A system for editorial comments

Boris Veytsman

Abstract

Copyediting a paper requires a back and forth exchange between the authors and editors. The traditional way to communicate editorial notes to the authors is marginalia. This paper discusses a system for typesetting layered marginal comments.

1 Introduction

Scientific publishing has been a wildly profitable enterprise, generating 40% profit margins [1], comparable to those in the illegal drug market [6]. However, preprint servers [2] provide a way to quickly communicate scientific results for free. This raises a question: what value is added by the publishers justifying the fees they collect either from the readers (in the traditional model), or from the authors (in the open access model), or even from both? Among the most often proposed arguments in favor of the traditional scientific publishers are (1) gatekeeping by peer review, and (2) post-submission work on a paper: copyediting and typesetting.

The first argument is beyond the framework of this paper. As to the second, it is obvious that clear prose and good typesetting help to convey the authors' thought and improve the readability of papers. Copyediting becomes even more important in the age of AI-generated prose, which is nowhere as clear and concise as is needed for the readers — especially since some authors now write in a style too close to AI slop.

Typesetting is also very important. Even if the authors do their own typesetting in (LA)TEX, a professional typesetter may greatly improve the paper's appearance — which influences both æsthetics and reading comprehension of the text. This is especially true with respect to mathematical typesetting, where small changes in the code often greatly improve the result.

We will loosely refer to both style and typesetting improvements as copyediting.

A copyeditor needs to communicate with the author, showing the changes made and asking questions about less obvious changes. Traditionally it was done by special marks and marginal comments on the galley proofs [10]. More modern systems try to imitate or replace this style using electronic documents. In this paper we discuss these attempts and propose a new system based on adding commands for typeset notes to the `.tex` file.

doi.org/10.47397/tb/46-2/tb143veytsman-comments

In the next sections we discuss the state of the art, and then the interface and implementation of the new solution. In the last section we lay out the plans for the future.

2 The state of the art

Consider a simple example of the author's manuscript shown in Figure 1. A copyeditor may find several problems with it:

1. "A most interesting relation" is not good English. Either "The most" or "A very" should be used.
2. Inline equations with fractions and integrals do not look good. If space permits, the equation in the first line should be displayed.
3. "Deriving" is an obvious typo.
4. Parentheses in the displayed equations are too short.
5. Depending on the house style one may insert an extra thin space before dx. Some journals also insist on making the differential symbol upright: $\mathrm{d}x$, but we will assume this is not the case.

A copyeditor with access to the TEX source can correct all these issues but the first one: it is up to the author to choose the right option. Also, a copyeditor must show the changes to the author and have them either approved or rejected.

An efficient way to do this is with PDF annotations. A number of modern tools allow to edit PDF files leading to the result shown in Figure 2. This is a very efficient way to communicate with the author, even though it is difficult to include TEX-formatted comments, which may be needed to discuss mathematics. Its disadvantage is transiency: the comments are bound to the current version of PDF and disappear if one recompiles the source. This makes their use not very convenient for the continuous workflow.

Another way to communicate the changes is to use the *latexdiff* tool [7]. An example of its use is shown on Figure 3. While *latexdiff* can be indispensable to show the differences between two LATEX files, its output can be rather messy. It does not show the final typeset text. Also, it is not clear how to ask the author about the proposed options.

A versatile and powerful way to communicate is to add comments in the source, as shown in Figure 4. However, it requires the author to switch between the source and the typeset output — a skill we cannot require from the authors.

A variation of source code comments is comments used by some integrated environments like Overleaf or TEXFolio (disclaimer: the parent companies of these environments have been sponsors of TUG2025). In these environments, comments are technically not part of the source code, but are associated with it (Figure 5). Often the environments integrate these comments in the workflow, allowing the authors and editors to open them, answer, close, etc. Still, these comments require switching between the source and the typeset material, which presents certain difficulties for the authors.

While the methods listed above help to communicate, they still lack the directness of "pen and paper" comments. Thus several journals, for which I wrote LATEX classes, asked me to develop a more convenient system. Interestingly enough, their requirements were close. This may indicate common practices and needs.

Below we describe the interface and implementation for this system.

3 New solution: requirements and interface

There is an inherent asymmetry between a copyeditor and an author. A copyeditor is supposed to be a TEXnician who may learn new LATEX commands, customize and use them. On the other hand, we do not expect any such knowledge or skills from the author. This prompted the main requirement for the system: it must be simple and intuitive for the consumer of the comments, but it may be relatively sophisticated and customizable to their producer.

Still, we want to make the copyeditor's life easier; thus another requirement is not to burden them with the painstaking deletion of comments. Therefore the comments should "disappear" in the final version of the paper. However, it is important that this disappearance does not change any line- or page-breaks. Indeed, part of the typesetter's work is the careful adjustment of breaks. It would be unacceptable if this work had to be done again in the final version.

Often, communication with authors includes several rounds of galley proofs exchange. This prompts another requirement: the editorial comments should be arranged in "layers", each of which can be switched on and off.

Based on these requirements, the following interface has been developed [8].

There are two editorial commands:

1. `\EditorialComment[`⟨*layer*⟩`]{`⟨*text*⟩`}[`⟨*shift*⟩`]`,
2. `\Highlight[`⟨*layer*⟩`]{`⟨*contents*⟩`}`.

The first command typesets `{`⟨*text*⟩`}` in the margins if the global option `manuscript` is chosen for the document class *and* the comments layer `[`⟨*layer*⟩`]` is selected for display. The color of the marginal comment corresponds to the `{`⟨*layer*⟩`}`. The second optional argument `[`⟨*shift*⟩`]` is discussed in the next section.

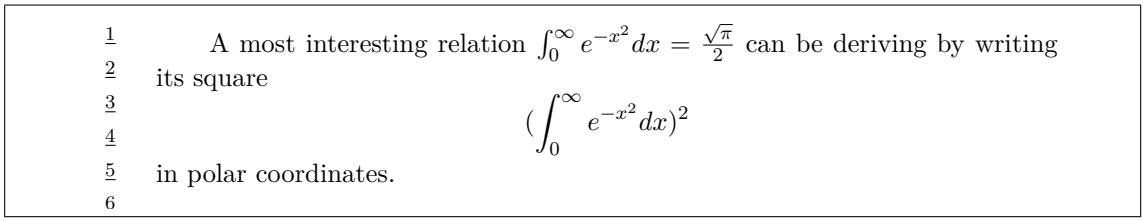

Figure 1: An example of a text requiring copyediting

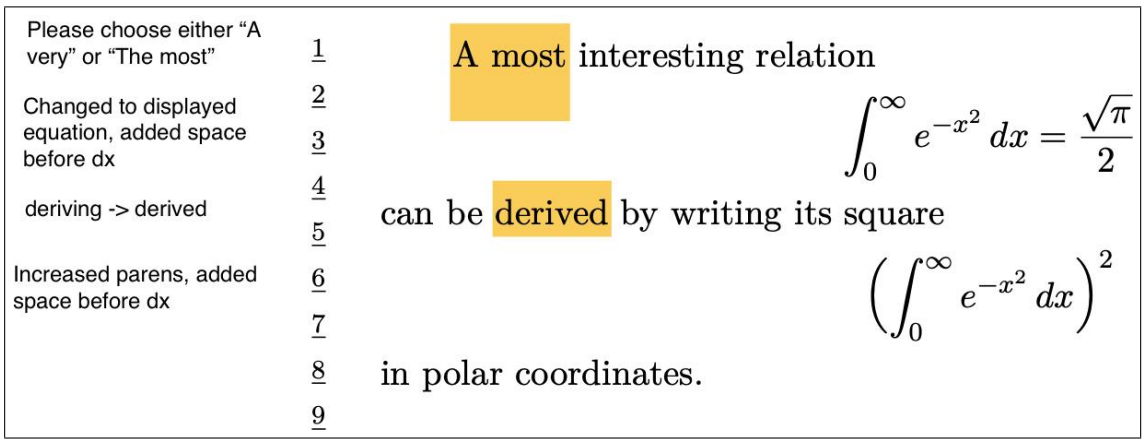

Figure 2: Communication with the author using PDF annotations

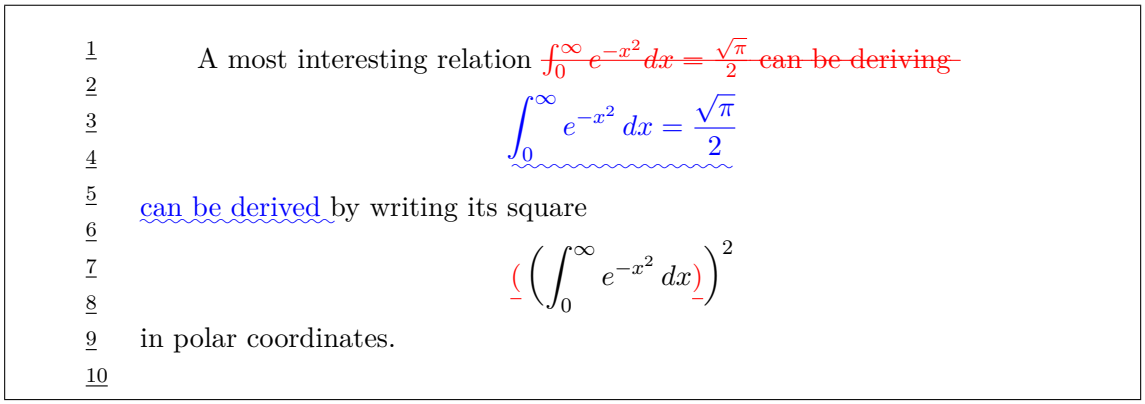

Figure 3: Example of *latexdiff* use

```
    A most interesting relation                % Either A very or The most
\[                                             % Displayed, added space
  \int_0^{\infty} e^{-x^2}\,dx =               % before dx
  \frac{\sqrt{\pi}}{2}
\]
can be derived by writing its square           % deriving -> derived
\[                                             % Increased parens,
 \left(\int_0^{\infty} e^{-x^2}\,dx\right)^2   % added space
\]                                             % before dx
in polar coordinates.
```

Figure 4: Adding comments to the source code

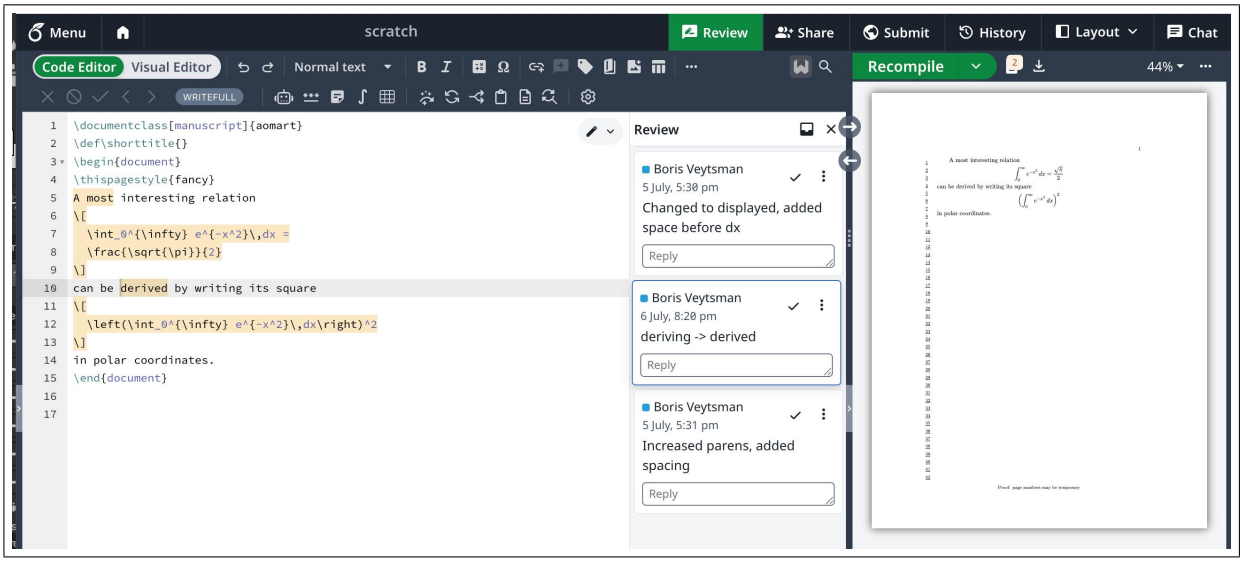

Figure 5: Overleaf environment with comments

The second command typesets `{`⟨*contents*⟩`}` in the color corresponding to `[`⟨*layer*⟩`]` if the global option `manuscript` is chosen for the document class *and* the highlighting layer `[`⟨*layer*⟩`]` is selected for display. If either of these conditions (or both) is not satisfied, `{`⟨*contents*⟩`}` is typeset without highlighting.

Currently, comments and highlighting layers are independent. They are selected for display with the commands `\ECSelect [`⟨*color*⟩`] {`⟨*layer*⟩`}` (for marginal comments) and `\HSelect [`⟨*color*⟩`] {`⟨*layer*⟩`}` (for highlighting). If `[`⟨*color*⟩`]` is not indicated, defaults are used. If these commands are not issued, the layer is not selected for display.

In Figures 6 and 7, we show the code for our copyediting example and the result. The commands seem to be reasonably easy to learn for a copyeditor with TEXnical skills, and the result seems to be intuitive enough for a novice.

4 Implementation notes

The `\EditorialComment` and `\Highlight` macros internally check whether the `manuscript` option is selected for the document class *and* whether the macros `\EditorialComment@layer` and `\Highlight@layer` are defined for the given `[`⟨*layer*⟩`]`. Unless both conditions are true, nothing is typeset or highlighted.

The commands `\ECSelect` and `\HSelect` define these macros for the given layer, storing in them the color to typeset marginal notes or highlight text (Figure 8). Note that for highlighting we suppress the coloring of hyperlinks done by *hyperref* commands.

Marginal notes present an additional problem. The LATEX kernel's `marginpar` (and its re-implementation in the *marginnote* package [3]) change page breaks. We need an implementation that does not affect breaks. The present solution is to wrap the contents into a zero-width and zero-height LATEX `picture`. To calculate the x offset of the picture we need to know the absolute coordinate of the comment anchor in the page. We use *zref-savepos* package from the bundle [4] to store the coordinate and retrieve it on the next `latex` run.

While the `marginpar` and `marginnote` implementations prevent overlapping, our typesetting of marginal notes in zero size boxes does not provide TEX with the information to do this. It also does not prevent notes falling below the page bottom. It would be possible to cure this by storing the heights of the notes and rearranging their positioning. Still, it is simpler to provide the copyeditor with a handle to deal with the problem manually. Thus the second optional argument of `\EditorialComment` macro, the vertical shift of the comment. If it is given, the note is moved up or down (for negative shift). The code is shown in Figure 9.

After this work was presented at TUG 2025, I received several interesting comments that I would like to share.

Frank Mittelbach noted that while zero boxes normally do not change breaks, it is possible to imagine exotic cases when they do. Thus it makes sense to produce them even in the final mode, just leaving them empty in this case. Ulrike Fischer suggested using PDF annotations instead of typesetting boxes for comments. Finally, Reinhard Kotucha showed me how to set marginal comments with a single `latex` run using the `\vadjust` command.

```
\ECSelect[red]{question}
\ECSelect[blue]{correction}
\HSelect[red]{question}
\HSelect[blue]{correction}
...
\Highlight[question]{A most}%
\EditorialComment[question]{Please choose either ``A very'' or ``The
  most''}[.5ex]
interesting relation
\[\EditorialComment[correction]{Converted to display and added
  spacing before $dx$}[-.5ex]
  \int_0^{\infty} e^{-x^2}\,dx =
  \frac{\sqrt{\pi}}{2}
\]
can be \Highlight[correction]{derived}%
\EditorialComment[correction]{deriving $\to$  derived}
by writing its square
\[\EditorialComment[correction]{Increased parens and added spacing
    before $dx$}
  \left(\int_0^{\infty} e^{-x^2}\,dx\right)^2
\]
in polar coordinates.
```

Figure 6: Code for editorial comments

Please choose either "A very" or "The most"

Converted to display and added spacing before *dx*

deriving → derived

Increased parens and added spacing before *dx*

A most interesting relation

$$\int_0^{\infty} e^{-x^2}\,dx = \frac{\sqrt{\pi}}{2}$$

can be derived by writing its square

$$\left(\int_0^{\infty} e^{-x^2}\,dx\right)^2$$

in polar coordinates.

Figure 7: Editorial comments produced by the code in Figure 6

```
\newcommand\ECSelect[2][black]{\if@aom@manuscript@mode
    \expandafter\gdef\csname
    EditorialComment@#2\endcsname{#1}%
  \fi}
\AtBeginDocument{\ECSelect{default}}
\newcommand\HSelect[2][red]{\if@aom@manuscript@mode
  \expandafter
    \gdef\csname Highlight@#2\endcsname##1{\bgroup
      \Hy@colorlinksfalse
      \def\Hy@colorlink####1{\begingroup}%
      \def\Hy@endcolorlink{\endgroup}%
      \color{#1}##1\egroup}%
  \fi}
\AtBeginDocument{\HSelect{default}}
```

Figure 8: `\ECselect` and `\HSelect` macros from [8]

```
\NewDocumentCommand\EditorialComment{O{default} +m O{0pt}}{%
 \if@aom@manuscript@mode\bgroup
 \expandafter\ifx\csname EditorialComment@#1\endcsname\relax
 \else
  \stepcounter{aom@EC@counter}%
  \setlength\unitlength{1sp}%
  \zsavepos{aom@EC@position\arabic{aom@EC@counter}}%
  \begin{picture}(0,0)%
  \put(\the\dimexpr
  -\zposx{aom@EC@position\arabic{aom@EC@counter}}sp + 15pt,#3){%
  \parbox{80pt}{\raggedright
  \normalfont\scriptsize\raggedright\expandafter\color
    \expandafter{\csname EditorialComment@#1\endcsname}#2}}%
   \end{picture}%
 \fi
 \egroup\fi}
```

Figure 9: `\EditorialComment` code

5 Conclusions and plans for the future

Currently, the code is a part of the *aomart* package [8]. The previous version of the code that used `marginpar` can be found in the *resphilosophica* package [9].

I plan to modularize the code and create a package that can be used by other journal styles. Among other improvements, we can tag the comments (for example, using the *Aside* or *Annot* tags [5, § 8.2.5.9 and § 8.2.5.21]). This will allow blind or low-vision authors to independently access editorial comments — probably for the first time in history.

It might be interesting to integrate this system into visual editors like Overleaf or TEXFolio to provide a graphic interface to editorial comments.

Acknowledgments

The financial support of the journals *Res Philosophica* (Saint Louis University) and *Annals of Mathematics* (Princeton University) is gratefully acknowledged. I am also grateful to the staff of these journals for many important comments and suggestions.

This paper was presented at TUG 2025, and I am grateful to the organizers and STMDocs for their hospitality and the great time in Kerala.

References

[1] Y. An, M.A. Williams, M. Xiao. High prices and market power of academic publishing reduce article citations. *ProMarket*, Apr. 2024. `www.promarket.org/2024/04/24/`

[2] R. Bhattacharjee, S. Khurana. Commonly used pre-print servers for researchers, March 2020. `medium.com/@sukantkhurana/pre-print-servers-for-researchers-2c924de9a402`

[3] M. Kohm. The marginnote package — notes in the margin, even where `\marginpar` fails, 2023-09-07. `ctan.org/pkg/marginnote`

[4] H. Oberdiek. The zref package — a new reference scheme for LaTeX, version 2.36, 2025-06-08. `ctan.org/pkg/zref`

[5] PDF Association. *Well-Tagged PDF (WTPDF). Using Tagged PDF for Accessibility and Reuse in PDF 2.0*, Feb. 2024. `pdfa.org/wtpdf`

[6] A. Thompson, C. Jeffords. Retail margins in illegal drug markets. *Economic Analysis and Policy*, 62:187–191, 2019. `doi.org/10.1016/j.eap.2019.03.001`

[7] F. Tilmann. The latexdiff package — determine and mark up significant differences between LaTeX files, version 1.3.4, 2024-07-20. `ctan.org/pkg/latexdiff`

[8] B. Veytsman. The aomart package — typeset articles for *The Annals of Mathematics*, version 1.34, 2025. `ctan.org/pkg/aomart`

[9] B. Veytsman. The resphilosophica package — typeset articles for the journal *Res Philosophica*, version 1.38, 2025-07-02. `ctan.org/pkg/resphilosophica`

[10] Wikipedia contributors. List of proofreader's marks, July 2025. `en.wikipedia.org/w/index.php?title=List_of_proofreader%27s_marks&oldid=1275570473`

◇ Boris Veytsman
TEX Users Group
`borisv (at) lk dot net`
`https://borisv.lk.net`