THE PLAIN TRUTH: DISPLAYLINES, IALIGN

Barbara Beeton
American Mathematical Society

This column continues the attempt, begun in the last issue, to illustrate the reasons for various changes to `plain.tex`. This issue's installment covers various changes made to macros controlling alignment.

Displayed Lines of Equations

The `\displaylines` macro lets you display any number of formulas in any way you want, without any alignment between formulas. Its original definition looked like this:

```
\def\displaylines#1{\displ@y
  \halign{\hbox to\displaywidth
    {$\hfil\displaystyle##\hfil$}\crcr
     #1\crcr}}
```

The first change never actually got into print in an errata list, being posted and rescinded almost immediately, but it may have been included in some distributions of version 1.1, or picked up via the Arpanet by unsuspecting users. This change consisted in putting braces around the aligned argument `##`.

These braces were added to be consistent with similar syntax in the definitions of `\eqalign`, `\eqalignno` and `\leqalignno`, but there is a significant difference—the `eqalign` macros were designed to produce only certain explicit structures, and `\displaylines` is provided to handle the nonstandard cases, where pieces may have to be moved around by hand. The following use of `\displaylines` is common:

```
\displaylines
   {\rlap{($\ast$)}\hfill a+b=c\hfill}
```

Without braces—`\displaystyle##`:

$$(*) \qquad a+b=c$$

With braces—`\displaystyle{##}`:

$$(*) \quad a+b=c$$

The `\hfill` instructions, intended to overpower occurrences of `\hfil` in the definition, lose their effect within braces.

Another bug in `\displaylines` was flushed out with the aid of this expression:

```
(m)\underbrace{x+y}(n)=0
```

This behaves nicely in a simple display, `$$...$$`:

$$(m)\underbrace{x+y}(n)=0$$

But with the "uncorrected" `plain`, the result of `\displaylines{...}` was unexpected:

$$(m)\underbrace{x+y}(n)=0$$

(Look closely at the baseline.) The explanation hinges on the complicated expansion of `\everycr` in the expansion of `\displ@y` (it appears in *The TEXbook* on page 362 and won't be repeated here); `\everycr` needs to be reset, and that was the nature of the fix to `plain`:

```
\def\@lign{\tabskip=0pt \everycr{}}
\def\displaylines#1{\displ@y
  \halign{\hbox to\displaywidth
    {$\@lign
      \hfil\displaystyle##\hfil$}\crcr
     #1\crcr}}
```

`\eqalignno` and `\leqalignno` were changed in a similar manner, by inserting `\@lign` before every instance of `\displaystyle` in their definitions. The `\tabskip=0pt` in `\@lign` locally resets `\tabskip=\centering` in `\*eqalignno` in case an alignment occurs in the argument.

Initialized Alignment

The `\ialign` macro provides an `\halign` for which `\tabskip` is initially zero. Its need for initialization similar to `\displaylines` had been discovered earlier. The original definition looked like this:

```
\def\ialign{\tabskip=0pt \halign}
```

The correction was:

```
\def\ialign
  {\everycr{}\tabskip=0pt \halign}
```

`\ialign` occurs internally in many `plain` macros, including ones dealing with alignment using tabs, accent placement, placement of arrows or braces above or below expressions, construction of "composite" characters from separate symbols (e.g. $\cong$ from $\sim$ and $=$), matrices, and displays aligned on equal signs.

To be continued...