mparhack.sty

Tom Sgouros
tomfool@top.gso.uri.edu

Stefan Ulrich
stefanulrich@users.sourceforge.net

v1.5 2021-05-02

CTAN: https://ctan.org/pkg/mparhack

Github: https://github.com/u-fischer/mparhack

Abstract

This package implements a workaround for the LaTeX bug that marginpars sometimes show up on the wrong margin.

Contents

1 Introduction

A persistent problem with the `\marginpar` command is that the marginalia produced often show up on the wrong margin. This has been noted in the LaTeX bugs database.[1]

The problem occurs most likely when a marginpar appears near the top of a page. The problem is exacerbated when the page break penalties are adjusted so

[1] See https://www.latex-project.org/cgi-bin/ltxbugs2html: latex/2361, latex/2484, latex/2617.

that the page look is quite ragged. In cases like this, if there is a deal of white space on a page, the marginpars on the next page will be on the wrong side for the same length of the page. The reason for this is that the LaTeX output routine is not in complete synchronization with the TeX page breaker, so that sometimes material might still move to a later page *after* LaTeX's marginpar algorithm has determined a page number for the marginpar.

The fix suggested in the bugs database is to insert `\pagebreak` commands at the places where the page breaks naturally fall; then the marginpars will appear on the correct side. However, this is awkward and unpleasant work for a document that changes regularly, and seems to fly against the spirit of LaTeX.

A different approach was suggested by D. E. Knuth in his *Macros for Jill*[2]: make two `.dvi` files, one with all notes in the left margin, the other one with all notes in the right margin. Then print the odd pages from one file, and the even pages from the other file; merge them by hand.

However, Knuth also mentions the possibility to use an auxiliar file to save the page information for each marginpar and use this information to position it correctly in a second TeX run; and this is the strategy adopted by `mparhack.sty`. Each marginpar is assigned a unique number; the output routine writes the number of the last marginpar of each page/column to the `.aux` file, and this information is used in `\@addmarginpar` to position the current marginpar on the correct margin.

2 Usage

Just insert `\usepackage{mparhack}` into your document preamble. After the first run, you'll probably get the warning message '`Marginpars may have changed. Rerun to get them right`'. This is similar to LaTeX's warning '`Label(s) may have changed`'; it indicates that the `.aux` file doesn't contain the information needed to position the marginpars correctly, due to some changes in the document since the last run. The warning should go away after running LaTeX again (eventually several times if there's also a table of contents etc.).

`debug` If you think something is going wrong, the package also has a `debug` option; this will output a lot of tracing information and will add numbers to the marginpars in the `.dvi` file for reference.

3 Bugs/Restrictions

- The package works with the `twocolumn` document option, but not with the `multicol` package (you can't use marginpars inside the `multicols` environment).
- The package uses one command sequence per page, and the total number of command sequences available is limited on most TeX systems. Note however

[2] Originally published in TUGboat **8**, pp. 309–314, reprinted in his *Digital Typography* pp. 185–193.

that most current TEX implementations can allocate more than several thousand command names, and some can be configured to allocate more than the default during run-time, so this shouldn't be a problem for moderately sized documents.

- The package has to hook into three commands of the LATEX2e output routine (by redefining some command), so it requires at least LATEX version 1997/04/14 and might break with future LATEX versions that change these commands. We try to detect this by requiring a version newer than 1997/04/14 and using some heuristics to check if the redefinitions have worked. You should watch out for this warning message:

  ```
  LaTeX Warning: You have requested release `1997/04/14' of LaTeX,
                 but only release `...' is available.
  ```

  It might be that the package also works with older versions of LATEX, but this hasn't been tested. If one of the heuristics detects an incompatible change, it will raise an error:

  ```
  Package mparhack Error: Couldn't hook into command `xyz'.
  ```

  It might also happen that the heuristics don't detect a change properly. If the package simply seems to fail, this could be a reason. In both cases, please send a bug report to stefanulrich@users.sourceforge.net (if neccessary including the `.log` file of your LATEX run with the `debug` package option enabled). Suggestions/improvements are also welcome, of course.

4 Acknowledgements

Thanks to Donald Arseneau and Robin Fairbairns for providing detailed hints[3] on how to implement the page referencing mechanism using only one label per page (instead of one label each marginpar, as in a previous version).

5 Implementation

5.1 Overview

Here's the algorithm for recording the positioning information for marginpars in more detail: Every marginpar has a unique number n and writes to the `.aux` file a string like

`\def\mph@nr{`n`}`.

The commands shipping out the columns and pages write to the `.aux` file strings like:

`\mph@setcol{`x`:\thepage}{\mph@nr}`,

[3] in `<23AUG199920123097@erich.triumf.ca>` and `<7prb8s$dpr$1@pegasus.csx.cam.ac.uk>`, respectively.

with x = i for the first column and x = ii otherwise. Then when reading the `.aux` file at the begin of the LaTeX run, `\mph@nr` will be substitued by the number of the last marginpar (the value of the `\mph@nr` command read before that), and `\mph@setcol` will assign that number to the command sequence `\csname mph@last@`x`:\thepage\endcsname`; so this will contain the number of the last marginpar on column/page x`:\thepage`.

Later on, in the document itself, `\@addmarginpar` will be able to check the value of `\csname mph@last@`x`:\thepage\endcsname`, using the current values for x and `\thepage`. Then there are two cases:

(a) `\csname mph@last@`x`:\thepage\endcsname` is undefined, which means that there has been no `.aux` file previously, or that there are more pages/marginpars than in the previous run; in this case, `\@addmarginpar` will fall back to using the value of `\c@page` to determine the current page number (yielding the same result as without this package), or

(b) `\csname mph@last@`x`:\thepage\endcsname` is defined as an integer z, the number of the last marginpar on `\thepage`. Then $z \leq n$, the number of the current marginpar.[4] While $z < n$, the page counter will be incremented (inside a group) until `\csname mph@last@`x`:\thepage\endcsname` for that new page is $\geq n$, or until we end up in case (a). The incremented value for the page counter now represents the actual page number of the current marginpar, and it will be used instead of the original page counter in the check for even-/oddness.

5.2 Code

We need at least LaTeX version 1997/04/14 which introduced `\normalsfcodes` into `\@outputdblcol`.

```
1 ⟨*sty⟩
2 \NeedsTeXFormat{LaTeX2e}[1997/04/14]
3 \ProvidesPackage{mparhack}
4       [2021-05-02 v1.5  (T. Sgouros and S. Ulrich)]
```

5.2.1 Booleans, Options, etc.

Some boolean switches which we will be using: `\if@mph@firstcol@` indicates that the current column is the first column, `\if@mph@warning@` toggles the warning message about changed marginpars, and `\if@debug@` toggles printing (lots of) debugging messages.

```
5 \newif\if@mph@firstcol@
6 \newif\if@mph@warning@
7 \newif\if@debug@
```

`\mph@debug` This command will print the debugging messages if `@debug@` is `true`:

```
8 \newcommand*\mph@debug[2][]{%
9     \if@debug@
```

[4] z cannot be greater than n: this would mean that TeX had shifted some material (including a marginpar) from later pages to earlier pages, which AFAIK cannot happen.

```
        \typeout{DBG: #2}%
    \else
        \ifx#1\@empty\else
            \typeout{DBG (#1): #2}%
        \fi
    \fi
}
```

The `debug` package option sets `@debug@` to true and adds numbers to each marginpar by redefining `\@xmpar` and `\@ympar`. The numbers are put into boxes of dimension zero so that they don't alter the spacing.

```
\DeclareOption{debug}{%
    \newcommand\mph@info{%
        \@tempcnta=\mph@cnt
        \advance\@tempcnta1
        \raisebox{0pt}[0pt][0pt]{\makebox[0pt][r]{\tiny\the\@tempcnta}}%
    }%
    \long\def\@xmpar[#1]#2{%
        \@savemarbox\@marbox{\mph@info#1}%
        \@savemarbox\@currbox{\mph@info#2}%
        \@xympar
    }%
    \long\def\@ympar#1{%
        \@savemarbox\@marbox{\mph@info#1}%
        \global\setbox\@currbox\copy\@marbox
        \@xympar
    }%
    \global\@debug@true
}%
\ProcessOptions*
```

5.2.2 Commands

`\mph@nr` A command sequence containing the current marginpar number (which will be written to the `.aux` file).

```
\newcommand*\mph@nr{0}
```

`\mph@pg@new` A command sequence for saving the newly determined value of `\thepage` for later use.

```
\newcommand*\mph@pg@new{0}
\newcommand*\mph@lastpage{0}
```

`\mph@cnt` Emulate a counter with a command name to save one count register. Oh well ... ;-)

```
\newcommand*\mph@cnt{0}
```

`\mph@step@cnt` Use this instead of `\stepcounter` to increment the 'counter' passed as the argument.

```
\newcommand*\mph@step@cnt[1]{%
    \begingroup
    \@tempcnta#1
    \advance\@tempcnta\@ne
    \protected@xdef#1{\the\@tempcnta}%
    \endgroup
}

```

`\mph@setcol` The command sequence written to the `.aux` file that will save the column number/ page number of the last marginpar on the column/page.

```
\newcommand*\mph@setcol[2]{%
%  \@ifundefined{mph@last@#1}{%
%        \relax
%  }{%
%        \expandafter\xdef\csname ignore@warn#1\endcsname{x}%
%  }%
  \expandafter\xdef\csname mph@last@#1\endcsname{#2}%
}
```

`\mph@check` Similar to LaTeX's `\@testdef` for labels, this will be called at the end of the LaTeX run to check if the meaning of `#2` is still equal to `mph@last@#1` or if it has changed since the last run.

```
\newcommand*\mph@tempa{}
\newcommand*\mph@check[2]{%
    \protected@edef\mph@tempa{#2}%
    \expandafter\ifx\csname mph@last@#1\endcsname\mph@tempa
        \mph@debug{\csname mph@last@#1\endcsname\space == \mph@tempa}%
    \else
        \mph@debug{\csname mph@last@#1\endcsname\space != \mph@tempa!}%
        \global\@mph@warning@true
    \fi
}
\newcommand*\mph@do@warn{%
    \if@mph@warning@
        \PackageWarningNoLine{mparhack}{Marginpars may have
            changed.\MessageBreak Rerun to get them right%
        }%
    \fi
}
```

To avoid warnings about undefined commands when someone stops using this package, we `\providecommand` the `\mph@setcol` command that we're going to write to the `.aux` file.

```
\AtBeginDocument{%
    \protected@write\@auxout{}{%
        \string\providecommand\string\mph@setcol[2]{}%
    }%
```

Fix for latex/3775[5]: eso-pic uses `\@picture` inside the output routine, which causes problems with our redefinition of `\hb@xt@` in `\@outputdblcol`. So, as suggested by M. Høgholm in his reply to the above bug report, we patch `\@picture` to revert to the original definition of `\hb@xt@`.

```
\@ifpackageloaded{eso-pic}{%
    \let\mph@orig@picture\@picture
    \renewcommand\@picture{\let\hb@xt@\mph@orig@hb@xt@\mph@orig@picture}%
}{%
    \relax
}%
}
```

Add the needed checks to `\@enddocumenthook`. Again, use `\csname` ... `\endcsname` to avoid warnings about undefined commands.

```
\AtEndDocument{%
    \let\mph@setcol\mph@check
    \clearpage
    \immediate\write\@auxout{%
        \string\gdef\string\mph@lastpage{\the\c@page}^^J%
        \string\csname\space mph@do@warn\string\endcsname
    }%
}
```

`\mph@ifundef@or@smaller` A helper macro that checks if the command sequence in `#1` is undefined or if its value is smaller than `\mph@cnt` (it assumes that `#1` is never empty). If one of these conditions is true, the second argument is evaluated, else the third argument is evaluated (similar to LaTeX's `\@ifundefined`). This macro will be used to shorten the body of the next macro definition.

```
\newcommand*\mph@ifundef@or@smaller[1]{%
    \expandafter\ifx#1\relax
        \let\mph@tempa\@firstoftwo
    \else
        \ifnum#1<\mph@cnt\relax
            \let\mph@tempa\@firstoftwo
        \else
            \let\mph@tempa\@secondoftwo
        \fi
    \fi
    \mph@tempa
}

```

`\mph@pg@orig` The macro `\mph@pg@orig` is used to save the original value of `\c@page` which is used in case the `.aux` file information is not up-to date and we can't determine the correct value; this should give the same results as if `mparhack.sty` hadn't been used.

```
\newcommand*\mph@pg@orig{}
```

[5] https://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/3775

\mph@get@margin This macro does most of the work described in section 5.1.

```
\newcommand*\mph@get@margin{%
    \begingroup
    \edef\mph@pg@orig{\the\c@page}
    \loop
```

This loop will be controlled by `\@tempswa`; all changes to that and to `\thepage` will be local to the current group.

```
    \@tempswafalse
    \mph@debug{--- checking marginpar \mph@cnt}%
    \if@twocolumn
```

When we are in twocolumn mode, first check if the current marginpar number is greater than the last marginpar in the first column. If it isn't, we're in the first column. If it is, do the check for the last marginpar of the entire page. If this returns false, we're in the second column. Else, advance the page number and try again.

```
        \mph@debug{last on page \thepage:
                \csname mph@last@i:\thepage\endcsname(i)
                \csname mph@last@ii:\thepage\endcsname(ii),
        }%
        \mph@ifundef@or@smaller{%
            \csname mph@last@i:\thepage\endcsname
        }{%
            \mph@ifundef@or@smaller{%
                \csname mph@last@ii:\thepage\endcsname
            }{%
                \global\@mph@firstcol@true
                \@tempswatrue
                \advance\c@page by 1
                \mph@debug{\mph@cnt\space >
                    \csname mph@last@ii:\thepage\endcsname,
                    incrementing \thepage, set col to i
                }%
            }{%
                \global\@mph@firstcol@false
                \@tempswafalse
                \mph@debug{\mph@cnt\space <=
                    \csname mph@last@ii:\thepage\endcsname,
                    exiting loop
                }%
            }%
        }{%
            \@tempswafalse
            \global\@mph@firstcol@true
            \mph@debug{exiting loop}%
        }%
    \else
```

Not in twocolumn mode, we need only check for `mph@last@ii`:

```
        \mph@debug{last on page \thepage\space is
                \csname mph@last@ii:\thepage\endcsname
        }%
        \mph@ifundef@or@smaller{%
            \csname mph@last@ii:\thepage\endcsname
        }{%
            \@tempswatrue
            \advance\c@page by 1
            \mph@debug{\mph@cnt\space >
                \csname mph@last@ii:\thepage\endcsname,
                incrementing page number
            }%
        }{%
            \@tempswafalse
            \mph@debug{\mph@cnt\space <=
                \csname mph@last@ii:\thepage\endcsname,
                exiting loop
            }%
        }%
    \fi
```

Now we have either found the correct page, or reached the end of the document without finding the appropriate page (e. g., if the `.aux` file hasn't existed before, or if the document has grown a lot since the last run). In this case we reset `\c@page` and `\if@mph@firstcol@` to their original values and jump out of the loop.

```
    \ifnum\mph@lastpage>\c@page
    \else
        \@tempswafalse
        \mph@debug{\c@page >= \mph@lastpage!}%
        \c@page=\mph@pg@orig
        \if@firstcolumn
            \global\@mph@firstcol@true
        \else
            \global\@mph@firstcol@false
        \fi
        \mph@debug{using original value: \c@page for
            \string\c@page and exiting loop.
        }%
    \fi
    \if@tempswa
        \mph@debug{iterating ...}%
    \repeat
```

After having finished the loop, we save the current (eventually modified) value of `\c@page` to `\mph@pg@new`:

```
    \mph@debug{=== marginpar \mph@cnt\space is on page \thepage%
        \if@twocolumn, col \if@mph@firstcol@ 1 \else 2 \fi\fi
        }%
    \protected@xdef\mph@pg@new{\the\c@page}%
    \endgroup
```

```
}
```

5.2.3 Patching the output routine

We have to hook into 3 commands that are part of LaTeX's output routine: `\@addmarginpar`, `\@outputdblcol` and `\@outputpage`. For the latter, we can use the default LaTeX mechanism `\g@addto@macro`. For the others, this is more difficult, since the changes need to go somewhere 'in between'.

One could use hard redefinitions (plus `\CheckCommand*` to verify that the correct version of the command was being defined); but this soon gets too bothersome with new LaTeX2e versions introducing minor changes that don't affect the stuff which we are concerned, but that nevertheless would require multiplying the redefinitions for the various versions.

So instead, we locally redefine some commands to hook into the existing definitions. As a heuristic of detecting whether we change the appropriate commands, we check the values of some special counters `\mph@chk@xyz` that get incremented by some of the 'hooks'.

`\mph@error` Error message if something suspicious happened with the redefintions:

```
\newcommand\mph@error[1]{%
    \PackageError{mparhack}{%
        Couldn't hook into command `#1'
    }{%
        This means that a LaTeX version incompatible with
        mparhack.sty^^J%
        has been used. See also the section on `Bugs/Restrictions'^^J%
        in mparhack.dvi. Please send an email about this bug to^^J%
        <stefanulrich@users.sourceforge.net>, along with the file `\jobname.log'.
    }%
}
```

`\@addmarginpar` This command adds the marginpar box to the current page. We change the `\c@page` counter and the `\if@firstcolumn` switch to the values that have been set correctly by the `\mph@get@margin` command. There's no further error checking done here.

```
\newcommand\mph@orig@c@page{}
\newcommand\mph@orig@addmarginpar{}
\let\mph@orig@addmarginpar\@addmarginpar

\renewcommand\@addmarginpar{%
    \mph@step@cnt\mph@cnt
    \protected@write\@auxout{}%
         {\string\def\string\mph@nr{\mph@cnt}}%
    \mph@get@margin
    \def\mph@orig@c@page{\c@page}
    \c@page=\mph@pg@new
    \if@mph@firstcol@
```

```
        \@firstcolumntrue
    \else
        \@firstcolumnfalse
    \fi
    \mph@orig@addmarginpar
    \c@page=\mph@orig@c@page
}

```

`\@outputpage` Add the `\mph@outputpage@hook` hook to `\@outputpage`. We need to decrement `\c@page`, since `\@outputpage` will have incremented it after dumping the page contents. The group in `\mph@outputpage@hook` is to keep this change local.

```
\newcommand*\mph@outputpage@hook{%
    \bgroup
    \advance\c@page\m@ne
    \immediate\write\@auxout{%
        \string\mph@setcol{ii:\thepage}{\string\mph@nr}%
    }%
    \egroup
}
\g@addto@macro{\@outputpage}{\mph@outputpage@hook}

```

`\@outputdblcol` In case an entire twocolumn page is finished, this command calls `\hb@xt@` with the boxes for the two columns inside, and each one needs to set a marker with the current column number. So we hook into `\hb@xt@` and check if its first argument equals `\columnwidth` (`\hb@xt@` is also called for the entire page, but in this case the first argument is `\textwidth`).

The current column number is represented by `\mph@chk@dcl`, and at the end we check if this number equals 0 (in which case the page hadn't been fininshed yet, and we didn't need to do anything) or 2 (the 'interesting' case); else we signal an error.

```
\newcommand\mph@orig@hb@xt@{}
\newcommand\mph@orig@outputdblcol{}
\newcommand\mph@chk@dcl{0}%
\let\mph@orig@hb@xt@\hb@xt@
\let\mph@orig@outputdblcol\@outputdblcol

\renewcommand\@outputdblcol{%
    \def\mph@chk@dcl{0}%
    \def\hb@xt@##1##2{%
        \ifx##1\columnwidth
            \mph@step@cnt\mph@chk@dcl
            \protected@edef\mph@tempa{%
                \noexpand\write\noexpand\@auxout{%
                    \noexpand\string\noexpand\mph@setcol{%
                        \romannumeral\mph@chk@dcl:\noexpand\thepage%
                    }{%
                        \noexpand\string\noexpand\mph@nr%
```

```
                    }%
                }%
            }%
            \mph@orig@hb@xt@##1{##2\mph@tempa}%
        \else
            \mph@orig@hb@xt@##1{##2}%
        \fi
    }%
    \mph@orig@outputdblcol
    \def\hb@xt@{\mph@orig@hb@xt@}%
    \ifnum\mph@chk@dcl=0\relax
    \else
        \ifnum\mph@chk@dcl=2\relax
        \else
            \mph@error{\string\@outputdblcol}%
        \fi
    \fi
}
```

`\mph@orig@picture`

```
\newcommand\mph@orig@picture{}
```

⟨/sty⟩

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Change History

v1.0

General: Created 1

v1.1

General: Rewrite to use only one