A Family of Libraries for Pitches and Intervals

Christoph Finkensiep[1], Robert Lieck[2], and Martin Rohrmeier[1]

[1] Digital and Cognitive Musicology Lab, École Polytechnique Fédérale de Lausanne
[2] Durham University

A Set of Libraries

Working with pitches and intervals can be difficult, especially with "spelled" pitches such as D♭ and the exact intervals between them. We have created a set of libraries that offer representations and operations for different interval types in various programming languages:

```
/DCMLab/pitchtypes
        haskell-musicology
        Pitches.jl
        purescript-pitches
        rust-pitches
```

These libraries provide:

- representations for different pitch and interval types,
- a generic API for manipulating pitches and intervals,
- special support for spelled pitch and intervals.

Pitch and Interval Spaces

Pitches and intervals can be considered points in a space and the vectors between them. They thus support very similar operations to points and vectors, as well as operations specific to musical pitch:

- addition ($i + i$), subtraction ($i - i$), negation ($-i$), and integer multiplication ($i \cdot \text{int}$) of intervals;
- computing the interval between two pitches ($p - p$);
- transposing a pitch by an interval ($p \pm i$);
- ordering ($p < p$, $i < i$) and direction $\sigma(i)$;
- projection to (octave-independent) pitch or interval classes ($i \to ic$, $p \to pc$), and a canonical inverse embedding ($ic \hookrightarrow i$, $pc \hookrightarrow p$);
- special intervals such as unison (neutral) and octave.

With these operations, algorithms can be specificed generically, independent of the pitch space:

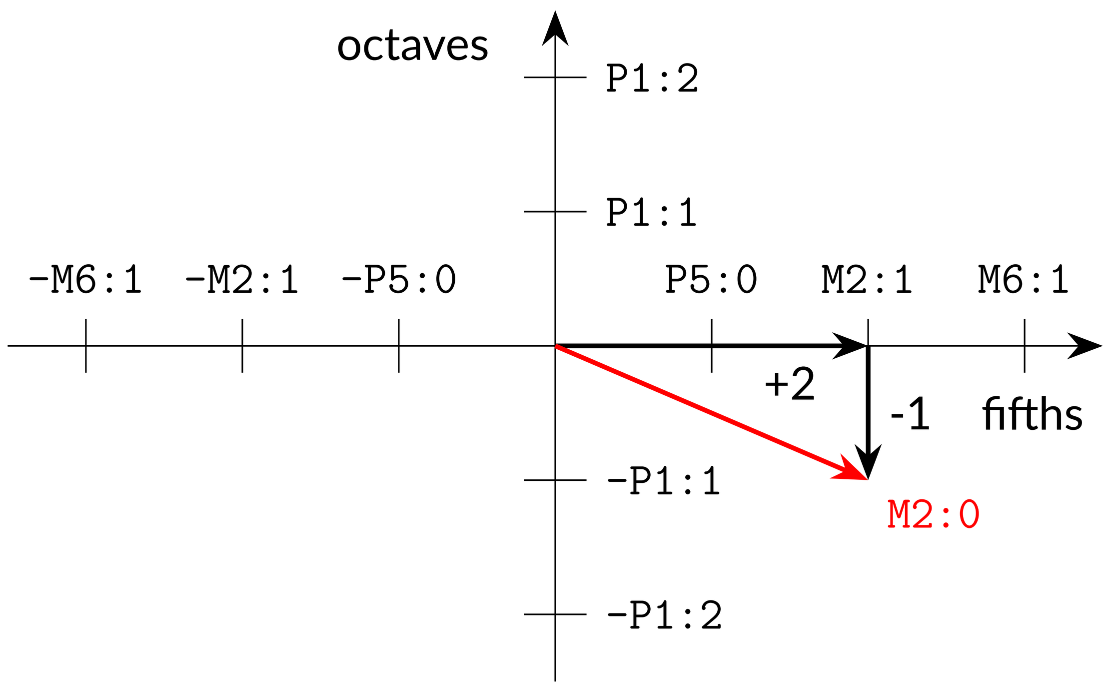
```
def relative_to_key(pitches, root):
    return [p.to_class() - root
            for p in pitches]

>>> relative_to_key([pt.SpelledPitch(p)
...         for p in ["C4", "Eb3", "G#3"]],
...      pt.SpelledPitchClass("D"))
[m7, m2, a4]

>>> relative_to_key([pt.EnharmonicPitch(p)
...         for p in [0,3,7]],
...      pt.EnharmonicPitchClass(2))
[10, 1, 5]
```

Spelled Pitches and Intervals

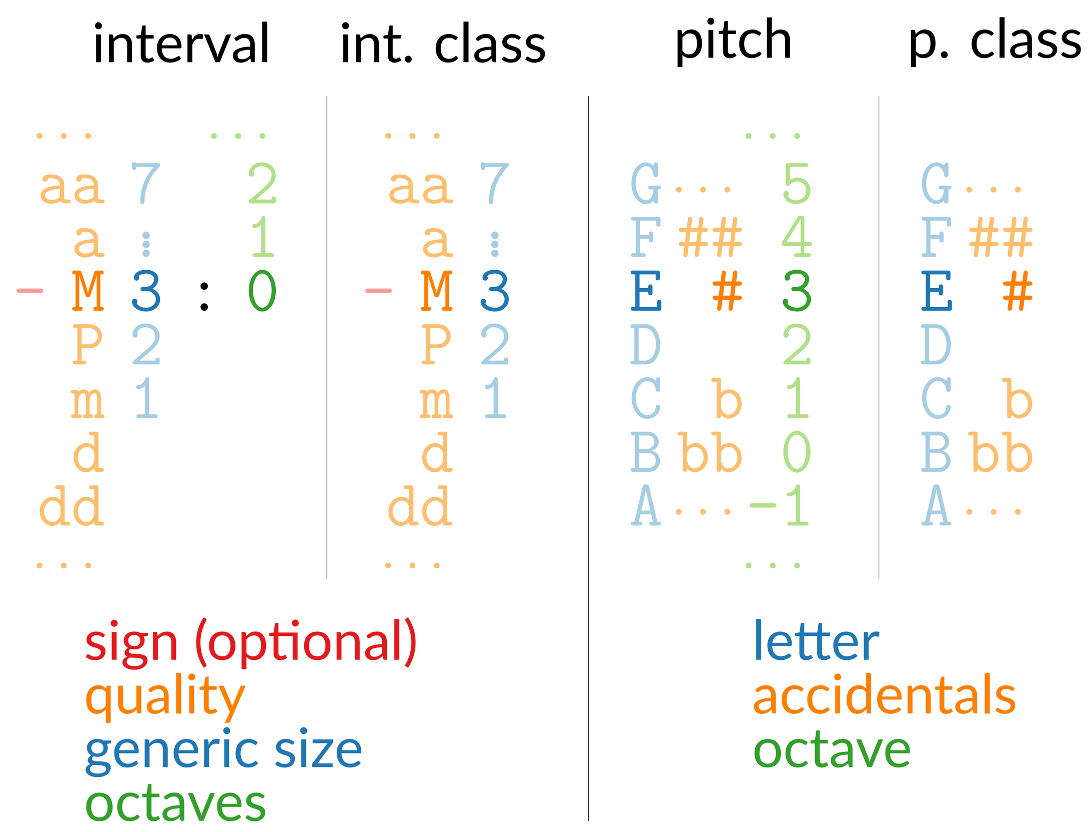
Spelled intervals are two-dimensional. They are encoded as sums of perfect fifths and perfect octaves. For example, a major second up (`M2:0`) is encoded as two fifths up (major ninth, `M2:1`) plus one octave down.



In addition to the general API, spelled pitches and intervals have special support for generic interval sizes, alterations, and string notation (see below).

Notation

All libraries implement a common string notation for spelled pitches and intervals:



Vectorized Types

The Python library supports vectorized variants for spelled types that internally use `numpy` arrays for fast operations. These array types can have arbitrary shapes.

intervals:
$$\begin{bmatrix} \texttt{P1:0} & \texttt{P5:0} \\ \texttt{M2:0} & \texttt{M6:0} \\ \texttt{M3:0} & \texttt{M7:0} \\ \texttt{P4:0} & \texttt{P1:1} \end{bmatrix}$$

internal arrays — fifths:
$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \\ -1 & 0 \end{bmatrix}$$

octaves:
$$\begin{bmatrix} 0 & 0 \\ -1 & -1 \\ -2 & -2 \\ 1 & 1 \end{bmatrix}$$

Array types support the standard APIs for collections as well as `numpy`s advanced indexing. Operations involving both arrays and scalar types are broadcast. Array types work with `pandas` dataframe columns and can be used to efficiently work with large notelist-like datasets.

One-Hot Encoding

Machine-learning applications often require pitches to be represented in a one-hot (or multi-hot) encoding. For spelled pitch and interval classes, these encodings are one-dimensional, representing a range on the line of fifths. Here is the encoding of a `G` with a fifth range of $[-2, 2]$:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \begin{matrix} -2 & (\text{B}\flat) \\ -1 & (\text{F}) \\ 0 & (\text{C}) \\ 1 & (\text{G}) \\ 2 & (\text{D}) \end{matrix} \quad \text{fifths}$$

For spelled pitches and intervals, the one-hot representation is two-dimensional, using fifths and *independent* octaves (i.e., as written) as dimensions. Here you can see the encoding of a `G3` with a fifth range of $[-2, 2]$ and an octave range of $[3, 5]$:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{matrix} -2 & (\text{B}\flat) \\ -1 & (\text{F}) \\ 0 & (\text{C}) \\ 1 & (\text{G}) \\ 2 & (\text{D}) \end{matrix} \quad \text{fifths}$$

3 4 5
octave

This representation is friendly to convolutional filters since fixed spacial relations correspond to constant intervals. When produced from array types, one-hot arrays will add the fifths and/or octaves dimension to the original array shape.

Funding

We thank Claude Latour for generously supporting this research. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 760081 – PMSB. This project was further supported by the Swiss National Science Foundation within the project "Distant Listening – The Development of Harmony over Three Centuries (1700–2000)" (Grant no. 182811).

Swiss National Science Foundation

European Research Council
Established by the European Commission