

# The `marginfit` package

Maurice Leclaire  
`leclaire@in.tum.de`

v1.0 – 2017/07/07

## Abstract

The `marginfit` package fixes various bugs with the margin paragraph implementation of  $\text{\LaTeX}$ . Those bugs include margin notes that are attached to the wrong side as well as those that stick out of the bottom of the page. This package provides a drop-in replacement solution.

## Contents

<b>1</b>	<b>Motivation</b>	<b>1</b>
<b>2</b>	<b>Usage</b>	<b>4</b>
<b>3</b>	<b>Implementation</b>	<b>4</b>

## 1 Motivation

The margin paragraph mechanism of  $\text{\LaTeX}$  has two main problems:

1. Near after a page break, it can happen that a margin note is placed at the wrong side. See Margin Note 2 in Figure 1.
2. Margin notes can stick out the bottom of the page. See Margin Note 5 in Figure 1. This happens if margin notes are too large for the remaining space on the page or if they are shifted down, because they do not fit at the callout position. However,  $\text{\LaTeX}$  does not shift margin notes upwards, even if there would be enough space.

Those problems are caused by calling the output routine for each margin paragraph and placing it separately, only considering previously placed margin notes. This can lead to wrong assumptions concerning the actual page break, which then results in Problem 1. Further, this mechanism prevents look ahead, which is the reason for Problem 2. Margin notes can be shifted down based on the previous margin notes, but there is no possibility to shift them up, if that would be necessary for later margin notes.

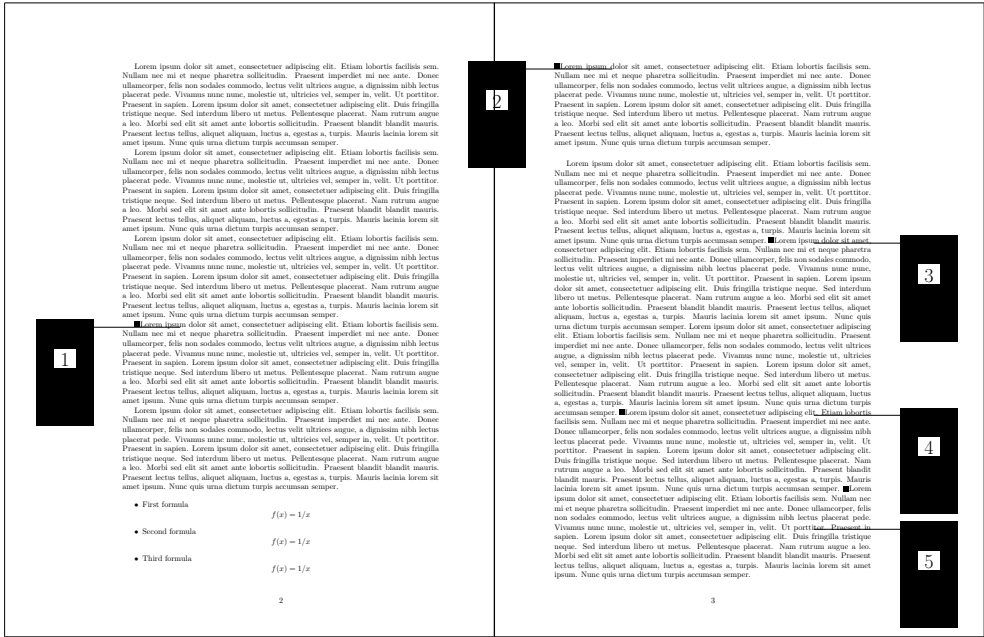


Figure 1: Plain L<sup>A</sup>T<sub>E</sub>X

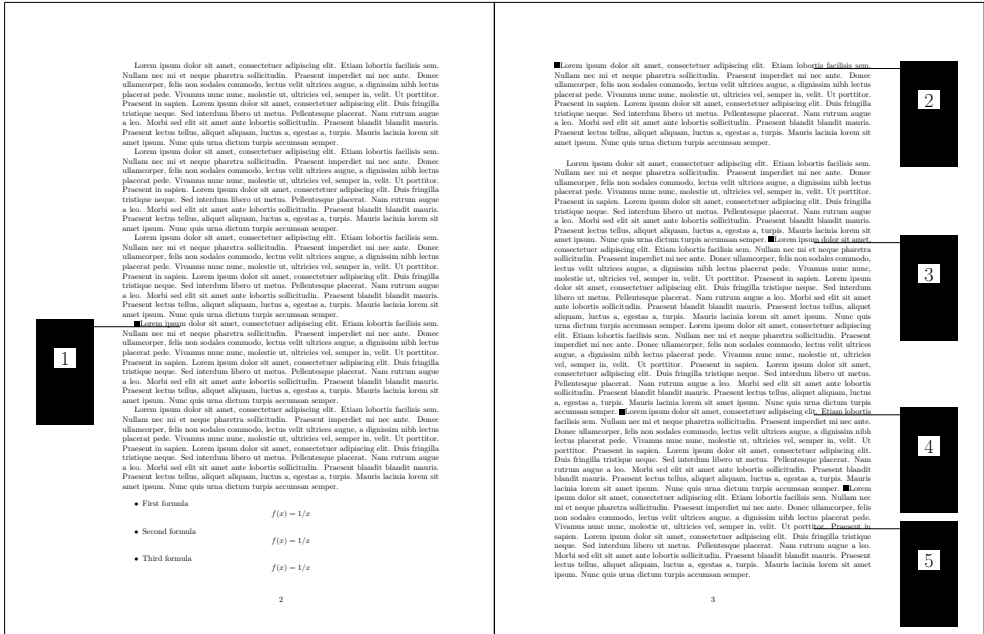


Figure 2: mparhack package

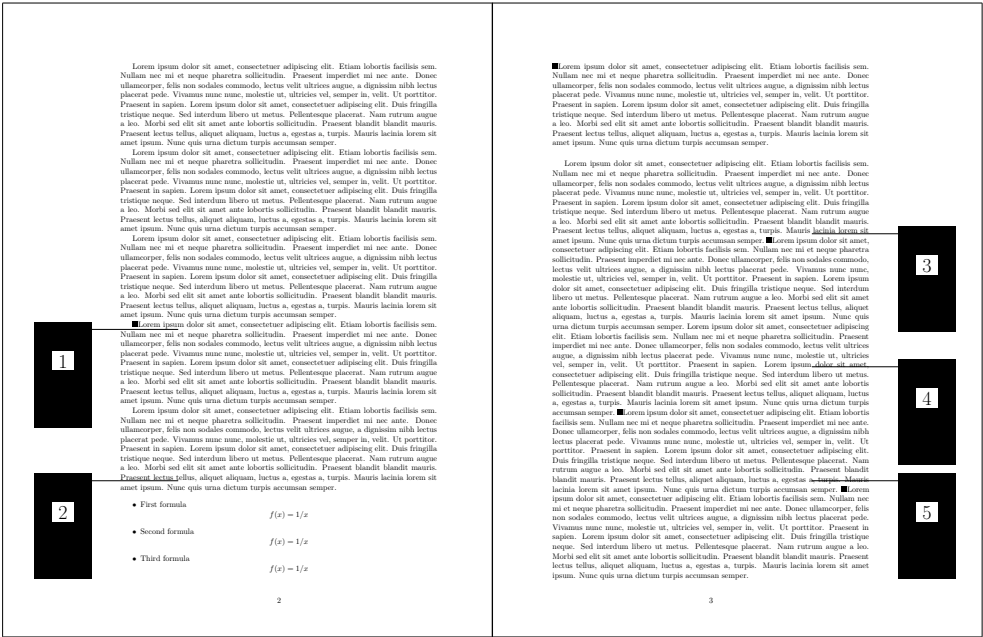


Figure 3: marginfix package

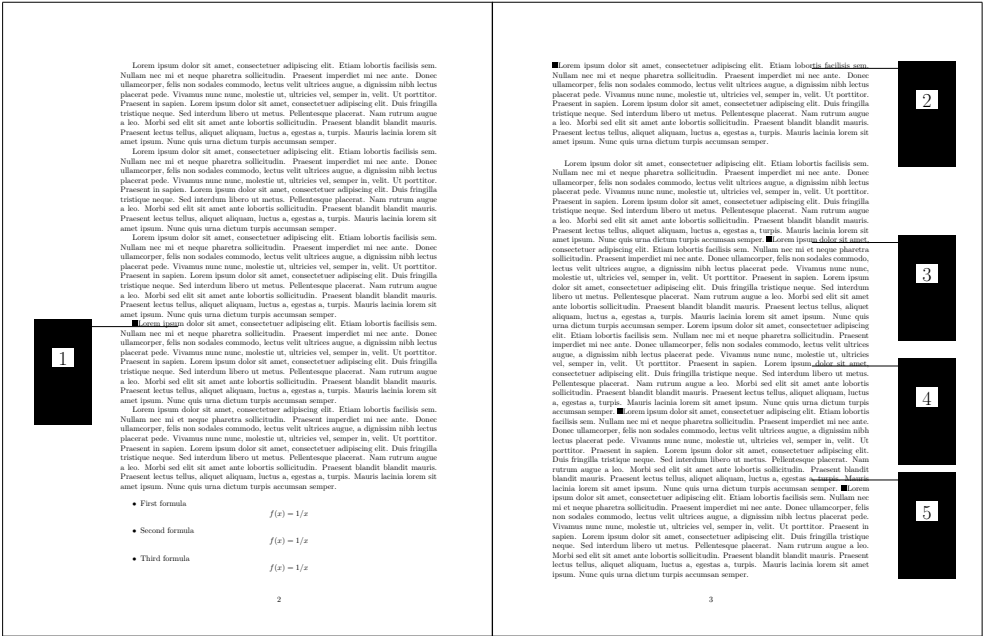


Figure 4: marginfit package

The `mparhack` package[1] addresses Problem 1. This is demonstrated in Figure 2. Problem 1 is completely fixed by this package without any known drawbacks and it also works in twocolumn mode. However, the package does not consider problem 2.

On the other hand, the `marginfix` package[2] addresses Problem 2. This is shown in Figure 3. However, this comes with a lot of disadvantages:

- The `marginfix` package does not support twocolumn mode.
- The  $\LaTeX$  Problem 1 manifests itself in another way: Margin notes are attached to the wrong page instead of the wrong side.
- The baseline (of the first line) of the margin note is not aligned with the baseline of the callout position. Instead the top of the margin note is aligned with the top of a strut at the callout position. This gives us the misalignment seen for Margin Note 1 as the height of this margin note is a bit more than that of a strut.
- The margin notes are positioned at the natural height of the callout position. However, the actual height can be different depending on the glue on the page. The `marginfix` package does not consider this glue and the effect can be seen for Margin Note 3.

In contrast to the packages above, the `marginfit` package is not a hack to the margin paragraph mechanism of  $\LaTeX$  but a complete reimplementation. Instead of calling the output routine for each margin note, it uses  $\TeX$  insertions. Hence, the margin notes are not placed individually at callout time but together at the time a complete column is assembled. Therefore, it does not suffer from the problems described above. The result can be observed in Figure 4.

## 2 Usage

The `marginfit` package is a drop-in replacement for the normal margin paragraph mechanism. Therefore, putting `\usepackage{marginfit}` into the preamble of your document is sufficient. This package requires the `\pdfsavepos` functionality from `pdfTeX`.

## 3 Implementation

`\marginfit@insert` We define a new insertion class named `\marginfit@insert`. All margin paragraphs will be inserted there. At shipout time the respective box will hold all the inserted margins for the respective page or column.

```
1 \newinsert\marginfit@insert
```

As a margin note does not account for the page goal, we set the corresponding magnification count to zero.

```
2 \count\marginfit@insert0
```

Also the respective skip with the amount of fixed size material is set to 0pt.

```
3 \skip\marginfit@insert\z@
```

As the magnification factor is zero, we can not track the amount of margin material. Also, we do not want margins to be shifted to another page. Hence, the corresponding dimension with the maximum amount of insertions per page is set to the maximum.

```
4 \dimen\marginfit@insert\maxdimen
```

`\marginfit@reverse` We allocate a new box register where we store the margin notes in reverse order.

```
5 \newbox\marginfit@reverse
```

`\marginfit@box` We also define a few aliases for temporary box registers.

```
\marginfit@left 6 \chardef\marginfit@box0
```

```
\marginfit@right 7 \chardef\marginfit@left0
```

```
8 \chardef\marginfit@right1
```

`\c@marginfit@w` A few counters are needed to enumerate labels. The first counter is used when writing the positions of margin notes to the aux file and the second is used when reading them in again. The third counter is used for the top positions of margin columns.

```
\c@marginfit@r
```

```
\c@marginfit@t
```

```
9 \newcount\c@marginfit@w
```

```
10 \newcount\c@marginfit@r
```

```
11 \newcount\c@marginfit@t
```

`\marginfit@top` We need some dimension registers to compute the placement of margin notes.

`\marginfit@min` Those are the current top of the margin column, the minimal position the current

`\marginfit@pos` margin note can be placed in and the callout position of the current margin note.

```
12 \newdimen\marginfit@top
```

```
13 \newdimen\marginfit@min
```

```
14 \newdimen\marginfit@pos
```

`\marginfit@writepos` The following macro saves the current position as a label to the aux file. The parameter takes the number / name of the label in already expanded form.

```
15 \def\marginfit@writepos#1{%
```

```
16   \pdfsavepos%
```

```
17   \write\@auxout{\string\@newl@bel{label@marginfit}{#1}{\the\pdfastypos}}%
```

```
18 }
```

`\marginfit@boxforeach` We define a macro that loops over a vbox to execute code for each hbox in that vbox. Due to the limited possibilities to access the inner boxes, we must do that in reverse order. The first parameter is the box register number and the second parameter is the code that should be executed for each hbox. During execution `\marginfit@box` holds the current hbox.

```
19 \def\marginfit@boxforeach#1#2{%
```

```
20   \setbox#1\vbox{\unvbox#1 \global\setbox\marginfit@box\lastbox}%
```

```
21   \ifvoid\marginfit@box\else%
```

```
22     #2%
```

```

23      \marginfit@boxforeach#1{#2}%
24      \fi%
25 }

```

`\marginfit@pass@i` This is the first pass of the margin positioning routine. For each margin note, the following macro splits the packed hbox to get the left and right version of the margin paragraph. In the following we will use the correct version, due to the setting of `\marginfit@side`. The minimum position is updated based on the size of the margin. The margin notes will be saved in reverse order in the reverse box.

```

26 \def\marginfit@pass@i{%
27   \setbox\marginfit@left\hbox{\unhbox\marginfit@box%
28     \global\setbox\marginfit@right\lastbox}%
29   \setbox\marginfit@right\hbox{\box\marginfit@right}%
30   \advance\marginfit@min\ht\marginfit@side%
31   \advance\marginfit@min\dp\marginfit@side%
32   \advance\marginfit@min\marginparpush%
33   \setbox\marginfit@reverse\vbox{\unvbox\marginfit@reverse%
34     \box\marginfit@side}%
35 }

```

`\marginfit@pass@ii` This is the second pass of the margin positioning routine. For each margin note, the callout position is read from the aux file. We compute the glue above the margin note. If the minimum position requires it, we shift the margin note upwards. If the current top position requires it, we shift the margin note downwards. Afterwards we update the top and minimum positions. Finally, the margin note is attached to the margin box.

```

36 \def\marginfit@pass@ii{%
37   \global\advance\c@marginfit@r\@ne%
38   \ifcsname label@marginfit@the\c@marginfit@r @m\endcsname%
39     \marginfit@pos\csname label@marginfit@the\c@marginfit@r @m\endcsname sp%
40   \fi%
41   \advance\marginfit@pos\ht\marginfit@box%
42   \ifdim\marginfit@pos<\marginfit@min%
43     \marginfit@pos\marginfit@min%
44   \fi%
45   \advance\marginfit@top-\marginfit@pos%
46   \ifdim\marginfit@top<\marginparpush%
47     \advance\marginfit@pos\marginfit@top%
48     \advance\marginfit@pos-\marginparpush%
49     \marginfit@top\marginparpush%
50   \fi%
51   \vskip\marginfit@top%
52   \marginfit@top\marginfit@pos%
53   \advance\marginfit@top-\ht\marginfit@box%
54   \advance\marginfit@top-\dp\marginfit@box%
55   \advance\marginfit@min-\ht\marginfit@box%
56   \advance\marginfit@min-\dp\marginfit@box%
57   \advance\marginfit@min-\marginparpush%
58   \hrule height\z@%

```

```

59     \box\marginfit@box%
60     \hrule height\z@%
61 }%

```

`\marginfit@marginbox` The following macro constructs the margin box. First, at the top of the box the position is saved and read in as `\marginfit@top` if available. Then, `\marginfit@min` is computed as the bottom of the box. The first `boxforeach` loop is executed on the box with the margin insertions. It computes the real `\marginfit@min` for the first margin note and saves the margin notes in reverse order into the reverse box. The second pass is executed on that reverse box and puts the margins into the right positions.

```

62 \def\marginfit@marginbox{%
63     \vbox to\@colht {%
64         \global\advance\c@marginfit@t\@ne%
65         \expandafter\marginfit@writepos\expandafter{\the\c@marginfit@t @t}%
66         \ifcsname label@marginfit@\the\c@marginfit@t @t\endcsname%
67             \marginfit@top\csname label@marginfit@\the\c@marginfit@t @t\endcsname sp%
68         \fi%
69         \marginfit@min\marginfit@top%
70         \advance\marginfit@min-\@colht%
71         \advance\marginfit@min-\marginparpush%
72         \advance\marginfit@top\marginparpush%
73         \vskip-\marginparpush%
74         \marginfit@boxforeach\marginfit@insert\marginfit@pass@i%
75         \marginfit@boxforeach\marginfit@reverse\marginfit@pass@ii%
76         \vfill%
77     }%
78 }%

```

`\marginfit@leftmargin` The following macro appends a left margin to the current column.

```

79 \def\marginfit@leftmargin{%
80     \let\marginfit@side\marginfit@left%
81     \hss%
82     \marginfit@marginbox%
83     \hskip\marginparsep%
84     \box\@outputbox%
85 }

```

`\marginfit@rightmargin` The following macro appends a right margin to the current column.

```

86 \def\marginfit@rightmargin{%
87     \let\marginfit@side\marginfit@right%
88     \box\@outputbox%
89     \hskip\marginparsep%
90     \marginfit@marginbox%
91     \hss%
92 }

```

`\@makecol` We redefine `\@makecol` and add the procedure to append a margin to the current column. First, the original `\@makecol` is executed. If we have margin material,

we decide on which side the margin column will be and add it to that side.

```

93 \let\marginfit@makecol\@makecol
94 \def\@makecol{%
95     \marginfit@makecol%
96     \ifvoid\marginfit@insert\else%
97         \setbox\@outputbox\vbox to\@colht {%
98             \hbox to\columnwidth {%
99                 \if@twocolumn%
100                     \if@firstcolumn%
101                         \marginfit@leftmargin%
102                     \else%
103                         \marginfit@rightmargin%
104                     \fi%
105                 \else%
106                     \ifodd\c@page%
107                         \marginfit@rightmargin%
108                     \else%
109                         \marginfit@leftmargin%
110                     \fi%
111                 \fi%
112             }%
113         }%
114     \fi%
115 }

```

`\marginpar` We redefine `\marginpar` to do an insertion instead of a call to the output routine. First, we save the callout position to the aux file. Then, we combine the left and right versions of the margin paragraph into an hbox and, finally, make an insertion with that hbox.

```

116 \def\marginpar{%
117     \global\advance\c@marginfit@w\@ne%
118     \expandafter\marginfit@writepos\expandafter{\the\c@marginfit@w @m}%
119     \ifnextchar[\marginfit@mpar@ii\marginfit@mpar@i%
120 }
121 \long\def\marginfit@mpar@i#1{%
122     \@savemarbox\marginfit@box{#1}%
123     \setbox\marginfit@box\hbox{\copy\marginfit@box\box\marginfit@box}%
124     \marginfit@mpar@iii%
125 }
126 \long\def\marginfit@mpar@ii[#1]#2{%
127     \@savemarbox\marginfit@left{#1}%
128     \@savemarbox\marginfit@right{#2}%
129     \setbox\marginfit@box\hbox{\box\marginfit@left\box\marginfit@right}%
130     \marginfit@mpar@iii%
131 }
132 \def\marginfit@mpar@iii{%
133     \insert\marginfit@insert{\box\marginfit@box}%
134 }

```



## References

- [1] Tom Sgouros and Stefan Ulrich. `mparhack.sty`. CTAN:  
<http://ctan.org/pkg/mparhack>
- [2] Stephen Hicks. `marginfix` package documentation. CTAN:  
<http://ctan.org/pkg/marginfix>

## Change History

v1.0  
 General: Initial version . . . . . 1

## 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.

Symbols	
<code>\@auxout</code> . . . . .	17
<code>\@colht</code> . . . . .	63, 70, 97
<code>\@makecol</code> . . . . .	<u>93</u>
<code>\@newl@bel</code> . . . . .	17
<code>\@outputbox</code> . . . . .	84, 88, 97
<code>\@savemarbox</code> . . . . .	122, 127, 128
C	
<code>\c@marginfit@r</code> . . . . .	<u>9</u> , 37–39
<code>\c@marginfit@t</code> . . . . .	<u>9</u> , 64–67
<code>\c@marginfit@w</code> . . . . .	<u>9</u> , 117, 118
<code>\c@page</code> . . . . .	106
<code>\columnwidth</code> . . . . .	98
I	
<code>\if@firstcolumn</code> . . . . .	100
<code>\if@twocolumn</code> . . . . .	99
<code>\insert</code> . . . . .	133
M	
<code>\marginfit@box</code> . . . . .	<u>6</u> , 20, 21, 27, 41, 53–56, 59, 122, 123, 129, 133
<code>\marginfit@boxforeach</code> . . . .	<u>19</u> , 74, 75
<code>\marginfit@insert</code> . . . .	<u>1</u> , 74, 96, 133
<code>\marginfit@left</code> . . .	<u>6</u> , 27, 80, 127, 129
<code>\marginfit@leftmargin</code> . .	<u>79</u> , 101, 109
<code>\marginfit@makecol</code> . . . . .	93, 95
<code>\marginfit@marginbox</code> . . . .	<u>62</u> , 82, 90
<code>\marginfit@min</code> . . . . .	.. <u>12</u> , 30–32, 42, 43, 55–57, 69–71
<code>\marginfit@mpar@i</code> . . . . .	119, 121
<code>\marginfit@mpar@ii</code> . . . . .	119, 126
<code>\marginfit@mpar@iii</code> . . .	124, 130, 132
<code>\marginfit@pass@i</code> . . . . .	<u>26</u> , 74
<code>\marginfit@pass@ii</code> . . . . .	<u>36</u> , 75
<code>\marginfit@pos</code> . . . . .	.. <u>12</u> , 39, 41–43, 45, 47, 48, 52
<code>\marginfit@reverse</code> . . . . .	<u>5</u> , 33, 75
<code>\marginfit@right</code> <u>6</u> , 28, 29, 87, 128, 129	
<code>\marginfit@rightmargin</code> .	86, 103, 107
<code>\marginfit@side</code> . . . .	30, 31, 34, 80, 87
<code>\marginfit@top</code> . . . . .	.. <u>12</u> , 45–47, 49, 51–54, 67, 69, 72
<code>\marginfit@writepos</code> . . . .	<u>15</u> , 65, 118
<code>\marginpar</code> . . . . .	<u>116</u>
<code>\marginparpush</code> 32, 46, 48, 49, 57, 71–73	
<code>\marginparsep</code> . . . . .	83, 89
N	
<code>\newbox</code> . . . . .	5
<code>\newcount</code> . . . . .	9–11
<code>\newdimen</code> . . . . .	12–14
<code>\newinsert</code> . . . . .	1

<b>P</b>	<b>W</b>
\pdflastypos .....	17
\pdfsavepos .....	16
	\write ..... 17