Merciadri packages: An overview

Luca Merciadri

1 Introduction

I have written four LaTeX 2$_\varepsilon$ packages: `bigints`, `dashundergaps`, `plantslabels`, and `turnthepage`. Because

◇ there are so many submitted packages, and

◇ TeX information sources are so heavily used that articles rapidly go to the archives,

their respective releases could have gone unnoticed by the majority of the community. I here propose a brief summary about what they do, and how they could be useful for your work. This idea came to my mind when reading Nicola Talbot's article [7]. I don't have the delusion that they will be useful to everybody all the time, but, except `plantslabels`, it's reasonable to think they might come in handy at least, *i.e.* for a TeX author to want to write a bigger integral than the one which is proposed, to dot or dash some text, or to indicate that the page needs to be turned.

I shall now describe them in alphabetical order, though I wrote fewer packages than Nicola!

The reader might notice that the text here is essentially extracted from their respective manuals: [2, 3, 4], where more detailed information and a bibliography of each is available. I am open to any suggestions, or remarks, concerning my packages.

After their respective descriptions, we will take a look at another tip: how to use a brace so that some elements of a given matrix are selected. We will here present the case where the brace is above the matrix.

2 The `bigints` package

The `bigints` package (`v1.1`) helps you to write *big integrals* when needed. When making a report for a course, during the 2009–2010 academic year, I realized that there was no satisfactory implementation of 'big integrals' in LaTeX 2$_\varepsilon$. (If there are, please let me know.) My idea, during the report, was to write a big integral in front of an $n \times 1$ (n rows, 1 column) matrix, to signify the integration of every element on the n rows. (For the doubtful, I asked mathematicians, and, yes, this is sometimes written like this, though it is not very standard.) I also thought about writing the integral in front of a big expression, which might be different from a matrix.

2.1 Example

Consider, for example, a rocket which is propelled in space thanks to motors, giving a thrust by mass unity ($\alpha > 0$) which is supposed constant. Using Mechanics' laws, the time which is necessary to go from Earth's surface ($r = R$) to an orbit of height $2R$ ($r = 3R$) is given by

$$T = \int_R^{3R} \frac{\mathrm{d}r}{\sqrt{2\alpha(r-R) + 2\mu\left(\frac{1}{r} - \frac{1}{R}\right)}}, \qquad (1)$$

where μ is a positive constant, associated to the gravitational force.

In Equation (1), the integral sign is too small. Consider now

$$T = \boxed{\displaystyle\int_R^{3R} \frac{\mathrm{d}r}{\sqrt{2\alpha(r-R) + 2\mu\left(\frac{1}{r} - \frac{1}{R}\right)}}}, \text{ or}$$

$$T = \int_R^{3R} \frac{\mathrm{d}r}{\sqrt{2\alpha(r-R) + 2\mu\left(\frac{1}{r} - \frac{1}{R}\right)}}.$$

They look reasonably better, simply because the integral sign's height is related to the integrand's height. (For skeptics: even without `displaystyle` in front of the opening parenthesis, the integral sign of the original expression is still too small.)

Creating integral signs which are adapted to their argument (the integrand) was the idea for the package, and this is what gave rise to this package.

2.2 Available commands

The available commands and their output are shown in Table 1.

Put briefly, I defined `\bigint`, `\bigints`, and so on, with their respective `o`-counterparts (for line integrals along a closed curve, for example). The only rule to keep in mind is that the more you add '`s`' to the integral command, the smaller the integral sign is. To use these functions, you simply need to load the `bigints` package.

3 The `dashundergaps` package

The `dashundergaps` package (`v1.2`) helps you to use a pattern or patterns from this list:

- dashing,
- dotting,
- underlining

for a word which can be either hidden, or not.

This can be useful in these situations:

1. You are writing a document for which you need to dash or (and) to dot text,

Command	Std. output	Command output
`\bigint`, `\bigoint`	$\int$, $\oint$	$\bigint$, $\bigoint$
`\bigints`, `\bigoints`	$\int$, $\oint$	$\bigints$, $\bigoints$
`\bigintss`, `\bigointss`	$\int$, $\oint$	$\bigintss$, $\bigointss$
`\bigintsss`, `\bigointsss`	$\int$, $\oint$	$\bigintsss$, $\bigointsss$
`\bigintssss`, `\bigointssss`	$\int$, $\oint$	$\bigintssss$, $\bigointssss$

Table 1: Commands in the `bigints` package.

2. You want to write a test for which students have to “fill in the gaps”, and you want to choose when to print the answers.

3.1 Examples

Here is an example of sentence dashing.

```
\documentclass[10pt]{article}
\usepackage[dash]{dashundergaps}
\begin{document}
\dashuline{This is a dashed sentence}
\end{document}
```

gives

This is a dashed sentence

Dotting is done in the same way.

And an example of dotted gaps for a student version (notice that gaps are always numbered):

```
\documentclass[10pt]{article}
\usepackage[dot, phantomtext]{dashundergaps}
\begin{document}
In Artificial Intelligence, ‘‘RL’’ means
‘‘Reinforcement \gap{Learning}.’’
\end{document}
```

results in

In Artificial Intelligence, “RL” means “Reinforcement .... (1).”

Despite its rather ugly appearance, several people asked me how to achieve something like

Head A	Head B	Head C
....		
Col 1	Col 2	Col 3
Col 1	Col 2	Col 3

For this, use:

```
 \begin{tabular}{lll}
\hline
Head A & Head B & Head C\\
\multicolumn{2}{l}{\dotuline{\hfill}}\\
Col 1 & Col 2 & Col 3\\
Col 1 & Col 2 & Col 3\\
\hline
\end{tabular}
```

where you want it to appear.

3.2 Available commands

Option(s)	**Consequence**		
	`\gap{text}`	`\dashuline{text}`	`\dotuline{text}`
`dash` (only)	×	text	×
`dot` (only)	×	×	text
`dash`, `dot`	×	text	text
`phantomtext` (only)	____ (1)	×	×
`phantomtext`, `dash`	- - (1)	text	×
`phantomtext`, `dot`	. . . (1)	×	text
`phantomtext`, `dash`, `dot`	____ (1)	text	text
`phantomtext`, `teachernotes`	**text**	×	×
`phantomtext`, `dash`, `teachernotes`	**text**	text	×
`phantomtext`, `dot`, `teachernotes`	**text**	×	text
`phantomtext`, `dash`, `dot`, `teachernotes`	**text**	text	text

Table 2: Possible calls of the `dashundergaps` package.

Without any option, the package will not do anything useful. Consequently, one of the following options should be specified:

- `dash`: will dash text if used with the command `\dashuline{text}` where you want "text" to be dashed (*i.e.* somewhere in the `document` environment).
- `dot`: will dot text if used with the command `\dotuline{text}` where you want "text" to be dotted.
- `phantomtext`: helps in writing a pattern at the place of the text. This pattern can be
  - dashing, if used with `dash` option;
  - dotting, if used with `dot` option;
  - underlining, if used with (`dash` *and* `dot`) options *or* with neither `dash` nor `dot`;
  - the text itself, if used with the `teachernotes` option.
- `teachernotes`: see the last above.
- `displaynbgaps`: produces, at the end of your document (in the center of the page), a summary of the number of gaps, like this:

GAPS: x.

All the commands (their order is immaterial) of `dashundergaps.sty` are given in Table 2 *except* the use of `displaynbgaps`, which can trivially be used iff `phantomtext` is used. Here, "×" means "not applicable".

3.3 Sectioning and dashundergaps

3.3.1 Numbering

Some users would like to have dashed or dotted section numbers. This can be done with, for example:

```
\usepackage[dash,dot]{dashundergaps}
\usepackage[calcwidth,pagestyles,...]{titlesec}
...
\titleformat{\section}
  {\normalfont\Huge\bfseries}
  {\dashuline{\thesection}}{1em}{}
\titleformat{\subsection}
  {\normalfont\LARGE\bfseries}
  {\dotuline{\thesubsection}}{1em}{}
```

It is possible, and will work. For example, here, sections and subsections will have their numbering respectively dashed and dotted.

3.3.2 Titles

However, this approach using `titlesec` does not work for the section titles. To do this, the present solution is to use, in each section command, code like this:

```
\section{\protect\dashuline{My section}}
```

At present, the `\protect` is required.

4 The plantslabels package

I have a somewhat wide-ranging collection of carnivorous plants. Once objects are categorized (be they plants, or anything else), it is useful to distinguish them easily. This is easy between plants whose characteristics are radically different, as is normally the case with distant species (as defined in biology). But once one has many objects sharing the same characteristics (which, here, occurs more frequently for plants belonging to the same species), it becomes more difficult not to mix up two plants. Or some plants belonging to the same species could have different reactions towards natural elements such as cold, etc., and it is thus interesting to distinguish them, which leads to the idea of labelling. This is what motivated me to write this package, which aims at making labels for plants, as its name suggests.

4.1 Example

Let's say that you have two kinds of plants that you want to label: "Myplant1" and "Myplant2." One habitually lives in the desert, and the other lives in tropical regions. You have, say, 2 specimens of the first, and 4 of the second. You can invoke, assuming `cactus.eps` is your image for the first one, that you have no image for the second one, and that they respect the conditions mentioned below:

```
\plant{1}{1}{2}{Myplant1}{5}{EUR}
{$-10\to +50$}{Peat moss, sand, %
perlite}{cactus.eps}
\plant{2}{2}{4}{Myplant2}{10}{EUR}
{$20\to +40$}{Peat moss, fertilizer}{}
```

This will create 2+4 = 6 labels (2 of the first, 4 of the second). The two labels are represented (without the captions, images and a slightly smaller length for the label) in Figure 1.

Name	*Myplant1*
Price	5 EUR
Temperature	$-10 \to +50$
Substratum	Peat moss, sand, perlite

(a) Label of *Myplant1*.

Name	*Myplant2*
Price	10 EUR
Temperature	$20 \to +40$
Substratum	Peat moss, fertilizer

(b) Label of *Myplant2*.

Figure 1: The two kinds of labels produced.

4.2 Available commands

This package has currently no option, and is rather limited, as it has only one command: `\plant`. This command takes nine arguments, of which only the first three are mandatory. Here is the syntax:

```
\plant
  {⟨cols_labels⟩}{⟨rows_labels⟩}{⟨no_labels⟩}
  {⟨generic_plant_name⟩}
  {⟨generic_price⟩}
  {⟨generic_currency⟩}
  {$⟨generic_temperature⟩$}
  {⟨generic_substratum⟩}
  {⟨generic_picture⟩}
```

where

1. (mandatory) ⟨*cols_labels*⟩ is the number of columns of labels,
2. (mandatory) ⟨*rows_labels*⟩ is the number of rows of labels,
3. (mandatory) ⟨*no_labels*⟩ is the number of labels, under the condition

   $$\langle \mathit{cols\_labels}\rangle \times \langle \mathit{rows\_labels}\rangle = \langle \mathit{no\_labels}\rangle,$$

4. ⟨*generic_plant_name*⟩ is the plant name to be written on each label,
5. ⟨*generic_price*⟩ is the plant's price to be written on each label,
6. ⟨*generic_currency*⟩ is the currency to be written on each label, after ⟨*generic_price*⟩,
7. ⟨*generic_temperature*⟩ is the temperature to be written on each label (it should be $t_{\min} \to t_{\max}$, *i.e.* the minimum and maximum temperatures for the plant),
8. ⟨*generic_substratum*⟩ is the plant's substratum to be written on each label,
9. ⟨*generic_picture*⟩ is the plant's picture to be drawn on each label.

As all the arguments after ⟨*no_labels*⟩ are optional, you can skip them. However, you still need to write empty braces for them. For example,

```
\plant{1}{1}{1}{Plant}{}{}{}{}{}
```

will simply draw one label with "Plant" on it.

5 The `turnthepage` package

On some exams, some teachers like to indicate to the reader that the 'page' can be turned. This avoids oversights by students, thereby avoiding them breaking out in a cold sweat at the end of an exam, realizing there were other questions they needed to answer. We have implemented a LaTeX package [5] so that each odd page (except the last one, if the last page is odd) displays a message to turn the page. The mechanism may also be useful for different kinds of documents.

This package (`v1.3`) can thus be used for exams, or special documents printed '`twoside`.'

5.1 Available commands

To load the package, use

```
\usepackage[⟨option⟩]{turnthepage}
```

where ⟨*option*⟩ determines the text that appears at the bottom of each odd page, in its right corner:

- `short`: displays '/...';
- `english`: displays '*Turn the page.*';
- `francais`: displays '*Tournez la page.*';
- `nederlands`: displays '*Sla de pagina om.*';
- `deutsch`: displays '*Bitte wenden.*'.

As the package calls `\turnthepage` where it needs to be placed, you can define `\turnthepage` as you want. For example, if you want a more polite way to say it, you can use

```
\renewcommand{\turnthepage}{%
\itshape Turn the page please.}
```

after the preamble.

The package is composed of three `.sty` files: one for the ε-TeX version (with the `zref-abspage` and `zref-lastpage` packages), one for the non-ε-TeX version (with the `pagesLTS` package), and one main file. The main file (`turnthepage.sty`) will automatically load `turnpageetex.sty` if ε-TeX is found, and `turnpagewoetex.sty` if no ε-TeX is detected.

You can force the use of `turnpageetex.sty` or `turnpagewoetex.sty` (provided you have the related LaTeX installation) using

```
\usepackage[option]{turnpageetex}
```

or

```
\usepackage[option]{turnpagewoetex}
```

where, in both cases, `option` is one of the aforementioned options. Notice the missing `the` after `turn` in these two specific files.

5.2 Thanks

I could not have achieved this work without some help. Special thanks go to

- Marc van Dongen for his help [8],
- Martin Münch for his personal communications and his version of `turnthepage` without ε-TeX,
- Philipp Stephani [6].

6 Matrices with borders

The code in this section comes from [1]; I thought it was valuable enough to describe. For pedagogical reasons, one may want matrices with borders, like this:

$$\left(\begin{array}{c|ccc} a & \multicolumn{3}{c}{\overbrace{b \quad \ldots \quad c}^{n \text{ times}}} \\ \hline d & & & \\ \vdots & & A & \\ e & & & \end{array}\right)$$

This can be achieved with a variety of approaches. First, here is what produced the above example:

```
\usepackage{multirow}
\makeatletter
\def\Biggg#1{{\hbox{$\left#1\vbox to32\p@{}
                     \right.\n@space$}}}
\newdimen\bracketwidth
\settowidth{\bracketwidth}{\Biggg(}
\makeatother
\[ \begin{array}{r@{}r@{\hspace{\arraycolsep}}rcc%
    c@{\hspace{\arraycolsep}}c@{}l}
    & & & & \multicolumn{3}{c}{\vspace{-.5em}
      \overbrace{\hphantom{b \hspace{2\arraycolsep}
      \cdots
      \hspace{2\arraycolsep} c}}
      ^{n \mbox{\scriptsize\ times}}}\\
    \multirow{5}{\bracketwidth}[3pt]{\Biggg(}
    & & a & \multirow{5}{1pt}[3pt]{\vrule height52pt}
    & b & \cdots & c &
    \multirow{5}{\bracketwidth}[3pt]{\Biggg)}\\
    \cline{2-7}
    & & d\\
    & & \vdots & & & A\\
    & & e
\end{array} \]
```

This is close to unreadable for me. A more readable solution:

```
\[\vbox{%
  \hskip2.8em$\overbrace{\hphantom{b %
  \hspace{2\arraycolsep} \cdots
    \hspace{2\arraycolsep} c}}^{n%
    \mbox{\scriptsize\ times}}$
  \vskip-.25em
  $\left(
  \begin{array}{r|ccc}
    a & b & \cdots & c\\
    \hline
    d\\
    \vdots & & A\\
    e
  \end{array} \right)$} \]
```

In another approach, one could define:

```
\def\moverbrace#1#2{%
 \newdimen\moverbracewd
 \settowidth\moverbracewd{#1}%
 \addtolength\moverbracewd{-2\arraycolsep}
 \vbox to 1.6ex{\hsize=\moverbracewd
  \centering\vss
  $\overbrace{#1}^{#2}$}}
```

and then use it, for example like this:

```
\[ \left(
 \begin{array}{r|c}
  a & \moverbrace{b \hspace{2\arraycolsep}
       \cdots \hspace{2\arraycolsep} c}
       {n \mbox{\scriptsize\ times}}\\
    \hline
    d\\
    \vdots & A\\
    e
 \end{array}\right) \]
```

In all the examples, \mbox could evidently be replaced by a \text equivalent. Please tell me if you know a simpler way to achieve this.

◇ Luca Merciadri
University of Liège
Luca.Merciadri (at) student dot ulg dot ac dot be
http://www.student.montefiore.ulg.ac.be/~merciadri/

References

[1] Glad Deschrijver. TEX Tricks, 2011. http://users.ugent.be/~gdschrij/LaTeX/textricks.html.

[2] Luca Merciadri. The bigints package, 2010. http://mirror.ctan.org/macros/latex/contrib/bigints.

[3] Luca Merciadri. The dashundergaps package, 2010. http://mirror.ctan.org/macros/latex/contrib/dashundergaps.

[4] Luca Merciadri. The plantslabels package, 2010. http://mirror.ctan.org/macros/latex/contrib/plantslabels.

[5] Luca Merciadri, Marc Van Dongen, and Martin Münch. The turnthepage package, 2011. http://mirror.ctan.org/macros/latex/contrib/turnthepage.

[6] Philipp Stephani and Luca Merciadri. New on CTAN: turnthepage, comp.text.tex, 2010.

[7] Nicola Talbot. Talbot packages: An overview. *TUGboat*, 31:65–67, 2010.

[8] Marc Van Dongen and Luca Merciadri. Turn the Page, comp.text.tex, 2011.