Improving margin paragraphs

Stephen Hicks

1 Introduction

A common frustration among authors using LaTeX to typeset books with significant margin material is that long margin notes often run off the bottom of the page, as shown in Figure 1. Several years ago, Andy Ruina approached me with an idea for solving this problem by completely rewriting LaTeX's margin code, at which point I developed an early version of the marginfix package, the subject of this article.

In this article, I'll start by looking in more detail at the problem and some common workarounds. After this, I'll explain the basic idea behind marginfix's solution. Finally, I'll describe a variety of options allowed by marginfix to further tweak margin note placement.

2 The problem

For a quick glance at the problem, the reader is encouraged to typeset the document in Figure 2. Doing so will yield a full-page paragraph with a margin note tied to the end of the paragraph, and this note clearly overflows past the bottom of the page.

What's happening here is that when the user calls `\marginpar`, LaTeX goes through a number of steps (Lamport, Mittelbach, and Rowley, 2004). To start, the content of the margin note (in the example, `\Large\lipsum[2]`) is saved into a pair of floating boxes (one for right-hand notes and the other for left-hand). LaTeX then inserts a specific large negative penalty into the outer vertical list, invoking TeX's `\output` routine, where the amount of text typeset so far on the page allows determining the vertical position of the `\marginpar` callout. This vertical position is compared to the position of the bottom of the previous margin note (if there is one), allowing LaTeX to shift the new note downward in an effort to prevent intersection. After this shift is computed, the box containing the margin note is shifted all the way to the left or right of the page (as appropriate), smashed (its height and depth are set to zero), and added to the outer vertical list.

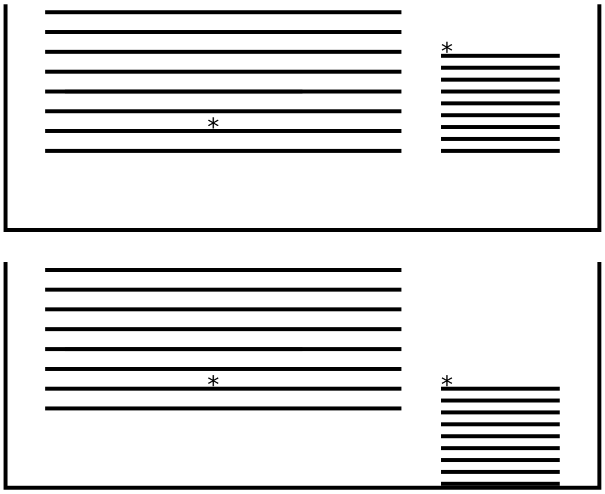

Figure 1: What the user expects (top) versus the default LaTeX output (bottom)

```
\documentclass{memoir}
\usepackage{lipsum}
\begin{document}
\sloppy\Huge\lipsum[1]
\marginpar{\Large\lipsum[2]}
\end{document}
```

Figure 2: Problems with LaTeX's built-in margin notes

This algorithm has both advantages and disadvantages. First, it's relatively straightforward and easy to understand, even if the behavior is less than desirable. Another advantage is that it automatically deals with vertical glue in the middle of a page. If a `\vskip` is stretched or shrunk and there is a `\marginpar` beneath it, the note is automatically moved along with the stretch or shrink, since its box is inside the lower paragraph. There are also a number of disadvantages. First, it's still possible for margin notes to intersect, if a skip between paragraphs with nearly-intersecting notes is shrunk. More importantly, there is no provision for notes to move upward on a page or to defer to later pages. Thus, if a note starts on the bottom of a page, it has nowhere to go but through the bottom margin.

3 Common workarounds

Any author who has written a substantial amount of LaTeX has developed a bag of tricks to coerce LaTeX's behavior when it's being particularly stubborn. There are a few such tricks for dealing with problematic margins.

- The simplest workaround is inserting a negative `\vskip` at the start of the marginnote (i.e. just before `\Large` in Figure 2). In Figure 2, the proper value is something like `-30pc`. But if the margin note changes (say, to `\lipsum[3]`), or if the page breaks change, then the value must be re-tweaked.
- The memoir package (Wilson, 2010) provides a workaround in the `\sidebar` macro, a complete (and incompatible) alternative to `\marginpar`

that provides a flowing column in the margin. This sufficiently addresses the problems with overflowing margins, but it makes no attempt to localize the margin material anywhere near its callout location (see the effect by changing `\marginpar` to `\sidebar` in Figure 2).

- Occasionally authors will find margin notes appearing on the wrong side of the page. This is due to LaTeX deciding on which side a margin note should appear at a different time from when it decides the page breaks. If a note looks like it will be on the bottom of an odd page, LaTeX attaches it on the right; but if a page break is inserted before the margin note, it will be wrong. The common solution here is provided by the package mparhack (Sgouros and Ulrich, 2005), which stores the actual page on which a note appears in the `.aux` file.

4 A new approach

Since LaTeX's margins already make use of floating boxes (that is, *inserts*), it is a natural extension to allow the notes to actually float. We can state our goals simply (the parenthetical notes refer to the step in our routine that enforces each goal):

- Each margin note is close to its callout (3b)
- No notes intersect (3b) or fall off the page (3c)
- Margin notes *may* float to later pages (3a)

Thus, the basic outline of the margin routines in marginfix is as follows:

1. When `\marginpar` is called, no change from LaTeX's behavior.
2. When the `\output` routine is entered for the first time, rather than setting the note immediately, instead append its boxes and the callout's page position to a token register (`\marginlist`).
3. When the `\output` routine is called and TeX actually builds the page, insert our own code to assemble the margin column, described below.
4. Once the margin column is assembled, we attach it to either side of the main column with an `\hbox`.

The interesting work here happens in step 3. We break this up into several substeps, as shown in Figure 3:

3a. Build a list of margin notes that are guaranteed to go on the current page by taking as many notes as possible from the front of `\marginlist` such that the total height of the notes does not exceed `\textheight`.

3b. Working down from the top, insert compressible "glue" into the list of notes on this page so that stacking all the boxes and glue places each note at, or below (in the case of notes that would otherwise intersect), its callout location (note that if we performed this step alone, we would reproduce LaTeX's margin routine).

3c. Working upwards from the bottom, remove as much glue as needed so that no notes overflow into the bottom margin.

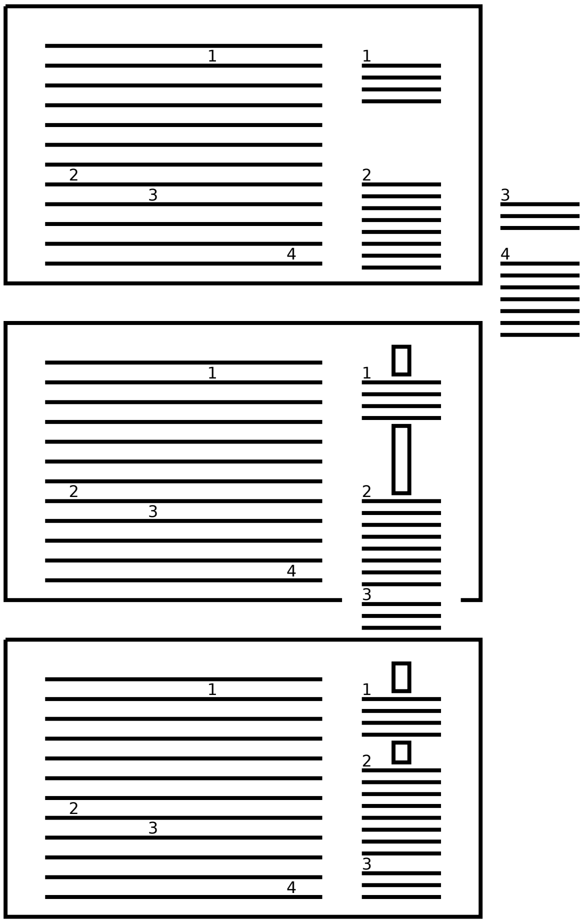

Figure 3: A new approach to building margins. We start by attaching each note to its callout position (top). We next push the notes down the margin until they don't intersect, and insert compressible glue (shown as narrow boxes) between non-abutting notes (middle). Note that the fourth note has been deferred because we only have room for 20 lines of margin material on the page. Finally, we work upwards, compressing glue as needed until all notes fit (bottom).

More concisely, we "expand down, compress up".

There are some interesting cases to consider here. If step 3a admits exactly `\textheight` of margin material then all the glue added in step 3b will be removed in 3c, resulting in a densely packed margin, comparable to `\sidebar`'s output. If there are not too many notes so that 3a doesn't defer anything, and if nothing occurs too close to the bottom of the page, so that 3c doesn't remove any glue, then we have reproduced LaTeX's output.

5 Options

The marginfix package provides a number of buttons and knobs that allow fine-tuning of the output.

- `\marginparpush` works the same as in standard LaTeX, adding a fixed incompressible space between notes.
- `\marginskip`⟨*length*⟩ appends an incompressible gap of a given *length* to the margin.
- `\mparshift`⟨*length*⟩ vertically shifts the next note by *length*.
- `\extendmargin`⟨*length*⟩ makes this page's margin longer by *length*.
- `\clearmargin` stops new material from going into this margin.
- `\blockmargin`...`\unblockmargin` makes a gap in the margin where no material can go, which is useful for extra-wide figures or equations.
- `\marginphantom`⟨*length*⟩ makes a gap, similar to `\(un)blockmargin`, except that one point is given as a vertical displacement from the other point, rather than specifying both points.

Outside of these macros, the package is simply a drop-in replacement of `\marginpar`.

6 Concerns and future work

A number of issues have recently come up in discussion with authors and developers. First, the initial version of step 2 used `\@pageht`, set by LaTeX's margin routines, to determine the callout position, but it was pointed out that this is inaccurate if any of the glue in the main column is stretched (or shrunk). Because LaTeX attaches the margin note to the main column immediately, the only effect this has in LaTeX is to allow the notes to possibly intersect if the glue shrinks enough; but since we put off note placement to the end, any stretch or shrink will cause misalignment. An initial thought is to calculate the stretchability above a note and then determine the glue set before building the margin column, but neither of these operations is possible (without doing cube roots of penalties, at least). Instead, we can use pdfTeX's `\pdflastypos` to determine (on the second pass) where the callout actually is. One benefit here is that we no longer need the premature output routines to find the vertical position.

Several features have been requested that are not yet implemented, but are not beyond the realm of possibility. LaTeX provides the ability to force a margin note to go into the opposite margin (with `\reversemarginpar`), so that both the left and right margin can have notes. This could be achieved by repeating the margin-building step twice, provided that each note clearly specifies into which margin it belongs (if notes are allowed to float between margins, the optimization seems to become more difficult). Another possible feature is allowing long notes to flow onto the following page, as memoir's `\sidebar` does. Because we can push margin notes upward, there is no reason to break a note unless the entire margin is full (much like the flowing of a full `\sidebar`). If necessary, however, a `\vsplit` would get the job done.

References

Lamport, Leslie, F. Mittelbach, and C. Rowley. "ltoutput.dtx". 2004. Available from CTAN, `macros/latex/base`.

Sgouros, Tom, and S. Ulrich. "mparhack.sty". 2005. Available from CTAN, `macros/latex/contrib/mparhack`.

Wilson, Peter. "The Memoir Class". 2010. Available from CTAN, `macros/latex/contrib/memoir`.

◇ Stephen Hicks
`sdh (at) google dot com`
(This work was conducted independent of the author's role at Google.)