Abstract

This article describes the use and the implementation of the \texttt{multicols} environment. This environment allows switching between one and multicolumn format on the same page. Footnotes are handled correctly (for the most part), but will be placed at the bottom of the page and not under each column. \LaTeX{}’s float mechanism, however, is partly disabled in the current implementation. At the moment only page-wide floats (i.e., star-forms) can be used within the scope of the environment.

Preface to version 1.5

This new release contains two major changes: \texttt{multicols} will now support up to 10 columns and two more tuning possibilities have been added to the balancing routine. The balancing routine now checks the badness of the resulting columns and rejects solutions that are larger than a certain threshold. At the same time \texttt{multicols} has been upgraded to run under \LaTeX{}2ε.

I apologise for the state of the code documentation but the work on \LaTeX{}2ε kept me too busy to do a proper job. This will hopefully be corrected in the near future.

1 Introduction

Switching between two column and one column layout is possible in \LaTeX{}, but every use of \texttt{\twocolumn} or \texttt{\onecolumn} starts a new page. Moreover, the last page of two column output isn’t balanced and this often results in an empty, or nearly empty, right column. When I started to write macros for \texttt{doc.sty} (see “The doc–Option”, TUGboat volume 10 #2, pp. 245–273) I thought that it would be nice to place the index on the same page as the bibliography. And balancing the last page would not only look better, it also would save space; provided of course that it is also possible to start the next article on the same page. Rewriting the index environment was comparatively easy, but the next goal, designing an environment which takes care of footnotes, floats etc., was a harder task. It took me a whole weekend\footnote{I started with the algorithm given in the \TeX{}book on page 417. Without this help a weekend would not have been enough.} to get together the few lines of code below and there is still a good chance that I missed something after all.

Try it and, hopefully, enjoy it; and please direct bug reports and suggestions back to Mainz.

2 The User Interface

To use the environment one simply says

\begin{verbatim}
\begin{multicols}{\langle number\rangle}
\langle multicolumn text\rangle
\end{multicols}
\end{verbatim}

where \texttt{\langle number\rangle} is the required number of columns and \texttt{\langle multicolumn text\rangle} may contain arbitrary \LaTeX{} commands, except that floats and marginpars are not allowed in

\footnote{This file has version number v1.5l, last revised 1995/10/19.}

\footnote{Note: This package is released under terms which affect its use in commercial applications. Please see the details at the head of the source file.}
As its first action, the \texttt{multicols} environment measures the current page to determine whether there is enough room for some portion of multicolumn output. This is controlled by the \texttt{\newdimen} variable \texttt{\premulticols} which can be changed by the user with ordinary \LaTeX{} commands. If the space is less than \texttt{\premulticols}, a new page is started. Otherwise, a \texttt{\vskip} of \texttt{\multicolsep} is added.\footnote{This is dictated by lack of time. To implement floats one has to reimplement the whole \LaTeX{} output routine.}

When the end of the \texttt{multicols} environment is encountered, an analogous mechanism is employed, but now we test whether there is a space larger than \texttt{\postmulticols} available. Again we add \texttt{\multicolsep} or start a new page.

It is often convenient to spread some text over all columns, just before the multicolumn output, without any page break in between. To achieve this the \texttt{multicols} environment has an optional second argument which can be used for this purpose. For example, the text you are now reading was started with

\begin{verbatim}
\begin{multicols}{3}
[\section{The User Interface}] ...
\end{verbatim}

If such text is unusually long (or short) the value of \texttt{\premulticols} might need adjusting to prevent a bad page break. We therefore provide a third argument which can be used to overwrite the default value of \texttt{\premulticols} just for this occasion. So if you want to combine some longer single column text with a multicols environment you could write

\begin{verbatim}
\begin{multicols}{3}
\end{verbatim}

Separation of columns with vertical rules is achieved by setting the parameter \texttt{\columnseprule} to some positive value. In this article a value of \texttt{.4pt} was used. Since narrow columns tend to need adjustments in interline spacing we also provide a \texttt{(skip)} parameter called \texttt{\multicolbaselineskip} which is added to the \texttt{\baselineskip} parameter inside the \texttt{multicols} environment. Please use this parameter with care or leave it alone; it is intended only for package file designers since even small changes might produce totally unexpected changes to your document.

\section{Balancing Columns}

Besides the previously mentioned parameters, some others are provided to influence the layout of the columns generated.

Paragraphing in \LaTeX{} is controlled by several parameters. One of the most important is called \texttt{\tolerance}: this controls the allowed 'looseness' (i.e. the amount of blank space between words). Its default value is 200 (the \LaTeX{} manual for further information about this command).\footnote{Look at the next paragraph, it was set with the \texttt{sloppy} declaration.}

Its default value is \texttt{200} (the \LaTeX{} manual for further information about this command).\footnote{The reason behind this behavior is the asynchronous character of the \LaTeX{} page builder. However, this could be avoided by defining very complicated output routines which don't use \LaTeX{} primitives like \texttt{\insert} but do everything by hand. This is clearly beyond the scope of a weekend problem.}

Generation of multicolumn output can be divided into two parts. In the first part we are collecting material for a page, shipping it out, collecting material for the next page, and so on. As a second step, balancing will be done when the end of the \texttt{multicols} environment is reached. In the first step \LaTeX{} might consider more material whilst finding the final columns than it actually uses when shipping out the page. This might cause a problem if a footnote is encountered in the part of the input, considered, but not used, on the current page. In this case the footnote might show up on the current page, while the footnotemark corresponding to this footnote might be set on the next one.\footnote{This message will be generated even if there are no footnotes in this part of the text.}

Therefore the \texttt{multicols} environment gives a warning message\footnote{This message will be generated even if there are no footnotes in this part of the text.} whenever it is unable to use all the material considered so far.

If you don’t use footnotes too often the chances of something actually going wrong are very slim, but if this happens you can help \LaTeX{} by using a \texttt{\pagebreak} command in
the final document. Another way to influence the behavior of \TeX in this respect is given by the counter variable \texttt{collectmore}. If you use the \texttt{setcounter} declaration to set this counter to \texttt{(number)}, \TeX will consider \texttt{(number)} more (or less) lines before making its final decision. So a value of \texttt{-1} may solve all your problems at the cost of slightly less optimal columns.

In the second step (balancing columns) we have other bells and whistles. First of all you can say \texttt{raggedcolumns} if you don’t want the bottom lines to be aligned. The default is \texttt{flushcolumns}, so \TeX will normally try to make both the top and bottom baselines of all columns align.

Additionally you can set another counter, the \texttt{unbalance} counter, to some positive \texttt{(number)}. This will make all but the right-most column \texttt{(number)} of lines longer than they would normally have been. \texttt{Lines} in this context refer to normal text lines (i.e. one \texttt{baselineskip} apart); thus, if your columns contain displays, for example, you may need a higher \texttt{(number)} to shift something from one column into another.

Unlike \texttt{collectmore}, the \texttt{unbalance} counter is reset to zero at the end of the environment so it only applies to one \texttt{multicols} environment.

The two methods may be combined but I suggest using these features only when fine tuning important publications.

Two more general tuning possibilities were added with version 1.5. \TeX allows to measure the badness of a column in terms of an integer value, where 0 means optimal and any higher value means a certain amount of extra white space. 10000 is considered to be infinitely bad (\TeX does not distinguish any further). In addition the special value 100000 means overfull (i.e., the column contains more text than could possibly fit into it).

The new release now measures every generated column and ignores solutions where at least one column has a badness being larger than the value of the counter \texttt{columnbadness}. The default value for this counter is 10000, thus \TeX will accept all solutions except those being overfull. By setting the counter to a smaller value you can force the algorithm to search for solutions that do not have columns with a lot of white space.

However, if the setting is too low, the algorithm may not find any acceptable solution at all and will then finally choose the extreme solution of placing all text into the first column.

Often, when columns are balanced, it is impossible to find a solution that distributes the text evenly over all columns. If that is the case the last column usually has less text than the others. In the earlier releases this text was stretched to produce a column with the same height as all others, sometimes resulting in really ugly looking columns.

In the new release this stretching is only done if the badness of the final column is not larger than the value of the counter \texttt{finalcolumnbadness}. The default setting is 9999, thus preventing the stretching for all columns that \TeX would consider infinitely bad. In that case the final column is allowed to run short which gives a much better result.

### 2.2 Floats inside a \texttt{multicols} environment

Within the \texttt{multicols} environment the usual star float commands are available but their function is somewhat different as in the two-column mode of standard \LaTeX. Stared floats, e.g., \texttt{figure*}, denote page wide floats that are handled in a similar fashion as normal floats outside the \texttt{multicols} environment. However, they will never show up on the page where they are encountered. In other words, one can influence their placement by specifying a combination of \texttt{t, b, and/or p} in their optional argument, but \texttt{h} doesn’t work because the first possible place is the top of the next page. One should also note, that this means that their placement behavior is determined by the values of \texttt{topfraction}, \texttt{bottomfraction}, etc. rather then by \texttt{dbl}.

### 2.3 Warnings

Under certain circumstances the use of the \texttt{multicols} environment may result in some warnings from \TeX or \LaTeX. Here is a list of the important ones and the possible cause:

**Underfull \hbox (badness ...)**

As the columns are often very narrow \TeX wasn’t able to find a good way to break the paragraph. Underfull denotes a loose line but as long the badness values is below 10000 the result is probably acceptable.

**Underfull \vbox ... while \output is active**

If a column contains a character with an unusual depth, for example a ‘(‘, in the bottom line then this message may show up. It usually has no significance as long as the value is not more than a few points.

**LaTeX Warning: I moved some lines to the next page**

As mentioned above, \texttt{multicols} sometimes screws up the footnote numbering. As a precaution, whenever there is a footnote on a page that where \texttt{multicols} had to leave a remainder for the following page this warning appears. Check the footnote numbering on this page. If it turns out that it is wrong you have to manually break the page using \texttt{\newpage} or \texttt{\pagebreak[..]}.
2.4 Tracing the output

To understand the reasoning behind the decisions \TeX makes when processing a \multicols environment, a tracing mechanism is provided. If you set the counter `\multicols` to a positive \langle number \rangle you then will get some tracing information on the terminal and in the transcript file:

\langle number \rangle = 1. \TeX will now tell you, whenever it enters or leaves a \multicols environment, the number of columns it is working on and its decision about starting a new page before or after the environment.

\langle number \rangle = 2. In this case you also get information from the balancing routine: the heights tried for the left and right-most columns, information about shrinking if the \raggedcolumns declaration is in force and the value of the `unbalance` counter if positive.

\langle number \rangle = 3. Setting \langle number \rangle to this value will additionally trace the mark handling algorithm. It will show what marks are found, what marks are considered, etc. To fully understand this information you will probably have to read carefully trough the implementation.

\langle number \rangle > 4. Setting \langle number \rangle to such a high value will additionally place an \hrule into your output, separating the part of text which had already been considered on the previous page from the rest. Clearly this setting should not be used for the final output. It will also activate even more debugging code for mark handling.

3 Prefaces to older versions

3.1 Preface to version 1.4

Beside fixing some bugs as mentioned in the \multicol.bug file this new release enhances the \multicols environment by allowing for balancing in arbitrary contexts. It is now, for example, possible to balance text within a \minipage as shown in 2 where a \multicols environment within a \quote environment was used. It is now even possible to nest \multicols environments.

The only restriction to such inner \multicols environments (nested, or within \TeX’s internal vertical mode) is that such variants will produce a box with the balanced material in it, so that they can not be broken across pages or columns.

Additionally I rewrote the algorithm for balancing so that it will now produce slightly better results.

I updated the source documentation but like to apologize in advance for some ‘left over’ parts that slipped through the revision.

A note to people who like to improve the balancing algorithm of \multicols: The balancing routine in now placed into a single macro which is called \balance@columns. This means that one can easily try different balancing routines by rewriting this macro. The interface for it is explained in table 1. There are several improvements possible, one can think of integrating the \badness function of \TeX3, define a faster algorithm for finding the right column height, etc. If somebody thinks he/she has an enhancement I would be pleased to learn about it. But please obey the copyright notice and don’t change \multicol.dtx directly!

3.2 Preface to version 1.2

After the article about the \multicols environment was published in \TUGboat 10#3, I got numerous requests for these macros. However, I also got a changed version of my style file, together with a letter asking me if I would include the changes to get better paragraphing results in the case of narrow lines. The main differences to my original style option were additional parameters (like \multicol@adjdemerits to be used for \adjdemerits, etc.) which would influence the line breaking algorithm.

But actually resetting such parameters to zero or even worse to a negative value won’t give better line breaks inside a \multicols environment. \TeX’s line breaking algorithm will only look at those possible line breaks which can be reached without a badness higher than the current value of \pretolerance (or \tolerance in the first pass). If this isn’t possible, then, as a last resort, \TeX will produce overfull boxes. All those (and only those) possible break points will be considered and finally the sequence which results in the fewest demerits will be chosen. This
The macro \balance@columns that contains the code for balancing gathered material is a macro without parameters. It assumes that the material for balancing is stored in the box \mult@box which is a \vbox. It also “knows” about all parameters set up by the \multicols environment, like \col@number, etc. It can also assume that \@col@room is the still available space on the current page.

When it finishes it must return the individual columns in boxes suitable for further processing with \page@sofar. This means that the left column should be stored in box register \mult@gfirstbox, the next in register \mult@firstbox + 2, ..., only the last one as an exception in register \mult@grightbox. Furthermore it has to set up two the macros \kept@firstmark and \kept@botmark to hold the values for the first and bottom mark as found in the individual columns. There are some helper functions defined in section ?? which may be used for this. Getting the marks right “by hand” is non-trivial and it may pay off to first take a look at the documentation and implementation of \balance@columns before trying anew.

Table 1: Interface description for \balance@columns

- \adjdemerits means that a value of $-1000$ for \adjdemerits instructs \TeX to prefer visibly incompatible lines instead of producing better line breaks.

However, with \TeX 3.0 it is possible to get decent line breaks even in small columns by setting \emergencystretch to an appropriate value. I implemented a version which is capable of running both in the old and the new \TeX (actually it will simply ignore the new feature if it is not available). The calculation of \emergencystretch is probably incorrect. I made a few tests but of course one has have much more experience with the new possibilities to achieve the maximum quality.

Version 1.1a had a nice ‘feature’: the penalty for using the forbidden floats was their ultimate removal from \TeXs \@freelist so that after a few \marginpars inside the \multicols environment floats where disabled forever. (Thanks to Chris Rowley for pointing this out.) I removed this misbehaviour and at the same time decided to allow at least floats spanning all columns, e.g., generated by the \figure* environment. You can see the new functionality in table 2 which was inserted at this very point. However single column floats are still forbidden and I don’t think I will have time to tackle this problem in the near future. As an advice for all who want to try: wait for \TeX 3.0. It has a few features which will make life much easier in multi-column surroundings. Nevertheless we are working here at the edge of \TeXs capabilities, really perfect solutions would need a different approach than it was done in \TeXs page builder.

The text below is nearly unchanged, I only added documentation at places where new code was added.
\texttt{\setemergencystretch}: This is a hook for people who like to play around. It is supposed to set the \emergencystretch \texttt{(dimen)} register provided in the new \TeX\ 3.0. The first argument is the number of columns and the second one is the current \hsize. At the moment the default definition is \texttt{4pt \times #1}, i.e. the \hsize isn’t used at all. But maybe there are better formulae.

\texttt{\set@floatcmds}: This is the hook for the experts who like to implement a full float mechanism for the \texttt{multicols} environment. The @ in the name should signal that this might not be easy.

Table 2: The new commands of multicol.sty version 1.2. Both commands might be removed if good solutions to these open problems are found. I hope that these commands will prevent that nearly identical style files derived from this one are floating around.