Glisterings

Peter Wilson

> Whose waves do glister by the Queen's
> bright beams.
> Which makes them murmure as they passe
> away.
>
> *Poems and Fancies*, Margaret Cavendish

The aim of this column is to provide odd hints or small pieces of code that might help in solving a problem or two while hopefully not making things worse through any errors of mine.

Corrections, suggestions, and contributions will always be welcome.

In recent times there seems to have been a spate of questions on the `comp.text.tex` newsgroup about controlling paragraphs.

> for life's not a paragraph
> and death I think is no parenthesis.
>
> *since feeling is first*, e e cummings

1 Paragraphs regular

The typical paragraph in a LaTeX document is typeset like:

> The typical paragraph in a LaTeX document looks like this, with the first line indented otherwise lines are set left and right justified except for the last line which is set ragged right.

The indent at the start of a paragraph is given by the length `\parindent`. To temporarily achieve a paragraph with no indentation of the first line put `\noindent` at the start of the paragraph.

The other regular available paragraph shapes are ragged left (flush right), ragged right (flush left) and centered (each line is centered).

On occasions it is useful to be able to set a 'hanging paragraph' where more than one line is indented. TeX provides two commands for specifying such a paragraph (the hanging package [7] provides these in a more LaTeX-like manner).

`\hangindent`⟨*length*⟩ specifies a 'hanging indentation' and `\hangafter`⟨*num*⟩ specifies the number of hung lines. If ⟨*num*⟩ is positive hanging indentation is applied to lines ⟨*num*⟩+1, ⟨*num*⟩+2, ..., while if negative hanging indentation is applied to the first ⟨*num*⟩ lines of the paragraph. When ⟨*length*⟩ is positive the indentation applies to the lefthand end of the lines and when it is negative the righthand ends are indented.

```
\hangindent=3pc\hangafter=-2
```

> Following the above incantation the first two lines of this paragraph are indented at the left by the given amount. You can use these commands in a LaTeX document.

Note that you have to repeat the hanging specification for each hung paragraph.

LaTeX has an internal command `\@hangfrom` that it uses for several purposes, such as the internal code for section titles. An author-friendly version of this is:

```
\makeatletter % unless in a .cls or .sty file
\newcommand*{\hangfrom}[1]{%
  \setbox\@tempboxa\hbox{{#1}}%
  \hangindent \wd\@tempboxa
  \noindent\box\@tempboxa}
\makeatother  % unless in a .cls or .sty file
```

Using `\hangfrom{`⟨*text*⟩`}` at the start of a paragraph produces a paragraph where the second and further lines are indented with respect to the first by the width of ⟨*text*⟩. For instance, the code below produces the result following.

```
\hangfrom{$\Longrightarrow$\space\space}Here
we get a paragraph that is hung in
relation to its first element, which can
sometimes be useful.
```

> ⟹ Here we get a paragraph that is hung in relation to its first element, which can sometimes be useful.

Much more exotically you can get very odd looking paragraphs.

> For more symbolic paragraph shapes like this one, TeX provides the `\parshape` command. For LaTeX users this is provided in a more friendly fashion via Donald Arseneau's shapepar package [1]. Use this kind of paragraph very, very rarely, and only then if you really cannot avoid it as it is a typesetting curiosity. This one has been made using the shapepar package and the `\nutshape` specification.

The shapepar package provides several shape specifications, and there are programs which will automatically generate shape specifications.

Table 1: LaTeX's paragraph settings

	regular	raggedleft	raggedright	centered
\leftskip	\Zeroskip	\Flushglue	\Zeroskip	\Flushglue
\rightskip	\Zeroskip	\Zeroskip	\Flushglue	\Flushglue
\parfillskip	\Flushglue	\Zeroskip	\Zeroskip	\Zeroskip

> For precept must be upon precept; precept upon precept; line upon line; line upon line; here a little, and there a little.
>
> *Isiah*, ch. 28, v 10

2 Paragraphs particular

Besides \hangafter, \hangindent and \parshape, TeX provides 4 parameters[1] for controlling the shape of regular paragraphs. The length \parindent sets the initial indentation of the first line. \leftskip and \rightskip are inserted at the start and end of each line, and \parfillskip is put at the end of the last line; these last three macros are akin to LaTeX's concept of rubber lengths. By changing these you can obtain some particular kinds of paragraph shapes. Assuming that:

```
\Zeroskip = 0pt plus 0pt minus 0pt
\Flushglue = 0pt plus 1fil
```

then LaTeX's settings for its regular paragraph styles are given in table 1.

By adjusting the parameters you can arrange that the middle lines of a paragaph have a particular shape, while the first and/or last lines can have different forms. For instance:

```
\newcommand*{\justlastragged}{%
  \leftskip=0pt plus 1fil
  \rightskip=-\leftskip
  \parfillskip=\leftskip
  \parindent=0pt}
```

Following a \justlastragged declaration paragraph(s) will look like the example below.

> The shape of this paragraph is not too strange. It is flush left and right, except for the last line which is ragged left.

Another particular paragraph shape is one with the lines flush left and right, except for the last which is to be centered, as in:

> The lines in this paragraph, with the normal indentation of the first line, should be flush left and right except for the last line which should be centered.

[1] There are really 5 parameters, the fifth being \everypar which is inserted between the indent and the start of the first line. Change this only if you really know what you are doing.

This can be achieved by using the declaration \centerlastline before the paragraphs(s) in question:

```
\newcommand{\centerlastline}{%
  \leftskip=0pt plus 1fil
  \rightskip=0pt plus -1fil
  \parfillskip=0pt plus 2fil}
```

For no initial indentation put \noindent at the start of the paragraph's text.

```
\newcommand*{\raggedrightthenleft}{%
  \leftskip=0pt plus 1fill
  \rightskip=0pt plus 1fil
  \parfillskip=0pt
  \everypar{\hskip 0pt plus -1fill}%
  \parindent=0pt}
```

After a \raggedrightthenleft declaration the first line of any paragraph will be ragged right and all the other lines will be set ragged left. This looks odd to me.

> This is a strangely shaped paragraph. The paragraph's first line is ragged right and all the remaining lines are ragged left.

One way or another the paragraph layouts depend on the amount of what Knuth terms 'glue' (I think, though, that 'spring' would be a more evocative term) at the start and end of each line (the values of fil). For fuller explanations of glue see the *TeXbook* [5, ch. 12] or, in my view, more accessibly by Victor Eijkhout [3, ch. 8, 16–18].

In the definition of the last macro the \leftskip has a glue (spring) of strength 1fill, which is infinitely stronger than the \rightskip with strength 1fil, so normally a line will get pushed to the right. The last line has a \parfillskip of 0pt, which will not affect the end of the line. The first line of the paragraph has a springiness of -1fill from the \everypar and a springiness of 1fill from the \leftskip, which cancel each other out, leaving the spring of 1fil at the right of the line, and consequently the line gets pushed to the left.

Nikos Platis [6] wanted to ensure that some words at the end of a paragraph were right justified — if there was not enough space on the current line for them then they should be moved to the next

line while leaving the current line ragged right. That is either:

A short line. Text at right

or

A much longer line than the first one.
Text at right

Several solutions were given but the one initially proposed by Knuth in *The TEXbook* [5, p.106] and submitted by Dirk Schlimm turned out to be the most robust in Nikos' tests. In LaTeX terms:

```
\newcommand*{\atright}[1]{{%
  \unskip\nobreak\hfil\penalty50
  \hskip2em\hbox{}\nobreak\hfil#1
  \parfillskip=0pt\finalhyphendemerits=0\par}}
```

and putting `\atright{`⟨*text*⟩`}` at the end of a paragraph ensures that ⟨*text*⟩ is flush right.

Another often occuring request is how to ensure that the last line of a paragraph is 'not too short'.

Following the declaration `\nottooshort`, which I have defined as

```
\newdimen\parabout
\newdimen\about
\about=2em
\newcommand*{\nottooshort}{%
  \parabout=\hsize
  \advance\parabout -\about
  \leftskip=0pt plus 0pt minus 0pt
  \rightskip=\leftskip
  \parfillskip=\parabout minus \parabout
  \parindent=2em}
```

then the last lines of paragraphs will be at least approximately `\about` long.

The last line in this paragraph should not be too short, for a suitable definition of short. 1 2 3 4

The last line in this paragraph should not be too short, for a suitable definition of short. 1 2 3 4 5

With short paragraphs, like the examples, the overall effect might not look as good as you might expect. The situation improves with more lines.

Peter Flynn [4] answered Mark's question posed below by providing code for what he termed a 'spring margin', noting that very few systems provided it.

Hi, I'd like left- and right-justified text on the same line. e.g., some text here... ... and some text over here. I've tried `tabularx` and `TabularC`, but they are not precise enough to line up with the margins. Any suggestions? Thanks, Mark

The above was produced, with appropriate replacements for the ..., by

```
\spring{0.3}{0.6}%
       {Hi, I'd like left- ...}%
       {\dots and some text ...}
```

where `\spring` is a slight extension of Peter's code. The first two arguments are the fractions of the overall line allocated to the left and right texts; the sum of these must be less than 1. The second pair of arguments are the left and right texts.

```
\newcommand{\spring}[4]{%
 \par\noindent\hbox to\columnwidth{\vtop{%
 \hsize=#1\columnwidth\flushleft#3\par}\hss
 \vtop{\hsize=#2\columnwidth\flushright#4\par}}}
```

I have seen legal documents where each line must be filled at the right so that no extra words can be added later.

This last example was created based on the following code.

```
\let\origpar\par
\newcommand*{\parrule}{%
  \hrule height 2.2pt depth -1.8pt\relax}
\newcommand*{\lastlinerule}{%
  \unskip\nobreak\space
  \leaders\parrule\hskip\Flushglue
  \vadjust{}{\parfillskip=0pt\origpar}}
```

If you have many paragraphs of this kind then following a

```
\let\par\lastlinerule
```

all paragraphs will potentially have the last line filled with a rule. Be aware that LaTeX considers many things to be paragraphs so you could be in for some surprises. To revert back to the regular paragraphs specify:

```
\let\par\origpar
```

Alternatively, do something along these lines:

```
\begingroup
\let\par\lastlinerule
A ruled paragraph ...

Another one...

Even more...

\endgroup
```

The end of a paragraph is normally signalled by either a blank line or the `\par` command. For an isolated ruled paragraph, just end the text with `\lastlinerule` instead of `\par` or a blank line.

3 Paragraphs Russian

A while after I had completed this column I was going through old papers, trying to winnow those

that were no longer useful. Doing this I came across an old issue of *Baskerville — The Annals of the UK TEX Users' Group* which included an article about Russian-style paragraphs [2]. Apparently in the Russian typographic tradition the last line of a multiline paragraph must be either at least as long as the `\parindent` and have at least `\parindent` space at the end, or it must be flush left and flush right.

This requirement can't be fulfilled by any simple adjustment of the paragraph setting parameters.

The article ended with two solutions. The first, shown below, was by Peter Schmitt. The basic technique is to end each paragraph by (`glue + hbox + glue`), where the empty `hbox` spans `\parindent`, the (`glue+hbox`) ranges from `\parindent` to (`\hsize - \parindent`) and the (`hbox+glue`) covers (`\hsize-\parindent`) to `\hsize`, where `\hsize` is the line length. According to TEX rules, a linebreak may occur either before the `glue+hbox` or just after the `hbox`, in either case giving the paragraph a final blank line, which has to be backed up over.

```
\def\Srussianpar{\ifhmode \unskip
  \hskip-2\parindent minus -2\parindent
  \hskip\hsize minus\hsize
  \hbox{\hskip\parindent}%
  \hskip0pt \hbox{\strut}%
  \hskip-\parindent
  \hskip\hsize plus\parindent
  \vadjust{\nobreak\vskip-\baselineskip}%
  \parfillskip0pt
  \origpar
  \fi}
```

How this looks in practice is shown below, with `\parindent` set to 2em, together with the definition `\let\par\Srussianpar`.

The last line in this paragraph should conform to the Russian typesetting tradition. 1 2 3 4

The last line in this paragraph should conform to the Russian typesetting tradition. 1 2 3 4 5

The last line in this paragraph should conform to the Russian typesetting tradition. 1 2 3 4 5 6

The last line in this paragraph should conform to the Russian typesetting tradition. 1 2 3 4 5 6 7

The last line in this paragraph should conform to the Russian typesetting tradition. 1 2 3 4 5 6 7 8

The last line in this paragraph should conform to the Russian typesetting tradition. 1 2 3 4 5 6 7 8 9

The last line in this paragraph should conform to the Russian typesetting tradition. 1 2 3 4 5 6 7 8 9 0

The second solution was by Donald Arseneau:

```
\def\Arussianpar{\ifhmode \unskip
  \strut\vadjust{}\nobreak
  \discretionary{}%
    {\hbox{\hskip2\parindent
           \vrule depth 273sp
           width 0sp height \ht\strutbox}}%
    {\hbox{\hskip\parindent}}%
  \hskip-2\parindent minus 2\parindent
  \hskip\hsize minus\hsize
  \kern0pt\parfillskip0pt
  \origpar
  \ifdim\prevdepth=273sp
    \nobreak
    \vskip-2\baselineskip
    \hbox{\strut}%
  \fi
  \fi}
```

This works in approximately the same manner as Peter Schmitt's but it does not always produce an unwanted extra blank line. A rule with a unique depth small enough to be invisible on the page is inserted with the glue items. If the break is such that this is left on the last line, which will be otherwise empty, it can be detected from the `\prevdepth` value and the line backed up.

References

[1] Donald Arseneau. shapepar.sty, 2002. (Available on CTAN in `latex/macros/generic/shapepar`).

[2] David Carlisle and Peter Schmitt. Russian paragraph shapes. *Baskerville*, 6(1):13–15, February 1996.

[3] Victor Eijkhout. *TEX by Topic, A TEXnician's Reference*. Addison-Wesley, 1991. ISBN 0–201–56882–9. (Available at `http://www.eijkhout.net/tbt/`).

[4] Peter Flynn. Re: simultaneous justification in latex. Post to `comp.text.tex` newsgroup, 20 September 2006.

[5] Donald E. Knuth. *The TEXbook*. Addison Wesley, 1984.

[6] Nikos Platis. Justify at right margin or in next line. Post to `comp.text.tex` newsgroup, 21 August 2006.

[7] Peter Wilson. The hanging package, April 2004. (Available on CTAN in `latex/macros/contrib/hanging`).

◇ Peter Wilson
18912 8th Ave. SW
Normandy Park, WA 98166
USA
herries dot press (at) earthlink dot net