LaTeX-Paragraphs Floating around Figures

Thomas Kneser
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, FRG

August 20, 1990

Abstract

Frequently figures do not fill the full pagewidth. If the width of such figures is only half of the pagewidth or even less, textlines should be set beside the figures, or – from another point of view – figures should 'float' in paragraphs. This article presents the LaTeX style option `FLOATFIG` which can be used to set such Floating Figures as easily as LaTeX's standard figures.

1 The name of the game

The macros which make up the `FLOATFIG` style option are based on PLAIN-TeX macros developed by Thomas Reid (TUGBoat Vol. 8 # 3 page 315), who pointed out how to set figures **right justified** with paragraphs floating around them. For such objects Th. Reid has chosen the term 'Floating Figures'.

This choice might cause some confusion, when we adapt these macros for LaTeX, since Leslie Lamport uses the term 'float' for objects which are realized as TeX-`\inserts`. While L. Lamports floats are floating in 'main vertical list', floats introduced by Th. Reid are floating in paragraphs. For the following we adopt the latter meaning.

2 How to use it

The Floating Figures style option is fully compatible with LaTeX's standard figure facility:

1. Floating Figures and standard figures may be requested in any sequence,
2. Floating Figures can be captioned like standard figures,
3. captioned Floating Figures are inserted in the list of figures which may be printed by the standard `\listoffigures` command.

A Floating Figure may be requested as follows:

```
\documentstyle[floatfig]{article}
\begin{document}
\initfloatingfigs
        .
        .
\begin{floatingfigure}{5.6cm}
\vspace{6.cm}
\caption{Intermolecular potential K-Xe}% optional !
\end{floatingfigure}
        .
        .
\end{document}
```

where `5.6cm` specifies the width of the figure space. A Floating Figure may only be requested in vertical mode, that is between paragraphs.

A Floating Figure will be set as soon as possible after its request has be encountered by TeX. That means, it will be tested if there is enough vertical space on the current page; otherwise the figure moves to the next page.

Floating Figures are set **alternating**, that is on the right hand side on odd and on the left hand side on even numbered pages.

2.1 Restrictions

1. The `FLOATFIG` style option may not be combined with the `TWOCOLUMN` style option,
2. a Floating Figure will never appear in a paragraph which begins on top of a page.

3 About the internals

We have extended the macros designed by Th. Reid with regard to:

1. implantation into the LaTeX context,
2. alternating setting of Floating Figures as explained above,
3. generation of warning messages for ‘collisions’ of two Floating Figures.

3.1 Implantation into the LaTeX context

The PLAIN-TeX implementation by Th. Reid is based on a redefined `\output` routine:

```
\edef\oldoutput{\the\output}%
\output={\the\outputpretest\ifoutput\oldoutput\fi}
\outputpretest={\outputtrue}
```

If a Floating Figure is requested, the content of the `\outputpretest` token register is prepared to decide:

1. if there is enough vertical space to set the Floating Figure,
2. if setting of a Floating Figure is in progress or
3. if indeed the current page has to be sent to the DVI file.

In general, TEX has to deal with more than one paragraph until a Floating Figure will be completely processed. During this process the redefined `\output` routine is called at the begin of **every** paragraph; this is done indirectly by expanding the control sequence `\tryfig`. Therefore, the `\everypar` token list is prepared by the following command sequence:

```
\edef\oldeverypar{\the\everypar}
\everypar={\tryfig\oldeverypar}
```

Now `\tryfig` triggers the (modified) `\output` routine, which then does the decisions and actions mentioned above.

Adopting this concept when using the macros in the LATEX context, we are faced with the following problems:

1. At the time `FLOATFIG.STY` is read in, the `\output` routine is still undefined; its definition is retarded until `\begin{document}` will be expanded; so the redefinition of the `\output` routine has to be done after `\begin{document}` by the command `\initfloatingfigs` (see section 'Known Problems' below).
2. There are situations where LATEX decides to redefine the `\everypar` token list without saving of the former content; this occurs for instance when expanding a `\section` control sequence. We overcome this by redefining `\everypar` whenever the `\floatingfigure` environment is entered. Hence to avoid problems, a Floating Figure should be requested early enough before any sectioning control sequence (see also subsection 'Misleading collision warnings').
   Furthermore, the concurrent definitions of `\everypar` are the reason why Floating Figures cannot move across section boundaries.

3.2 Alternating figure setting

The problem to be solved is to decide if a certain figure has to be set left- or rightjustified. This decision has to made according to the value of the pagecount (left if even, right if odd). That is we are dealing with the wellknown problem to associate a certain part of input text with the number of the page on which it will be set finally.

As pointed out by D.E. Knuth in 'The TEXbook' this association is done not before `\output` routine time. But the problem is not so hard to solve, since in Th. Reid's version there is already a modified `\output` routine which decides if a certain figure will fit on the current page. As a by-product of this decision one easily gains the information 'odd' or 'even' for the pagecount of the current page. So our problem reduces to the following simple decision:

```
\ifodd\count0 %
   \hbox to \hsize{\hss\copy\figbox}%
   \global\oddpagestrue
 \else% leftsetting
   \hbox to \hsize{\copy\figbox\hss}%
   \global\oddpagesfalse
 \fi% \ifodd\count0
```

3.3 Collisions of Floating Figures

We define a collision as a situation where:

1. a Floating Figure is requested before the predecessor has been finished,
2. some sectioning is requested before a Floating Figure has been finished.

While the `FLOATFIG` style option cannot avoid such collisions, it will recognize them. For diagnostic purposes we have therefore defined the switch `\iffigprocessing` and another count register called `\ffigcount`. This count register is used to attach a sequence number to each Floating Figure, so they can be identified uniquely within collision warning messages. These sequence numbers are not to be confused with the figure count maintained by standard LaTeX.

4 Known Problems

4.1 Need of Initialization

The present version of the style option needs to be initialized by control sequence `\initfloatingfigs` as mentioned above.[1]

We hope a later version will initialize itself, when the first request to a Floating Figure is encountered. One problem with such an automatic initialization seems to be, that we are grouped down since we are inside a LaTeX environment. Our first attempt failed to make the respective settings `\global`.

4.2 Misleading collision warnings

As mentioned above, Floating Figures do not move across section boundaries. If a Floating Figure is requested near a section end, the figure will be truncated, if it does not fit in the current section.

[1] This is not needed with LaTeX 2_ε. /MD

If this occurs for instance with floating figure number 4, a collision will be reported when the request for floating figure number 5 is encountered; so the respective warning message may be substantially retarded. In fact, the message will tell us that there is a problem with figure 4, but there will be no further hint. So one has to analyze 'by hand' that the problem is **not** caused by collision with figure 5 but with a section heading; even worse: if there is no floating figure 5, one gets no warning message at all.

Furthermore, warning messages will be generated if a Floating Figure ends in paragraph n and the next one begins with paragraph $n+1$. These warning messages are to be ignored; they are due to some retardation caused by the `\everypar` mechanism.

5 Conclusions

Working on `FLOATFIG.STY` we had some unexpected problems which were caused by LaTeX's somewhat unsafe assignments to the `\everypar` token list. This is due to the fact that the use of this token list is fundamental to the algorithm designed by Th. Reid. On the other hand we did expect problems to couple the Floating Figures with LaTeX's figure caption adminstration, that is to achieve a single figure caption numbering for the standard figures and the Floating Figures and to get both types listed by the `\listoffigures` control sequence. But these problems could easily be solved by the following local definiton within the `FLOATFIG` environment:

```
\def\@captype{figure}
```

Obviously this is due to the fact that LaTeX's caption apparatus is thoroughly parameterized.

The `FLOATFIG.STY` file is stored in the `EARN/BITNET` listserver in Heidelberg; `VM/CMS` users should enter the commands:

```
TELL LISTSERV AT DHDURZ1 GET LATEXSTY INDEX
TELL LISTSERV AT DHDURZ1 GET LATEXSTY README
TELL LISTSERV AT DHDURZ1 GET LATEXSTY FILELIST
TELL LISTSERV AT DHDURZ1 GET FLOATFIG ZOOUUE LATEXSTY
```

to obtain information about how to obtain the style option file.

5.1 Note by Mats Dahlgren

Several features described in this document still apply to the LaTeX 2_ε `floatflt` package, whereas others do not. Please run LaTeX on `floatflt.drv` and read the resulting documentation before you use the package. (This note was added on October 20, 1994; changed on December 27, 1994.)