspired symbolic dynamics.

References

[1] Gilberto Bernardes et al. “A multi-level tonal interval space for modelling pitch relatedness and musical consonance”. In: *Journal of New Music Research* 45 (May 2016), pp. 1–14. DOI: 10.1080/09298215.2016.1182192.

[2] Gilberto Bernardes et al. “Conchord: An Application for Generating Musical Harmony by Navigating in the Tonal Interval Space”. en. In: *Music, Mind, and Embodiment*. Ed. by Richard Kronland-Martinet, Mitsuko Aramaki, and Sølvi Ystad. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 243–260. ISBN: 978-3-319-46282-0. DOI: 10.1007/978-3-319-46282-0_15.

[3] Gino Brunner et al. “JamBot: Music Theory Aware Chord Based Generation of Polyphonic Music with LSTMs”. In: *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)* (Nov. 2017), pp. 519–526. DOI: 10.1109/ICTAI.2017.00085.

[4] Mike Cuthbert and Christopher Ariza. “Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data”. In: *International Society for Music Information Retrieval Conference*. 2010.

[5] Diana S. Dabby. “Musical variations from a chaotic mapping”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 6.2 (June 1996), pp. 95–107. ISSN: 1054-1500. DOI: 10.1063/1.166171.

[6] Simon Danisch and Julius Krumbiegel. “Makie.jl: Flexible high-performance data visualization for Julia”. In: *Journal of Open Source Software* 6.65 (2021), p. 3349. DOI: 10.21105/joss.03349. URL: https://doi.org/10.21105/joss.03349.

[7] S. R. Holtzman. “Using Generative Grammars for Music Composition”. In: *Computer Music Journal* 5.1 (1981), pp. 51–64. ISSN: 0148-9267. DOI: 10.2307/3679694.

[8] Andrew Horner and David Goldberg. “Genetic Algorithms and Computer-Assisted Music Composition”. In: *Urbana* 51 (Jan. 1991), pp. 437–441.

[9] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[10] María Navarro-Cáceres, Marcelo Caetano, and Gilberto Bernardes. "Objective Evaluation of Tonal Fitness for Chord Progressions Using the Tonal Interval Space". en. In: *Artificial Intelligence in Music, Sound, Art and Design*. Ed. by Juan Romero et al. Vol. 12103. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 150–164. ISBN: 978-3-030-43858-6. DOI: 10.1007/978-3-030-43859-3_11. URL: http://link.springer.com/10.1007/978-3-030-43859-3_11.

[11] María Navarro-Cáceres et al. "ChordAIS: An assistive system for the generation of chord progressions with an artificial immune system". en. In: *Swarm and Evolutionary Computation* 50 (Nov. 2019), p. 100543. ISSN: 2210-6502. DOI: 10.1016/j.swevo.2019.05.012.

[12] Erik Neumann. *Double Pendulum*. 2021. URL: https://www.myphysicslab.com/pendulum/double-pendulum-en.html.

[13] António Ramires et al. *TIV.lib: an open-source library for the tonal description of musical audio*. Aug. 2020.

[14] Joan Serra et al. "Assessing the Tuning of Sung Indian Classical Music". en. In: *Oral Session* (2011).

[15] Troy Shinbrot et al. "Chaos in a double pendulum". In: *American Journal of Physics* 60.6 (1992), pp. 491–499.

[16] Dmitri Tymoczko. "The Generalized Tonnetz". In: *Journal of Music Theory* 56.1 (2012), pp. 1–52. ISSN: 0022-2909.