`rotpages.sty` — Multiple page rotation in LaTeX

Sergio Callegari

January 25th, 2002

Abstract

This LaTeX package permits to format documents where small sets of pages are rotated by 180 degrees and rearranged, so that they can be read by turning the printed copy upside-down. It was primarily meant for collecting exercises and solutions: the package permits to print the exercise text normally and the solution text upside-down in order to make it a bit harder for the students to unintentionally read the solutions before solving the exercises autonomously. Other uses are obviously possible.

1 Introduction

The package *rotpages.sty* is meant to be used when parts of a document should be made purposely difficult to read or obfuscated. An obvious way to make something difficult to read is to print it upside-down. This is well known by all those keen on puzzles and cross-words as in puzzle magazines the solutions are often printed "the wrong way" to prevent the readers from spoiling the pleasure of the quiz by unintentionally reading the solutions.

A similar situation emerges in collecting exercises and solutions as an aid for students training. The solutions must be there, so that the students can check their results and avoid getting stuck when they cannot answer some questions. At the same time the solutions should not be easily readable, otherwise the temptation to read them before attempting to solve a problem can be too strong. A typical format for exercise collections consists in putting all the exercises at the beginning and all the answers at the end. However, this makes people flap pages back and forth all the time. Having the solutions close to the relevant exercises and obfuscating them by page rotation can be an alternative practice.

In LaTeX, it is very easy to rotate small amounts of text. The package `graphics` is a valid aid in this sense. However, it is not easy to deal with large amounts of text. If the rotated material has to span many pages, it cannot be put in a `\rotatebox` command. Furthermore, if many pages need to be rotated, it is not sufficient to flip them individually: they must also be rearranged so that they appear in the proper order when the printed copy of the document is read "upside-down".

The package `rotpages.sty` permits to rotate a few pages, rearranging them consistently and preserving the normal LaTeX algorithm for breaking the material into pages.

2 Usage

The package defines two basic commands: `\rotboxpages` and `\endrotboxpages`, which take no arguments.

The command `\rotboxpages` closes the current page and delimits the beginning of the text which should be typeset upside-down. The command `\endrotboxpages` marks the end of the region which is typeset upside-down and makes sure that the rotated pages are flushed onto the `dvi` file. Then it opens a fresh page. The rotated material is by default printed surrounded by a frame (but this can be changed). Page headers and footers continue display the proper way.

When used in two-column mode, the package correctly rotates the columns independently, rearranging them if necessary. The rotated columns get individually framed. Note that the package might not operate correctly with other packages that also alter the paging algorithms.

The files in the `Examples` directory provide some usage examples.

2.1 Personalization

The behaviour of `rotpages.sty` can be largely personalized. Rotpages can frame the rotated pages, scale them, enhance them with notes (e.g. "rotate the book to read this page properly") and so on. All this is done by means of two "hooks".

The first hook is set by renewing the command `\rotboxAtRotationHook`. Whenever LaTeX has finished preparing a page or a column and rotation is activated, `rotpages.sty` assures that the material is not directly shipped to the dvi file. On the contrary, the material is put into a box and this box is saved onto a stack. When the rotation mode is terminated, all the boxes saved onto the stack are shipped to the dvi file in inverse order. The stack mechanism is obviously necessary because the pages need to be rearranged while flipped upside down. The `\rotboxAtRotationHook` is invoked on the pages/columns just before saving them onto the stack (i.e. right after rotation). Hence the `\rotboxAtRotationHook` takes a box as its argument and should return a box. In the simplest case the `\rotboxAtRotationHook` command can just return its argument. In more complex scenarios it can put a frame around it and so on.

The second hook is set by renewing the command `\rotboxAtShippingHook`. The idea is the same as for the `\rotboxAtRotationHook`, however this hook is invoked when taking the pages off the stack rather than while putting them onto the stack.

In most cases the two hooks can be used indifferently. However there is a very subtle difference among the two. When putting the pages onto the stack one can easily keep trace of the relative number of the page among the lot that is being rotated, but cannot know the actual page number as this will be determined only at shipping time. On the contrary, when taking the pages out of the stack one can know the actual page number, because the shipping is effectively taking place. This differenc can be used to add numbering labels and rotated/non-rotated text to the pages being rotated as one of the files in the `Examples` directory shows.

In any case, consider that `rotpages.sty` has no problems in dealing with labels and page-references and does so in the expected way, i.e. using the "actual" page numbers.

The default for the two hooks `\rotboxAtShippingHook` and `\rotboxAtRotationHook` is the following: the `\rotboxAtRotationHook` frames the page, and the `\rotboxAtShippingHook` does nothing.

The package `rotpages.sty` also defines a couple of lengths: `\rotboxheight` and `\rotboxwidth`. These are related to the size of the rotated pages/columns. It is the user responsibility to assure that the rotated pages are exactly as large as the normal ones. However, if a page is framed, one must take into account that the frame occupies space. If this fact were not considered, the rotated pages would come out with smaller characters than the normal ones. In fact, the text *plus the frame* should occupy as much space as a normal page. The solution to overcome this problem is to put slightly less text on the pages which get rotated, i.e. to format the text for a *virtual* page which is smaller than the normal ones, accounting for the frame. The lengths `\rotboxheight` and `\rotboxwidth` have exactly this purpose, setting the size of the virtual pages on which the text to be rotated gets formatted.

The default for `\rotboxheight` and `\rotboxwidth` is the size of a normal page, less the space occupied by the default frame.

To conclude, the `rotpages.sty` package defines a boolean variable which can be used to test whether the current text is being rotated or not. This is the variable `\boolean{rotboxactive}` which can be tested using the `\ifthenelse` command.

3 Known bugs and quirks

The package is unable to deal with float material. Avoid float material in the rotated pages. Equations, quotations, displayed math and other non-float material can safely be used.

Apart from this, the package is relatively young and no bugs are known so far. It should be observed that the `.dvi` files produced with `rotpages.sty` cannot be viewed properly with `xdvi`, `kdvi` and most dvi previewers. However, this is not a problem of `rotpages.sty`, rather of `xdvi` which does not understand correctly some of the `\specials` produced by the `graphics` package which is internally invoked by `rotpages` for the page rotation. The postscript files produced by `dvips` display and print correctly. The package is also compatible with pdfLaTeX.

In order to re-arrange the pages which are rotated, the shipping of the formatted material to the `dvi` file needs to be deferred until all the rotated pages are processed. This puts a limit on the number of pages which can be rotated at once, since the deferring mechanisms uses a stack which eats up the LaTeX memory.

4 Internal operation

The package works by redefining the `\@makecol` macro, which is normally used by LaTeX to format and ship a page to the dvi file. Particularly, at the `\rotboxpages` command, the `\@opcol` and the `\@startcolumn` macros are temporarily disabled, so that no page gets shipped to the `dvi` file and no page number can be updated. Also the current page size is altered, following the `\rotboxheight` and `\rotboxwidth` lengths. Furthermore, the `\@makecol` macro is temporarily modified, so that the formatted pages get processed by the `\rotboxAtRotationHook` command, rotated and accumulated into a stack.

At the `\endrotboxpages` command, the behaviour of LaTeX gets reset to the standard. After that, the pages that had been accumulated onto the `rotpages` stack are processed by the `\rotboxAtShippingHook` command, scaled to a normal page size and flushed to the `dvi` file. The stack mechanism assures that the pages get to the `dvi` file in the correct (rearranged) order.

For the page rotation, `rotpages.sty` relies on the standard `graphics` package and so does for scaling.

5 To-do

- Allow the package to pass options to the underlying graphics package, particularly concerning the drivers (dvips, xdvi, etc.) to use.
- Make the package deal correctly with float material (maybe hard to do).
- Check the compatibility of the package with other packages.