Macros

A complement to \smash, \llap, and \rlap

Alexander R. Perlis

Abstract

In both plain TEX and LATEX, most local alignment issues are addressed using \smash, \phantom, \vphantom, \hphantom, \llap, and \rlap. (LATEX also provides \makebox. All these macros are reviewed in this article.) However, conspicuously missing is a horizontal version of \smash, which is necessary, for example, to eliminate the excessive whitespace surrounding the large operator in

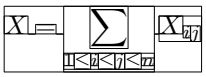
$$X = \sum_{1\le i\le j\le n} X_{ij}.$$

Another snag: whereas \smash and \phantom behave as expected in both horizontal mode and math mode, \llap, \rlap, and \makebox are not suited for use in math mode.

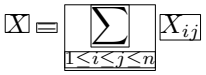
This article introduces the macro \clap (simultaneously a centered version of \llap/\rlap and a horizontal version of \smash), and the three macros \mathllap, \mathrlap, and \mathclap (versions of \llap, \rlap, and \clap designed for math mode).

1 Think in terms of boxes

To understand how alignment works, we should follow a TEX guru's mantra: *think in terms of boxes.*

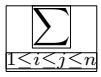


If we ignore some of the details, we are left with:



Evidently every item typeset by TEX has two components: the ink component, and the box component. The latter is often called the "boundary box" of the item, but this can be misleading, as the box may differ radically from the tightest one surrounding the ink. *TEX's alignment calculations are performed entirely in terms of box components.* In fact, TEX understands nothing about the ink component (such as its shape), and merely passes it along to the output file.

For individual glyphs, the box is part of the font's design and is encoded in the font's TFM file. That box may admit a margin or allow ink to spill out: $\sum$, $\int$. But when TEX constructs a big box from many small boxes, the new outer box will be the tightest one around all inner boxes:[1]



Finally, TEX composes a line by placing boxes side-by-side without overlap.

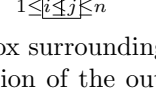
Thus, to eliminate the excessive whitespace surrounding the large operator, we must reduce the width of $1\le i\le j\le n$ yet preserve the ink. We might as well set the width to 0. In pictures, we want to change

$$1 \le i \le j \le n$$

to

$$1 \le i \not\le j \le n,$$

which is a box of width 0 (indicated by a line) with ink sticking out equally on either side. The result:



(What looks like a box surrounding "$i \le j$" is actually the bottom portion of the outer box that surrounds the box of the operator and the zero-width box of the subscript.)

In the next section, we'll review the macros, available in plain TEX and LATEX, that affect alignment by altering boxes. Then we'll introduce the new macro for achieving the effect discussed above.

2 Review of existing macros

The macro \smash boxes up its material but sets the height and depth of the box to 0. Thus it is the box itself that gets smashed, not the ink in the box. To smash *both the box and the ink*, i.e., to smash the box and eliminate the ink, use \hphantom in place of \smash. Thus \hphantom produces no ink: the horizontal phantom remaining after the smashing is an infinitely thin horizontal line segment just as wide as the original material.

align	**align**	
original	\smash	\hphantom

(By the way: it may be easier to understand these macros by their effect in the context of neighboring

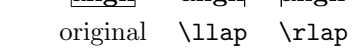
[1] In the example here, the extra whitespace at the bottom is due to a another box (more precisely: a kern) that lies below the subscript but is not indicated in our image. This mysterious box arises from the elaborate rules TEX follows for generating boxes in math mode. As explained in *The TEXbook*, Appendix G, Rule 13a: additional whitespace below subscripts (and above superscripts) of math operators is determined by \fontdimen13 of the math extension font. Rule 13a furthermore explains how TEX avoids an underfull box: it starts with the two boxes $\sum$ and $1\le i\le j\le n$, repackages them as $\sum_{1\le i\le j\le n}$ and $1\le i\le j\le n$, and puts them (along with some kerns) on top of each other.

material, which is shown in the two tables at the end of this article.) Now compare the above with:

original `\phantom` `\vphantom`

Evidently `\phantom` eliminates the ink without changing the box, while `\vphantom` eliminates the ink and smashes the box horizontally. (The first letter of `\hphantom` and `\vphantom` refers not to the direction of smashing but to the shape of the result.)

To smash the box horizontally, without affecting the ink, there are `\llap` and `\rlap`. The former aligns the smashed box at the right end of the ink (so that we end up with a "left overlap"), while the latter aligns the smashed box at the left end (resulting in a "right overlap").



original `\llap` `\rlap`

Finally, plain TEX and LATEX diverge as follows. Missing in plain TEX is a macro we'll define in section 4 and call `\clap`, which aligns the smashed box halfway between the left and right ends of the ink (we might call this a "centered overlap"). LATEX already provides it under a different name:
`\makebox[0pt][l]{...}` behaves like `\llap{...}`
`\makebox[0pt][r]{...}` behaves like `\rlap{...}`
`\makebox[0pt][c]{...}` behaves like `\clap{...}`

3 Concerning math mode

Whereas `\smash` and the three `\phantom` macros work correctly both in horizontal mode and in math mode, `\llap` and `\rlap` (and the LATEX-only `\makebox`) are suited only for horizontal mode. To use them in math mode, we must resort to monstrosities like

```
\rlap{$\mathsurround=0pt\scriptstyle{...}$}.
```

Here `\rlap` exited math mode, so we had to:

- use `$` to get back into math mode,
- use `\mathsurround` to eliminate whitespace introduced whenever we enter math mode, and
- reintroduce whatever math style was in effect before the `\rlap`.

With `\smash` and `\phantom` such shenanigans are unnecessary (indeed errors) because those macros use `\ifmmode` to test for the current mode and use `\mathpalette` to maintain the current math style. Thus where `\smash` and `\phantom` are flexible, `\llap` and `\rlap` are efficient.[2] Why the dichotomy? Perhaps Knuth can explain, but the matter is moot: plain TEX is essentially frozen, and future versions of LATEX are unlikely to deviate. All we can do is introduce new macros to fill in the gaps. They will be called `\mathllap`, `\mathrlap`, and `\mathclap`.[3] (For LATEX consistency we might also define `\mathmakebox` as a math mode analogue of `\makebox`, but don't show the code here.)

[2] The LATEX macro `\makebox` is neither flexible (in the sense under discussion) nor efficient, but has the benefit of being consistent with the rest of LATEX in its use of optional parameters.

4 The new macros

Use these macros with plain TEX or with LATEX.

```
% For comparison, the existing overlap macros:
% \def\llap#1{\hbox to 0pt{\hss#1}}
% \def\rlap#1{\hbox to 0pt{#1\hss}}

\def\clap#1{\hbox to 0pt{\hss#1\hss}}

\def\mathllap{\mathpalette\mathllapinternal}
\def\mathrlap{\mathpalette\mathrlapinternal}
\def\mathclap{\mathpalette\mathclapinternal}

\def\mathllapinternal#1#2{%
           \llap{$\mathsurround=0pt#1{#2}$}}
\def\mathrlapinternal#1#2{%
           \rlap{$\mathsurround=0pt#1{#2}$}}
\def\mathclapinternal#1#2{%
           \clap{$\mathsurround=0pt#1{#2}$}}
```

5 Applications

5.1 Large operators

Excessive whitespace may be eliminated as follows:

```
X = \sum_{\mathclap{1\le i\le j\le n}} X_{ij}
```

$$X = \sum_{\mathclap{1\le i\le j\le n}} X_{ij}.$$

5.2 Tabular alignments

Consider a complicated alignment, such as polynomial long division. Fiddling with `\ialign` yields:

```
\vcenter{\def\ministrut{\vrule height2pt
depth2pt width0pt}\offinterlineskip\ialign{%
$\mathstrut#$&&\hfil$\mathsurround=0pt#$\cr
&&&x+{}&1+\alpha\cr
\omit&\multispan{4}\rlap{\ministrut
                \vrule height0pt}\hrulefill\cr
x-\alpha\;&\vrule\;\;&x^2+{}&x+{}&2\cr
&&x^2-{}&\alpha x\phantom{{}+{}}&\cr
\omit&&\multispan{3}\ministrut\hrulefill\cr
```

[3] Concerning names, initially I used `\hsmash` in place of `\clap`, and had it test for math mode and maintain the current math style. Thus, whereas `\smash` is like `\hphantom`, my `\hsmash` was like `\vphantom`. Concerned that the names were confusing, I pondered: clapping one's hands together might be the horizontal analogue of smashing one's hands on, say, a desk. Thus the name `\clap` was born, but the definition still mimicked `\smash`. Only later did I realize the obvious connection (both in name and behavior) with `\llap`/`\rlap`. Remembering the separate need for math versions of those macros, I arrived at the design presented in this article: three overlap macros for horizontal mode, and separately three overlap macros for math mode.

```
&&&(1+\alpha)x+{}&2\cr
&&&(1+\alpha)x-{}&\alpha-\alpha^2\cr
\omit&&&\multispan{2}\ministrut\hrulefill\cr
&&&&2+\alpha+\alpha^2\cr
}}
```

$$\begin{array}{r|l} & x+\quad 1+\alpha \\ x-\alpha & \begin{array}{llr} x^2+ & x+ & 2 \\ x^2- & \alpha x & \end{array} \\ \hline & \begin{array}{lr} (1+\alpha)x+ & 2 \\ (1+\alpha)x- & \alpha-\alpha^2 \end{array} \\ \hline & 2+\alpha+\alpha^2 \end{array}$$

By inserting `\mathllap` thrice, `\mathrlap` twice, and `\quad` once (exercise: determine where), we reduce whitespace and allow α^2 to stick out:

$$\begin{array}{r|l} & x+1+\alpha \\ x-\alpha & \begin{array}{llr} x^2+ & x+ & 2 \\ x^2- & \alpha x & \end{array} \\ \hline & \begin{array}{lr} (1+\alpha)x+ & 2 \\ (1+\alpha)x-\alpha-\alpha^2 & \end{array} \\ \hline & 2+\alpha+\alpha^2 \end{array}$$

5.3 Commutative diagrams

Consider the alignment of arrows, objects, and arrow labels in commutative diagrams. Because many diagram packages exist, instead of showing the source for the following simple diagrams, the onus is on the reader to reproduce the following effect using the diagram package of choice.

By putting all the primes inside `\mathrlap`,

$$\begin{array}{ccc} C & \xrightarrow{\phi} & C' \\ \downarrow & & \downarrow \\ C'' & \xrightarrow{\phi''} & C''' \end{array} \quad \text{might become} \quad \begin{array}{ccc} C & \xrightarrow{\phi} & C' \\ \downarrow & & \downarrow \\ C'' & \xrightarrow{\phi''} & C''' \end{array} .$$

I wrote *might* because some minor additional fiddling may be necessary due to the difficulty that the box of an entry such as `C\mathrlap{''}` surrounds only the "C"; consequently, the neighboring arrow may land on top of the primes. Depending on your choice of diagram package, you might work around this problem by demanding some entries to have a wider margin, or by defining a new horizontal arrow that has some extra space at one end. (The diagrams above were produced using `\ialign`, so I simply preceded each arrow with `\mskip\thinmuskip`.)

6 Conclusion and acknowledgment

The new macros complement the existing ones by filling in the obvious gaps, as is evident from the tables below. My hope is that these macros (along with `\mathmakebox`—see section 3) will be incorporated into a future version of LaTeX, or at least become part of the `amsmath` package of $\mathcal{AMS}$-LaTeX.

MACRO	MODE	EFFECT	IN CONTEXT
none	either	**align**	**align**
`\smash`	either	**align**	**align**
`\phantom`	either		**align**
`\hphantom`	either		**align**
`\vphantom`	either	|	**align**
`\llap`	horiz.	**align**|	**align**
`\rlap`	horiz.	|**align**	**align**
`\clap`	horiz.	**al**|**gn**	**align**
`\mathllap`	math	**align**|	**align**
`\mathrlap`	math	|**align**	**align**
`\mathclap`	math	**al**|**gn**	**align**

Table 1: Effect of existing macros (first six) and new macros (last four).

COMBINATION	MODE	IN CONTEXT
`\llap{\smash{...}}`	horiz.	**align**
`\rlap{\smash{...}}`	horiz.	**align**
`\clap{\smash{...}}`	horiz.	**align**
`\mathllap{\smash{...}}`	math	**align**
`\mathrlap{\smash{...}}`	math	**align**
`\mathclap{\smash{...}}`	math	**align**

Table 2: The remaining effects are achieved using combinations.

For useful comments on an earlier version of this article, and in particular for telling me about `\makebox`, I thank Michael Downes.

◇ Alexander R. Perlis
Department of Mathematics
The University of Arizona
Tucson, AZ 85721 USA
aprl@math.arizona.edu