

A package for rotated objects in L^AT_EX^{*}

Robin Fairbairns Sebastian Rahtz Leonor Barroca

printed October 31, 2025

This file is maintained by the L^AT_EX Project team.
Bug reports can be opened (category `graphics`) at
<https://latex-project.org/bugs.html>.

Contents

1	Introduction	1
2	Usage	2
2.1	Package options	2
2.2	Float environments	2
2.3	Other environments and commands	2
2.4	Positioning	3
3	Setup	3
4	Turning and rotation environments	4
5	Sideways figures and tables	5
5.1	Rotated captions only	9
6	Last-minute infrastructure	10
7	History	10

Abstract

This article documents a L^AT_EX package, ‘rotating.sty’, which performs most sorts of rotation one might like, including rotation of complete floating figures and tables.

1 Introduction

The package provides:

- two new environments, `sidewaystable` and `sidewaysfigure`, each of which produces a single page-size float with contents rotated ± 90 degrees; and

^{*}This file has version number v2.16e, last revised 2025/01/14

- a variety of other rotation-related commands and environments.

Note that the package uses rotation facilities from the `graphicx` package. When generating DVI output, users should note that rotation is typically *not* visible in a DVI viewer: conversion to, and viewing, PostScript or PDF is necessary.

2 Usage

2.1 Package options

Sideways figures and tables always take up the whole page. In single-sided documents, they may be rotated so that the bottom of the figures is on the left (package option ‘`counterclockwise`’) or the right (package option ‘`clockwise`’). The default is to turn so that the bottom is on the right (option ‘`clockwise`’).

Option ‘`anticlockwise`’ is an alias for ‘`counterclockwise`’.

If the ‘`twoside`’ option has been given to the main document class (either explicitly, or implicitly as in the default for book class), the package will rotate sideways figures according to the page number (this requires at least two passes through L^AT_EX). If you want the ‘`twoside`’ option, but want the figures always in one direction, use the ‘`figuresright`’ or ‘`figuresleft`’ options to the package.

The package can produce a lot of logging information; the amount of information is controlled by the package options ‘`quiet`’ (fewest messages; default), ‘`log`’ and ‘`chatter`’ (most messages).

All other options are passed to the `graphicx` package when it is loaded to provide rotation functions.

2.2 Float environments

The environments `sidewaystable` and `sidewaysfigure` introduce landscape-form floating tables and figures, respectively. (Each of the environments has a “starred” version, such as `sidewaystable*`, for a single-column float in a double-column area of the document.)

New rotated environments may be declared using the combined facilities of the `float` and `rotfloat` packages.

2.3 Other environments and commands

The package provides other L^AT_EX environments:

`sideways` prints the contents of the environment turned through 90 degrees counterclockwise;

`turn` prints the contents turned through an arbitrary angle (the argument to the environment);

`rotate` prints the contents turned through an arbitrary angle but does *not* leave any space for the result

The command `\turnbox{⟨angle⟩}{⟨matter to turn⟩}` is a macro version of the `rotate` environment.

A set of examples is given in the file `rotex.tex`

2.4 Positioning

Floats appear one to a page, and are positioned by spacer skips that appear (logically) above and below the floating object. The skips, `\rotFPtop` and `\rotFPbot`, are initialised from the standard L^AT_EX (internal) `\@fptop` and `\@fpbot` skips. As a result, by default, rotated floats appear horizontally centred on their float pages.

Some sensible values for the registers are:

<code>\rotFPtop</code>	<code>\rotFPbot</code>	Effect
0pt plus 1fil	0pt plus 1fil	figure/table appears in middle (default value)
0pt	0pt plus 1fil	figure/table appears with its top nearest the edge of the page
0pt plus 1fil	0pt plus 2fil	figure/table's bottom appears twice as far from the edge as the top does

3 Setup

Now we present the documented code. First, package options.

Note that the `clockwise` and `counterclockwise` options are present for compatibility only.

```

1 (*package)
2 \DeclareOption{clockwise}{%
3   \AtBeginDocument{\setkeys{Grot}{units=360}}%
4 }
5 \DeclareOption{counterclockwise}{%
6   \AtBeginDocument{\setkeys{Grot}{units=-360}}%
7 }
8 \DeclareOption{anticlockwise}{\ds@counterclockwise}

```

Control figure orientation

```

9 \DeclareOption{figuresleft}{%
10   \@rot@twosidefalse
11   \def\rot@LR{0}%
12 }
13 \DeclareOption{figuresright}{%
14   \@rot@twosidefalse
15   \def\rot@LR{-1}%
16 }

```

control messages:

```

17 \newif\ifrot@messages
18 \DeclareOption{quiet}{%
19   \rot@messagesfalse
20   \let\rot@message\@gobble % pro tem -- should suppress altogether
21 }
22 \DeclareOption{log}{%
23   \rot@messagestrue
24   \def\rot@message{\PackageInfo{rotating}}%
25 }
26 \DeclareOption{chatter}{%
27   \rot@messagestrue
28   \def\rot@message{\PackageWarning{rotating}}%
29 }

```

`\if@rot@twoside` A couple of commands for passing rotation direction around

```
\rot@LR 30 \newif\if@rot@twoside
31 \if@twoside
32   \@rot@twosidetrue
33 \else
34   \@rot@twosidefalse
35 \fi
36 \def\rot@LR{-1}
```

Pass any unknown options to the `graphicx` package, and set up defaults and process the options.

```
37 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{graphicx}}
38 \ExecuteOptions{clockwise,quiet}
39 \ProcessOptions
Other initialisation
40 \RequirePackage{graphicx}
41 \RequirePackage{ifthen}
```

`\rotdriver` The command `\rotdriver` allows a user to specify an initialisation file, a sort of non-automatically-loaded driver (in the graphics, hyperref sense).

```
42 \def\rotdriver#1{\makeatletter\input{#1.def}\makeatother}
```

The `r@tfl@t` counter is used when generating ‘labels’ for determining what side of the page the float is on, in twoside mode.

```
43 \newcounter{r@tfl@t}
44 \setcounter{r@tfl@t}{0}
```

Positioning skips (see above).

```
45 \newskip\rotFPtop \rotFPtop=\@fptop
46 \newskip\rotFPbot \rotFPbot=\@fpbot
```

4 Turning and rotation environments

`sideways` Environment to turn the contents through 90 degrees. `\Grot@box` issues a `\leavevmode` but for tagging (when using `lualatex`) this is too late, it should happen before the box is set.

```
47 \def\sideways{%
48   \leavevmode
49   \Grot@setangle{90}%
50   \setbox\z@\color@hbox\ignorespaces}
51 \def\endsideways{%
52   \unskip\color@endbox
53   \Grot@x\z@
54   \Grot@y\z@
55   \Grot@box
56 }
```

`turn` Rotate the contents of the environment, leaving the appropriate space

```
57 \def\turn#1{%
58   \leavevmode
59   \Grot@setangle{#1}%
```

```

60 \setbox\z@\color@hbox\ignorespaces}
61 \def\endturn{%
62 \unskip\color@endbox
63 \Grot{x}\z@
64 \Grot{y}\z@
65 \Grot@box
66 }

```

rotate Rotate the contents of the environment, leaving *no space*.

```

67 \def\rotate#1{%
68 \leavevmode
69 \Grot@setangle{#1}%
70 \setbox\z@\color@hbox\ignorespaces}
71 \def\endrotate{%
72 \unskip\color@endbox
73 \Grot{x}\z@
74 \Grot{y}\z@
75 \wd0\z@\dp0\z@\ht0\z@
76 \Grot@box
77 }

```

\turnbox A macro version of the ‘rotate’ environment.

```

78 \def\turnbox#1#2{%
79 \leavevmode
80 \Grot@setangle{#1}%

```

Note: grouping within the box makes `\color@hbox` unnecessary, I think.

```

81 \setbox\z@\hbox{{#2}}%
82 \Grot{x}\z@\Grot{y}\z@
83 \wd0\z@\dp0\z@\ht0\z@
84 \Grot@box
85 }

```

5 Sideways figures and tables

Now for the macros to provide a complete environment for sideways figures and tables. We define two environments **sidewaysfigure** and **sidewaystable** that fit in with the normal table and figure floats. These are ‘fixed’ environments that just do 90 degree rotation, but it would be easy to parameterize this to do other rotations if needed (the mind boggles...)

\@rotfloat First a generalised ‘rotfloat’ environment. We need to intercept L^AT_EX’s float macros, in order to change the assumed width of a float being `\columnwidth`. We want it to work on a width of `\textheight` so that when we rotate the float, it comes out the right height. This is not actually very satisfactory, since what we *really* want is for rotated floats to occupy the space they actually *use*. The captions are a problem — since they can precede the figure or table, we cannot set them in a box of the right width (ie the *height* of the forthcoming object), because it has not happened yet. The result of these difficulties is that rotated figures always end up as full page figures.

```

86 \newsavebox\rot@float@box
87 \def\@rotfloat#1{%

```

```

88 \@ifnextchar[%
89   {\@xrotfloat{#1}}%
90   {\edef\@tempa{\noexpand\@xrotfloat{#1}[\csname fps@#1\endcsname]}\@tempa}%
91 }
92 \def\@xrotfloat#1[#2]{%
93   \@float{#1}[#2]%

```

Set the float contents in a box of width `\textheight` instead of `\columnwidth`.

```

94   \begin{lrbox}\rot@float@box
95   \SuspendTagging{\minipage}%
96   \begin{minipage}\textheight
97   \ResumeTagging{\minipage}%
98   \UseTaggingSocket{para/restore}%
99 }

```

`\end@rotfloat` We call L^AT_EX's `\end@float` macro having previously rotated the box `\@currbox`. The rotation is either clockwise or anti-clockwise, depending on whether the page is odd or even; in oneside mode it is always odd.

```

100 \def\end@rotfloat{%

```

If we are going to know whether pages are odd or even, we need to use the a variant `\pageref` mechanism and our own specialised labels.

```

101   \par\SuspendTagging{\minipage}%
102   \end{minipage}%
103   \ResumeTagging{\minipage}%
104   \end{lrbox}%
105   \stepcounter{r@tfl@t}%
106   \rot@label{RF\ther@tfl@t}%
107   \rot@pageref{RF\ther@tfl@t}{\R@@page}%
108   \edef\@tempa{Adding sideways \@capytype\space on page \R@@page\space}
109   \rot@mess@toks\expandafter{\@tempa}
110   \wd\rot@float@box\z@
111   \ht\rot@float@box\z@
112   \dp\rot@float@box\z@
113   \vbox to \textheight{%

```

We need to know for sure which direction rotation is going to be in, so locally reset the graphics units.

```

114   \setkeys{Grot}{units=360}%
115   \if@rot@twoside
116   \else
117     \let\R@@page\rot@LR
118   \fi
119   \ifthenelse{\isodd{\R@@page}}{%
120     \if@rot@twoside
121       \rot@mess@toks\expandafter{\the\rot@mess@toks (right hand page)}%
122     \fi
123     \vfill
124     \@@line{%
125       \hskip\rotFPtop
126       \rotatebox{90}{\box\rot@float@box}%
127       \hskip\rotFPbot
128     }%
129   }{%
130     \if@rot@twoside

```



```

175 \ht\rot@float@box\z@
176 \dp\rot@float@box\z@
177 \vbox to \textheight{%

```

We need to know for sure which direction rotation is going to be in, so locally reset the graphics units.

```

178 \setkeys{Grot}{units=360}%
179 \if@rot@twoside
180 \else
181 \let\R@@page\rot@LR
182 \fi
183 \ifthenelse{\isodd{\R@@page}}{%
184 \ifrot@messages
185 \if@rot@twoside
186 \rot@mess@toks\expandafter{\the\rot@mess@toks (right hand page)}}%
187 \fi
188 \fi
189 \vfill
190 \@@line{%
191 \hskip\rotFPtop
192 \rotatebox{90}{\box\rot@float@box}%
193 \hskip\rotFPbot
194 }%
195 }{%
196 \ifrot@messages
197 \if@rot@twoside
198 \rot@mess@toks\expandafter{\the\rot@mess@toks (left hand page)}}%
199 \fi
200 \fi%
201 \@@line{%
202 \hskip\rotFPbot
203 \rotatebox{-90}{\box\rot@float@box}%
204 \hskip\rotFPtop
205 }%
206 \vfill
207 }%
208 \rot@message{\the\rot@mess@toks}%
209 }%
210 \end@dblfloat
211 }

```

```

sidewaysfigure* (env.)
sidewaystable* (env.) 212 \newenvironment{sidewaystable*}
213 \rot@dblfloat{table}}
214 \end@rot@dblfloat}
215 \newenvironment{sidewaysfigure*}
216 \rot@dblfloat{figure}}
217 \end@rot@dblfloat}
218

```

\rot@label Note that we used \rot@label, not \label; this variant writes (just) the *true* page number, not the value of \thepage; this “true” value then needs special treatment in \protected@write, just as \thepage already has. \rot@pageref{\generated label name} {\cs to set to pageno} then returns the labelled page’s number (or 0 if@rot@refundefined

if label not yet defined). If label not defined, flags using `\rot@refundefinedtrue` for end-document to pick up. (later...)

```

219 \def\rot@thepage{\@arabic\c@page}
220 \def\rot@label#1{\@bsphack
221   \rot@protected@write{\@auxout}{\}%
222     {\string\newlabel{#1}{\rot@thepage}}%
223   \@esphack}
224 \def\rot@pageref#1#2{%
225   \expandafter\ifx\csname r@#1\endcsname\relax
226     \global\@rot@refundefinedtrue
227     \def#2{0}%
228   \else
229     \edef#2{\csname r@#1\endcsname}%
230   \fi
231 }
232 \long\def\rot@protected@write#1#2#3{%
233   \begingroup
234     \let\rot@thepage\relax
235     #2%
236     \let\protect\@unexpandable@protect
237     \edef\reserved@a{\write#1{#3}}%
238     \reserved@a
239   \endgroup
240   \if@nobreak\ifvmode\nobreak\fi\fi
241 }
242 \newif\if@rot@refundefined
243 \global\@rot@refundefinedfalse

```

`\rot@mess@toks` A token register to build up debugging messages

```

244 \newtoks\rot@mess@toks

```

5.1 Rotated captions only

`\rotcaption` Sometimes you may find that the rotation of complete figures does not give quite the right result, since they always take up the whole page. You may prefer to rotate the caption and the float contents separately within a conventional figure. Here we offer a suggestion for a `\rotcaption` command, which inserts the caption rotated by 90 degrees. It is essentially a copy of the normal captioning code. Packages which define the `\makecaption` command may also need to define `\@makerotcaption`.

```

245 \def\rotcaption{\refstepcounter\@captype\@dblarg{\@rotcaption\@captype}}
246 \long\def\@rotcaption#1[#2]#3{%
247   \addcontentsline{\csname ext@#1\endcsname}{#1}{%
248     \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}%
249   \par
250   \begingroup
251     \@parboxrestore
252     \normalsize
253     \@makerotcaption{\csname fnum@#1\endcsname}{#3}%
254   \endgroup}
255 \long\def\@makerotcaption#1#2{%
256   \setbox\@tempboxa\color@hbox#1: #2\color@endbox
257   \ifdim \wd\@tempboxa > .8\vsize

```

```

258      \rotatebox{90}{%
259        \begin{minipage}{.8\textheight}#1: #2\end{minipage}%
260      }%
261      \else%
262        \rotatebox{90}{\box\@tempboxa}%
263      \fi
264      \nobreak
265      \hspace{12pt}%
266 }
267 \end{package}

```

6 Last-minute infrastructure

`\color@hbox` These macros aren't provided in L^AT_EX, by default (I seem to have assumed that
`\color@endbox` they were...)

```

268 \AtBeginDocument{%
269   \providecommand\color@hbox{\hbox\bgroup}%
270   \providecommand\color@vbox{\vbox\bgroup}%
271   \providecommand\color@endbox{\egroup}%
272 }

```

7 History

Version 2.0 is a complete re-write, with most of the work now being done by the L^AT_EX 2_ε graphics package.

Version 2.1 provides a 'clockwise' option to reinstate the behaviour described in the 'L^AT_EX Companion'

Version 2.2 just intercepts the standard float macros instead of copying and changing the. The 'twoside' option is obeyed.

Version 2.5 corrects problems in sideways figures.

Version 2.6 is a rewrite of the sideways floats via Frank Mittelbach (to whom many thanks for looking at the mangy code).

Version 2.7 is checked for L^AT_EX of December 94, and adds the option of twoside behaviour independent of the general twoside.

Version 2.8 cleans up some mistakes pointed out by Harald Axel Sommerfeldt.

Version 2.9 cleans up some (more) mistakes pointed out by Harald Axel Sommerfeldt.

Version 2.13a permits positioning of rotated floats in the same way as they are positioned in 'normal' floats.

Version 2.14 is the first to be published anywhere as the outcome of maintenance by Robin Fairbairns.

Version 2.15 deals with page-numbering bug for auto-float-rotation, and tidying of messages; published to ctan

Version 2.16 uses colour boxes as necessary; published to ctan. Version 2.16a provides the colour box commands `\AtBeginDocument`.