JUST INTONATION IN BACH

Andrea Agostini
University and Conservatoire of Antwerp
Orpheus Institute Ghent
andreaagostini@bachproject.net

Daniele Ghisi
Conservatory of Turin
danieleghisi@bachproject.net

ABSTRACT

This article discusses the experimental implementation of just intonation within the *bach* [1] software tool, a package for Max [2] dedicated to musical representation and computer-aided composition. This implementation augments the already extant pitch data type with a representation of Helmholtz-Ellis Just Intonation pitches, as well as a set of operations upon them, providing users with the virtually unique ability, among software tools of this kind, to manipulate algorithmically, as well as representing graphically, such kind of pitch representation.

We shall firstly discuss the context for this new feature set. We will then provide a brief introduction to the main concepts of modern just intonation systems. Next, an overview of the current status of pitch representation in *bach* will be given, complemented by some general considerations about tuning systems in computer music software. Then we will delve into the details of the actual implementation of just intonation in *bach*, from the perspectives of both the user and the underlying workings, describing the syntax of the pitch representation and the possible mathematical operations on justly tuned pitches, and providing an overview of the *bach* modules dealing with them. Finally, we will present some simple examples of the new possibilities afforded by this new addition to the *bach* package.

1. INTRODUCTION

Pitch representation in computer music software is a complex topic. Simplistic representations, such as frequency values or MIDI pitches, are easily implemented and allow for a basic set of fundamental operations such as transpositions, expansions and inversions in the chromatic or frequency space. The conversion between the two spaces is mathematically simple, and involves logarithmic and exponentials (a corollary of the fact that human perception of intervals relates to frequency ratios, rather than to frequency differences).

On the other hand, such representations neglect a number of details and possibilities that may or may not be relevant according to the aesthetic, theoretical and technical context in which the representation is employed. These include enharmonicity, that is, the distinction between e.g. C♯ and D♭, and the expression of pitches in different tuning systems.

Software tools for computer aided composition approach these problems differently. The venerable PatchWork family, including Patchwork itself and PWGL (both obsolete nowadays) as well as OpenMusic, [3] used to rely on MIDIcents, which are just a higher-resolution variant of MIDI pitches, with limited support for enharmonic information. As mentioned above, this, and more specifically the fact that real-valued MIDIcents are supported, in principle allows one to express any pitch in any tuning system. In this scenario, native semantic support — including, but not limited to, graphical representation — is only provided for divisions of the octave in 12, 24 or 48 equal steps. Recent versions of OpenMusic provide semantic support for other equal divisions of the octave. On the other hand, as we shall see in section 2, this only covers a part of the problem.

OpusModus [4] and Music21 [5] support operations with both pitch classes (thus disregarding enharmonic information) and an enharmonically-aware pitch data type; operations upon these two different representations are however expressed very differently. [6]

MaxScore [7] provides extensive support for all the aforementioned features, with a strong focus on alternative tuning systems, but it is aimed at graphical representation much more than algorithmic processing, albeit in principle (and, to some extent, in practice) the latter is also possible [1].

The latest publicly distributed version of the *bach* package (version 0.8.1) supports both MIDIcents representation and a pitch type, which features advanced support for micro-intervals and enharmonicity. On the other hand, its support for alternative tuning systems is quite limited, and essentially applies to equal-tempered microtonal subdivisions of the octave.

[1] www.bachproject.net
[2] http://cycling74.com

[3] https://openmusic-project.github.io
[4] https://opusmodus.com
[5] www.music21.org
[6] In computer science terms, the pitch data type of OpusModus, as well as that of OpenMusic and Music21, is a class, that is, a construct associating data and operations upon it. This collides with the fact that "pitch class" has a very specific, and completely unrelated, meaning in music theory — the meaning referred to when discussing OpusModus and Music21 a few lines above. For this reason we shall keep referring to "pitch type" or "pitch data type" when it comes to the computer representation, even if it is technically a class; and use the term "pitch class" only for the music theory concept introduced by Allen Forte.
[7] www.computermusicnotation.com

It seemed to us that it would be an interesting feature for *bach* to include full support for just intonation, which is not just a variant of the widespread equal temperaments, but rather is based upon on a different approach altogether. We think that such a feature, which at the time of writing (October 2025) is functional in a pre-release version of *bach*, may be of interest to a growing community of composers and musical artists interested in working with alternative tuning systems.

2. A SHORT INTRODUCTION TO JUST INTONATION

The history of Western tuning systems is a fascinating and complex topic in itself, and we are certainly not attempting to tackle it in this article [2, 3, 4]. Still, from a modern, non-historical perspective, two influential ways of determining which pitches and intervals are afforded to musicians are equal temperaments and just intonation.

Equal temperaments (from now on, ET) take a reference interval and subdivide it uniformly in a number of steps. From a frequential point of view, if R is the size of the reference intervals, every one of the n steps corresponds to a frequency ratio $r = \sqrt[n]{R}$, such that when we pile up n of them — i.e., when we perform n consecutive multiplications of r — we get back R.

The reference interval R is often set to be the octave, corresponding to a $2/1$ frequency ratio. ET tunings based upon the octave are called *octaviant*.

Subdividing the octave into $n = 12$ equal steps provides the widespread 12-EDO tuning, with EDO standing for 'equal divisions of the octave' (also known as 12-TET, for 'twelve-tone equal temperament'). This tuning informs the vast majority of the music that we hear and instruments that we play in the Western world. In this case, $r = \sqrt[12]{2} = 1.059\dots$ corresponds to a tempered semitone. It is important to notice that this is an *irrational* number, i.e., there is no way to obtain it as a proportion of integer numbers. By stacking semitones, i.e., by multiplying r by itself multiple times, we obtain every interval in the system. The traditional Western system adds on top of this a complex naming policy based upon the concepts of degree and alteration, but this is beyond the scope of this introduction.

Subdividing the octave into $n = 24$ equal steps gives the 24-EDO, i.e., the temperament of quarter-tones; with $n = 48$ we get eighth-tones; and so on. Of course, historically, not all values of n have the same relevance, and the profound reasons why the 12-EDO has emerged as a standard have to do with acoustics, psychoacoustics, algebra and instrument making, and are certainly beyond the scope of this article [5, 4, 2].

In any case, in principle, one can choose R and n freely. At least a few non-octaviant ET system are in relatively widespread usage, if only in the somewhat rarified community of tuning systems aficionados [6]. The most widely known one is probably the Bohlen-Pierce equal-tempered scale, with R set to 3 (the so-called 'tritave') and n to 13.

A profoundly different way of thinking about intervals, traditionally traced back to Pythagoras of Samos, is to notice that small-integer frequency ratios provide blueprints for the most consonant intervals. Physiologically and cognitively, this holds true only to a degree — consonance is certainly a many-headed beast that we are not going to tame here. You can refer to [7] for an insightful introduction. In any case, the choice of the octave as the reference interval for an ET system aligns with this observation — the octave $2/1$ being the 'most consonant' interval in this respect, after the unison $(1/1)$. This idea highlights the role of ratios such as $3/2$ (just fifth) or $4/3$ (just fourth), all corresponding to intervals borrowed from the lowest portion of the harmonic series. Systems based upon this kind of approach are often called *just-intonation* (JI from now on) systems.

In a very profound sense, the friction between equal *frequential* subdivisions (JI systems and the harmonic series) and equal *pitch* subdivisions (ET systems) is the foundational clash of worlds of Western music theory, as well as the driving force that has shaped many of the features of Western music. Among other things, one of the most prominent reasons why 12-EDO has emerged as a standard is that it approximates relatively well a subset of these small-ratio intervals, and particularly the fifth and fourth. On the other hand, its approximations of the just major third $5/4$ and the just minor third $6/5$ are suboptimal.

In a way, JI systems are simpler than ET systems since the calculation of frequencies only requires multiplications of rational numbers, whereas ET systems are based upon irrational frequency ratios.

It is worth noting that, if we extend the idea of ET system to a continuum of pitches (or logarithmic frequency), then any frequency and frequency ratio can be represented within it, including of course JI ratios. The case of 12-EDO representation is particularly relevant because we often use cents, that is, hundredths of 12-EDO semitones, as a sort of 'neutral' unit of measurement for pitch. Still, while representing JI pitches and intervals in terms of cents happens to be convenient, the cents value will generally be irrational, so any truncation will necessarily be approximated.

The opposite is *almost* possible: as any irrational number can be approximated to any desired degree of precision to a rational one, [8] then any ET pitch can be approximated to a JI one with an arbitrary degree of precision. Deciding which rational number makes a good approximation involves balancing simplicity and accuracy. For example, while $\pi = 3.14159\dots$ can be approximated quite precisely by $31415/10000$, a classic approximation such like $22/7$ is arguably more expressive. Though less accurate, it is much simpler — or, in musical terms, more 'consonant', in the sense that it involves ratios between lower harmonics. (Such approximations are often derived using continued fractions, which we'll briefly touch on in Section 5.)

Combining steps of an EDO system is straightforward, as every step is identical and one can stack them up as LEGO bricks. JI provides a potentially infinite palette of bricks all with different heights, so building your cathedral requires further attention. This is to say that putting into practice

[8] In formal terms, this is to say that the set of rational numbers is *dense* in the number line.

the concept of just intonation is not without complications, as the combination of different harmonics can produce a potentially infinite number of variants of the same pitch.

For instance, consider the interval of major third C-E. This can be obtained in two ways: by stacking four perfect fifths (as in C-G-D-A-E) and adjusting by two descending octaves ($(3/2)^4/4 = 81/64$), or by associating it by definition with the pure ratio $5/4$ ($80/64$). In fact, virtually any interval can have different, equally reasonable variants — not to mention the "unreasonable", but not necessarily uninteresting, ones.

One way to curb this potentially overwhelming combinatorics is to generate tuning systems by stacking of a single interval, a bit like in the EDO case, but with this interval being a justly tuned one. The most prominent historical examples are Pythagorean tunings (used throughout the Middle Ages, and still informing the tuning of string instruments today), which generate all of their pitches as combination of pure fifths. Their E would then be at an interval of $81/64$ above C. Of course, this comes at the price of massively limiting the palette of just intervals at one's disposal. As a matter of fact, the 'natural' $5/4$ major third is 'more consonant' than the Pythagorean $81/64$ one, in that it correspond to a simpler ratio, but theoretically more complicated as well, because it requires a generator interval ($5/4$) higher up in the harmonic series, with respect to the octave ($2/1$) and the fifth ($3/2$).

When thirds became unavoidable building blocks of consonance, roughly during the Renaissance, just intonation system emerged that would borrow the E from the harmonic series, using this purer interval of $5/4$ above C. The two ratios $81/64$ and $5/4$ are close but not identical: their discrepancy, given by their ratio $81/80$, is known as 'syntonic comma'.

Theories of tuning are full of similar little discrepancies, called commas, each corresponding to the distance (or, more precisely, the ratio) between different variants of the same interval. Their implications with respect to the musical language are massive. One way to face them is to *temper* them, compromising the purity of integer ratios in order to achieve, or at least approach, some target characteristics — say, transposability, or enharmonic equivalence. Many different tempered systems have emerged in history, and the one that has today gained widespread usage, especially from the 19th century, is the twelve-tone equal temperament. It can easily be said that the harmonic language of many Western composers, from Schubert to Schönberg and beyond, requires by necessity, and was actually prompted by, its uniformity. [9] On the other hand, the aforementioned classical Western pitch naming policy, its notation system, and the inner core of the highly normative set of compositional rules are deeply rooted in pre-ET practice and theory, in a complex and often intimidating multi-layered construction.

Another posture is to accept the variants of JI, embracing the diversity of intervals that rational frequency ratios propose. Compositional interest in this approach resurged in the mid-20th century, through the seminal works, writings and teachings of composers such as Harry Partch, Lou Harrison and LaMonte Young, who started reconsidering the affordances of JI theory and tunings with respect to modernist music. This prompted the development of new theoretical frameworks and notational systems aimed at representing nuances that, in Medieval, Baroque and Renaissance music, were just outsourced to the preliminary choice of a specific tuning system.

Modern approaches to JI are not just a matter of local technical choices: the preference for JI over ET tends to be a defining and affirming fact for composers who embrace it, often identifying (or being identified) as JI composers, and not seldom has philosophical and even political undercurrents [8]. Within this context, tuning discrepancies are often seen as fundamental compositional devices and, as such, they require a notational framework to accommodate it. Intervals such as $9/7$ or $11/7$, for instance, are not something one finds in a classical music theory book, but they provide unique flavours that one can learn to appreciate. To borrow a famous Partch's metaphor, all these ratios lie before us like a "large table of aromatic and unusual viands" [9].

This is where modern-day JI notational systems come in: they provide detailed ways, usually through the redefinition of the frequency ratios associated with the white-key pitches and the introduction of special symbols and accidentals, to notate the different varieties of just intervals. In the following sections, we shall briefly describe a few of them.

2.1 Primes and limits

A crucial facet of just intonation is its connection with the Fundamental Theorem of Arithmetic: the numerators and denominators of a frequency ratio can be factorized, and this factorization gives readable insight on the type of harmonics (and intervals) involved. For example, $81/64 = 2^{-6} \times 3^4$, and $5/4 = 2^{-2} \times 5^1$. Larger primes correspond to higher harmonics, and the highest prime factor that shows up in an interval ratio is called its *limit*. For instance, $5/4$ is in the 5-limit, while $81/64$ is in the 3-limit. This means that it can be built by combining intervals up to the third harmonic (octaves and fifths), whereas, in order to build $5/4$, we need intervals up to the fifth harmonic (that is, we need to include pure major thirds) .

From a slightly different point of view, an interval can be defined by a list containing the exponents of its factorisation, so that $81/64$ is $[-6, 4]$ and $5/4$ is $[-2, 0, 1]$. Every element in the list corresponds to one of the prime numbers, in ascending order.

2.2 Notating just intervals

But one thing is to have a consistent mathematical representation for just intervals; another thing entirely is to have a way to notate them that does not collide (too much) with the traditional musical staff. This is clearly a bit of a bat-

[9] It seems to be established, on the other hand, that the tuning system for which Johann Sebastian Bach composed his *Well-Tempered Clavier* is not the modern 12-EDO, but a different one on whose details entire bookshelves of hypotheses have been written. Also, as a bit of trivia fact, 12-EDO is certainly not the longest-standing 'standard' — meantone temperaments enjoyed a much longer period of prominence.

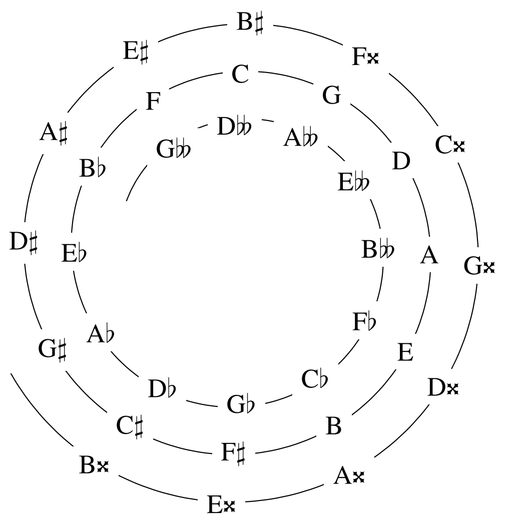


Figure 1. The Pythagorean spiral of fifths (enharmonic pairs, such as D♯ and E♭, refer to different frequencies).

tle against windmills, because of the foundational clash of worlds we mentioned above.

To find a compromise solution, a number of systems have emerged in the last few decades. A relatively popular one is Ben Johnston's notation, which consider 'white-key' pitches (C, D, E, F, G, A, B) as belonging to the pure diatonic scale of the Renaissance, and introduces accidentals to alter them by specific commas [10]. Another one is Sagittal notation, a comprehensive system for notating microtonal scales and tunings using arrow-like symbols [11]. A third system, and one that is arguably gaining traction at least in Europe, is the Extended Helmholtz-Ellis JI Pitch notation (or HEJI for short) [12]. [10] Due to its consistency and relative simplicity, it is the notational system that we have decided to embrace in *bach*.

Differently from Johnston's notation, HEJI considers all 'white-key' pitches (C, D, E, F, G, A, B), as well as any of their alterations with sharps and flats, as Pythagorean, i.e., organised in a spiral of 3/2 fifths (see Figure 1).

In other words, the core elements of the system are, by default, in the 3-limit. For instance, HEJI considers E_5 as $81/64$ above C_5. In order to vary the interpretation of intervals, HEJI introduces specific accidentals, each representing a comma. The crucial connection is that every prime number comes with a specific comma that concerns it.

The comma for the 5-limit is the syntonic comma $81/80$, and its graphical HEJI sign corresponds to adding an upward or downward arrow to the existing Pythagorean accidental (or to a natural, if no accidental was provided). Of course, raising by two or more syntonic commas is also possible, and corresponds to adding more arrows. This means that we can represent the $5/4$ E (with respect to C) in HEJI starting from the Pythagorean E and lower it by one syntonic comma. But nothing prevents us to for example lower it by one more syntonic comma, or to raise it by any number of syntonic commas instead.

[10] Whereas Ben Johnston's notation was indeed invented by Ben Johnston, the Helmholts-Ellis system was inspired by the works of Hermann von Helmholtz and Alexander Ellis, but actually developed in the early 2000s by Marc Sabat and Wolfgang von Schweinitz.

There is also a comma for the 7-limit, called septimal comma, whose accidental is inspired by the digit 7; it amounts to $64/63$, that is, the discrepancy between a Pythagorean minor seventh and a harmonic seventh (the interval between the 4th and 7th partial of a harmonic series). There is a comma for the 11-limit ($33/32$), which, due to its proximity with the tempered quarter-tone, is called undecimal quarter-tone; and so on. Since any positive fraction can be factorised uniquely, any frequency ratio can be expressed uniquely as a combination of a note name with optional sharps or flats (corresponding to the factors 2 and 3) and a set of commas (introducing any higher prime). For instance, the quite exotic fraction $15309/8800$, representing yet another kind of minor seventh, can be decomposed as

$$\frac{15309}{8800} = 2^{-5} \times 3^{7} \times 5^{-2} \times 7^{1} \times 11^{-1}$$

but by introducing the necessary denominators and numerators we can change every one of the primes beyond 2 and 3 into its respective comma. Now, each comma is the discrepancy between a given prime-indexed harmonic, shifted by one or more octaves (and thus multiplied by a power of 2), and the nearest Pythagorean pitch (whose factors are, by definition, powers of 2 and 3). Therefore, changing a prime into a comma requires us to counterbalance the exponents of the Pythagorean part of the expression. After some number juggling, we are left with a decomposition into a Pythagorean part (the 16/9 fraction) and a sequence of commas:

$$\frac{15309}{8800} = \frac{16}{9} \times \left(\frac{81}{80}\right)^{2} \times \left(\frac{64}{63}\right)^{-1} \times \left(\frac{33}{32}\right)^{-1}.$$

HEJI features commas, and accidentals, for limits up to prime 47 (the first few are displayed in Figure 2). The mathematics is made easier by the fact that all these commas only involve a single prime, in addition to powers of 2 and 3.

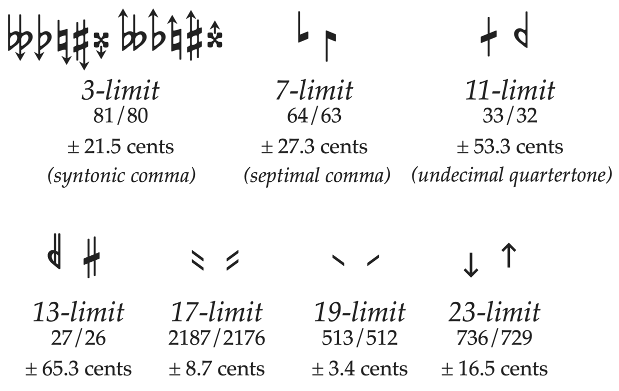


Figure 2. HEJI commas, for the first few primes, along with their accidental representations. The deviations expressed in cents are approximations of irrational values.

As an example, our interval of ratio $15309/8800$ would be notated as in Figure 3, while Figure 4 shows the first few elements of a harmonic series written in HEJI notation.

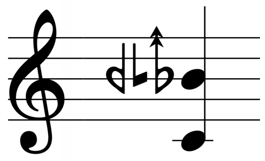


Figure 3. HEJI representation of the interval 15309/8800: starting from a Pythagorean minor seventh, we raise by two syntonic commas, and lower by both one septimal comma and an undecimal quarter-tone.

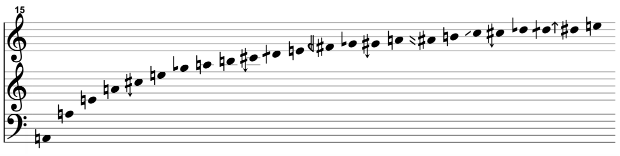


Figure 4. A portion of harmonic series displayed in HEJI notation. Increasingly more unusual accidental signs appear as the series goes higher, as higher prime factors require higher-order commas.

3. PITCHES IN THE *BACH* PACKAGE FOR MAX

bach is a software package aimed at augmenting the Max programming environment with advanced musical representation capabilities [13]. At its forefront are two musical notation editors and sequencers called bach.roll and bach.score, both relying upon a set of custom data types added to Max by the package, namely rational numbers, multi-level lists (*llll*s in the *bach* jargon) and musical pitches (more about these later). The two notation objects are complemented by more than 200 other modules dealing with a wide array of operations, ranging from basic ones (such as arithmetic operations and elementary list processing on the new data types) to advanced ones (such as rhythmic quantisation and the solution of constraint satisfaction problems). *bach* is free and open source software; it was conceived in 2010 as a personal project and has been constantly developed by the authors of this article ever since. As of today, we estimate that it has some thousands of users worldwide.

3.1 Pitch representation and operations

As hinted at before, one notable feature of *bach* is the fact that it treats musical pitches as a full-fledged data type. In the current (at the time of writing) publicly distributed version of the system, a pitch is represented internally through the following components:

- an integer representing the white-key class, from 0 (C) to 6 (B);
- a rational number representing the chromatic alteration with respect to the white key, measured in whole tones or fractions thereof;
- an integer representing the octave, according to the convention by which the middle C is located at the beginning of octave 5. Such relatively unusual convention has been chosen for simplicity's sake with respect to the MIDIcents representation: in this way, the octave of a pitch expressed in MIDIcents can be calculated simply through an integer division by 1200. (For the reasons behind this seemingly idiosyncratic choice, as well as many more details about all the information contained in this section, see [14].)

So, for example $E\flat_6$, that is, the E♭ in the topmost space of the treble clef, will be represented through white-key class 2 (E), alteration $-1/2$ (one chromatic semitone lower) and octave 6. It should be remarked that $D\sharp_6$, that is, an enharmonic pitch corresponding to the same key on a piano keyboard (and hence to the same MIDI note) is represented differently from E♭.

One could rather see pitches as intervals counted from C_0: in this sense, G_0 represents not only a very low G, but also a perfect fifth. This makes arithmetic operations on pitches conceptually meaningful. Sums represent the stacking of intervals (stacking a major third and a minor third produces a perfect fifth, thus $E_0+E\flat_0 = G_0$) as well as transposition (transposing D_4 a fifth above yields A_4, thus $D_4+G_0 = A_4$). Multiplying a pitch by an integer represents stacking an interval on top of itself multiple times (three stacked major thirds yield an augmented seventh, so $E_0\times 3 = B\sharp_0$). Dividing a pitch by an integer means dividing an interval in equal parts — which sometimes is exactly possible and other times can only return an enharmonic equivalent to the result: for example, $B\sharp_0/3 = E_0$, as per the previous example, but C_1 (which is enharmonic to $B\sharp_0$) divided by 3 does not have a solution proper, since there is no interval that, stacked three times, forms an octave. [11] Yet, taking inspiration from what happens with integer division, where (save of course for the case of division by 0) a result is always defined even if it entails approximation, as in $7 \div 3 = 2$, we implemented pitch division so that it always produces a result, with a potential enharmonic approximation: so, $C_1/3$ will anyway return E_0. A corresponding pitch modulo operation returns the remainder of such division, always an enharmonic equivalent of the unison. Multiplication and division of a pitch by rational numbers then immediately follow as combinations of integer multiplications and divisions. Quotitive division of a pitch by a pitch is also possible, telling how many times a given interval contains another — and always allowing for enharmonic approximation when needed. For all the operations and mathematical functions that do not have a specific meaning with respect to pitches, the pitch is preliminarily converted into MIDIcents. This ensures that, for example, summing an actual pitch and a value in MIDIcents (as in C_5+500) produces a meaningful value (in the example, 6500). [12]

[11] Of course this implies that, in agreement with classical Western music theory, we are considering enharmonic distinctions in the picture, something that sometimes escapes more informal discourses about intervals.

[12] The fact that, in the example in parentheses, the return value of the

Pitch comparisons also deserve a brief explanation. Equality is trivial: two pitches are considered equal if all their components are equal. So, B♯$_4$ is considered different from C$_5$, although they are enharmonically equivalent and both correspond to 6000 MIDIcents. Converting to MIDIcents and comparing for equality is also possible: in this case, the two pitches will be considered equal. As for inequalities, the ordering is always firstly defined by the octave, subsequently by the white key if the octave is the same, and finally alterations are taken into account only if the two previous values are the same. For example, if we compare F𝄫$_0$ with D𝄪$_0$, the former will be considered greater, because the white key is. On the other hand, if those pitches are converted to MIDIcents, then the second one is greater, as it corresponds to 400, while the first is 300.

Besides the arithmetic operations, *bach* features a number of other modules dealing with pitches, for finding enharmonic equivalents, constructing computationally pitches from various kinds of data and so on. Generally speaking, the easiest way to represent a pitch in *bach* is just through a text symbol resembling the ones above, and composed by a mandatory white-key name (`A`, `B`, `C`, `D`, `E`, `F`, or `G`), an optional sequence of accidentals expressed through specific characters (`#` for sharp, `b` for flat, `q` for one quarter-tone up and so on, covering ascending and descending chromatic semitones, quarter-tones and eighth-tones), a mandatory octave number and an optional further deviation, expressed in tones or fractions thereof, followed by the letter `t`. Examples of valid pitch symbols (that, as stated above, can also be considered as the representation of intervals corresponding to their respective distances from C$_0$) include the following: `C3`, `D#4`, `Fq0` (F$_0$ plus one quarter-tone), `G#v6` (G♯$_6$ minus one eighth-tone), `Bbb^5-3/19t` (B𝄫$_5$ plus one eighth-tone minus 3/19 of a tone).

Besides the actual pitch data type, all the objects dealing with notation also accept and understand MIDIcents.

3.2 Limitations of the *bach* pitch representation

The pitch system of *bach* discussed above is essentially focused on the representation of equal-tempered, octaviant tuning systems. More specifically, it is straightforward to express through it semitones, quarter-tones and eight-tones, since there are specific accidental symbols for them. It is not difficult to express any other kind of division of the semitone, by specifying the rational deviation from a reference pitch through the `t` part of the pitch symbol. Moreover, one can easily define custom structures of MIDIcents associated with correspondingly custom structures of frequencies. Formally, because there is a relatively simple and universal definition of the mathematical relation between cents and frequency, one should assume to define their custom MIDIcents structure so as to reflect the desired frequency structure, and vice versa. Practically, if one considers the MIDI representation as a numeric encoding of the layout of a physical instrument, then it can be perfectly meaningful to elude this mathematical relation altogether.

In fact, one can almost as easily associate note names and accidentals with arbitrary frequencies, conceptually not unlike re-tuning a piano to whatever collection of fundamental frequencies one may fancy. As long as a concept of octave and a subdivision in 12 non necessarily equal steps is preserved, such an operation poses no special problems, although it may as well be that processes often given for granted (such as transposition) cannot be meaningfully applied any longer.[13] It should also be mentioned that such an operation would require the user to choose univocally between, say, C♯ and D♭: if the black key between C and D is tuned as a C♯ (calculated, for example, as a 5:4 ratio starting from A), then D♭'s (calculated, for example, as a 4:5 ratio starting from F) are out of the game, and vice versa. For this reason, the whole pitch system of *bach*, with its accounting for virtually infinite enharmonic distinctions,[14] is rendered useless and the user is better off to stick to MIDIcents, since a value of 6100 (the black key placed at the immediate right of the middle C) only has a single possible interpretation in this scenario. Accounting for more keys in an octave, as in split-sharp harpsichord keyboards, would not change the gist of the reasoning, but only require more MIDIcents values mapped to frequency values within an octave.

Still, this is only a part of the representation problem: the other part is semantic, and has to do with the inherent relations and affordances that are built into a pitch structure, and the operations one can apply to it. In this sense, the more one strays away from the aforementioned kinds of division of the octave, the more the representation becomes cumbersome and problematic, at least within the context of *bach*. Dividing the octave in a number of steps that is not a multiple of 12 (that is, creating an octaviant ET system that is not based upon the semitone) delegitimates both note (and accidental) names and the diatonic staff representation. Non-octaviant ET systems delegitimate the octave concept altogether. Moreover, in such scenarios all the pitch operations described above lose their meanings. Established systems such as ET Bohlen-Pierce generally have their naming schemes and notational practices, but custom ET systems, that can be defined by just plugging in the equation arbitrary generator ratio and num-

sum is 6500 rather than F$_4$ is linked to the problem of enharmonicity: whereas intuitively we may think that 500 MIDIcents and F$_0$ (and hence a perfect fourth) are equivalent, they are not: in an enharmonically aware system, 500 MIDIcents may as well be E♯$_0$ (an augmented third), G𝄫$_0$ (a doubly-diminished fifth) and infinitely many more enharmonic variants.

[13] One specific and very relevant example of this scenario is the tuning of keyboard instruments for music predating the ET era. Most of the times, the keyboards of such instruments featured an arrangement of keys exactly identical to the one of the modern piano, with all the embedded knowledge of a diatonic seven-note scale punctuated by a specific structure of five altered pitches and repeating at distances of octaves. As hinted at above, such instruments can be tuned to a plethora of different kinds of tunings, something that historically informed harpsichord players actually do all the times, according to the specific repertoire. Once the instrument is tuned, though, it can only play in a few selected keys, and the modern concept of transposition, by which a skilled pianist can just move any piece of music up and down the keyboard in whatever tonality she or he likes, is just inapplicable, as the same operation on an instrument with such a tuning will almost invariably result in horrendously dissonant chords and melodies.

[14] Technically, due to the limited precision of numeric representation in computers, the amount of enharmonic possibilities — and of pitches altogether — in *bach* is not infinite. On the other hand, it is hardly imaginable that in a realistic musical scenario the limits of their potential quantity are hit, or even just approached.

ber of steps, are even more problematic: naming systems notations addressing broader families of tunings have been proposed, but their implementation would be difficult to say the least. The ideal solution would be to provide users with ways to implement their own pitch grammars, arithmetics and notation system, something along the lines of MaxScore's *scorepions* [1], but such a device would risk to pose other problems, especially in terms of efficiency. The simple truth is that, as of now, *bach* is not currently well-suited to such kinds of scenario.

4. JUST INTONATION PITCHES IN *bach*

The kinds of problems with generalised ET systems that have been discussed above are magnified when it comes to representing generalised JI systems. As hinted at above, one specific historical just tuning system may as well be implemented as a specific MIDIcents-to-frequency mapping. On the other hand, the generalisation of just intonation systems, with their arbitrarily expanding series of commas and potential interpretations of the same note names and piano keys, poses substantially different problems from the ones that are faced in ET representation.

These considerations, combined with a personal fascination with the aesthetics and formalism of just-intonation music, has prompted us to embark in the experimental project of supporting generalised just intonation through the HEJI system described above. [15]

The implementation of just intonation has required a thorough rethinking of the pitch data type and its corresponding operations. Whereas a preliminary hypothesis was to keep the extant pitch type unchanged and add a further type for JI, careful reflection prompted us to embed the two representation in a single data type. Although this has caused some additional complexity in the codebase, it also poses great practical advantages that will be discussed further on.

4.1 Textual syntax: curly brackets

The fundamental principle is that, at the lowest level, a pitch in *bach* is composed of two separate parts — an equal-tempered part (already described in [14]) and a justly tuned part. Both parts can be represented with letters, accidentals and octaves, and to distinguish between them we use curly brackets. For instance, `E5` is the tempered E, 400 cents away from middle C, while `E{}5` is about 408 cents away from middle C, or more precisely a $81/64$ ratio, corresponding to a Pythagorean major third. We shall discuss now the JI part, and leave for a further section an explanation of how the combination of ET and JI works.

C's have a special status of 'amphibian' entities, in that they constitute the reference point for both the *bach* ET and JI systems: `C0` corresponds to 0 cents. Similarly, `C0` as a justly tuned pitch has an interval of $1/1$ from the equal-tempered `C0`, so it corresponds to it. `C1` is both equal tempered and just, with an interval of $2/1$ (octave), and so on. All C's will be both equal tempered and just so that `C5` and `C{}5` are exactly the same thing: *bach* will understand both representations, but will always output the former.

Working with ratios with respect to `C0` is of course consistent, but perhaps cumbersome operationally, and one can change the reference pitch in several ways. A powerful one is by writing an explicit frequency ratio in terms of a rational deviation; for instance, to get a pitch a pure major third above `G{}5`, one can write `G{}5+5/4r`. As a matter of fact, one can consider the `r` part as the non-logarithmic counterpart of the aforementioned `t` component representing the alteration expressed in terms of fractions of a tone (e.g., `E5+7/12t`). The `r` part can as well appear on its own, as a simple way to enter JI pitches directly in terms of their generator fraction. Thus, writing `48r` is equivalent to writing `G{}5`, that is, $48 = 2^4 \times 3$, that is, 4 octaves plus one twelfth from C0.

Notice that `G{}5+5/4r` does not correspond to the Pythagorean `B{}5`, but actually differs by it precisely by one descending syntonic comma. A more convenient way to notate this specific note is as `B{-1}5`: we are starting to use the space between curly brackets to account for prime commas, and the first term that appears corresponds to the 5-limit deviation, measured in number of syntonic commas (81/80). Similarly, a pure major third over middle C we find `E{-1}5`.

One can introduce additional commas, each corresponding to a further prime number, by concatenating them via colons. For instance, a harmonic seventh (the interval between the 4th and 7th partials of a harmonic spectrum) above middle C we find `Bb{0:-1}5`, where 0 signals that there are no syntonic commas, and -1 stands for the lowering by a septimal comma (64/63). This corresponds to the decomposition of $7/4$ into a Pythagorean and a septimal part, whose exponent is precisely -1:

$$\frac{7}{4} = \frac{16}{9} \times \left(\frac{64}{63}\right)^{-1}.$$

Of course one can lower or raise by more than one comma at a time; and you can pile up commas up to the 47-limit, which is the highest prime for which the newest version of HEJI specifications has an official representation [12]. For instance, the $7/4$ ratio raised by an undecimal comma (almost a quarter tone) gives `Bb{0:-1:1}5`.

4.2 Just Intonation Arithmetics

An arithmetic system for JI pitches, parallel to the ET one, has been implemented. The fundamental concept that lies at the heart of JI arithmetics is that any possible representation of a JI pitch is equivalent to a rational number expressing the frequency ratio: the various different representations (exponents of prime factors; note name with optional commas and sharps or flats; Helmoltz-Ellis, Sagittal or Johnston's; or any combination of those) are just different ways of writing that rational number. With this in mind, we can briefly discuss the main principles of JI arithmetics in *bach*.

[15] Alongside just intonation, the forthcoming version of *bach* introduces a third notation type, with a linear vertical pitch domain, as in a true piano-roll scenario where all the semitone steps have the same graphical distance. This notation type, which will not be discussed further here, is meant among other things to enable more flexible display of non-standard tuning systems.

- Sums and subtractions of JI pitches produce transpositions, thus they amount to respectively multiplying and dividing the corresponding frequency ratios.

- Multiplications and divisions of a JI pitch by an integer or a rational make more sense if the pitch is thought of as an interval. They amount to raising the frequency ratio of the pitch to respectively the integer or rational, or its opposite. This corresponds respectively to stack multiple JI intervals on top of each other, and to calculate how many times a given interval contains a different one (for example, `D{}1 / G{}0` returns 2, because a Pythagorean major ninth is composed exactly of 2 Pythagorean fifths).

- Equalities and inequalities are calculated by simply comparing the frequency ratios. Importantly, unlike with ET enharmonics, the result always corresponds to that of the comparison of the corresponding MIDIcents values.

- For all the other possible operations, the pitch is preliminarily converted into MIDIcents.

Through either direct expression or arithmetic operations, it is possible to generate a JI pitch that should in principle exceed the 47-limit that is hard-coded within the *bach* system. Because such a pitch cannot inherently be represented, there is an approximation mechanism that comes into play in this situation: the desired pitch is approximated to one with a smaller limit (47 by default, but a different one can be chosen), within a range of error that can be specified in MIDIcents (67 by default, roughly corresponding to one third of a tone).

4.3 Internal Representation of Pitches

As mentioned above, in the forthcoming version of *bach*, all pitches are actually composed of two parts: an ET one and a JI one. The ET part is represented internally in terms of the white-key class and a rational alteration. The JI part is represented internally in terms of its corresponding frequency ratio with respect to C0. The two parts share the octave component, because the octave is the only interval whose corresponding frequency ratio is the same, and as such it would make no sense to treat it separately. In fact the octave of any pitch is always stored internally within the JI part of the pitch. In the simplest cases, the exponent of the 2 factor actually coincides with the octave. On the other hand, this is not true in general: for instance, a just G_0 has a ratio of $3 : 2$ (or $3^1 \times 2^{-1}$), because the numerator 3 actually pushes the pitch to octave 1, and this must be compensated by the denominator 2 so as to bring back the pitch to octave 0.

Unlike the standard *bach* rational numbers, which are stored internally as a simple pair of integers, the rational representing the JI component of a pitch is stored in terms of a vector of exponents of the first 15 primes, from 2 to 47. This vector is precisely the vector we mentioned at the end of Section 2.1. The first 7 primes, from 2 to 17, are allocated 8 bits each, allowing for exponents from -128 to 127; the subsequent 8 primes, from 19 to 47, are allocated 4 bits each, allowing for exponents from -8 to 7.

This peculiar representation is motivated by the fact that the whole data structure for a pitch should fit into 16 bytes; one byte is taken by the ET white key [16]; four bytes are taken by the ET alteration rational number (we deemed that reducing the alteration to a two-byte rational number, with 8 bits each for numerator and denominator, was not enough for the precision that is expected from the system); this leaves 11 bytes for the JI part.

On the other hand, the JI part is not expected to be just any rational number whose numerator and denominator fall within a given range, as is the case for standard *bach* rationals as used, for instance, for the ET alteration. Rather, frequency ratio are conceived and computed in terms of the exponents of their prime factors. This means that the range of their numerator and denominator is potentially very wide (each would require more than 300 bits to be represented as an integer), but also extremely sparse, in that very few values over the range can be obtained as a product of such primes with exponents the allowed range described above. Thus, representing the ratio precisely in terms of its prime factor exponents is extremely more space-efficient than employing a standard rational, and hopefully not much more expensive in computational terms. If anything, the basic mathematical pitch operations are simpler: summing pitches, that is, multiplying ratios, actually amounts to summing 11 bytes of exponents rather than multiplying an almost 80-byte rational (more than 300 bits for the numerator and as many for the denominator) and then having to simplify the fraction. The same line of reasoning goes for pitch multiplication. As a matter of fact, albeit technically possible, a standard rational representation of JI ratios within the desired 47-limit would have required very complicated and inefficient workarounds.

The choice of the ranges for the individual exponents, and specifically the fact that lower prime factors exponents have a greater range that higher ones, has been motivated by space considerations, and by the observation that low primes such as 2 and 3 practically require a relatively wide range of exponents (a HEJI $A\sharp_0$, for instance, is $2^{-15} \times 3^{10}$), whereas prime factors higher than 13 or 17 are very seldom used, and if they are they probably have very small exponents.

An important check that is carried out at the beginning of virtually every operations on a pitch is whether the concerned pitch is ET-pure or JI-pure, that is, respectively, if the JI exponent vector is 0 for all the values except optionally the first (which means that only the octave JI component is present); and if the ET white key and alteration are both 0. Since, in practice, most pitches are expected to be ET- or JI-pure, this check is specifically meant to avoid the extra computational burden of dealing with the complexity of the JI part for pitches that do not have one, save for the octave. Among other things, ET-pure pitches always have their octave coinciding with the exponent of the 2 factor,

[16] Actually, 3 bits would be sufficient for representing 7 possible values, but using the remaining 5 bits for other purposes would have required deeper and more extensive and dangerous changes to the whole *bach* codebase than we were willing to face.

thus making their manipulation even easier. As mentioned above, C's are exceptional in that they are both ET-pure and JI-pure.

4.4 Mixing just and equal-tempered parts

The ability of expressing non-pure pitches has been added in order to keep the *bach* type system simpler; to avoid a proliferation of object attributes to distinguish between the types of accepted and returned pitches; and, more importantly, to allow for the specific musical problem of setting the reference for individual JI pitches.

Even more than ET pitches, JI pitches are inherently intervals. A frequency ratio is not a frequency: it is a deviation from a reference frequency, corresponding to a reference pitch. The reference pitch may as well be a JI pitch, but at some point, as in the myth of the turtle supporting the world, an arbitrary, non-relative pitch must be given. In terms of the *bach* world, this means that this base reference must be an ET pitch. The choice of C_0 is the most elegant, because it is both an ET and a JI pitch, and also because it corresponds to 0 MIDIcents, which is a natural starting point. On the other hand, it is not the only possible choice. To express, for example, a pitch located a just fifth above A_3, we can simply write `A3+G{}0`. In fact, entering a sum or a subtraction of an ET and a JI component will result in a non-pure pitch. Incidentally, when the expression is parsed and the pitch is generated and stored, the octave number that here is attached to the ET part will be summed to the one attached to the JI part (0 in this case) and to the optional frequency ratio, and stored as the exponent of prime 2 of the JI part.

Arithmetics of non pure-pitches works according to the same rules as for pure pitches, with the two parts computed separately and then summed together again. On the other hand, the two parts are not fully independent as they share the octave specification. This means that the same pitch could be represented for instance as `A3+G{}0`, `A0+G{}3` or even `A1+G{}2`. All those writings are actually equivalent and will have the same internal representation. When required to output a textual representation of non-pure pitches, *bach* will follow a standard form by which the octave number is attached to the ET part except in the case of JI-pure pitches.

Non-pure pitches can be also leveraged to solve elegantly specific musical problems, such as representing the harmonic series of a given ET pitch such as `A3` — something that can be expressed as, for instance, the sequence `A3`, `A3+2r`, `A3+3r`, `A3+4r` and so on.

5. AN ECOSYSTEM FOR JUST INTONATION

Just intonation comes with a set of objects dedicated to make it work properly.

There are two objects that can be seen as Swiss-army knives for pitches. One is aptly called bach.pitch, and it performs various kinds of conversion, query, packing and unpacking for pure and non-pure pitches. It can, among other things, separate the individual components of a pitch (white-key class for the ET and JI parts, ET alteration, JI commas, octave); convert a pitch in MIDIcents; and retrieve the frequency ratio or exponent vector for a JI pitch. Also, it can build pitches starting from all those separate components.

The other general-purpose object is bach.eval. This is a very advanced object, implementing a small functional programming language called *bell* and meant to facilitate the expression of complex algorithmic problems in *bach* [15]. Among its features, *bell* provides a set of functions operating on pitches and reproducing within the context of the language the various possible behaviours of bach.pitch, and more. As a matter of fact, *bell* is probably the most advanced and comprehensive way to deal algorithmically with the complexity of pitch representation.

bach.eval can perform all the arithmetic operations mentioned above, but they are also accessible from simpler dedicated modules such as bach.+, bach.- and so on, as well as the bach.expr object, which is a general evaluator for mathematical expressions. [17]

A way to generate sequences of rational numbers is the bach.fareyser module, which produces Farey series, i.e., ordered sequence of reduced fractions with denominators less than or equal to some number. These sequences are often used by composers interested in just intonation and spectralism as 'reservoirs of consonances'.

Another important facet of dealing with rational numbers is to handle their approximations. The *bach* package already includes bach.approx, approximating to equal tempered grids, but approximating just intonation ratios is much trickier as rational numbers are dense in the real number line. To this end, we have included bach.continuedfraction, computing rational approximations by using the convergents (and, optionally, semiconvergents) of continued fractions. This object is not just useful to approximate rational numbers with simpler ones, but rather any floating point number, and indeed approximations of irrational number have profound significance [16], as exemplified in Figure 5. [18]

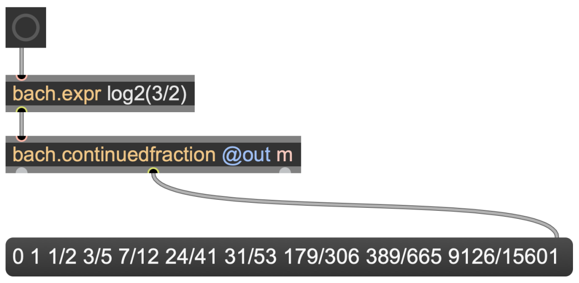
In order to explore Partch's "large table of aromatic and unusual viands", we have added a brand new object, called bach.jiwheel, where one can explore pitches and intervals interactively by zooming and selecting. The bach.jiwheel object shows JI pitches up to a given limit arranged in a circular fashion, and returns visual information about their graphical representation and mathematical properties.

Finally, one problematic aspect with tunings other than 12-EDO is the difficulty of handling them in MIDI-based contexts. For a long time, the only practical way to play even just quarter-tones was to set up clumsy contraptions of MIDI channels and pitch bend: this is, among other things, how microtonal playback works in OpenMusic, and how it is implemented in the bach.ezmidiplay module.

Luckily, in recent time, a number of microtonal MIDI protocols have emerged to ease this process. In partic-

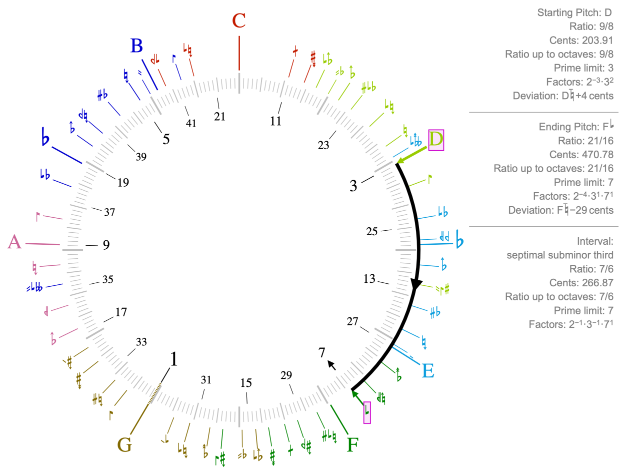
[17] Actually, the simpler arithmetic modules are wrappers for bach.expr.

[18] The *bach* package also already included the bach.float2rat module, designed for rational approximation of floating point numbers, according to different types of criteria, such as error thresholds and maximum denimonators.



```
bach.expr log2(3/2)
bach.continuedfraction @out m
0 1 1/2 3/5 7/12 24/41 31/53 179/306 389/665 9126/15601
```

Figure 5. The continued fraction convergents for $\log_2 3/2$ are profoundly connected with equal temperaments that best approximate the fifth $3/2$. You will notice, in particular $7/12$ (corresponding to the common 12-EDO, whose perfect fifth is on the 7th step) and $31/53$ (corresponding to 53-EDO, whose fifth is on the 31th step).



Starting Pitch: D
Ratio: 9/8
Cents: 203.91
Ratio up to octaves: 9/8
Prime limit: 3
Factors: $2^{-3} \cdot 3^{2}$
Deviation: D♮+4 cents

Ending Pitch: F
Ratio: 21/16
Cents: 470.78
Ratio up to octaves: 21/16
Prime limit: 7
Factors: $2^{-4} \cdot 3^{1} \cdot 7^{1}$
Deviation: F♮−29 cents

Interval:
septimal subminor third
Ratio: 7/6
Cents: 266.87
Ratio up to octaves: 7/6
Prime limit: 7
Factors: $2^{-1} \cdot 3^{-1} \cdot 7^{1}$

Figure 6. The bach.jiwheel object provides an interactive way to explore just intonation pitches and intervals, while monitoring their mathematical relationships.

ular, the MTS-ESP protocol [19] is becoming a standard, and it is now implemented in a large number of plugins and applications. To take advantage of it, we have endowed bach.playkeys (the object responsible for extracting and formatting playback information from the output of notation editors) with the ability to act as a MTS-ESP master, handle all microtonal adjustments internally, and communicating them on a note-by-note basis to MTS-ESP-compliant plugins. This allows seamless microtonal playback of *bach* scores with very little effort on the user's side. There are inevitable limitations anyway, due to the fact that all these are essentially hacks on the MIDI protocol, and there may be situations in which a note cannot be reproduced. On the other hand, Max allows one to build their own sound generation modules directly in the environment. Users willing to do so will be able to overcome all the limitations of MIDI.

[19] https://oddsound.com

6. EXAMPLES

As a first, simple example of application, consider the so-called 'Benedetti's puzzle': a looping progression of four chords that, when performed in rigourous JI, prompts the tuning to raise by a syntonic comma every time [17]. The patch in Figure 7 shows how, starting from the initial cell, one can build this comma-raising sequence just by using bach.+ C{1}0.

As a second example, consider the patch in Figure 8, which uses the concept of Farey sequence (also connected to mediants) to produce a descending sequence of intervals, from the octave to the unison, along with the fundamentals that their respective harmonic series underpin.

7. CONCLUSIONS AND FUTURE WORK

As mentioned above, at the time of writing the pitch system is fully implemented and is undergoing an extensive testing phase. We will hopefully be able to publicly release the new version of *bach* before the end of 2025.

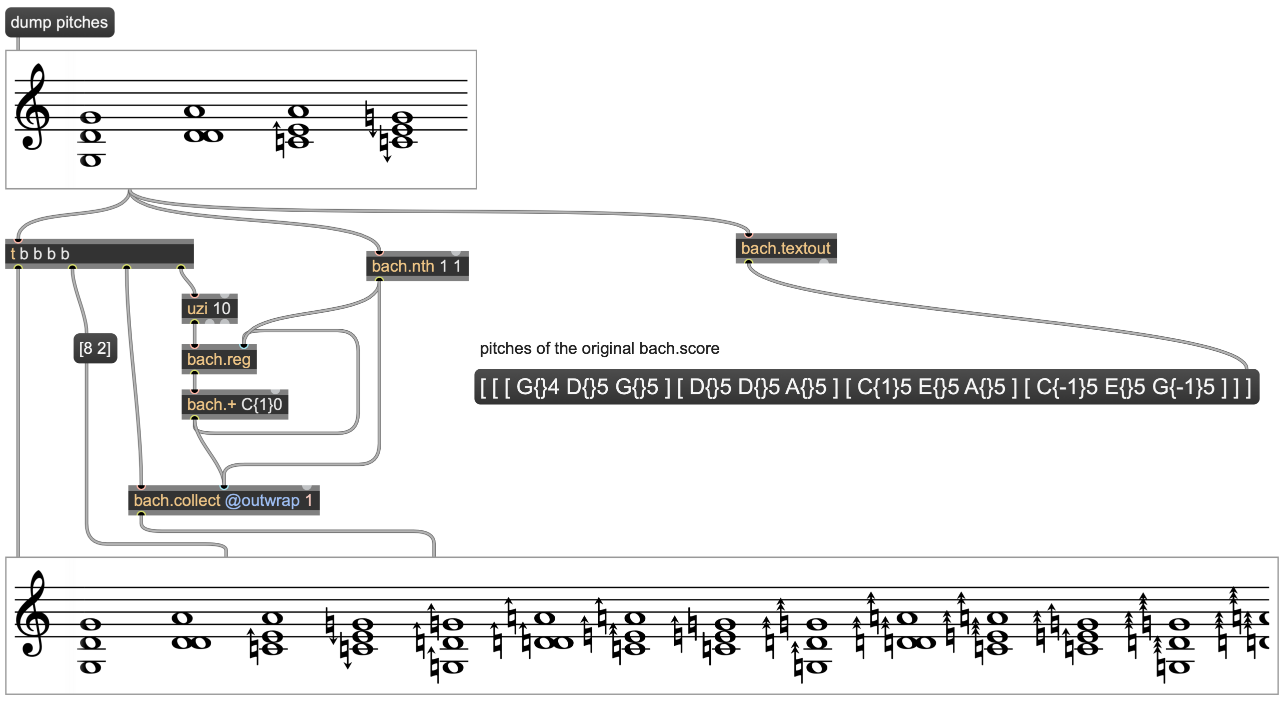
This being said, the current implementation of JI pitches can be seen as the starting point for a much ampler work on supporting more features related to pitch representation and tuning systems. One interesting addition could be supporting Scala files. On the other hand, those are already supported by Max's native objects mtof and ftom, and their very nature — just a list of cents or frequency ratios — makes them only adapted to the basic way of dealing with tuning systems other than 12-TET through, so to speak, "retuning the piano strings", that is, associating arbitrary MIDIcents values to the data describing each specific scale or tuning.

In order to complement this limitation, we could add native support for at least a few established non-standard tunings such as the aforementioned Bohlen-Pierce. On the other hand, the notation editors should integrate the respective accidental sets and general paradigms of behaviour; and, even more importantly, the *bach* type system should be enhanced so as to accommodate these additional tunings — something probably not impossible, but surely rather complicated.

Finally, *bach* is at the core of a wider ecosystem of Max packages, also including *cage* (for high-level musical processes), *dada* (for creative graphical interfaces and data corpus management) and *ears* (for non-realtime algorithmic audio processing). It would definitely be useful to at least include in cage some new tools for making the usage of the admittedly complex new pitch system easier and less intimidating.

8. REFERENCES

[1] G. Hajdu and N. Didkovsky, "Maxscore: Recent developments," in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 138–146.



```
[ [ [ G{}4 D{}5 G{}5 ] [ D{}5 D{}5 A{}5 ] [ C{1}5 E{}5 A{}5 ] [ C{-1}5 E{}5 G{-1}5 ] ] ]
```

Figure 7. So called ‘Benedetti’s puzzle’ in *bach*’s just intonation: in order to maintain the purity of the harmonic intervals, every repetition of the phrase must be shifted up by one syntonic comma.

[2] J. M. Barbour, *Tuning and Temperament: A Historical Survey*. Michigan State College Press, 1951.

[3] K. Gann, *The Arithmetic of Listening: Tuning Theory and History for the Impractical Musician*. University of Illinois Press, 2019.

[4] P. Barbieri, *Tuning and Temperament: Practice vs Science. 1450-2020*. Gangemi Editore, 2023.

[5] D. Tymoczko, *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice*. Oxford University Press, 2011.

[6] D. Blackwood, *The Structure of Recognizable Diatonic Tunings*. Princeton University Press, 1985.

[7] J. Tenney, *A History of ‘Consonance’ and ‘Dissonance’*. Excelsior Music, 1988.

[8] P. Oliveros, “My “american music”: Soundscape, politics, technology, community,” *American Music*, vol. 25, no. 4, pp. 389–404, 2007.

[9] H. Partch, *Genesis Of A Music: An Account Of A Creative Work, Its Roots, And Its Fulfillments (2nd Edition)*. Da Capo, 1979.

[10] K. Gann. (2018) How to Use Ben Johnston’s Just Intonation Notation. [Online]. Available: www.kylegann.com/BJNotation.html

[11] G. D. Secor and D. C. Keenan, “Sagittal: A microtonal notation system,” *Xenharmonikôn, An Informal Journal of Experimental Music*, vol. 18, 2006.

[12] M. Sabat, *The Extended Helmholtz-Ellis Pitch Notation*. Plainsound Music Edition, 2005.

[13] A. Agostini and D. Ghisi, “A max library for musical notation and computer-aided composition,” *Computer Music Journal*, vol. 29, no. 2, pp. 11–27, 2015.

[14] ——, “Pitches in bach,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR’18*, S. Bhagwati and J. Bresson, Eds. Montreal, Canada: Concordia University, 2018, pp. 128–137.

[15] A. Agostini and J.-L. Giavitto, “Bell, a textual language for the bach library,” in *Proceedings of the 2019 International Computer Music Conference*. Michigan Publishing Services, 2019.

[16] D. Benson, *Music: A mathematical offering*. Cambridge University Press, 2006.

[17] R. W. Duffin, “Cipriano de Rore, Giovanni Battista Benedetti, and the Just Tuning Conundrum,” *Journal of the Alamire Foundation*, vol. 9, no. 2, pp. 239–266, 2017.

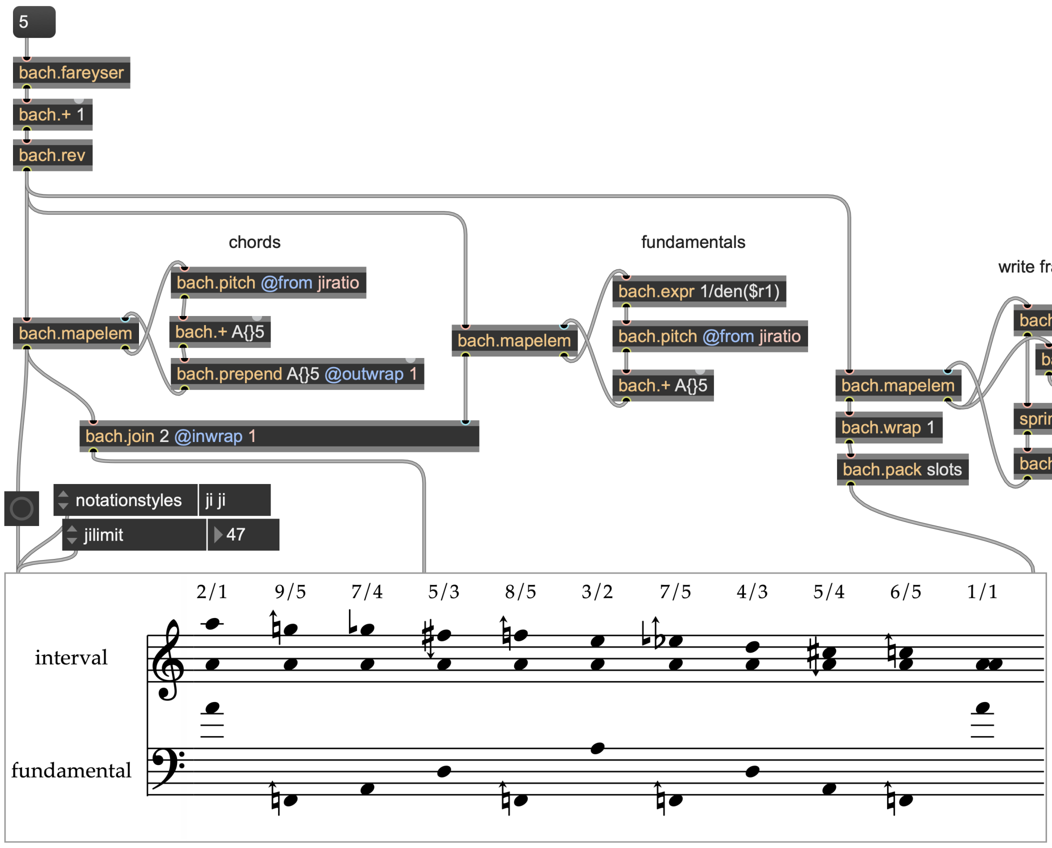


Figure 8. A descending Farey sequence of fractions built over `A{}5`, with the fundamentals of their corresponding harmonic series highlighted in the lower staff.