

Alignment of Scores in Text Documents

`lyluaTeX`¹ is a package for `LuaTeX` that manages the engraving and integration of musical scores in text documents. The scores are encoded in chunks of LilyPond input which can be written directly in the document or maintained in standalone files. They are compiled using LilyPond and cached intelligently. Line width and staff size can be inferred from the surrounding text (but also configured explicitly). The following Beethoven excerpt is included twice, first normally and then with a smaller staff size in a two-column section:

Allegro con brio

VI. I
VI. II
Vla.
Vc.

p

6

f

11

p

¹<https://github.com/jperon/lyluaTeX>

Example 1: Score automatically generated from an included LilyPond file

And now the same score is engraved in two columns with the staff size manually reduced (in the \LaTeX include). Note that the same LilyPond file is simply used again.

Two important features can be demonstrated in these two examples: automatic engraving and individual handling of systems. Depending on the current line width the same score input is rendered twice with different settings, so it is not necessary to manually make the score “fit” into the text in the notation software. And if the settings change (for example when rendering to different output media or adjusting the space between the columns) one doesn’t have to go back to the notation software to update *all* used scores. Instead they are recompiled automatically upon the next \LaTeX run.

The other aspect is that the resulting systems of the score are generated as independent image files. This makes the score float freely through the text, allowing \LaTeX to break pages and columns automatically.

Allegro con brio



Example 2: Engraved from the same file

But the main point of this document is to provide material for discussing the *alignment* of scores to the surrounding text. The examples so far show an achievement of the *lyLuaTeX* package: when aligning the score to the text the *staff lines* exactly match the line width of the text, while brackets and instrument names protrude into the left margin. This is actually a non-trivial task, as the width of that margin material is arbitrary and can't reasonably be predicted, at least not automatically and reliably. In order to properly handle that task *lyLuaTeX* reads that width from an intermediate PostScript file and applies the corresponding negative offset when including the image file for each system.

However, while this is a great default behaviour there are valid reasons for *not* wanting such protrusion to (always) happen. For example it might be considered inappropriate to use the margin at all (maybe there are other elements to be printed in the margin) or it may look awkward or actually run off the page when the margin text is very wide:



Example 3: Long text protruding in the left margin

Which of the following approaches would be correct or acceptable from the typographical point of view? Additional suggestions are welcome!

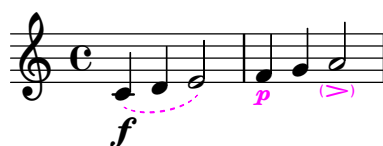
- indent the score so the margin text is aligned with the margin of the text?
- indent by *some* extent, if so, what extent?
- Do *not* handle this and expect the relevant information to be given differently, for example with text markup or in a caption?

If it is desirable to provide a way to include scores *without* protrusion, i. e. aligning the leftmost printed element with the surrounding text, there are technical challenges because we don't know the width of the margin text beforehand.

If we would simply not apply the negative offset the score would of course protrude into the *right* margin, which is unacceptable. It would on the other hand be simple to scale the image to fit exactly into the line width, but this would scale it down and make it inconsistent with other scores. Somewhat less unacceptable but still ugly and unprofessional. Obviously the native line width of the score would have to be reduced, which could be done in two ways:

- Give the user an option to manually specify an indentation width. However, this will *not* make the resulting image match with the text’s line width (at least not without tedious trial & error), and there would be the question of whether it should be left-aligned, right-aligned, or centered to the paragraph.
- Determine the actual value in a two-pass process. In a first compilation LilyPond produces the score with the desired line width. From that the width of the margin elements can be read, and then LilyPond can compile the score once more, with the line width reduced accordingly. This approach will perfectly align both the leftmost printed item and the right edge of the system with the surrounding text while keeping the staff height consistent. But it will require the score to be compiled twice, which requires non-neglectable amounts of time. And it has to be noted that shortening the line width *may* affect the line breaking in unexpected ways.

Another situation to be discussed is the alignment of shorter music examples. Typically these are not justified to span full lines but centered, as is demonstrated in the following example:



Example 4: Short centered music example

This example didn’t include any margin items so the system is nicely centered together with the paragraph and the caption. If on the other hand there *is* something printed in the margin the question arises what should be centered, the staff lines or the whole image. The following two examples show the behaviour *with* protrusion (as currently *ly*Lua \TeX can not do otherwise (which is the reason for this discussion anyway)), one of them with a ridiculously long text, just to show the effect. The question is if in such cases it would be better to center the *whole* image or at least give the user the option to choose (globally and per score).



Example 5: Short centered music example – staff lines centered



Example 6: Short centered music example with loooong text in front

For continuing the development some questions about the interface for \LaTeX users have to be discussed. It seems we are dealing with two terms: *protrusion* and *indentation*. The current behaviour as shown in this document is clearly “with protrusion”. But if the whole image is

aligned instead of the staff lines is that “without protrusion” or “with indentation”? And would it make a difference in terminology if the user chooses to automatically align the whole image or if they create an explicit indentation (shorten the line width)?

Independently from the *naming* of the features it should be decided which of the features should be configurable globally (package option), in the middle of the document (command to change behaviour/values), or on a per-score basis (optional arguments). Please give any feedback on the package’s issue tracker².

²<https://github.com/jperon/lylualatex/issues/36>