Ziffers - Numbered Notation for Algorithmic Composition

Miika Alonen
Department of Computer Science. Aalto University, Finland
miika.alonen@aalto.fi

Raphaël Maurice Forment
ECLLA. Université Jean Monnet, France
raphael.forment@gmail.com

ABSTRACT

Ziffers is an algorithmic number based musical notation system offering a terse syntax for the improvisation and composition of complex melodies and rhythms. Ziffers gradually evolved from empirical experiments aiming at closing the gap between several number-based notational practices spanning from numerical and staff notation to pitch class set and post-tonal theories. Ziffers places a special emphasis on providing support for live algorithmic and generative composition, embedding a toolset of algorithmic processes directly in its syntax model. The system also tries to accommodate these fixed media types of notation to the liveness experienced when using live coding environments. Ziffers has been influenced by TidalCycles (McLean and Wiggins 2010) and Ixi Lang (Magnusson 2011), two popular live coding libraries designed around the use of a custom domain specific language, called mininotation, allowing the live manipulation and transformation of musical patterns. In this article, we propose an environment agnostic presentation of the Ziffers musical notation system. As a proof of concept, this article also introduces two implementations running in different environments: Sonic Pi (Aaron 2016), a live coding platform designed for education and music performance and Musescore, a general purpose scorewriter and score playback software. In doing so, we are hoping to highlight the versatility of our approach, which seems to allow the use of a unified syntax for different contexts of execution and interpretation.

1 INTRODUCTION: ZIFFERS, QUICKLY EDITABLE NUMBER BASED NOTATION

Ziffers — named and inspired by the older *Ziffersystem* (Warkentin 2022) — proposes a system focused on the conciseness and expressiveness of musical notation. Ziffers is designed to enable generation and transformation of musical patterns (McCormack et al. 1996; McLean 2020) in minimal amounts of time and typing. To do so, Ziffers uses a text-based syntax for representing various short-hand symbolic notations used in music theory and concepts in programming languages. The need for concise and succinct notation in the context of live coding performance is a well-known constraint (Charlie Roberts and Wakefield 2018) that gave rise to many practices and techniques aiming to reduce frictions in the conversational feedback loop between the musician and its programming interface (REFERENCE). Domain specific languages (DSLs) and the use of terse mini notations have become an important design pattern in the conception of live coding interfaces (Hoogland 2019; ADD OTHER), and can also be noted to be a key concept in the larger realm of exploratory or conversational open-ended programming where reactivity and fast decision-making is of prime importance to achieve a creative state of flow (Nash 2015; Kery and Myers 2017).

Ziffers is designed as a terse and platform independent syntax capable of embedding algorithmic and generative processes at the notational level. Basic notation tokens denoting pitch, rhythm or expression marks form the base notation, enriched by a set of generative operators and tokens denoting algorithmic transformations of that first layer of notation. Implementation of the Ziffers system is specific to each targeted platform, in conformity with the notation described in this article and the documentation for the prototype implementation. The first prototype of Ziffers has been created in 2018 (Alonen 2018) as an inline parser for Sonic Pi. It was thought as an attempt to speed up the process of writing melodic lines on-the-fly, taking advantage of the large number of predefined methods and compositional helpers offered by Sonic Pi's rich internal library. Its design aimed at resolving the dissociation between pitch and rhythm imposed by Sonic Pi data structure and imperative-oriented programming model. After a few years of evolutionary prototyping, Ziffers 2.0 was released in 2022 (Alonen 2022), including a new parser aiming at extending Ziffers capabilities with new operators, nested structures and better support for stochastic and aleatoric composition.

2 METHOD: DESIGNING BY DRAWING FROM MUSIC NOTATION HISTORY

The Ziffers music notation project initially started as an experiment to confront historical numeric pitch-based music notation practices with the kairotic notations (Cocker 2013) typically found in live coding and algorithmic music performance systems (Blackwell et al. 2022). Therefore, the system has been designed iteratively in conversation with the diverse and complex historical legacy of said practices. Documents pertaining to the systems of numeric pitch notation in common practice can be traced back as early as the European baroque up to the contemporan era through examples as varied as the American Nashville system, the Chinese Jianpu simplified notation (Kaminski 2022; MNMA 2022) or the theoretical pitch-class set numbering of musical structures and chords used in post-tonal music theory (Forte 1973).

Numerical systems for score, harmonic or melodic writing – less commonly encountered than their staff-based counterparts – have historically served as compositional tools for polyphonic contrapuntal notation (Davantès 1560), early theoretical endeavours in algorithmic composition (Kircher 1650) or in pedagogy and teaching (Rousseau 1781). Finding a new breath during the late nineteenth-century through the Galin-Paris-Chevré notation (Dauphin 2012), the practice spread to Asia where it is widely used, known and practiced, sometimes concurrently with standard staff notation. Some specialized applications of numbered notation can also be found in the toolset of contemporary ethnomusicologists and ethnomathematicians that often take advantage of the versatility of symbolic cipher notation to devise notation systems capable of formalizing musical systems or previously non-written musical practices (Chemillier 2002).

We perceive – as a thought experiment – the use of mininotations (Magnusson and McLean 2018) as modern live coding analogues to previously mentioned numerical music notation systems. In that regard, with their conciseness and economy of expression, we observe that older non-computer based numerical notations systems are adequate fits for the domain but have never been fully transposed and applied to the realm of live coding performance. Strikingly, we see them being used by musicians and music theorists alike as pragmatic and practical tools for draft notation and information sharing. In our opinion, the versatility of pitch-based number notations as well as their adequacy to the traditional way of inputting complex data structures (as lists, arrays or ordered data) in the most music programming languages (Roads 1996; Dannenberg 2018) make this type of notation a fruitful exploration domain for live coding and computer-based music notation alike. Generative numeric notation can also mitigate the separation between static and dynamic music notation languages as defined by Dannenberg's typology of the domain. Examples such as Adagio (Dannenberg 1986), ABC (Walshaw 2011), Guido (Hoos et al. 1998), MML (Izumi 2001), MusicXML (Good 2001), Lilypond (Nienhuys and Nieuwenhuizen 2003) or MusicTXT (Li and Li 2021), as static languages, focus on the encoding of musical information on fixed medium while other dynamic languages such as Nyquist (Dannenberg 1997), SuperCollider (McCartney 2002), Sonic Pi (Aaron 2016), TidalCycles (McLean and Wiggins 2010), Open Music (Bresson et al. 2017), OpusModus (Opusmodus 2022), often extends musical notation by blending it with computational models of control flow, sound processing capabilities or binding different means of sonic writing: "The instrument incorporates notational elements, but conversely the notational is becoming increasingly instrumental and systematic" (Magnusson 2019).

3 OBJECTIVES: SPECIFICATION OF A CROSS-PLATFORM NOTATION SYSTEM

Objectives of the Ziffers numerical notation adheres to the cognitive dimensions of music notations as defined by Nash (2015) and to the criterion defined by the Music Notation Project (MNMA 2022). The notation must allow users of different ages and backgrounds to compose music. The Ziffers extension built for Sonic Pi presented in latter sections of this paper builds on the accessibility and pedagogical efforts pursued by Aaron (2016). The development of the numerical notation aims to remove the exclusivity that comes with the music theory and provide a beneficial live coding experience for the uninitiated. Use of numeric notation should make it possible to write melodic sequences and interval movements using numbers regardless of the key and scale. The notation should also be minimal enough to be easily manipulated through any writing tool by relying solely on symbols taken from the ASCII table. To easily conceptualize pitches in time, defining a combination of pitch and duration should be as straightforward as possible. The notation should be able to function as an educational tool for teaching music theory and composition. Furthermore, by making the user aware of the relationship between musical and mathematical notation, the system should also support teaching advanced music theory concepts such as musical set theory and the manipulation of pitch-sets or any complex musical objects.

The number of live coding environments have steadily increased in recent years (Toplap 2022). We argue that there is now a need for some amount of notational interoperability between different live coding environments. We interpret the non coordinated emergence of Tidal-like mininotations in tools such as Gibber (Charles Roberts and Pachon-Puentes 2019), Bacalao (Fraser 2020), Petal (kn1kn1 2017) and Hydra (Jack 2019) as an attempt from live-coders to agree on a common rhythmic pattern language despite the use of different and often highly personalized environments. In a similar fashion, the Ziffers project is hoping to bring some cross-environment support for concise and rich melodic notation. Moreover, this project can also be considered as an attempt to bridge the gap between traditional staff-based notation tools such as Avid's Sibelius, Steinberg's Dorico or Musescore and kairotic pattern notations used by live-coders. As demonstrated by the Musescore interpreter (REF TO FIGURE), Ziffers can be used both as a live and fixed-media notation, allowing any musician learning the system to quickly jot down or hear their ideas directly in any Ziffers supported software system capable of audio playback.

The Ziffers system is open for the multiple interpretations of any integer sequence. Depending on context and purpose, numbers can receive multiple interpretations, denoting musical objects such as integer notation (0-9 and T=10 and E=11), scale degrees (1-9, 0=rest), midi notation (1-127), MIDI channels (0-15) or even roman numerals for chord sequence writing (i-vii). The notation also supports commenting and providing supplementary information to clarify the meaning of any number sequence, thus adding support for the definition of custom scales, temperaments and events of any kind. The base syntax has also been designed to support common score-centric notation symbols (codas, bar lines, repetitions) and extended generative symbols (random numbers, mathematical operations) more typically found in a live coding context. In doing so, the Ziffers system allows for some amount of transduction between different notation mediums and contexts, allowing the displacement of live coding ideas into the realm of score-based music composition. From another perspective, Ziffers also opens the field for quickly bringing musical ideas from staff-based musical notation in to the typically improvised and delayed feedback (Rohrhuber, Dean, and McLean 2018) oriented performing context permitted by live coding environments.

4 SONIC-PI BASED PROTOTYPE IMPLEMENTATION

(INSERT PICTURE HERE)

Ziffers has evolved from a simple script embedded in one of many Sonic Pi's text buffers to a fully fledged Ruby extension and framework for computer aided algorithmic composition based on a PEG parser. The current software architecture for the Sonic Pi's based Ziffers implementation is presented in the figure above. A Musescore 3.0 plugin allowing the transformation of ziffers notation to MIDI files and staff-based score format has also been created and proposed as a proof of concept tool for analyzing musical structures and learning the new numeric notation.

(INSERT MUSESCORE PICTURE HERE)

```
ziffers " // Aphex Twin - Avril 14th in Ziffers notation with some example transformations
/ synth: piano
| q ^ 2 _ 5 ^ 0 e r s 0  | sq. 5 e 5 ^ 4 3 2 0  | / retrograde: true
| q 2 _ 5 ^ 0 e r s 0 _  | sq. _ 5 e 5 ^ 4 3 2 0  | / octave: -2
| e __ 0 5 ^ 0 2 _ 2 ^ 0 2 4  | e __ 3 ^ 0 3 4 _ 1 ^ 0 3 2  | / inverse: 2"
```

# Conclusion

5 Acknowledgment

Authors would like to thank Sam Aaron and the whole Sonic Pi community for the important work on improving the accessibility on programming and for the inspiration to pursue research in the field of live coding. Raphaël would like to thank Laurent Pottier and Alain Bonardi for their support, as well as the doctoral school *3LA* from the University of Lyon for the funding it provided to this research.

References

10 Aaron, Sam. 2016. "Sonic Pi–Performance in Education, Technology and Art." *International Journal of Performance Arts and Digital Media* 12 (2): 171–78.

Alonen, Miika. 2018. *Ziffers: Numbered Notation for Algorithmic Composition* (version 0.1). https://github.com/amiika/ziffers/commit/c12516d2b1604836925cbe829e488e033578405c.

———. 2022. *Ziffers: Numbered notation for algorithmic composition* (version 2.0). https://github.com/amiika/ziffers.

Blackwell, Alan F, Emma Cocker, Geoff Cox, Alex McLean, and Thor Magnusson. 2022. *Live Coding: A User's Manual.* MIT Press.

Bresson, Jean, Dimitri Bouche, Thibaut Carpentier, Diemo Schwarz, and Jérémie Garcia. 2017. "Next-Generation Computer-Aided Composition Environment: A New Implementation of OpenMusic." In *International Computer Music Conference (ICMC'17)*.

Chemillier, Marc. 2002. "Ethnomusicology, Ethnomathematics. The Logic Underlying Orally Transmitted Artistic Practices." In *Mathematics and Music*, 161–83. Springer.

Cocker, Emma. 2013. "Live Notation:–Reflections on a Kairotic Practice." *Performance Research* 18 (5): 69–76.

Dannenberg, Roger B. 1986. "The Cmu Midi Toolkit." In *ICMC*, 86:53–56.

———. 1997. "Machine Tongues XIX: Nyquist, a Language for Composition and Sound Synthesis." *Computer Music Journal* 21 (3): 50–60.

———. 2018. "Languages for Computer Music." *Frontiers in Digital Humanities* 5: 26.

Dauphin, Claude. 2012. "Le Devenir Du Système de Notation Musicale de Jean-Jacques Rousseau." *Rousseau En Musique, Orages*, no. 11: 79–98.

Davantès, Pierre. 1560. "Pseaumes de David, Nouvelle Et Facile Methode Pour Chanter Chacun Couplet Des Pseaumes Sans Recour Au Premier." 1560. https://gallica.bnf.fr/ark:/12148/bpt6k3135388/f23.item.

Forte, Allen. 1973. *The Structure of Atonal Music*. Vol. 304. Yale University Press.

Fraser, Glen. 2020. *Balacao - Somewhat Fishy Live Cod(e) Extensions to SuperCollider*. (version 0.1). https://github.com/totalgee/bacalao.

Good, Michael. 2001. "MusicXML for Notation and Analysis." *The Virtual Score: Representation, Retrieval, Restoration* 12 (113-124): 160.

Hoos, Holger H, Keith A Hamel, Kai Renz, and Jürgen Kilian. 1998. "The GUIDO Notation Format–a Novel Approach for Adequately Representing Score-Level Music."

Izumi. 2001. "Music Macro Language." 2001. http://www.vgmpf.com/Wiki/index.php/Music_Macro_Language.

Jack, Olivia. 2019. "Hydra: Live Coding Networked Visuals." In *Proceedings of the Fourth International Conference on Live Coding*, 353–54.

Kaminski, Joseph S. 2022. "Jianpu Simplified Notation and the Transnational in Musical Repertoires of New York's Chinatown." In *Material Cultures of Music Notation*, 110–23. Routledge.

Kery, Mary Beth, and Brad A Myers. 2017. "Exploring Exploratory Programming." In *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 25–29. IEEE.

Kircher, Athanasius. 1650. "Musurgia Universalis Sive Ars Magna Consoni Et Dissoni." 1650. https://archive.org/details/bub_gb_97xCAAAAcAAJ/mode/2up.

kn1kn1. 2017. *Petal - a Small Language on Sonic Pi with Similar Syntax to TidalCycles*. https://github.com/siaflab/petal.

Li, Kelian, and Wanwan Li. 2021. "MusicTXT: A Text-Based Interface for Music Notation." In *Proceedings of the 11th Workshop on Ubiquitous Music (UbiMus 2021)*, 62–71. g-ubimus.

Magnusson, Thor. 2011. "The Ixi Lang: A Supercollider Parasite for Live Coding." In *ICMC*.

———. 2019. *Sonic Writing: Technologies of Material, Symbolic, and Signal Inscriptions*. Bloomsbury Academic.

Magnusson, Thor, and Alex McLean. 2018. "Performing with Patterns of Time."

McCartney, James. 2002. "Rethinking the Computer Music Language: Super Collider." *Computer Music Journal* 26 (4): 61–68.

McCormack, Jon et al. 1996. "Grammar Based Music Composition." *Complex Systems* 96: 321–36.

McLean, Alex. 2020. "Algorithmic Pattern." In *Proceedings of the 20th Conference on New Interfaces for Musical Expression. Birmingham, UK*.

McLean, Alex, and Geraint Wiggins. 2010. "Tidal–Pattern Language for the Live Coding of Music." In *Proceedings of the 7th Sound and Music Computing Conference*, 331–34.

MNMA. 2022. "The Music Notation Project." 2022. https://musicnotation.org/tutorials/numerical-notation-systems/.

Nash, Chris. 2015. "The Cognitive Dimensions of Music Notations."

Nienhuys, Han-Wen, and Jan Nieuwenhuizen. 2003. "LilyPond, a System for Automated Music Engraving." In *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, 1:167–71. Citeseer.

Opusmodus. 2022. “Opusmodus - Structural Composition System.” 2022. https://opusmodus.com/.

Roads, Curtis. 1996. *The Computer Music Tutorial*. MIT press.

Roberts, Charles, and Mariana Pachon-Puentes. 2019. “Bringing the Tidalcycles Mini-Notation to the Browser.” In *Proceedings of the Web Audio Conference*.

Roberts, Charlie, and Graham Wakefield. 2018. “Tensions and Techniques in Live Coding Performance.”

Rohrhuber, Julian, RT Dean, and A McLean. 2018. “Algorithmic Music and the Philosophy of Time.”

Rousseau, Jean-Jacques. 1781. *Projet Concernant de Nouveaux Signes Pour La Musique*. éditeur non identifié.

Toplap. 2022. “Awesome Live Coding - a Curated List of Live Coding Languages and Tools.”

Walshaw, Christopher. 2011. “The Abc Music Standard 2.1.” *URL: Http://Abcnotation.com/Wiki/Abc: Standard: V2* 1.

Warkentin, Emily Grace. 2022. “The Story of the Ziffersystem and the Russian Mennonites: Counting Blessings.”