

LilyPond Contemporary Notation Cookbook: Snippets and Their Grammars

Yoshiaki Onishi
School of Music, University of Delaware
info@yoshionishi.com

Version: December 23, 2025

MIT License

©2024 by Yoshiaki Onishi.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Cite: Onishi, Yoshiaki. “LilyPond Contemporary Notation Cookbook: Snippets and Their Grammars,” (Version December 23, 2025), GitHub repository, <https://github.com/yoshiakionishi/lilypond-snippets>

Contents

Foreword	viii
0.1 Preamble	viii
0.2 README	viii
0.3 Background	ix
0.4 How This Document Is Structured	x
0.5 LilyPond Version Used	x
0.6 Acknowledgements	xi
1 Articulations	1
1.1 Accent-Staccato	1
1.1.1 Description	1
1.1.2 Grammar	1
1.1.3 Code	1
1.2 Accent-Tenuto	4
1.2.1 Description	4
1.2.2 Grammar	4
1.2.3 Code	4
1.3 Jeté (Ricochet)	6
1.3.1 Description	6
1.3.2 Grammar	6
1.3.3 Code	6
2 Beams	8
2.1 Wiggle Beam (zig-zag shaped beam)	8
2.1.1 Description	8
2.1.2 Grammar	8
2.1.3 Code	9
2.1.4 Discussion	13
3 Clefs	14
3.1 String Position Clef	14
3.1.1 Description	14
3.1.2 Grammar	14
3.1.3 Code	14
3.1.4 Discussion	16
3.2 String Position Clef (revised)	18

3.2.1	Description	18
3.2.2	Grammar	18
3.2.3	Code	18
3.3	String Position Clef 2	22
3.3.1	Description	22
3.3.2	Grammar	22
3.3.3	Code	22
3.3.4	Discussion	24
3.4	String Position Clef 3a	25
3.4.1	Description	25
3.4.2	Grammar	25
3.4.3	Code	25
3.4.4	Discussion	28
3.5	String Position Clef 3b (Longer Span)	29
3.5.1	Description	29
3.5.2	Grammar	29
3.5.3	Code	29
3.5.4	Discussion	32
4	Dynamics	33
4.1	Dynamics in Quotation Marks	33
4.1.1	Description	33
4.1.2	Grammar	33
4.1.3	Code	33
4.1.4	Discussion	36
4.2	Enhanced Constante Hairpins I (Under The Staff)	37
4.2.1	Description	37
4.2.2	Grammar	37
4.2.3	Code	38
4.2.4	Discussion	48
4.3	Enhanced Constante Hairpins II (Over The Staff)	49
4.3.1	Description	49
4.3.2	Grammar	49
4.3.3	Code	50
4.3.4	Discussion	60
4.4	Enhanced Flared Hairpins I (Original Version)	61
4.4.1	Description	61
4.4.2	Grammar	62
4.4.3	Code	62
4.4.4	Discussion	77
4.5	Enhanced Flared Hairpins II (Curvy Version)	78
4.5.1	Description	78
4.5.2	Grammar	78
4.5.3	Code	79
4.5.4	Discussion	94
5	Noteheads	95
5.1	Jet Whistle (for flute)	95

5.1.1	Description	95
5.1.2	Grammar	95
5.1.3	Code	95
5.2	Jet Whistle (for flute) (Version 2)	97
5.2.1	Description	97
5.2.2	Grammar	97
5.2.3	Code	97
5.3	Line as a Notehead	99
5.3.1	Description	99
5.3.2	Grammar	99
5.3.3	Code	99
5.3.4	Discussion	102
5.4	Line as a Notehead 2	103
5.4.1	Description	103
5.4.2	Grammar	103
5.4.3	Code	103
5.5	Noteheadless	108
5.5.1	Description	108
5.5.2	Grammar	108
5.5.3	Code	108
5.6	Slap Tongue, Type A	110
5.6.1	Description	110
5.6.2	Grammar	110
5.6.3	Code	110
5.7	Slap Tongue, Type B	111
5.7.1	Description	111
5.7.2	Grammar	111
5.7.3	Code	111
5.7.4	Discussion	111
5.8	Slashed Notehead	112
5.8.1	Description	112
5.8.2	Grammar	112
5.8.3	Code	112
5.9	Square Notehead	115
5.9.1	Description	115
5.9.2	Grammar	115
5.9.3	Code	115
5.10	Tone Cluster	117
5.10.1	Description	117
5.10.2	Grammar	117
5.10.3	Code	117
5.10.4	Discussion	120
5.11	Tongue Ram (for flute)	121
5.11.1	Description	121
5.11.2	Grammar	121
5.11.3	Code	121
5.11.4	Discussion	122
5.12	X In A Hollow Notehead	123

5.12.1	Description	123
5.12.2	Grammar	123
5.12.3	Code	123
6	Markups	125
6.1	Conducting Patterns	125
6.1.1	Description	125
6.1.2	Grammar	125
6.1.3	Code	125
6.2	Mute Sign	129
6.2.1	Description	129
6.2.2	Grammar	129
6.2.3	Code	129
7	Rhythm	130
7.1	Incomplete Tuplet Bracket for Irrational Time Signatures	130
7.1.1	Description	130
7.1.2	Grammar	130
7.1.3	Code	131
7.1.4	Discussion	132
7.2	Metric Modulation Equation (Regular Flag)	133
7.2.1	Description	133
7.2.2	Grammar	133
7.2.3	Code	134
7.3	Metric Modulation Equation (Straight Flag)	151
7.3.1	Description	151
7.3.2	Grammar	151
7.3.3	Code	152
8	Spanners	169
8.1	Arrow Lines	170
8.1.1	Description	170
8.1.2	Grammar	170
8.1.3	Code	170
8.1.4	Discussion	173
8.2	Boulez Tie	175
8.2.1	Description	175
8.2.2	Grammar	175
8.2.3	Code	175
8.2.4	Discussion	176
8.3	Dashed Arrow Lines	177
8.3.1	Description	177
8.3.2	Grammar	177
8.3.3	Code	177
8.3.4	Discussion	180
8.4	Dashed Flat Slurs	181
8.4.1	Description	181
8.4.2	Grammar	181

8.4.3	Code	181
8.5	Flat Slurs	185
8.5.1	Description	185
8.5.2	Grammar	185
8.5.3	Code	185
8.5.4	Discussion	188
8.6	Grace Note Brackets I	189
8.6.1	Description	189
8.6.2	Grammar	189
8.6.3	Code	189
8.7	Grace Note Brackets II	194
8.7.1	Description	194
8.7.2	Grammar	194
8.7.3	Code	194
8.8	Tempo Arrows	199
8.8.1	Description	199
8.8.2	Grammar	199
8.8.3	Code	199
9	Staff Lines	203
9.1	Expanding, Shrinking and Bloated Staff Lines	203
9.1.1	Description	203
9.1.2	Grammar	203
9.1.3	Code	203
10	Stems	207
10.1	"M" on Stem	207
10.1.1	Description	207
10.1.2	Grammar	207
10.1.3	Code	207
10.2	Mute Sign on Stem	209
10.2.1	Description	209
10.2.2	Grammar	209
10.2.3	Code	209
10.3	"S" on Stem	212
10.3.1	Description	212
10.3.2	Grammar	212
10.3.3	Code	212
10.4	"V" on Stem	214
10.4.1	Description	214
10.4.2	Grammar	214
10.4.3	Code	214
11	Time Signatures	216
11.1	Fractional Time Signatures, Style A	216
11.1.1	Description	217
11.1.2	Grammar	217
11.1.3	Code	218

11.2 Fractional Time Signatures, Style B	223
11.2.1 Description	223
11.2.2 Grammar	223
11.2.3 Code	224
11.3 Fractional Time Signatures, Style C	230
11.3.1 Description	230
11.3.2 Grammar	230
11.3.3 Code	230
11.4 Compound Meter with Two Fractional Time Signatures, Style A	235
11.4.1 Description	235
11.4.2 Grammar	235
11.4.3 Code	235
11.4.4 Discussion	240
11.5 Compound Meter with Two Fractional Time Signatures, Style B	242
11.5.1 Description	242
11.5.2 Grammar	242
11.5.3 Code	242
11.5.4 Discussion	249
11.6 Compound Meter with Two Fractional Time Signatures, Style C	250
11.6.1 Description	250
11.6.2 Grammar	250
11.6.3 Code	250
11.6.4 Discussion	256
11.7 Compound Meter with Three Fractional Time Signatures, Style A	257
11.7.1 Description	257
11.7.2 Grammar	257
11.7.3 Code	257
11.7.4 Discussion	264
11.8 Compound Meter with Three Fractional Time Signatures, Style B	266
11.8.1 Description	266
11.8.2 Grammar	266
11.8.3 Code	266
11.8.4 Discussion	275
11.9 Compound Meter with Three Fractional Time Signatures, Style C	276
11.9.1 Description	276
11.9.2 Grammar	276
11.9.3 Code	276
11.9.4 Discussion	284
11.10 Time Signature with Decimals	285
11.10.1 Description	285
11.10.2 Grammar	285
11.10.3 Code	286
11.10.4 Discussion	289
12 Combinations	290
12.1 Prescriptive Notation for String Instruments	290
12.1.1 Description	290
12.1.2 Variables Used	290

12.1.3 Code	290
12.2 Multiple Instances Of Spanners At Once	292
12.2.1 Description	292
12.2.2 Variables Used	292
12.2.3 Code	292
13 Miscellanies	295
13.1 Shifting Staves, Rotated Clef and Time Signature	295
13.1.1 Description	295
13.1.2 Code	295
14 Exploring Scheme	298
14.1 Introduction	298
14.1.1 Step 1a: Focus on the Scheme Language Itself	298
14.1.2 Step 1b: Get Used to Prefix Notation	298
14.1.3 Step 2: Study Lots of Snippets	299
14.1.4 Step 3: Hack the Codes	301
14.2 Example 1: Creating a Time Signature with Its Compound Meter Form	304
14.2.1 Step 1: Analyze What Could Be Automatized	306
14.2.2 Step 2: Write the Code	306
Bibliography	309
Appendices	311
Appendix A: Resources	312

Foreword

0.1 Preamble

This document houses all the codes I built on LilyPond since September 2024. Because I deal with contemporary notations in my compositional practice, I found myself creating codes and turning them into variables in order to repeatedly use them in my projects. I created a dedicated `.ly` file to store these codes for use, which quickly became very lengthy. I thought it would be useful to organize them into a document where I could easily consult and remind myself what they are and how to use them. This is that document.

Because I use LilyPond actively in my daily compositional and musical typesetting activities, this document is a work in progress.

0.2 README

This document and the codes contained herein are under the MIT License. So long as you include the copyright as well as the MIT License permission notices, please feel free to use my codes in your LilyPond files or modify them according to your specific need. Furthermore, crediting in the following manner is greatly appreciated:

```
% Original Code written by Yoshiaki Onishi  
% https://github.com/yoshiakionishi/lilypond-snippets
```

I make this document public because I wish to return something useful to the LilyPond community, but also to seek and implement any improvements other users may find in my codes. Please feel free to reach out to the email address shown on the title page of this document.

In the interest of making the codes found in this document available to as many people as possible, I have avoided using copyrighted musical examples. However, wherever appropriate, I have provided bibliographical sources. Furthermore, I acknowledge that, just as academic work in humanities goes, my ideas are built on those that are formulated by others; as such, whenever there is a direct source of inspiration for formulating a code, I provide sources.

In creating this document, I make no claim that my notational choices represent an absolute standard that everyone should adhere to. Once the basic principles of notation and typesetting are established (e.g., avoiding collisions, etc.), notation becomes a personal decision for each composer, shaped by careful study of preexisting scores and an evaluation of their musical contexts.

For example, in his book *The Bass Clarinet – A Personal History*, Harry Sparnaay lists nine

variants of noteheads for the slap tongue technique.¹ In my work, I created two subcategories of the slap tongue technique: one followed by a pitch and another followed by an air sound (which produces the slap tongue effect that sounds “empty”). To distinguish between the two, I decided to use encircled noteheads—both filled and hollow—and attribute them to each subcategory. Again, this is a method that I have found works for my music, but I would be reluctant to suggest that others should follow the same.²

Readers are encouraged to modify my codes in order to suit their desired techniques. This document serves as a record of how I arrived at certain notational choices, because learning LilyPond meant that I would also need to become familiar with Scheme, which proved to be somewhat challenging—even though I have used Common Lisp before owing to programming in OpenMusic—because I had to make many guessworks as I navigated various Scheme codes in other snippets available online. I have also gained familiarity in PostScript language as I continued to familiarize myself with LilyPond.³

0.3 Background

After [MakeMusic](#) announced that they would cease development of the music notation software program [Finale](#), which I had used for the past twenty-four years, I decided to explore a few other music notation programs to determine the best alternative. At the time of writing this document in late November 2024, a little under three months have passed since I started using [LilyPond](#) for my daily typesetting needs. I now open LilyPond more often than Finale and am committed to using it for the foreseeable future. LilyPond appears to me as the way forward both as a composer and a musical typesetter, as other proprietary notation programs, such as Dorico (which MakeMusic has claimed to be the leading program in the industry) and Sibelius, fall short of what I wish to accomplish.

While LilyPond is “just” a music notation software program that I happened to choose, it is, in a way, more than a toolkit for a composer. It appears that way to me, at least, because choosing an open-source platform with strong community support and engagement, rather than a proprietary program where desired functionality is subject to the priorities of a small group of salaried developers, reflects a critique of the capitalist/commercialist mindset that often pervades a composer’s life.

For example, before transitioning to LilyPond, I briefly explored Dorico. However, as of late September 2024, its functionality for displaying straight flags was very limited; the angle of the straight flags provided by the software was too steep. I consulted the online forum and discovered that another user had posted a question similar to mine. The chief developer of Dorico responded to that post, noting that implementing improvements to this feature was possible but “not currently a high priority.”⁴ In this tiered structure typical of capitalism, composers may find themselves with increasingly limited creative “freedom.”

MakeMusic has heavily advertised on social media platforms that Finale users should migrate to Dorico because it is the “next industry standard.” However, this advertising seems to discourage

1. Harry Sparnaay, *The Bass Clarinet: A Personal History* (Periferia Sheet Music, 2012), 66.

2. This particular notation becomes quickly problematic in terms of rhythmic notation when a bar is longer than a half note (e.g. 1/2, 2/4, 4/8...) For this reason, I tend to favor time signatures that avoids the use of a half note, such as 3/8 or 5/16.

3. See Appendix A for some resources I referred to for Scheme- and PostScript-related matters.

4. See: <https://forums.steinberg.net/t/straight-flags-angle/766503>.

thoughtful consideration of alternatives, leaving little room for reflection or exploration. I became increasingly disillusioned as I witnessed the coercion to invest in a program—however exciting it may appear—with no definite promise of its long-term security and stability.

Of course, it is not my intent to claim that all composers should abandon their proprietary programs of choice, particularly those they have invested money in and/or have been using for many years. It is, however, important to note that:

1. All proprietary programs are at the mercy of the executives who run the companies behind them. “Oh, [insert the name of a proprietary program] is operated by [insert the name of its company], and I just don’t see them closing the program down,” someone might say. Yet, it happened to Finale.
2. All notation programs, owing to the ways they operate, exert some degree of influence on the way composers compose. As early as the 1980s, Finale’s *Mass Mover*, *Note Mover*, and MIDI playback features were already influential in shaping the way composers worked on their music.⁵ On the one hand, these features may have helped composers save time. On the other hand, their ready availability may have invited overuse.
3. The lack or underdevelopment of certain functionalities may also push composers to work in certain ways rather than others. Finale benefitted from having the flexibility to implement graphical notation, but even then, many of my composer friends found it practical to use external graphical editing programs to further refine their scores. Even from my personal experience using Finale, I encountered situations where I had to devise creative alternatives to meet my notational goals.

These points implicitly highlight the benefits of learning an additional notation program, ideally an open-source one, alongside the program one primarily uses. LilyPond resonated with me most because of its text-based interface, which I have become increasingly familiar with through my involvement in computer programming. As other users have remarked, I have also found it to be very flexible and extensible. All the snippets I list in this document can be reused with relative ease, allowing me to save time in the long run when using specialized notations in my music. This was not necessarily the case when working on the music notation of extended techniques in Finale.

0.4 How This Document Is Structured

Each chapter of this document addresses a specific element of music notation, such as noteheads, stems, beams, and so on. Some chapters, however, cover topics specific to LilyPond coding, such as Markups and Spanners. Snippets that use more than one snippet covered in earlier chapters, thus simulating practical applications of these snippets, are covered in the chapter *Combinations*. Snippets that do not appear to belong to earlier chapters find their home in the chapter *Miscellanies*.

Each snippet entry includes a musical example, a description, the relevant grammar, the code required for the snippet to function, and, whenever necessary, a “Discussion” section.

0.5 LilyPond Version Used

The version of LilyPond used to create these snippets is 2.24.4.

5. For example, watch from 15:20 of <https://youtu.be/T1IRlg87Qks>.

0.6 Acknowledgements

I thank the supportive community of LilyPond users, whose exchanges on `lilypond-user` mailing list have inspired me greatly.

Even though I have not met him, I am grateful to Ben Lemon for his generosity in creating and sharing his LilyPond tutorial videos on YouTube. These videos were immensely helpful during the initial stages of learning LilyPond.

I also want to thank my friends who inspired me to start using LilyPond. It was Cole Ingraham who first introduced me to the program in 2016. My initial attempt at using it was not successful, but more recently, Santiago Beis composed and typeset his orchestral piece *Spletna* entirely in LilyPond, which compelled me to give it another try.

I extend my gratitude to my composition students at the University of Delaware School of Music, with whom I embarked on this journey of learning LilyPond. Even though they were not directly affected by Finale's discontinuation, they remained curious and enthusiastic about exploring this program. I hope that if the programs of their choice ever face a fate similar to Finale (though I sincerely hope they do not), they will be better equipped to adapt without the annoyance and arduous work often associated with transitioning to a new tool.

[Table of Contents](#)

Chapter 1

Articulations

1.1 Accent-Staccato



1.1.1 Description

While *accent-staccato* is not specific to contemporary music, in LilyPond, specifying accent and staccato via `->-` could cause the two articulation marks to be separated from each other. This happens because of the default setting for accents is to have them placed *outside* the staff line. For example, if you wrote `\stemUp g' '->-`, the following results:



This code implements a combination of the music glyphs `scripts.sforzato` and `scripts.staccato` as one entity.

1.1.2 Grammar

```
NOTE \accentStaccato
NOTE \accentStaccatoUp
NOTE \accentStaccatoDown
```

1.1.3 Code

```
1 \version "2.24.4"
2
3 #(append! default-script-alist
4           (list
5             `(\accentStaccatoUp
```

```

6         . (
7             (stencil . ,ly:text-interface::print)
8             (text . ,#{ \markup
9
10
11                 \center-align
12                 \combine \musicglyph "scripts.sforzato"
13                 \translate #'(0 . -0.75)
14                 \musicglyph "scripts.staccato" #})
15             ; any other properties
16             (toward-stem-shift-in-column . 1.0)
17             (outside-staff-priority . #t)
18             (padding . 0.5)
19             (avoid-slur . around)
20             (direction . ,UP))))
21
22 (list
23   `(accentStaccatoDown
24     . (
25         (stencil . ,ly:text-interface::print)
26         (text . ,#{ \markup \center-align
27                     \combine \musicglyph "scripts.staccato"
28                     \translate #'(0 . -0.75)
29                     \musicglyph "scripts.sforzato" #})
30         ; any other properties
31         (toward-stem-shift-in-column . 1.0)
32         (outside-staff-priority . #t)
33         (padding . 0.5)
34         (avoid-slur . around)
35         (direction . ,DOWN))))
36
37 accentStaccato = #(make-articulation 'accentStaccatoUp)
38 accentStaccatoUp = #(make-articulation 'accentStaccatoUp)
39 accentStaccatoDown = #(make-articulation 'accentStaccatoDown)
40
41
42 {
43   \override Staff.TimeSignature.stencil = ##f
44   \time 5/4
45
46   c'4\accentStaccato c'4 \accentStaccatoDown c''\accentStaccatoUp
47   \stopStaff
48   \override Staff.StaffSymbol.line-positions = #'(4 -4)
49   \startStaff
50
51   \stemUp g'' \tweak Y-offset #1.5 \accentStaccatoDown
52   \stemDown d' \tweak Y-offset #-0.5 \accentStaccatoUp
53 }

```

Table of Contents

1.2 Accent-Tenuto



1.2.1 Description

For the same rationale as explained in the [Accent-Staccato](#) entry, this code implements a combination of the music glyphs `scripts.sforzato` and `scripts.tenuto` as one entity.

1.2.2 Grammar

```
NOTE \accentTenuto
NOTE \accentTenutoUp
NOTE \accentTenutoDown
```

1.2.3 Code

```
1 \version "2.24.4"
2
3 #(append! default-script-alist
4   (list
5     `(accentTenutoUp
6       . (
7         (stencil . ,ly:text-interface::print)
8         (text . ,#{ \markup
9
10
11           \center-align
12           \combine \musicglyph "scripts.sforzato"
13           \translate #'(0 . -0.75)
14           \musicglyph "scripts.tenuto" #})
15       ; any other properties
16       (toward-stem-shift-in-column . 1.0)
17       (outside-staff-priority . #t)
18       (padding . 0.5)
19       (avoid-slur . around)
20       (direction . ,UP))))
21
22 (list
23   `(accentTenutoDown
24     . (
25       (stencil . ,ly:text-interface::print)
26       (text . ,#{ \markup \center-align
27         \combine \musicglyph "scripts.tenuto"
28         \translate #'(0 . -0.75)
```

```

29             \musicglyph "scripts.sforzato" #})
30         ; any other properties
31         (toward-stem-shift-in-column . 1.0)
32         (outside-staff-priority . #t)
33         (padding . 0.5)
34         (avoid-slur . around)
35         (direction . ,DOWN))))))
36
37 accentTenuto = #(make-articulation 'accentTenutoUp)
38 accentTenutoUp = #(make-articulation 'accentTenutoUp)
39 accentTenutoDown = #(make-articulation 'accentTenutoDown)
40
41
42 {
43   \override Staff.TimeSignature.stencil = ##f
44   \time 5/4
45
46   c'4\accentTenuto c'4 \accentTenutoDown c''\accentTenutoUp
47   \stopStaff
48   \override Staff.StaffSymbol.line-positions = #'(4 -4)
49   \startStaff
50
51   \stemUp g'' \tweak Y-offset #1.5 \accentTenutoDown
52   \stemDown d' \tweak Y-offset #-0.5 \accentTenutoUp
53 }

```

[Table of Contents](#)

1.3 Jeté (Ricochet)



1.3.1 Description

I use this notation to designate jeté/ricochet for string instruments, adding that the number of bounces are undetermined.¹

I apply this indication *above* the note regardless of how high or low the note is; however, in case of need, I have supplied the version to be used *under* the note.

1.3.2 Grammar

```
NOTE \jete
NOTE \jeteUp
NOTE \jeteDown
```

1.3.3 Code

```
1 \version "2.24.4"
2
3 jeteDesign =
4 \markup
5 \center-align
6 \combine \combine \combine
7 \override #'(filled . #t)
8 \path #0.1
9 #'((moveto -0.25 0.5)
10 (curveto 0.35 1.1 0.85 1.1 1.45 0.5)
11 (curveto 0.85 0.8 0.35 0.8 -0.25 0.5)
12 (closepath))
13 \draw-circle #0.2 #0 ##t
14 \translate #'(0.6 . 0) \draw-circle #0.2 #0 ##t
15 \translate #'(1.2 . 0)\draw-circle #0.2 #0 ##t
16 #(append! default-script-alist
17 (list
18 `(jetelistUp
19 . (
20 (stencil . ,ly:text-interface::print)
21 (text . ,#{ \markup \jeteDesign #})
22 ; any other properties
23 (toward-stem-shift-in-column . 1.0)
```

1. Concerning the technique of adding articulation designs to an internal alist, I was inspired by the following thread on lilypond-user mailing list: <https://lists.gnu.org/archive/html/lilypond-user/2015-04/msg00105.html>

```

24         (outside-staff-priority . #t)
25         (padding . 0.5)
26         (avoid-slur . around)
27         (direction . ,UP))))
28
29     (list
30       `(jetelistDown
31         . (
32           (stencil . ,ly:text-interface::print)
33           (text . ,#{ \markup \rotate #180 \jeteDesign #})
34           ; any other properties
35           (toward-stem-shift-in-column . 1.0)
36           (outside-staff-priority . #t)
37           (padding . 0.5)
38           (avoid-slur . around)
39           (direction . ,DOWN))))))
40
41     jete = #(make-articulation 'jetelistUp)
42     jeteUp = #(make-articulation 'jetelistUp)
43     jeteDown = #(make-articulation 'jetelistDown)
44
45
46     {c'4\jete c'4 \jeteDown c''\jeteUp }

```

[Table of Contents](#)

Chapter 2

Beams

2.1 Wiggle Beam (zig-zag shaped beam)



2.1.1 Description

Ordinary beams are replaced with zig-zag beams. A set of forward then backward beams are printed in the amount specified in the argument. I use this notation in such pieces as *jeux enjeux* (2022) for brass quintet, in order to designate somewhat uneven rhythmic figures, which are nonetheless to be played within the time frame indicated.

`\wiggleBeamOne` replaces an 8th-note beam.

`\wiggleBeamTwo` replaces a 16th-note beam.

`\wiggleBeamThree` replaces a 32nd-note beam.

`\wiggleBeam_markup` adds a zig-zag beam at will. This allows beaming of mixed note durations,

such as: 

`\wiggleBeamStemAdjust` allows the adjustment of a stem length, in the event the wiggle beam and the stem do not touch each other.

2.1.2 Grammar

`\wiggleBeamOne` `#vOffset` `#howMany` `#width` `#rotation`

`\wiggleBeamTwo` `#vOffset` `#howMany` `#width` `#rotation`

`\wiggleBeamThree` `#vOffset` `#howMany` `#width` `#rotation`

NB

- `hOffset` = (`\wiggleBeam_markup` only) the horizontal offset value originating from where the ordinary beam is placed.

- `vOffset` = the vertical offset value originating from where the ordinary beam is placed.
- `howMany` = how many "wiggles" to print. It only accepts integers.
- `width` = how wide each "wiggle" should appear. When in doubt, start with `#1`.
- `rotation` = a positive value would rotate the beam upward, and the negative value would rotate the beam downward.

NOTE `\wobbleBeam_markup #hOffset #vOffset #howMany #width #rotation`

NB

- `hOffset` = the horizontal offset value originating from where the ordinary beam is placed.
- `vOffset` = the vertical offset value originating from where an above-staff markup is placed. Thus, `#0` would place a wiggle beam above the staff line.
- `howMany` = how many "wiggles" to print. It only accepts integers.
- `width` = how wide each "wiggle" should appear. When in doubt, start with `#1`.
- `rotation` = a positive value would rotate the beam upward, and the negative value would rotate the beam downward.
- More than one `\wobbleBeam_markup` may be added in sequence, provided that for each instance all the arguments are defined.

`\wobbleBeamStemAdjust #fromMiddleLine #howFar` NOTE

NB

- `fromMiddleLine` = (`\wobbleBeamStemAdjust` only) = determines one end of the stem, `#0` being the middle line of an ordinary 5-line staff.
- `howFar` = (`\wobbleBeamStemAdjust` only) = computes how long the stem should be extended. A positive value would draw the stem upward, and a negative value would draw the stem downward. An integer corresponds to the distance between two staff lines of an ordinary 5-line staff.

2.1.3 Code

```

1 wobbleBeamOne =
2 #(define-music-function (vOffset howMany howWide howTilted)
3   (number? number? number? number?) #{
4     \once \override Voice.Beam.stencil = #ly:text-interface::print
5     \once \override Voice.Beam.text = \markup {
6       \translate #(cons 0 vOffset)
7       \postscript #(string-append
8         "newpath
9         1 setlinejoin
10        1 setlinecap

```

```

11         0.35 setlinewidth
12         0.13 0 moveto "
13         (number->string howMany)
14         " {" (number->string (* 0.6 howWide)) " "
15         (number->string (+ 0.5 howTilted)) " rlineto "
16         (number->string (* 0.6 howWide))
17         " -0.5 rlineto} repeat
18         stroke"
19     )
20
21 }
22 #})
23
24 wiggleBeamTwo =
25 #(define-music-function (vOffset howMany howWide howTilted )
26   (number? number? number? number?) #{
27     \once \override Voice.Beam.stencil = #ly:text-interface::print
28     \once \override Voice.Beam.text = \markup {
29       \translate #(cons 0 vOffset)
30       \postscript #(string-append
31         "newpath
32         1 setlinejoin
33         1 setlinecap
34         0.35 setlinewidth
35         0.13 0 moveto "
36         (number->string howMany)
37         " {" (number->string (* 0.6 howWide)) " "
38         (number->string (+ 0.5 howTilted)) " rlineto "
39         (number->string (* 0.6 howWide))
40         " -0.5 rlineto} repeat
41         stroke newpath
42         0.35 setlinewidth
43         1 setlinejoin
44         0.13 -0.75 moveto "
45         (number->string howMany)
46         " {" (number->string (* 0.6 howWide)) " "
47         (number->string (+ 0.5 howTilted)) " rlineto "
48         (number->string (* 0.6 howWide))
49         " -0.5 rlineto} repeat
50         stroke"
51     )
52   }
53   #})
54
55 wiggleBeamThree =
56 #(define-music-function (vOffset howMany howWide howTilted )
57   (number? number? number? number?)
58   #{

```

```

59     \once \override Voice.Beam.stencil = #ly:text-interface::print
60     \once \override Voice.Beam.text = \markup          {
61         \translate #(cons 0 vOffset)
62         \postscript #(string-append
63             "newpath
64             1 setlinejoin
65             1 setlinecap
66             0.35 setlinewidth
67             0.13 0 moveto "
68             (number->string howMany) " {"
69             (number->string (* 0.6 howWide)) " "
70             (number->string (+ 0.5 howTilted)) " rlineto "
71             (number->string (* 0.6 howWide))
72             " -0.5 rlineto} repeat
73             stroke
74             newpath
75             0.35 setlinewidth
76             1 setlinejoin
77             0.13 -0.75 moveto "
78             (number->string howMany) " {"
79             (number->string (* 0.6 howWide)) " "
80             (number->string (+ 0.5 howTilted)) " rlineto "
81             (number->string (* 0.6 howWide))
82             " -0.5 rlineto} repeat
83             stroke
84             newpath
85             0.35 setlinewidth
86             1 setlinejoin
87             0.13 -1.5 moveto "
88             (number->string howMany) " {"
89             (number->string (* 0.6 howWide)) " "
90             (number->string (+ 0.5 howTilted)) " rlineto "
91             (number->string (* 0.6 howWide))
92             " -0.5 rlineto} repeat
93             stroke"
94         )
95     }
96     #})
97
98 wiggleBeam_markup =
99 #(define-music-function (hOffset vOffset howMany howWide howTilted )
100   (number? number? number? number? number?)
101   #{
102     ^\markup          {
103       \translate #(cons hOffset vOffset)
104       \postscript #(string-append
105           "newpath
106           1 setlinejoin

```



```

107         1 setlinecap
108         0.35 setlinewidth
109         0.17 0 moveto "
110         (number->string howMany) " {"
111         (number->string (* 0.6 howWide)) " "
112         (number->string (+ 0.5 howTilted)) " rlineto "
113         (number->string (* 0.6 howWide))
114         " -0.5 rlineto} repeat
115         stroke"
116     )
117
118 }
119 #})
120
121 wiggleBeamStemAdjust =
122 #(define-music-function (fromMiddleLine howFar)
123   (number? number?)
124   #{
125     \once \override Stem.stencil = #ly:text-interface::print
126     \once \override Stem.text = \markup {
127       \postscript #(string-append
128         "newpath
129         0.12 setlinewidth
130         0 " (number->string fromMiddleLine) " moveto
131         0 " (number->string howFar) " rlineto
132         stroke"
133       )
134     }
135   #})
136
137 {
138   \wiggleBeamTwo #0 #9 #1.01 #0 c'16 c'
139   \wiggleBeamStemAdjust #-3 #3.4 c' c'
140   \wiggleBeamTwo #0 #5 #1.82 #0 g''
141   \wiggleBeamStemAdjust #2.5 #-3 g''
142   \wiggleBeamStemAdjust #2.5 #-3 g'' g''
143   \wiggleBeamTwo #-1 #9 #1.01 #-0.15 f''
144   \wiggleBeamStemAdjust #1.5 #-3.5 e''
145   \wiggleBeamStemAdjust #1 #-3.5 d''
146   \wiggleBeamStemAdjust #0.5 #-3.5 c''
147   \wiggleBeamOne #-3.5 #5 #1.4 #0.15 b'8
148   c''16 \wiggleBeam_markup #0 #-4.8 #2 #1.4 #0.15 d''
149   \wiggleBeamThree #-1.3 #19 #0.73 #0 g''32
150   \wiggleBeamStemAdjust #1.5 #-4 e''
151   \wiggleBeamStemAdjust #0.5 #-3 c'' g'' e''
152   \wiggleBeamStemAdjust #0.5 #-3 c''
153   \wiggleBeamStemAdjust #2.5 #-5 g'' e''
154   \bar "...

```

155 }

2.1.4 Discussion

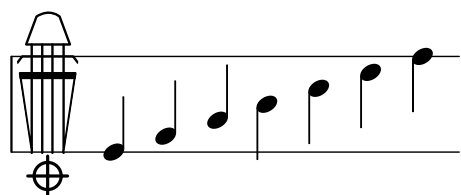
1. Admittedly, while the current setup allows great flexibility in making the wiggle beams appear, it is entirely possible that some of the parameters be automated.
2. When using many wiggle beams, it may be easier to make the score proportionally notated, in order to avoid the micromanagement of the parameters.

[Table of Contents](#)

Chapter 3

Clefs

3.1 String Position Clef



3.1.1 Description

String position clef to indicate bowing position. See Discussion for the associated command, `\normalClef`.

3.1.2 Grammar

`\stringPositionClef`

3.1.3 Code

```
1 stringPositionClefDesign = #(ly:make-stencil (list 'embedded-ps
2 "gsave
3 currentpoint translate
4 /fingboardpath
5 {
6 newpath
7
8 -0.55 7.5 moveto
9 0 -3 rlineto
10 1 -6.5 rlineto
11 -1 -1 rlineto
12 0 -3 rlineto
```

```
13 4.1 0 rlineto
14 0 3 rlineto
15 -1 1 rlineto
16 1 6.5 rlineto
17 0 3 rlineto
18 closepath
19
20 } def
21
22 fingboardpath clip
23 newpath
24 0.15 setlinewidth
25 0.5 4.75 moveto
26 0 -6.8 rlineto
27 -0.75 5 rlineto
28 3.5 0 rlineto
29 -0.75 -5 rlineto
30 0. 6.8 rlineto
31 stroke
32 0.35 setlinewidth
33 -0.4 2.75 moveto
34 3.75 0 rlineto
35 stroke
36
37 %inner two line
38 newpath
39 0.15 setlinewidth
40 1.16 4.75 moveto
41 0. -6.8 rlineto
42 1.8 4.75 moveto
43 0. -6.8 rlineto
44 stroke
45
46 %bridge
47 newpath
48 -0.4 3.6 moveto
49 0.3 0.4 rlineto
50 3.2 0 rlineto
51 0.3 -0.4 rlineto
52 stroke
53
54 %tailpiece
55 0.15 4.75 moveto
56 1 setlinecap
57 1 setlinejoin
58 2.75 0 rlineto
59 -0.65 1.75 rlineto
60 -0 -0 -0.6 0.55 -1.45 0 rcurveto
```

```

61 closepath
62 stroke
63
64 %mutesign
65 newpath
66 0.2 setlinewidth
67 1 setlinecap
68 1.5 -2.25 moveto
69 0 -2.5 rlineto
70 0.25 -3.5 moveto
71 2.5 0 rlineto
72 stroke
73 newpath
74 1.5 -3.5 0.85 0 360 arc
75 stroke
76 grestore")
77         (cons 0 3)
78         (cons 0 1))
79
80 stringPositionClefSize =
81 #(lambda (grob)
82   (let* ((sPCS (ly:grob-property grob 'font-size 0.0))
83         (mult (magstep sPCS)))
84     (ly:stencil-scale
85       stringPositionClef
86       mult mult)))
87
88 stringPositionClef = {
89   \override Staff.Clef.stencil = \stringPositionClefDesign
90 }
91
92 normalClef = {
93   \revert Staff.Clef.stencil
94 }
95
96 {
97   \override Staff.StaffSymbol.line-positions = #'(6 -6)
98   \override Staff.LedgerLineSpanner.stencil = ##f
99   \override Staff.TimeSignature.stencil = ##f
100   \override Staff.BarLine.stencil = ##f
101   \stringPositionClef c'4 e' g' b' d' f' a'
102 }

```

3.1.4 Discussion

1. With the current design, `c'` would place a note at the lower end of the fingerboard. `a''` would place a note on the same line as the bridge.

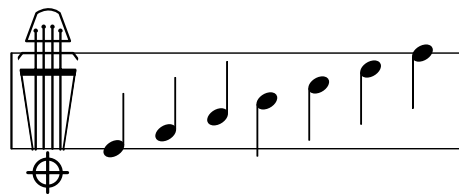
2. The current design comes with the mute sign. If the mute sign is not needed, remove the following portion of the code above:

```
64 %mutesign
65 newpath
66 0.2 setlinewidth
67 1 setlinecap
68 1.5 -2.25 moveto
69 0 -2.5 rlineto
70 0.25 -3.5 moveto
71 2.5 0 rlineto
72 stroke
73 newpath
74 1.5 -3.5 0.85 0 360 arc
75 stroke
```

3. Once `\stringPositionClef` is used, in order to revert back to the normal clef, `\normalClef` must be used.
4. See [Prescriptive Notation for String Instruments](#) for a possible use of this clef.

[Table of Contents](#)

3.2 String Position Clef (revised)



3.2.1 Description

This is a revised version of the [String Position Clef](#), where the fine-tuner pins are graphically represented, as well as the four strings are arranged tighter than the previous version.

3.2.2 Grammar

`\stringPositionClef`

3.2.3 Code

```

1
2 % revision june 24 2025
3
4 stringPositionClefDesignRev = #(ly:make-stencil (list 'embedded-ps
5                                                     "gsave
6   currentpoint translate
7   /fingboardpath
8   {
9
10  newpath
11  %0 1 .7 0 setcmykcolor
12  -0.5 7.5 moveto
13  0 -3 rlineto
14  1 -6.5 rlineto
15  -1 -1 rlineto
16  0 -3 rlineto
17  4.05 0 rlineto
18  0 3 rlineto
19  -1 1 rlineto
20  1 6.5 rlineto
21  0 3 rlineto
22  closepath
23
24  } def
25
26 % fingboardpath
27  fingboardpath clip

```

```
28
29 newpath
30 0.15 setlinewidth
31 0.75 5.25 moveto
32 0 -7.3 rlineto
33 -0.2 0 rmoveto
34 -0.75 5 rlineto
35 3.45 0 rlineto
36 -0.75 -5 rlineto
37 -0.2 0 rmoveto
38 0. 7.3 rlineto
39 stroke
40 0.35 setlinewidth
41 -0.4 2.75 moveto
42 3.75 0 rlineto
43 stroke
44
45 %inner two lines
46 newpath
47 0.15 setlinewidth
48 1.25 5.5 moveto
49 0. -7.5 rlineto
50 1.8 5.5 moveto
51 0. -7.5 rlineto
52 stroke
53
54
55 %finetuner pins
56 0.75 5.4 0.14 0 360 arc
57 fill
58 1.25 5.65 0.14 0 360 arc
59 fill
60 1.8 5.65 0.14 0 360 arc
61 fill
62 2.3 5.4 0.14 0 360 arc
63 fill
64
65
66
67 %bridge
68 newpath
69 -0.4 3.6 moveto
70 0.3 0.4 rlineto
71 3.2 0 rlineto
72 0.3 -0.4 rlineto
73 stroke
74
75 %tailpiece
```



```

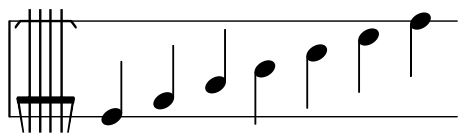
76 0.15 4.75 moveto
77 1 setlinecap
78 1 setlinejoin
79 2.75 0 rlineto
80 -0.65 1.75 rlineto
81 -0 -0 -0.6 0.55 -1.45 0 rcurveto
82 closepath
83 stroke
84
85 %mute sign, delete if not needed
86 newpath
87 0.2 setlinewidth
88 1 setlinecap
89 1.5 -2.25 moveto
90 0 -2.5 rlineto
91 0.25 -3.5 moveto
92 2.5 0 rlineto
93 stroke
94 newpath
95 1.5 -3.5 0.85 0 360 arc
96 stroke
97
98 grestore")
99
100                                     (cons 0 3)
101                                     (cons 0 1))
102
102 strPosClefSize =
103 #(lambda (grob)
104   (let* ((sPCS (ly:grob-property grob 'font-size 0.0))
105          (mult (magstep sPCS)))
106     (ly:stencil-scale
107      strPosClef
108      mult mult)))
109
110 stringPositionClefDesignRev = {
111   \override Staff.Clef.stencil = \stringPositionClefDesignRev
112 }
113
114 normalClef = {
115   \revert Staff.Clef.stencil
116 }
117
118 {
119   \override Staff.StaffSymbol.line-positions = #'(6 -6)
120   \override Staff.LedgerLineSpanner.stencil = ##f
121   \override Staff.TimeSignature.stencil = ##f
122   \override Staff.BarLine.stencil = ##f
123   \stringPositionClefDesignRev c'4 e' g' b' d'' f'' a''

```

124 }

[Table of Contents](#)

3.3 String Position Clef 2



3.3.1 Description

String position clef to indicate bowing position, but this one provides more space between bridge and the edge of the fingerboard, allowing the visual-timbre correspondence between *sul ponticello* and *sul tasto*.

3.3.2 Grammar

`\stringPositionClef_two`

3.3.3 Code

```

1  \version "2.24.4"
2
3  stringPositionClefDesign_two = #(ly:make-stencil (list 'embedded-ps
4                                                    "gsave
5  currentpoint translate
6  /fingboardpath
7  {
8
9  newpath
10 -0.45 4.75 moveto
11  0 -5 rlineto
12  0.5 -2.75 rlineto
13  2.9 0 rlineto
14  0.5 2.75 rlineto
15  0 5 rlineto
16  closepath
17 } def
18
19 fingboardpath clip
20 newpath
21 0.15 setlinewidth
22 0.5 8 moveto
23 0 -13.8 rlineto
24 -0.75 5 rlineto
25 3.5 0 rlineto
26 -0.75 -5 rlineto

```

```

27 0 11 rlineto
28 stroke
29 0.35 setlinewidth
30 -0.4 -1 moveto
31 3.75 0 rlineto
32 stroke
33
34 %inner two line
35 newpath
36 0.15 setlinewidth
37 1.16 4.75 moveto
38 0. -7.75 rlineto
39 1.8 4.75 moveto
40 0. -7.75 rlineto
41 stroke
42
43 %bridge
44 newpath
45 -0.4 3.6 moveto
46 0.3 0.4 rlineto
47 3.2 0 rlineto
48 0.3 -0.4 rlineto
49 stroke
50
51 grestore")
52         (cons 0 3)
53         (cons 0 1))
54
55 stringPositionClef_two = {
56   \override Staff.Clef.stencil = \stringPositionClefDesign_two
57 }
58
59 normalClef = {
60   \revert Staff.Clef.stencil
61 }
62
63 {
64   \override Staff.StaffSymbol.line-positions = #'(6 -6)
65   \override Staff.LedgerLineSpanner.stencil = ##f
66   \override Staff.TimeSignature.stencil = ##f
67   \override Staff.BarLine.stencil = ##f
68   \stringPositionClef_two c'4^\markup {
69     \translate #'(-3 . 2)
70     \musicglyph "space"
71   }
72   _\markup {
73     \translate #'(-3 . -3)
74     \musicglyph "space"

```

```

75    }
76    e' g' b' d'' f'' a''
77    }

```

3.3.4 Discussion

1. With the current design, `c'` would place a note at the lower end of the fingerboard. `a''` would place a note on the same line as the bridge.
2. Once `\stringPositionClef_two` is used, in order to revert back to the normal clef, `\normalClef` must be used.

[Table of Contents](#)

3.4 String Position Clef 3a



3.4.1 Description

In contrast to the two types of String Position Clefs introduced earlier, this clef helps facilitate the showing of the positions between the edge of the fingerboard and bridge, as well as between the bridge and the edge of the tailpiece.

3.4.2 Grammar

`\stringPositionClef_three`

3.4.3 Code

```

1
2 stringPositionClef_three_Design = #(ly:make-stencil (list 'embedded-ps
3                                     "gsave
4   currentpoint translate
5   /fingboardpath
6   {
7     newpath
8     0 1 .7 0 setcmykcolor
9     0.3 4.75 moveto
10    0 -4.5 rlineto
11    -0.2 -0.5 rlineto
12    0.5 -2.15 rlineto
13    -1 0 rlineto
14    0 -3 rlineto
15    3.75 0 rlineto
16    0 3 rlineto
17    -1 0 rlineto
18    0.45 2.15 rlineto
19    -0.15 0.5 rlineto
20    0 4.5 rlineto
21    closepath
22    %stroke
23    .1 .4 .5 .9 setcmykcolor
24  } def
25
26  %fingboardpath
27
```

```
28 fingboardpath clip
29 newpath
30 0.15 setlinewidth
31 0.8 3.5 moveto
32 0 -5.85 rlineto
33 -0.2 0 rmoveto
34 -0.25 1.3 rlineto
35 2.2 0 rlineto
36 -0.2 -1.3 rlineto
37 -0.2 0 rmoveto
38 0. 5.85 rlineto
39 stroke
40 0.35 setlinewidth
41 0.15 -1 moveto
42 3.2 0 rlineto
43 stroke
44
45 %inner two line
46 newpath
47 0.15 setlinewidth
48 1.25 3.6 moveto
49 0. -5.95 rlineto
50 1.7 3.6 moveto
51 0. -5.95 rlineto
52 stroke
53
54 0.8 3.5 0.14 0 360 arc
55 fill
56 2.1525 3.5 0.14 0 360 arc
57 fill
58 1.25 3.7 0.14 0 360 arc
59 fill
60 1.7 3.7 0.14 0 360 arc
61 fill
62
63
64
65 %bridge
66 newpath
67 0.25 0.6 moveto
68 0.3 0.4 rlineto
69 1.85 0 rlineto
70 0.3 -0.4 rlineto
71 stroke
72
73 %tailpiece
74 0.425 3 moveto
75 1 setlinecap
```

```

76 1 setlinejoin
77 2.15 0 rlineto
78 -0.35 1.25 rlineto
79 -0 -0 -0.65 0.75 -1.55 0 rcurveto
80 closepath
81 stroke
82
83 %%% mute sign; commentify if not needed %%%
84 newpath
85 0.2 setlinewidth
86 1 setlinecap
87 1.5 -2.25 moveto
88 0 -2.5 rlineto
89 0.25 -3.5 moveto
90 2.5 0 rlineto
91 stroke
92 newpath
93 1.5 -3.5 0.85 0 360 arc
94 stroke
95 %%% end of mute sign for commenting/uncommenting %%%
96
97 grestore
98
99 ")
100                                     (cons 0 3)
101                                     (cons 0 1))
102
103 stringPositionClefSize =
104 #(lambda (grob)
105   (let* ((bCS (ly:grob-property grob 'font-size 0.0))
106          (mult (magstep bCS)))
107     (ly:stencil-scale
108       stringPositionClef
109       mult mult)))
110
111 stringPositionClef_three = {
112   \override Staff.Clef.stencil = \stringPositionClef_three_Design
113 }
114
115 {
116   \override Staff.StaffSymbol.line-positions = #'(4 0 -4)
117   \stringPositionClef_three
118   c'4 e' g' b' d' f' a'
119 }

```


3.4.4 Discussion

1. With the current design, **e'** places a note at the lower end of the fingerboard. **b'** places a note at the bridge line, and **f''** places a note on the line indicating the edge of the tailpiece.
2. The current design comes with the mute sign. If the mute sign is not needed, remove the following portion of the code above:

```

64  %%% mute sign; commentify if not needed %%%
65  newpath
66  0.2 setlinewidth
67  1 setlinecap
68  1.5 -2.25 moveto
69  0 -2.5 rlineto
70  0.25 -3.5 moveto
71  2.5 0 rlineto
72  stroke
73  newpath
74  1.5 -3.5 0.85 0 360 arc
75  stroke
76  %%% end of mute sign for commenting/uncommenting %%%

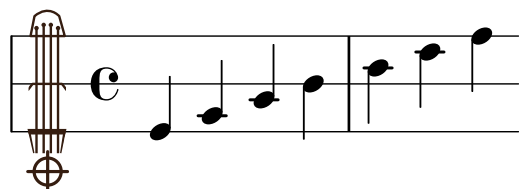
```

3. If you do not wish the ledger lines to appear within the staff line, consider using:

```
\override Staff.NoteHead.no-ledgers = ##t.
```

[Table of Contents](#)

3.5 String Position Clef 3b (Longer Span)



3.5.1 Description

The design of this clef is similar to [String Position Clef 3a](#); however, here the distance among the edge of the fingerboard, bridge, and the edge of the tailpiece is set wider.

3.5.2 Grammar

`\stringPositionClef_three`

3.5.3 Code

```

1
2 stringPositionClef_three_longer_Design = #(ly:make-stencil (list 'embedded-ps
3                                     "gsave
4   currentpoint translate
5   /fingboardpath
6   {
7     newpath
8     0 1 .7 0 setcmykcolor
9     0.3 5.75 moveto
10    0 -6.5 rlineto
11    -0.2 -0.5 rlineto
12    0.5 -2.15 rlineto
13    -0.5 0 rlineto
14    0 -2.5 rlineto
15    2.75 0 rlineto
16    0 2.5 rlineto
17    -0.5 0 rlineto
18    0.45 2.15 rlineto
19    -0.15 0.5 rlineto
20    0 6.5 rlineto
21    closepath
22    %stroke
23    .1 .4 .5 .9 setcmykcolor
24  } def
25
26  %fingboardpath
27
```

```
28 fingboardpath clip
29 newpath
30 0.15 setlinewidth
31 0.8 4.5 moveto
32 0 -7.85 rlineto
33 -0.2 0 rmoveto
34 -0.25 1.3 rlineto
35 2.2 0 rlineto
36 -0.2 -1.3 rlineto
37 -0.2 0 rmoveto
38 0. 7.85 rlineto
39 stroke
40 0.35 setlinewidth
41 0.15 -2 moveto
42 3.2 0 rlineto
43 stroke
44
45 %inner two line
46 newpath
47 0.15 setlinewidth
48 1.25 4.6 moveto
49 0. -7.95 rlineto
50 1.7 4.6 moveto
51 0. -7.95 rlineto
52 stroke
53
54 0.8 4.5 0.14 0 360 arc
55 fill
56 2.1525 4.5 0.14 0 360 arc
57 fill
58 1.25 4.7 0.14 0 360 arc
59 fill
60 1.7 4.7 0.14 0 360 arc
61 fill
62
63
64
65 %bridge
66 newpath
67 0.25 0.6 moveto
68 0.3 0.4 rlineto
69 1.85 0 rlineto
70 0.3 -0.4 rlineto
71 stroke
72
73 %tailpiece
74 0.425 4 moveto
75 1 setlinecap
```

```

76 1 setlinejoin
77 2.15 0 rlineto
78 -0.35 1.25 rlineto
79 -0 -0 -0.65 0.75 -1.55 0 rcurveto
80 closepath
81 stroke
82
83 %%% mute sign; commentify if not needed %%%
84 newpath
85 0.2 setlinewidth
86 1 setlinecap
87 1.5 -3.25 moveto
88 0 -2.5 rlineto
89 0.25 -4.5 moveto
90 2.5 0 rlineto
91 stroke
92 newpath
93 1.5 -4.5 0.85 0 360 arc
94 stroke
95 %%% end of mute sign for commenting/uncommenting %%%
96
97 grestore
98
99 ")
100                                     (cons 0 3)
101                                     (cons 0 1))
102
103 stringPositionClefSize =
104 #(lambda (grob)
105   (let* ((bCS (ly:grob-property grob 'font-size 0.0))
106         (mult (magstep bCS)))
107     (ly:stencil-scale
108       stringPositionClef
109       mult mult)))
110
111 stringPositionClef_three_longer = {
112   \override Staff.Clef.stencil = \stringPositionClef_three_longer_Design
113 }
114
115 {
116   \override Staff.StaffSymbol.line-positions = #'(6 0 -6)
117   \stringPositionClef_three_longer
118   c'4 e' g' b' d'' f'' a''
119 }

```

3.5.4 Discussion

1. With the current design, **c'** places a note at the lower end of the fingerboard. **b'** places a note at the bridge line, and **a'** places a note on the line indicating the edge of the tailpiece.
2. The current design comes with the mute sign. If the mute sign is not needed, remove the following portion of the code above:

```

64  %%% mute sign; commentify if not needed %%%
65  newpath
66  0.2 setlinewidth
67  1 setlinecap
68  1.5 -3.25 moveto
69  0 -2.5 rlineto
70  0.25 -4.5 moveto
71  2.5 0 rlineto
72  stroke
73  newpath
74  1.5 -4.5 0.85 0 360 arc
75  stroke
76  %%% end of mute sign for commenting/uncommenting %%%

```

3. If you do not wish the ledger lines to appear within the staff line, consider using:

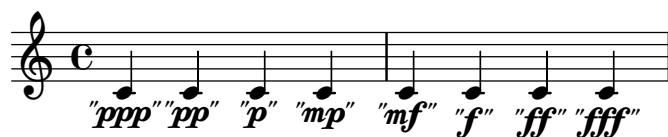
```
\override Staff.NoteHead.no-ledgers = ##t.
```

[Table of Contents](#)

Chapter 4

Dynamics

4.1 Dynamics in Quotation Marks



4.1.1 Description

Updated: November 12, 2025: hairpins will stop closer to the dynamics in quotation marks.

Dynamics in quotation marks, also known as *effort dynamics*, indicate those with which certain techniques must be carried on, understanding that the perceived dynamics will be quieter than what are indicated. Examples abound in scores by Helmut Lachenmann and others for such techniques as air sound, bowing directly on the bridge, etc..

4.1.2 Grammar

```
NOTE \qppp
NOTE \qpp
NOTE \qp
NOTE \qmp
NOTE \qmf
NOTE \qf
NOTE \qff
NOTE \qfff
```

4.1.3 Code

```
1 \version "2.24.4"
2
3 qmp = #(make-dynamic-script
4         (markup
```

```

5      #:pad-x -1
6      ( #:combine
7        #:combine
8        #:translate '(-0.85 . -0.1)
9        #:normal-text (#:italic #:fontsize 0.75 "\"")
10       #:dynamic "mp"
11       #:translate '(3.25 . -0.1)
12       #:normal-text (#:italic #:fontsize 0.75 "\"")))))
13 qp = #(make-dynamic-script
14   (markup
15     #:pad-x -1
16     (
17       #:combine
18       #:combine
19       #:translate '(-0.95 . -0.1)
20       #:normal-text (#:italic #:fontsize 0.75 "\"")
21       #:dynamic "p"
22       #:translate '(1.35 . -0.1)
23       #:normal-text (#:italic #:fontsize 0.75 "\"")))))
24 qpp = #(make-dynamic-script
25   (markup
26     #:pad-x -1
27     ( #:combine
28       #:combine
29       #:translate '(-0.95 . -0.1)
30       #:normal-text (#:italic #:fontsize 0.75 "\"")
31       #:dynamic "pp"
32       #:translate '(2.75 . -0.1)
33       #:normal-text (#:italic #:fontsize 0.75 "\"")))))
34 qppp = #(make-dynamic-script
35   (markup
36     #:pad-x -1
37     ( #:combine
38       #:combine
39       #:translate '(-0.95 . -0.1)
40       #:normal-text (#:italic #:fontsize 0.75 "\"")
41       #:dynamic "ppp"
42       #:translate '(4.25 . -0.1)
43       #:normal-text (#:italic #:fontsize 0.75 "\"")))))
44 qmf = #(make-dynamic-script
45   (markup
46     #:pad-x -1
47     ( #:combine
48       #:combine
49       #:translate '(-0.85 . 0)
50       #:normal-text (#:italic #:fontsize 0.75 "\"")
51       #:dynamic "mf"
52       #:translate '(3.25 . 0)

```

```

53         #:normal-text ( #:italic #:fontsize 0.75 "\"")))))
54 qf = #(make-dynamic-script
55       (markup
56         #:pad-x -1
57         ( #:combine
58           #:combine
59           #:translate '(-0.75 . 0)
60           #:normal-text ( #:italic #:fontsize 0.75 "\"")
61           #:dynamic "f"
62           #:translate '(1.65 . 0)
63           #:normal-text ( #:italic #:fontsize 0.75 "\"")))))
64 qff = #(make-dynamic-script
65       (markup
66         #:pad-x -1
67         ( #:combine
68           #:combine
69           #:translate '(-0.75 . 0)
70           #:normal-text ( #:italic #:fontsize 0.75 "\"")
71           #:dynamic "ff"
72           #:translate '(2.75 . 0)
73           #:normal-text ( #:italic #:fontsize 0.75 "\"")))))
74 qfff = #(make-dynamic-script
75       (markup
76         #:pad-x -1
77         ( #:combine
78           #:combine
79           #:translate '(-0.75 . 0)
80           #:normal-text ( #:italic #:fontsize 0.75 "\"")
81           #:dynamic "fff"
82           #:translate '(3.85 . 0)
83           #:normal-text ( #:italic #:fontsize 0.75 "\"")))))
84
85 {
86
87   c'4\qppp
88   c'4\qpp
89   c'4\qp
90   c'4\qmp
91
92   c'4\qmf
93   c'4\qf
94   c'4\qff
95   c'4\qfff
96
97 }
98
99 \layout {
100   \context {

```



```
101     \Score      proportionalNotationDuration = #(ly:make-moment 1/9)
102   }
103 }
```

4.1.4 Discussion

In scores by Lachenmann, in concordance with German quotation marks (*Anführungszeichen*), the opening quotation mark points left, and placed on the bottom line, and the closing quotation mark points right and sits at the top of the last character. It would be possible to achieve this by adjusting the parameters in the Scheme code.¹

Table of Contents

1. See: <https://lilypond.org/doc/v2.24/Documentation/extending/markup-construction-in-scheme>

4.2 Enhanced Constante Hairpins I (Under The Staff)

Four musical staves illustrating enhanced constante hairpins. Each staff shows a sequence of notes (G4, A4, B4, C5, B4, A4, G4) on a treble clef staff. The hairpins are placed below the staff, indicating dynamic changes over time. The first staff shows a hairpin starting at *mf* and tapering to *p*. The second staff shows a hairpin starting at *(p)* and remaining constant. The third staff shows a hairpin starting at *(p)* and flaring to *"f"*. The fourth staff shows a hairpin starting at *f* and flaring to *ff*. The fifth staff shows a hairpin starting at *f* and flaring to *ff*.

4.2.1 Description

NB: Slight revision (December 23, 2025) has been carried out to make sure the hairpin terminates at the end of the staff line when it breaks out to the next system.

A constante hairpin, a horizontal line with a short hook at the end, indicates that a dynamic remains unchanged throughout the notes to which it is applied. There are many contemporary scores that utilize this type of hairpins, e.g. works by Brian Ferneyhough. In LilyPond, it is possible to specify constante hairpins by writing an override function:

```
\override Hairpin.stencil = #constante-hairpin
```

However, similar to the issues raised in the [Flared Hairpins](#), LilyPond's default constante hairpins could use some enhancements, particularly in regards to accommodating three separate components when they break into new systems: 1. beginning; 2. intermediate; and; 3. ending. Furthermore, when the constante hairpins break into a new system, it would sometimes be helpful to have the dynamic in parentheses to remind what the dynamic had been in the previous system, which could be taking place before the page turn.

4.2.2 Grammar

NOTE `\constante` `\DYNAMIC` NOTE(S) `\DYNAMIC` OR `\!`

This `\constante` accepts three optional arguments:

1. Vertical offset of the `constante` hairpin, indicated by a direction `#UP`, `#CENTER`, or `#DOWN`. Without specifying this argument, the default is `#CENTER`. You may use `#UP` when it is followed by a decrescendo hairpin, while you may use `#DOWN` when it is followed by a crescendo hairpin.
2. Whether or not to use cautionary dynamics in subsequent systems when the hairpin breaks into new systems, indicated by `##f` or `##t`. The default is `##f`.
3. Extra vertical adjustment is possible with the third argument by specifying a number. The default is `#0`.

4.2.3 Code

```

1  \version "2.24.4"
2
3
4  qf = #(make-dynamic-script
5      (markup
6          #:pad-x -1
7          ( #:combine
8              #:combine
9              #:translate '(-0.75 . 0)
10             #:normal-text (:#italic #:fontsize 0.75 "\"")
11             #:dynamic "f"
12             #:translate '(1.65 . 0)
13             #:normal-text (:#italic #:fontsize 0.75 "\"")))))
14
15  qmp = #(make-dynamic-script
16      (markup
17          #:pad-x -1
18          ( #:combine
19              #:combine
20              #:translate '(-0.85 . -0.1)
21              #:normal-text (:#italic #:fontsize 0.75 "\"")
22              #:dynamic "mp"
23              #:translate '(3.25 . -0.1)
24              #:normal-text (:#italic #:fontsize 0.75 "\"")))))
25
26
27  constante =
28  #(define-music-function (vOff cautionary moreOff dyn)
29      ( (ly:dir? CENTER) (boolean? #f) (number? 0) ly:event? )
30      (define-public
31          ((constante-hairpin-revised coords mirrored?) grob)
32          "This scheme code was originally the scheme code 'elbowed-hairpin
33 taken from output-lib.scm, and was revised to suit the need."
34          (define (scale-coords coords-list x y)
35              (map

```

```

36      (lambda (coord) (cons (* x (car coord)) (* y (cdr coord))))
37      coords-list))
38
39      (define (hairpin::print-part points decresc? me)
40        (let ((stil (make-connected-line points me)))
41          (ly:stencil-scale stil 1 1)))
42
43      ;; outer let to trigger suicide
44      (let ((sten (ly:hairpin::print grob)))
45        (if (grob::is-live? grob)
46            (let*
47              (
48                ;I've left decresc? function but removed all the if clause
49                ;associated with it, to simplify the code a little bit
50                (verticalOffset vOff)
51                (useCautionary? cautionary)
52                (moreOffset moreOff)
53                (hairpinHeight (ly:grob-property grob 'height))
54                (decresc? (eqv? (ly:grob-property grob 'grow-direction) LEFT))
55                (toBarline? (ly:grob-property grob 'to-barline))
56                ; (circled-tip? (ly:grob-property grob 'circled-tip))
57                (hairpinTypeA?
58                  (and (eqv? (end-broken-spanner? grob )           #f)
59                       (eqv? (first-broken-spanner? grob )         #f)
60                       (eqv? (middle-broken-spanner? grob )        #f)
61                       (eqv? (not-first-broken-spanner? grob )      #f)
62                       (eqv? (not-last-broken-spanner? grob )       #f)
63                       (eqv? (unbroken-or-first-broken-spanner? grob ) #t)
64                       (eqv? (unbroken-or-last-broken-spanner? grob ) #t)
65                       (eqv? (unbroken-spanner? grob )              #t)
66                      )) ;Hairpins that end within one line without
67                ;reaching to the end of the system
68                (hairpinTypeB?
69                  (or (and (eqv? (end-broken-spanner? grob )           #t)
70                             (eqv? (first-broken-spanner? grob )         #t)
71                             (eqv? (middle-broken-spanner? grob )        #f)
72                             (eqv? (not-first-broken-spanner? grob )      #f)
73                             (eqv? (not-last-broken-spanner? grob )       #f)
74                             (eqv? (unbroken-or-first-broken-spanner? grob ) #t)
75                             (eqv? (unbroken-or-last-broken-spanner? grob ) #t)
76                             (eqv? (unbroken-spanner? grob )              #f)
77                             )
78                      (and (eqv? (end-broken-spanner? grob )           #f)
79                           (eqv? (first-broken-spanner? grob )         #t)
80                           (eqv? (middle-broken-spanner? grob )        #f)
81                           (eqv? (not-first-broken-spanner? grob )      #f)
82                           (eqv? (not-last-broken-spanner? grob )       #t)
83                           (eqv? (unbroken-or-first-broken-spanner? grob ) #t)

```

```

84          (eqv? (unbroken-or-last-broken-spanner? grob )    #f)
85          (eqv? (unbroken-spanner? grob )                  #f)
86          )
87      )) ;Hairpins that breaks into next line
88  (hairpinTypeC?
89      (and (eqv? (end-broken-spanner? grob )                #f)
90            (eqv? (first-broken-spanner? grob )              #f)
91            (eqv? (middle-broken-spanner? grob )             #t)
92            (eqv? (not-first-broken-spanner? grob )          #t)
93            (eqv? (not-last-broken-spanner? grob )           #t)
94            (eqv? (unbroken-or-first-broken-spanner? grob ) #f)
95            (eqv? (unbroken-or-last-broken-spanner? grob ) #f)
96            (eqv? (unbroken-spanner? grob )                  #f)
97            )) ;Midway-through Hairpins
98  (hairpinTypeD?
99      (and (eqv? (end-broken-spanner? grob )                #t)
100           (eqv? (first-broken-spanner? grob )              #f)
101           (eqv? (middle-broken-spanner? grob )             #f)
102           (eqv? (not-first-broken-spanner? grob )          #t)
103           (eqv? (not-last-broken-spanner? grob )           #f)
104           (eqv? (unbroken-or-first-broken-spanner? grob ) #f)
105           (eqv? (unbroken-or-last-broken-spanner? grob ) #t)
106           (eqv? (unbroken-spanner? grob )                  #f)
107           )) ;multi-system Hairpin that terminates
108  (xex
109      (cons (car (ly:stencil-extent sten X))
110            (- (cdr (ly:stencil-extent sten X)) 1.25))
111      )
112  (lenx (interval-length xex))
113  (yex (ly:stencil-extent sten Y))
114  (leny (interval-length yex))
115  (xtrans (+ (car xex) 0))
116  (ytrans (car yex))
117  (uplist (scale-coords coords lenx (/ leny 2)))
118
119  (crescStencil
120
121      (if toBarline?
122          ;I added here to make it default to stretch
123          ;the hairpin immediately to the dynamics
124
125          (ly:stencil-translate
126              (hairpin::print-part
127                  (scale-coords coords
128                      (+ lenx 2.35) (/ leny 2))
129                  decresc? grob)
130
131              '(-0.5 . 0))

```

```

132         ;if not to barline
133         (ly:stencil-translate
134         (hairpin::print-part
135         (scale-coords coords
136           (+ lenx 2.35) (/ leny 2))
137         decresc? grob)
138
139         '(-0.5 . 0))
140
141     )
142 )
143 (crescStencilBrokenBeginning
144
145   (if toBarline?
146     ;I added here to make it default to stretch
147     ;the hairpin immediately to the dynamics
148
149     (hairpin::print-part (scale-coords
150       '((0 . 0) (1 . 0))
151       (+ lenx 1) (/ leny 2))
152     decresc? grob)
153
154     ;if not to barline
155     (ly:stencil-translate
156     (hairpin::print-part
157     (scale-coords '((0 . 0) (1 . 0))
158       (+ lenx 1.5) (/ leny 2))
159     decresc? grob)
160
161     '(-0.5 . 0))
162
163   )
164 )
165 (crescStencilBrokenContinuation
166   (ly:stencil-combine-at-edge
167   ;if the cautionary accidentals are turned on
168   (if cautionary
169     (if (list? (ly:music-property dyn 'text))
170       ;this means, when the custom-made dynamics
171       ;(e.g. \qmp) are used
172       (grob-interpret-markup
173       grob
174       (markup
175         #:hspace 2
176         #:whiteout
177         #:center-align
178         (
179           #:concat
180           ( #:normal-text

```

```

180             (:translate '(0 . -0.25)
181                 #:italic (:fontsize -0.5 "( "
182                             ))
183             #:pad-x
184             0.35
185             #:dynamic (ly:music-property dyn 'text)
186             #:normal-text
187             (:translate '(0.5 . -0.25)
188                 #:italic (:fontsize -0.5 ")"))))
189         )
190     )
191 )
192 ;when in-house dynamics (e.g. \p) are used
193 (grob-interpret-markup
194   grob
195   (markup
196     #:hspace 2
197     #:center-align
198     #:whiteout
199     #:concat
200     (:normal-text
201       (:translate '(0 . -0.25)
202           #:italic (:fontsize 0.5 "("
203                       ))
204       (:translate '(0 . -0.25)
205           #:italic (:fontsize -10 " "
206                       ))
207       #:pad-x
208       0
209       #:dynamic (ly:music-property dyn 'text)
210       #:normal-text
211       (:translate '(0 . -0.25)
212           #:italic (:fontsize 0.5 ")"))))
213     )
214   ))
215   empty-stencil)
216 1
217 1
218 (hairpin::print-part (scale-coords
219                       '((0.01 . 0.15) (1 . 0.15))
220                       (+ lenx 1) (/ leny 2))
221                       decresc? grob)
222 -1.5
223 )
224 )
225 (crescStencilBrokenEnding
226   ;if the flare end is applied
227   ;on the dynamic at the beginning

```

```

228      ;of the system, meaning lenx is super low
229
230      (cond
231        ((< lenx 1)
232         (ly:stencil-combine-at-edge
233          ;if the cautionary accidentals are turned on
234          (if cautionary
235            (if (list? (ly:music-property dyn 'text))
236              ;this means, when the custom-made dynamics
237              ;(e.g. \qmp) are used
238              (grob-interpret-markup
239               grob
240               (markup
241                #:hspace -1
242                #:whiteout
243                #:center-align
244                #:concat
245                (:#:normal-text
246                 (:#:translate '(0 . -0.25)
247                  #:italic (:#:fontsize -0.5 "( "
248                           ))
249
250                #:pad-x
251                0.25
252                #:dynamic (ly:music-property dyn 'text)
253                #:normal-text
254                (:#:translate '(0.5 . -0.25)
255                 #:italic (:#:fontsize -0.5 ")"))))
256              )
257              )
258          ;when in-house dynamics (e.g. \p) are used
259          (grob-interpret-markup
260           grob
261           (markup
262            #:hspace -1
263            #:whiteout
264            #:center-align
265            (
266             #:concat
267             (:#:normal-text
268              (:#:translate '(0 . -0.25)
269               #:italic (:#:fontsize 0.5 "( "
270                        ))
271
272             (:#:translate '(0 . -0.25)
273              #:italic (:#:fontsize -10 " "
274                        ))
275
276             #:pad-x 0
277             #:dynamic (ly:music-property dyn 'text)

```



```

276             #:normal-text
277             (#:translate '(0 . -0.25)
278                     #:italic (#:fontsize 0.5 "))))
279         ))
280     ))
281     empty-stencil
282 )
283 1
284 1
285 (hairpin::print-part
286   (scale-coords coords
287     (+ lenx 1.85) (/ leny 2) )
288   decresc? grob)
289 ;vertical alignment (to be refined possibly...?)
290 -1.75
291 )
292 )
293 ((>= lenx 1)
294  ;if the normal ending flare crescendo
295  ;is applied...
296  (ly:stencil-combine-at-edge
297   ;if the cautionary accidentals are turned on
298   (if cautionary
299     (if (list? (ly:music-property dyn 'text))
300       ;this means, when the custom-made dynamics
301       ;(e.g. \qmp) are used
302       (grob-interpret-markup
303         grob
304         (markup
305           #:hspace 2
306           #:whiteout
307           #:center-align
308           #:concat
309           (#:normal-text
310             (#:translate '(0 . -0.25)
311                     #:italic (#:fontsize -0.5 "( "
312                               ))
313             #:pad-x 0.25
314             #:dynamic (ly:music-property dyn 'text)
315             #:normal-text
316             (#:translate '(0.5 . -0.25)
317                     #:italic (#:fontsize -0.5 "))))
318         )
319       )
320   ;if the in-house dynamics are used
321
322   (grob-interpret-markup
323     grob

```

```

324         (markup
325           #:hspace 2
326           #:whiteout
327           #:center-align
328           #:concat
329           (#:normal-text
330             (#:translate '(0 . -0.25)
331               #:italic (#:fontsize 0.5 "("
332                 ))
333             (#:translate '(0 . -0.25)
334               #:italic (#:fontsize -10 " "
335                 ))
336             #:pad-x
337             0
338             #:dynamic (ly:music-property dyn 'text)
339             #:normal-text
340             (#:translate '(0 . -0.25)
341               #:italic (#:fontsize 0.5 ")"))))
342         )
343       ))
344     empty-stencil
345   )
346   1
347   1
348   (hairpin::print-part
349     (scale-coords coords
350       (+ lenx 1.85) (/ leny 2) )
351     decresc? grob)
352   ;vertical alignment (to be refined possibly...?)
353   -1.75
354   )
355   )
356   )
357   )
358
359
360   (cresc
361     (cond (hairpinTypeA? crescStencil)
362           (hairpinTypeB? crescStencilBrokenBeginning)
363           (hairpinTypeC? crescStencilBrokenContinuation)
364           (hairpinTypeD? crescStencilBrokenEnding)
365           )
366   )
367
368   (stil
369     (ly:stencil-aligned-to
370       (ly:stencil-translate
371         cresc

```

```

372         (cons xtrans ytrans))
373     Y CENTER))
374     (stil-y-extent (ly:stencil-extent stil Y)))
375 ; FOR DEBUG ONLY
376 ; (display (list? (ly:music-property dyn 'text)))
377 ; (newline)
378 ; (display dyn)
379 ; (newline)
380 ; (newline)
381 ; (display "Is this type A Hairpin? " )
382 ; (display hairpinTypeA?)
383 ; (newline)
384 ; (display "Is this type B Hairpin? " )
385 ; (display hairpinTypeB?)
386 ; (newline)
387 ; (display "Is this type C Hairpin? " )
388 ; (display hairpinTypeC?)
389 ; (newline)
390 ; (display "Is this type D Hairpin? " )
391 ; (display hairpinTypeD?)
392 ;
393 ; (newline)
394 ; (display "Is it decrescendo? " )
395 ; (display decresc?)
396 ; (newline)
397 ; (display "end-broken-spanner??? " )
398 ; (display (end-broken-spanner? grob ))
399 ; (newline)
400 ; (display "(first-broken-spanner?) Is spanner broken
401 ;         and the first of its broken siblings? " )
402 ; (display (first-broken-spanner? grob ) )
403 ; (newline)
404 ; (display "middle-broken-spanner??? " )
405 ; (display (middle-broken-spanner? grob ) )
406 ; (newline)
407 ;
408 ; (display "Is spanner broken and not the first of
409 ;         its broken siblings? " )
410 ; (display (not-first-broken-spanner? grob ) )
411 ; (newline)
412 ;
413 ; (display "Is spanner broken and not the last of
414 ;         its broken siblings? " )
415 ; (display (not-last-broken-spanner? grob ) )
416 ; (newline)
417 ;
418 ; (display "unbroken-or-first-broken-spanner??? " )
419 ; (display (unbroken-or-first-broken-spanner? grob ) )

```

```

420         ; (newline)
421         ;
422         ; (display "unbroken-or-first-broken-spanner???" )
423         ; (display (unbroken-or-last-broken-spanner? grob ) )
424         ; (newline)
425         ; (display "unbroken spanner?? " )
426         ; (display (unbroken-spanner? grob ) )
427         ; (newline)
428         ; (newline)
429         ; (display "distance of xex " )
430         ; (display lenx )
431         ; (newline)
432         ; (newline)
433         (ly:make-stencil (ly:stencil-expr stil) xex stil-y-extent))
434     ;; return empty, if no Hairpin.stencil present.
435     '()))
436
437     #{
438     #dyn
439     \tweak Hairpin.stencil
440     #(lambda (grob)
441         ;depending on the hairpin opening width,
442         ;you can adjust the value here vvv
443         (constante-hairpin-revised '((0 . 0) (1.0 . 0) (1.0 . 0.9)) #f)
444         )
445     % This calculates the tweak value for
446     % the value of the constante V-offset
447     \tweak Y-offset #(lambda (grob)
448         (+ (* vOff (ly:grob-property grob 'height))
449            (* (ly:grob-property grob 'height) -0.5 vOff )
450            moreOff
451            )
452         )
453
454     \<
455     #}
456     )
457 \layout {
458     indent = #0
459     line-width = #100
460     ragged-last = ##f
461 }
462 {
463
464     \override Hairpin.height = #0.5
465     \override Hairpin.to-barline = ##f
466     % \override Hairpin.stencil = #constante-hairpin
467     \override Hairpin.after-line-breaking = ##t

```

```

468 c'2 \constante #UP \mf c'2 c'2 \> c'2
469 c'2 \constante #DOWN ##t #0.3 \p c'2
470 \break c'4 4 4 4 c'4 4 4 4 \break
471 c''2 \< c'2 \! \constante #DOWN ##f \qf \break
472 c'2 c'2\< c'2 c'2 c'2
473 \revert Hairpin.to-barline
474 \afterGrace 31/32
475 { c'2 \constante #CENTER ##f \ff} s32\! \break
476 c'2\f c'2\< c'2 c'2 c'2
477 \revert Hairpin.to-barline
478 \after 1*25/64 \!
479 { c'2 \constante #CENTER ##f \ff }
480 \bar "|."
481 }
482

```

4.2.4 Discussion

For the purpose of using it in my own musical typesetting, this constante hairpin snippet has been designed to work with a narrower hairpin height, e.g.

`\override Hairpin.height = #0.5` (the default is 0.666.)

[Table of Contents](#)

4.3 Enhanced Constante Hairpins II (Over The Staff)



4.3.1 Description

NB: Slight revision (December 23, 2025) has been carried out to make sure the hairpin terminates at the end of the staff line when it breaks out to the next system.

The motivation for creating an enhancement to the default constante hairpin provided by LilyPond was that it would not allow the hook to be inverted when placed over the staff.

See [Enhanced Constante Hairpins I \(Under The Staff\)](#) for more information.

4.3.2 Grammar

NOTE \constante \DYNAMIC NOTE(S) \DYNAMIC OR \!

This \constante accepts three optional arguments:

1. Vertical offset of the constante hairpin, indicated by a direction #UP, #CENTER, or #DOWN. Without specifying this argument, the default is #CENTER. You may use #DOWN when it is followed by a decrescendo hairpin, while you may use #UP when it is followed by a crescendo hairpin.²
2. Whether or not to use cautionary dynamics in subsequent systems when the hairpin breaks into new systems, indicated by ##f or ##t. The default is ##f.

² Note that the directions are flipped from Enhanced Constante Hairpins I!

3. Extra vertical adjustment is possible with the third argument by specifying a number. The default is #0.

Furthermore, when using the dynamic *p*, *pp*, etc., it is possible that the initial horizontal line may be too high. In this case, use `\tweak Y-offset #0` to be placed between the numerical value for the extra vertical adjustment and the dynamics.

4.3.3 Code

```

1  \version "2.24.4"
2
3
4  qf = #(make-dynamic-script
5      (markup
6          #:pad-x -1
7          ( #:combine
8              #:combine
9              #:translate '(-0.75 . 0)
10             #:normal-text (#:italic #:fontsize 0.75 "\"")
11             #:dynamic "f"
12             #:translate '(1.65 . 0)
13             #:normal-text (#:italic #:fontsize 0.75 "\"")))))
14
15  qmp = #(make-dynamic-script
16      (markup
17          #:pad-x -1
18          ( #:combine
19              #:combine
20              #:translate '(-0.85 . -0.1)
21              #:normal-text (#:italic #:fontsize 0.75 "\"")
22              #:dynamic "mp"
23              #:translate '(3.25 . -0.1)
24              #:normal-text (#:italic #:fontsize 0.75 "\"")))))
25
26
27  constanteUP =
28  #(define-music-function ( vOff cautionary moreOff dyn)
29      ( (ly:dir? CENTER) (boolean? #t) (number? 0) ly:event?  )
30      (define-public
31          ((constante-hairpin-revised coords mirrored?) grob)
32          "This scheme code was originally the scheme code 'elbowed-hairpin
33 taken from output-lib.scm, and was revised to suit the need."
34          (define (scale-coords coords-list x y)
35              (map
36                  (lambda (coord) (cons (* x (car coord)) (* y (cdr coord))))
37                  coords-list))
38
39      (define (hairpin::print-part points decresc? me)
```

```

40     (let ((stil (make-connected-line points me)))
41       (ly:stencil-scale stil 1 1)))
42
43 ;; outer let to trigger suicide
44 (let ((sten (ly:hairpin::print grob)))
45   (if (grob::is-live? grob)
46       (let*
47         (
48           ;I've left decresc? function but removed all the if clause
49           ;associated with it, to simplify the code a little bit
50           (verticalOffset vOff)
51           (useCautionary? cautionary)
52           (moreOffset moreOff)
53           (hairpinHeight (ly:grob-property grob 'height))
54           (decresc? (eqv? (ly:grob-property grob 'grow-direction) LEFT))
55           (toBarline? (ly:grob-property grob 'to-barline))
56           ; (circled-tip? (ly:grob-property grob 'circled-tip))
57           (hairpinTypeA?
58             (and (eqv? (end-broken-spanner? grob )           #f)
59                  (eqv? (first-broken-spanner? grob )          #f)
60                  (eqv? (middle-broken-spanner? grob )         #f)
61                  (eqv? (not-first-broken-spanner? grob )       #f)
62                  (eqv? (not-last-broken-spanner? grob )        #f)
63                  (eqv? (unbroken-or-first-broken-spanner? grob ) #t)
64                  (eqv? (unbroken-or-last-broken-spanner? grob ) #t)
65                  (eqv? (unbroken-spanner? grob )              #t)
66                  )) ;Hairpins that end within one line without
67           ;reaching to the end of the system
68           (hairpinTypeB?
69             (or (and (eqv? (end-broken-spanner? grob )           #t)
70                      (eqv? (first-broken-spanner? grob )          #t)
71                      (eqv? (middle-broken-spanner? grob )         #f)
72                      (eqv? (not-first-broken-spanner? grob )       #f)
73                      (eqv? (not-last-broken-spanner? grob )        #f)
74                      (eqv? (unbroken-or-first-broken-spanner? grob ) #t)
75                      (eqv? (unbroken-or-last-broken-spanner? grob ) #t)
76                      (eqv? (unbroken-spanner? grob )              #f)
77                      )
78              (and (eqv? (end-broken-spanner? grob )           #f)
79                   (eqv? (first-broken-spanner? grob )          #t)
80                   (eqv? (middle-broken-spanner? grob )         #f)
81                   (eqv? (not-first-broken-spanner? grob )       #f)
82                   (eqv? (not-last-broken-spanner? grob )        #t)
83                   (eqv? (unbroken-or-first-broken-spanner? grob ) #t)
84                   (eqv? (unbroken-or-last-broken-spanner? grob ) #f)
85                   (eqv? (unbroken-spanner? grob )              #f)
86                   )
87              )) ;Hairpins that breaks into next line

```



```

88      (hairpinTypeC?
89        (and (eqv? (end-broken-spanner? grob )           #f)
90              (eqv? (first-broken-spanner? grob )         #f)
91              (eqv? (middle-broken-spanner? grob )        #t)
92              (eqv? (not-first-broken-spanner? grob )      #t)
93              (eqv? (not-last-broken-spanner? grob )       #t)
94              (eqv? (unbroken-or-first-broken-spanner? grob ) #f)
95              (eqv? (unbroken-or-last-broken-spanner? grob ) #f)
96              (eqv? (unbroken-spanner? grob )             #f)
97              )) ;Midway-through Hairpins
98      (hairpinTypeD?
99        (and (eqv? (end-broken-spanner? grob )           #t)
100             (eqv? (first-broken-spanner? grob )         #f)
101             (eqv? (middle-broken-spanner? grob )        #f)
102             (eqv? (not-first-broken-spanner? grob )      #t)
103             (eqv? (not-last-broken-spanner? grob )       #f)
104             (eqv? (unbroken-or-first-broken-spanner? grob ) #f)
105             (eqv? (unbroken-or-last-broken-spanner? grob ) #t)
106             (eqv? (unbroken-spanner? grob )             #f)
107             )) ;multi-system Hairpin that terminates
108      (xex
109        (cons (car (ly:stencil-extent sten X))
110              (- (cdr (ly:stencil-extent sten X)) 1.25))
111      )
112      (lenx (interval-length xex))
113      (yex (ly:stencil-extent sten Y))
114      (leny (interval-length yex))
115      (xtrans (+ (car xex) 0))
116      (ytrans (car yex))
117      (uplist (scale-coords coords lenx (/ leny 2)))
118
119      (crescStencil
120
121        (if toBarline?
122          ;I added here to make it default to stretch
123          ;the hairpin immediately to the dynamics
124
125          (ly:stencil-translate
126            (hairpin::print-part
127              (scale-coords coords
128                (+ lenx 2.35) (/ leny 2))
129              decresc? grob)
130
131            '(-0.5 . 0))
132          ;if not to barline
133          (ly:stencil-translate
134            (hairpin::print-part
135              (scale-coords coords

```



```

184         0.35
185         #:dynamic (ly:music-property dyn 'text)
186         #:normal-text
187         (#:translate '(0.5 . -0.25)
188           #:italic (#:fontsize -0.5 "))))
189     )
190   )
191 )
192 ;when in-house dynamics (e.g. \p) are used
193 (grob-interpret-markup
194   grob
195   (markup
196     #:hspace 2
197     #:center-align
198     #:whiteout
199     #:concat
200     (#:normal-text
201       (#:translate '(0 . -0.25)
202         #:italic (#:fontsize 0.5 "("
203           ))
204       (#:translate '(0 . -0.25)
205         #:italic (#:fontsize -10 " "
206           ))
207       #:pad-x
208       0
209       #:dynamic (ly:music-property dyn 'text)
210       #:normal-text
211       (#:translate '(0 . -0.25)
212         #:italic (#:fontsize 0.5 "))))
213   )
214 ))
215 empty-stencil)
216 1
217 1
218 (hairpin::print-part (scale-coords
219   '((0.01 . 0.15) (1 . 0.15))
220   (+ lenx 1) (/ leny 2))
221   decresc? grob)
222 -1.5
223 )
224 )
225 (crescStencilBrokenEnding
226   ;if the flare end is applied
227   ;on the dynamic at the beginning
228   ;of the system, meaning lenx is super low
229
230   (cond
231     (< lenx 1)

```

```

232 (ly:stencil-combine-at-edge
233   ;if the cautionary accidentals are turned on
234   (if cautionary
235     (if (list? (ly:music-property dyn 'text))
236       ;this means, when the custom-made dynamics
237       ;(e.g. \qmp) are used
238       (grob-interpret-markup
239         grob
240         (markup
241           #:hspace 0
242           #:whiteout
243           #:center-align
244           #:concat
245           (#:normal-text
246             (#:translate '(0 . -0.25)
247               #:italic (#:fontsize -0.5 "( "
248                 ))
249             #:pad-x
250             0.25
251             #:dynamic (ly:music-property dyn 'text)
252             #:normal-text
253             (#:translate '(0.5 . -0.25)
254               #:italic (#:fontsize -0.5 ")"))))
255         )
256       )
257   ;when in-house dynamics (e.g. \p) are used
258   (grob-interpret-markup
259     grob
260     (markup
261       #:hspace 0
262
263       #:whiteout
264       #:center-align
265       (
266         #:concat
267         (#:normal-text
268           (#:translate '(0 . -0.25)
269             #:italic (#:fontsize 0.5 "( "
270               ))
271           (#:translate '(0 . -0.25)
272             #:italic (#:fontsize -10 " "
273               ))
274         #:pad-x
275         0
276         #:dynamic (ly:music-property dyn 'text)
277         #:normal-text
278         (#:translate '(0 . -0.25)
279           #:italic (#:fontsize 0.5 ")"))))

```

```

280             ))
281         ))
282         empty-stencil
283     )
284     1
285     1
286     (hairpin::print-part
287       (scale-coords coords
288         (+ lenx 1.85) (/ leny 2) )
289       decresc? grob)
290     ;vertical alignment (to be refined possibly...?)
291     -1.75
292   )
293 )
294 ((>= lenx 1)
295   ;if the normal ending flare crescendo
296   ;is applied...
297   (ly:stencil-combine-at-edge
298     ;if the cautionary accidentals are turned on
299     (if cautionary
300       (if (list? (ly:music-property dyn 'text))
301         ;this means, when the custom-made dynamics
302         ;(e.g. \qmp) are used
303         (grob-interpret-markup
304           grob
305           (markup
306             #:hspace 2
307             #:whiteout
308             #:center-align
309             #:concat
310             (#:normal-text
311               (#:translate '(0 . -0.25)
312                 #:italic (#:fontsize -0.5 "( "
313                   ))
314               #:pad-x 0.25
315               #:dynamic (ly:music-property dyn 'text)
316               #:normal-text
317               (#:translate '(0.5 . -0.25)
318                 #:italic (#:fontsize -0.5 ")"))))
319             )
320           )
321     ;if the in-house dynamics are used
322
323     (grob-interpret-markup
324       grob
325       (markup
326         #:hspace 2
327         #:whiteout

```

```

328         #:center-align
329         #:concat
330         (#:normal-text
331           (#:translate '(0 . -0.25)
332             #:italic (#:fontsize 0.5 "("
333                       ))
334           (#:translate '(0 . -0.25)
335             #:italic (#:fontsize -10 " "
336                       ))
337         #:pad-x
338         0
339         #:dynamic (ly:music-property dyn 'text)
340         #:normal-text
341         (#:translate '(0 . -0.25)
342           #:italic (#:fontsize 0.5 ")"))))
343       )
344     ))
345     empty-stencil
346   )
347   1
348   1
349   (hairpin::print-part
350     (scale-coords coords
351       (+ lenx 1.85) (/ leny 2) )
352     decresc? grob)
353   ;vertical alignment (to be refined possibly...?)
354   -1.75
355   )
356 )
357 )
358 )
359
360
361 (cresc
362   (cond (hairpinTypeA? crescStencil)
363         (hairpinTypeB? crescStencilBrokenBeginning)
364         (hairpinTypeC? crescStencilBrokenContinuation)
365         (hairpinTypeD? crescStencilBrokenEnding)
366   )
367 )
368
369 (stil
370   (ly:stencil-aligned-to
371     (ly:stencil-translate
372       cresc
373       (cons xtrans ytrans))
374     Y CENTER))
375   (stil-y-extent (ly:stencil-extent stil Y)))

```

```

376      ; FOR DEBUG ONLY
377      ; (display (list? (ly:music-property dyn 'text)))
378      ; (newline)
379      ; (display dyn)
380      ; (newline)
381      ; (newline)
382      ; (display "Is this type A Hairpin? " )
383      ; (display hairpinTypeA?)
384      ; (newline)
385      ; (display "Is this type B Hairpin? " )
386      ; (display hairpinTypeB?)
387      ; (newline)
388      ; (display "Is this type C Hairpin? " )
389      ; (display hairpinTypeC?)
390      ; (newline)
391      ; (display "Is this type D Hairpin? " )
392      ; (display hairpinTypeD?)
393      ;
394      ; (newline)
395      ; (display "Is it decrescendo? " )
396      ; (display decresc?)
397      ; (newline)
398      ; (display "end-broken-spanner???" )
399      ; (display (end-broken-spanner? grob ))
400      ; (newline)
401      ; (display "(first-broken-spanner?) Is spanner broken
402      ;           and the first of its broken siblings? " )
403      ; (display (first-broken-spanner? grob ) )
404      ; (newline)
405      ; (display "middle-broken-spanner???" )
406      ; (display (middle-broken-spanner? grob ) )
407      ; (newline)
408      ;
409      ; (display "Is spanner broken and not the first of
410      ;           its broken siblings? " )
411      ; (display (not-first-broken-spanner? grob ) )
412      ; (newline)
413      ;
414      ; (display "Is spanner broken and not the last of
415      ;           its broken siblings? " )
416      ; (display (not-last-broken-spanner? grob ) )
417      ; (newline)
418      ;
419      ; (display "unbroken-or-first-broken-spanner???" )
420      ; (display (unbroken-or-first-broken-spanner? grob ) )
421      ; (newline)
422      ;
423      ; (display "unbroken-or-first-broken-spanner???" )

```

```

424         ; (display (unbroken-or-last-broken-spanner? grob ) )
425         ; (newline)
426         ; (display "unbroken spanner?? " )
427         ; (display (unbroken-spanner? grob ) )
428         ; (newline)
429         ; (newline)
430         ; (display "distance of xex " )
431         ; (display lenx )
432         ; (newline)
433         ; (newline)
434         (ly:make-stencil (ly:stencil-expr stil) xex stil-y-extent))
435     ;; return empty, if no Hairpin.stencil present.
436     '()))
437
438     #{
439     #dyn
440     \tweak Hairpin.stencil
441     #(lambda (grob)
442       ;depending on the hairpin opening width,
443       ;you can adjust the value here vvv
444       (constante-hairpin-revised '((0 . 0) (1.0 . 0) (1.0 . -0.9)) #f)
445
446     )
447     % This calculates the tweak value for
448     % the value of the constante V-offset
449     \tweak Y-offset #(lambda (grob)
450       (+ (* vOff (ly:grob-property grob 'height))
451          (* (ly:grob-property grob 'height) -0.5 vOff )
452          moreOff
453        )
454     )
455
456     \<
457     #}
458     )
459   \layout {
460     indent = #0
461     line-width = #100
462     ragged-last = ##f
463   }
464   {
465     \dynamicUp
466     \override Hairpin.height = #0.5
467     \override Hairpin.to-barline = ##t
468     % \override Hairpin.stencil = #constante-hairpin
469     \override Hairpin.after-line-breaking = ##t
470     c'2 \constanteUP #DOWN \mf c'2 c'2 \> c'2
471     c'2 \constanteUP #UP ##t #0.3 \tweak Y-offset #0 \p c'2

```



```

472 \break c'4 4 4 4 c'4 4 4 4 \break
473 c''2 \< c'2 \! \constanteUP #UP ##f \qf \break
474 c'2 c'2\< c'2 c'2 c'2
475 \revert Hairpin.to-barline
476 \afterGrace 31/32
477 { c'2 \constanteUP #CENTER ##f \ff} s32\! \break
478 c'2\f c'2\< c'2 c'2 c'2
479 \revert Hairpin.to-barline
480 \after 1*25/64 \!
481 { c'2 \constanteUP #CENTER ##f \ff }
482 \bar "|."
483 }
484

```

4.3.4 Discussion

For the purpose of using it in my own musical typesetting, this `constante` hairpin snippet has been designed to work with a narrower hairpin height, e.g.

```
\override Hairpin.height = #0.5 (the default is 0.666.)
```

[Table of Contents](#)

4.4 Enhanced Flared Hairpins I (Original Version)



4.4.1 Description

NB: Significantly updated on December 23, 2025

(See [Enhanced Flared Hairpins II](#) for the flared hairpins with curved flares.)

In LilyPond, it is possible to specify flared hairpins by writing an override command:

```
\override Hairpin.stencil = #flared-hairpin
```

There are several aspects to LilyPond’s default flared hairpin that could be seen as limitations and would be best overcome:

1. Because the flare of the hairpin is drawn using the scheme code `'elbowed-hairpin` in the file `output-lib.scm` that uses a list of points from 0 (beginning) to 1 (end), the longer the hairpin is, the longer its flare becomes. While there may be a case where the longer flare is preferred, in the case of a long hairpin, it becomes more difficult to immediately discern a flared hairpin.
2. While LilyPond’s ordinary hairpins do have three separate components when they break into new systems: 1. beginning; 2. intermediate; and; 3. ending, the default flared hairpins draw identical “flared” hairpins when the lines break. This could invite confusions as it may tempt players to do flared crescendo/diminuendos at the end of each system.
3. When the default flared hairpins are activated, `\override Hairpin.circled-tips` (i.e. to apply circled tips at the onset of the crescendo or at the end of decrescendo) ceases to work.

In developing this enhanced flared hairpin, I have attempted to accommodate different case scenarios, e.g. having `\override Hairpin.to-barline`, `\override Hairpin.after-line-breaking` set to either `##t` or `##f`. It is subject to further refinements. I have also coded the circled-tip dynamic sign as well as tweak functions to be used when the circled-tip dynamic signs are used in tandem with the hairpins.

4.4.2 Grammar

```
\override Hairpin.stencil = #flared-hairpin-new
\override Hairpin.to-barline = ##f OR ##t
\override Hairpin.after-line-breaking = ##t OR ##g
```

If you wish to use circled tips, instead of using `\override Hairpin.circled-tip = ##t`, do:

```
NOTE \o \nCrescTweak \< NOTE(S) \DYNAMIC OR \! \nDecrescTweak \> NOTE(S) \o
```

Update: December 23, 2025 `\nCrescTweak` and `\nDecrescTweak` accepts a number as an optional argument. This functionality has been added to allow the adjustment of the hairpin at the arrival point. Think of this as an `X-offset` for the arrival point. See **Code**.

4.4.3 Code

```
1 \version "2.24.4"
2
3 % similar to the original version
4
5 #(define-public ((elbowed-hairpin-revised coords mirrored?) grob)
6   "This scheme code was originally the scheme code 'elbowed-hairpin
7   taken from output-lib.scm, and was revised to suit the need.
8   "
9   (define (scale-coords coords-list x y)
10     (map
11       (lambda (coord) (cons (* x (car coord)) (* y (cdr coord))))
12       coords-list))
13
14   (define (hairpin::print-part points decresc? me)
15     (let ((stil (make-connected-line points me)))
16       (if decresc? (ly:stencil-scale stil -1 1)
17         (ly:stencil-scale stil 1 1))))
18
19   ;; outer let to trigger suicide
20   (let ((sten (ly:hairpin::print grob)))
21     (if (grob::is-live? grob)
22       (let*
23         ((decresc? (eqv? (ly:grob-property grob 'grow-direction) LEFT))
24          (hairpinHeight (ly:grob-property grob 'height))
25          (toBarline? (ly:grob-property grob 'to-barline))
26          (orig (ly:grob-original grob))
27          (siblings (if (ly:grob? orig)
28            (ly:spanner-broken-into orig)
```

```

29         ' ()))
30     (hairpinTypeA?
31         (and (eqv? (end-broken-spanner? grob )           #f)
32              (eqv? (first-broken-spanner? grob )          #f)
33              (eqv? (middle-broken-spanner? grob )         #f)
34              (eqv? (not-first-broken-spanner? grob )      #f)
35              (eqv? (not-last-broken-spanner? grob )       #f)
36              (eqv? (unbroken-or-first-broken-spanner? grob ) #t)
37              (eqv? (unbroken-or-last-broken-spanner? grob ) #t)
38              (eqv? (unbroken-spanner? grob )              #t)
39              )) ;Hairpins that end within one line without reaching
40 ;to the end of the system
41     (hairpinTypeB?
42         (or (and (eqv? (end-broken-spanner? grob )           #t)
43                 (eqv? (first-broken-spanner? grob )          #t)
44                 (eqv? (middle-broken-spanner? grob )         #f)
45                 (eqv? (not-first-broken-spanner? grob )      #f)
46                 (eqv? (not-last-broken-spanner? grob )       #f)
47                 (eqv? (unbroken-or-first-broken-spanner? grob ) #t)
48                 (eqv? (unbroken-or-last-broken-spanner? grob ) #t)
49                 (eqv? (unbroken-spanner? grob )              #f)
50                 )
51             (and (eqv? (end-broken-spanner? grob )           #f)
52                  (eqv? (first-broken-spanner? grob )          #t)
53                  (eqv? (middle-broken-spanner? grob )         #f)
54                  (eqv? (not-first-broken-spanner? grob )      #f)
55                  (eqv? (not-last-broken-spanner? grob )       #t)
56                  (eqv? (unbroken-or-first-broken-spanner? grob ) #t)
57                  (eqv? (unbroken-or-last-broken-spanner? grob ) #f)
58                  (eqv? (unbroken-spanner? grob )              #f)
59                  )
60             )) ;Hairpins that breaks into next line
61     (hairpinTypeC?
62         (and (eqv? (end-broken-spanner? grob )           #f)
63              (eqv? (first-broken-spanner? grob )          #f)
64              (eqv? (middle-broken-spanner? grob )         #t)
65              (eqv? (not-first-broken-spanner? grob )      #t)
66              (eqv? (not-last-broken-spanner? grob )       #t)
67              (eqv? (unbroken-or-first-broken-spanner? grob ) #f)
68              (eqv? (unbroken-or-last-broken-spanner? grob ) #f)
69              (eqv? (unbroken-spanner? grob )              #f)
70              )) ;Midway-through Hairpins
71     (hairpinTypeD?
72         (and (eqv? (end-broken-spanner? grob )           #t)
73              (eqv? (first-broken-spanner? grob )          #f)
74              (eqv? (middle-broken-spanner? grob )         #f)
75              (eqv? (not-first-broken-spanner? grob )      #t)
76              (eqv? (not-last-broken-spanner? grob )       #f)

```

```

77         (eqv? (unbroken-or-first-broken-spanner? grob ) #f)
78         (eqv? (unbroken-or-last-broken-spanner? grob )  #t)
79         (eqv? (unbroken-spanner? grob )                  #f)
80         )) ;multi-system Hairpin that terminates
81 (xex (if decresc?
82       (cons (+ (car (ly:stencil-extent sten X)) 0.75)
83             (cdr (ly:stencil-extent sten X)))
84       (cons (car (ly:stencil-extent sten X))
85             (- (cdr (ly:stencil-extent sten X)) 1.25))
86       ))
87 (lenx (interval-length xex))
88 (yex (ly:stencil-extent sten Y))
89 (leny (interval-length yex))
90 (xtrans (+ (car xex) (if decresc? lenx 0)))
91 (ytrans (car yex))
92 (uplist (scale-coords coords lenx (/ leny 2)))
93 (downlist (scale-coords coords lenx (/ leny -2)))
94 ; crescStencil is used when the flared crescendo takes place
95 ; and finishes within one line.
96 (crescStencil
97   (if toBarline?
98     ;I added here to make it default to stretch
99     ;the hairpin immediately to the dynamics
100    (ly:stencil-combine-at-edge
101      (ly:stencil-combine-at-edge
102        (hairpin::print-part
103          (scale-coords
104            coords
105            (+ lenx 0.95) (/ leny 2)) decresc? grob)
106        1
107        -1
108        (if mirrored?
109          (hairpin::print-part
110            (scale-coords
111              coords
112              (+ lenx 0.95) (/ leny -2)) decresc? grob)
113          empty-stencil)
114        -0.1 ;vertical padding value here
115      )
116    0
117    1
118    (make-path-stencil
119      (list
120        'moveto 0 (* 0.2 (/ hairpinHeight 0.666) )
121        'rlineto 0.25 0.3
122        'moveto 0 (* -0.2 (/ hairpinHeight 0.666) )
123        'rlineto 0.25 -0.3
124      )

```

```

125         0.1
126         2
127         2
128         #f
129     )
130     -0.075
131 )
132 ;if not to barline
133 (ly:stencil-combine-at-edge
134   (ly:stencil-combine-at-edge
135     (hairpin::print-part
136       (scale-coords
137         coords
138         (+ lenx 0.95) (/ leny 2)) decresc? grob)
139     1
140     -1
141     (if mirrored?
142       (hairpin::print-part
143         (scale-coords
144           coords
145           (+ lenx 0.95) (/ leny -2)) decresc? grob)
146       empty-stencil)
147     -0.1 ;vertical padding value here
148   )
149   0
150   1
151   (make-path-stencil
152
153     (list
154       'moveto 0 (* 0.2 (/ hairpinHeight 0.666) )
155       'rlineto 0.25 0.3
156       'moveto 0 (* -0.2 (/ hairpinHeight 0.666) )
157       'rlineto 0.25 -0.3
158     )
159     0.1
160     2
161     2
162     #f
163   )
164   -0.075
165 ))
166 )
167 ; when crescendo begins but breaks off to
168 ; the subsequent system(s)
169 (crescStencilBrokenBeginning
170   ; if the hairpin is set only to barline...
171   (if toBarline?
172     ; ... AND if the hairpin finishes at the beginning

```

```

173         ; of the next line
174         ( if (= (length siblings) 1)
175             (ly:stencil-combine-at-edge
176               (ly:stencil-combine-at-edge
177                 (hairpin::print-part
178                   (scale-coords
179                     coords
180                     (+ lenx 0.95) (/ leny 2)) decresc? grob)
181                 1
182                 -1
183                 (if mirrored?
184                   (hairpin::print-part
185                     (scale-coords
186                       coords
187                       (+ lenx 0.95) (/ leny -2)) decresc? grob)
188                   empty-stencil)
189                 -0.1 ;vertical padding value here
190               )
191             0
192             1
193             (make-path-stencil
194               (list
195                 'moveto 0 (* 0.2 (/ hairpinHeight 0.666) )
196                 'rlineto 0.25 0.3
197                 'moveto 0 (* -0.2 (/ hairpinHeight 0.666) )
198                 'rlineto 0.25 -0.3
199               )
200             0.1
201             2
202             2
203             #f
204             )
205             -0.075
206             )
207         (ly:stencil-combine-at-edge
208           (hairpin::print-part
209             (scale-coords
210               coords
211               (+ lenx 1) (/ leny 1.5)) decresc? grob)
212           1
213           -1
214           (if mirrored?
215             (hairpin::print-part
216               (scale-coords
217                 coords
218
219                 (+ lenx 1) (/ leny -1.5)) decresc? grob)
220             empty-stencil)

```

```

221             -0.1 ;vertical padding value here
222         )
223     )
224     (ly:stencil-combine-at-edge
225     (hairpin::print-part
226     (scale-coords
227     coords
228     (+ lenx 1) (/ leny 1.5)) decresc? grob)
229     1
230     -1
231     (if mirrored?
232     (hairpin::print-part
233     (scale-coords
234     coords
235
236     (+ lenx 1) (/ leny -1.5)) decresc? grob)
237     empty-stencil)
238     -0.1 ;vertical padding value here
239     )
240     )
241 )
242 (crescStencilBrokenContinuation
243 (ly:stencil-combine-at-edge
244 (hairpin::print-part
245 (scale-coords
246 coords
247 (+ lenx 1) (/ leny 2)) decresc? grob)
248 1
249 -1
250 (if mirrored?
251 (hairpin::print-part
252 (scale-coords
253 coords
254 (+ lenx 1) (/ leny -2)) decresc? grob)
255 empty-stencil)
256 0.1 ;vertical padding value here
257 )
258 )
259 (crescStencilBrokenEnding
260 ;if the flare end is applied
261 ;on the dynamic at the beginning
262 ;of the system, meaning lenx is super low
263 (cond ((< lenx 0.2)
264 (make-path-stencil
265 (list 'moveto 0.45 (* 0.2 (/ hairpinHeight 0.666) )
266 'rlineto -0.65 -0.1
267 'moveto 0.45 (* 0.2 (/ hairpinHeight 0.666) )
268 'rlineto 0.25 0.3

```



```

269             'moveto 0.45 (* -0.2 (/ hairpinHeight 0.666) )
270             'rlineto -0.65 0.1
271             'moveto 0.45 (* -0.2 (/ hairpinHeight 0.666) )
272             'rlineto 0.25 -0.3
273         )
274     0.1
275     2
276     2
277     #f
278 ))
279 ((and (>= lenx 0.2)(< lenx 2))
280 ;if the flare end is applied
281 ;on the note at the beginning of the system
282 ;without the dynamics (lenx is about 1 to 2)
283 (ly:stencil-combine-at-edge
284 (ly:stencil-combine-at-edge
285 (hairpin::print-part
286 (scale-coords
287 (list
288 (cons (- (car (car coords)) 0)
289 (cdr (car coords) )))
290 (cons (- (car (cadr coords)) 0)
291 (cdr (cadr coords) )))
292 (+ lenx 0.25) (/ leny 2.75)) decresc? grob)
293 1
294 -1
295 (if mirrored?
296 (hairpin::print-part
297 (scale-coords
298 (list
299 (cons (- (car (car coords)) 0)
300 (cdr (car coords) )))
301 (cons (- (car (cadr coords)) 0)
302 (cdr (cadr coords) )))
303 (+ lenx 0.25) (/ leny -2.75)) decresc? grob)
304 empty-stencil)
305 0.1 ;vertical padding value here
306 )
307 0
308 1
309 (make-path-stencil
310 (list 'moveto 0 (* 0.15 (/ hairpinHeight 0.666) )
311 'rlineto 0.25 0.3
312 'moveto 0
313 (- (* -0.15 (/ hairpinHeight 0.666) ) 0.1)
314 'rlineto 0.25 -0.3
315 )
316 0.1

```

```

317         2
318         2
319         #f
320     )
321     -0.075
322 ))
323 ((>= lenx 2)
324   ;if the normal ending flare crescendo
325   ;is applied...
326   (ly:stencil-combine-at-edge
327     (ly:stencil-combine-at-edge
328       (hairpin::print-part
329         (scale-coords
330           coords
331           (+ lenx 0.75) (/ leny 3)) decresc? grob)
332     1
333     -1
334     (if mirrored?
335       (hairpin::print-part
336         (scale-coords
337           coords
338           (+ lenx 0.75) (/ leny -3)) decresc? grob)
339       empty-stencil)
340     0.1 ;vertical padding value here
341   )
342   0
343   1
344   (make-path-stencil
345     (list 'moveto 0 (* 0.133 (/ hairpinHeight 0.666) )
346           'rlineto 0.25 0.3
347           'moveto 0
348           (- (* -0.133 (/ hairpinHeight 0.666) ) 0.1)
349           'rlineto 0.25 -0.3
350         )
351     0.1
352     2
353     2
354     #f
355     )
356     -0.075
357   )
358   (ly:stencil-combine-at-edge
359     (ly:stencil-combine-at-edge
360       (hairpin::print-part
361         (scale-coords
362           coords
363           (+ lenx 0.75) (/ leny 3)) decresc? grob)
364     1

```

```

365         -1
366         (if mirrored?
367           (hairpin::print-part
368             (scale-coords
369               coords
370               (+ lenx 0.75) (/ leny -3)) decresc? grob)
371           empty-stencil)
372         0.1 ;vertical padding value here
373       )
374     0
375     1
376     (make-path-stencil
377       (list 'moveto 0 (* 0.133 (/ hairpinHeight 0.666) )
378             'rlineto 0.25 0.3
379             'moveto 0
380             (- (* -0.133 (/ hairpinHeight 0.666) ) 0.1)
381             'rlineto 0.25 -0.3
382             )
383       0.1
384       2
385       2
386       #f
387       )
388     -0.075
389   ))
390 )
391 )
392 (decrescStencil
393   (if toBarline?
394     ;I added here to make it default to stretch
395     ;the hairpin immediately to the dynamics
396     (ly:stencil-combine-at-edge
397       (ly:stencil-combine-at-edge
398         (hairpin::print-part
399           (scale-coords
400             coords
401             (+ lenx 0.25) (/ leny 2)) decresc? grob)
402         1
403         -1
404         (if mirrored?
405           (hairpin::print-part
406             (scale-coords
407               coords
408               (+ lenx 0.25) (/ leny -2)) decresc? grob)
409           empty-stencil)
410         -0.1 ;vertical padding value here
411       )
412     0

```

```

413         -1
414     (make-path-stencil
415       (list 'moveto 0 (* 0.19 (/ hairpinHeight 0.666) )
416             'rlineto -0.25 0.3
417             'moveto 0 (* -0.19 (/ hairpinHeight 0.666) )
418             'rlineto -0.25 -0.3
419           )
420       0.1
421       2
422       2
423       #f
424     )
425     -0.075
426   )
427 ;if not to barline
428 (ly:stencil-combine-at-edge
429 (ly:stencil-combine-at-edge
430   (hairpin::print-part
431     (scale-coords
432       coords
433       (+ lenx 0.25) (/ leny 2)) decresc? grob)
434   1
435   -1
436   (if mirrored?
437     (hairpin::print-part
438       (scale-coords
439         coords
440         (+ lenx 0.25) (/ leny -2)) decresc? grob)
441     empty-stencil)
442   -0.1 ;vertical padding value here
443   )
444   0
445   -1
446   (make-path-stencil
447     (list
448       'moveto 0 (* 0.2 (/ hairpinHeight 0.666) )
449       'rlineto -0.25 0.3
450       'moveto 0 (* -0.2 (/ hairpinHeight 0.666) )
451       'rlineto -0.25 -0.3
452     )
453     0.1
454     2
455     2
456     #f
457   )
458   -0.075
459   ))
460 )

```

```

461
462      (decreascStencilBrokenBeginning
463      (ly:stencil-combine-at-edge
464      (ly:stencil-combine-at-edge
465      (hairpin::print-part
466      (scale-coords
467      coords
468      (+ lenx 0.2) (/ leny 2.5))
469      decreasc? grob)
470      1
471      -1
472      (if mirrored?
473      (hairpin::print-part
474      (scale-coords
475      coords
476      (+ lenx 0.2) (/ leny -2.5)) decreasc? grob)
477      empty-stencil)
478      0.1 ;vertical padding value here
479      )
480      0
481      -1
482      (make-path-stencil
483      (list 'moveto 0 (* 0.15 (/ hairpinHeight 0.666) )
484      'rlineto -0.25 0.3
485      'moveto 0
486      (- (* -0.15 (/ hairpinHeight 0.666) ) 0.1)
487      'rlineto -0.25 -0.3
488      )
489      0.1
490      2
491      2
492      #f
493      )
494      -0.075
495      )
496      )
497      (decreascStencilBrokenEnding
498      (ly:stencil-combine-at-edge
499      (hairpin::print-part uplist decreasc? grob)
500      1
501      -1
502      (if mirrored?
503      (hairpin::print-part downlist decreasc? grob)
504      empty-stencil)
505      -0.1 ;vertical padding value here
506      )
507      )
508      (cresc

```

```

509      (cond (hairpinTypeA? crescStencil)
510            (hairpinTypeB? crescStencilBrokenBeginning)
511            (hairpinTypeC? crescStencilBrokenContinuation)
512            (hairpinTypeD? crescStencilBrokenEnding)
513            )
514    )
515    (decrec
516      (cond (hairpinTypeA? decrescStencil)
517            (hairpinTypeB? decrescStencilBrokenBeginning)
518            (hairpinTypeC? crescStencilBrokenContinuation)
519            ;crescStencilBrokenContinuation can be used for decresc
520            ;as well, because it gets inverted.
521            (hairpinTypeD? decrescStencilBrokenEnding)
522            )
523    )
524    (stil
525      (ly:stencil-aligned-to
526        (ly:stencil-translate
527          (if decresc?
528            decrec
529            cresc
530          )
531        (cons xtrans ytrans))
532      Y CENTER))
533    (stil-y-extent (ly:stencil-extent stil Y)))
534  ; for debugging purposes
535  ;      (display "Is this type A Hairpin? " )
536  ;      (display hairpinTypeA?)
537  ;      (newline)
538  ;      (display "Is this type B Hairpin? " )
539  ;      (display hairpinTypeB?)
540  ;      (newline)
541  ;      (display "Is this type C Hairpin? " )
542  ;      (display hairpinTypeC?)
543  ;      (newline)
544  ;      (display "Is this type D Hairpin? " )
545  ;      (display hairpinTypeD?)
546  ;
547  ;      (newline)
548  ;      (display "Is it decrescendo? " )
549  ;      (display decresc?)
550  ;      (newline)
551  ;      (display "end-broken-spanner??? " )
552  ;      (display (end-broken-spanner? grob ))
553  ;      (newline)
554  ;      (display "(first-broken-spanner?) Is spanner broken
555  ;and the first of its broken siblings? " )
556  ;      (display (first-broken-spanner? grob ) )

```

```

557         ;      (newline)
558         ;      (display "middle-broken-spanner???" )
559         ;      (display (middle-broken-spanner? grob ) )
560         ;      (newline)
561         ;
562         ;      (display "Is spanner broken and not the first of
563 ;its broken siblings?" )
564         ;      (display (not-first-broken-spanner? grob ) )
565         ;      (newline)
566         ;
567         ;      (display "Is spanner broken and not the last of
568 ;its broken siblings?" )
569         ;      (display (not-last-broken-spanner? grob ) )
570         ;      (newline)
571         ;
572         ;      (display "unbroken-or-first-broken-spanner???" )
573         ;      (display (unbroken-or-first-broken-spanner? grob ) )
574         ;      (newline)
575         ;
576         ;      (display "unbroken-or-first-broken-spanner???" )
577         ;      (display (unbroken-or-last-broken-spanner? grob ) )
578         ;      (newline)
579         ;      (display "unbroken spanner?? " )
580         ;      (display (unbroken-spanner? grob ) )
581         ;      (newline)
582         ;      (newline)
583         ;      (display "distance of xex " )
584         ;      (display lenx )
585         ;      (newline)
586         ;      (newline)
587         (ly:make-stencil (ly:stencil-expr stil) xex stil-y-extent))
588     ;; return empty, if no Hairpin.stencil present.
589     '()))))
590
591     #(define-public flared-hairpin-new
592       (elbowed-hairpin-revised '((0 . 0) (1 . 0.5))
593         #t))
594
595     #(define-public o
596       (make-music 'AbsoluteDynamicEvent
597         'text
598         (markup
599           #:pad-x -inf.0
600           #:concat (
601             #:pad-to-box '(0 . 0.1) '(-0.75 . 1.75)
602             #:translate '(0 . 0.6)
603             #:musicglyph "scripts.flageolet"
604             ))

```

```

605         ))
606
607 nCrescTweak =
608 # (define-music-function (num dyn) ((number? 0) ly:event? )
609   (let* ((hOffset num))
610
611     #{ \tweak Hairpin.shorten-pair
612       # (lambda (grob)
613         (let* (
614           ;; have we been split?
615           (orig (ly:grob-original grob))
616
617           ;; if yes, get the split pieces (our siblings)
618           (siblings (if (ly:grob? orig)
619                         (ly:spanner-broken-into orig)
620                         '())))
621
622           (if (or (= (length siblings) 0)
623                 (and (>= (length siblings) 0)
624                       (eq? (first siblings) grob)))
625
626             )
627           (cons -0.6 num)))
628       )
629     \tweak Hairpin.after-line-breaking
630     # (lambda (grob)
631       (let* (
632         ;; have we been split?
633         (orig (ly:grob-original grob))
634
635         ;; if yes, get the split pieces (our siblings)
636         (siblings (if (ly:grob? orig)
637                       (ly:spanner-broken-into orig)
638                       '()))
639         (toBarline? (ly:grob-property grob 'to-barline)))
640
641       )
642
643       (if (and (>= (length siblings) 2)
644             (eq? (car (last-pair siblings)) grob))
645         (if toBarline?
646             (ly:grob-set-property!
647               grob 'shorten-pair
648               (cons 0 (+ (* 0.6 -1) hOffset)))
649             (ly:grob-set-property!
650               grob 'shorten-pair
651               (cons (+ -0.6 hOffset) -0.6))
652             )

```



```

653         )
654
655     )
656     ) #dyn #}))
657
658 nDecrescTweak =
659 #(define-music-function (num dyn) ((number? 0) ly:event? )
660   (let* ((hOffset num))
661     #{
662       \tweak Hairpin.shorten-pair
663       #(lambda (grob)
664         (let* (
665           ;; have we been split?
666           (orig (ly:grob-original grob))
667
668           ;; if yes, get the split pieces (our siblings)
669           (siblings (if (ly:grob? orig)
670                         (ly:spanner-broken-into orig)
671                         '()))))
672
673         (if (= (length siblings) 0)
674             (cons 0 (+ -0.6 (* hOffset -1))))))
675       )
676       \tweak Hairpin.after-line-breaking
677       #(lambda (grob)
678         (let* (
679           ;; have we been split?
680           (orig (ly:grob-original grob))
681
682           ;; if yes, get the split pieces (our siblings)
683           (siblings (if (ly:grob? orig)
684                         (ly:spanner-broken-into orig)
685                         '()))))
686
687         (if (and (>= (length siblings) 1)
688             (eq? (car (last-pair siblings)) grob))
689
690             (ly:grob-set-property! grob 'shorten-pair '(-0.6 . -0.6))))
691         ) #dyn #}))
692
693 \layout {
694   indent = #0
695   line-width = #100
696   ragged-last = ##t
697 }
698
699
700 {

```

```

701 % \override Hairpin.height = #0.5
702 \override Hairpin.stencil = #flared-hairpin-new
703 \override Hairpin.to-barline = ##f
704 \override Hairpin.after-line-breaking = ##f
705
706 c' \o \nCrescTweak \< c' c' c' \f \nDecrescTweak \>
707 c' c' \o \nCrescTweak #-0.75 \< c' c' \break
708 \repeat unfold 3 {c' c' c' c'} \break
709 c' \ff \nDecrescTweak \> c' c' c'
710 c' \p \nCrescTweak \< c' c' \nDecrescTweak #-0.6 \> c'
711 c' \p \< c' c' c' \ff \nDecrescTweak \> \break
712 \repeat unfold 3 {c' c' c' c'} \break
713 c' \f \nDecrescTweak \> c' c' c' \o \nCrescTweak \<
714 c' c' c' c' \!
715 }
716

```

4.4.4 Discussion

[Table of Contents](#)

4.5 Enhanced Flared Hairpins II (Curvy Version)



4.5.1 Description

NB: Significantly updated on December 23, 2025

See [Enhanced Flared Hairpins I](#) for the flared hairpins similar to [LilyPond's default flared hairpins](#).

This version implements hairpins with flares that are curved. Initially Santiago Beis inspired me to take advantage of Phi Curve in making the curvy flares as natural as possible, though I have departed from that approach as I found using `rcurve` useful in `make-path-stencil`, which I then combined with the crescendo lines using `ly:stencil-combine-at-edge`.

4.5.2 Grammar

```
\override Hairpin.stencil = #flared-hairpin-curvy
\override Hairpin.to-barline = ##f OR ##t
\override Hairpin.after-line-breaking = ##t OR ##g
```

If you wish to use circled tips, instead of using `\override Hairpin.circled-tip = ##t`, do:

```
NOTE \o \nCrescTweak \< NOTE(S) \DYNAMIC OR \! \nDecrescTweak \> NOTE(S) \o
```

Update: December 23, 2025 `\nCrescTweak` and `\nDecrescTweak` accepts a number as an optional argument. This functionality has been added to allow the adjustment of the hairpin at the arrival point. Think of this as an `X-offset` for the arrival point. See [Code](#).

4.5.3 Code

```

1  \version "2.24.4"
2
3  % curvy version
4
5  #(define-public ((curvy-elbowed-hairpin coords mirrored?) grob)
6    "This scheme code was originally the scheme code 'elbowed-hairpin
7    taken from output-lib.scm, and was revised to suit the need.
8    "
9    (define (scale-coords coords-list x y)
10      (map
11        (lambda (coord) (cons (* x (car coord)) (* y (cdr coord)))))
12      coords-list))
13
14    (define (hairpin::print-part points decresc? me)
15      (let ((stil (make-connected-line points me)))
16        (if decresc? (ly:stencil-scale stil -1 1)
17          (ly:stencil-scale stil 1 1))))
18
19    ;; outer let to trigger suicide
20    (let ((sten (ly:hairpin::print grob)))
21      (if (grob::is-live? grob)
22        (let*
23          ((decresc? (eqv? (ly:grob-property grob 'grow-direction) LEFT))
24           (hairpinHeight (ly:grob-property grob 'height))
25           (toBarline? (ly:grob-property grob 'to-barline))
26           (orig (ly:grob-original grob))
27           (siblings (if (ly:grob? orig)
28                         (ly:spanner-broken-into orig)
29                         '())))
30          (hairpinTypeA?
31            (and (eqv? (end-broken-spanner? grob )           #f)
32                  (eqv? (first-broken-spanner? grob )         #f)
33                  (eqv? (middle-broken-spanner? grob )        #f)
34                  (eqv? (not-first-broken-spanner? grob )      #f)
35                  (eqv? (not-last-broken-spanner? grob )       #f)
36                  (eqv? (unbroken-or-first-broken-spanner? grob ) #t)
37                  (eqv? (unbroken-or-last-broken-spanner? grob ) #t)
38                  (eqv? (unbroken-spanner? grob )              #t)
39                  )) ;Hairpins that end within one line without
40          ;reaching to the end of the system
41          (hairpinTypeB?
42            (or (and (eqv? (end-broken-spanner? grob )           #t)
43                     (eqv? (first-broken-spanner? grob )         #t)
44                     (eqv? (middle-broken-spanner? grob )        #f)
45                     (eqv? (not-first-broken-spanner? grob )      #f)
46                     (eqv? (not-last-broken-spanner? grob )       #f)

```

```

47         (eqv? (unbroken-or-first-broken-spanner? grob ) #t)
48         (eqv? (unbroken-or-last-broken-spanner? grob ) #t)
49         (eqv? (unbroken-spanner? grob ) #f)
50     )
51     (and (eqv? (end-broken-spanner? grob ) #f)
52         (eqv? (first-broken-spanner? grob ) #t)
53         (eqv? (middle-broken-spanner? grob ) #f)
54         (eqv? (not-first-broken-spanner? grob ) #f)
55         (eqv? (not-last-broken-spanner? grob ) #t)
56         (eqv? (unbroken-or-first-broken-spanner? grob ) #t)
57         (eqv? (unbroken-or-last-broken-spanner? grob ) #f)
58         (eqv? (unbroken-spanner? grob ) #f)
59     )
60     )) ;Hairpins that breaks into next line
61 (hairpinTypeC?
62     (and (eqv? (end-broken-spanner? grob ) #f)
63         (eqv? (first-broken-spanner? grob ) #f)
64         (eqv? (middle-broken-spanner? grob ) #t)
65         (eqv? (not-first-broken-spanner? grob ) #t)
66         (eqv? (not-last-broken-spanner? grob ) #t)
67         (eqv? (unbroken-or-first-broken-spanner? grob ) #f)
68         (eqv? (unbroken-or-last-broken-spanner? grob ) #f)
69         (eqv? (unbroken-spanner? grob ) #f)
70     )) ;Midway-through Hairpins
71 (hairpinTypeD?
72     (and (eqv? (end-broken-spanner? grob ) #t)
73         (eqv? (first-broken-spanner? grob ) #f)
74         (eqv? (middle-broken-spanner? grob ) #f)
75         (eqv? (not-first-broken-spanner? grob ) #t)
76         (eqv? (not-last-broken-spanner? grob ) #f)
77         (eqv? (unbroken-or-first-broken-spanner? grob ) #f)
78         (eqv? (unbroken-or-last-broken-spanner? grob ) #t)
79         (eqv? (unbroken-spanner? grob ) #f)
80     )) ;multi-system Hairpin that terminates
81 (xex (if decresc?
82     (cons (+ (car (ly:stencil-extent sten X)) 0.75)
83         (cdr (ly:stencil-extent sten X)))
84     (cons (car (ly:stencil-extent sten X))
85         (- (cdr (ly:stencil-extent sten X)) 1.75))
86 ))
87 (lenx (interval-length xex))
88 (yex (ly:stencil-extent sten Y))
89 (leny (interval-length yex))
90 (xtrans (+ (car xex) (if decresc? lenx 0)))
91 (ytrans (car yex))
92 (uplist (scale-coords coords lenx (/ leny 2)))
93 (downlist (scale-coords coords lenx (/ leny -2)))
94 ; crescStencil is used when the flared crescendo takes place

```

```

95         ; and finishes within one line.
96     (crescStencil
97       (if toBarline?
98         ;I added here to make it default to stretch
99         ;the hairpin immediately to the dynamics
100        (ly:stencil-combine-at-edge
101          (ly:stencil-combine-at-edge
102            (hairpin::print-part
103              (scale-coords
104                coords
105                (+ lenx 0.95) (/ leny 2)) decresc? grob)
106              1
107              -1
108              (if mirrored?
109                (hairpin::print-part
110                  (scale-coords
111                    coords
112                    (+ lenx 0.95) (/ leny -2)) decresc? grob)
113                  empty-stencil)
114              -0.1 ;vertical padding value here
115            )
116            0
117            1
118            (make-path-stencil
119              (list
120                'moveto 0 (* 0.2 (/ hairpinHeight 0.666) )
121                'rcurveto 0 0 0.3 -0.05 0.4 0.3
122                'moveto 0 (* -0.2 (/ hairpinHeight 0.666) )
123                'rcurveto 0 0 0.3 0.05 0.4 -0.3
124              )
125              0.1
126              2
127              2
128              #f
129            )
130            -0.075
131          )
132        ;if not to barline
133        (ly:stencil-combine-at-edge
134          (ly:stencil-combine-at-edge
135            (hairpin::print-part
136              (scale-coords
137                coords
138                (+ lenx 0.95) (/ leny 2)) decresc? grob)
139            1
140            -1
141            (if mirrored?
142              (hairpin::print-part

```

```

143         (scale-coords
144         coords
145         (+ lenx 0.95) (/ leny -2)) decresc? grob)
146     empty-stencil)
147     -0.1 ;vertical padding value here
148 )
149 0
150 1
151 (make-path-stencil
152
153     (list
154     'moveto 0 (* 0.2 (/ hairpinHeight 0.666) )
155     'rcurveto 0 0 0.3 -0.05 0.4 0.3
156     'moveto 0 (* -0.2 (/ hairpinHeight 0.666) )
157     'rcurveto 0 0 0.3 0.05 0.4 -0.3
158     )
159     0.1
160     2
161     2
162     #f
163     )
164     -0.075
165     ))
166 )
167 ; when crescendo begins but breaks off to
168 ; the subsequent system(s)
169 (crescStencilBrokenBeginning
170 ; if the hairpin is set only to barline...
171 (if toBarline?
172     ; ... AND if the hairpin finishes at the beginning
173     ; of the next line
174     ( if (= (length siblings) 1)
175         (ly:stencil-combine-at-edge
176         (ly:stencil-combine-at-edge
177         (hairpin::print-part
178         (scale-coords
179         coords
180         (+ lenx 0.95) (/ leny 2)) decresc? grob)
181         1
182         -1
183         (if mirrored?
184             (hairpin::print-part
185             (scale-coords
186             coords
187             (+ lenx 0.95) (/ leny -2)) decresc? grob)
188             empty-stencil)
189         -0.1 ;vertical padding value here
190         )

```

```

191         0
192         1
193         (make-path-stencil
194         (list
195         'moveto 0 (* 0.2 (/ hairpinHeight 0.666) )
196         'rcurveto 0 0 0.3 -0.05 0.4 0.3
197         'moveto 0 (* -0.2 (/ hairpinHeight 0.666) )
198         'rcurveto 0 0 0.3 0.05 0.4 -0.3
199         )
200         0.1
201         2
202         2
203         #f
204         )
205         -0.075
206         )
207         (ly:stencil-combine-at-edge
208         (hairpin::print-part
209         (scale-coords
210         coords
211         (+ lenx 1) (/ leny 1.5)) decresc? grob)
212         1
213         -1
214         (if mirrored?
215         (hairpin::print-part
216         (scale-coords
217         coords
218
219         (+ lenx 1) (/ leny -1.5)) decresc? grob)
220         empty-stencil)
221         -0.1 ;vertical padding value here
222         )
223         )
224         (ly:stencil-combine-at-edge
225         (hairpin::print-part
226         (scale-coords
227         coords
228         (+ lenx 1) (/ leny 1.5)) decresc? grob)
229         1
230         -1
231         (if mirrored?
232         (hairpin::print-part
233         (scale-coords
234         coords
235
236         (+ lenx 1) (/ leny -1.5)) decresc? grob)
237         empty-stencil)
238         -0.1 ;vertical padding value here

```



```

239         )
240     )
241 )
242 (crescStencilBrokenContinuation
243   (ly:stencil-combine-at-edge
244     (hairpin::print-part
245       (scale-coords
246         coords
247         (+ lenx 1) (/ leny 2)) decresc? grob)
248     1
249     -1
250     (if mirrored?
251       (hairpin::print-part
252         (scale-coords
253           coords
254           (+ lenx 1) (/ leny -2)) decresc? grob)
255       empty-stencil)
256     0.1 ;vertical padding value here
257   )
258 )
259 (crescStencilBrokenEnding
260   ;if the flare end is applied
261   ;on the dynamic at the beginning
262   ;of the system, meaning lenx is super low
263   (cond ((< lenx 0.2)
264     (make-path-stencil
265       (list 'moveto 0.45 (* 0.2 (/ hairpinHeight 0.666) )
266         'lineto -0.5 0.1
267         'moveto 0.45 (* 0.2 (/ hairpinHeight 0.666) )
268         'rcurveto 0 0 0.3 0 0.4 0.3
269         'moveto 0.45 (* -0.2 (/ hairpinHeight 0.666) )
270         'lineto -0.5 -0.1
271         'moveto 0.45 (* -0.2 (/ hairpinHeight 0.666) )
272         'rcurveto 0 0 0.3 0 0.4 -0.3
273       )
274     0.1
275     2
276     2
277     #f
278   ))
279   ((and (>= lenx 0.2)(< lenx 2))
280     ;if the flare end is applied
281     ;on the note at the beginning of the system
282     ;without the dynamics (lenx is about 1 to 2)
283     (ly:stencil-combine-at-edge
284       (ly:stencil-combine-at-edge
285         (hairpin::print-part
286           (scale-coords

```

```

287         (list
288           (cons (- (car (car coords)) 0)
289                 (cdr (car coords))))
290         (cons (- (car (cadr coords)) 0)
291               (cdr (cadr coords))))
292       (+ lenx 0.25) (/ leny 2.75)) decresc? grob)
293   1
294   -1
295   (if mirrored?
296       (hairpin::print-part
297         (scale-coords
298           (list
299             (cons (- (car (car coords)) 0)
300                   (cdr (car coords))))
301             (cons (- (car (cadr coords)) 0)
302                   (cdr (cadr coords))))
303           (+ lenx 0.25) (/ leny -2.75)) decresc? grob)
304     empty-stencil)
305   0.1 ;vertical padding value here
306   )
307   0
308   1
309   (make-path-stencil
310     (list 'moveto 0 (* 0.15 (/ hairpinHeight 0.666) )
311           'rcurveto 0 0 0.3 -0.05 0.4 0.3
312           'moveto 0
313           (- (* -0.15 (/ hairpinHeight 0.666) ) 0.1)
314           'rcurveto 0 0 0.3 0.05 0.4 -0.3
315           )
316     0.1
317     2
318     2
319     #f
320     )
321     -0.075
322     ))
323   ((>= lenx 2)
324     ;if the normal ending flare crescendo
325     ;is applied...
326     (ly:stencil-combine-at-edge
327       (ly:stencil-combine-at-edge
328         (hairpin::print-part
329           (scale-coords
330             coords
331             (+ lenx 0.75) (/ leny 3)) decresc? grob)
332         1
333         -1
334         (if mirrored?

```

```

335         (hairpin::print-part
336         (scale-coords
337         coords
338         (+ lenx 0.75) (/ leny -3)) decresc? grob)
339     empty-stencil)
340     0.1 ;vertical padding value here
341 )
342 0
343 1
344 (make-path-stencil
345   (list 'moveto 0 (* 0.133 (/ hairpinHeight 0.666) )
346         'rcurveto 0 0 0.3 -0.05 0.4 0.3
347         'moveto 0
348         (- (* -0.133 (/ hairpinHeight 0.666) ) 0.1)
349         'rcurveto 0 0 0.3 0.05 0.4 -0.3
350         )
351     0.1
352     2
353     2
354     #f
355     )
356     -0.075
357     )
358 (ly:stencil-combine-at-edge
359 (ly:stencil-combine-at-edge
360   (hairpin::print-part
361   (scale-coords
362   coords
363   (+ lenx 0.75) (/ leny 3)) decresc? grob)
364   1
365   -1
366   (if mirrored?
367     (hairpin::print-part
368     (scale-coords
369     coords
370     (+ lenx 0.75) (/ leny -3)) decresc? grob)
371     empty-stencil)
372     0.1 ;vertical padding value here
373     )
374     0
375     1
376     (make-path-stencil
377       (list 'moveto 0 (* 0.133 (/ hairpinHeight 0.666) )
378             'rcurveto 0 0 0.3 -0.05 0.4 0.3
379             'moveto 0
380             (- (* -0.133 (/ hairpinHeight 0.666) ) 0.1)
381             'rcurveto 0 0 0.3 0.05 0.4 -0.3
382             )

```

```

383         0.1
384         2
385         2
386         #f
387     )
388     -0.075
389 ))
390 )
391 )
392 (decreascStencil
393   (if toBarline?
394     ;I added here to make it default to stretch
395     ;the hairpin immediately to the dynamics
396     (ly:stencil-combine-at-edge
397       (ly:stencil-combine-at-edge
398         (hairpin::print-part
399           (scale-coords
400             coords
401             (+ lenx 0.25) (/ leny 2)) decreasc? grob)
402         1
403         -1
404         (if mirrored?
405           (hairpin::print-part
406             (scale-coords
407               coords
408               (+ lenx 0.25) (/ leny -2)) decreasc? grob)
409           empty-stencil)
410         -0.1 ;vertical padding value here
411       )
412       0
413       -1
414       (make-path-stencil
415         (list 'moveto 0 (* 0.19 (/ hairpinHeight 0.666) )
416               'rcurveto 0 0 -0.3 -0.05 -0.4 0.3
417               'moveto 0 (* -0.19 (/ hairpinHeight 0.666) )
418               'rcurveto 0 0 -0.3 0.05 -0.4 -0.3
419             )
420         0.1
421         2
422         2
423         #f
424       )
425       -0.075
426     )
427     ;if not to barline
428     (ly:stencil-combine-at-edge
429       (ly:stencil-combine-at-edge
430         (hairpin::print-part

```

```

431         (scale-coords
432         coords
433         (+ lenx 0.25) (/ leny 2)) decresc? grob)
434     1
435     -1
436     (if mirrored?
437     (hairpin::print-part
438     (scale-coords
439     coords
440     (+ lenx 0.25) (/ leny -2)) decresc? grob)
441     empty-stencil)
442     -0.1 ;vertical padding value here
443     )
444     0
445     -1
446     (make-path-stencil
447     (list
448     'moveto 0 (* 0.2 (/ hairpinHeight 0.666) )
449     'rcurveto 0 0 -0.3 -0.05 -0.4 0.3
450     'moveto 0 (* -0.2 (/ hairpinHeight 0.666) )
451     'rcurveto 0 0 -0.3 0.05 -0.4 -0.3
452     )
453     0.1
454     2
455     2
456     #f
457     )
458     -0.075
459     ))
460 )
461
462 (decrescStencilBrokenBeginning
463 (ly:stencil-combine-at-edge
464 (ly:stencil-combine-at-edge
465 (hairpin::print-part
466 (scale-coords
467 coords
468 (+ lenx 0.2) (/ leny 2.5))
469 decresc? grob)
470 1
471 -1
472 (if mirrored?
473 (hairpin::print-part
474 (scale-coords
475 coords
476 (+ lenx 0.2) (/ leny -2.5)) decresc? grob)
477 empty-stencil)
478 0.1 ;vertical padding value here

```

```

479         )
480     0
481     -1
482     (make-path-stencil
483       (list 'moveto 0 (* 0.15 (/ hairpinHeight 0.666) )
484             'rcurveto 0 0 -0.3 -0.05 -0.4 0.3
485             'moveto 0
486             (- (* -0.15 (/ hairpinHeight 0.666) ) 0.1)
487             'rcurveto 0 0 -0.3 0.05 -0.4 -0.3
488             )
489       0.1
490       2
491       2
492       #f
493     )
494     -0.075
495   )
496 )
497 (decreascStencilBrokenEnding
498   (ly:stencil-combine-at-edge
499     (hairpin::print-part uplist decreasc? grob)
500     1
501     -1
502     (if mirrored?
503       (hairpin::print-part downlist decreasc? grob)
504       empty-stencil)
505     -0.1 ;vertical padding value here
506   )
507 )
508 (cresc
509   (cond (hairpinTypeA? crescStencil)
510         (hairpinTypeB? crescStencilBrokenBeginning)
511         (hairpinTypeC? crescStencilBrokenContinuation)
512         (hairpinTypeD? crescStencilBrokenEnding)
513       )
514 )
515 (decreas
516   (cond (hairpinTypeA? decreascStencil)
517         (hairpinTypeB? decreascStencilBrokenBeginning)
518         (hairpinTypeC? crescStencilBrokenContinuation)
519         ;crescStencilBrokenContinuation can be used for
520         ;decreasc as well, because it gets inverted.
521         (hairpinTypeD? decreascStencilBrokenEnding)
522       )
523 )
524 (stil
525   (ly:stencil-aligned-to
526     (ly:stencil-translate

```

```

527         (if decresc?
528             decrec
529             cresc
530         )
531     (cons xtrans ytrans))
532     Y CENTER))
533     (stil-y-extent (ly:stencil-extent stil Y)))
534 ; for debugging purposes
535 ;     (display "Is this type A Hairpin? " )
536 ;     (display hairpinTypeA?)
537 ;     (newline)
538 ;     (display "Is this type B Hairpin? " )
539 ;     (display hairpinTypeB?)
540 ;     (newline)
541 ;     (display "Is this type C Hairpin? " )
542 ;     (display hairpinTypeC?)
543 ;     (newline)
544 ;     (display "Is this type D Hairpin? " )
545 ;     (display hairpinTypeD?)
546 ;
547 ;     (newline)
548 ;     (display "Is it decrescendo? " )
549 ;     (display decresc?)
550 ;     (newline)
551 ;     (display "end-broken-spanner??? " )
552 ;     (display (end-broken-spanner? grob ))
553 ;     (newline)
554 ;     (display "(first-broken-spanner?) Is spanner broken
555 ;and the first of its broken siblings? " )
556 ;     (display (first-broken-spanner? grob ) )
557 ;     (newline)
558 ;     (display "middle-broken-spanner??? " )
559 ;     (display (middle-broken-spanner? grob ) )
560 ;     (newline)
561 ;
562 ;     (display "Is spanner broken and not the first of
563 ;its broken siblings? " )
564 ;     (display (not-first-broken-spanner? grob ) )
565 ;     (newline)
566 ;
567 ;     (display "Is spanner broken and not the last of
568 ;its broken siblings? " )
569 ;     (display (not-last-broken-spanner? grob ) )
570 ;     (newline)
571 ;
572 ;     (display "unbroken-or-first-broken-spanner??? " )
573 ;     (display (unbroken-or-first-broken-spanner? grob ) )
574 ;     (newline)

```

```

575      ;
576      ;      (display "unbroken-or-first-broken-spanner???" )
577      ;      (display (unbroken-or-last-broken-spanner? grob ) )
578      ;      (newline)
579      ;      (display "unbroken spanner?? " )
580      ;      (display (unbroken-spanner? grob ) )
581      ;      (newline)
582      ;      (newline)
583      ;      (display "distance of xex " )
584      ;      (display lenx )
585      ;      (newline)
586      ;      (newline)
587      (ly:make-stencil (ly:stencil-expr stil) xex stil-y-extent))
588      ;; return empty, if no Hairpin.stencil present.
589      '()))
590
591      #(define-public flared-hairpin-curved
592        (curvy-elbowed-hairpin '((0 . 0) (1 . 0.5))
593          #t))
594
595      #(define-public o
596        (make-music 'AbsoluteDynamicEvent
597          'text
598          (markup
599            #:pad-x -inf.0
600            #:concat (
601              #:pad-to-box '(0 . 0.1) '(-0.75 . 1.75)
602              #:translate '(0 . 0.6)
603              #:musicglyph "scripts.flageolet"
604              ))
605          ))
606
607      nCrescTweak =
608      #(define-music-function (num dyn) ((number? 0) ly:event? )
609        (let* ((hOffset num))
610
611          #{ \tweak Hairpin.shorten-pair
612            #(lambda (grob)
613              (let* (
614                ;; have we been split?
615                (orig (ly:grob-original grob))
616
617                ;; if yes, get the split pieces (our siblings)
618                (siblings (if (ly:grob? orig)
619                              (ly:spanner-broken-into orig)
620                              '()))
621
622                (if (or (= (length siblings) 0)

```



```

671             '()))))
672
673         (if (= (length siblings) 0)
674             (cons 0 (+ -0.6 (* hOffset -1))))))
675     )
676     \tweak Hairpin.after-line-breaking
677     #(lambda (grob)
678         (let* (
679             ;; have we been split?
680             (orig (ly:grob-original grob))
681
682             ;; if yes, get the split pieces (our siblings)
683             (siblings (if (ly:grob? orig)
684                           (ly:spanner-broken-into orig)
685                           '()))))
686
687         (if (and (>= (length siblings) 1)
688             (eq? (car (last-pair siblings)) grob))
689
690             (ly:grob-set-property! grob 'shorten-pair '(-0.6 . -0.6))))
691     ) #dyn #}))
692
693 \layout {
694     indent = #0
695     line-width = #100
696     ragged-last = ##t
697 }
698
699
700 {
701     % \override Hairpin.height = #0.5
702     \override Hairpin.stencil = #flared-hairpin-curvy
703     \override Hairpin.to-barline = ##f
704     \override Hairpin.after-line-breaking = ##f
705
706     c' \o \nCrescTweak \< c' c' c' \f \nDecrescTweak \>
707     c' c' \o \nCrescTweak #-0.75 \< c' c' \break
708     \repeat unfold 3 {c' c' c' c'} \break
709     c' \ff \nDecrescTweak \> c' c' c'
710     c' \p \nCrescTweak \< c' c' \nDecrescTweak #-0.6 \> c'
711     c' \p \< c' c' c' \ff \nDecrescTweak \> \break
712     \repeat unfold 3 {c' c' c' c'} \break
713     c' \f \nDecrescTweak \> c' c' c' \o \nCrescTweak \<
714     c' c' c' c' \!
715 }
716

```

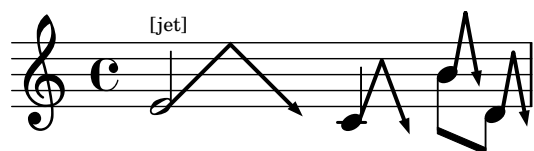
4.5.4 Discussion

[Table of Contents](#)

Chapter 5

Noteheads

5.1 Jet Whistle (for flute)



5.1.1 Description

Implementation of the jet whistle, as described in *The Techniques of Flute Playing* by Carin Levine and Christina Mitropoulos-Bott.¹

NB: (Aug. 10 2025) There is now [a new version of the jet whistle implementation](#).

5.1.2 Grammar

`\jet NOTE #X-length`

5.1.3 Code

```
1 jet = #(define-music-function (pitchthing width) (ly:music? number?)
2       (define p1 (ly:music-property pitchthing 'pitch))
3       (define steps (+ -6 (ly:pitch-steps p1)))
4       (define radToDeg (* 180 (/ 1 3.141592653589793)))
5       #{ #pitchthing ^\markup {
6           \postscript
7           #(string-append "gsave newpath 0.2 setlinewidth 1.15 "
8                           (number->string
9                             (+ -2.5 (* 0.5 steps))) " moveto "
10                          (number->string
```

1. Carin Levine and Christina Mitropoulos-Bott, *The techniques of flute playing = Die Spieltechnik der Flöte* (Kassel ; New York: Bärenreiter, 2003), 18.

```

11             (* 0.5 width)) " 4 rlineto "
12             (number->string
13             (* 0.5 width)) " -4 rlineto
14
15         stroke
16         newpath
17         0.1 setlinewidth "
18             (number->string (+ 1.15 width)) " "
19             (number->string (+ -2.55 (* 0.5 steps)))
20             " moveto "
21             (number->string
22             (* radToDeg (atan (/ (* width 0.5) 4))))
23             " rotate
24
25         0 -1 rlineto
26         -0.35 1 rlineto
27         0.7 0 rlineto
28         -0.35 -1 rlineto
29         closepath
30         fill
31         grestore
32         ")
33     } #}})
34
35 \score {
36   {
37     \jet e'2^\markup {\fontsize #-5 {[jet]}} #8
38     \jet c'4 #3
39     \stemDown \jet b'8 #1.5
40     \jet d'8 #1.5
41   }
42
43   \layout {
44     \context {
45       \Score proportionalNotationDuration = #(ly:make-moment 1/10)
46       \override SpacingSpanner.uniform-stretching = ##t
47     }
48   }
49 }

```

[Table of Contents](#)

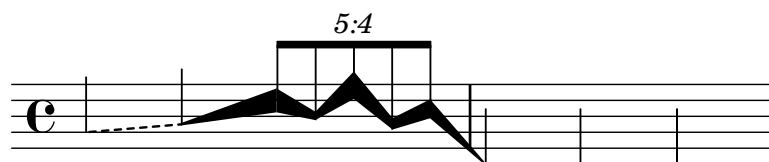

```

31 0.7 0 rlineto
32 -0.35 -1 rlineto
33 closepath
34 fill
35 grestore
36 ")
37         } #})
38
39
40 \score {
41   {
42     \jet e'2^\markup {\fontsize #-5 {[jet]}} #5 #8
43     \jet c'4 #1 #2
44     \stemDown
45     \jet b'8 #4 #1.5
46     \jet d'8 #6 #1.5
47   }
48
49   \layout {
50     \context {
51       \Score proportionalNotationDuration = #(ly:make-moment 1/10)
52       \override SpacingSpanner.uniform-stretching = ##t
53     }
54   }
55 }

```

[Table of Contents](#)

5.3 Line as a Notehead



5.3.1 Description

These functions replace an ordinary notehead with a dashed or a continuous line. For the continuous line, it is possible to adjust the beginning and ending thicknesses.

5.3.2 Grammar

```
\dashedLineNotehead NOTE1 PITCH #x-dist
\modularLineNotehead NOTE1 PITCH #beginningThick #endingThick #x-dist
```

NB

1. NOTE1 specifies with which note the line starts. If necessary, the duration must be set, as well.
2. PITCH specifies with which pitch the line ends. Enter only the pitch; this information is used to determine the angle of the line, and it has no effect in displaying the rhythm.
3. x-dist specifies how long the line is.
4. beginningThick (for modularLineNotehead only) specifies how thick the beginning part of the line should be. #15 gives a thin line, similar to the \dashedLineNotehead line. #100 is as thick as a space between two neighboring lines of a staff.
5. endingThick (for modularLineNotehead only) specifies how thick the ending part of the line should be. #15 gives a thin line, similar to the \dashedLineNotehead line. #100 is as thick as a space between two neighboring lines of a staff.

5.3.3 Code

```
1
2 % See the entry on "Noteheadless" for its code;
3 % it is required for the snippet to run properly.
4
5 dashedLineNotehead =
6 #(define-music-function
7   (beginning end x-distance) (ly:music? ly:music? number?)
8   (let*
9     (
10      (p1 (ly:music-property beginning 'pitch))
11      (p2 (ly:music-property end 'pitch))
12      (steps
13        (-
14          (+ (* 7 (ly:pitch-octave p2)) (ly:pitch-notename p2))
15          (+ (* 7 (ly:pitch-octave p1)) (ly:pitch-notename p1))
```



```

16      )
17    )
18  )
19  #{
20    {
21
22      \once \override Voice.NoteHead.stencil = #ly:text-interface::print
23      \once \override Voice.NoteHead.stem-attachment = #'(0 . 0)
24      \once \override Staff.LedgerLineSpanner.stencil = ##f
25      \once \override Voice.NoteHead.text = \markup          {
26        % \translate #(cons 0 0)
27        \postscript
28        #(string-append
29          "newpath 1 setlinecap
30            0.15 setlinewidth
31            0 0 moveto
32            [.4 .4 .4 .4] 3 setdash "
33            (number->string x-distance) " " (number->string (* steps 0.5))
34            " rlineto stroke"
35          )
36        }
37        #beginning
38        \revert Voice.NoteHead.stencil
39        \revert Staff.LedgerLineSpanner.stencil
40      }
41    #})
42  )
43
44
45  modularLineNotehead =
46  #(define-music-function
47    (beginning end beginningThickness endingThickness x-distance)
48    (ly:music? ly:music? number? number? number?)
49    (let*
50      (
51        (p1 (ly:music-property beginning 'pitch))
52        (p2 (ly:music-property end 'pitch))
53        (steps
54          (-
55            (+ (* 7 (ly:pitch-octave p2)) (ly:pitch-notename p2))
56            (+ (* 7 (ly:pitch-octave p1)) (ly:pitch-notename p1))
57          )
58        )
59      )
60    #{
61      {
62
63        \once \override Voice.NoteHead.stencil = #ly:text-interface::print

```

```

64     \once \override Voice.NoteHead.stem-attachment = #'(0 . 0)
65     \once \override Voice.LedgerLineSpanner.transparent = ##t
66     \once \override Voice.NoteHead.text = \markup          {
67         \postscript
68         #(string-append
69             "newpath 1 setlinecap 0.1 setlinewidth -0.05 0 moveto 0 "
70             (number->string (* beginningThickness 0.005)) " rlineto "
71             (number->string x-distance) " "
72             (number->string (+ (- (* endingThickness 0.005)
73                                   (* beginningThickness 0.005) )
74                                   (* steps 0.5)))
75             " rlineto 0 "
76             (number->string (* endingThickness -0.01)) " rlineto "
77             (number->string (* -1 x-distance)) " "
78             (number->string (- (* endingThickness 0.005)
79                                   (* beginningThickness 0.005)
80                                   (* steps 0.5)))
81             " rlineto
82             closepath
83             fill"
84         )
85     }
86     #beginning
87     \revert Voice.NoteHead.stencil
88     \revert Staff.LedgerLineSpanner.stencil
89 }
90 #}})
91 )
92
93
94 \score {
95 {
96     \omit Staff.Clef
97     \dashedLineNotehead g'4 a' #6
98     \modularLineNotehead a' d'' #15 #150 #6
99     \override TupletNumber.text = #tuplet-number::calc-fraction-text
100
101     \stemUp \tuplet 5/4 {
102         \modularLineNotehead d''8 b' #150 #50 #2.5
103         \modularLineNotehead b' f'' #50 #175 #2.5
104         \modularLineNotehead f'' a' #175 #70 #2.5
105         \modularLineNotehead a' c'' #70 #120 #2.5
106         \modularLineNotehead c'' c' #120 #15 #3.5
107     }
108     |
109     \modularLineNotehead c'4 c' #15 #15 #12
110     \noteheadless c'
111     \dashedLineNotehead c' c' #5

```

```
112     }
113
114     \layout {
115         \context {
116             \Score proportionalNotationDuration = #(ly:make-moment 1/10)
117             \override SpacingSpanner.uniform-stretching = ##t
118         }
119     }
120 }
121
122
```

5.3.4 Discussion

See [Prescriptive Notation for String Instruments](#) for a possible use of this notehead.

[Table of Contents](#)

5.4 Line as a Notehead 2



5.4.1 Description

These functions replace ordinary noteheads with a dashed or a continuous line. However, unlike the [First Version](#), these functions use `\glissando` as the basis for drawing the line.

5.4.2 Grammar

```
\lineNotehead #THICKNESS NOTE
\lineNoteheadOn #THICKNESS STARTING_NOTE NOTES...
\lineNoteheadOff ARRIVING_NOTE
\lineNoteheadWithRhythm #THICKNESS NOTE
\lineNoteheadWithRhythmOn #THICKNESS STARTING_NOTE NOTES...
\lineNoteheadWithRhythmOff ARRIVING_NOTE
```

NB

1. `\lineNotehead` only shows the line on the staff.
2. `\lineNoteheadWithRhythm` retains the rhythmic information.
3. `\lineNotehead` and `\lineNoteheadWithRhythm` applies the line from one note to another, without the line spanning multiple notes.
4. If the line must span over more than a note, use `\lineNoteheadOn` or `\lineNoteheadWithRhythmOn`.
5. In order to exit the line-as-a-notehead mode, use `\lineNoteheadOff` for both `\lineNotehead` and `\lineNoteheadWithRhythm`. In case the notehead must be disguised at the arrival, you may reduce the font size of the Notehead very drastically. See the Code for an example of this.
6. When using `\lineNoteheadWithRhythm` and `\lineNoteheadWithRhythmOn`, cautions must be paid to the placements of the augmentation dots and the intermediate stems. In the Code, I use:
`\once \override Voice.Dots.extra-offset = #'(0 . -1)`
 And place this *before* the `\lineNoteheadWithRhythmOn`.

5.4.3 Code

```
1 \version "2.24.4"
2
```

```

3 % revised on January 25 2025
4
5 lineNotehead =
6 #(define-music-function (thickness note) (number? ly:music? )
7   #{
8     \once \override NoteHead.stencil = #ly:text-interface::print
9     \once \override NoteHead.text = \markup{ \char ##x200A }
10    \once \override Dots.stencil = ##f
11    \once \override Glissando.breakable = ##t
12    \once \override Glissando.after-line-breaking = ##t
13    \once \override Glissando.thickness = #thickness
14    \once \override Glissando.bound-details =
15    #'(
16      (left (padding . 0))
17      (right (padding . 0))
18    )
19    #note
20    \glissando
21
22  #})
23
24 lineNoteheadOn =
25 #(define-music-function (thickness note) (number? ly:music?)
26   #{
27     \override Stem.stencil = ##f
28     \override Flag.stencil = ##f
29     \override TupletBracket.stencil = ##f
30     \override TupletNumber.stencil = ##f
31     \override Beam.stencil = ##f
32     \override NoteHead.stencil = #ly:text-interface::print
33     \override NoteHead.text = \markup{ \char ##x200A }
34     \override Dots.stencil = ##f
35     \override Glissando.breakable = ##t
36     \override Glissando.after-line-breaking = ##t
37     \override Glissando.thickness = #thickness
38     \override Glissando.bound-details =
39     #'(
40       (left (padding . 0))
41       (right (padding . 0))
42     )
43     #note
44     \glissando
45     \override NoteColumn.glissando-skip = ##t
46   #})
47
48
49 lineNoteheadWithRhythm =
50 #(define-music-function (thickness note) (number? ly:music?)

```

```

51   #{
52   \once \override NoteHead.stencil = #ly:text-interface::print
53   \once \override NoteHead.text = \markup{ \char ##x200A }
54   \once \override Glissando.breakable = ##t
55   \once \override Glissando.after-line-breaking = ##t
56   \once \override Glissando.thickness = #thickness
57   \once \override Glissando.bound-details =
58   #'(
59     (left (padding . 0))
60     (right (padding . 0))
61   )
62   #note
63   \glissando
64
65   #})
66
67   lineNoteheadWithRhythmOn =
68   #(define-music-function (thickness note) (number? ly:music?)
69     #{
70       \override NoteHead.stencil = #ly:text-interface::print
71       \override NoteHead.text = \markup{ \char ##x200A }
72       \override Glissando.breakable = ##t
73       \override Glissando.after-line-breaking = ##t
74       \override Glissando.thickness = #thickness
75       \override Glissando.bound-details =
76       #'(
77         (left (padding . 0))
78         (right (padding . 0))
79       )
80       #note
81       \glissando
82       \override NoteColumn.glissando-skip = ##t
83     #})
84
85
86   lineNoteheadOff =
87   {
88     \revert Stem.stencil
89     \revert Flag.stencil
90     \revert Beam.stencil
91     \revert NoteHead.stencil
92     \revert Dots.stencil
93     \revert Glissando.breakable
94     \revert Glissando.after-line-breaking
95     \revert Glissando.thickness
96     \revert Glissando.bound-details
97     \revert NoteColumn.glissando-skip
98     \revert TupletBracket.stencil

```

```

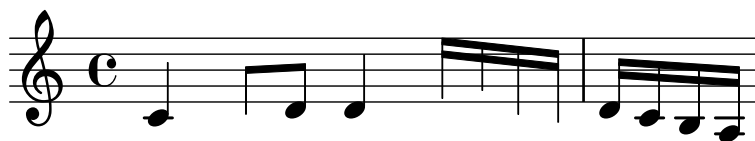
99  \revert TupletNumber.stencil
100 \revert Beam.stencil
101 }
102
103
104 {
105
106  \lineNotehead #1 e'1
107  \lineNoteheadOn #3
108  e''1
109  b'1
110  \lineNoteheadOff
111  \lineNotehead #5
112  e'1
113  \lineNotehead #7
114  e''1
115  \lineNoteheadOn #9 e'4
116  e''4. e'8
117  \lineNoteheadOff
118  \omit Stem
119  e''4
120
121 }
122
123 \score {
124   {
125     \time 3/4
126     \once \override Voice.Dots.extra-offset = #'(0 . -1)
127     \lineNoteheadWithRhythm #5 e'4.
128     \stemDown
129     \lineNoteheadWithRhythmOn #5
130
131     g'8
132     b'4
133     \lineNoteheadOff
134     \lineNoteheadWithRhythmOn #5
135     g''4
136     f''4
137     e''4
138     d''4
139     c''4
140     \lineNoteheadOff
141     \once \override NoteHead.font-size = #-30
142     b'4
143   }
144   \layout {
145     \context{
146       \Score proportionalNotationDuration = #(ly:make-moment 1/8)

```

147 }
148 }
149 }

[Table of Contents](#)

5.5 Noteheadless



5.5.1 Description

This snippet is hardly my own idea, as I largely quoted this technique from one of the snippets available on LSR.² However, I list it here because:

1. it took a while for me to find the workaround for maintaining the musical spacing as a result of omitting noteheads. It is worth noting that because merely disabling `NoteHead.stencil` will render the spacing to be squished, the approach of specifying `##t` for `NoteHead.transparent` (which itself will *not* eliminate the ledger lines) then `##t` for `NoteHead.no-ledgers` is effective in maintaining the general spacing.
2. I use this in conjunction with other notehead alterations, e.g. [Line as a notehead](#).

5.5.2 Grammar

```
\noteheadless NOTE
\noteheadlessOn NOTE
\noteheadlessOff
```

NB

1. `\noteheadless` affects only one note immediately following.
2. For a group of notes, use `\noteheadlessOn` to toggle on the function. `\noteheadlessOff` will toggle off the function.

5.5.3 Code

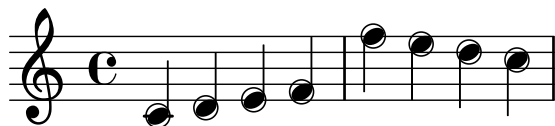
```
1
2 %% Inspired by:
3 %% http://lsr.di.unimi.it/LSR/Item?id=796
4
5
6 noteheadless = {
7   \once \override Voice.NoteHead.transparent = ##t
8   \once \override Voice.NoteHead.no-ledgers = ##t
9 }
10
11 noteheadlessOn = {
12   \override Voice.NoteHead.transparent = ##t
13   \override Voice.NoteHead.no-ledgers = ##t
14 }
15 noteheadlessOff = {
```

2. See: <http://lsr.di.unimi.it/LSR/Item?id=796>

```
16    \revert Voice.NoteHead.transparent
17    \revert Voice.NoteHead.no-ledgers
18  }
19
20
21  {
22    c'4 \noteheadless c'8 d' d'4
23    \noteheadlessOn e'16 f' c' b |
24    \noteheadlessOff d' c' b a
25  }
26
```

[Table of Contents](#)

5.6 Slap Tongue, Type A



5.6.1 Description

In my music, I use encircled noteheads to denote slap tongues. Type A, encircled filled notehead, is used for a slap tongue with a regular note immediately following.

5.6.2 Grammar

`\slapA NOTE`

NB It only affects one note, owing to the `\once \override` functions within the code.

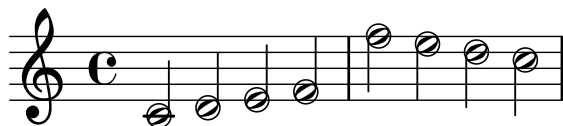
5.6.3 Code

```

1  slapA = #(define-music-function (note) (ly:music?)
2      #{ \once \override Voice.NoteHead.stencil =
3          #ly:text-interface::print
4          \once \override Voice.NoteHead.text =
5          \markup {
6              \concat {
7                  \musicglyph "noteheads.s2"
8                  \postscript "newpath
9                      -0.675 0.025 0.75 0 360 arc
10                     closepath stroke"
11              }
12          }
13      $note #})
14
15  {
16      \slapA c'4 \slapA d' \slapA e' \slapA f'
17      \slapA f'' \slapA e'' \slapA d'' \slapA c''
18  }
19
```

[Table of Contents](#)

5.7 Slap Tongue, Type B



5.7.1 Description

In my music, I use encircled noteheads to denote slap tongues. Type B, encircled hollow notehead, is used for a slap tongue with an air sound immediately following.

5.7.2 Grammar

`\SlapB NOTE`

NB It only affects one note, owing to the `\once \override` functions within the code.

5.7.3 Code

```

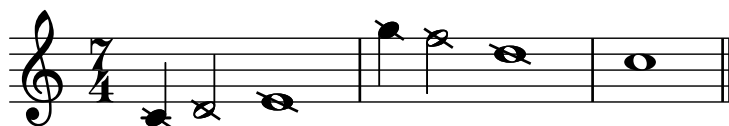
1  slapB = #(define-music-function (note) (ly:music?)
2      #{ \once \override Voice.NoteHead.stencil =
3          #ly:text-interface::print
4          \once \override Voice.NoteHead.text =
5          \markup {
6              \concat {
7                  \musicglyph "noteheads.s1"
8                  \postscript "newpath
9                      -0.675 0.025 0.75 0 360 arc
10                     closepath stroke"
11              }
12          }
13      $note #})
14  {
15      \SlapB c'4 \SlapB d' \SlapB e' \SlapB f'
16      \SlapB f'' \SlapB e'' \SlapB d'' \SlapB c''
17  }
18
```

5.7.4 Discussion

As the musical example shows, when the Type B Slap Tongue notehead is applied to a quarter note, it could invite confusion in terms of rhythm. As a slap tongue itself is a short sound, I only use the slap tongue noteheads on eighth notes or shorter note durations.

[Table of Contents](#)

5.8 Slashed Notehead



5.8.1 Description

Noteheads with backslashes applied.³ I use this notehead to indicate, for example, notes on the piano whose strings are prepared, thus producing pitch/sound different from what is expected normally.

5.8.2 Grammar

```
\slashNote NOTE
\slashNoteOn NOTE
\slashNoteOff
```

NB `\slashNote` only affects one note, owing to the `\once` `\override` functions within the code. For a group of notes to have slashes applied, use `\slashNoteOn`. `\slashNoteOff` cancels the application.

5.8.3 Code

```
1
2 % Inspired by the code provided by Jean Abou Samra
3 % https://lists.gnu.org/archive/html/lilypond-user/2022-11/msg00333.html
4
5 slashNote =
6 \once \override Voice.NoteHead.stencil =
7 #(grob-transformer
8   'stencil
9   (lambda (grob original)
10     (let* ((added-markup
11             #{
12               \markup \general-align #Y #CENTER
13               #(case (ly:grob-property grob 'duration-log)
14                 ((0) #{ \markup \concat {
15                       \musicglyph "noteheads.s0"
16                       \postscript
17                       "gsave
18                       0.17 setlinewidth
19                       -2.3 0.6 moveto
20                       0.3 -0.6 lineto
21                       stroke
22                       grestore"
23                       } #})
```

3. The code provided by Jean Abou Samra in the following discussion thread on lilypond-user was very helpful in creating this code: <https://lists.gnu.org/archive/html/lilypond-user/2022-11/msg00333.html>

```

24
25      ((1) #{ \markup \concat {
26        \musicglyph "noteheads.s1"
27        \postscript
28        "gsave
29          0.17 setlinewidth
30          -1.5 0.6 moveto
31          0.3 -0.6 lineto
32          stroke
33          grestore"
34        } #})
35
36      ((2) #{ \markup \concat {
37        \musicglyph "noteheads.s2"
38        \postscript
39        "gsave
40          0.17 setlinewidth
41          -1.5 0.6 moveto
42          0.3 -0.6 lineto
43          stroke
44          grestore"
45        } #}))
46    #})
47    (added-stencil (grob-interpret-markup grob added-markup)))
48  (if (ly:stencil? original)
49      (ly:stencil-add original added-stencil)
50      added-stencil)))
51
52
53
54 slashNoteOn =
55 \override Voice.NoteHead.stencil =
56 #(grob-transformer
57   'stencil
58   (lambda (grob original)
59     (let* ((added-markup
60             #{
61               \markup \general-align #Y #CENTER
62               #(case (ly:grob-property grob 'duration-log)
63                 ((0) #{ \markup \concat {
64                   \musicglyph "noteheads.s0"
65                   \postscript
66                   "gsave
67                     0.17 setlinewidth
68                     -2.3 0.6 moveto
69                     0.3 -0.6 lineto
70                     stroke
71                     grestore"

```

```

72         } #})
73     ((1) #{ \markup \concat {
74         \musicglyph "noteheads.s1"
75         \postscript
76         "gsave
77         0.17 setlinewidth
78         -1.5 0.6 moveto
79         0.3 -0.6 lineto
80         stroke
81         grestore"
82         } #})
83     ((2) #{ \markup \concat {
84         \musicglyph "noteheads.s2"
85         \postscript
86         "gsave
87         0.17 setlinewidth
88         -1.5 0.6 moveto
89         0.3 -0.6 lineto
90         stroke
91         grestore"
92         } #}))
93     #})
94     (added-stencil (grob-interpret-markup grob added-markup)))
95     (if (ly:stencil? original)
96         (ly:stencil-add original added-stencil)
97         added-stencil)))
98
99
100 slashNoteOff = \revert Voice.NoteHead.stencil
101
102 {
103     \time 7/4
104     \slashNote c'4
105     \slashNote d'2
106     \slashNote e'1
107     \slashNoteOn g''4 f''2 d''1
108     \slashNoteOff c''1 \bar "||"
109 }

```

[Table of Contents](#)

5.9 Square Notehead



5.9.1 Description

Filled and hollow square noteheads.

5.9.2 Grammar

```
\squareHollowNotehead NOTE
\squareHollowNoteheadOn NOTES
\squareHollowNoteheadOff
\squareFilledNotehead NOTE
\squareFilledNoteheadOn NOTES
\squareFilledNoteheadOff
```

```
\slashNoteOn NOTE
\slashNoteOff
```

NB `\squareHollowNotehead` and `\squareFilledNotehead` only affect one note, owing to the `\once \override` functions within the code. For a group of notes, use `\squareHollowNoteheadOn` or `\squareFilledNoteheadOn`. `\squareHollowNoteheadOff` and `\squareFilledNoteheadOff` cancel the application.

5.9.3 Code

```
1 \version "2.24.4"
2
3 % See also: https://lsr.di.unimi.it/LSR/Item?id=516
4 % UPDATED June 13 2025
5
6 squareHollowNoteheadDesign =
7 #(ly:make-stencil '(path 0.15 (moveto 0.0 0.425
8                               rlineto 1.2 0
9                               rlineto 0 -0.875
10                              rlineto -1.2 0
11                              closepath)
12                               )
13      (cons -0.15 1.275)
14      (cons -1 1))
15
16 squareHollowNotehead =
17 #(define-music-function (note) (ly:music?)
18   #{\once \override Voice.NoteHead.stencil =
19     \squareHollowNoteheadDesign $note #})
```



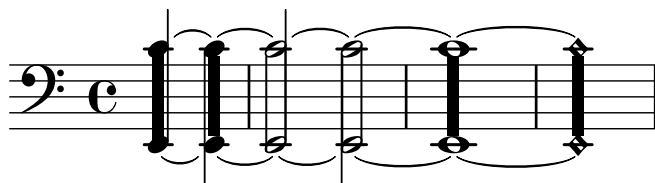
```

20
21 squareHollowNoteheadOn =
22 #(define-music-function (note) (ly:music?)
23   #{\override Voice.NoteHead.stencil =
24     \squareHollowNoteheadDesign $note #})
25
26 squareHollowNoteheadOff = \revert Voice.NoteHead.stencil
27
28 squareFilledNoteheadDesign =
29 #(ly:make-stencil '(path 0.15 (moveto 0.0 0.425
30                               rlineto 1.2 0
31                               rlineto 0 -0.875
32                               rlineto -1.2 0
33                               closepath)
34
35                               round
36                               round
37                               #t)
38   (cons -0.15 1.275)
39   (cons -1 0))
40
41 squareFilledNotehead =
42 #(define-music-function (note) (ly:music?)
43   #{\once \override Voice.NoteHead.stencil =
44     \squareFilledNoteheadDesign $note #})
45 squareFilledNoteheadOn =
46 #(define-music-function (note) (ly:music?)
47   #{\override Voice.NoteHead.stencil =
48     \squareFilledNoteheadDesign $note #})
49
50 squareFilledNoteheadOff = \revert Voice.NoteHead.stencil
51
52 {
53   \squareHollowNotehead c'8
54   \squareHollowNoteheadOn d' e' f'
55   \squareHollowNoteheadOff
56   \squareFilledNotehead c'8
57   \squareFilledNoteheadOn d' e' f'
58   \squareFilledNoteheadOff
59   \squareHollowNotehead a''8
60   \squareHollowNoteheadOn g'' f'' e''
61   \squareHollowNoteheadOff
62   \squareFilledNotehead a''8
63   \squareFilledNoteheadOn g'' f'' e''
64   \squareFilledNoteheadOff
65 }

```

[Table of Contents](#)

5.10 Tone Cluster



5.10.1 Description

Inspired by the tone cluster notation of Henry Cowell and others. See [Discussion](#).

5.10.2 Grammar

```
\toneClusterBar NOTE1 NOTE2 yOffset yLengthAdjust
\toneClusterBarHollow NOTE1 NOTE2 yOffset yLengthAdjust
\toneClusterBarWhole NOTE1 NOTE2 yOffset yLengthAdjust
```

NB

1. The order of pitch boundaries as shown by NOTE1 and NOTE2 does not matter; NOTE1 can be upper or lower pitch boundary, and vice versa for NOTE2. See [Code](#).
2. `yOffset` indicates where the upper part of the cluster sign begins. When set to #0, it starts right at the top line of the ordinary 5-line staff. Each positive/negative integer will bring the beginning point up/down by a space of two neighboring lines of the staff.
3. `yLengthAdjust` indicates any value by which the cluster bar may be extended or reduced. When set to #0, the cluster bar will be as long as the distance between the lower boundary of the upper notehead and upper boundary of the lower notehead. Each positive/negative integer will add/reduce the length of the bar by a space of two neighboring lines of the staff.

For this reason, when the tone cluster sign is applied to a quarter-note dyad, you may wish to set the upper part of the cluster bar right in the middle of the notehead. In the snippet shown, the first cluster's `yOffset` is set to #1. `yLengthAdjust` is also set to #1, meaning that the cluster bar will go down to the center of the lower notehead. The second cluster intentionally shows what happens when the bar only touches the two boundaries of the noteheads.

4. `\toneClusterBarHollow` shows the notation (quite à la Cowell) specifically for hollowed noteheads. Some people may prefer this notation, instead.
5. `\toneClusterBarWhole` is specifically for the tone cluster notation as applied to a whole-note dyad, owing to width being wider than the quarter or half noteheads.
6. These functions may be used in tandem with other noteheads, as well as ties. See [Code](#).

5.10.3 Code

```
1
2 \toneClusterBar =
3 #(define-music-function (note1 note2 yOffset yLengthAdjust)
4   (ly:music? ly:music? number? number?)
5   (let* (
```

```

6      (note1p (ly:music-property note1 'pitch))
7      (note2p (ly:music-property note2 'pitch))
8      (note1pnumber (+ (* 7 (ly:pitch-octave note1p))
9                      (ly:pitch-notename note1p)))
10     (note2pnumber (+ (* 7 (ly:pitch-octave note2p))
11                     (ly:pitch-notename note2p)))
12     (pitchDistance (abs (- note1pnumber note2pnumber)))
13   )
14   #{
15     < #note1
16     #note2 > ^\markup {
17       \postscript
18       #(string-append
19         "gsave
20         newpath
21         0.3 " (number->string (- yOffset 0.5)) " moveto
22         0.7 0 rlineto
23         0 " (number->string (- (* -0.5 pitchDistance)
24                               (- yLengthAdjust 1))) " rlineto
25         -0.7 0 rlineto
26         closepath
27         fill
28         grestore")
29     }
30   #}
31   )
32 )
33
34
35 toneClusterBarHollow =
36 #(define-music-function (note1 note2 yOffset yLengthAdjust)
37   (ly:music? ly:music? number? number?)
38   (let* (
39     (note1p (ly:music-property note1 'pitch))
40     (note2p (ly:music-property note2 'pitch))
41     (note1pnumber (+ (* 7 (ly:pitch-octave note1p))
42                     (ly:pitch-notename note1p)))
43     (note2pnumber (+ (* 7 (ly:pitch-octave note2p))
44                     (ly:pitch-notename note2p)))
45     (pitchDistance (abs (- note1pnumber note2pnumber)))
46   )
47   #{
48     < #note1
49     #note2 > ^\markup {
50       \postscript
51       #(string-append
52         "gsave
53         newpath

```

```

54         0.1 " (number->string (- yOffset 0.5)) " moveto
55         0 " (number->string (- (* -0.5 pitchDistance)
56                               (+ 0.5 yLengthAdjust))) " rlineto
57         0.125 setlinewidth
58         1.3 "(number->string (+ 0.75 (- yOffset 0.5))) " moveto
59         0 " (number->string (- (* -0.5 pitchDistance)
60                               (+ 0.75 yLengthAdjust))) " rlineto
61         stroke
62         grestore")
63     }
64     #}
65 )
66 )
67
68
69 toneClusterBarWhole =
70 # (define-music-function (note1 note2 yOffset yLengthAdjust)
71   (ly:music? ly:music? number? number?)
72   (let* (
73     (note1p (ly:music-property note1 'pitch))
74     (note2p (ly:music-property note2 'pitch))
75     (note1pnumber (+ (* 7 (ly:pitch-octave note1p))
76                      (ly:pitch-notename note1p)))
77     (note2pnumber (+ (* 7 (ly:pitch-octave note2p))
78                      (ly:pitch-notename note2p)))
79     (pitchDistance (abs (- note1pnumber note2pnumber)))
80   )
81   #{
82     < #note1
83     #note2 > ^\markup {
84       \postscript
85       # (string-append
86         "gsave
87         newpath
88         0.125 setlinewidth
89         0.55 " (number->string (- yOffset 0.5)) " moveto
90         0 " (number->string (- (* -0.5 pitchDistance)
91                               (- yLengthAdjust 1))) " rlineto
92         0.75 0 rlineto
93         0 " (number->string (abs (- (* -0.5 pitchDistance)
94                               (- yLengthAdjust 1)))) " rlineto
95         closepath fill
96         grestore")
97     }
98     #}
99   )
100 )
101

```

```

102
103 {
104   \time 4/4
105   \partial 2
106   \clef "F"
107   \stemUp \toneClusterBar c'4~ e,~ #1 #1
108   \stemDown \toneClusterBar e,~ c'4~ #0.5 #0
109   \stemUp \toneClusterBarHollow c'2~ e,~ #0.5 #-0.5
110   \stemDown \toneClusterBarHollow c'2~ e,~ #0.5 #-0.5
111   \toneClusterBarWhole c'1~ e,~ #0.5 #0
112   \toneClusterBar c'1~\harmonic e,~\harmonic #0.5 #0
113 }

```

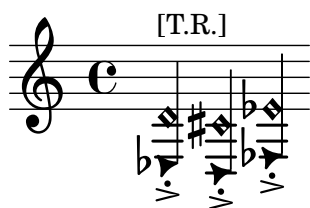
5.10.4 Discussion

There have been some discussions on `lilypond-user` mailing list in the past that readers may consult for further ideas on implementing different types of tone cluster notation:

- <https://lists.gnu.org/archive/html/lilypond-user/2008-10/msg00484.html> (This one in particular lists other notational conventions established by other composers)
- <https://lists.gnu.org/archive/html/lilypond-user/2020-12/msg00130.html>

[Table of Contents](#)

5.11 Tongue Ram (for flute)



5.11.1 Description

Implementation of the tongue ram notation, as described in *The Techniques of Flute Playing* by Carin Levine and Christina Mitropoulos-Bott.⁴

5.11.2 Grammar

`\tgrWithIndication NOTE`

`\tgr NOTE`

NB

1. `\language "english"` needs to be specified.
2. `\tgr` and `\tgrWithIndication` are followed by a pitch to be fingered on the instrument. The code will copy and reproduce a resultant pitch a major seventh down. Use `\tgrWithIndication` for showing the markup with the indication "T.R." (tongue ram). For more details, see: [FluteXpansions](#).

5.11.3 Code

```

1  \tgrWithIndication = #(define-music-function (note1) (ly:music?)
2      (let*
3          (
4              (p1 #{ #(ly:music-deep-copy note1) \harmonic #})
5              (p2 #{ \transpose c df, #(ly:music-property note1 'pitch)#})
6              (d1 (ly:music-property note1 'duration))
7              )
8          #{ < $p1
9              \single \override NoteHead.stencil = #ly:text-interface::print
10             \single \override NoteHead.text =
11             \markup \musicglyph "noteheads.s2triangle"
12             %\single \override Stem.stencil
13             $p2 > $d1 ^\markup {\override #'(font-size . -2) {[T.R.]} } #}
14          ))
15  \tgr = #(define-music-function (note1) (ly:music?)
16      (let*
17          (
18              (p1 #{ #(ly:music-deep-copy note1) \harmonic #})
19              (p2 #{ \transpose c df, #(ly:music-property note1 'pitch)#})
20              (d1 (ly:music-property note1 'duration))
21              )
22          )

```

4. Levine and Mitropoulos-Bott, *The techniques of flute playing = Die Spieltechnik der Flöte*, 28.

```

20      #{ < $p1
21      \single \override NoteHead.stencil = #ly:text-interface::print
22      \single \override NoteHead.text =
23      \markup \musicglyph "noteheads.s2triangle"
24      %\single \override Stem.stencil
25      $p2 > $d1  #}
26  ))
27
28  {\language "english" \tgrWithIndication d'4-.-> \tgr cs'4-.-> \tgr ef'4-.->}

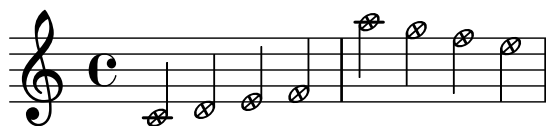
```

5.11.4 Discussion

I want to improve this code so that I can add markups to the note. It is slightly awkward at the moment.

[Table of Contents](#)

5.12 X In A Hollow Notehead



5.12.1 Description

While LilyPond Notation Reference provides an example of an X-in-a-circle notehead, its shape differs from the regular notehead.⁵ This implementation simulates a hollow notehead with which the X notehead is combined.

5.12.2 Grammar

`\cirX NOTE`

5.12.3 Code

```

1 % Stem attachment function inspired by:
2 % https://lsr.di.unimi.it/LSR/Snippet?id=518
3 cirX = #(define-music-function (note) (ly:music?)
4   #{
5     \temporary \override NoteHead.stencil =
6     #ly:text-interface::print
7     \temporary \override NoteHead.text =
8     \markup
9     \translate #'(0.6 . 0)
10    \pad-x #-0.22
11    \rotate #35
12    \scale #'(1 . 0.65)
13    \combine \combine \combine \combine
14    \override #'(thickness . 2)
15    \draw-line #'(0.05 . 0.6)
16    \override #'(thickness . 2)
17    \draw-line #'(-0.05 . -0.6)
18    \override #'(thickness . 2)
19    \draw-line #'(0.6 . 0.1 )
20    \override #'(thickness . 2)
21    \draw-line #'(-0.6 . -0.1 )
22    \draw-circle #0.65 #0.175 ##f
23
24    \temporary \override NoteHead.stem-attachment =
25    #(lambda (grob)
26      (let* ((stem (ly:grob-object grob 'stem))
27             (dir (ly:grob-property stem 'direction UP))
28             (is-up (eqv? dir UP)))
29        (cons dir (if is-up 0.2 -0.2))))

```

5. <https://lilypond.org/doc/v2.24/Documentation/notation/modifying-stencils>


```
30         #note
31         \revert NoteHead.stencil
32         \revert NoteHead.text
33         \revert NoteHead.stem-attachment
34     #})
35 {
36     \cirX c'4 \cirX d' \cirX e' \cirX f'
37     \cirX a''4 \cirX g'' \cirX f'' \cirX e''
38 }
```

[Table of Contents](#)

Chapter 6

Markups

6.1 Conducting Patterns



6.1.1 Description

Conducting patterns. While there are several examples of conducting patterns available on LSR,¹ the conducting shapes in my implementation are not affected by the horizontal length of given durations.

6.1.2 Grammar

```
NOTE \condOne
NOTE \condTwoA
NOTE \condTwoB
NOTE \condThree
NOTE \condDoubleTwoA
NOTE \condDoubleTwoB
NOTE \condDoubleThree
```

6.1.3 Code

```
1
2 condOnePattern =
3 #'((moveto 0.25 1.75)
4   (rlineto 0 -1.75))
5
```

1. See: <https://lsr.di.unimi.it/LSR/Item?id=523> and <https://lsr.di.unimi.it/LSR/Item?id=259>

```

6  condTwoPatternA =
7  #'((moveto 0.25 1.75)
8      (rlineto 0 -1.75)
9      (rlineto 2 0)
10     (rlineto 0 1.75))
11
12  condDoubleTwoPatternA =
13  #'((moveto 0.25 1.75)
14      (rlineto 0 -1.75)
15      (rlineto 2 0)
16      (rlineto 0 1.75)
17      (moveto 0.65 1.75)
18      (rlineto 0 -1.35)
19      (rlineto 1.2 0)
20      (rlineto 0 1.35))
21
22  condTwoPatternB =
23  #'((moveto 0.25 1.75)
24      (rlineto 0 -1.75)
25      (rlineto 1.25 1.75))
26
27  condDoubleTwoPatternB =
28  #'((moveto 0.25 1.75)
29      (rlineto 0 -1.75)
30      (rlineto 1.25 1.75)
31      (moveto 0.6 1.75)
32      (rlineto 0 -0.7)
33      (rlineto 0.5 0.7))
34
35  condThreePattern =
36  #'((moveto 1.15 1.75)
37      (rlineto -1 -1.75)
38      (rlineto 2 0)
39      (closepath))
40
41  condDoubleThreePattern =
42  #'((moveto 1.15 1.75)
43      (rlineto -1 -1.75)
44      (rlineto 2 0)
45      (closepath)
46      (moveto 1.15 1.05)
47      (rlineto -0.385 -0.7)
48      (rlineto 0.75 0)
49      (closepath))
50
51
52  condOne = ^\markup {
53      \override #'(line-join-style . round)

```

```

54   \path #0.25 #condOnePattern
55 }
56
57 condTwoA = ^\markup {
58   \override #'(line-join-style . round)
59   \path #0.25 #condTwoPatternA
60 }
61 condTwoB = ^\markup {
62   \override #'(line-join-style . round)
63   \path #0.25 #condTwoPatternB
64 }
65 condDoubleTwoA = ^\markup {
66   \override #'(line-join-style . round)
67   \path #0.25 #condDoubleTwoPatternA
68 }
69
70 condDoubleTwoB = ^\markup {
71   \override #'(line-join-style . round)
72   \path #0.25 #condDoubleTwoPatternB
73 }
74
75 condThree = ^\markup {
76   \override #'(line-join-style . round)
77   \path #0.25 #condThreePattern
78 }
79
80 condDoubleThree = ^\markup {
81   \override #'(line-join-style . round)
82   \path #0.25 #condDoubleThreePattern
83 }
84
85 %% Source inspired by
86 %% and adapted from: http://lsr.di.unimi.it/LSR/Item?id=629
87 spacerVoice = \new Voice {
88   \override MultiMeasureRest.transparent = ##t
89   \override MultiMeasureRest.minimum-length = #14
90   R16*5
91 }
92
93
94 \score {
95   {
96     \time 5/8
97     b'4 \condTwoA b'4. \condThree \bar "||"
98     b'4 \condTwoB b'4. \condThree \bar "||"
99     b'8 \condOne b'4 \condTwoA b'4 \condTwoA \bar "||"
100    \time 5/16
101    << {b'8 \condDoubleTwoA b'8. \condDoubleThree}

```

```
102         \spacerVoice >> \bar "||"  
103     << {b'8 \condDoubleTwoB b'8. \condDoubleThree}  
104         \spacerVoice >> \bar "||"  
105     }  
106  
107 }  
108
```

[Table of Contents](#)

6.2 Mute Sign



6.2.1 Description

Implementation of the mute sign, used to indicate that vibrating strings must be dampened at a specified moment. Its provenance can be traced back to Carlos Salzedo's *Modern Study of the Harp*.²

6.2.2 Grammar

NOTE/REST[^]\mutesign

6.2.3 Code

```

1  mutesign = \markup {
2    \translate #'(0.5 . 0)
3    \postscript
4
5    "newpath
6    0.2 setlinewidth
7    1 setlinecap
8    0 0 moveto
9    0 2.5 rlineto
10   -1.25 1.25 moveto
11   2.5 0 rlineto
12   stroke
13   newpath
14   0 1.25 0.85 0 360 arc
15   stroke"
16
17   { c'2. r4^\mutesign }
18
```

[Table of Contents](#)

2. Carlos Salzedo, *L'étude moderne de la harpe... Modern study of the harp* (New York - Boston, G. Schirmer, 1921), 19.

Chapter 7

Rhythm

7.1 Incomplete Tuplet Bracket for Irrational Time Signatures



7.1.1 Description

This entry implements the irrational time signatures¹ as seen on LSR.² Concerning the irrational time signatures, in her *Behind Bars: the Definitive Guide to Music Notation*, Elaine Gould suggests the use of denominator as any division of the semibreve/whole note..³ However, in these pages there has not been a mention of the use of tuplet brackets while the non-conventional time signature is in place. There are examples, such as *Asyla* for large orchestra by Thomas Adès,⁴ where tuplet brackets are placed atop "incomplete" tuplets.

While it is still prudent to spend a paragraph explaining the nature of the irrational time signatures in the preface, my preference has also been to utilize incomplete tuplet brackets, in order to allow the reading of the rhythm consistent and smooth from bars with ordinary time signatures. It is also helpful to have the brackets shown in cases of compound time signatures that use irrational time signatures in part (see the first measure of the example).

7.1.2 Grammar

```
\incompleteTupletBracket \tuplet ...  
\incompleteSmallTupletBracket \tuplet ...
```

NB

1. For incomplete tuplets with two or more notes, use `\incompleteTupletBracket`.

1. See [Chapter Time Signatures](#) for discussion on the variants of the irrational/fractional time signatures.
2. <https://lsr.di.unimi.it/LSR/Snippet?id=552>
3. Elaine Gould, *Behind bars : the definitive guide to music notation* (London: Faber Music, 2011), 180–181, Book.
4. Thomas Adès, *Asyla : for large orchestra* (Faber Music, 1997).

2. For incomplete tuplets with one note, use `\incompleteSmallTupletBracket`. This was created specifically to ensure that the brackets appear properly in tight space that one-note tuplet customarily gives.

7.1.3 Code

```

1  \version "2.24.4"
2
3  %% "suppressWarning" function comes from:
4  %% http://lsr.di.unimi.it/LSR/Item?id=552
5
6  % Warnings may be suppressed using 'ly:expect-warning'
7  % Or use the here defined 'suppressWarning'-function, working since 2.20.
8
9  suppressWarning =
10 # (define-void-function (amount message) (number? string?)
11     (for-each
12         (lambda (warning)
13             (ly:expect-warning message))
14         (iota amount 1 1)))
15
16 \suppressWarning 3 "strange time signature found"
17
18 incompleteTupletBracket = {
19     \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
20     \once \override Voice.TupletBracket.bracket-visibility = ##t
21
22 }
23 incompleteSmallTupletBracket = {
24     \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
25     \once \override Voice.TupletBracket.bracket-visibility = ##t
26     \once \override Voice.TupletNumber.X-offset =
27     # (lambda (grob)
28         (if (= UP (ly:grob-property grob 'direction))
29             2.2
30             1.2))
31
32     \once \override Voice.TupletBracket.shorten-pair =
33     # (lambda (grob)
34         (if (= UP (ly:grob-property grob 'direction))
35             '(-0.7 . 0.15)
36             '(-0.3 . 0.8)))
37     \once \override Voice.TupletBracket.X-positions =
38     # (lambda (grob)
39         (if (= UP (ly:grob-property grob 'direction))
40             '(1.8 . 3)
41             '(0.3 . 2.7)))
42 }

```



```

43
44
45 {
46   \compoundMeter #'((2 4) (4 12))
47   f'4 f'
48   \tuplet 3/2 {g'8[ g' g']}
49   \incompleteSmallTupletBracket
50   \tuplet 3/2 {a'8 }|
51
52   \time 4/20
53   \incompleteTupletBracket
54   \tuplet 5/4 {b'16[ b' b' b']} |
55   \time 4/12
56   \tuplet 3/2 {c''8[ g' e']}
57   \incompleteSmallTupletBracket
58   \tuplet 3/2 {c'8} |
59   \tuplet 3/2 {c'8[ e' g']}
60   \incompleteSmallTupletBracket
61   \tuplet 3/2 {c''8} |
62 }

```

7.1.4 Discussion

In the preceding code, I have opted to notate the tuplets within the bars with irrational time signatures in an ordinary manner, using `\tuplet`. This is to ensure that the incomplete tuplet bracket appears. Compare this with the quoted LSR No. 552, which has a different way of reducing the note duration in order to fit them into the bar with irrational time signature. Observe the way duration is multiplied by fractions, e.g. Line 6.

```

1 {
2   \time 4/4
3   \tempo 4 = 60
4   fis4 fis fis fis
5   \time 2/6
6   g4*2/3 g |
7   g4*2/3 g |
8   \time 4/5
9   as4*4/5 as as as8*4/5 g |
10  \tuplet 3/2 { as4*4/5 as as } as4*4/5 as8*4/5 g |
11  \time 3/7
12  fis4*4/7 fis fis |
13  fis4*4/7 fis fis |
14 }

```

[Table of Contents](#)

7.2 Metric Modulation Equation (Regular Flag)

The image shows two musical staves illustrating the Metric Modulation Equation (Regular Flag).
 The first staff is in 2/4 time, starting with a tempo of 60. It features a 5-measure phrase followed by a 3-measure phrase, with a tempo change to 75. Above the staff, there are two equations: $\frac{5}{4} = \frac{3}{4}$ and $\frac{5:4}{4:3} = \frac{3}{4}$.
 The second staff is in 4/4 time, starting with a tempo of 83.33. It features a 3-measure phrase followed by a 3-measure phrase, with a tempo change to 6/8. Above the staff, there are two equations: $\frac{5:4}{4:3} = \frac{3}{4}$ and $\frac{5}{4} = \frac{3}{4}$.

7.2.1 Description

This entry implements a metric modulation formula that indicates a note value of one measure being equal to another note value of the subsequent measure. While this notation has existed for a very long time, it was in the twentieth century where composers such as Elliott Carter used them extensively in their works. This code follows the convention commonly seen in Carter's scores, where the formula is accompanied by the right and left arrows to state explicitly that the note value on the left refers to that of the preceding measure, while the one on the right refers to that of the subsequent measure.⁵

Another version with notes with straight flags can be found [here](#).

7.2.2 Grammar

```
\MModEquation
  LEFT_NOTE_DURATION #'((LEFT_TUPLET_1)(LEFT_TUPLET_2))
  RIGHT_NOTE_DURATION #'((RIGHT_TUPLET_1)(RIGHT_TUPLET_2))
  #V_OFFSET #H_OFFSET
\MModEquationBegin
  LEFT_NOTE_DURATION #'((LEFT_TUPLET_1)(LEFT_TUPLET_2))
  RIGHT_NOTE_DURATION #'((RIGHT_TUPLET_1)(RIGHT_TUPLET_2))
  #V_OFFSET
```

NB

5. While it may be beyond the scope of this Cookbook, stating the orientation of the note value clarifies any ambiguities, as the convention in the classical period had the two note values flipped, where the note on the left indicated the note value of the forthcoming measure, and the one on the right, that of the preceding measure. See Arthur Weisberg, *Performing twentieth-century music : a handbook for conductors and instrumentalists* (New Haven: Yale University Press, 1996), 52.

1. `\MModEquation` utilizes LilyPond's `\textEndMark` functions. It is therefore meant to be used at the end of a measure where the metric modulation is about to take place. Then you may use `\tempo` function on LilyPond to indicate the new tempo in the subsequent measure.
2. `\MModEquationBegin`, on the other hand, uses LilyPond's `\textMark` functions, which places texts at the beginning of a measure. It is meant to be used as part of the editing process. Because `\MModEquation` places the formula at the end of a measure, when that measure is at the end of a system, requiring another formula at the beginning of the subsequent measure (i.e. at the beginning of a new system) may be plausible. Using `\MModEquationBegin` may be the solution.
3. For `LEFT_NOTE_DURATION` and `RIGHT_NOTE_DURATION`, use a numerical value *without* `#`. You may add dot(s) to indicate augmentation dot(s).
4. For the variables 2 and 4, you may put up to two instances of tuplet values, in the form of list(s) within a list. At the minimum, you will need to supply an empty list, i.e. `#'()`.

For example, if it is a simple triplet you wish to implement, you would write:

```
#'( ( 3 ) )
```

If you wish to indicate tuplet in ratio, e.g. quintuplet in the span of 4 of the note values, you would create a list with two items, i.e.:

```
#'( ( 5 4 ) )
```

You may set another tuplet instance *above* the initial tuplet, so the following list:

```
#'( ( 4 3 ) ( 5 4 ) )
```

...would produce the tuplet indication at the end of measure 3 in the example shown above.

5. For `#V_OFFSET` and `#H_OFFSET` (only for `\MModEquation`), enter numbers preceded by `#`. These numbers adjust the placement of the formula. When in doubt, start with `#0`.

7.2.3 Code

```
1 \version "2.24.4"
2
3 MModEquation =
4 #(define-music-function
5   (notevalue1 ratio1 notevalue2 ratio2 verticaloffset horizontaloffset)
6   (ly:duration? list? ly:duration? list? number? number?)
7   (let* (
8     (noteone notevalue1)
9     (notetwo notevalue2)
10    (ratioone ratio1)
11    (ratiotwo ratio2)
12    )
13   #{
14     \tweak X-offset #(- horizontaloffset 0.35)
15     \tweak Y-offset #verticaloffset
```

```

16 \textEndMark \markup \right-align {
17 \hspace #-0.5
18 \raise #0
19 \fontsize #-4.5
20 \override #'(self-alignment . LEFT)
21 {
22   {
23     \note { $noteone } #(if (= (ly:duration-log noteone) 4) 1 1 )
24   }
25   {
26     \hspace #(if (< (ly:duration-log noteone) 3) -4.15
27                (+ (* (ly:duration-dot-count noteone) -0.5) -4.5))
28     \concat {
29       \combine
30       \musicglyph "arrowheads.open.OM1"
31       \draw-line #'(2 . 0) \fontsize #-5 \musicglyph "space"
32     }
33   }
34   {
35     \hspace #(cond ((<= (ly:duration-log noteone) 2) 0.75)
36                   ((> (ly:duration-log noteone) 2) 0.25) )
37     \raise #(cond
38              ((= (ly:duration-log noteone) 0) -2)
39              ((and (> (ly:duration-log noteone) 0)
40                  (<= (ly:duration-log noteone) 4) ) -.5)
41              ((> (ly:duration-log noteone) 4)
42               (+ (* (- (ly:duration-log noteone) 5) 0.6) 0))
43            )
44     #(cond
45        ((= (length ratioone) 1)
46         (markup
47          #:line
48          (#:hspace
49           -4.5
50           #:raise
51           2.75
52           (#:center-column
53            (#:concat
54             (#:translate
55              (cons 3.5 0.5)
56              (#:override
57               (cons 'thickness 1.5)
58               (#:translate
59                (cons -3.5 0)
60                (#:draw-line (cons -4.25 0))))))
61          #:hspace
62          -0.1
63          #:override

```

```

64         (cons 'thickness 1.5)
65         (:translate
66         (cons 0 0.5)
67         (:draw-line (cons 0 -0.5))))
68     #:vspace
69     -0.45
70     #:whiteout
71     (:halign
72     1.75
73     (:concat
74     (:fontsize
75     -1
76     (:italic (:fontsize -5 (:musicglyph "space"))
77     #:fontsize
78     -1
79     #:italic
80     (if (= (length (car ratioone)) 2)
81         (begin (string-append
82                 (number->string (car (car ratioone)))
83                 ":"
84                 (number->string (cadr (car ratioone)))) )
85         (number->string (car (car ratioone)))
86         )
87
88     (:italic (:fontsize -5 (:musicglyph "space"))))))))
89
90     ((= (length ratioone) 2)
91     (markup
92     #:line
93     (:hspace
94     -4.5
95     #:raise
96     2.75
97     (:center-column
98     (:concat
99     (:translate
100    (cons 3.5 0.5)
101    (:override
102    (cons 'thickness 1.5)
103    (:translate
104    (cons -3.5 0)
105    (:draw-line (cons -4.25 0))))
106    #:hspace
107    -0.1
108    #:override
109    (cons 'thickness 1.5)
110    (:translate
111    (cons 0 0.5)

```

```

112         (:draw-line (cons 0 -0.5))))
113 #:vspace
114 -0.45
115 #:whiteout
116 (:halign
117 1.75
118 (:concat
119 (:fontsize
120 -1
121 (:italic (:fontsize -5 (:musicglyph "space")))
122 #:fontsize
123 -1
124 #:italic
125 (if (= (length (car ratioone)) 2)
126     (begin (string-append
127             (number->string (car (car ratioone)))
128             ":"
129             (number->string (cadr (car ratioone)))) )
130     (number->string (car (car ratioone)))
131     )
132     (:italic (:fontsize -5 (:musicglyph "space"))))))))
133 #:line
134 (:hspace
135 -4.8
136 #:raise
137 4
138 (:center-column
139 (:concat
140 (:translate
141 (cons 3.5 0.5)
142 (:override
143 (cons 'thickness 1.5)
144 (:translate
145 (cons -3.5 0)
146 (:draw-line (cons -4.25 0))))
147 #:hspace
148 -0.1
149 #:override
150 (cons 'thickness 1.5)
151 (:translate
152 (cons 0 0.5)
153 (:draw-line (cons 0 -0.5))))
154 #:vspace
155 -0.45
156 #:whiteout
157 (:halign
158 1.75
159 (:concat

```

```

160         (:fontsize
161         -1
162         (:italic (:fontsize -5 (:musicglyph "space")))
163         (:fontsize
164         -1
165         (:italic
166         (if (= (length (cadr ratioone)) 2)
167             (begin (string-append
168                     (number->string (car (cadr ratioone)))
169                     ":"
170                     (number->string (cadr (cadr ratioone)))) )
171             (number->string (car (cadr ratioone)))
172             )
173         (:italic (:fontsize -5 (:musicglyph "space"))))))))
174     ))
175 )
176 }
177 }
178 }
179 \tweak self-alignment-X #CENTER
180 \tweak Y-offset #(- verticaloffset 0.4)
181 \tweak X-offset #(- horizontaloffset 0.35)
182 \textEndMark \markup { \fontsize #-3 "="}
183
184 \tweak Y-offset #verticaloffset
185 \tweak self-alignment-X #RIGHT
186 \tweak X-offset #(- horizontaloffset 0.35)
187 \tweak self-alignment-Y -1
188 \textEndMark \markup \left-align {
189   \hspace #1.9
190   \raise #0
191   \fontsize #-4.5
192   \concat {
193     {
194       \hspace #(cond ((<= (ly:duration-log notetwo) 2) 4)
195                     ((> (ly:duration-log notetwo) 2) 3.75) )
196       \raise #(cond
197                 ((= (ly:duration-log notetwo) 0) -2)
198                 ((and (> (ly:duration-log notetwo) 0)
199                      (<= (ly:duration-log notetwo) 4) ) -.5)
200                 ((> (ly:duration-log notetwo) 4)
201                  (+ (* (- (ly:duration-log notetwo) 5) 0.6) 0))
202                 )
203       #(cond
204         ((= (length ratiotwo) 1)
205          (markup
206           #:line
207           (:hspace

```

```

208      -4.5
209      #:raise
210      3.25
211      (#:center-column
212        (#:concat
213          (#:translate
214            (cons 3.5 0.5)
215            (#:override
216              (cons 'thickness 1.5)
217              (#:translate
218                (cons 0 -0.5 )
219                (#:draw-line (cons 0 -0.5)))
220            )
221          #:hspace
222          -0.1
223          #:override
224          (cons 'thickness 1.5)
225          (#:translate
226            (cons -3.5 0)
227            (#:draw-line (cons -4.25 0))))
228      #:vspace
229      -0.35
230      #:whiteout
231      (#:halign
232        1.75
233        (#:concat
234          (#:fontsize
235            -1
236            (#:italic (#:fontsize -5 (#:musicglyph "space"))))
237          #:fontsize
238          -1
239          #:italic
240          (if (= (length (car ratiotwo)) 2)
241            (begin (string-append
242                  (number->string (car (car ratiotwo)))
243                  ":"
244                  (number->string (cadr (car ratiotwo)))) )
245            (number->string (car (car ratiotwo)))
246          )
247          (#:italic
248            (#:fontsize -5
249              (number->string (car (car ratiotwo))))))
250      ((= (length ratiotwo) 2)
251      (markup
252        #:line
253        (#:hspace
254          -4.5
255          #:raise

```



```

256      3.25
257      (:center-column
258        (:concat
259          (:translate
260            (cons 3.5 0.5)
261            (:override
262              (cons 'thickness 1.5)
263
264              (:translate
265                (cons 0 -0.5 )
266                (:draw-line (cons 0 -0.5)))
267              )
268              #:hspace
269              -0.1
270              #:override
271              (cons 'thickness 1.5)
272              (:translate
273                (cons -3.5 0)
274                (:draw-line (cons -4.25 0))))
275              #:vspace
276              -0.35
277              #:whiteout
278              (:halign
279                1.75
280                (:concat
281                  (:fontsize
282                    -1
283                    (:italic (:fontsize -5 (:musicglyph "space"))))
284                    #:fontsize
285                    -1
286                    #:italic
287                    (if (= (length (car ratiotwo)) 2)
288                      (begin (string-append
289                        (number->string (car (car ratiotwo)))
290                        ":"
291                        (number->string (cadr (car ratiotwo)))) )
292                      (number->string (car (car ratiotwo)))
293                      )
294                    (:italic (:fontsize -5 (:musicglyph "space"))))))))
295      #:line
296      (:hspace
297        -4.8
298        #:raise
299        4.5
300        (:center-column
301          (:concat
302            (:translate
303              (cons 3.5 0.5)

```

```

304         (:override
305         (cons 'thickness 1.5)
306         (:translate
307         (cons 0 -0.5 )
308         (:draw-line (cons 0 -0.5)))
309         )
310         #:hspace
311         -0.1
312         #:override
313         (cons 'thickness 1.5)
314         (:translate
315         (cons -3.5 0)
316         (:draw-line (cons -4.25 0)))
317         )
318         #:vspace
319         -0.35
320         #:whiteout
321         (:halign
322         1.75
323         (:concat
324         (:fontsize
325         -1
326         (:italic (:fontsize -5 (:musicglyph "space"))))
327         #:fontsize
328         -1
329         #:italic
330         (if (= (length (cadr ratiotwo)) 2)
331             (begin (string-append
332                     (number->string (car (cadr ratiotwo)))
333                     ":"
334                     (number->string (cadr (cadr ratiotwo))))))
335             (number->string (car (cadr ratiotwo)))
336             )
337         )
338         (:italic (:fontsize -5 (:musicglyph "space"))))))))
339     ))
340 )
341 }
342 {
343     \hspace #-4.25
344     \note { $notetwo } #(if (= (ly:duration-log notetwo) 4) 1 1 )
345 }
346 {
347     \hspace #0.5
348     \combine
349     \draw-line #'(-2 . 0)
350     \musicglyph "arrowheads.open.01"
351 }

```

```

352     }
353   }
354   #})
355 )
356
357
358
359 MModEquationBegin =
360 #(define-music-function
361   (notevalue1 ratio1 notevalue2 ratio2 verticaloffset )
362   (ly:duration? list? ly:duration? list? number? )
363   (let* (
364     (noteone notevalue1)
365     (notetwo notevalue2)
366     (ratioone ratio1)
367     (ratiotwo ratio2)
368   )
369     #{
370
371       \tweak Y-offset #verticaloffset
372       \textMark \markup \left-align {
373
374         \raise #0
375         \fontsize #-4.5
376         \override #'(self-alignment . LEFT)
377         {
378           {
379             \hspace #(if (< (ly:duration-log noteone) 3) -4.15
380                           (+ (* (ly:duration-dot-count noteone) -0.5) -4.5))
381             \concat {
382               \combine
383               \musicglyph "arrowheads.open.OM1"
384               \draw-line #'(2 . 0) \fontsize #-5 \musicglyph "space"
385             }
386           }
387
388           {
389             \hspace #-0.5
390             \note { $noteone } #(if (= (ly:duration-log noteone) 4) 1 1 )
391           }
392
393           {
394             \hspace #(cond
395               ((<= (ly:duration-dot-count noteone) 1) -0.5)
396               ((> (ly:duration-dot-count noteone) 1)
397                (+ (* (ly:duration-dot-count noteone) -0.5) -0.25)))
398             \right-align

```

```

400     \raise #(cond
401         ((= (ly:duration-log noteone) 0) -2)
402         ((and (> (ly:duration-log noteone) 0)
403              (<= (ly:duration-log noteone) 4) ) -.5)
404         ((> (ly:duration-log noteone) 4)
405          (+ (* (- (ly:duration-log noteone) 5) 0.6) 0))
406     )
407
408     #(cond
409         ((= (length ratioone) 1)
410          (markup
411           #:line
412           (#:hspace
413            -4.5
414            #:raise
415            2.75
416            (#:center-column
417             (#:concat
418              (#:translate
419               (cons 3.5 0.5)
420              (#:override
421               (cons 'thickness 1.5)
422              (#:translate
423               (cons -3.5 0)
424              (#:draw-line (cons -4.25 0))))))
425             #:hspace
426             -0.1
427             #:override
428             (cons 'thickness 1.5)
429             (#:translate
430              (cons 0 0.5)
431              (#:draw-line (cons 0 -0.5))))))
432           #:vspace
433           -0.45
434           #:whiteout
435           (#:halign
436            1.75
437            (#:concat
438             (#:fontsize
439              -1
440              (#:italic (#:fontsize -5 (#:musicglyph "space"))))
441             #:fontsize
442             -1
443             #:italic
444             (if (= (length (car ratioone)) 2)
445                  (begin (string-append
446                          (number->string (car (car ratioone)))
447                          " : "

```

```

448             (number->string (cadr (car ratioone)))) )
449         (number->string (car (car ratioone)))
450     )
451
452     (#:italic
453     (#:fontsize -5
454     (#:musicglyph "space")))))))))))
455
456 ((= (length ratioone) 2)
457 (markup
458   #:line
459   (#:hspace
460   -4.5
461   #:raise
462   2.75
463   (#:center-column
464   (#:concat
465   (#:translate
466   (cons 3.5 0.5)
467   (#:override
468   (cons 'thickness 1.5)
469   (#:translate
470   (cons -3.5 0)
471   (#:draw-line (cons -4.25 0))))))
472   #:hspace
473   -0.1
474   #:override
475   (cons 'thickness 1.5)
476   (#:translate
477   (cons 0 0.5)
478   (#:draw-line (cons 0 -0.5))))))
479   #:vspace
480   -0.45
481   #:whiteout
482   (#:halign
483   1.75
484   (#:concat
485   (#:fontsize
486   -1
487   (#:italic (#:fontsize -5 (#:musicglyph "space")))
488   #:fontsize
489   -1
490   #:italic
491   (if (= (length (car ratioone)) 2)
492       (begin (string-append
493               (number->string (car (car ratioone)))
494               ":"
495               (number->string (cadr (car ratioone))))))

```

```

496         (number->string (car (car ratioone)))
497     )
498     (#:italic (:fontsize -5 (:musicglyph "space"))))))))
499
500 #:line
501 (:hspace
502   -4.8
503   #:raise
504   4
505   (:center-column
506     (:concat
507       (:translate
508         (cons 3.5 0.5)
509         (:override
510           (cons 'thickness 1.5)
511           (:translate
512             (cons -3.5 0)
513             (:draw-line (cons -4.25 0))))))
514     #:hspace
515     -0.1
516     #:override
517     (cons 'thickness 1.5)
518     (:translate
519       (cons 0 0.5)
520       (:draw-line (cons 0 -0.5))))))
521 #:vspace
522 -0.45
523 #:whiteout
524 (:halign
525   1.75
526   (:concat
527     (:fontsize
528       -1
529       (:italic (:fontsize -5 (:musicglyph "space"))))
530     #:fontsize
531     -1
532     #:italic
533     (if (= (length (cadr ratioone)) 2)
534       (begin (string-append
535               (number->string (car (cadr ratioone)))
536               ":"
537               (number->string (cadr (cadr ratioone)))) )
538         (number->string (car (cadr ratioone)))
539       )
540     )
541     (:italic (:fontsize -5 (:musicglyph "space"))))))))
542 ))
543 )

```

```

544
545   }
546   {
547     \hspace #(+ (* (ly:duration-dot-count noteone) 0.35) -0.2)
548     \fontsize #2 \lower #0.5  ""
549   }
550   \concat {
551
552   {
553     \hspace #(cond ((<= (ly:duration-log notetwo) 2) 4.15 )
554                   ((> (ly:duration-log notetwo) 2) 4.15) )
555     \raise #(cond
556              ((= (ly:duration-log notetwo) 0) -2)
557              ((and (> (ly:duration-log notetwo) 0)
558                  (<= (ly:duration-log notetwo) 4) ) -.5)
559              ((> (ly:duration-log notetwo) 4)
560               (+ (* (- (ly:duration-log notetwo) 5) 0.6) 0))
561              )
562
563     #(cond
564       ((= (length ratiotwo) 1)
565        (markup
566         #:line
567         (#:hspace
568          -4.5
569          #:raise
570          3.25
571          (#:center-column
572           (#:concat
573            (#:translate
574             (cons 3.5 0.5)
575             (#:override
576              (cons 'thickness 1.5)
577
578              (#:translate
579               (cons 0 -0.5 )
580               (#:draw-line (cons 0 -0.5))))
581             )
582
583             #:hspace
584             -0.1
585
586             #:override
587             (cons 'thickness 1.5)
588             (#:translate
589              (cons -3.5 0)
590              (#:draw-line (cons -4.25 0))))))
591

```

```

592
593      #:vspace
594      -0.35
595      #:whiteout
596      (#:halign
597      1.75
598      (#:concat
599      (#:fontsize
600      -1
601      (#:italic (#:fontsize -5 (#:musicglyph "space"))))
602      #:fontsize
603      -1
604      #:italic
605      (if (= (length (car ratiotwo)) 2)
606          (begin (string-append
607                  (number->string (car (car ratiotwo)))
608                  ":"
609                  (number->string (cadr (car ratiotwo)))))
610              (number->string (car (car ratiotwo)))
611              )
612          (#:italic
613          (#:fontsize -5
614          (#:musicglyph "space")))))))))))
615
616      ((= (length ratiotwo) 2)
617      (markup
618      #:line
619      (#:hspace
620      -4.5
621      #:raise
622      3.25
623      (#:center-column
624      (#:concat
625      (#:translate
626      (cons 3.5 0.5)
627      (#:override
628      (cons 'thickness 1.5)
629      (#:translate
630      (cons 0 -0.5 )
631      (#:draw-line (cons 0 -0.5)))
632      )
633      #:hspace
634      -0.1
635      #:override
636      (cons 'thickness 1.5)
637      (#:translate
638      (cons -3.5 0)
639      (#:draw-line (cons -4.25 0))))))

```



```

640      #:vspace
641      -0.35
642      #:whiteout
643      (#:halign
644      1.75
645      (#:concat
646      (#:fontsize
647      -1
648      (#:italic (#:fontsize -5 (#:musicglyph "space"))))
649      #:fontsize
650      -1
651      #:italic
652      (if (= (length (car ratiotwo)) 2)
653          (begin (string-append
654                  (number->string (car (car ratiotwo)))
655                  ":"
656                  (number->string (cadr (car ratiotwo)))))
657              (number->string (car (car ratiotwo)))
658              )
659          (#:italic
660          (#:fontsize -5
661          (#:musicglyph "space"))))))))
662
663      #:line
664      (#:hspace
665      -4.8
666      #:raise
667      4.5
668      (#:center-column
669      (#:concat
670      (#:translate
671      (cons 3.5 0.5)
672      (#:override
673      (cons 'thickness 1.5)
674      (#:translate
675      (cons 0 -0.5 )
676      (#:draw-line (cons 0 -0.5)))
677      )
678      #:hspace
679      -0.1
680      #:override
681      (cons 'thickness 1.5)
682      (#:translate
683      (cons -3.5 0)
684      (#:draw-line (cons -4.25 0)))
685      )
686      #:vspace
687      -0.35

```

```

688         #:whiteout
689         (#:halign
690         1.75
691         (#:concat
692         (#:fontsize
693         -1
694         (#:italic (#:fontsize -5 (:#musicglyph "space"))))
695         #:fontsize
696         -1
697         #:italic
698         (if (= (length (cadr ratiotwo)) 2)
699         (begin (string-append
700         (number->string (car (cadr ratiotwo)))
701         ":"
702         (number->string (cadr (cadr ratiotwo))))))
703         (number->string (car (cadr ratiotwo)))
704         )
705         (#:italic
706         (#:fontsize -5
707         (:#musicglyph "space"))))))))
708     ))
709 )
710 }
711 {
712     \hspace #-4
713     \note { $notetwo }
714     #(if (= (ly:duration-log notetwo) 4) 1 1 )
715 }
716 {
717     \hspace #0.5
718     \combine
719     \draw-line #'(-2 . 0)
720     \musicglyph "arrowheads.open.01"
721 }
722 }
723 }
724 }
725 #})
726 )
727
728 {
729     \time 2/4
730     \tempo 4 = 60
731     c'16
732     ^\markup {\translate #'(0 . 10) " "}
733     [ c'16 c'16 c'16 ]
734     \tuplet 5/4 {c'16 [ c'16 c'16 c'16 c'16 ] }
735     \MModEquation 16 #'((5)) 16 #'() #8 #0

```

```

736 \tempo 4 = 75
737 c'16 [ c'16 c'16 c'16 ]
738 \tuplet 3/2 {c'8 [ c'8 c'8 ] }
739 \override TupletNumber.text = #tuplet-number::calc-fraction-text
740 \tuplet 5/4 {c'8 [ c'8 ] \tuplet 4/3 { c'8 [ c'8 c'8 c'8 ] }}
741 \MModEquation 8 #'((4 3)(5 4)) 8 #'((3)) 8 #-3
742 \revert TupletNumber.text
743 \break
744 \MModEquationBegin 8 #'((4 3)(5 4)) 8 #'((3)) #6.5
745 \tuplet 3/2 {
746   c'8
747   ^\markup {\translate #'(0 . 10) " "}
748   ^\markup
749   {\fontsize #-1 \raise #0.25 \note {4} #1 "= 83.33"}
750   [ c'8 c'8 ]
751 }
752 \tuplet 3/2 {
753   c'8 c'8 c'8
754   \MModEquation 4 #'() 4. #'() #6 #0
755 }
756 \time 6/8
757 c'8 c'8 c'8 c'4.
758 \bar "|."
759 }

```

[Table of Contents](#)

7.3 Metric Modulation Equation (Straight Flag)

The image shows two staves of musical notation. The first staff is in 2/4 time, starting with a tempo of 60. It contains a measure of four eighth notes, followed by a measure of five eighth notes (labeled with a '5' above a bracket), and then a measure of three eighth notes (labeled with a '3' above a bracket). Above the staff, there are two diagrams: the first shows a half note equal to a dotted quarter note with a '5' above a bracket, and the second shows a half note equal to a dotted quarter note with a '3' above a bracket. The second staff is in 4/4 time, starting with a tempo of 83.33. It contains a measure of four eighth notes (labeled with a '3' above a bracket), followed by a measure of three eighth notes (labeled with a '3' above a bracket), and then a measure of two eighth notes (labeled with a '6' above a bracket). Above the staff, there are two diagrams: the first shows a half note equal to a dotted quarter note with a '5:4' and '4:3' above a bracket, and the second shows a half note equal to a dotted quarter note with a '3' above a bracket.

7.3.1 Description

This is an alternative implementation of the metric modulation formula, introduced in the [previous entry](#), except this time it uses notes with the modern straight flag.

7.3.2 Grammar

```
\MModEquationSTR
LEFT_NOTE_DURATION #'((LEFT_TUPLET_1)(LEFT_TUPLET_2))
RIGHT_NOTE_DURATION #'((RIGHT_TUPLET_1)(RIGHT_TUPLET_2))
#V_OFFSET #H_OFFSET
\MModEquationBeginSTR
LEFT_NOTE_DURATION #'((LEFT_TUPLET_1)(LEFT_TUPLET_2))
RIGHT_NOTE_DURATION #'((RIGHT_TUPLET_1)(RIGHT_TUPLET_2))
#V_OFFSET
```

NB

1. `\MModEquationSTR` utilizes LilyPond's `\textEndMark` functions. It is therefore meant to be used at the end of a measure where the metric modulation is about to take place. Then you may use `\tempo` function on LilyPond to indicate the new tempo in the subsequent measure.
2. `\MModEquationBeginSTR`, on the other hand, uses LilyPond's `\textMark` functions, which places texts at the beginning of a measure. It is meant to be used as part of the editing process. Because `\MModEquationSTR` places the formula at the end of a measure, when that measure is at the end of a system, requiring another formula at the beginning of the subsequent measure (i.e. at the beginning of a new system) may be plausible. Using `\MModEquationBeginSTR` may be the solution.
3. For `LEFT_NOTE_DURATION` and `RIGHT_NOTE_DURATION`, use a numerical value *without* #. You may add dot(s) to indicate augmentation dot(s).

4. For the variables 2 and 4, you may put up to two instances of tuplet values, in the form of list(s) within a list. At the minimum, you will need to supply an empty list, i.e. `#'()`.

For example, if it is a simple triplet you wish to implement, you would write:

```
#'( ( 3 ) )
```

If you wish to indicate tuplet in ratio, e.g. quintuplet in the span of 4 of the note values, you would create a list with two items, i.e.:

```
#'( ( 5 4 ) )
```

You may set another tuplet instance *above* the initial tuplet, so the following list:

```
#'( ( 4 3 ) ( 5 4 ) )
```

...would produce the tuplet indication at the end of measure 3 in the example shown above.

5. For `#V_OFFSET` and `#H_OFFSET` (only for `\MModEquationSTR`), enter numbers preceded by `#`. These numbers adjust the placement of the formula. When in doubt, start with `#0`.

7.3.3 Code

```
1 \version "2.24.4"
2
3
4 MModEquationSTR =
5 #(define-music-function
6   (notevalue1 ratio1 notevalue2 ratio2 verticaloffset horizontaloffset)
7   (ly:duration? list? ly:duration? list? number? number?)
8   (let* (
9       (noteone notevalue1)
10      (notetwo notevalue2)
11      (ratioone ratio1)
12      (ratiotwo ratio2)
13      )
14     #{
15       \tweak X-offset #(- horizontaloffset 0.35)
16       \tweak Y-offset #verticaloffset
17       \textEndMark \markup \right-align {
18         \hspace #-0.5
19         \raise #0
20         \fontsize #-4.5
21         \override #'(self-alignment . LEFT)
22         {
23           {
24             \override #'(flag-style . modern-straight-flag)
25             \note { $noteone } #(if (= (ly:duration-log noteone) 4) 1 1 )
26           }
27           {
```

```

28     \hspace #(if (< (ly:duration-log noteone) 3) -4.15
29                (+ (* (ly:duration-dot-count noteone) -0.5) -4.5))
30     \concat {
31         \combine
32         \musicglyph "arrowheads.open.OM1"
33         \draw-line #'(2 . 0) \fontsize #-5 \musicglyph "space"
34     }
35 }
36 {
37     \hspace #(cond ((<= (ly:duration-log noteone) 2) 0.75)
38                  ((> (ly:duration-log noteone) 2) 0.25) )
39     \raise #(cond
40              ((= (ly:duration-log noteone) 0) -2)
41              ((and (> (ly:duration-log noteone) 0)
42                  (<= (ly:duration-log noteone) 4) ) -.5)
43              ((> (ly:duration-log noteone) 4)
44               (+ (* (- (ly:duration-log noteone) 5) 0.6) 0))
45              )
46     #(cond
47         ((= (length ratioone) 1)
48          (markup
49           #:line
50           (:#hspace
51            -4.5
52           #:raise
53            2.75
54           (:#center-column
55            (:#concat
56             (:#translate
57              (cons 3.5 0.5)
58             (:#override
59              (cons 'thickness 1.5)
60             (:#translate
61              (cons -3.5 0)
62              (:#draw-line (cons -4.25 0))))))
63           (:#hspace
64            -0.1
65           (:#override
66            (cons 'thickness 1.5)
67            (:#translate
68             (cons 0 0.5)
69             (:#draw-line (cons 0 -0.5))))))
70           #:vspace
71           -0.45
72           #:whiteout
73           (:#halign
74            1.75
75           (:#concat

```

```

76         (:fontsize
77         -1
78         (:italic (:fontsize -5 (:musicglyph "space")))
79         (:fontsize
80         -1
81         (:italic
82         (if (= (length (car ratioone)) 2)
83             (begin (string-append
84                     (number->string (car (car ratioone)))
85                     ":"
86                     (number->string (cadr (car ratioone)))) )
87             (number->string (car (car ratioone)))
88             )
89
90         (:italic (:fontsize -5 (:musicglyph "space"))))))))
91
92 ((= (length ratioone) 2)
93 (markup
94   #:line
95   (:hspace
96   -4.5
97   #:raise
98   2.75
99   (:center-column
100   (:concat
101     (:translate
102     (cons 3.5 0.5)
103     (:override
104     (cons 'thickness 1.5)
105     (:translate
106     (cons -3.5 0)
107     (:draw-line (cons -4.25 0))))))
108   #:hspace
109   -0.1
110   #:override
111   (cons 'thickness 1.5)
112   (:translate
113   (cons 0 0.5)
114   (:draw-line (cons 0 -0.5))))
115   #:vspace
116   -0.45
117   #:whiteout
118   (:halign
119   1.75
120   (:concat
121   (:fontsize
122   -1
123   (:italic (:fontsize -5 (:musicglyph "space"))))

```

```

124         #:fontsize
125         -1
126         #:italic
127         (if (= (length (car ratioone)) 2)
128             (begin (string-append
129                     (number->string (car (car ratioone)))
130                     ":"
131                     (number->string (cadr (car ratioone)))) )
132             (number->string (car (car ratioone)))
133         )
134         (#:italic (#:fontsize -5 (:#musicglyph "space"))))))))
135 #:line
136 (:#hspace
137   -4.8
138   #:raise
139   4
140   (:#center-column
141     (:#concat
142       (:#translate
143         (cons 3.5 0.5)
144         (:#override
145           (cons 'thickness 1.5)
146           (:#translate
147             (cons -3.5 0)
148             (:#draw-line (cons -4.25 0))))
149       (:#hspace
150         -0.1
151         (:#override
152           (cons 'thickness 1.5)
153           (:#translate
154             (cons 0 0.5)
155             (:#draw-line (cons 0 -0.5))))
156       (:#vspace
157         -0.45
158         (:#whiteout
159           (:#halign
160             1.75
161             (:#concat
162               (:#fontsize
163                 -1
164                 (#:italic (#:fontsize -5 (:#musicglyph "space")))
165               (:#fontsize
166                 -1
167                 (:#italic
168                   (if (= (length (cadr ratioone)) 2)
169                       (begin (string-append
170                               (number->string (car (cadr ratioone)))
171                               ":"

```



```

172             (number->string (cadr (cadr ratioone)))) )
173             (number->string (car (cadr ratioone)))
174             )
175             (:italic (:fontsize -5 (:musicglyph "space"))))))))
176         ))
177     )
178 }
179 }
180 }
181 \tweak self-alignment-X #CENTER
182 \tweak Y-offset #(- verticaloffset 0.4)
183 \tweak X-offset #(- horizontaloffset 0.35)
184 \textEndMark \markup { \fontsize #-3 "="}
185
186 \tweak Y-offset #verticaloffset
187 \tweak self-alignment-X #RIGHT
188 \tweak X-offset #(- horizontaloffset 0.35)
189 \tweak self-alignment-Y -1
190 \textEndMark \markup \left-align {
191   \hspace #1.9
192   \raise #0
193   \fontsize #-4.5
194   \concat {
195     {
196       \hspace #(cond ((<= (ly:duration-log notetwo) 2) 4)
197                     ((> (ly:duration-log notetwo) 2) 3.75) )
198       \raise #(cond
199         ((= (ly:duration-log notetwo) 0) -2)
200         ((and (> (ly:duration-log notetwo) 0)
201              (<= (ly:duration-log notetwo) 4) ) -.5)
202         ((> (ly:duration-log notetwo) 4)
203          (+ (* (- (ly:duration-log notetwo) 5) 0.6) 0))
204       )
205     #(cond
206       ((= (length ratiotwo) 1)
207        (markup
208         #:line
209         (:hspace
210          -4.5
211          #:raise
212          3.25
213          (:center-column
214           (:concat
215            (:translate
216             (cons 3.5 0.5)
217             #:override
218             (cons 'thickness 1.5)
219             (:translate

```

```

220         (cons 0 -0.5 )
221         (#:draw-line (cons 0 -0.5)))
222     )
223     #:hspace
224     -0.1
225     #:override
226     (cons 'thickness 1.5)
227     (#:translate
228     (cons -3.5 0)
229     (#:draw-line (cons -4.25 0))))
230     #:vspace
231     -0.35
232     #:whiteout
233     (#:halign
234     1.75
235     (#:concat
236     (#:fontsize
237     -1
238     (#:italic (#:fontsize -5 (#:musicglyph "space"))))
239     #:fontsize
240     -1
241     #:italic
242     (if (= (length (car ratiotwo)) 2)
243         (begin (string-append
244                 (number->string (car (car ratiotwo)))
245                 ":"
246                 (number->string (cadr (car ratiotwo)))) )
247         (number->string (car (car ratiotwo)))
248         )
249     (#:italic
250     (#:fontsize -5
251     (#:musicglyph "space")))))))))))
252 ((= (length ratiotwo) 2)
253 (markup
254   #:line
255   (#:hspace
256   -4.5
257   #:raise
258   3.25
259   (#:center-column
260   (#:concat
261   (#:translate
262   (cons 3.5 0.5)
263   (#:override
264   (cons 'thickness 1.5)
265
266   (#:translate
267   (cons 0 -0.5 )

```

```

268         (#:draw-line (cons 0 -0.5)))
269     )
270     #:hspace
271     -0.1
272     #:override
273     (cons 'thickness 1.5)
274     (#:translate
275     (cons -3.5 0)
276     (#:draw-line (cons -4.25 0))))
277 #:vspace
278 -0.35
279 #:whiteout
280 (#:halign
281 1.75
282 (#:concat
283 (#:fontsize
284 -1
285 (#:italic (#:fontsize -5 (#:musicglyph "space"))))
286 #:fontsize
287 -1
288 #:italic
289 (if (= (length (car ratiotwo)) 2)
290     (begin (string-append
291             (number->string (car (car ratiotwo)))
292             ":"
293             (number->string (cadr (car ratiotwo)))) )
294     (number->string (car (car ratiotwo)))
295     )
296     (#:italic (#:fontsize -5 (#:musicglyph "space"))))))))
297 #:line
298 (#:hspace
299 -4.8
300 #:raise
301 4.5
302 (#:center-column
303 (#:concat
304 (#:translate
305 (cons 3.5 0.5)
306 (#:override
307 (cons 'thickness 1.5)
308 (#:translate
309 (cons 0 -0.5 )
310 (#:draw-line (cons 0 -0.5)))
311 )
312 #:hspace
313 -0.1
314 #:override
315 (cons 'thickness 1.5)

```

```

316         (:translate
317         (cons -3.5 0)
318         (:draw-line (cons -4.25 0)))
319     )
320     #:vspace
321     -0.35
322     #:whiteout
323     (:halign
324     1.75
325     (:concat
326     (:fontsize
327     -1
328     (:italic (:fontsize -5 (:musicglyph "space"))))
329     #:fontsize
330     -1
331     #:italic
332     (if (= (length (cadr ratiotwo)) 2)
333         (begin (string-append
334                 (number->string (car (cadr ratiotwo)))
335                 ":"
336                 (number->string (cadr (cadr ratiotwo))))))
337         (number->string (car (cadr ratiotwo)))
338         )
339     )
340     (:italic (:fontsize -5 (:musicglyph "space"))))))))
341 ))
342 )
343 }
344 {
345     \hspace #-4.25
346     \override #'(flag-style . modern-straight-flag)
347     \note { $notetwo } #(if (= (ly:duration-log notetwo) 4) 1 1 )
348 }
349 {
350     \hspace #0.5
351     \combine
352     \draw-line #'(-2 . 0)
353     \musicglyph "arrowheads.open.01"
354 }
355 }
356 }
357 #})
358 )
359
360
361 MModEquationBeginSTR =
362 #(define-music-function
363     (notevalue1 ratio1 notevalue2 ratio2 verticaloffset )

```

```

364 (ly:duration? list? ly:duration? list? number? )
365 (let* (
366     (noteone notevalue1)
367     (notetwo notevalue2)
368     (ratioone ratio1)
369     (ratiotwo ratio2)
370 )
371 #{
372
373     \tweak Y-offset #verticaloffset
374     \textMark \markup \left-align {
375
376         \raise #0
377         \fontsize #-4.5
378         \override #'(self-alignment . LEFT)
379         {
380             {
381                 \hspace #(if (< (ly:duration-log noteone) 3) -4.15
382                             (+ (* (ly:duration-dot-count noteone) -0.5) -4.5))
383                 \concat {
384                     \combine
385                     \musicglyph "arrowheads.open.OM1"
386                     \draw-line #'(2 . 0) \fontsize #-5 \musicglyph "space"
387                 }
388             }
389
390             {
391                 \hspace #-0.5
392                 \override #'(flag-style . modern-straight-flag)
393                 \note { $noteone } #(if (= (ly:duration-log noteone) 4) 1 1 )
394             }
395
396
397             {
398                 \hspace #(cond
399                     ((<= (ly:duration-dot-count noteone) 1) -0.5)
400                     ((> (ly:duration-dot-count noteone) 1)
401                      (+ (* (ly:duration-dot-count noteone) -0.5) -0.25)))
402                 \right-align
403                 \raise #(cond
404                     ((= (ly:duration-log noteone) 0) -2)
405                     ((and (> (ly:duration-log noteone) 0)
406                          (<= (ly:duration-log noteone) 4) ) -.5)
407                     ((> (ly:duration-log noteone) 4)
408                      (+ (* (- (ly:duration-log noteone) 5) 0.6) 0))
409                     )
410
411             #(cond

```

```

412      ((= (length ratioone) 1)
413      (markup
414        #:line
415        (#:hspace
416          -4.5
417          #:raise
418          2.75
419          (#:center-column
420            (#:concat
421              (#:translate
422                (cons 3.5 0.5)
423                (#:override
424                  (cons 'thickness 1.5)
425                  (#:translate
426                    (cons -3.5 0)
427                    (#:draw-line (cons -4.25 0))))))
428              #:hspace
429              -0.1
430              #:override
431              (cons 'thickness 1.5)
432              (#:translate
433                (cons 0 0.5)
434                (#:draw-line (cons 0 -0.5))))))
435        #:vspace
436        -0.45
437        #:whiteout
438        (#:halign
439          1.75
440          (#:concat
441            (#:fontsize
442              -1
443              (#:italic (#:fontsize -5 (#:musicglyph "space"))))
444            #:fontsize
445            -1
446            #:italic
447            (if (= (length (car ratioone)) 2)
448              (begin (string-append
449                      (number->string (car (car ratioone)))
450                      ":"
451                      (number->string (cadr (car ratioone)))) )
452              (number->string (car (car ratioone)))
453              )
454            (#:italic
455              (#:fontsize -5
456                (#:musicglyph "space"))))))))
458      ((= (length ratioone) 2)

```

```

460      (markup
461      #:line
462      (#:hspace
463      -4.5
464      #:raise
465      2.75
466      (#:center-column
467      (#:concat
468      (#:translate
469      (cons 3.5 0.5)
470      (#:override
471      (cons 'thickness 1.5)
472      (#:translate
473      (cons -3.5 0)
474      (#:draw-line (cons -4.25 0))))))
475      #:hspace
476      -0.1
477      #:override
478      (cons 'thickness 1.5)
479      (#:translate
480      (cons 0 0.5)
481      (#:draw-line (cons 0 -0.5))))))
482      #:vspace
483      -0.45
484      #:whiteout
485      (#:halign
486      1.75
487      (#:concat
488      (#:fontsize
489      -1
490      (#:italic (#:fontsize -5 (#:musicglyph "space"))))
491      #:fontsize
492      -1
493      #:italic
494      (if (= (length (car ratioone)) 2)
495          (begin (string-append
496                  (number->string (car (car ratioone)))
497                  ":"
498                  (number->string (cadr (car ratioone))))))
499          (number->string (car (car ratioone))))
500      )
501      (#:italic (#:fontsize -5 (#:musicglyph "space"))))))))
502
503      #:line
504      (#:hspace
505      -4.8
506      #:raise
507      4

```

```

508         (:center-column
509         (:concat
510         (:translate
511         (cons 3.5 0.5)
512         (:override
513         (cons 'thickness 1.5)
514         (:translate
515         (cons -3.5 0)
516         (:draw-line (cons -4.25 0))))))
517         #:hspace
518         -0.1
519         (:override
520         (cons 'thickness 1.5)
521         (:translate
522         (cons 0 0.5)
523         (:draw-line (cons 0 -0.5))))))
524         #:vspace
525         -0.45
526         #:whiteout
527         (:halign
528         1.75
529         (:concat
530         (:fontsize
531         -1
532         (:italic (:fontsize -5 (:musicglyph "space"))))
533         #:fontsize
534         -1
535         #:italic
536         (if (= (length (cadr ratioone)) 2)
537             (begin (string-append
538                     (number->string (car (cadr ratioone)))
539                     ":"
540                     (number->string (cadr (cadr ratioone)))) )
541             (number->string (car (cadr ratioone)))
542             )
543         (:italic (:fontsize -5 (:musicglyph "space"))))))))
544     ))
545 )
546
547 }
548 {
549 \hspace #(+ (* (ly:duration-dot-count noteone) 0.35) -0.2)
550 \fontsize #2 \lower #0.5 "="
551 }
552 \concat {
553
554 {
555

```



```

556      \hspace #(cond ((<= (ly:duration-log notetwo) 2) 4.15 )
557                    ((> (ly:duration-log notetwo) 2) 4.15) )
558      \raise #(cond
559              ((= (ly:duration-log notetwo) 0) -2)
560              ((and (> (ly:duration-log notetwo) 0)
561                  (<= (ly:duration-log notetwo) 4) ) -.5)
562              ((> (ly:duration-log notetwo) 4)
563                  (+ (* (- (ly:duration-log notetwo) 5) 0.6) 0))
564              )
565
566      #(cond
567        ((= (length ratiotwo) 1)
568         (markup
569          #:line
570          (#:hspace
571           -4.5
572           #:raise
573           3.25
574           (#:center-column
575            (#:concat
576             (#:translate
577              (cons 3.5 0.5)
578              (#:override
579               (cons 'thickness 1.5)
580
581               (#:translate
582                (cons 0 -0.5 )
583                (#:draw-line (cons 0 -0.5))))
584             )
585
586             #:hspace
587             -0.1
588
589             #:override
590             (cons 'thickness 1.5)
591             (#:translate
592              (cons -3.5 0)
593              (#:draw-line (cons -4.25 0))))))
594
595
596          #:vspace
597          -0.35
598          #:whiteout
599          (#:halign
600           1.75
601           (#:concat
602            (#:fontsize
603             -1

```

```

604      (#:italic (:fontsize -5 (:musicglyph "space")))
605      #:fontsize
606      -1
607      #:italic
608      (if (= (length (car ratiotwo)) 2)
609          (begin (string-append
610                  (number->string (car (car ratiotwo)))
611                  ":"
612                  (number->string (cadr (car ratiotwo)))))
613              (number->string (car (car ratiotwo)))
614              )
615      (#:italic
616      (:fontsize -5
617      (:musicglyph "space")))))))))))
618
619  ((= (length ratiotwo) 2)
620  (markup
621  #:line
622  (:hspace
623  -4.5
624  #:raise
625  3.25
626  (:center-column
627  (:concat
628  (:translate
629  (cons 3.5 0.5)
630  (:override
631  (cons 'thickness 1.5)
632  (:translate
633  (cons 0 -0.5 )
634  (:draw-line (cons 0 -0.5))))
635  )
636  #:hspace
637  -0.1
638  #:override
639  (cons 'thickness 1.5)
640  (:translate
641  (cons -3.5 0)
642  (:draw-line (cons -4.25 0))))
643  #:vspace
644  -0.35
645  #:whiteout
646  (:halign
647  1.75
648  (:concat
649  (:fontsize
650  -1
651  (:italic (:fontsize -5 (:musicglyph "space"))))

```

```

652         #:fontsize
653         -1
654         #:italic
655         (if (= (length (car ratiotwo)) 2)
656             (begin (string-append
657                     (number->string (car (car ratiotwo)))
658                     ":"
659                     (number->string (cadr (car ratiotwo)))))
660                 (number->string (car (car ratiotwo)))
661             )
662         #:italic
663         (#:fontsize -5
664             (#:musicglyph "space"))))))))
665
666 #:line
667 (#:hspace
668     -4.8
669     #:raise
670     4.5
671     (#:center-column
672         (#:concat
673             (#:translate
674                 (cons 3.5 0.5)
675                 (#:override
676                     (cons 'thickness 1.5)
677                     (#:translate
678                         (cons 0 -0.5 )
679                         (#:draw-line (cons 0 -0.5)))
680                 )
681             #:hspace
682             -0.1
683             #:override
684             (cons 'thickness 1.5)
685             (#:translate
686                 (cons -3.5 0)
687                 (#:draw-line (cons -4.25 0)))
688             )
689         #:vspace
690         -0.35
691         #:whiteout
692         (#:halign
693             1.75
694             (#:concat
695                 (#:fontsize
696                     -1
697                     (#:italic (#:fontsize -5 (#:musicglyph "space")))
698                     #:fontsize
699                     -1

```

```

700             #:italic
701             (if (= (length (cadr ratiotwo)) 2)
702                 (begin (string-append
703                     (number->string (car (cadr ratiotwo)))
704                     ":"
705                     (number->string (cadr (cadr ratiotwo))))))
706                 (number->string (car (cadr ratiotwo)))
707                 )
708             (#:italic
709             (#:fontsize -5
710                 (#:musicglyph "space"))))))))
711         ))
712     )
713 }
714 {
715     \hspace #-4
716     \override #'(flag-style . modern-straight-flag)
717     \note { $notetwo }
718     #(if (= (ly:duration-log notetwo) 4) 1 1 )
719 }
720 {
721     \hspace #0.5
722     \combine
723     \draw-line #'(-2 . 0)
724     \musicglyph "arrowheads.open.01"
725 }
726 }
727 }
728 }
729 #})
730 )
731
732
733 {
734     \time 2/4
735     \tempo 4 = 60
736     c'16
737     ^\markup {\translate #'(0 . 10) " "}
738     [ c'16 c'16 c'16 ]
739     \tuplet 5/4 {c'16 [ c'16 c'16 c'16 c'16 ] }
740     \MModEquationSTR 16 #'((5)) 16 #'() #8 #0
741     \tempo 4 = 75
742     c'16 [ c'16 c'16 c'16 ]
743     \tuplet 3/2 {c'8 [ c'8 c'8 ] }
744     \override TupletNumber.text = #tuplet-number::calc-fraction-text
745     \tuplet 5/4 {c'8 [ c'8 ] \tuplet 4/3 { c'8 [ c'8 c'8 c'8 ] }}
746     \MModEquationSTR 8 #'((4 3)(5 4)) 8 #'((3)) 8 #-3
747     \revert TupletNumber.text

```

```

748 \break
749 \MModEquationBeginSTR 8 #'((4 3)(5 4)) 8 #'((3)) #6.5
750 \tuplet 3/2 {
751   c'8
752   ^\markup {\translate #'(0 . 10) " "}
753   ^\markup
754   {\fontsize #-1 \raise #0.25 \note {4} #1 "= 83.33"}
755   [ c'8 c'8 ]
756 }
757 \tuplet 3/2 {
758   c'8 c'8 c'8
759   \MModEquationSTR 4 #'() 4. #'() #6 #0
760 }
761 \time 6/8
762 c'8 c'8 c'8 c'4.
763 \bar "|."
764 }
765
766 \layout {
767   \context{
768     \Score
769     harmonicDots = ##t
770     \override Flag.stencil = #modern-straight-flag
771     \override MetronomeMark.flag-style = #'modern-straight-flag
772     \override StemTremolo.shape = #'beam-like
773     \override StemTremolo.slope = #0.4
774
775   }
776 }
777

```

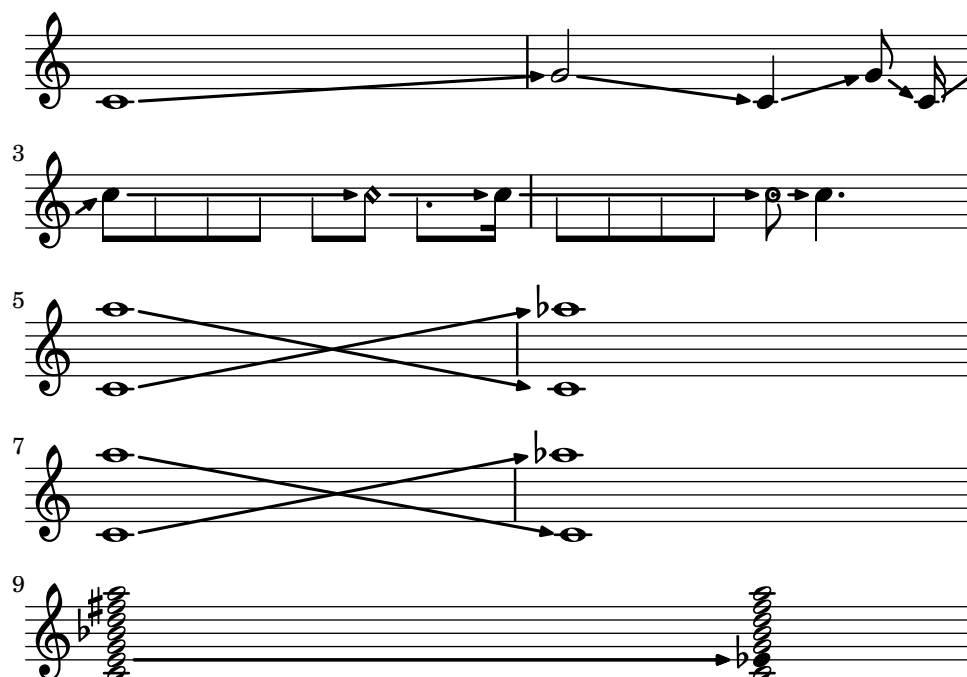
[Table of Contents](#)

Chapter 8

Spanners

This chapter covers snippets that take advantages of spanners (text, line, etc.) in one way or another. Because functions such as `\startTextSpan` and `\stopTextSpan` activate and deactivate these snippets, caution must be paid when using more than one of them at the same time. See [Example in Combinations](#) to avoid conflicts between or among the spanner snippets.

8.1 Arrow Lines



8.1.1 Description

Implementation of arrow lines. It takes advantage of the `\glissando` function. It is possible to have the arrow line span over multiple notes, as `glissando-skip` parameter is set to `##t`. When the arrow line spans over multiple systems, the arrow mark will not appear at the end of the system.¹ Furthermore, it is possible to use the function on dyads and chords. The placement of the beginning of the arrow is adjusted according to the different types of notehead.

8.1.2 Grammar

```
\arrowLineOn STARTING_NOTE (NOTES...)
\arrowLineOff ARRIVAL_NOTE
```

8.1.3 Code

```
1 \version "2.24.4"
2
3 arrowLineOn =
4 #(define-music-function (note)(ly:music?)
5   (define paddingvalue (if (music-is-of-type? note 'event-chord)
6     (ly:duration-log
7       (ly:music-property
8         (first
9           (ly:music-property note 'elements))
10        'duration))
11     (ly:duration-log
```

1. See [Discussion](#) for more details.

```

12                                     (ly:music-property note 'duration))))
13
14   #{
15
16     \override Glissando.breakable = ##t
17     \override Glissando.after-line-breaking = ##t
18     \override Glissando.thickness = #2.35
19     \override Glissando.bound-details.right.arrow = ##t
20     \override Glissando.bound-details.right-broken.arrow = ##f
21     \override Glissando.bound-details.right-broken.padding = #-1
22     \override Glissando.bound-details.left.padding =
23       #(cond ((= paddingvalue 0) 0.85)
24              ((= paddingvalue 1) 0.65)
25              ((>= paddingvalue 2) 0.65))
26
27     \override Glissando.bound-details.right.padding = #0.25
28     #note
29     \glissando \override NoteColumn.glissando-skip = ##t
30   #})
31
32
33 arrowLineOff =
34 {
35   \revert Glissando.breakable
36   \revert Glissando.after-line-breaking
37   \revert Glissando.thickness
38   \revert Glissando.bound-details.right.arrow
39   \revert Glissando.bound-details.right-broken.arrow
40   \revert Glissando.bound-details.right-broken.padding
41   \revert Glissando.bound-details.left.padding
42   \revert Glissando.bound-details.right.padding
43   \revert NoteColumn.glissando-skip
44 }
45
46
47
48 \score {
49
50   {
51     \override Score.TimeSignature.stencil = ##f
52
53     \arrowLineOn
54     c'1
55     \arrowLineOff
56
57     \arrowLineOn
58     g'2
59     \arrowLineOff

```



```

60
61 \arrowLineOn
62 c'4
63 \arrowLineOff
64
65 \arrowLineOn
66 g'8 \noBeam
67 \arrowLineOff
68 \arrowLineOn
69 c'16 s16 |
70
71 \break
72 \arrowLineOff
73 \arrowLineOn
74 c''8
75 \override Voice.NoteHead.transparent = ##t
76 8 8 8 8
77 \revert Voice.NoteHead.transparent
78 \arrowLineOff
79
80 \arrowLineOn
81 8 \harmonic
82 \override Voice.NoteHead.transparent = ##t
83 \once \override Voice.Dots.extra-offset = #'(-1 . -0.75)
84
85 8.
86 \revert Voice.NoteHead.transparent
87 \arrowLineOff
88
89 \arrowLineOn
90 16
91
92 \override Voice.NoteHead.transparent = ##t
93 8 8 8 8
94 \revert Voice.NoteHead.transparent
95 \arrowLineOff
96 \easyHeadsOn
97 \arrowLineOn
98 8
99 \arrowLineOff
100 \easyHeadsOff
101 4.
102 \break
103 \arrowLineOn
104 <c' a''>1
105
106 \arrowLineOff
107 <aes'' c'>1

```

```

108
109 <<
110 { \arrowLineOn a''1 \arrowLineOff c'1} \\\
111 {\arrowLineOn c'1 \arrowLineOff aes''1}
112 >>
113
114 \break
115 \override Voice.Stem.stencil = ##f
116 \override Voice.NoteHead.stencil = #ly:text-interface::print
117 \override Voice.NoteHead.text = \markup{\musicglyph "noteheads.s1"}
118 \set glissandoMap = #'((1 . 1) (1 . 1))
119 \arrowLineOn
120 <c' e' g' bes' d'' fis'' a''>2
121 s4
122 \arrowLineOff
123 <c'
124 \single \override NoteHead.text =
125 \markup{\musicglyph "noteheads.s2"} es'
126 g' bes' d'' fis'' a''>4
127
128 }
129
130
131 \layout {
132
133   indent = #0
134   line-width = #125
135   ragged-last = ##f
136
137   \context {
138     \Score
139     proportionalNotationDuration = #(ly:make-moment 1/7)
140   }
141 }
142 }

```

8.1.4 Discussion

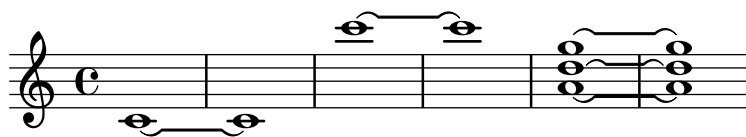
There are a few things to consider when using the arrow lines on dyads and chords:

- By default, all pair of notes will have arrow lines. In order to selectively show the arrow lines, use `\set glissandoMap`. See [1.3.3 Expressive marks as lines](#) in LilyPond's *Notation Reference* for details.
- Just as the ordinary `\glissando` function, the X coordinate of the terminating point for *all* of the lines between two dyads/chords is determined by the presence of accidentals in the arrival dyads/chords. Thus, if there is an accidental on one or more of the pitches in the arriving dyad/chord, there may be a space between the tip of arrow and the pitches *without* the accidentals. Should this be avoided, it is best to apply the arrow lines in different layers,

so that each of the layers will have a different X-coordinate value of the terminating point of the arrow lines.

Table of Contents

8.2 Boulez Tie



8.2.1 Description

Implementation of ties seen in Movement 5 *Tombeau of Pli selon pli* by Pierre Boulez.² The shape is roughly a combination of a short, *laissez vibrer* sign on both originating and arriving notes, connected by a horizontal line. In the score, these are manually written in, and the shape is somewhat irregular.

8.2.2 Grammar

```
\override Tie.stencil = \boulezTie
```

8.2.3 Code

```
1 \version "2.24.4"
2
3 \language "english"
4
5 \pointAndClickOff
6
7 boulezTie =
8 #(lambda (grob)
9
10   (let (
11     ;defines four control points, each of which containing
12     ;x and y, defined from a to h...
13     (a (caar (ly:grob-property grob 'control-points)) )
14     (b (cdar (ly:grob-property grob 'control-points)) )
15     (c (car (cadr (ly:grob-property grob 'control-points))))
16     (d (cdr (cadr (ly:grob-property grob 'control-points))))
17     (e (car (caddr (ly:grob-property grob 'control-points))))
18     (f (cdr (caddr (ly:grob-property grob 'control-points))))
19     (g (car (cadddr (ly:grob-property grob 'control-points))))
20     (h (cdr (cadddr (ly:grob-property grob 'control-points))))
21   )
22   (make-path-stencil
23
24     (list 'moveto a b
25           'curveto a b (* (+ a (+ c 0.5)) 0.5) d (+ c 0.5) b
26           'lineto (- e 0.5) b
```

2. Pierre Boulez, *Pli selon pli: V. Tombeau* (London: Universal Edition, 1971).

```

27         'curveto (- e 0.5) b (+ (- e 0.5) (* (- g (- e 0.5)) 0.5)) f g h
28
29         'curveto g h (+ (- e 0.5) (* (- g (- e 0.5)) 0.6))
30         (+ f 0.1) (- e 0.5) (+ b 0.15)
31         'lineto (+ c 0.5) (+ b 0.15)
32         'curveto (+ c 0.5) (+ b 0.15) (* (+ a (+ c 0.5)) 0.5) (+ d 0.1) a b
33         'closepath
34     )
35
36     0.01
37     1
38     1
39     #t
40 )
41 )
42 )
43
44 {
45
46     \override Tie.stencil = \boulezTie
47
48     c'1 ~ c'1
49     c''' ~ c'''
50     <a' d'' g''> ~ <a' d'' g''>
51 }}
52
53

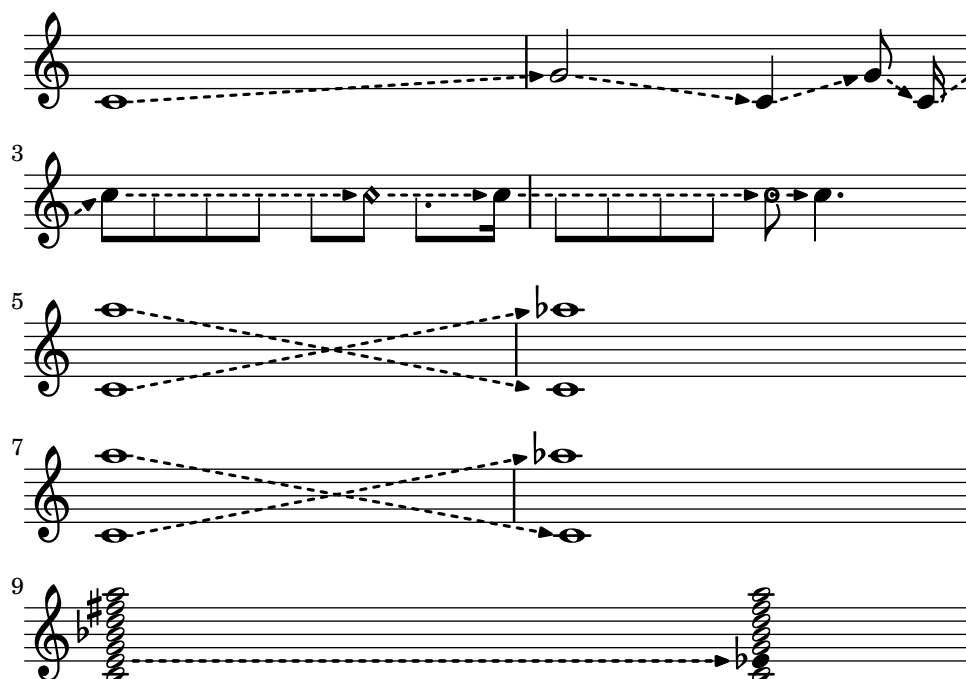
```

8.2.4 Discussion

It is possible to adjust the shape of the tie using `\shape` and `\vshape` commands.

[Table of Contents](#)

8.3 Dashed Arrow Lines



8.3.1 Description

Implementation of dashed arrow lines. Its design is nearly identical to that of the [Arrow Lines](#).

8.3.2 Grammar

```
\dashedArrowLineOn STARTING_NOTE (NOTES...)
```

```
\dashedArrowLineOff ARRIVAL_NOTE
```

8.3.3 Code

```
1 \version "2.24.4"
2
3 dashedArrowLineOn =
4 #(define-music-function (note)(ly:music?)
5   (define paddingvalue (if (music-is-of-type? note 'event-chord)
6     (ly:duration-log
7       (ly:music-property
8         (first
9           (ly:music-property note 'elements))
10          'duration))
11     (ly:duration-log
12       (ly:music-property note 'duration))))
13
14   #{
15
16     \override Glissando.breakable = ##t
```

```

17   \override Glissando.after-line-breaking = ##t
18   \override Glissando.thickness = #2.35
19   \override Glissando.style = #'dashed-line
20   \override Glissando.bound-details.right.arrow = ##t
21   \override Glissando.bound-details.right-broken.arrow = ##f
22   \override Glissando.bound-details.right-broken.padding = #-1
23   \override Glissando.bound-details.left.padding =
24   #(cond ((= paddingvalue 0) 0.85)
25          ((= paddingvalue 1) 0.65)
26          ((>= paddingvalue 2) 0.65))
27
28   \override Glissando.bound-details.right.padding = #0.25
29   #note
30   \glissando \override NoteColumn.glissando-skip = ##t
31   #})
32
33
34 dashedArrowLineOff =
35 {
36   \revert Glissando.breakable
37   \revert Glissando.after-line-breaking
38   \revert Glissando.thickness
39   \revert Glissando.style
40   \revert Glissando.bound-details.right.arrow
41   \revert Glissando.bound-details.right-broken.arrow
42   \revert Glissando.bound-details.right-broken.padding
43   \revert Glissando.bound-details.left.padding
44   \revert Glissando.bound-details.right.padding
45   \revert NoteColumn.glissando-skip
46 }
47
48
49
50 \score {
51
52   {
53     \override Score.TimeSignature.stencil = ##f
54
55     \dashedArrowLineOn
56     c'1
57     \dashedArrowLineOff
58
59     \dashedArrowLineOn
60     g'2
61     \dashedArrowLineOff
62
63     \dashedArrowLineOn
64     c'4

```

```

65   \dashedArrowLineOff
66
67   \dashedArrowLineOn
68   g'8 \noBeam
69   \dashedArrowLineOff
70   \dashedArrowLineOn
71   c'16 s16 |
72
73   \break
74   \dashedArrowLineOff
75   \dashedArrowLineOn
76   c''8
77   \override Voice.NoteHead.transparent = ##t
78   8 8 8 8
79   \revert Voice.NoteHead.transparent
80   \dashedArrowLineOff
81
82   \dashedArrowLineOn
83   8 \harmonic
84   \override Voice.NoteHead.transparent = ##t
85   \once \override Voice.Dots.extra-offset = #'(-1 . -0.75)
86
87   8.
88   \revert Voice.NoteHead.transparent
89   \dashedArrowLineOff
90
91   \dashedArrowLineOn
92   16
93
94   \override Voice.NoteHead.transparent = ##t
95   8 8 8 8
96   \revert Voice.NoteHead.transparent
97   \dashedArrowLineOff
98   \easyHeadsOn
99   \dashedArrowLineOn
100  8
101  \dashedArrowLineOff
102  \easyHeadsOff
103  4.
104  \break
105  \dashedArrowLineOn
106  <c' a''>1
107
108  \dashedArrowLineOff
109  <aes'' c'>1
110
111  <<
112  { \dashedArrowLineOn a''1 \dashedArrowLineOff c'1} \

```



```

113     {\dashedArrowLineOn c'1 \dashedArrowLineOff aes''1}
114     >>
115     \break
116     \override Voice.Stem.stencil = ##f
117     \override Voice.NoteHead.stencil = #ly:text-interface::print
118     \override Voice.NoteHead.text = \markup{\musicglyph "noteheads.s1"}
119     \set glissandoMap = #'((1 . 1) (1 . 1))
120     \dashedArrowLineOn
121     <c' e' g' bes' d'' fis'' a''>2
122     s4
123     \dashedArrowLineOff
124     <c'
125     \single \override NoteHead.text =
126     \markup{\musicglyph "noteheads.s2"} es'
127     g' bes' d'' fis'' a''>4
128
129   }
130
131   \layout {
132
133     indent = #0
134     line-width = #125
135
136     ragged-last = ##f
137
138     \context {
139       \Score
140       proportionalNotationDuration = #(ly:make-moment 1/7)
141     }
142   }
143 }

```

8.3.4 Discussion

See [Discussion of the Arrow Lines](#). [Table of Contents](#)

8.4 Dashed Flat Slurs



8.4.1 Description

See [Flat Slurs](#) for the ordinary non-dashed version.

Because the [Flat Slurs](#) is based on a rewritten stencil, further overrides within `Slur`, such as [Making slurs with complex dash structure](#) will not work. This particular implementation addresses the issue of achieving dashed lines for the flat slurs. Here, three different stencils are created, defined as variables, then combined using `ly:stencil-add`.

See [Flat Slurs](#) for the ordinary non-dashed version, as well as [Discussion](#) for hints on tweaking.

8.4.2 Grammar

```
\override Slur.stencil = \flatSlur
```

8.4.3 Code

```
1 \version "2.24.4"
2
3 \language "english"
4
5 \pointAndClickOff
6
7 dashedFlatSlur =
8 #(lambda (grob)
9
10   (let* (
11     ;defines four control points, each of which containing
12     ;x and y, defined from a to h...
```

```

13      (a (caar (ly:grob-property grob 'control-points)) )
14      (b (cdar (ly:grob-property grob 'control-points)) )
15      (c (car (cadr (ly:grob-property grob 'control-points))))
16      (d (cdr (cadr (ly:grob-property grob 'control-points))))
17      (e (car (caddr (ly:grob-property grob 'control-points))))
18      (f (cdr (caddr (ly:grob-property grob 'control-points))))
19      (g (car (caddr (ly:grob-property grob 'control-points))))
20      (h (cdr (caddr (ly:grob-property grob 'control-points))))
21      (firstCurve (make-path-stencil
22                    (list 'moveto a b
23                          'curveto a b a (- d 0.2) c (- d 0.2)
24                          'lineto c (- d 0.4)
25                          'curveto c (- d 0.4) (+ a 0.1) (- d 0.3) a b
26                          'closepath
27                    )
28                    0.01
29                    1
30                    1
31                    #t
32                    ))
33      (secondCurve (make-path-stencil
34                     (list 'moveto e (- f 0.2)
35                           'curveto e (- f 0.2) g (- f 0.2) g h
36                           'curveto g h (+ g 0.1) (- f 0.5) e (- f 0.4)
37                           'closepath
38                     )
39                     0.01
40                     1
41                     1
42                     #t
43                     ))
44
45      (dashedLine (grob-interpret-markup
46                    grob
47                    (markup
48                      #:line
49                      (:#:postscript
50                      (string-append
51                        "0.2 setlinewidth [0.4 0.4] 0 setdash "
52                        (number->string (+ c 0.2)) " "
53                        (number->string (- d 0.3))
54                        " moveto "
55                        (number->string (- e 0.2)) " "
56                        (number->string (- f 0.3))
57                        " lineto stroke"))
58                      ))
59      )
60  )

```

```

61      (ly:stencil-add firstCurve dashedLine secondCurve
62
63          )
64      )
65  )
66  \layout {
67      indent = 0
68      line-width = 80
69  }
70  {
71      \time 2/4
72      \override Slur.stencil = \dashedFlatSlur
73
74      \override Slur.details =
75      #'(
76          (max-slope . 0)
77          (max-slope-factor . 100)
78          (free-head-distance . 0.2)
79          (free-slur-distance . 0)
80          (extra-encompass-free-distance . 0.3)
81          (extra-encompass-collision-distance . 0.8)
82          (head-slur-distance-max-ratio . 3)
83          (head-slur-distance-factor . 10)
84      )
85
86      \shape #'(
87          (0 . 1)
88          (0 . 0.5)
89          (0 . 0.5)
90          (0 . 0.25)
91      ) Slur
92
93      e'16 ( d' c' d' e' d' c'8 )
94
95      \shape #'(
96          (0 . 0)
97          (0 . -0.5)
98          (0 . -0.5)
99          (0 . -1)
100      ) Slur
101      g''16 ( f'' e'' f'' g'' f'' e''8 ) \break
102
103
104      \shape #'(
105          (( 0 . 0) (0 . 0) (0 . 0) (0 . 0))
106          ( (-1.25 . 3.5) (-1.25 . 3.75)(0 . 3.75)(0 . 2.5) )
107      ) Slur
108      c''4 ( b' a' g' ~ | \break

```

```
109
110   g'8 )
111
112   \shape #'(
113           (0 . 0)
114           (0 . 0.2)
115           (0.75 . 2.5)
116           (0 . 3)
117           ) Slur
118
119
120   a ( c' e' g' b' ) r4 |
121
122
123   }
124
```

[Table of Contents](#)

8.5 Flat Slurs



8.5.1 Description

Implementation of a straight-line slur in which only the beginning and ending parts are curved. This type of slurs can be found in scores by such composers as Brian Ferneyhough, Claus-Steffen Mahnkopf, and Donald Martino, where such slurs were manually drawn using a ruler. Even in the age of digital musical engraving, composers such as Stefan Beyer have emulated this slur using the musical engraving software programs.

This flat slur requires careful tweaking in order to make it look optimal on the score. This is discussed further in [Discussion](#).

See also [Dashed Flat Slurs](#).

8.5.2 Grammar

```
\override Slur.stencil = \flatSlur
```

8.5.3 Code

```
1 \version "2.24.4"
2
3 \language "english"
4
5 \pointAndClickOff
6
7 flatSlur =
8 #(lambda (grob)
9
10   (let (
11     ;defines four control points, each of which containing
12     ;x and y, defined from a to h...
```

```

13      (a (caar (ly:grob-property grob 'control-points)) )
14      (b (cdar (ly:grob-property grob 'control-points)) )
15      (c (car (cadr (ly:grob-property grob 'control-points))))
16      (d (cdr (cadr (ly:grob-property grob 'control-points))))
17      (e (car (caddr (ly:grob-property grob 'control-points))))
18      (f (cdr (caddr (ly:grob-property grob 'control-points))))
19      (g (car (caddr (ly:grob-property grob 'control-points))))
20      (h (cdr (caddr (ly:grob-property grob 'control-points))))
21    )
22  (make-path-stencil
23    (if (= (ly:grob-property grob 'direction) 1)
24      ;this is if the slur/tie direction is up...
25      (list 'moveto a b
26            'curveto a b a (- d 0.2) c (- d 0.2)
27            'lineto e (- f 0.2)
28            'curveto e (- f 0.2) g (- f 0.2) g h
29
30            'curveto g h (+ g 0.1) (- f 0.5) e (- f 0.4)
31            'lineto c (- d 0.4)
32            'curveto c (- d 0.4) (+ a 0.1) (- d 0.3) a b
33            'closepath
34          )
35      ;if the direction is down...
36      (list 'moveto a b
37            'curveto a b a (+ d 0.2) c (+ d 0.2)
38            'lineto e (+ f 0.2)
39
40            'curveto e (+ f 0.2) g (+ f 0.2) g h
41
42            'curveto g h (- g 0.1) (+ f 0.5) e (+ f 0.4)
43            'lineto c (+ d 0.4)
44            'curveto c (+ d 0.4) (+ a 0.3) (+ d 0.3) a b
45            'closepath
46          ))
47
48    0.01
49    1
50    1
51    #t
52  )
53 )
54 )
55 \layout {
56   indent = 0
57   line-width = 80
58 }
59 {
60   \time 2/4

```

```

61  \override Slur.stencil = \flatSlur
62
63  \override Slur.details =
64  #'(
65    (max-slope . 0)
66    (max-slope-factor . 100)
67    (free-head-distance . 0.2)
68    (free-slur-distance . 0)
69    (extra-encompass-free-distance . 0.3)
70    (extra-encompass-collision-distance . 0.8)
71    (head-slur-distance-max-ratio . 3)
72    (head-slur-distance-factor . 10)
73  )
74
75  \shape #'(
76    (0 . 1)
77    (0 . 0.5)
78    (0 . 0.5)
79    (0 . 0.25)
80  ) Slur
81
82  e'16 ( d' c' d' e' d' c'8 )
83
84  \shape #'(
85    (0 . 0)
86    (0 . -0.5)
87    (0 . -0.5)
88    (0 . -1)
89  ) Slur
90  g''16 ( f'' e'' f'' g'' f'' e''8 ) \break
91
92
93
94  c''4 ( b' a' g' ~ | \break
95
96  g'8 )
97
98  \shape #'(
99    (0 . 0)
100    (0 . 0.2)
101    (0.75 . 2.5)
102    (0 . 3)
103  ) Slur
104
105
106  a ( c' e' g' b' ) r4 |
107
108

```



```
109 }
```

8.5.4 Discussion

While this code can be used out of box by overriding `Slur.stencil` to `\flatSlur`, you may find that tweak will be needed. Initial experiments have shown that the following overrides of `Slur.details` help:

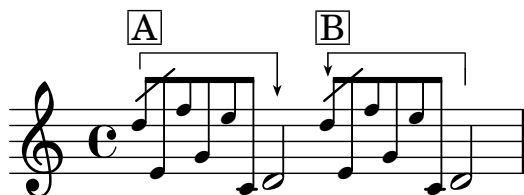
```
1  \override Slur.details =
2  #'(
3    (max-slope . 0)
4    (max-slope-factor . 100)
5    (free-head-distance . 0.2)
6    (free-slur-distance . 0)
7    (extra-encompass-free-distance . 0.3)
8    (extra-encompass-collision-distance . 0.8)
9    (head-slur-distance-max-ratio . 3)
10   (head-slur-distance-factor . 10)
11   )
```

There may be other effective `Slur.details` configurations.

Beyond this, further refinements can be made using `\shape` and `\vshape`.

[Table of Contents](#)

8.6 Grace Note Brackets I



8.6.1 Description

NB: See [Grace Note Brackets II](#) for the updated version of this code.) Replication of grace note brackets seen in scores by Pierre Boulez (e.g. *Sur Incises*,³ ...*explosante-fixe*...⁴). Bracket A in the example shows that the grace notes are to be played *before* the beat to which they are applied. Whereas Bracket B shows that the grace notes are to be played *on* the beat to which they are applied.

8.6.2 Grammar

```
\graceNoteBeforeBeatOn NOTE
\graceNoteBeforeBeatOff NOTE
\graceNoteAfterBeatOn NOTE
\graceNoteAfterBeatOff NOTE
```

8.6.3 Code

```
1 \version "2.24.4"
2
3 \language "english"
4
5 % This code includes snippet for grace note
6 % slashes, which has been taken from:
7 % https://lsr.di.unimi.it/LSR/Item?id=1048
8
9
10 graceNoteBeforeBeatOn =
11 #(define-music-function (starting_note) (ly:music?)
12   #{
13     \once \override TextSpanner.style = #'line
14     \once \override TextSpanner.bound-details.left.text =
15     \markup { \draw-line #'(0 . -1) }
16     \once \override TextSpanner.bound-details.right.text =
17     \markup {
```

3. Pierre Boulez, *Sur incises : pour trois pianos, trois harpes et trois percussions-claviers* (1996/1998) (Universal Edition, 1998).

4. Pierre Boulez, ... *explosante-fixe ... transitoire VII : (version 1991/93)* (Universal Edition, 1991).

```

18      \postscript
19      "newpath 0 0 moveto
20 0 -2.5 rlineto
21 stroke
22 newpath
23 -0.275 -2 moveto
24 0.275 -0.75 rlineto
25 0.275 0.75 rlineto
26 -0.275 -0.2 rlineto
27 closepath
28 fill"
29   }
30   \once \override TextSpanner.Y-offset = #5
31   \once \override TextSpanner.bound-details.left.padding = #0.5
32   \once \override TextSpanner.bound-details.right.padding = #-0.25
33   #starting_note
34   \startTextSpan
35   #})
36
37
38 graceNoteBeforeBeatOff =
39 #(define-music-function (ending_note) (ly:music?)
40   #{
41     #ending_note
42     \stopTextSpan
43   #})
44
45
46 graceNoteAfterBeatOn =
47 #(define-music-function (starting_note) (ly:music?)
48   #{
49     \once \override TextSpanner.style = #'line
50     \once \override TextSpanner.bound-details.right.text =
51     \markup {
52       \combine \draw-line #'(0 . -1)
53       \postscript "newpath
54 0 -1 moveto
55 0 -1 rlineto
56 stroke"
57     }
58     \once \override TextSpanner.bound-details.left.text =
59     \markup {
60       \postscript
61       "newpath 0 0 moveto
62 0 -1 rlineto
63 stroke
64 newpath
65 -0.275 -0.75 moveto

```

```

66 0.275 -0.75 rlineto
67 0.275 0.75 rlineto
68 -0.275 -0.2 rlineto
69 closepath
70 fill"
71   }
72   \once \override TextSpanner.Y-offset = #2
73   \once \override TextSpanner.bound-details.left.padding = #0.5
74   \once \override TextSpanner.bound-details.right.padding = #-0.25
75   #starting_note
76   \startTextSpan
77 #})
78
79
80 graceNoteAfterBeatOff =
81 #(define-music-function (ending_note) (ly:music?)
82   #{
83     #ending_note
84     \stopTextSpan
85   #})
86
87 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%LSR SNIPPET START%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
88
89 #(define (degrees->radians deg)
90   (* PI (/ deg 180.0)))
91
92 slash =
93 #(define-music-function (ang stem-fraction protrusion)
94   (number? number? number?)
95   (remove-grace-property 'Voice 'Stem 'direction)
96   #{
97     \once \override Stem.stencil =
98     #(lambda (grob)
99       (let* ((x-parent (ly:grob-parent grob X))
100              (is-rest? (ly:grob?
101                          (ly:grob-object x-parent 'rest))))
102         (beam (ly:grob-object grob 'beam))
103         (stil (ly:stem::print grob)))
104     (cond
105      (is-rest? empty-stencil)
106      ((ly:grob? beam)
107       (let* ((refp (ly:grob-system grob))
108              (stem-y-ext (ly:grob-extent grob Y))
109              (stem-length
110                (- (cdr stem-y-ext) (car stem-y-ext))))
111         (beam-X-pos (ly:grob-property beam 'X-positions))
112         (beam-Y-pos (ly:grob-property beam 'positions))
113         (beam-slope (/ (- (cdr beam-Y-pos) (car beam-Y-pos))

```

```

114         (- (cdr beam-X-pos) (car beam-X-pos))))
115     (beam-angle (atan beam-slope))
116     (dir (ly:grob-property grob 'direction))
117     (line-dy (* stem-length stem-fraction))
118     (line-dy-with-protrusions (if (= dir 1)
119         (+ (* 4 protrusion) beam-angle)
120         (- (* 4 protrusion) beam-angle)))
121     (ang (if (> beam-slope 0)
122         (if (= dir 1)
123             (+ (degrees->radians ang) (* beam-angle 0.7))
124             (degrees->radians ang))
125         (if (= dir 1)
126             (degrees->radians ang)
127             (- (degrees->radians ang) (* beam-angle 0.7)))))
128     (line-dx (/ line-dy-with-protrusions (tan ang)))
129     (protrusion-dx (/ protrusion (tan ang)))
130     (corr (if (= dir 1) (car stem-y-ext) (cdr stem-y-ext)))
131 (ly:stencil-add
132   stil
133   (grob-interpret-markup grob
134     (markup
135       #:translate
136       (cons (- protrusion-dx)
137         (+ corr
138           (* dir
139             (- stem-length
140               (+ stem-fraction protrusion))))))
141       #:override '(thickness . 1.7)
142       #:draw-line
143       (cons line-dx
144         (* dir line-dy-with-protrusions))))))
145   (else stil)))
146 #}}
147
148 startSlashedGraceMusic = {
149   \slash 40 1 0.5
150   \override Flag.stroke-style = #"grace"
151 }
152 stopSlashedGraceMusic = {
153   \revert Flag.stroke-style
154 }
155
156 startAcciaccaturaMusic = {
157   \slash 40 1 0.5
158   s1*0(
159   \override Flag.stroke-style = #"grace"
160 }
161 stopAcciaccaturaMusic = {

```

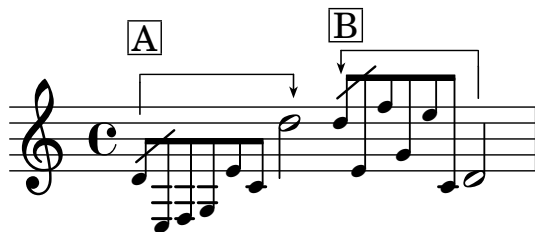
```

162   \revert Flag.stroke-style
163   s1*0)
164 }
165 %%%%%%%%%%%%%%% LSR SNIPPET END %%%%%%%%%%%%%%%
166
167
168 {
169   \grace {
170     \startSlashedGraceMusic
171     \graceNoteBeforeBeatOn d''8^\markup{\box A} e' f'' g' e'' c'
172   }
173   \graceNoteBeforeBeatOff d'2
174   \grace {
175     \startSlashedGraceMusic
176     \graceNoteAfterBeatOn d''8^\markup{\box B} e' f'' g' e'' c'
177   }
178   \graceNoteAfterBeatOff d'2
179 }

```

[Table of Contents](#)

8.7 Grace Note Brackets II



8.7.1 Description

This is an updated version of Grace Note Brackets I. It differs from the original version in that this version takes a list of three parameters in order to finely adjust the shape of the bracket in order to accommodate various shapes of grace notes and the actual note.

8.7.2 Grammar

```
\graceNoteBeforeBeatOn #'(OVERALL LEFT RIGHT) NOTE
\graceNoteBeforeBeatOff #'(OVERALL LEFT RIGHT) NOTE
\graceNoteAfterBeatOn #'(OVERALL LEFT RIGHT) NOTE
\graceNoteAfterBeatOff #'(OVERALL LEFT RIGHT) NOTE
```

NB The list accepts three integers as parameters, i.e.:

1. **OVERALL** is a value of the distance between the top line of the staff and the horizontal line of the grace note bracket. This value cannot be smaller than the skyline value established by the staff line and the notes; when the skyline value is greater than what is specified in this bracket, the skyline value is favored. When in doubt, start with 0, then increase the amount gradually.
2. **LEFT** and **RIGHT** values (negative value only!) adjust the lengths of the left and right hooks.

8.7.3 Code

```
1 \version "2.24.4"
2 \language "english"
3
4 % This code includes snippet for grace note
5 % slashes, which has been taken from:
6 % https://lsr.di.unimi.it/LSR/Item?id=1048
7
8 % Slightly revised, Jan. 19/22 2025 - YO
9
10 graceNoteBeforeBeatOn =
11 #(define-music-function (setting-list starting_note) (list? ly:music? )
12   #{
13     \once \override TextSpanner.style = #'line
```

```

14 \once \override TextSpanner.bound-details.left.text =
15 \markup {
16 \combine
17 \draw-line #(cons 0 -0.5)
18 \postscript #(string-append "newpath
19 0 -0.5 moveto
20 0 " (number->string (cadr setting-list)) " rlineto
21 stroke")
22 }
23 \once \override TextSpanner.bound-details.right.text =
24 \markup {
25 \postscript
26 #(string-append "newpath 0 0 moveto
27 0 " (number->string (caddr setting-list)) " rlineto
28 stroke
29 newpath
30 -0.275 " (number->string (+ (caddr setting-list) 0.25)) " moveto
31 0.275 -0.75 rlineto
32 0.275 0.75 rlineto
33 -0.275 -0.2 rlineto
34 closepath
35 fill")
36 }
37 \once \override TextSpanner.extra-offset = #(cons 0 (car setting-list))
38 \once \override TextSpanner.bound-details.left.padding = #0.5
39 \once \override TextSpanner.bound-details.right.padding = #-0.25
40 #starting_note
41 \startTextSpan
42 #})
43
44
45 graceNoteBeforeBeatOff =
46 #(define-music-function (ending_note) (ly:music?)
47 #{
48 #ending_note
49 \stopTextSpan
50 #})
51
52
53 graceNoteAfterBeatOn =
54 #(define-music-function (setting-list starting_note) (list? ly:music?)
55 #{
56 \once \override TextSpanner.style = #'line
57 \once \override TextSpanner.bound-details.right.text =
58 \markup {
59 \combine
60 \draw-line #(cons 0 -1)
61 \postscript #(string-append "newpath

```



```

62 0 -1 moveto
63 0 " (number->string (caddr setting-list)) " rlineto
64 stroke")
65 }
66 \once \override TextSpanner.bound-details.left.text =
67 \markup {
68 \postscript
69 # (string-append "newpath 0 0 moveto
70 0 " (number->string (cadr setting-list)) " rlineto
71 stroke
72 newpath
73 -0.275 " (number->string (+ (cadr setting-list) 0.25)) " moveto
74 0.275 -0.75 rlineto
75 0.275 0.75 rlineto
76 -0.275 -0.2 rlineto
77 closepath
78 fill")
79 }
80 \once \override TextSpanner.extra-offset = #(cons 0 (car setting-list))
81 \once \override TextSpanner.bound-details.left.padding = #0.5
82 \once \override TextSpanner.bound-details.right.padding = #-0.25
83 #starting_note
84 \startTextSpan
85 #})
86
87
88 graceNoteAfterBeatOff =
89 #(define-music-function (ending_note) (ly:music?)
90 #{
91 #ending_note
92 \stopTextSpan
93 #})
94
95 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%LSR SNIPPET START%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
96
97 #(define (degrees->radians deg)
98 (* PI (/ deg 180.0)))
99
100 slash =
101 #(define-music-function (ang stem-fraction protrusion)
102 (number? number? number?)
103 (remove-grace-property 'Voice 'Stem 'direction)
104 #{
105 \once \override Stem.stencil =
106 # (lambda (grob)
107 (let* ((x-parent (ly:grob-parent grob X))
108 (is-rest? (ly:grob?
109 (ly:grob-object x-parent 'rest))))

```

```

110         (beam (ly:grob-object grob 'beam))
111         (stil (ly:stem::print grob)))
112     (cond
113       (is-rest? empty-stencil)
114       ((ly:grob? beam)
115        (let* ((refp (ly:grob-system grob))
116               (stem-y-ext (ly:grob-extent grob grob Y))
117               (stem-length
118                (- (cdr stem-y-ext) (car stem-y-ext)))
119               (beam-X-pos (ly:grob-property beam 'X-positions))
120               (beam-Y-pos (ly:grob-property beam 'positions))
121               (beam-slope (/ (- (cdr beam-Y-pos) (car beam-Y-pos))
122                              (- (cdr beam-X-pos) (car beam-X-pos))))
123               (beam-angle (atan beam-slope))
124               (dir (ly:grob-property grob 'direction))
125               (line-dy (* stem-length stem-fraction))
126               (line-dy-with-protrusions (if (= dir 1)
127                                              (+ (* 4 protrusion) beam-angle)
128                                              (- (* 4 protrusion) beam-angle)))
129               (ang (if (> beam-slope 0)
130                       (if (= dir 1)
131                           (+ (degrees->radians ang) (* beam-angle 0.7))
132                           (degrees->radians ang))
133                       (if (= dir -1)
134                           (degrees->radians ang)
135                           (- (degrees->radians ang) (* beam-angle 0.7)))))
136               (line-dx (/ line-dy-with-protrusions (tan ang)))
137               (protrusion-dx (/ protrusion (tan ang)))
138               (corr (if (= dir 1) (car stem-y-ext) (cdr stem-y-ext))))
139       (ly:stencil-add
140        stil
141        (grob-interpret-markup grob
142         (markup
143          #:translate
144          (cons (- protrusion-dx)
145                (+ corr
146                  (* dir
147                    (- stem-length
148                      (+ stem-fraction protrusion))))))
149          #:override '(thickness . 1.7)
150          #:draw-line
151          (cons line-dx
152                (* dir line-dy-with-protrusions))))))
153      (else stil)))
154   #})
155
156 startSlashedGraceMusic = {
157   \slash 40 1 0.5

```

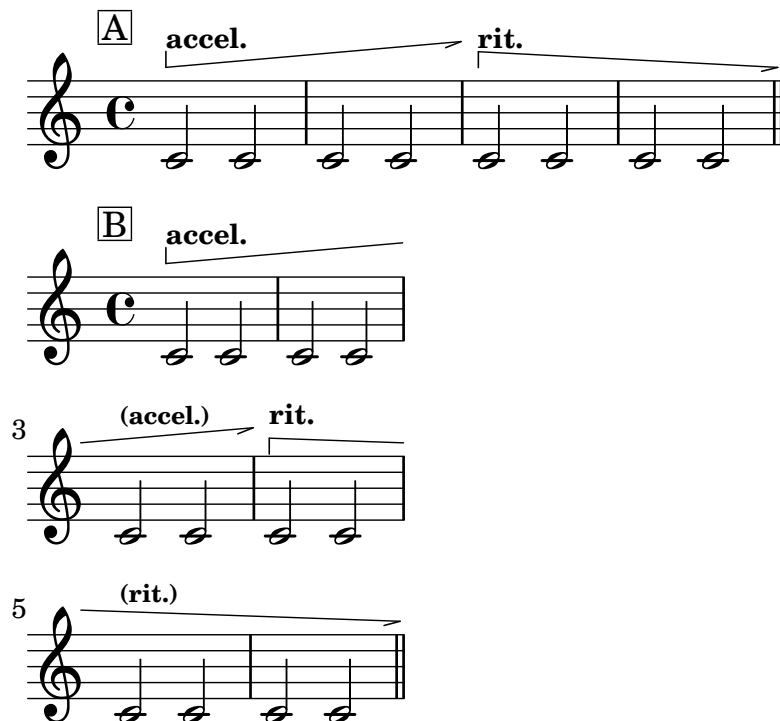
```

158 \override Flag.stroke-style = #"grace"
159 }
160 stopSlashedGraceMusic = {
161 \revert Flag.stroke-style
162 }
163
164 startAcciaccaturaMusic = {
165 \slash 40 1 0.5
166 s1*0(
167 \override Flag.stroke-style = #"grace"
168 }
169 stopAcciaccaturaMusic = {
170 \revert Flag.stroke-style
171 s1*0)
172 }
173 %%%%%%%%%%% LSR SNIPPET END %%%%%%%%%%%
174
175 {
176 \grace {
177 \startSlashedGraceMusic
178 \graceNoteBeforeBeatOn #'(1 -2 -1) d'8^\markup{\translate #'(0 . 3) \box A}
179 e f g e' c'
180 }
181 \graceNoteBeforeBeatOff d''2
182 \grace {
183 \startSlashedGraceMusic
184 \graceNoteAfterBeatOn #'(0 -1 -2) d''8^\markup{\box B} e' f'' g' e'' c'
185 }
186 \graceNoteAfterBeatOff d'2
187 }

```

[Table of Contents](#)

8.8 Tempo Arrows



8.8.1 Description

Replication of accelerando and rallentando arrows chiefly seen in scores by Tōru Takemitsu.⁵ The snippets also handle line break.

8.8.2 Grammar

```
\accelArrow #Line_angle ... \stopTextSpan
\rallArrow #Line_angle ... \stopTextSpan
```

NB

1. `#Line_angle` sets how angled the horizontal line should be. `#5` should be more than sufficient for a short line. When it goes over a line break or it extends for a long time, a smaller number may be recommended, such as `#2`.
2. These commands only set the tempo arrows; as such, indications such as `accel.` and `rall.` need to be added separately.

8.8.3 Code

```
1 \version "2.24.4"
2
```

5. Examples abound, but see: Tōru Takemitsu, *Fantasma/cantos : for clarinet and orchestra* (Schott ; Schott Japan, 1993) and Tōru Takemitsu, *Les yeux clos II : for piano* (Schott ; Schott Japan, 1990) Other composers from the same publishing company, e.g. Toshio Hosokawa, have also adopted variants of the arrows in their music.

```

3 % freely modified from: https://lsr.di.unimi.it/LSR/Item?id=1168
4 % as well as http://lsr.di.unimi.it/LSR/Item?id=1023
5
6
7 accelArrow =
8 #(define-music-function (line_angle) (number?)
9
10     (define x_value (cos (* (/ 3.14159265358979 180) (- 90 line_angle))))
11     (define y_value (sin (* (/ 3.14159265358979 180) (- 90 line_angle))))
12     #{
13         \tweak direction #up
14         \tweak style #'line
15         \tweak thickness #1
16         \tweak to-barline ##t
17         \tweak rotation #(list line_angle -1 0 )
18         \tweak bound-details.left.stencil #ly:text-interface::print
19         \tweak bound-details.left.text \markup \postscript
20         #(string-append
21           "gsave newpath
22 0 0 moveto "
23           (number->string x_value) " "
24           (number->string y_value)
25           " rlineto
26 stroke
27 grestore")
28         \tweak bound-details.left-broken.stencil #ly:text-interface::print
29         \tweak bound-details.left-broken.text ##f
30
31         \tweak bound-details.right.stencil #ly:text-interface::print
32         \tweak bound-details.right.text \markup \postscript
33         "newpath
34 0 0 moveto
35 -1 -0.3 rlineto
36 stroke"
37         \tweak bound-details.right-broken.stencil #ly:text-interface::print
38         \tweak bound-details.right-broken.text ##f
39         \tweak font-shape #'upright
40         \tweak bound-details.left.padding #0
41         \tweak bound-details.right.padding #0
42         \tweak breakable ##t
43         \tweak after-line-breaking ##t
44
45         \startTextSpan
46     #})
47
48 rallArrow =
49 #(define-music-function (line_angle) (number?)
50

```

```

51      (define x_value (cos (* (/ 3.14159265358979 180) (- 90 line_angle))))
52      (define y_value (sin (* (/ 3.14159265358979 180) (- 90 line_angle))))
53      #{
54          \tweak direction #up
55          \tweak style #'line
56          \tweak thickness #1
57          \tweak to-barline ##t
58          \tweak rotation #(list (* -1 line_angle) 1 0 )
59          \tweak bound-details.left.stencil #ly:text-interface::print
60          \tweak bound-details.left.text \markup \postscript
61          #(string-append
62             "gsave
63 newpath
64 0 0 moveto "
65             (number->string x_value) " "
66             (number->string (* -1 y_value))
67             " rlineto
68 stroke
69 grestore")
70          \tweak bound-details.left-broken.stencil #ly:text-interface::print
71          \tweak bound-details.left-broken.text ##f
72
73          \tweak bound-details.right.stencil #ly:text-interface::print
74          \tweak bound-details.right.text \markup \postscript
75          "newpath
76 0 0 moveto
77 -1 -0.3 rlineto
78 stroke"
79          \tweak bound-details.right-broken.stencil #ly:text-interface::print
80          \tweak bound-details.right-broken.text ##f
81          \tweak font-shape #'upright
82          \tweak bound-details.left.padding #0
83          \tweak bound-details.right.padding #0
84          \tweak breakable ##t
85          \tweak after-line-breaking ##t
86
87          \startTextSpan
88          #})
89
90 \score {
91   \layout {
92     indent = 0
93   }
94   {
95     c'2~\markup{\translate #'(-4 . 2) \box "A"}
96     ^\markup {\translate #'(0 . 1.5) \tiny \bold "accel."}
97       \accelArrow #5    c'2
98     c'2 \after 2 \stopTextSpan c'2

```

```

99      c'2 ^\markup {\translate #'(0 . 1.5) \tiny \bold "rit."}
100          \rallArrow #3 c'2
101      c'2 \after 2 \stopTextSpan c'2 \bar "||"
102  }
103 }
104
105 \score {
106   \layout {
107     indent = 0
108     line-width = 40
109   }
110   {
111     c'2^\markup{\translate #'(-4 . 2) \box "B"}
112     ^\markup {\translate #'(0 . 1.5) \tiny \bold "accel."}
113       \accelArrow #5 c'2
114     c'2 c'2
115     c'2^\markup {\translate #'(0 . 1.5) \teeny \bold "(accel.)"}
116       \after 2 \stopTextSpan c'2
117     c'2 ^\markup {\translate #'(0 . 1.5) \tiny \bold "rit."}
118       \rallArrow #2 c'2 \break
119     c'2^\markup {\translate #'(0 . 1.5) \teeny \bold "(rit.)"} c'2
120     c'2 \after 2 \stopTextSpan c'2 \bar "||"
121   }
122 }

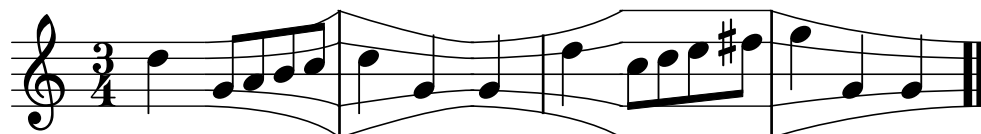
```

[Table of Contents](#)

Chapter 9

Staff Lines

9.1 Expanding, Shrinking and Bloated Staff Lines



9.1.1 Description

I made this code as a proof of concept after having read some excellent snippets on LSR.¹

9.1.2 Grammar

```
\expandingStaff #X-length  
\shrinkingStaff #X-length  
\bloatedStaff  
\normalStaff
```

9.1.3 Code

```
1  
2  shrinkingStaff =  
3  #(define-music-function  
4    (staffDist)  
5    (number?)  
6  
7    #{  
8      \stopStaff  
9      \once \override Staff.StaffSymbol.stencil = #ly:text-interface::print  
10     \once \override Staff.StaffSymbol.text = \markup {
```

1. See: <https://lsr.di.unimi.it/LSR/Item?id=878>, <https://lsr.di.unimi.it/LSR/Item?id=1005>, and <https://lsr.di.unimi.it/LSR/Item?id=1007>.


```

11      \postscript #(string-append
12          "newpath
13          0 4 moveto
14          0 4 6 2 " (number->string staffDist) " 2 curveto
15          0 2 moveto
16          0 2 6 1 " (number->string staffDist) " 1 curveto
17          0 0 moveto "
18          (number->string staffDist) " 0 lineto
19          0 -2 moveto
20          0 -2 6 -1 " (number->string staffDist) " -1 curveto
21          0 -4 moveto
22          0 -4 6 -2 " (number->string staffDist) " -2 curveto
23          stroke")
24
25
26      }
27      \override Staff.StaffSymbol.line-positions = #'(-4 -2 0 2 4 )
28      \startStaff
29      #})
30
31  normalStaff = {
32      \stopStaff
33      \revert Staff.StaffSymbol.line-positions
34      \revert Staff.StaffSymbol.stencil
35      \startStaff
36  }
37
38  expandingStaff =
39  #(define-music-function
40      (staffDist)
41      (number?)
42
43      #{
44
45          \stopStaff
46          \once \override Staff.StaffSymbol.stencil = #ly:text-interface::print
47          \once \override Staff.StaffSymbol.text = \markup {
48              \postscript #(string-append
49                  "newpath
50                  0 2 moveto
51                  0 2 6 2 " (number->string staffDist) " 4 curveto
52                  0 1 moveto
53                  0 1 6 1 " (number->string staffDist) " 2 curveto
54                  0 0 moveto "
55                  (number->string staffDist) " 0 lineto
56                  0 -1 moveto
57                  0 -1 6 -1 " (number->string staffDist) " -2 curveto
58                  0 -2 moveto

```

```

59         0 -2 6 -2 " (number->string staffDist) " -4 curveto
60         stroke ")
61     }
62
63     \startStaff
64     \override Staff.StaffSymbol.line-positions = #'(-8 -4 0 4 8 )
65     #})
66
67     bloatedStaff = {
68         \stopStaff
69         \override Staff.StaffSymbol.line-positions = #'(-8 -4 0 4 8 )
70         \override Staff.LedgerLineSpanner.stencil = ##f
71         \startStaff}
72
73
74
75 % to adjust the length of the individual barlines, see:
76 % https://lilypond.org/doc/v2.24/Documentation/internals/barline
77
78 {
79
80     \override Staff.LedgerLineSpanner.transparent = ##t
81     \numericTimeSignature
82     \time 3/4
83     \once \override Staff.BarLine.bar-extent = #'(-2 . 2)
84     d''4 \expandingStaff #8.5
85
86     g'8 a' b' c''
87     \once \override Staff.BarLine.bar-extent = #'(-4 . 4)
88     \shrinkingStaff #8.5
89     d''4 g' \expandingStaff #9.5 g'
90     \once \override Staff.BarLine.bar-extent = #'(-2.5 . 2.5)
91
92
93     e''4 \bloatedStaff c''8 d'' e'' fs''
94     \once \override Staff.BarLine.bar-extent = #'(-4 . 4)
95
96     \shrinkingStaff #13.5
97
98     g''4 g' g'
99     \bar ".."
100
101 }
102
103 \layout {
104     \context{
105         \Score    proportionalNotationDuration = #(ly:make-moment 1/6)
106     }

```

107 }

108

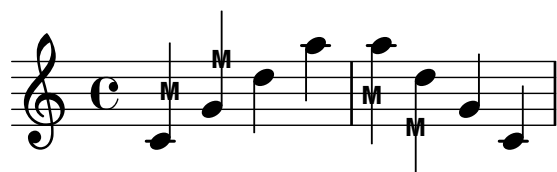
109

Table of Contents

Chapter 10

Stems

10.1 "M" on Stem



10.1.1 Description

This function attaches "M" to the stem. I have used this to indicate **M**ultiphonics on woodwind instruments in my pieces. This function lengthens the stem in order to give a balanced look, especially combined with stems/flags.

10.1.2 Grammar

```
\MOnStemOn NOTE ...  
\MOnStemOff
```

NB `\MOnStemOn` toggles the feature on, while `\MOnStemOff` toggles it off.

10.1.3 Code

```
1 MOnStemOn = {  
2   \override Stem.length = #12  
3   \override Stem.details.beamed-lengths = #'(5.5)  
4   \override Stem.stencil =  
5   #(lambda (grob)  
6     (let* ((x-parent (ly:grob-parent grob X))  
7             (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))  
8       (if is-rest?  
9           empty-stencil  
10          (ly:stencil-combine-at-edge  
11            (ly:stem::print grob)  
12            Y
```

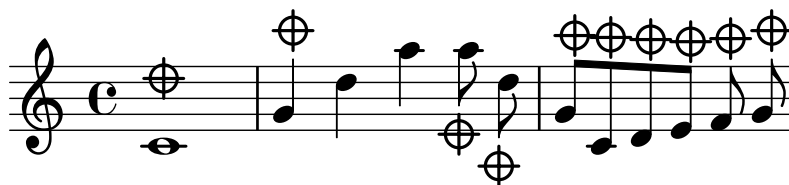
```

13         (- (ly:grob-property grob 'direction))
14         (grob-interpret-markup grob
15             (markup
16                 #:center-align
17                 #:teeny #:sans #:bold "M"))
18         -3.5))))
19     }
20
21     MOnStemOff = {
22         \revert Stem.length
23         \revert Stem.details.beamed-lengths
24         \revert Stem.stencil
25         \revert Flag.stencil
26     }
27
28     {
29         \MOnStemOn c'4 g' \MOnStemOff d'' a''
30         \MOnStemOn a'' d'' \MOnStemOff g' c'
31     }

```

[Table of Contents](#)

10.2 Mute Sign on Stem



10.2.1 Description

This function attaches a mute sign *above/below* the stem.

10.2.2 Grammar

```
\muteSignOnStemOn NOTE ...
\muteSignOnStemOff
```

NB `\muteSignOnStemOn` toggles the feature on, while `\muteSignOnStemOff` toggles it off.

10.2.3 Code

```
1 \version "2.24.4"
2 \pointAndClickOff
3
4 muteSignOnStemOn = {
5
6   % Somewhat rough state; I think I will revisit at a later date.
7
8   % \override Stem.length =
9   % #(lambda (grob)
10  %       (if (= UP (ly:grob-property grob 'direction ))
11  %
12  %   7.5
13  %   7.5))
14
15  % \override Stem.details.beamed-lengths = #'(5.5)
16
17  \override Stem.stencil =
18  #(lambda (grob)
19    (let* ((x-parent (ly:grob-parent grob X))
20           (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
21      (if is-rest?
22          empty-stencil
23
24          (if (= UP (ly:grob-property grob 'direction))
25
26              (ly:stencil-combine-at-edge
27                (ly:stem::print grob)
28
```

```

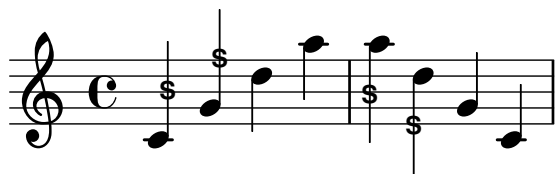
29      (+ (ly:grob-property grob 'direction))
30      (grob-interpret-markup
31      grob
32      (markup
33
34          #:postscript
35          "newpath
36      0.2 setlinewidth
37      1 setlinecap
38      0 0 moveto
39      0 2.5 rlineto
40      -1.25 1.25 moveto
41      2.5 0 rlineto
42      stroke
43      newpath
44      0 1.25 0.85 0 360 arc
45      stroke"
46      ))
47      0.5)
48
49      (ly:stencil-combine-at-edge
50      (ly:stem::print grob)
51      Y
52      (+ (ly:grob-property grob 'direction))
53      (grob-interpret-markup
54      grob
55      (markup
56          #:rotate 180
57          #:postscript
58          "newpath
59      0.2 setlinewidth
60      1 setlinecap
61      0 0 moveto
62      0 2.5 rlineto
63      -1.25 1.25 moveto
64      2.5 0 rlineto
65      stroke
66      newpath
67      0 1.25 0.85 0 360 arc
68      stroke"
69      ))
70      0.5)
71      ))))
72  }
73
74  muteSignOnStemOff = {
75      \revert Stem.length
76      \revert Stem.details.beamed-lengths

```

```
77 \revert Stem.stencil
78 \revert Flag.stencil
79 }
80
81 {
82 \muteSignOnStemOn c'1 g'4 \muteSignOnStemOff d'' a''
83 \muteSignOnStemOn a''8 \noBeam d'' g' c' d' e' f' g'
84 }
```

[Table of Contents](#)

10.3 "S" on Stem



10.3.1 Description

This function attaches "S" to the stem. I have used this to indicate **S**plit tone on clarinet/bass clarinet in my pieces. This function lengthens the stem in order to give a balanced look, especially combined with stems/flags.

10.3.2 Grammar

```
\SOnStemOn NOTE ...
\SOnStemOff
```

NB `\SOnStemOn` toggles the feature on, while `\SOnStemOff` toggles it off.

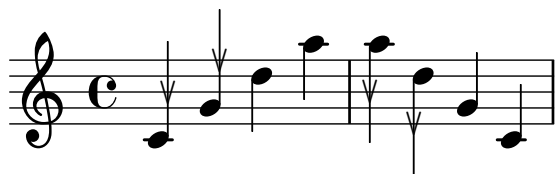
10.3.3 Code

```
1  SOnStemOn = {
2    \override Stem.length = #12
3    \override Stem.details.beamed-lengths = #'(5.5)
4    \override Stem.stencil =
5    #(lambda (grob)
6      (let* ((x-parent (ly:grob-parent grob X))
7              (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
8        (if is-rest?
9            empty-stencil
10           (ly:stencil-combine-at-edge
11             (ly:stem::print grob)
12             Y
13             (- (ly:grob-property grob 'direction))
14             (grob-interpret-markup grob
15                                   (markup
16                                     #:center-align
17                                     #:teeny #:sans #:bold "S")))
18           -3.5))))
19 }
20
21 SOnStemOff = {
22   \revert Stem.length
23   \revert Stem.details.beamed-lengths
24   \revert Stem.stencil
25   \revert Flag.stencil
26 }
27
```

```
28 {  
29   \SOnStemOn c'4 g' \SOnStemOff d'' a''  
30   \SOnStemOn a'' d'' \SOnStemOff g' c'  
31 }
```

[Table of Contents](#)

10.4 "V" on Stem



10.4.1 Description

This function attaches "V" to the stem. I have used this to designate a note with a differentiated timbre from others, for example "brassy tone" for bassoon in my *Gz III* (2019-21) for bass clarinet and bassoon. This function lengthens the stem in order to give a balanced look, especially combined with stems/flags.

10.4.2 Grammar

```
\VOnStemOn NOTE ...
\VOnStemOff
```

NB `\VOnStemOn` toggles the feature on, while `\VOnStemOff` toggles it off.

10.4.3 Code

```
1  \VOnStemOn = {
2    \override Stem.no-stem-extend = ##f
3    \override Stem.length = #12
4    \override Stem.details.beamed-lengths = #'(5.5)
5    \override Stem.stencil =
6    #(lambda (grob)
7      (let* ((x-parent (ly:grob-parent grob X))
8              (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
9        (if is-rest?
10           empty-stencil
11           (ly:stencil-combine-at-edge
12             (ly:stem::print grob)
13             Y
14             (- (ly:grob-property grob 'direction))
15             (grob-interpret-markup grob
16                                   (markup
17                                     #:center-align
18                                     #:teeny #:sans #:musicglyph "scripts.upbow")))
19             -3.5))))
20  }
21
22  \VOnStemOff = {
23    \revert Stem.length
24    \revert Stem.stencil
25    \revert Flag.stencil
26  }
```

```
27
28
29 {
30     \VOnStemOn c'4 g' \VOnStemOff d'' a''
31     \VOnStemOn a'' d'' \VOnStemOff g' c'
32 }
```

[Table of Contents](#)

Chapter 11

Time Signatures

First nine entries of this chapter discuss fractional time signatures (variants of the irrational time signatures) and their compound forms. I have been inspired to create these implementations after chancing upon the email exchanges on `lilypond-user` dated from 2014.¹

While Gould discourages the use of time signatures with numerators as fractions,² there are cases where the use of such time signatures seems justified, particularly when the fractions deal with some form of tuplets. This is a form of time signature notation widely seen in works by Chaya Czernowin, Stefan Beyer, myself, and so many others.

I present the implementation of fractional time signatures in three different styles, A, B, and C. There are implementations for compound meters for each of the styles, in two and three time signatures.

11.1 Fractional Time Signatures, Style A



1. See: <https://lists.gnu.org/archive/html/lilypond-user/2014-06/msg00209.html>. However, in the process of writing this documentation I have come across another email thread on the same mailing list: <https://mail.gnu.org/archive/html/lilypond-user/2020-04/msg00423.html>

2. Gould, *Behind bars : the definitive guide to music notation*, 180.

11.1.1 Description

This particular style of fractional time signatures³ can be seen in scores by Stefan Beyer, for example *Marsch* (2013-14),⁴ *Mittel und Zwecke (Boulevard)* (2014),⁵ *Bleib hier. Schau zu. Mach kein Geräusch.* (2017),⁶ and *Most of My Clients Come Back* (2012-13).⁷ In the case of *Mass und Gewicht* (2021), Beyer uses fractions on the denominator of the time signatures.⁸

Because the size the fractions is a half of the ordinary time signatures, it may be difficult to see from afar.⁹

11.1.2 Grammar

```
\fractionalTimeSignatureA
    #'(NUM1 NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureA
    #'(NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureAPlus
    #'(NUM1 NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureAPlus
    #'(NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT

\backToNormalTimeSignature
```

NB

1. `\fractionalTimeSignatureA` lists time signatures *without* the use of the + (plus) sign.
2. `\fractionalTimeSignatureAPlus` lists time signatures with the + (plus) sign, when the list with four NUMs are given.
3. NUM1, NUM2, NUM3, and NUM4 can be understood as follows:

$$\frac{1 + \frac{2}{3}}{4}$$

where NUM1 is optional. The code has `cond` clause, which adjusts the appearance of the time signature according to the length of the list, either having 3 or 4 numbers.

4. `MEASURE_SPAN` denotes how the measure may be written using an *irrational time signature*. In the example snippet, this would be:

3. After having come up with this code, there were other implementations that could be seen on this email thread: <https://mail.gnu.org/archive/html/lilypond-user/2020-04/msg00423.html>

4. Stefan Beyer, *Marsch* (Manuscript, 2013-14).

5. Stefan Beyer, *Mittel und Zwecke (Boulevard)* (Manuscript, 2014).

6. Stefan Beyer, *Bleib hier. Schau zu. Mach kein Geräusch.* (Manuscript, 2017).

7. Stefan Beyer, *Most of My Clients Come Back* (Manuscript, 2012-13).

8. It would be relatively easy to modify the Scheme code so that the fraction appears next to the denominator of the time signature, instead.

9. It should be noted that in other works such as Lotte Reiniger's *The Sleeping Beauty* (2020-21), Beyer also uses the irrational time signatures as seen in the *Incomplete Tuplet Bracket for Irrational Time Signatures* section of this document.

$$\frac{3}{4} + \frac{2}{12} = \frac{11}{12}$$

5. BEAT_STRUCT indicates beat structure, by which the beaming of the measure abides.
6. When you wish to go back to a regular time signature, use `\backToNormalTimeSignature`, otherwise the identical fractional time signature will keep showing up.

11.1.3 Code

```

1 % Inspired by:
2 % https://lists.gnu.org/archive/html/lilypond-user/2014-06/msg00209.html
3
4 % Revised Aug 10 2025 to include the function to revert to a
5 % regular time signature
6
7 \version "2.24.4"
8 \language "english"
9
10 suppressWarning =
11 #(define-void-function (amount message)(number? string?)
12   (for-each
13     (lambda (warning)
14       (ly:expect-warning message))
15     (iota amount 1 1)))
16
17 \suppressWarning 3 "strange time signature found"
18
19 incompleteTupletBracket = {
20   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
21   \once \override Voice.TupletBracket.bracket-visibility = ##t
22
23 }
24 incompleteSmallTupletBracket = {
25   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
26   \once \override Voice.TupletBracket.bracket-visibility = ##t
27   \once \override Voice.TupletNumber.X-offset =
28     #(lambda (grob)
29       (if (= UP (ly:grob-property grob 'direction))
30         2.2
31         1.2))
32
33   \once \override Voice.TupletBracket.shorten-pair =
34     #(lambda (grob)
35       (if (= UP (ly:grob-property grob 'direction))
36         '(-0.7 . 0.15)
37         '(-0.3 . 0.8)))
38   \once \override Voice.TupletBracket.X-positions =

```

```

39  #(lambda (grob)
40    (if (= (ly:grob-property grob 'direction))
41        '(1.8 . 3)
42        '(0.3 . 2.7)))
43  }
44
45  fractionalTimeSignatureA =
46  #(define-music-function
47    (timeSignatureToShow underlyingMeter beatStructure)
48    (list? fraction? number-list?)
49    #{
50      \time $underlyingMeter
51      \set beatStructure = $beatStructure
52
53      \override Staff.TimeSignature.stencil =
54      #ly:text-interface::print
55
56      \override Staff.TimeSignature.text =
57      #(if (= (length timeSignatureToShow) 4)
58
59          (markup
60            #:override
61            (cons 'baseline-skip 0)
62            (#:center-column
63              (#:number
64                (#:concat
65                  (#:simple
66                    (number->string (car timeSignatureToShow))
67                    #:halign -1.5
68                    (#:center-column
69                      ((#:translate
70                        (cons 0 1)
71                        (#:fontsize -6
72                          (number->string
73                            (cadr timeSignatureToShow))))
74                      (#:translate
75                        (cons 0 0)
76                        (#:fontsize -6
77                          (number->string
78                            (caddr timeSignatureToShow))))))
79                      #:number
80                      (number->string (caddr timeSignatureToShow))))
81
82          (markup
83            #:override
84            (cons 'baseline-skip 0)
85            (#:center-column
86              (#:number

```



```

87         (:translate
88         (cons 0 1)
89         (:fontsize -6 (number->string
90                       (car timeSignatureToShow))))
91     #:number
92     (:translate
93     (cons 0 0)
94     (:fontsize -6 (number->string
95                   (cadr timeSignatureToShow))))
96     #:number
97     (number->string (caddr timeSignatureToShow))))
98
99     )
100 #})
101
102 fractionalTimeSignatureAPlus =
103 #(define-music-function
104   (timeSignatureToShow underlyingMeter beatStructure)
105   (list? fraction? number-list?)
106   #{
107     \time $underlyingMeter
108     \set beatStructure = $beatStructure
109
110     \override Staff.TimeSignature.stencil =
111     #ly:text-interface::print
112
113     \override Staff.TimeSignature.text =
114     #(if (= (length timeSignatureToShow) 4)
115
116         (markup
117         #:override
118         (cons 'baseline-skip 0)
119         (:center-column
120         (:number
121         (:concat
122         (:simple
123         (number->string (car timeSignatureToShow))
124
125         (:fontsize -12 (string-append " "))
126         (string-append "+")
127         (:fontsize -12 (string-append " ")))
128
129         #:center-column
130         ((:translate
131         (cons 0 1)
132         (:fontsize -6
133         (number->string
134         (cadr timeSignatureToShow))))

```

```

135         (:translate
136         (cons 0 0)
137         (:fontsize -6
138         (number->string
139         (caddr timeSignatureToShow))))))
140     #:number
141     (number->string (caddr timeSignatureToShow)))
142
143 (markup
144   #:override
145   (cons 'baseline-skip 0)
146   (:center-column
147     (:number
148       (:translate
149       (cons 0 1)
150       (:fontsize -6 (number->string
151       (car timeSignatureToShow))))
152     #:number
153     (:translate
154     (cons 0 0)
155     (:fontsize -6 (number->string
156     (cadr timeSignatureToShow))))
157     #:number
158     (number->string (caddr timeSignatureToShow))))
159   )
160   #})
161
162
163
164 backToNormalTimeSignature =
165 {
166   \unset beatStructure
167   \revert Timing.TimeSignature.stencil
168   \revert Timing.TimeSignature.text
169   \revert Staff.TimeSignature.stencil
170   \revert Staff.TimeSignature.text
171 }
172
173
174 \new Staff \with { instrumentName = \markup {\fontsize #4 \box "A"}} {
175   \fractionalTimeSignatureA #'(3 2 3 4) 11/12 3,3,3,2
176   \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
177   \tuplet 3/2 {c' c' c'}
178   \incompleteTupletBracket \tuplet 3/2 {c' c'}
179   \backToNormalTimeSignature
180   \time 3/4
181   c'2.
182 }

```

```

183 \new Staff \with { instrumentName = \markup {\fontsize #4 \box "B"}} {
184   \fractionalTimeSignatureAPlus #'(3 2 3 4) 11/12 3,3,3,2
185   \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
186   \tuplet 3/2 {c' c' c'}
187   \incompleteTupletBracket \tuplet 3/2 {c' c'}
188   \backToNormalTimeSignature
189   \time 3/4
190   c'2.
191 }
192 \new Staff \with { instrumentName = \markup {\fontsize #4 \box "C"}} {
193   \fractionalTimeSignatureA #'(11 3 4) 11/12 3,3,3,2
194   \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
195   \tuplet 3/2 {c' c' c'}
196   \incompleteTupletBracket \tuplet 3/2 {c' c'}
197   \backToNormalTimeSignature
198   \time 3/4
199   c'2.
200 }
201

```

[Table of Contents](#)

11.2 Fractional Time Signatures, Style B



11.2.1 Description

Style B differs from Style A, as the fraction has a bigger font size. This is similar to the design of fractional time signatures I have used in works such as *Gz II* (2017-22).¹⁰

11.2.2 Grammar

```
\fractionalTimeSignatureB
    #'(NUM1 NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureB
    #'(NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureBPlus
    #'(NUM1 NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureBPlus
    #'(NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT

\backToNormalTimeSignature
```

NB

1. `\fractionalTimeSignatureB` lists time signatures *without* the use of the + (plus) sign.
2. `\fractionalTimeSignatureBPlus` lists time signatures with the + (plus) sign, when the list with four NUMs are given.
3. See [Grammar of Fractional Time Signatures, Style A](#) for the explanation on the arguments.
4. When you wish to go back to a regular time signature, use `\backToNormalTimeSignature`, otherwise the identical fractional time signature will keep showing up.

10. Yoshiaki Onishi, *Gz II : for two accordions* (Brühl and Berlin: Edition Gravis, 2024).

11.2.3 Code

```

1  \version "2.24.4"
2  \language "english"
3
4  % Revised Jan 2 2025 for improving the appearance of fractions
5  % Revised Aug 10 2025 to include the function to revert to a
6  % regular time signature
7
8  suppressWarning =
9  #(define-void-function (amount message)(number? string?)
10    (for-each
11      (lambda (warning)
12        (ly:expect-warning message))
13      (iota amount 1 1)))
14
15  \suppressWarning 3 "strange time signature found"
16
17  incompleteTupletBracket = {
18    \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
19    \once \override Voice.TupletBracket.bracket-visibility = ##t
20
21  }
22  incompleteSmallTupletBracket = {
23    \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
24    \once \override Voice.TupletBracket.bracket-visibility = ##t
25    \once \override Voice.TupletNumber.X-offset =
26    #(lambda (grob)
27      (if (= UP (ly:grob-property grob 'direction))
28          2.2
29          1.2))
30
31    \once \override Voice.TupletBracket.shorten-pair =
32    #(lambda (grob)
33      (if (= UP (ly:grob-property grob 'direction))
34          '(-0.7 . 0.15)
35          '(-0.3 . 0.8)))
36    \once \override Voice.TupletBracket.X-positions =
37    #(lambda (grob)
38      (if (= UP (ly:grob-property grob 'direction))
39          '(1.8 . 3)
40          '(0.3 . 2.7)))
41  }
42
43  fractionalTimeSignatureB =
44  #(define-music-function
45    (timeSignatureToShow underlyingMeter beatStructure)
46    (list? fraction? number-list?)

```

```

47   #{
48
49   \time $underlyingMeter
50   \set beatStructure = $beatStructure
51
52   \override Staff.TimeSignature.stencil =
53   #ly:text-interface::print
54   \override Staff.TimeSignature.text =
55
56   #(if (= (length timeSignatureToShow) 4)
57
58       (markup
59         (make-override-markup
60           (cons 'baseline-skip 0)
61           (make-center-column-markup
62             (list
63               (make-line-markup
64                 (list
65                   (make-number-markup
66                     (number->string (car timeSignatureToShow)))
67
68
69                 (make-hspace-markup -0.5)
70                 (make-right-align-markup
71                   (make-number-markup
72                     (make-translate-markup
73                       (cons 0 1.5)
74                       (make-fontsize-markup
75                         -3
76                         (number->string (cadr timeSignatureToShow))))))
77
78                 (make-hspace-markup -1.5)
79
80                 (make-override-markup
81                   (cons 'alignment 0)
82                   (make-translate-markup
83                     (cons 0 0.8)
84                     (make-draw-line-markup (cons 1.5 1.35))))))
85
86                 (make-hspace-markup -1.5)
87
88                 (make-number-markup
89                   (make-left-align-markup
90                     (make-fontsize-markup
91                       -3
92                       (number->string (caddr timeSignatureToShow))))))
93
94       (make-number-markup

```

```

95         (number->string (caddr timeSignatureToShow))))))
96
97
98     (markup
99     (make-override-markup
100     (cons 'baseline-skip 0)
101     (make-center-column-markup
102     (list
103     (make-line-markup
104     (list
105     (make-number-markup
106     (make-right-align-markup
107     (make-translate-markup
108     (cons 0 1.5)
109     (make-fontsize-markup
110     -3
111     (number->string (car timeSignatureToShow))))))
112
113     (make-hspace-markup -1.5)
114
115     (make-override-markup
116     (cons 'alignment 0)
117     (make-translate-markup
118     (cons 0 0.8)
119     (make-draw-line-markup (cons 1.5 1.35))))
120
121     (make-hspace-markup -1.5)
122
123     (make-translate-markup
124     (cons 0 0)
125     (make-fontsize-markup
126     -3
127     (make-number-markup
128     (number->string (cadr timeSignatureToShow))))))
129
130     (make-number-markup
131     (number->string (caddr timeSignatureToShow))))))
132
133     )
134     #})
135
136 fractionalTimeSignatureBPlus =
137 #(define-music-function
138   (timeSignatureToShow underlyingMeter beatStructure)
139   (list? fraction? number-list?)
140   #{
141
142     \time $underlyingMeter

```

```

143 \set beatStructure = $beatStructure
144
145 \override Staff.TimeSignature.stencil =
146 #ly:text-interface::print
147 \override Staff.TimeSignature.text =
148
149 #(if (= (length timeSignatureToShow) 4)
150
151
152 (markup
153   (make-override-markup
154     (cons 'baseline-skip 0)
155     (make-center-column-markup
156       (list
157         (make-line-markup
158           (list
159             (make-number-markup
160               (number->string (car timeSignatureToShow)))
161             (make-fontsize-markup
162               -12
163               (make-simple-markup " "))
164
165             (make-hspace-markup -1.25)
166             (make-translate-markup
167               (cons 0 0.4)
168               (make-bold-markup
169                 (make-simple-markup "+")))
170
171             (make-hspace-markup -0.25)
172
173             (make-hspace-markup -0.5)
174             (make-right-align-markup
175               (make-number-markup
176                 (make-translate-markup
177                   (cons 0 1.5)
178                   (make-fontsize-markup
179                     -3
180                     (number->string (cadr timeSignatureToShow))))))
181
182             (make-hspace-markup -1.5)
183
184             (make-override-markup
185               (cons 'alignment 0)
186               (make-translate-markup

```



```

191         (cons 0 0.8)
192         (make-draw-line-markup (cons 1.5 1.35))))
193
194     (make-hspace-markup -1.5)
195
196     (make-number-markup
197       (make-left-align-markup
198         (make-fontsize-markup
199           -3
200           (number->string (caddr timeSignatureToShow))))))
201
202     (make-number-markup
203       (number->string (caddr timeSignatureToShow))))
204
205
206 (markup
207   (make-override-markup
208     (cons 'baseline-skip 0)
209     (make-center-column-markup
210       (list
211         (make-line-markup
212           (list
213             (make-number-markup
214               (make-right-align-markup
215                 (make-translate-markup
216                   (cons 0 1.6)
217                   (make-fontsize-markup
218                     -3
219                     (number->string (car timeSignatureToShow))))))
220
221             (make-hspace-markup -1.5)
222
223             (make-override-markup
224               (cons 'alignment 0)
225               (make-translate-markup
226                 (cons 0 0.8)
227                 (make-draw-line-markup (cons 1.5 1.35))))
228
229             (make-hspace-markup -1.5)
230
231             (make-translate-markup
232               (cons 0 0)
233               (make-fontsize-markup
234                 -3
235                 (make-number-markup
236                   (number->string (cadr timeSignatureToShow))))))
237
238             (make-number-markup

```

```

239         (number->string (caddr timeSignatureToShow)))))))))
240     )
241     #})
242
243     backToNormalTimeSignature =
244     {
245         \unset beatStructure
246         \revert Timing.TimeSignature.stencil
247         \revert Timing.TimeSignature.text
248         \revert Staff.TimeSignature.stencil
249         \revert Staff.TimeSignature.text
250     }
251
252
253     {
254         \fractionalTimeSignatureB #'(1 2 3 4) 11/12 3,3,3,2
255         \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
256         \tuplet 3/2 {c' c' c'}
257         \incompleteTupletBracket \tuplet 3/2 {c' c'}
258         \backToNormalTimeSignature
259         \time 3/4
260         c'2.
261     }
262
263     {
264         \fractionalTimeSignatureBPlus #'(3 2 3 4) 11/12 3,3,3,2
265         \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
266         \tuplet 3/2 {c' c' c'}
267         \incompleteTupletBracket \tuplet 3/2 {c' c'}
268         \backToNormalTimeSignature
269         \time 3/4
270         c'2.
271     }
272
273     {
274         \fractionalTimeSignatureB #'(11 3 4) 11/12 3,3,3,2
275         \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
276         \tuplet 3/2 {c' c' c'}
277         \incompleteTupletBracket \tuplet 3/2 {c' c'}
278         \backToNormalTimeSignature
279         \time 3/4
280         c'2.
281     }

```

[Table of Contents](#)

11.3 Fractional Time Signatures, Style C



11.3.1 Description

Style C of the fractional time signatures offers the largest font size for displaying the fractions. This design is commonly seen in scores by Chaya Czernowin, in such works as *String Quartet* (1995),¹¹ *Lovesong* (2010),¹² *Streams (Slow Summer Stay I)* (2012),¹³ and *At the fringe of our gaze* (2012/13).¹⁴

11.3.2 Grammar

```
\fractionalTimeSignatureC
    #'(NUM1 NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT
\fractionalTimeSignatureC
    #'(NUM2 NUM3 NUM4) MEASURE_SPAN BEAT_STRUCT

\backToNormalTimeSignature
```

NB

1. By default, `\fractionalTimeSignatureC` shows + (plus) sign when four NUMs are given. As the font size for the ordinary numerator and the fractions is the same, without + it becomes very confusing to read the time signature. Thus, contrary to Styles A and B, there is no separate function for the time signature with the + sign given.
2. See [Grammar of Fractional Time Signatures, Style A](#) for the explanation on the arguments.
3. When you wish to go back to a regular time signature, use `\backToNormalTimeSignature`, otherwise the identical fractional time signature will keep showing up.

11.3.3 Code

```
1 \version "2.24.4"
2 \language "english"
3
4
```

11. Chaya Czernowin, *String Quartet* (Schott, 1995).

12. Chaya Czernowin, *Lovesong : for mixed ensemble* (Schott, 2010).

13. Chaya Czernowin, *Streams (Slow Summer Stay I) : for 8 players* (Schott, 2012).

14. Chaya Czernowin, *At the fringe of our gaze : for Orchestra and Concertino Group* (Schott, 2012/13).

```

5 % Revised Jan 2 2025 for improving the appearance of fractions
6 % Revised Aug 10 2025 to include the function to revert to a
7 % regular time signature
8
9 suppressWarning =
10 #(define-void-function (amount message)(number? string?)
11   (for-each
12     (lambda (warning)
13       (ly:expect-warning message))
14     (iota amount 1 1)))
15
16 \suppressWarning 2 "strange time signature found"
17
18 incompleteTupletBracket = {
19   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
20   \once \override Voice.TupletBracket.bracket-visibility = ##t
21
22 }
23 incompleteSmallTupletBracket = {
24   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
25   \once \override Voice.TupletBracket.bracket-visibility = ##t
26   \once \override Voice.TupletNumber.X-offset =
27     #(lambda (grob)
28       (if (= UP (ly:grob-property grob 'direction))
29         2.2
30         1.2))
31
32   \once \override Voice.TupletBracket.shorten-pair =
33     #(lambda (grob)
34       (if (= UP (ly:grob-property grob 'direction))
35         '(-0.7 . 0.15)
36         '(-0.3 . 0.8)))
37   \once \override Voice.TupletBracket.X-positions =
38     #(lambda (grob)
39       (if (= UP (ly:grob-property grob 'direction))
40         '(1.8 . 3)
41         '(0.3 . 2.7)))
42 }
43
44 fractionalTimeSignatureC =
45 #(define-music-function
46   (timeSignatureToShow underlyingMeter beatStructure)
47   (list? fraction? number-list?)
48   #{
49
50     \time $underlyingMeter
51     \set beatStructure = $beatStructure
52

```



```

101
102      (make-number-markup
103        (number->string
104          (caddr timeSignatureToShow))))))
105
106
107      (markup
108        (make-override-markup
109          (cons 'baseline-skip 0)
110          (make-center-column-markup
111            (list
112              (make-line-markup
113                (list
114
115                  (make-right-align-markup
116                    (make-number-markup
117                      (number->string
118                        (car timeSignatureToShow))))
119
120                    (make-hspace-markup -0.6)
121
122                    (make-override-markup
123                      (list (cons 'alignment 0)
124                            (cons 'thickness 2))
125                      (make-draw-line-markup
126                        (cons 0.5 2)))
127
128                    (make-hspace-markup -0.6)
129
130                    (make-number-markup
131                      (make-left-align-markup
132                        (number->string
133                          (cadr timeSignatureToShow))))))
134
135                    (make-number-markup
136                      (number->string
137                        (caddr timeSignatureToShow))))))
138          ))
139      #})
140
141 backToNormalTimeSignature =
142 {
143   \unset beatStructure
144   \revert Timing.TimeSignature.stencil
145   \revert Timing.TimeSignature.text
146   \revert Staff.TimeSignature.stencil
147   \revert Staff.TimeSignature.text
148 }

```

```

149
150
151 {
152   \fractionalTimeSignatureC #'(3 2 3 4) 11/12 3,3,3,2
153   \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
154   \tuplet 3/2 {c' c' c'}
155   \incompleteTupletBracket \tuplet 3/2 {c' c'}
156
157   \backToNormalTimeSignature
158   \time 3/4
159   c'2.
160 }
161
162
163 {
164   \fractionalTimeSignatureC #'(11 3 4) 11/12 3,3,3,2
165   \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
166   \tuplet 3/2 {c' c' c'}
167   \incompleteTupletBracket \tuplet 3/2 {c' c'}
168   \backToNormalTimeSignature
169   \time 3/4
170   c'2.
171 }

```

[Table of Contents](#)

11.4 Compound Meter with Two Fractional Time Signatures, Style A



11.4.1 Description

This is an implementation of a compound meter with two fractional time signatures with Style A.

11.4.2 Grammar

```
\compoundFractionalTimeSignatureATwo
    #'((TIME_SIG1)(TIME_SIG2)) MEASURE_SPAN BEAT_STRUCT

\backToNormalTimeSignature
```

NB

1. Following the convention of `\compoundMeter` to enter the two time signatures, you will create a list of lists. Each `TIME_SIG` accepts:
 - an ordinary time signature (list with two numbers);
 - a time signature with a fraction (list with three numbers), or;
 - a time signature with an ordinary numerator and a fraction.

See [Grammar of Fractional Time Signatures, Style A](#) for the explanation on the arguments for the order of arguments to specify time signatures.
2. `MEASURE_SPAN` and `BEAT_STRUCT` follow the same convention as before.
3. When you wish to go back to a regular time signature, use `\backToNormalTimeSignature`, otherwise the identical fractional time signature will keep showing up.

11.4.3 Code

```
1 \version "2.24.4"
2 \language "english"
3
4 % Revised Aug 10 2025 to include the function to revert to a
5 % regular time signature
6
7 suppressWarning =
8 #(define-void-function (amount message)(number? string?)
9   (for-each
10     (lambda (warning)
```



```

11      (ly:expect-warning message))
12      (iota amount 1 1)))
13
14 \suppressWarning 1 "strange time signature found"
15
16 incompleteTupletBracket = {
17   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
18   \once \override Voice.TupletBracket.bracket-visibility = ##t
19
20 }
21 incompleteSmallTupletBracket = {
22   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
23   \once \override Voice.TupletBracket.bracket-visibility = ##t
24   \once \override Voice.TupletNumber.X-offset =
25     #(lambda (grob)
26       (if (= UP (ly:grob-property grob 'direction))
27           2.2
28           1.2))
29
30   \once \override Voice.TupletBracket.shorten-pair =
31     #(lambda (grob)
32       (if (= UP (ly:grob-property grob 'direction))
33           '(-0.7 . 0.15)
34           '(-0.3 . 0.8)))
35   \once \override Voice.TupletBracket.X-positions =
36     #(lambda (grob)
37       (if (= UP (ly:grob-property grob 'direction))
38           '(1.8 . 3)
39           '(0.3 . 2.7)))
40 }
41
42 compoundFractionalTimeSignatureATwo =
43 #define-music-function
44 (timeSignatureToShow underlyingMeter beatStructure)
45 (list? fraction? number-list?)
46 (define mkup
47 (markup
48   #:concat
49   (
50     #:override
51     (cons 'baseline-skip 0)
52     (cond ((= (length (car timeSignatureToShow)) 2)
53           (make-center-column-markup
54             (list (make-number-markup
55                     (number->string
56                     (car (car timeSignatureToShow))))
57                   (make-number-markup
58                     (number->string

```

```

59             (cadr (car timeSignatureToShow)))))))))
60
61     ((= (length (car timeSignatureToShow)) 3)
62       (make-override-markup
63         (cons 'baseline-skip 0)
64         (make-center-column-markup
65           (list
66
67             (make-center-column-markup
68               (list
69                 (make-translate-markup
70                   (cons 0 1)
71                   (make-fontsize-markup
72                     -6
73                     (make-number-markup
74                       (number->string
75                         (car (car timeSignatureToShow))))))
76                 (make-translate-markup
77                   (cons 0 0)
78                   (make-fontsize-markup
79                     -6
80                     (make-number-markup
81                       (number->string
82                         (cadr (car timeSignatureToShow))))))
83                 (make-number-markup
84                   (number->string
85                     (caddr (car timeSignatureToShow))))))
86               ))
87
88
89
90     ((= (length (car timeSignatureToShow)) 4)
91
92       (make-override-markup
93         (cons 'baseline-skip 0)
94         (make-center-column-markup
95           (list
96
97             (make-concat-markup
98               (list (make-number-markup
99                 (number->string
100                   (car (car timeSignatureToShow))))
101                 (make-halign-markup
102                   -1.5
103                   (make-center-column-markup
104                     (list
105                       (make-translate-markup
106                         (cons 0 1)

```

```

107             (make-fontsize-markup
108               -6
109               (make-number-markup
110                 (number->string
111                   (cadr (car timeSignatureToShow))))))
112             (make-translate-markup
113               (cons 0 0)
114               (make-fontsize-markup
115                 -6
116                 (make-number-markup
117                   (number->string
118                     (caddr (car timeSignatureToShow)))))))))
119         (make-number-markup
120           (number->string
121             (caddr (car timeSignatureToShow))))))
122     ))
123 )
124
125
126 #:translate
127 (cons 0 -0.5)
128 (#:fontsize -12 " ")
129 #:translate
130 (cons 0 -0.5)
131 (#:bold "+")
132 #:translate
133 (cons 0 -0.5)
134 (#:fontsize -12 " ")
135
136 #:override
137 (cons 'baseline-skip 0)
138 (cond ((= (length (cadr timeSignatureToShow)) 2)
139        (make-center-column-markup
140          (list (make-number-markup
141                  (number->string
142                    (car (cadr timeSignatureToShow))))
143                (make-number-markup
144                  (number->string
145                    (cadr (cadr timeSignatureToShow)))))))
146        ((= (length (cadr timeSignatureToShow)) 3)
147          (make-override-markup
148            (cons 'baseline-skip 0)
149            (make-center-column-markup
150              (list

```

```

155         (make-center-column-markup
156         (list
157         (make-translate-markup
158         (cons 0 1)
159         (make-fontsize-markup
160         -6
161         (make-number-markup
162         (number->string
163         (car (cadr timeSignatureToShow))))))
164         (make-translate-markup
165         (cons 0 0)
166         (make-fontsize-markup
167         -6
168         (make-number-markup
169         (number->string
170         (cadr (cadr timeSignatureToShow))))))
171         (make-number-markup
172         (number->string
173         (caddr (cadr timeSignatureToShow))))))
174     ))
175
176
177     ((= (length (cadr timeSignatureToShow)) 4)
178
179     (make-override-markup
180     (cons 'baseline-skip 0)
181     (make-center-column-markup
182     (list
183
184     (make-concat-markup
185     (list (make-number-markup
186     (number->string
187     (car (cadr timeSignatureToShow))))
188     (make-halign-markup
189     -1.5
190     (make-center-column-markup
191     (list
192     (make-translate-markup
193     (cons 0 1)
194     (make-fontsize-markup
195     -6
196     (make-number-markup
197     (number->string
198     (cadr (cadr timeSignatureToShow))))))
199     (make-translate-markup
200     (cons 0 0)
201     (make-fontsize-markup
202     -6

```

```

203             (make-number-markup
204               (number->string
205                 (caddr (cadr timeSignatureToShow)))))))))
206         (make-number-markup
207           (number->string
208             (caddr (cadr timeSignatureToShow))))))
209       ))
210     )
211   )))
212
213   #{
214     \time $underlyingMeter
215     \set beatStructure = $beatStructure
216
217     \override Timing.TimeSignature.stencil =
218     #ly:text-interface::print
219     \override Timing.TimeSignature.text =
220     #mkup
221     #})
222
223
224   backToNormalTimeSignature =
225   {
226     \unset beatStructure
227     \revert Timing.TimeSignature.stencil
228     \revert Timing.TimeSignature.text
229     \revert Staff.TimeSignature.stencil
230     \revert Staff.TimeSignature.text
231   }
232
233
234   {
235     \compoundFractionalTimeSignatureATwo #'((3 4)(2 3 4)) 11/12 3,3,3,2
236     \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
237     \tuplet 3/2 {c' c' c'}
238     \incompleteTupletBracket \tuplet 3/2 {c' c'}
239     \backToNormalTimeSignature
240     \time 3/4
241     c'2.
242   }

```

11.4.4 Discussion

1. This was a tricky one to make, as I had to resort to building the Scheme code without using the syntactic sugars, i.e. `#:`.¹⁵ If any modification are to be made to this code, it is recommended to carefully examine where the corresponding parenthesis of a starting parenthesis is located.

15. See *Known issues and warnings* at: <https://lilypond.org/doc/v2.24/Documentation/extending/markup-construction-in-scheme>

It is also helpful to watch LilyPond Log for any errors, as it seems to give hints for how many argument(s) a function is looking for.

2. I am hoping to find ways to simplify the code, as the same bits (with variations in variables that are called upon) of the codes are used to streamline the formatting of the time signature appearances.
3. When you wish to go back to a regular time signature, use `\backToNormalTimeSignature`, otherwise the identical fractional time signature will keep showing up.

[Table of Contents](#)

11.5 Compound Meter with Two Fractional Time Signatures, Style B



11.5.1 Description

This is an implementation of a compound meter with two fractional time signatures with Style B.

11.5.2 Grammar

```
\compoundFractionalTimeSignatureBTwo
    #'((TIME_SIG1)(TIME_SIG2)) MEASURE_SPAN BEAT_STRUCT

\backToNormalTimeSignature
```

NB

1. Following the convention of `\compoundMeter` to enter the two time signatures, you will create a list of lists. Each `TIME_SIG` accepts: an ordinary time signature (list with two numbers), a time signature with a fraction (list with three numbers), or a time signature with an ordinary numerator and a fraction. See [Grammar of Fractional Time Signatures, Style A](#) for the explanation on the arguments for the order of arguments to specify time signatures.
2. `MEASURE_SPAN` and `BEAT_STRUCT` follow the same convention as before.
3. When you wish to go back to a regular time signature, use `\backToNormalTimeSignature`, otherwise the identical fractional time signature will keep showing up.

11.5.3 Code

```
1 \version "2.24.4"
2 \language "english"
3
4
5 % Revised Jan 2 2025 for improving the appearance of fractions
6 % Revised Aug 10 2025 to include the function to revert to a
7 % regular time signature
8
9 suppressWarning =
10 #(define-void-function (amount message)(number? string?)
11   (for-each
12     (lambda (warning)
13       (ly:expect-warning message))
14     (iota amount 1 1)))
```

```

15
16 \suppressWarning 1 "strange time signature found"
17
18 incompleteTupletBracket = {
19   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
20   \once \override Voice.TupletBracket.bracket-visibility = ##t
21
22 }
23 incompleteSmallTupletBracket = {
24   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
25   \once \override Voice.TupletBracket.bracket-visibility = ##t
26   \once \override Voice.TupletNumber.X-offset =
27     #(lambda (grob)
28       (if (= UP (ly:grob-property grob 'direction))
29           2.2
30           1.2))
31
32   \once \override Voice.TupletBracket.shorten-pair =
33     #(lambda (grob)
34       (if (= UP (ly:grob-property grob 'direction))
35           '(-0.7 . 0.15)
36           '(-0.3 . 0.8)))
37   \once \override Voice.TupletBracket.X-positions =
38     #(lambda (grob)
39       (if (= UP (ly:grob-property grob 'direction))
40           '(1.8 . 3)
41           '(0.3 . 2.7)))
42 }
43
44 compoundFractionalTimeSignatureBTwo =
45 #(define-music-function
46   (timeSignatureToShow underlyingMeter beatStructure)
47   (list? fraction? number-list?)
48   (define mkup
49     (markup
50       #:concat
51       (
52
53         #:override
54         (cons 'baseline-skip 0)
55         (cond ((= (length (car timeSignatureToShow)) 2)
56               (make-center-column-markup
57                 (list (make-number-markup
58                       (number->string
59                        (car (car timeSignatureToShow))))
60                       (make-number-markup
61                        (number->string
62                         (cadr (car timeSignatureToShow))))))))

```



```

63
64      ((= (length (car timeSignatureToShow)) 3)
65      (make-override-markup
66      (cons 'baseline-skip 0)
67      (make-center-column-markup
68      (list
69      (make-line-markup
70      (list
71      (make-number-markup
72      (make-right-align-markup
73      (make-translate-markup
74      (cons 0 1.6)
75      (make-fontsize-markup
76      -3
77      (number->string
78      (car (car timeSignatureToShow))))))))))
79
80      (make-hspace-markup -1.5)
81
82      (make-override-markup
83      (cons 'alignment 0)
84      (make-translate-markup
85      (cons 0 0.8)
86      (make-draw-line-markup (cons 1.5 1.35))))))
87
88      (make-hspace-markup -1.5)
89
90      (make-translate-markup
91      (cons 0 0)
92      (make-fontsize-markup
93      -3
94      (make-number-markup
95      (number->string
96      (cadr (car timeSignatureToShow))))))))))
97
98      (make-number-markup
99      (number->string
100      (caddr (car timeSignatureToShow)))))))))
101
102
103      ((= (length (car timeSignatureToShow)) 4)
104
105      (make-override-markup
106      (cons 'baseline-skip 0)
107      (make-center-column-markup
108      (list
109      (make-line-markup
110      (list

```

```

111         (make-number-markup
112           (number->string
113             (car (car timeSignatureToShow))))
114         (make-fontsize-markup
115           -12
116           (make-simple-markup " ")))
117
118         (make-hspace-markup -1.25)
119         (make-translate-markup
120           (cons 0 0.4)
121           (make-bold-markup
122             (make-simple-markup "+"))))
123
124         (make-hspace-markup -0.25)
125
126         (make-hspace-markup -0.5)
127         (make-right-align-markup
128           (make-number-markup
129             (make-translate-markup
130               (cons 0 1.5)
131               (make-fontsize-markup
132                 -3
133                 (number->string
134                   (cadr (car timeSignatureToShow)))))))
135
136         (make-hspace-markup -1.5)
137
138         (make-override-markup
139           (cons 'alignment 0)
140           (make-translate-markup
141             (cons 0 0.8)
142             (make-draw-line-markup
143               (cons 1.5 1.35))))))
144
145         (make-hspace-markup -1.5)
146
147         (make-number-markup
148           (make-left-align-markup
149             (make-fontsize-markup
150               -3
151               (number->string
152                 (caddr (car timeSignatureToShow)))))))
153
154         (make-number-markup
155           (number->string
156             (caddr (car timeSignatureToShow))))))
157
158

```

```

159      #:translate
160      (cons 0 -0.5)
161      (#:fontsize -12 " ")
162      #:translate
163      (cons 0 -0.5)
164      (#:bold "+")
165      #:translate
166      (cons 0 -0.5)
167      (#:fontsize -12 " ")
168
169      #:override
170      (cons 'baseline-skip 0)
171
172      (cond ((= (length (cadr timeSignatureToShow)) 2)
173            (make-center-column-markup
174              (list (make-number-markup
175                    (number->string
176                      (car (cadr timeSignatureToShow))))
177                    (make-number-markup
178                      (number->string
179                        (cadr (cadr timeSignatureToShow)))))))
180
181            ((= (length (cadr timeSignatureToShow)) 3)
182              (make-override-markup
183                (cons 'baseline-skip 0)
184                (make-center-column-markup
185                  (list
186                    (make-line-markup
187                      (list
188                        (make-number-markup
189                          (make-right-align-markup
190                            (make-translate-markup
191                              (cons 0 1.6)
192                              (make-fontsize-markup
193                                -3
194                                (number->string
195                                  (car (cadr timeSignatureToShow)))))))
196
197                      (make-hspace-markup -1.5)
198
199                      (make-override-markup
200                        (cons 'alignment 0)
201                        (make-translate-markup
202                          (cons 0 0.8)
203                          (make-draw-line-markup (cons 1.5 1.35))))
204
205                      (make-hspace-markup -1.5)
206

```

```

207         (make-translate-markup
208           (cons 0 0)
209           (make-fontsize-markup
210             -3
211             (make-number-markup
212               (number->string
213                 (cadr (cadr timeSignatureToShow)))))))))
214
215     (make-number-markup
216       (number->string
217         (caddr (cadr timeSignatureToShow))))))
218
219
220     ((= (length (cadr timeSignatureToShow)) 4)
221
222     (make-override-markup
223       (cons 'baseline-skip 0)
224       (make-center-column-markup
225         (list
226           (make-line-markup
227             (list
228               (make-number-markup
229                 (number->string
230                   (car (cadr timeSignatureToShow))))
231               (make-fontsize-markup
232                 -12
233                 (make-simple-markup " "))
234
235               (make-hspace-markup -1.25)
236               (make-translate-markup
237                 (cons 0 0.4)
238                 (make-bold-markup
239                   (make-simple-markup "+")))
240
241               (make-hspace-markup -0.25)
242
243               (make-hspace-markup -0.5)
244               (make-right-align-markup
245                 (make-number-markup
246                   (make-translate-markup
247                     (cons 0 1.5)
248                     (make-fontsize-markup
249                       -3
250                       (number->string
251                         (cadr (cadr timeSignatureToShow))))))
252
253               (make-hspace-markup -1.5)
254

```

```

255         (make-override-markup
256         (cons 'alignment 0)
257         (make-translate-markup
258         (cons 0 0.8)
259         (make-draw-line-markup
260         (cons 1.5 1.35))))))
261
262         (make-hspace-markup -1.5)
263
264         (make-number-markup
265         (make-left-align-markup
266         (make-fontsize-markup
267         -3
268         (number->string
269         (caddr (cadr timeSignatureToShow))))))))))
270
271         (make-number-markup
272         (number->string
273         (caddr (cadr timeSignatureToShow))))))))))
274     )))
275
276     #{
277     \time $underlyingMeter
278     \set beatStructure = $beatStructure
279
280     \override Timing.TimeSignature.stencil =
281     #ly:text-interface::print
282     \override Timing.TimeSignature.text =
283     #mkup
284     #})
285
286
287     backToNormalTimeSignature =
288     {
289     \unset beatStructure
290     \revert Timing.TimeSignature.stencil
291     \revert Timing.TimeSignature.text
292     \revert Staff.TimeSignature.stencil
293     \revert Staff.TimeSignature.text
294     }
295
296
297     {
298     \compoundFractionalTimeSignatureBTWO #'((3 4)(2 3 4)) 11/12 3,3,3,2
299     \tuplet 3/2 { c'8 c' c'} \tuplet 3/2 {c' c' c'}
300     \tuplet 3/2 {c' c' c'}
301     \incompleteTupletBracket \tuplet 3/2 {c' c'}
302     \backToNormalTimeSignature

```

```
303 \time 3/4
304 c'2.
305 }
306
```

11.5.4 Discussion

See [Discussion](#) of the entry *Compound Meter with Two Fractional Time Signatures, Style A*.

[Table of Contents](#)

11.6 Compound Meter with Two Fractional Time Signatures, Style C



11.6.1 Description

This is an implementation of a compound meter with two fractional time signatures with Style C.

11.6.2 Grammar

```
\compoundFractionalTimeSignatureCTwo
    #'((TIME_SIG1)(TIME_SIG2)) MEASURE_SPAN BEAT_STRUCT

\backToNormalTimeSignature
```

NB

1. Following the convention of `\compoundMeter` to enter the two time signatures, you will create a list of lists. Each `TIME_SIG` accepts: an ordinary time signature (list with two numbers), a time signature with a fraction (list with three numbers), or a time signature with an ordinary numerator and a fraction. See [Grammar of Fractional Time Signatures, Style A](#) for the explanation on the arguments for the order of arguments to specify time signatures.
2. `MEASURE_SPAN` and `BEAT_STRUCT` follow the same convention as before.
3. When you wish to go back to a regular time signature, use `\backToNormalTimeSignature`, otherwise the identical fractional time signature will keep showing up.

11.6.3 Code

```
1 \version "2.24.4"
2 \language "english"
3
4
5 % Revised Jan 2 2025 for improving the appearance of fractions
6 % Revised Aug 10 2025 to include the function to revert to a
7 % regular time signature
8
9 suppressWarning =
10 #(define-void-function (amount message)(number? string?)
11   (for-each
12     (lambda (warning)
13       (ly:expect-warning message))
14     (iota amount 1 1)))
```

```

15
16 \suppressWarning 1 "strange time signature found"
17
18 incompleteTupletBracket = {
19   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
20   \once \override Voice.TupletBracket.bracket-visibility = ##t
21
22 }
23 incompleteSmallTupletBracket = {
24   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
25   \once \override Voice.TupletBracket.bracket-visibility = ##t
26   \once \override Voice.TupletNumber.X-offset =
27     #(lambda (grob)
28       (if (= UP (ly:grob-property grob 'direction))
29           2.2
30           1.2))
31
32   \once \override Voice.TupletBracket.shorten-pair =
33     #(lambda (grob)
34       (if (= UP (ly:grob-property grob 'direction))
35           '(-0.7 . 0.15)
36           '(-0.3 . 0.8)))
37   \once \override Voice.TupletBracket.X-positions =
38     #(lambda (grob)
39       (if (= UP (ly:grob-property grob 'direction))
40           '(1.8 . 3)
41           '(0.3 . 2.7)))
42 }
43
44 compoundFractionalTimeSignatureCTwo =
45 #(define-music-function
46   (timeSignatureToShow underlyingMeter beatStructure)
47   (list? fraction? number-list?)
48   (define mkup
49     (markup
50       #:concat
51       (
52
53         #:override
54         (cons 'baseline-skip 0)
55         (cond ((= (length (car timeSignatureToShow)) 2)
56               (make-center-column-markup
57                 (list (make-number-markup
58                       (number->string
59                        (car (car timeSignatureToShow))))
60                       (make-number-markup
61                        (number->string
62                         (cadr (car timeSignatureToShow))))))))))

```



```

63
64      ((= (length (car timeSignatureToShow)) 3)
65      (make-override-markup
66      (cons 'baseline-skip 0)
67      (make-center-column-markup
68      (list
69      (make-line-markup
70      (list
71
72      (make-right-align-markup
73      (make-number-markup
74      (number->string
75      (car (car timeSignatureToShow))))))
76
77      (make-hspace-markup -0.6)
78
79      (make-override-markup
80      (list (cons 'alignment 0)
81            (cons 'thickness 2))
82      (make-draw-line-markup
83      (cons 0.5 2)))
84
85      (make-hspace-markup -0.6)
86
87      (make-number-markup
88      (make-left-align-markup
89      (number->string
90      (cadr (car timeSignatureToShow))))))
91
92      (make-number-markup
93      (number->string
94      (caddr (car timeSignatureToShow))))))
95
96      ((= (length (car timeSignatureToShow)) 4)
97
98      (make-override-markup
99      (cons 'baseline-skip 0)
100      (make-center-column-markup
101      (list
102      (make-line-markup
103      (list
104      (make-number-markup
105      (number->string
106      (car (car timeSignatureToShow))))
107      (make-fontsize-markup
108      -12
109      (make-simple-markup " ")))

```

```

111
112
113         (make-hspace-markup -1.25)
114         (make-translate-markup
115         (cons 0 0.4)
116         (make-bold-markup
117         (make-simple-markup "+")))
118
119         (make-hspace-markup -0.25)
120
121         (make-hspace-markup -0.5)
122         (make-right-align-markup
123         (make-number-markup
124         (number->string
125         (cadr (car timeSignatureToShow))))))
126
127         (make-hspace-markup -0.6)
128
129         (make-override-markup
130         (list (cons 'alignment 0)
131         (cons 'thickness 2))
132         (make-draw-line-markup (cons 0.5 2)))
133
134         (make-hspace-markup -0.6)
135
136         (make-number-markup
137         (make-left-align-markup
138         (number->string
139         (caddr (car timeSignatureToShow))))))
140
141         (make-number-markup
142         (number->string
143         (caddr (car timeSignatureToShow))))))
144
145
146 #:translate
147 (cons 0 -0.5)
148 (#:fontsize -12 " ")
149 #:translate
150 (cons 0 -0.5)
151 (#:bold "+")
152 #:translate
153 (cons 0 -0.5)
154 (#:fontsize -12 " ")
155
156 #:override
157 (cons 'baseline-skip 0)
158

```

```

159      (cond ((= (length (cadr timeSignatureToShow)) 2)
160             (make-center-column-markup
161               (list (make-number-markup
162                     (number->string
163                     (car (cadr timeSignatureToShow))))
164                     (make-number-markup
165                     (number->string
166                     (cadr (cadr timeSignatureToShow)))))))
167
168      ((= (length (cadr timeSignatureToShow)) 3)
169       (make-override-markup
170         (cons 'baseline-skip 0)
171         (make-center-column-markup
172           (list
173             (make-line-markup
174               (list
175
176                 (make-right-align-markup
177                   (make-number-markup
178                     (number->string
179                     (car (cadr timeSignatureToShow))))))
180
181               (make-hspace-markup -0.6)
182
183                 (make-override-markup
184                   (list (cons 'alignment 0)
185                         (cons 'thickness 2))
186                   (make-draw-line-markup
187                     (cons 0.5 2)))
188
189               (make-hspace-markup -0.6)
190
191                 (make-number-markup
192                   (make-left-align-markup
193                     (number->string
194                     (cadr (cadr timeSignatureToShow)))))))))
195
196         (make-number-markup
197           (number->string
198             (caddr (cadr timeSignatureToShow))))))
199
200      ((= (length (cadr timeSignatureToShow)) 4)
201       (make-override-markup
202         (cons 'baseline-skip 0)
203         (make-center-column-markup
204           (list

```

```

207         (make-line-markup
208         (list
209         (make-number-markup
210         (number->string
211         (car (cadr timeSignatureToShow))))))
212         (make-fontsize-markup
213         -12
214         (make-simple-markup " ")))
215
216
217         (make-hspace-markup -1.25)
218         (make-translate-markup
219         (cons 0 0.4)
220         (make-bold-markup
221         (make-simple-markup "+"))))
222
223         (make-hspace-markup -0.25)
224
225         (make-hspace-markup -0.5)
226         (make-right-align-markup
227         (make-number-markup
228         (number->string
229         (cadr (cadr timeSignatureToShow))))))
230
231         (make-hspace-markup -0.6)
232
233         (make-override-markup
234         (list (cons 'alignment 0)
235         (cons 'thickness 2))
236         (make-draw-line-markup (cons 0.5 2)))
237
238         (make-hspace-markup -0.6)
239
240         (make-number-markup
241         (make-left-align-markup
242         (number->string
243         (caddr (cadr timeSignatureToShow))))))
244
245         (make-number-markup
246         (number->string
247         (caddr (cadr timeSignatureToShow))))))
248     )))
249
250     #{
251     \time $underlyingMeter
252     \set beatStructure = $beatStructure
253     \override Timing.TimeSignature.stencil =
254     #ly:text-interface::print

```

```

255     \override Timing.TimeSignature.text =
256     #mkup
257     #})
258
259
260 backToNormalTimeSignature =
261 {
262   \unset beatStructure
263   \revert Timing.TimeSignature.stencil
264   \revert Timing.TimeSignature.text
265   \revert Staff.TimeSignature.stencil
266   \revert Staff.TimeSignature.text
267 }
268
269
270 {
271
272   \compoundFractionalTimeSignatureCTwo #'((3 4)(2 3 4)) 11/12 3,3,3,2
273   \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
274   \tuplet 3/2 {c' c' c'}
275   \incompleteTupletBracket \tuplet 3/2 {c' c'}
276   \backToNormalTimeSignature
277   \time 3/4
278   c'2.
279 }

```

11.6.4 Discussion

See [Discussion](#) of the entry *Compound Meter with Two Fractional Time Signatures, Style A*.

[Table of Contents](#)

11.7 Compound Meter with Three Fractional Time Signatures, Style A



11.7.1 Description

This is an implementation of a compound meter with three fractional time signatures with Style A.

11.7.2 Grammar

```
\compoundFractionalTimeSignatureAThree
    #'((TIME_SIG1)(TIME_SIG2)(TIME_SIG3)) MEASURE_SPAN BEAT_STRUCT

\backToNormalTimeSignature
```

NB

1. Following the convention of `\compoundMeter` to enter the two time signatures, you will create a list of lists. Each `TIME_SIG` accepts:
 - an ordinary time signature (list with two numbers);
 - a time signature with a fraction (list with three numbers), or;
 - a time signature with an ordinary numerator and a fraction.

See [Grammar of Fractional Time Signatures, Style A](#) for the explanation on the arguments for the order of arguments to specify time signatures.

2. In the code of the given snippet, the value for `MEASURE_SPAN` may appear absurd. However, this results from following the same convention as before, i.e. adding the constituent time signatures to give a general irrational time signature for the entire bar. Thus:

$$\frac{3}{4} + \frac{4}{20} + \frac{2}{12} = \frac{67}{60}$$

3. `BEAT_STRUCT` follows the same convention as before; however, as the given code shows, it may be necessary to still use `[` and `]` to explicitly specify the beaming.
4. When you wish to go back to a regular time signature, use `\backToNormalTimeSignature`, otherwise the identical fractional time signature will keep showing up.

11.7.3 Code

```
1 \version "2.24.4"
2 \language "english"
```

```

3
4 % Revised Aug 10 2025 to include the function to revert to a
5 % regular time signature
6
7 suppressWarning =
8 #(define-void-function (amount message)(number? string?)
9   (for-each
10     (lambda (warning)
11       (ly:expect-warning message))
12     (iota amount 1 1)))
13
14 \suppressWarning 3 "strange time signature found"
15
16 incompleteTupletBracket = {
17   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
18   \once \override Voice.TupletBracket.bracket-visibility = ##t
19
20 }
21 incompleteSmallTupletBracket = {
22   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
23   \once \override Voice.TupletBracket.bracket-visibility = ##t
24   \once \override Voice.TupletNumber.X-offset =
25     #(lambda (grob)
26       (if (= UP (ly:grob-property grob 'direction))
27         2.2
28         1.2))
29
30   \once \override Voice.TupletBracket.shorten-pair =
31     #(lambda (grob)
32       (if (= UP (ly:grob-property grob 'direction))
33         '(-0.7 . 0.15)
34         '(-0.3 . 0.8)))
35   \once \override Voice.TupletBracket.X-positions =
36     #(lambda (grob)
37       (if (= UP (ly:grob-property grob 'direction))
38         '(1.8 . 3)
39         '(0.3 . 2.7)))
40 }
41
42 compoundFractionalTimeSignatureAThree =
43 #(define-music-function
44   (timeSignatureToShow underlyingMeter beatStructure)
45   (list? fraction? number-list?)
46   (define mkup
47     (markup
48       #:concat
49       (
50         #:override

```



```

99             (make-halign-markup
100             -1.5
101             (make-center-column-markup
102             (list
103             (make-translate-markup
104             (cons 0 1)
105             (make-fontsize-markup
106             -6
107             (make-number-markup
108             (number->string
109             (cadr (car timeSignatureToShow))))))
110             (make-translate-markup
111             (cons 0 0)
112             (make-fontsize-markup
113             -6
114             (make-number-markup
115             (number->string
116             (caddr (car timeSignatureToShow))))))))))
117             (make-number-markup
118             (number->string
119             (caddr (car timeSignatureToShow))))))
120         ))
121     )
122
123
124     #:translate
125     (cons 0 -0.5)
126     (#:fontsize -12 " ")
127     #:translate
128     (cons 0 -0.5)
129     (#:bold "+")
130     #:translate
131     (cons 0 -0.5)
132     (#:fontsize -12 " ")
133
134     #:override
135     (cons 'baseline-skip 0)
136     (cond ((= (length (cadr timeSignatureToShow)) 2)
137           (make-center-column-markup
138           (list (make-number-markup
139                 (number->string
140                 (car (cadr timeSignatureToShow))))
141                 (make-number-markup
142                 (number->string
143                 (cadr (cadr timeSignatureToShow)))))))
144           ((= (length (cadr timeSignatureToShow)) 3)

```

```

147      (make-override-markup
148      (cons 'baseline-skip 0)
149      (make-center-column-markup
150      (list
151
152
153          (make-center-column-markup
154          (list
155          (make-translate-markup
156          (cons 0 1)
157          (make-fontsize-markup
158          -6
159          (make-number-markup
160          (number->string
161          (car (cadr timeSignatureToShow))))))
162          (make-translate-markup
163          (cons 0 0)
164          (make-fontsize-markup
165          -6
166          (make-number-markup
167          (number->string
168          (cadr (cadr timeSignatureToShow))))))
169          (make-number-markup
170          (number->string
171          (caddr (cadr timeSignatureToShow))))))
172      ))
173
174
175      ((= (length (cadr timeSignatureToShow)) 4)
176
177      (make-override-markup
178      (cons 'baseline-skip 0)
179      (make-center-column-markup
180      (list
181
182          (make-concat-markup
183          (list (make-number-markup
184          (number->string
185          (car (cadr timeSignatureToShow))))
186          (make-halign-markup
187          -1.5
188          (make-center-column-markup
189          (list
190          (make-translate-markup
191          (cons 0 1)
192          (make-fontsize-markup
193          -6
194          (make-number-markup

```

```

195             (number->string
196               (cadr (cadr timeSignatureToShow))))))
197       (make-translate-markup
198         (cons 0 0)
199         (make-fontsize-markup
200           -6
201           (make-number-markup
202             (number->string
203               (caddr (cadr timeSignatureToShow))))))))))
204     (make-number-markup
205       (number->string
206         (caddr (cadr timeSignatureToShow))))))
207   ))
208 )
209 #:translate
210 (cons 0 -0.5)
211 (#:fontsize -12 " ")
212 #:translate
213 (cons 0 -0.5)
214 (#:bold "+")
215 #:translate
216 (cons 0 -0.5)
217 (#:fontsize -12 " ")
218
219 #:override
220 (cons 'baseline-skip 0)
221 (cond ((= (length (caddr timeSignatureToShow)) 2)
222        (make-center-column-markup
223          (list (make-number-markup
224                  (number->string
225                    (car (caddr timeSignatureToShow))))
226                (make-number-markup
227                  (number->string
228                    (cadr (caddr timeSignatureToShow)))))))
229        ((= (length (caddr timeSignatureToShow)) 3)
230          (make-override-markup
231            (cons 'baseline-skip 0)
232            (make-center-column-markup
233              (list
234                (make-center-column-markup
235                  (list
236                    (make-translate-markup
237                      (cons 0 1)
238                      (make-fontsize-markup

```

```

243         -6
244         (make-number-markup
245         (number->string
246         (car (caddr timeSignatureToShow))))))
247     (make-translate-markup
248     (cons 0 0)
249     (make-fontsize-markup
250     -6
251     (make-number-markup
252     (number->string
253     (cadr (caddr timeSignatureToShow)))))))))
254 (make-number-markup
255 (number->string
256 (caddr (caddr timeSignatureToShow))))))
257 ))
258
259
260 ((= (length (caddr timeSignatureToShow)) 4)
261
262 (make-override-markup
263 (cons 'baseline-skip 0)
264 (make-center-column-markup
265 (list
266
267 (make-concat-markup
268 (list (make-number-markup
269 (number->string
270 (car (caddr timeSignatureToShow))))))
271 (make-halign-markup
272 -1.5
273 (make-center-column-markup
274 (list
275 (make-translate-markup
276 (cons 0 1)
277 (make-fontsize-markup
278 -6
279 (make-number-markup
280 (number->string
281 (cadr (caddr timeSignatureToShow))))))
282 (make-translate-markup
283 (cons 0 0)
284 (make-fontsize-markup
285 -6
286 (make-number-markup
287 (number->string
288 (caddr (caddr timeSignatureToShow)))))))))
289 (make-number-markup
290 (number->string

```

```

291             (caddr (caddr timeSignatureToShow))))))
292         ))
293     )
294
295
296     )))
297
298     #{
299     \time $underlyingMeter
300     \set beatStructure = $beatStructure
301
302     \override Timing.TimeSignature.stencil =
303     #ly:text-interface::print
304     \override Timing.TimeSignature.text =
305     #mkup
306     #})
307
308     backToNormalTimeSignature =
309     {
310     \unset beatStructure
311     \revert Timing.TimeSignature.stencil
312     \revert Timing.TimeSignature.text
313     \revert Staff.TimeSignature.stencil
314     \revert Staff.TimeSignature.text
315     }
316
317
318     {
319
320     \compoundFractionalTimeSignatureAThree #'((3 4)(4 5 4)(2 3 4)) 67/60 3,3,3,4,2
321     \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
322     \tuplet 3/2 {c'[ c' c']}
323     \incompleteTupletBracket \tuplet 5/4 {c'16[ c' c' c']}
324     \incompleteTupletBracket \tuplet 3/2 {c'8 c'}
325
326     \backToNormalTimeSignature
327     \time 3/4
328     c'2.
329     }

```

11.7.4 Discussion

1. As mentioned in the **Grammar** section, it appears that specifying the beaming in the LilyPond code is still necessary. This is probably because of the unusual value of the fraction that needs to be given in the second argument of the function, `MEASURE_SPAN`.
2. As in the case of the other compound meters introduced in this document, I am hoping to find ways to simplify the code.

3. When you wish to go back to a regular time signature, use `\backToNormalTimeSignature`, otherwise the identical fractional time signature will keep showing up.

[Table of Contents](#)

11.8 Compound Meter with Three Fractional Time Signatures, Style B



11.8.1 Description

This is an implementation of a compound meter with three fractional time signatures with Style B.

11.8.2 Grammar

```
\compoundFractionalTimeSignatureBThree
    #'((TIME_SIG1)(TIME_SIG2)(TIME_SIG3)) MEASURE_SPAN BEAT_STRUCT

\backToNormalTimeSignature
```

NB

1. See [Grammar](#) of the entry *Compound Meter with Three Fractional Time Signatures, Style A*.
2. When you wish to go back to a regular time signature, use `\backToNormalTimeSignature`, otherwise the identical fractional time signature will keep showing up.

11.8.3 Code

```
1 \version "2.24.4"
2 \language "english"
3
4 % Revised Jan 2 2025 for improving the appearance of fractions
5 % Revised Aug 10 2025 to include the function to revert to a
6 % regular time signature
7
8 suppressWarning =
9 #(define-void-function (amount message)(number? string?)
10   (for-each
11     (lambda (warning)
12       (ly:expect-warning message))
13     (iota amount 1 1)))
14
15 \suppressWarning 1 "strange time signature found"
16
17 incompleteTupletBracket = {
18   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
```

```

19 \once \override Voice.TupletBracket.bracket-visibility = ##t
20
21 }
22 incompleteSmallTupletBracket = {
23 \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
24 \once \override Voice.TupletBracket.bracket-visibility = ##t
25 \once \override Voice.TupletNumber.X-offset =
26 #(lambda (grob)
27   (if (= UP (ly:grob-property grob 'direction))
28       2.2
29       1.2))
30
31 \once \override Voice.TupletBracket.shorten-pair =
32 #(lambda (grob)
33   (if (= UP (ly:grob-property grob 'direction))
34       '(-0.7 . 0.15)
35       '(-0.3 . 0.8)))
36 \once \override Voice.TupletBracket.X-positions =
37 #(lambda (grob)
38   (if (= UP (ly:grob-property grob 'direction))
39       '(1.8 . 3)
40       '(0.3 . 2.7)))
41 }
42
43 compoundFractionalTimeSignatureBThree =
44 #(define-music-function
45   (timeSignatureToShow underlyingMeter beatStructure)
46   (list? fraction? number-list?)
47   (define mkup
48     (markup
49       #:concat
50
51       (
52
53         #:override
54         (cons 'baseline-skip 0)
55         (cond ((= (length (car timeSignatureToShow)) 2)
56               (make-center-column-markup
57                 (list (make-number-markup
58                       (number->string
59                        (car (car timeSignatureToShow))))
60                       (make-number-markup
61                        (number->string
62                         (cadr (car timeSignatureToShow)))))))
63
64               ((= (length (car timeSignatureToShow)) 3)
65                 (make-override-markup
66                  (cons 'baseline-skip 0)

```



```

67      (make-center-column-markup
68      (list
69      (make-line-markup
70      (list
71      (make-number-markup
72      (make-right-align-markup
73      (make-translate-markup
74      (cons 0 1.6)
75      (make-fontsize-markup
76      -3
77      (number->string
78      (car (car timeSignatureToShow)))))))
79
80      (make-hspace-markup -1.5)
81
82      (make-override-markup
83      (cons 'alignment 0)
84      (make-translate-markup
85      (cons 0 0.8)
86      (make-draw-line-markup (cons 1.5 1.35))))
87
88      (make-hspace-markup -1.5)
89
90      (make-translate-markup
91      (cons 0 0)
92      (make-fontsize-markup
93      -3
94      (make-number-markup
95      (number->string
96      (cadr (car timeSignatureToShow)))))))
97
98      (make-number-markup
99      (number->string
100      (caddr (car timeSignatureToShow))))))
101
102      ((= (length (car timeSignatureToShow)) 4)
103
104      (make-override-markup
105      (cons 'baseline-skip 0)
106      (make-center-column-markup
107      (list
108      (make-line-markup
109      (list
110      (make-number-markup
111      (number->string
112      (car (car timeSignatureToShow))))
113      (make-fontsize-markup

```

```

115         -12
116         (make-simple-markup " ")
117
118         (make-hspace-markup -1.25)
119         (make-translate-markup
120         (cons 0 0.4)
121         (make-bold-markup
122         (make-simple-markup "+"))))
123
124         (make-hspace-markup -0.25)
125
126         (make-hspace-markup -0.5)
127         (make-right-align-markup
128         (make-number-markup
129         (make-translate-markup
130         (cons 0 1.5)
131         (make-fontsize-markup
132         -3
133         (number->string
134         (cadr (car timeSignatureToShow)))))))
135
136         (make-hspace-markup -1.5)
137
138         (make-override-markup
139         (cons 'alignment 0)
140         (make-translate-markup
141         (cons 0 0.8)
142         (make-draw-line-markup
143         (cons 1.5 1.35))))))
144
145         (make-hspace-markup -1.5)
146
147         (make-number-markup
148         (make-left-align-markup
149         (make-fontsize-markup
150         -3
151         (number->string
152         (caddr (car timeSignatureToShow)))))))
153
154         (make-number-markup
155         (number->string
156         (caddr (car timeSignatureToShow))))))
157
158
159 #:translate
160 (cons 0 -0.5)
161 (:fontsize -12 " ")
162 #:translate

```

```

163      (cons 0 -0.5)
164      (:bold "+")
165      #:translate
166      (cons 0 -0.5)
167      (:fontsize -12 " ")
168
169      #:override
170      (cons 'baseline-skip 0)
171
172      (cond ((= (length (cadr timeSignatureToShow)) 2)
173            (make-center-column-markup
174              (list (make-number-markup
175                    (number->string
176                      (car (cadr timeSignatureToShow))))
177                    (make-number-markup
178                      (number->string
179                        (cadr (cadr timeSignatureToShow)))))))
180
181            ((= (length (cadr timeSignatureToShow)) 3)
182              (make-override-markup
183                (cons 'baseline-skip 0)
184                (make-center-column-markup
185                  (list
186                    (make-line-markup
187                      (list
188                        (make-number-markup
189                          (make-right-align-markup
190                            (make-translate-markup
191                              (cons 0 1.6)
192                              (make-fontsize-markup
193                                -3
194                                (number->string
195                                  (car (cadr timeSignatureToShow)))))))
196
197                        (make-hspace-markup -1.5)
198
199                        (make-override-markup
200                          (cons 'alignment 0)
201                          (make-translate-markup
202                            (cons 0 0.8)
203                            (make-draw-line-markup (cons 1.5 1.35))))
204
205                        (make-hspace-markup -1.5)
206
207                        (make-translate-markup
208                          (cons 0 0)
209                          (make-fontsize-markup
210                            -3

```

```

211         (make-number-markup
212           (number->string
213             (cadr (cadr timeSignatureToShow)))))))))
214
215     (make-number-markup
216       (number->string
217         (caddr (cadr timeSignatureToShow)))))))))
218
219
220     ((= (length (cadr timeSignatureToShow)) 4)
221
222       (make-override-markup
223         (cons 'baseline-skip 0)
224         (make-center-column-markup
225           (list
226             (make-line-markup
227               (list
228                 (make-number-markup
229                   (number->string
230                     (car (cadr timeSignatureToShow))))
231                 (make-fontsize-markup
232                   -12
233                   (make-simple-markup " "))
234
235                 (make-hspace-markup -1.25)
236                 (make-translate-markup
237                   (cons 0 0.4)
238                   (make-bold-markup
239                     (make-simple-markup "+")))
240
241                 (make-hspace-markup -0.25)
242
243                 (make-hspace-markup -0.5)
244                 (make-right-align-markup
245                   (make-number-markup
246                     (make-translate-markup
247                       (cons 0 1.5)
248                       (make-fontsize-markup
249                         -3
250                         (number->string
251                           (cadr (cadr timeSignatureToShow)))))))))
252
253                   (make-hspace-markup -1.5)
254
255                   (make-override-markup
256                     (cons 'alignment 0)
257                     (make-translate-markup
258                       (cons 0 0.8)

```

```

259             (make-draw-line-markup
260               (cons 1.5 1.35))))
261
262             (make-hspace-markup -1.5)
263
264             (make-number-markup
265               (make-left-align-markup
266                 (make-fontsize-markup
267                   -3
268                   (number->string
269                     (caddr (cadr timeSignatureToShow))))))))
270
271             (make-number-markup
272               (number->string
273                 (caddr (cadr timeSignatureToShow))))))
274
275     #:translate
276     (cons 0 -0.5)
277     (#:fontsize -12 " ")
278     #:translate
279     (cons 0 -0.5)
280     (#:bold "+")
281     #:translate
282     (cons 0 -0.5)
283     (#:fontsize -12 " ")
284
285     #:override
286     (cons 'baseline-skip 0)
287
288     (cond ((= (length (caddr timeSignatureToShow)) 2)
289            (make-center-column-markup
290              (list (make-number-markup
291                     (number->string
292                       (car (caddr timeSignatureToShow))))
293                    (make-number-markup
294                      (number->string
295                        (cadr (caddr timeSignatureToShow)))))))
296
297            ((= (length (caddr timeSignatureToShow)) 3)
298             (make-override-markup
299               (cons 'baseline-skip 0)
300               (make-center-column-markup
301                 (list
302                   (make-line-markup
303                     (list
304                       (make-number-markup
305                         (make-right-align-markup
306                           (make-translate-markup

```

```

307         (cons 0 1.6)
308         (make-fontsize-markup
309         -3
310         (number->string
311         (car (caddr timeSignatureToShow))))))
312
313     (make-hspace-markup -1.5)
314
315     (make-override-markup
316     (cons 'alignment 0)
317     (make-translate-markup
318     (cons 0 0.8)
319     (make-draw-line-markup (cons 1.5 1.35))))
320
321     (make-hspace-markup -1.5)
322
323     (make-translate-markup
324     (cons 0 0)
325     (make-fontsize-markup
326     -3
327     (make-number-markup
328     (number->string
329     (cadr (caddr timeSignatureToShow))))))
330
331     (make-number-markup
332     (number->string
333     (caddr (caddr timeSignatureToShow))))))
334
335
336     ((= (length (caddr timeSignatureToShow)) 4)
337
338     (make-override-markup
339     (cons 'baseline-skip 0)
340     (make-center-column-markup
341     (list
342     (make-line-markup
343     (list
344     (make-number-markup
345     (number->string
346     (car (caddr timeSignatureToShow))))
347     (make-fontsize-markup
348     -12
349     (make-simple-markup " ")))
350
351     (make-hspace-markup -1.25)
352     (make-translate-markup
353     (cons 0 0.4)
354     (make-bold-markup

```

```

355             (make-simple-markup "+"))
356
357         (make-hspace-markup -0.25)
358
359         (make-hspace-markup -0.5)
360         (make-right-align-markup
361         (make-number-markup
362         (make-translate-markup
363         (cons 0 1.5)
364         (make-fontsize-markup
365         -3
366         (number->string
367         (cadr (caddr timeSignatureToShow))))))))
368
369         (make-hspace-markup -1.5)
370
371         (make-override-markup
372         (cons 'alignment 0)
373         (make-translate-markup
374         (cons 0 0.8)
375         (make-draw-line-markup
376         (cons 1.5 1.35))))
377
378         (make-hspace-markup -1.5)
379
380         (make-number-markup
381         (make-left-align-markup
382         (make-fontsize-markup
383         -3
384         (number->string
385         (caddr (caddr timeSignatureToShow))))))))
386
387         (make-number-markup
388         (number->string
389         (caddr (caddr timeSignatureToShow))))))
390     )
391
392 ))
393
394 #{
395   \time $underlyingMeter
396   \set beatStructure = $beatStructure
397
398   \override Timing.TimeSignature.stencil =
399   #ly:text-interface::print
400   \override Timing.TimeSignature.text =
401   #mkup
402   #})

```

```

403
404 backToNormalTimeSignature =
405 {
406   \unset beatStructure
407   \revert Timing.TimeSignature.stencil
408   \revert Timing.TimeSignature.text
409   \revert Staff.TimeSignature.stencil
410   \revert Staff.TimeSignature.text
411 }
412
413 {
414
415   \compoundFractionalTimeSignatureBThree #'((3 4)(4 5 4)(2 3 4)) 67/60 3,3,3,4,2
416   \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
417   \tuplet 3/2 {c'[ c' c']}
418   \incompleteTupletBracket \tuplet 5/4 {c'16[ c' c' c']}
419   \incompleteTupletBracket \tuplet 3/2 {c'8 c'}
420   \backToNormalTimeSignature
421   \time 3/4
422   c'2.
423
424 }

```

11.8.4 Discussion

See [Discussion](#) of the entry *Compound Meter with Three Fractional Time Signatures, Style A*.

[Table of Contents](#)


```

18 incompleteTupletBracket = {
19   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
20   \once \override Voice.TupletBracket.bracket-visibility = ##t
21
22 }
23 incompleteSmallTupletBracket = {
24   \once \override Voice.TupletBracket.edge-height = #'(0.7 . 0)
25   \once \override Voice.TupletBracket.bracket-visibility = ##t
26   \once \override Voice.TupletNumber.X-offset =
27     #(lambda (grob)
28       (if (= UP (ly:grob-property grob 'direction))
29           2.2
30           1.2))
31
32   \once \override Voice.TupletBracket.shorten-pair =
33     #(lambda (grob)
34       (if (= UP (ly:grob-property grob 'direction))
35           '(-0.7 . 0.15)
36           '(-0.3 . 0.8)))
37   \once \override Voice.TupletBracket.X-positions =
38     #(lambda (grob)
39       (if (= UP (ly:grob-property grob 'direction))
40           '(1.8 . 3)
41           '(0.3 . 2.7)))
42 }
43
44 compoundFractionalTimeSignatureCThree =
45 # (define-music-function
46   (timeSignatureToShow underlyingMeter beatStructure)
47   (list? fraction? number-list?)
48   (define mkup
49     (markup
50       #:concat
51       (
52
53         #:override
54         (cons 'baseline-skip 0)
55         (cond ((= (length (car timeSignatureToShow)) 2)
56               (make-center-column-markup
57                 (list (make-number-markup
58                       (number->string
59                        (car (car timeSignatureToShow))))
60                       (make-number-markup
61                        (number->string
62                         (cadr (car timeSignatureToShow)))))))
63               ((= (length (car timeSignatureToShow)) 3)
64                 (make-override-markup

```

```

66      (cons 'baseline-skip 0)
67      (make-center-column-markup
68        (list
69          (make-line-markup
70            (list
71
72              (make-right-align-markup
73                (make-number-markup
74                  (number->string
75                    (car (car timeSignatureToShow))))))
76
77              (make-hspace-markup -0.6)
78
79              (make-override-markup
80                (list (cons 'alignment 0)
81                      (cons 'thickness 2))
82                (make-draw-line-markup
83                  (cons 0.5 2)))
84
85              (make-hspace-markup -0.6)
86
87              (make-number-markup
88                (make-left-align-markup
89                  (number->string
90                    (cadr (car timeSignatureToShow))))))
91
92              (make-number-markup
93                (number->string
94                  (caddr (car timeSignatureToShow))))))
95
96      ((= (length (car timeSignatureToShow)) 4)
97
98      (make-override-markup
99        (cons 'baseline-skip 0)
100        (make-center-column-markup
101          (list
102            (make-line-markup
103              (list
104                (make-number-markup
105                  (number->string
106                    (car (car timeSignatureToShow))))
107                (make-fontsize-markup
108                  -12
109                  (make-simple-markup " "))
110
111                (make-hspace-markup -1.25)

```

```

114         (make-translate-markup
115         (cons 0 0.4)
116         (make-bold-markup
117         (make-simple-markup "+")))
118
119         (make-hspace-markup -0.25)
120
121         (make-hspace-markup -0.5)
122         (make-right-align-markup
123         (make-number-markup
124         (number->string
125         (cadr (car timeSignatureToShow))))))
126
127         (make-hspace-markup -0.6)
128
129         (make-override-markup
130         (list (cons 'alignment 0)
131               (cons 'thickness 2))
132         (make-draw-line-markup (cons 0.5 2)))
133
134         (make-hspace-markup -0.6)
135
136         (make-number-markup
137         (make-left-align-markup
138         (number->string
139         (caddr (car timeSignatureToShow))))))
140
141         (make-number-markup
142         (number->string
143         (caddr (car timeSignatureToShow))))))
144
145 #:translate
146 (cons 0 -0.5)
147 (#:fontsize -12 " ")
148 #:translate
149 (cons 0 -0.5)
150 (#:bold "+")
151 #:translate
152 (cons 0 -0.5)
153 (#:fontsize -12 " ")
154
155 #:override
156 (cons 'baseline-skip 0)
157
158
159 (cond ((= (length (cadr timeSignatureToShow)) 2)
160        (make-center-column-markup
161        (list (make-number-markup

```

```

162         (number->string
163         (car (cadr timeSignatureToShow))))
164     (make-number-markup
165     (number->string
166     (cadr (cadr timeSignatureToShow))))))
167
168 ((= (length (cadr timeSignatureToShow)) 3)
169 (make-override-markup
170 (cons 'baseline-skip 0)
171 (make-center-column-markup
172 (list
173 (make-line-markup
174 (list
175
176 (make-right-align-markup
177 (make-number-markup
178 (number->string
179 (car (cadr timeSignatureToShow))))))
180
181 (make-hspace-markup -0.6)
182
183 (make-override-markup
184 (list (cons 'alignment 0)
185 (cons 'thickness 2))
186 (make-draw-line-markup
187 (cons 0.5 2)))
188
189 (make-hspace-markup -0.6)
190
191 (make-number-markup
192 (make-left-align-markup
193 (number->string
194 (cadr (cadr timeSignatureToShow))))))
195
196 (make-number-markup
197 (number->string
198 (caddr (cadr timeSignatureToShow))))))
199
200
201 ((= (length (cadr timeSignatureToShow)) 4)
202
203 (make-override-markup
204 (cons 'baseline-skip 0)
205 (make-center-column-markup
206 (list
207 (make-line-markup
208 (list
209 (make-number-markup

```

```

210         (number->string
211         (car (cadr timeSignatureToShow))))
212 (make-fontsize-markup
213 -12
214 (make-simple-markup " "))
215
216
217 (make-hspace-markup -1.25)
218 (make-translate-markup
219 (cons 0 0.4)
220 (make-bold-markup
221 (make-simple-markup "+")))
222
223 (make-hspace-markup -0.25)
224
225 (make-hspace-markup -0.5)
226 (make-right-align-markup
227 (make-number-markup
228 (number->string
229 (cadr (cadr timeSignatureToShow)))))
230
231 (make-hspace-markup -0.6)
232
233 (make-override-markup
234 (list (cons 'alignment 0)
235 (cons 'thickness 2))
236 (make-draw-line-markup (cons 0.5 2)))
237
238 (make-hspace-markup -0.6)
239
240 (make-number-markup
241 (make-left-align-markup
242 (number->string
243 (caddr (cadr timeSignatureToShow))))))
244
245 (make-number-markup
246 (number->string
247 (caddr (cadr timeSignatureToShow))))))
248
249 #:translate
250 (cons 0 -0.5)
251 (#:fontsize -12 " ")
252 #:translate
253 (cons 0 -0.5)
254 (#:bold "+")
255 #:translate
256 (cons 0 -0.5)
257 (#:fontsize -12 " ")

```

```

258
259     #:override
260     (cons 'baseline-skip 0)
261
262     (cond ((= (length (caddr timeSignatureToShow)) 2)
263           (make-center-column-markup
264             (list (make-number-markup
265                   (number->string
266                     (car (caddr timeSignatureToShow))))
267                   (make-number-markup
268                     (number->string
269                       (cadr (caddr timeSignatureToShow)))))))
270
271           ((= (length (caddr timeSignatureToShow)) 3)
272             (make-override-markup
273               (cons 'baseline-skip 0)
274               (make-center-column-markup
275                 (list
276                   (make-line-markup
277                     (list
278                       (make-right-align-markup
279                         (make-number-markup
280                           (number->string
281                             (car (caddr timeSignatureToShow))))))
282                     (make-hspace-markup -0.6)
283                     (make-override-markup
284                       (list (cons 'alignment 0)
285                             (cons 'thickness 2))
286                       (make-draw-line-markup
287                         (cons 0.5 2)))
288                     (make-hspace-markup -0.6)
289                     (make-number-markup
290                       (make-left-align-markup
291                         (number->string
292                           (cadr (caddr timeSignatureToShow))))))
293                     (make-number-markup
294                       (number->string
295                         (caddr (caddr timeSignatureToShow))))))
296               (make-number-markup
297                 (number->string
298                   (caddr (caddr timeSignatureToShow))))))
298
299           ((= (length (caddr timeSignatureToShow)) 4)
300             (make-override-markup
301               (cons 'baseline-skip 0)
302               (make-center-column-markup
303                 (list
304                   (make-line-markup
305                     (list

```

```

306      (make-override-markup
307      (cons 'baseline-skip 0)
308      (make-center-column-markup
309      (list
310      (make-line-markup
311      (list
312      (make-number-markup
313      (number->string
314      (car (caddr timeSignatureToShow))))
315      (make-fontsize-markup
316      -12
317      (make-simple-markup " ")))
318
319
320      (make-hspace-markup -1.25)
321      (make-translate-markup
322      (cons 0 0.4)
323      (make-bold-markup
324      (make-simple-markup "+"))))
325
326      (make-hspace-markup -0.25)
327
328      (make-hspace-markup -0.5)
329      (make-right-align-markup
330      (make-number-markup
331      (number->string
332      (cadr (caddr timeSignatureToShow))))))
333
334      (make-hspace-markup -0.6)
335
336      (make-override-markup
337      (list (cons 'alignment 0)
338            (cons 'thickness 2))
339      (make-draw-line-markup (cons 0.5 2)))
340
341      (make-hspace-markup -0.6)
342
343      (make-number-markup
344      (make-left-align-markup
345      (number->string
346      (caddr (caddr timeSignatureToShow))))))
347
348      (make-number-markup
349      (number->string
350      (caddr (caddr timeSignatureToShow))))))
351    )
352  ))
353

```



```

354   #{
355     \time $underlyingMeter
356     \set beatStructure = $beatStructure
357
358     \override Timing.TimeSignature.stencil =
359     #ly:text-interface::print
360     \override Timing.TimeSignature.text =
361     #mkup
362     #})
363
364   backToNormalTimeSignature =
365   {
366     \unset beatStructure
367     \revert Timing.TimeSignature.stencil
368     \revert Timing.TimeSignature.text
369     \revert Staff.TimeSignature.stencil
370     \revert Staff.TimeSignature.text
371   }
372
373
374   {
375
376     \compoundFractionalTimeSignatureCThree #'((3 4)(4 5 4)(2 3 4)) 67/60 3,3,3,4,2
377     \tuplet 3/2 {c'8 c' c'} \tuplet 3/2 {c' c' c'}
378     \tuplet 3/2 {c'[ c' c']}
379     \incompleteTupletBracket \tuplet 5/4 {c'16[ c' c' c']}
380     \incompleteTupletBracket \tuplet 3/2 {c'8 c'}
381     \backToNormalTimeSignature
382     \time 3/4
383     c'2.
384   }
385

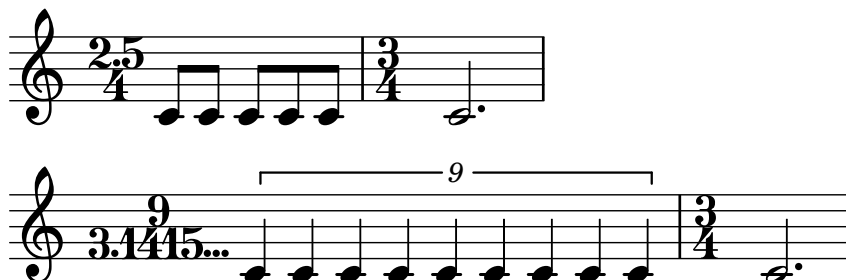
```

11.9.4 Discussion

See [Discussion](#) of the entry *Compound Meter with Three Fractional Time Signatures, Style A*.

[Table of Contents](#)

11.10 Time Signature with Decimals



11.10.1 Description

This is an implementation of a time signature with decimals. This function allows the user to use decimals for both numerator and denominator values of the time signature. While there are many examples where the numerator value contains decimals, notable examples for denominators that contain decimals include works by Mark Andre, such as *Un-fini I* (1995) for harp,¹⁶ *Un-fini II* (1994-95) for harpsichord,¹⁷ and *Contrapunctus* (1998/99) for piano..¹⁸ This function also has the option of showing ellipsis, as discussed in **Grammar**.

11.10.2 Grammar

```
\decimalPointTimeSignature
    #'((NUMERATOR)(DENOMINATOR)) MEASURE_SPAN BEAT_STRUCT
```

```
\backToNormalTimeSignature
```

NB

1. The first argument takes a list of two lists. For both `NUMERATOR` and `DENOMINATOR`, one or two numbers can be placed.
 - (a) If only one number is placed, it is treated as an integer. For example, `#'((3)(4))` would print: $\frac{3}{4}$.
 - (b) If two numbers are placed, the first number is the integer portion of the number, and the second number is the decimals. For example, `#'((3 5)(4 232))` would print: $\frac{3.5}{4.232}$.
 - (c) Placing a dot `.` at the end of the second number will print the ellipsis `...` at the end. This is useful for notating infinite decimal representations. For example, `#'((3 14159.)(4))` would print: $\frac{3.14159...}{4}$.
2. `MEASURE_SPAN` denotes how the measure may be written without the use of "decimal point" time signature.
3. `BEAT_STRUCT` indicates beat structure, by which the beaming of the measure abides.

16. Mark Andre, *Un-fini I : 1995, für eine Harfenistin/einen Harfenisten (Harfe, Tam-tam, grosse Trommel)*, Neue Musik bei Carus (Stuttgart: Carus, 1997).

17. Mark Andre, *Un-fini II : pour clavecin (1996)* (Paris: Editions Durand, 1998).

18. Mark Andre, *Contrapunctus : pour piano* (Paris: Durand, 2006).

4. When you wish to go back to a regular time signature, use `\backToNormalTimeSignature`, otherwise the identical fractional time signature will keep showing up.

11.10.3 Code

```

1  \version "2.24.4"
2  \language "english"
3
4  decimalPointTimeSignature =
5  #(define-music-function
6    (timeSignatureToShow underlyingMeter beatStructure)
7    (list? fraction? number-list?)
8
9    (define (is-float? x)
10      (and (number? x) (inexact? x)))
11
12    #{
13      \time $underlyingMeter
14      \set beatStructure = $beatStructure
15      \override Staff.TimeSignature.stencil =
16      #ly:text-interface::print
17      \override Staff.TimeSignature.text =
18      #(markup
19        (make-override-markup
20          (cons 'baseline-skip 0)
21          (make-center-column-markup
22            (list
23              (if (= (length (car timeSignatureToShow)) 1)
24                (make-number-markup
25                  (number->string
26
27                    (car (car timeSignatureToShow))))
28                (make-line-markup
29                  (list
30                    (make-number-markup
31                      (number->string
32                        (car (car timeSignatureToShow))))
33                    (make-hspace-markup -0.5)
34                    (make-translate-markup
35                      '(0 . 0.15)
36                      (make-musicglyph-markup "period"))
37                    (make-hspace-markup -0.5)
38                    (if (not (is-float? (cadr (car timeSignatureToShow))))
39                      (make-number-markup
40                        (number->string
41                          (inexact->exact (cadr (car timeSignatureToShow))))
42                      ))
43                    (make-line-markup

```

```

44         (list (make-number-markup
45                 (number->string
46                 (inexact->exact
47                 (cadr (car timeSignatureToShow)))
48                 ))
49         (make-hspace-markup -0.5)
50         (make-translate-markup
51         '(0 . 0.15)
52         (make-musicglyph-markup "period"))
53         (make-hspace-markup -0.5)
54         (make-translate-markup
55         '(0 . 0.15)
56         (make-musicglyph-markup "period"))
57         (make-hspace-markup -0.5)
58         (make-translate-markup
59         '(0 . 0.15)
60         (make-musicglyph-markup "period"))))
61     )
62 )
63 )
64 )
65 (if (= (length (cadr timeSignatureToShow)) 1)
66     (make-number-markup
67     (number->string
68     (car (cadr timeSignatureToShow))))
69     (make-line-markup
70     (list
71     (make-number-markup
72     (number->string
73     (car (cadr timeSignatureToShow))))
74     (make-hspace-markup -0.5)
75     (make-translate-markup
76     '(0 . 0.15)
77     (make-musicglyph-markup "period"))
78     (make-hspace-markup -0.5)
79     (if (not (is-float? (cadr (cadr timeSignatureToShow))))
80         (make-number-markup
81         (number->string
82         (inexact->exact (cadr (cadr timeSignatureToShow))))
83         ))
84     (make-line-markup
85     (list (make-number-markup
86             (number->string
87             (inexact->exact
88             (cadr (cadr timeSignatureToShow))))
89             ))
90     (make-hspace-markup -0.5)
91     (make-translate-markup

```

```

92             '(0 . 0.15)
93             (make-musicglyph-markup "period"))
94         (make-hspace-markup -0.5)
95         (make-translate-markup
96         '(0 . 0.15)
97         (make-musicglyph-markup "period"))
98         (make-hspace-markup -0.5)
99         (make-translate-markup
100        '(0 . 0.15)
101        (make-musicglyph-markup "period"))))
102     )
103   )
104   )
105   )
106   )
107   ))
108   (make-hspace-markup -1))
109   #})
110
111   backToNormalTimeSignature =
112   {
113     \unset beatStructure
114     \revert Timing.TimeSignature.stencil
115     \revert Timing.TimeSignature.text
116     \revert Staff.TimeSignature.stencil
117     \revert Staff.TimeSignature.text
118   }
119
120
121   {
122     \decimalPointTimeSignature #'((2 5)(4)) 5/8 2,3
123     c'8 c' c' c' c'
124     \backToNormalTimeSignature
125     \time 3/4
126     c'2.
127   }
128
129
130   {
131     \decimalPointTimeSignature #'((9)(3 1415.)) 9/4 3,3,3
132     \tuplet 9/9 {c'4 4 4 4 4 4 4 4 4}
133     \backToNormalTimeSignature
134     \time 3/4
135     c'2.
136   }
137

```

11.10.4 Discussion

The structure of the code where the user specifies integer and decimal portions of either numerator, denominator, or both, resulted from the fact that the period "." by default appeared too close to the staff line and the denominator, possibly rendering the time signature difficult to read. In the code I made these periods appear via `\translate` feature, where I offset the period upward by the value of 0.15, allowing the period sign to be separated from the staff line and the denominator.

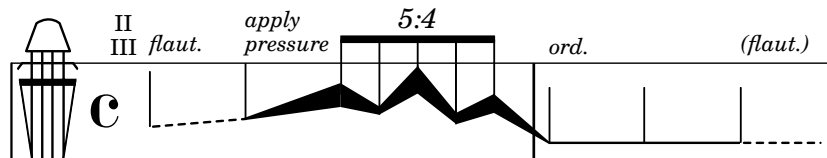
[Table of Contents](#)

Chapter 12

Combinations

This chapter presents examples that combine several snippets from the previous chapters. **Variables Used** provides a comprehensive list of all the variables required to generate the snippet. Among these, indented variables indicate "variables of a variable," i.e., dependent variables necessary for the main variables to function. The **Code** section only lists the score portion of the LilyPond code.

12.1 Prescriptive Notation for String Instruments



12.1.1 Description

An example of a prescriptive notation for a string instrument. Vertical placement of the notehead corresponds to the position at which bowing takes place. Horizontally it shows the change of the bow pressure against the string(s).

12.1.2 Variables Used

```
\stringPositionClef
  \stringPositionClefDesign
\dashedLineNotehead
\modularLineNotehead
\noteheadless
```

12.1.3 Code

```
1
2 \score {
```

```

3   {
4   \override Staff.StaffSymbol.line-positions = #'(6 -6)
5   \stringPositionClef
6   \dashedLineNotehead g'4
7       ^\markup {\fontsize #-4 \italic flaut.}
8       ^\markup \translate #'(-2.5 . -0) \center-column
9           {\translate #'(0 . -1.5) \fontsize #-4 II
10              \fontsize #-4 III}
11       a' #6
12   \modularLineNotehead a'
13       ^\markup \column {\translate #'(0 . -1.5)
14           \fontsize #-4 \italic apply \fontsize #-4
15           \italic pressure}
16       d'' #15 #150 #6
17   \override TupletNumber.text = #tuplet-number::calc-fraction-text
18   \stemUp \tuplet 5/4 {
19       \modularLineNotehead d''8 b' #150 #50 #2.5
20       \modularLineNotehead b' f'' #50 #175 #2.5
21       \modularLineNotehead f'' a' #175 #70 #2.5
22       \modularLineNotehead a' c'' #70 #120 #2.5
23       \modularLineNotehead c'' e' #120 #15 #3.5
24   }
25   |
26   \modularLineNotehead e'4
27       ^\markup {\fontsize #-4 \italic ord.}
28       e' #15 #15 #12
29   \noteheadless e'
30   \dashedLineNotehead e'
31       ^\markup {\fontsize #-4 \italic (flaut.)}
32       e' #5
33   }
34
35   \layout {
36       \context {
37           \Score proportionalNotationDuration = #(ly:make-moment 1/10)
38               \override SpacingSpanner.uniform-stretching = ##t
39       }
40   }
41 }
42

```

[Table of Contents](#)

12.2 Multiple Instances Of Spanners At Once

Two musical staves, A and B, illustrating multiple instances of spanners. Both staves are in 5/4 time and start with a tempo of 100. A "rall." (rallentando) spanner is applied to the second measure, with an arrow indicating the tempo change to 50. The music includes eighth notes, quarter notes, and a triplet of eighth notes in the third measure. Staff A has a box "A" at the beginning, and Staff B has a box "B" at the beginning.

12.2.1 Description

Invoking two or more Text Spanners (that require `\stopTextSpan` for them to finish their processes) all on one single layer could cause the spanners to behave unexpectedly. This entry is an attempt to avoid such unexpected behaviors by invoking a spanner per layer (A), or per staff line (B).

12.2.2 Variables Used

```
\startSlashedGraceMusic
\stopSlashedGraceMusic
\graceNoteBeforeBeatOn
\graceNoteBeforeBeatOff
\graceNoteAfterBeatOn
\graceNoteAfterBeatOff
\rallArrow
```

12.2.3 Code

```
1
2 %%%%%%%%%%%%%%% A %%%%%%%%%%%%%%%
3 \score {
4   \new Staff = "allElementsCombined"
5   \with {instrumentName = \markup {\fontsize #4 \box "A"}} {
6     \numericTimeSignature
7     \override Score.MetronomeMark.Y-offset = #5.75
8     \tempo 4 = 100
9     \time 5/4
```

```

10      <<
11      {
12          \tieNeutral \stemNeutral d'4~
13          \tuplet 3/2 {d'8 d'4}
14          \stemUp \grace {
15              \startSlashedGraceMusic \graceNoteBeforeBeatOn e'8 f'
16              \stopSlashedGraceMusic
17          } \graceNoteBeforeBeatOff g'4~
18          \stemNeutral g'8.[ \grace {
19              \startSlashedGraceMusic \graceNoteAfterBeatOn
20              e''16 c'' e' c' \stopSlashedGraceMusic
21          }
22          \graceNoteAfterBeatOff d''16]~
23          \tuplet 3/2 {d''8 d'8 d'8~} |
24          \time 4/4
25          d'1 \bar"||"
26      }
27      \\\
28      {
29          s4 \tuplet 3/2 {
30              s8 \override Voice.TextSpanner.Y-offset = #6.5
31              s4~\markup {\translate #'(0 . 6.5) \bold "rall."}
32              \rallArrow #4
33          } s2. \tempo 4 = 50 s4*4 \stopTextSpan
34      }
35      >>
36  }
37 }
38
39
40
41
42
43 %%%%%%%%%%%%%%% B %%%%%%%%%%%%%%%
44 \score {
45     <<
46     \new Staff = "tempoLine" \with {
47         \remove Clef_engraver
48         \remove Staff_symbol_engraver
49         \remove Time_signature_engraver
50     }
51     {
52         \numericTimeSignature
53         \override Score.MetronomeMark.Y-offset = #6
54         \tempo 4 = 100
55         \time 5/4
56         s4 \tuplet 3/2 {
57             s8 \override Voice.TextSpanner.Y-offset = #-2.25

```

```

58      s4~\markup {\translate #'(0 . 0) \bold "rall."}
59      \rallArrow #4} s2 \after 64*15 \stopTextSpan s8*2 |
60      \tempo 4 = 50  s4*4
61    }
62    \new Staff = "music"
63    \with { instrumentName = \markup {\fontsize #4 \box "B"}}
64    {
65      \tieNeutral \stemNeutral d'4~
66      \tuplet 3/2 {d'8 d'4}
67      \grace {
68        \startSlashedGraceMusic \graceNoteBeforeBeatOn e'8 f''
69        \stopSlashedGraceMusic
70      } \graceNoteBeforeBeatOff g'4~
71      g'8.[ \grace { \startSlashedGraceMusic \graceNoteAfterBeatOn
72        e''16 c'' e' c' \stopSlashedGraceMusic
73      }
74      \graceNoteAfterBeatOff d''16]~
75      \tuplet 3/2 {d''8 d'8 d'8~} |
76      \time 4/4
77      d'1 \bar"||"
78    }
79    >>
80  }

```

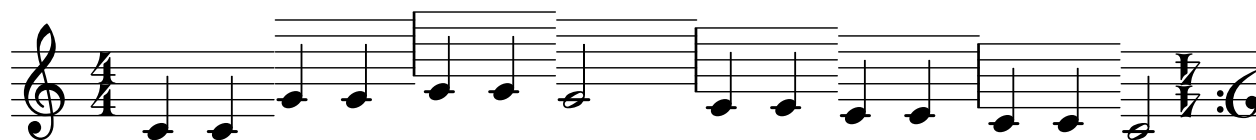
[Table of Contents](#)

Chapter 13

Miscellanies

This chapter presents snippets that do not really belong to any of the other preceding chapters but I learned tremendously from making. Quite often I have made these snippets as a diversion.

13.1 Shifting Staves, Rotated Clef and Time Signature



13.1.1 Description

Staff lines that are shifted so that, when the note moves away from the middle C, the staff lines move accordingly. The excerpt ends with a time signature and a clef that are rotated 180 degrees.

13.1.2 Code

```
1 \version "2.24.4"
2 \language "english"
3
4 staone = {
5   \stopStaff
6   \override Staff.StaffSymbol.line-positions =
7   #'(0 2 4 6 8)
8   \startStaff
9 }
10 statwo = {
11   \stopStaff
12   \override Staff.StaffSymbol.line-positions =
13   #'(1 3 5 7 9)
14   \startStaff
15 }
16 stathree = {
```

```

17   \stopStaff
18   \override Staff.StaffSymbol.line-positions =
19   #'(-1 1 3 5 7)
20   \startStaff
21 }
22 stafour = {
23   \stopStaff
24   \override Staff.StaffSymbol.line-positions =
25   #'(-2 0 2 4 6)
26   \startStaff
27 }
28 stafive = {
29   \stopStaff
30   \override Staff.StaffSymbol.line-positions =
31   #'(-3 -1 1 3 5)
32   \startStaff
33 }
34 stanorm = {
35   \stopStaff
36   \revert Staff.StaffSymbol.line-positions
37   \startStaff
38 }
39 {
40   \numericTimeSignature
41   \time 4/4
42
43   c'4 c' \staone g' g' \statwo a' a' \staone g'2
44   \stathree f'4 f' \stafour e' e' \stafive d' d' \stanorm
45   \override TextScript.outside-staff-priority = ##f
46   \once \override TextScript.extra-offset = #'(0 . -4.5)
47   c'2 ^\markup \concat {
48     {
49       \hspace #3 \rotate #180
50       {\compound-meter #'(4 4)}
51     }
52     {
53       \translate-scaled #'(1 . 0.5)
54       \rotate #180 \musicglyph "clefs.F"
55     }
56   }
57   \bar ""
58
59 }
60
61 \layout {
62   \context{
63     \Score    proportionalNotationDuration = #(ly:make-moment 1/7)
64   }

```

65 }

[Table of Contents](#)

Chapter 14

Exploring Scheme

14.1 Introduction

Scheme, one of the dialects of the Lisp family of programming languages, is used in LilyPond as its extension language. Scheme allows LilyPond users to explore the inner workings of the program, enabling significant customization. The snippets in this document would not exist without taking advantage of it.¹

However, learning Scheme can be daunting. In his unfinished book on Scheme and LilyPond, Urs Liska refers to its "thorny path."² While I have experience with Common Lisp (another Lisp dialect) from my work with OpenMusic, adjusting to Scheme's grammatical nuances still took some time.

This chapter does not aim to be a comprehensive guide to using Scheme in LilyPond.³ Instead, it offers suggestions for newcomers to familiarize themselves with Scheme.

14.1.1 Step 1a: Focus on the Scheme Language Itself

Scheme is a language distinct from LilyPond, and understanding this distinction is essential. For simpler LilyPond tasks, Scheme may not be necessary. However, when working with internal parameters, Scheme allows deeper customization. It is beneficial to first study Scheme independently, learning its syntax and concepts by writing simple code.

14.1.2 Step 1b: Get Used to Prefix Notation

Scheme, like its Lisp relatives, uses prefix notation (Cambridge Polish Notation). Here are examples:

```
(+ 12 34)
>> This expression results in the value of 46.
```

1. For newcomers: parts of LilyPond code written in Scheme are often enclosed in `#(` and `)`. Numerical values preceded by `#`, and number pairs such as `\#' (1 . -2)`, are also part of the Scheme language.

2. Urs Liska, *Understanding Scheme In LilyPond*, vol. 2024, December 19 (2020), Web Page, <https://scheme-book.readthedocs.io/en/latest/>.

3. For a deeper dive, refer to the resource by Liska, as well as Jean Abou Samra, *Extending LilyPond*, vol. 2024, December 19 (2021), Web Page, <https://extending-lilypond.gitlab.io/en/index.html>. LilyPond also provides its own Extending Manual: <https://lilypond.org/doc/v2.24/Documentation/extending/index>

```
(+ 4 (* 3 9))
```

```
>> This expression first resolves the multiplication: (+ 4 27), which is 31.
```

If you are new to this, I recommend starting with Daniel P. Friedman and Matthias Felleisen, *The little Schemer (4th ed.)* (Cambridge, MA, USA: MIT Press, 1996), ISBN: 0262560992. While you might be eager to dive into using Scheme in LilyPond, learning Scheme as a programming language will make the process smoother.⁴

14.1.3 Step 2: Study Lots of Snippets

Once familiar with Scheme, study how it integrates with LilyPond by reviewing snippets from LSR. Start with shorter examples and analyze their structure. Here is an example snippet for adding the *Schleifer* ornament:⁵ The corresponding code:⁶



Figure 14.1: LSR No. 1185: *Schleifer* Ornament.

```

1  % Implementation by Martin Straeten of the Schleifer ornament
2  % as used by Johann Sebastian Bach, contributed to the user
3  % mailing list. In this case, it functions like a set of (always?)
4  % two grace notes, hence using a modified grace note to represent
5  % it in LilyPond makes sense.
6  %
7  % Code styling and user interface by Simon Albrecht 2024.
8
9  schleiferMarkup = \markup {
10    \large \halign #.2 \raise #0.0
11    \combine
12    \halign #.8 \musicglyph "scripts.prall"
13    \rotate #140 \normalsize \raise #2.4 \musicglyph "flags.u3"
14  }
15  schleiferGrace =
16  #(define-music-function (note) (ly:music?)
17    #{
18      \grace {
19        \once\override NoteHead.stencil = #ly:text-interface::print
20        \once\override NoteHead.X-extent = #'(-2 . -0)
21        \once\override NoteHead.text = \schleiferMarkup
22        \once\omit Stem
23        \once\omit Flag

```

4. Liska and Samra's resources serve as excellent refreshers later on.

5. <https://lsr.di.unimi.it/LSR/Item?id=1185>

6. The mailing list thread referenced in the preamble is available at <https://lists.gnu.org/archive/html/lilypond-user/2021-09/msg00352.html>


```

24         $note
25     }
26     #})
27
28 \relative {
29     \time 3/8
30     \partial 8
31     \clef bass
32     \key c \minor
33     g8
34     \schleiferGrace c es8. d16 c8
35     c4
36 }
37 \addlyrics {
38     Ich ha -- be ge -- nug
39 }

```

The `\schleiferGrace` variable creates a customized ornament using Scheme’s `define-music-function` macro. For a deeper understanding of the macro syntax, refer to the *LilyPond – Internals Reference*.⁷

Taking the variable `\schleiferGrace`, we see that invoking it creates an instance of activating a Scheme function that starts at Line 16. `define-music-function` is a macro that allows you to create a function that operates on LilyPond.

According to *LilyPond – Internals Reference*, the syntax for `define-music-function` is:

```

(define-music-function (arg1 arg2 ...)
  (type1? type2? ...)
  function-body)

```

In the code, the argument’s name is `note`, and it is tested according to the type specified in `type1?`, which in this case is `ly:music?`. According to the *Internal Reference*, `ly:music?` is a function that checks whether the object—in this case, `note`—is a `Music` object. Thus, it becomes clear that this function will not work unless it is followed by a musical note.

From Line 17 to Line 26, we see that a LilyPond code snippet has been inserted, as `#{` and `#}` signify the boundary of the LilyPond code within the Scheme code. This means that as part of invoking the variable `\schleiferGrace`, it passes through this LilyPond fragment, which is responsible for creating a grace note. Here, the notehead of the grace note is replaced with `\schleiferMarkup`, which is defined in Lines 9 to 14 of the code.⁸

Lines 22 and 23 show that the stem and flag are omitted from the grace note, while Line 24’s `$note` signifies that the original argument `note` is called upon.⁹ In this way, the *Schleifer* ornament is

7. <https://lilypond.org/doc/v2.24/Documentation/internals/scheme-functions>

8. The technique of sequential overrides, invoking the Scheme command `#ly:text-interface::print`, sets the `.stencil` of the notehead to use whatever is defined in the `.text` parameter. This technique is frequently used and is very useful in customizing notation. See also: <https://lilypond.org/doc/v2.24/Documentation/notation/modifying-stencils>.

9. Refer to this page for the difference between `#` and `$`: <https://lilypond.org/doc/v2.24/Documentation/extending/lilypond-scheme-syntax>

created from a note that follows the variable `\schleiferGrace`. This note is transformed into a grace note with a customized stencil setting, all done within the Scheme code.

14.1.4 Step 3: Hack the Codes

Once you study a code and become familiar with how it operates, experimenting with the code by hacking is a good way to deepen your understanding. Below, I give one example using the preceding *Schleifer* ornament snippet.

The *LilyPond – Internal Reference* reveals that the object `NoteHead` has its own standard settings, as well as support for about a dozen other interfaces.¹⁰ One of them is the `grob-interface`, which makes it possible to change the color of a graphical object, or *Grob*.¹¹ Further reading in the *LilyPond – Notation Reference* shows that it is possible to override the color of an object.¹² Let us now tweak the *Schleifer* ornament code to allow us to change the ornament's color.

Following the reference, add the following line underneath `\once\override NoteHead.X-extent`:

```
\once\override NoteHead.color = #red
```

Running LilyPond now should produce the following result: Hard-coding a change like this may



Figure 14.2: LSR No. 1185: *Schleifer* Ornament in red.

be good for testing the waters, but we may want the *Schleifer* ornament in more than just one color. The beauty of extending LilyPond is that we can customize the Scheme code to allow for this flexibility.

Let us move on. We should now let the `define-music-function` know that we are adding an additional argument to specify the color. The first part of the code will look like this:

```
#(define-music-function (note schleiferColor) (ly:music? color?)
```

This adds the argument `schleiferColor`, which only accepts color, as indicated by the corresponding test function `color?`.

Then, implement this argument in the sequence of `\once\override` processes. The line `NoteHead.color` can now be changed to:

```
\once\override NoteHead.color = #schleiferColor
```

Now, the variable `\schleiferGrace` requires one more argument to specify the ornament's color. The entire code should look like this:

```
1 schleiferMarkup = \markup {
2   \large \halign #.2 \raise #0.0
3   \combine
4   \halign #.8 \musicglyph "scripts.prall"
```

10. <https://lilypond.org/doc/v2.24/Documentation/internals/notehead>

11. https://lilypond.org/doc/v2.24/Documentation/internals/grob_002dinterface

12. <https://lilypond.org/doc/v2.24/Documentation/notation/inside-the-staff#coloring-objects>

```

5   \rotate #140 \normalsize \raise #2.4 \musicglyph "flags.u3"
6   }
7
8   schleiferGrace =
9   #(define-music-function (note schleiferColor) (ly:music? color?)
10     #{
11       \grace {
12         \once\override NoteHead.stencil = #ly:text-interface::print
13         \once\override NoteHead.X-extent = #'(-2 . 0)
14         \once\override NoteHead.color = #schleiferColor
15         \once\override NoteHead.text = \schleiferMarkup
16         \once\omit Stem
17         \once\omit Flag
18         $note
19       }
20     #})
21   \relative {
22     \time 3/8
23     \partial 8
24     \clef bass
25     \key c \minor
26     g8
27     \schleiferGrace c #green es8. d16 c8
28     c4
29   }
30   \addlyrics {
31     Ich ha -- be ge -- nug
32   }

```

This produces the following output:



Figure 14.3: LSR No. 1185: *Schleifer* Ornament in green.

Notice that on Line 27, `#green` has been added. You can change this to any of the colors listed under "Normal Colors" in the *Notation Reference*,¹³ such as `#'"lightsalmon"`, `#(x11-color "medium turquoise")`, or even `#'"#5e45ad"`.

As an exercise, try replicating the following excerpt:¹⁴



Figure 14.4: Can you replicate this?

Table of Contents

13. <https://lilypond.org/doc/v2.24/Documentation/notation/list-of-colors>

14. See [LSR1185e3.ly](https://lilypond.org/doc/v2.24/Documentation/notation/list-of-colors) for the answer.

14.2 Example 1: Creating a Time Signature with Its Compound Meter Form

On January 1, 2025, I came across a post by an anonymous user on Facebook.¹⁵ The post asked if it would be possible to create a time signature that had its beat structure expressed in the form of a compound meter. Something like this:



Figure 14.5: What the anonymous user wanted to achieve.

I responded to the post with relevant email threads on `lilypond-user` mailing list. I commented that it would be possible to override `TimeSignature.stencil` with custom-made time signatures. Incidentally, I was making a series of [Fractional Time Signatures](#), which used this method.

The code for the aforementioned example is as follows:

```
1 \version "2.24.4"
2
3 {
4   \clef "G"
5   \time 9/8
6   \set beatStructure = #'(2 2 2 3)
7   \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
8   \once \override Timing.TimeSignature.text = \markup
9   {
10    \override #'(baseline-skip . 0)
11    \center-column \number {9 8}
12    \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
13    \hspace #-0.75
14    \override #'(baseline-skip . 0)
15    \center-column \number {{2+2+2+3} 8}
16    \hspace #-0.75
17    \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
18    \hspace #-1
19  }
20  \repeat unfold 9 {<e' g'>8 }
21 }
```

I realized that, while this might be an acceptable method if such time signatures appeared only once or twice in a piece, it may become problematic if I had to copy and paste this code every time I have such a time signature. Normally this could easily be resolved by making a variable out of `\override` clauses; however, a piece of music may use time signatures of this form in different configurations, just as the following example:

The code:

```
1 \version "2.24.4"
```

15. <https://www.facebook.com/groups/gnulilypond/posts/10162467719483529/>



Figure 14.6: More compound meters.

```

2
3 {
4   \clef "G"
5   \time 9/8
6   \set beatStructure = #'(2 2 2 3)
7   \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
8   \once \override Timing.TimeSignature.text = \markup
9   {
10    \override #'(baseline-skip . 0)
11    \center-column \number {9 8}
12    \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
13    \hspace #-0.75
14    \override #'(baseline-skip . 0)
15    \center-column \number {{2+2+2+3} 8}
16    \hspace #-0.75
17    \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
18    \hspace #-1
19  }
20  \repeat unfold 9 {<e' g'>8}
21
22  \time 11/8
23  \set beatStructure = #'(2 2 2 3 2)
24  \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
25  \once \override Timing.TimeSignature.text = \markup
26  {
27    \override #'(baseline-skip . 0)
28    \center-column \number {11 8}
29    \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
30    \hspace #-0.75
31    \override #'(baseline-skip . 0)
32    \center-column \number {{2+2+2+3+2} 8}
33    \hspace #-0.75
34    \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
35    \hspace #-1
36  }
37  \repeat unfold 11 {<e' g'>8 }
38
39  \time 7/8
40  \set beatStructure = #'(2 3 2)
41  \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
42  \once \override Timing.TimeSignature.text = \markup

```

```

43 {
44   \override #'(baseline-skip . 0)
45   \center-column \number {7 8}
46   \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
47   \hspace #-0.75
48   \override #'(baseline-skip . 0)
49   \center-column \number {{2+3+2} 8}
50   \hspace #-0.75
51   \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
52   \hspace #-1
53 }
54 \repeat unfold 7 {<e' g'>8 }
55
56 }

```

Writing as long of a code as this (for just three measures!) would be cumbersome, indeed. What could help is to come up with a music function, using the Scheme.

14.2.1 Step 1: Analyze What Could Be Automatized

I quote the code for the first example of this section again. This time, however, I turn the variables that could change each time I create an instance of this kind of time signature, into **red**:

```

1  \version "2.24.4"
2
3  {
4    \clef "G"
5    \time 9/8
6    \set beatStructure = #'(2 2 2 3)
7    \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
8    \once \override Timing.TimeSignature.text = \markup
9    {
10     \override #'(baseline-skip . 0)
11     \center-column \number {9 8}
12     \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
13     \hspace #-0.75
14     \override #'(baseline-skip . 0)
15     \center-column \number {{2+2+2+3}{8}}
16     \hspace #-0.75
17     \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
18     \hspace #-1
19   }
20   \repeat unfold 9 {<e' g'>8 }
21 }

```

14.2.2 Step 2: Write the Code

It would be good if this function could accept the following as arguments:

- Time signature of the measure as defined normally in the LilyPond function `\time`. For this, I will set `timesig` as the name of the argument, that tests its value with `fraction?`.
- The customized stencil of the time signature. I need to declare how it looks, namely:
 - Overall time signature;
 - Numerator portion of the compound meter, and;
 - Denominator portion of the compound meter.

It should look similar to how the LilyPond function `\compoundMeter` that accepts a list of lists. For this, I will set `beatstruct` as the name of the argument, that tests its value with `list?`.

I will now build the rest of the function. Notice the way the Scheme code references various locations of a list, using `car`, `cadr`, and so on:

```

1  \version "2.24.4"
2
3  compoundTimeWithBeatStructure =
4  #(define-music-function (timesig beatstruct) (fraction? list?)
5    #{
6      \time #timesig
7      \set beatStructure = #(cadr beatstruct)
8      \once \override Timing.TimeSignature.stencil = #ly:text-interface::print
9      \once \override Timing.TimeSignature.text = \markup
10     {
11       \override #'(baseline-skip . 0)
12       \center-column \number
13       {
14         #(number->string (car (car beatstruct)))
15         #(number->string (cadr (car beatstruct)))
16       }
17       \center-column {\fontsize #6 \musicglyph "accidentals.leftparen"}
18       \hspace #-0.75
19       \override #'(baseline-skip . 0)
20       \center-column \number
21       {
22         {#(string-join (map number->string (cadr beatstruct)) "+")}
23         #(number->string (car (caddr beatstruct)))
24       }
25       \hspace #-0.75
26       \center-column {\fontsize #6 \musicglyph "accidentals.rightparen"}
27       \hspace #-1
28     }
29   #}
30
31 )
32
33 {

```



```

34 \compoundTimeWithBeatStructure 9/8 #'((9 8)(2 2 2 3)(8))
35 \repeat unfold 9 {<e' g'>8}
36 \compoundTimeWithBeatStructure 11/8 #'((11 8)(2 2 2 3 2)(8))
37 \repeat unfold 11 {<e' g'>8}
38 \compoundTimeWithBeatStructure 7/8 #'((7 8)(2 3 2)(8))
39 \repeat unfold 7 {<e' g'>8}
40 }

```

Thus, there is now a function called `\compoundTimeWithBeatStructure`, whose grammar is:

```

\compoundTimeWithBeatStructure
    TIME_SIGNATURE #'((TIME_SIGNATURE)(BEAT_STRUCTURE)(DENOMINATOR))

```

Running the code will result in the identical snippet as [the previous figure](#):



Figure 14.7: The same result as before with a shorter code.

[Table of Contents](#)

Bibliography

- Adès, Thomas. *Asyla : for large orchestra*. Faber Music, 1997.
- Andre, Mark. *Contrapunctus : pour piano*. Paris: Durand, 2006.
- . *Un-fini I : 1995, für eine Harfenistin/einen Harfenisten (Harfe, Tam-tam, grosse Trommel)*. Neue Musik bei Carus. Stuttgart: Carus, 1997.
- . *Un-fini II : pour clavecin (1996)*. Paris: Editions Durand, 1998.
- Beyer, Stefan. *Bleib hier. Schau zu. Mach kein Geräusch*. Manuscript, 2017.
- . *Marsch*. Manuscript, 2013-14.
- . *Mittel und Zwecke (Boulevard)*. Manuscript, 2014.
- . *Most of My Clients Come Back*. Manuscript, 2012-13.
- Boulez, Pierre. ... *explosante-fixe ... transitoire VII : (version 1991/93)*. Universal Edition, 1991.
- . *Pli selon pli: V. Tombeau*. London: Universal Edition, 1971.
- . *Sur incises : pour trois pianos, trois harpes et trois percussions-claviers (1996/1998)*. Universal Edition, 1998.
- Czernowin, Chaya. *At the fringe of our gaze : for Orchestra and Concertino Group*. Schott, 2012/13.
- . *Lovesong : for mixed ensemble*. Schott, 2010.
- . *Streams (Slow Summer Stay I) : for 8 players*. Schott, 2012.
- . *String Quartet*. Schott, 1995.
- Friedman, Daniel P., and Matthias Felleisen. *The little Schemer (4th ed.)*. Cambridge, MA, USA: MIT Press, 1996. ISBN: 0262560992.
- Gould, Elaine. *Behind bars : the definitive guide to music notation*. London: Faber Music, 2011. Book.
- Levine, Carin, and Christina Mitropoulos-Bott. *The techniques of flute playing = Die Spieltechnik der Flöte*. Kassel ; New York: Bärenreiter, 2003.
- Liska, Urs. *Understanding Scheme In LilyPond*. Vol. 2024. December 19. 2020. Web Page. <https://scheme-book.readthedocs.io/en/latest/>.
- Onishi, Yoshiaki. *Gz II : for two accordions*. Brühl and Berlin: Edition Gravis, 2024.

- Salzedo, Carlos. *L'étude moderne de la harpe... Modern study of the harp*. 3 p.l., 53 p. New York - Boston, G. Schirmer, 1921.
- Samra, Jean Abou. *Extending LilyPond*. Vol. 2024. December 19. 2021. Web Page. <https://extending-lilypond.gitlab.io/en/index.html>.
- Sparnaay, Harry. *The Bass Clarinet: A Personal History*. Periferia Sheet Music, 2012.
- Takemitsu, Tōru. *Fantasma/cantos : for clarinet and orchestra*. Schott ; Schott Japan, 1993.
- . *Les yeux clos II : for piano*. Schott ; Schott Japan, 1990.
- Weisberg, Arthur. *Performing twentieth-century music : a handbook for conductors and instrumentalists*. New Haven: Yale University Press, 1996.

Appendices

Appendix A: Resources

As I taught LilyPond in a special topic course at the University of Delaware in Fall 2024, I compiled a list of links to useful websites and pages. It is in no way intended as a comprehensive list; instead, I list some essential pages that I have frequently looked up and found very useful. This page is subject to frequent revision.

On LilyPond

- Website: <https://lilypond.org/>
- Installing: <https://lilypond.org/doc/v2.24/Documentation/learning/installing>
- Manuals: <https://lilypond.org/manuals.html>

Text Editor for LilyPond

- Frescobaldi (Editor): <https://frescobaldi.org/>

Coding LilyPond

- Cheat Sheet: <https://lilypond.org/doc/v2.24/Documentation/notation/cheat-sheet>
- Snippets: <https://lilypond.org/doc/v2.24/Documentation/web/snippets>
- LilyPond Snippet Repository: <https://lsr.di.unimi.it/>

Mailing List

- Mailing list: <https://lists.gnu.org/mailman/listinfo/lilypond-user>
- Archives 1 <https://lists.gnu.org/archive/html/lilypond-user/>
- Archives 2 <https://www.mail-archive.com/lilypond-user@gnu.org/>

Advanced Topic on LilyPond

- LilyPond – Extending v2.24.4: <https://lilypond.org/doc/v2.24/Documentation/extending/index#top>
- Scheme (in LilyPond): <https://scheme-book.readthedocs.io/en/latest/>
- Extending LilyPond: <https://extending-lilypond.gitlab.io/en/extending/index.html>
- Scheme Resources <https://www.gnu.org/software/guile/learn/#scheme-resources>
- PostScript Manual: <https://www.adobe.com/jp/print/postscript/pdfs/PLRM.pdf>
- PostScript Tutorial: <https://paulbourke.net/dataformats/postscript/>

Troubleshooting

- [The default text font for LilyPond doesn't seem to work \(Mac\)](#)
- [Frescobaldi freezes upon loading](#)

Miscellaneous Items

- About Emmentaler font: <https://lilypond.org/doc/v2.25/Documentation/notation/the-emmentaler-font>

[Table of Contents](#)