

The spacingtricks package*

Antoine Missier
antoine.missier@ac-toulouse.fr

July 28, 2023

1 Introduction

This package offers several macros to address spacing issues:

- `\centered` offers horizontal centering without introducing vertical spacing.
- `\footnote` has been redefined to avoid unsuitable spacing in certain cases.
- `\vstrut` generates a strut with variable height or depth for maintaining consistent vertical spacing.
- `\indent` has been redefined to indent a line at the beginning of a particular paragraph even if `\parindent` has been set to 0.
- The `indentblock` environment allows indentation of all its content.
- The `compactlist` environment creates a compact list, as shown here, without vertical spacing between items. Additionally, aliases are provided for shorter **list symbols**: `\bul`, `\dash`, `\ddash`, `\aster`, `\hand`, `\checksymb`, `\arrowsymb`.
- Common abbreviations, *i.e.* and *e.g.*, are typeset using the macros `\ie` and `\eg` ensuring correct spacing.
- The `\dualboxes` command facilitates placing two boxes (figures, tables, text) side by side by adjusting the vertical positioning.

Three other “spacing” packages are loaded by `spacingtricks`: `setspace` [1] (natively in $\text{\LaTeX 2}\mathcal{E}$), used for setting line spacing in a piece of text, `xspace` [2], which adds an interword space unless the macro is followed by a punctuation character, and `centeredline` [3] which offers an interesting alternative to our `\centered` macro (see further). We also took over the fine `juxtapose` environment from the `mafr` [4] distribution, as another way to place boxes side by side.

Furthermore, we have developed the package `arraycols` [5], which allows for effective management of spacing in `tabular` or `array` environments, and `mismath` [6] providing several macros to improve spacing in mathematical formulas.

2 Usage

`\centered` The `\centered{\text}` command yields a centered line without vertical spacing. It

*This document corresponds to `spacingtricks v1.7`, dated 2023/07/28.

functions similarly to `\centerline`, except in lists or tables where its behavior is notably improved (see the following examples). Additionally, the line break before (but not after) the macro is applied automatically.

Here is a comparative example of the centering commands inside a list:

1. Here is a centered line using `\centered`:
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Here is another centered line using `\\ \centerline`:
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
3. Here is another centered line using `\par\centerline`:
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
4. Here is a centered line using the `center` environment:
Lorem ipsum dolor sit amet, consectetur adipiscing elit.

After using `\centered{...}`, the end of line must be explicit, otherwise `\centered` will not function correctly. You can use `\\`, which is equivalent to `\linebreak`, or you can use a blank line to begin a new paragraph (equivalent to `\par`). In the former case, no vertical space is added after the centered line, while in the latter case, the vertical space following the centered line is slightly larger. In the example provided, the centered line comes to an end, without line breaking because of the next `\item` command, and the vertical space is determined by the `enumerate` environment.

In tables, `\centered` allows you to center a specific line in a cell independently of the column alignment¹, Here's an example using `\begin{tabular}{|l|r|}`:

left-aligned content	right-aligned content
the second cell determines the width	another centered content
a centered cell	the last cell determines the width

`\centeredline` A limitation of our `\centered` macro is that it doesn't allow the use of `\verb` commands inside its argument. However, you have the `\centeredline` command from the package `centeredline` by Jean-François Burnol [3] to address this limitation. This small package is now loaded by `spacingtricks`. One other advantage of `\centeredline` is that it does not require an explicit end; the text can follow the command on the same line, and the line break will be automatic. Nevertheless, it doesn't work in tables (like in the example above), except when used with the `p` column declaration.

`onehalfspace`
`doublespace`
`spacing` The `setspace` package [1], which is loaded by `spacingtricks`, provides environments for changing the spacing between lines. The current paragraph is nested in a `onehalfspace` environment, as you can see. The `spacing` environment can be used to achieve other interline spacing, for example `\begin{spacing}{2.5}`.

`\footnote` The `\footnote` command doesn't handle spacing issues well, especially when the

¹In tables, we can also utilize the powerful `\makecell` command from the `makecell` package [8]. However, neither the `\centerline` nor `\centering` commands work for a single line in a cell.

hyperref package has been loaded. In English tradition, there is no space before numbers (or symbols) of note calls, and similarly at the beginning of footnotes, the text begins immediately after the note number². To avoid undesirable spaces, you don't have to leave any space before writing `\footnote{`, for instance:

There is no space here`\footnote{The old command}` before the macro. However, it may be convenient to place the `\footnote` command on a new line. To achieve this, `\footnote` has been redefined to completely eliminate unwanted spaces³.

<pre>This is a note \footnote{ The old command.} with bad spacing. This is a note^a with bad spacing.</pre>	<pre>This is a note \footnote{ The new command.} with good spacing. This is a note^a with good spacing.</pre>
<hr style="width: 20%; margin: auto;"/> ^a The old command.	<hr style="width: 20%; margin: auto;"/> ^a The new command.

`\footnotespace`
`\footnoteindent`

Like the old `\footnote` command, the new one can take also an optional argument to force the number of the note. Additionally, we have always had customization macros such as `\footnotesize`, `\footnotesep`, `\footnoterule`. However, two new macros have been added to manage spacing: `\footnotespace` which produces the space before the note call symbol, and `\footnoteindent` which produces the space at the beginning of the footnote text. For instance, with `\renewcommand{\footnotespace}{\,}` and `\renewcommand{\footnoteindent}{\enskip}`⁴ we get:

This is a note^a with particular space settings.

^a en dash spacing at the beginning of the note.

`\vstrut`

The command `\vstrut[⟨depth⟩]{⟨height⟩}` produces a strut with a variable height or depth, allowing you to increase the line's height above the baseline or its depth below (optional). This command can be used in various contexts, such as a text line, a table, a list, a formula, and more. If the values of `⟨height⟩` and `⟨depth⟩` are smaller than the height and depth of the current line, the command has no effect. Here are some examples:

$$\left[\frac{\sqrt{0.5p}}{10}\right]_{0.5p} = \frac{\sqrt{0.5p}}{10} = \frac{\sqrt{0.5p}}{10}$$

`\fbox{\vstrut{2ex}$\sigma(X)=1$}` gives $\sigma(X) = 1$ better than $\sigma(X) = 1$.

Adjusting the height using `\vstrut` is done by trial and error. While a vertical phantom box would provide a good result, for example `\vphantom{\bar{t}}` in the previous square root, it's not obvious to know what to put in the phantom box. Moreover, `\vstrut` allows a finer tuning.

²The typesetting of footnotes and note calls depends on national typographic rules which are, in principle, managed by the `babel` package. For instance, when activating the `french` option of `babel`, a thin space is added before the note calls, and the new `\footnote` macro does not alter this behavior.

³Indeed, recommending the use of the `%` symbol at the end of a line would suffice (but not in `dtx` files); its effect is to cancel the space produced by a line break, but it's not always something that comes to mind immediately. Furthermore let's note that our definition of `\footnote` will have no effect when using the `\VerbatimFootnotes` command from the `fancyvrb` package by Timothy Van Zandt [7].

⁴`\enskip` is equivalent to `\hspace{0.5em}`.

In a table, using `\renewcommand{\arraystretch}{\stretch}` allows you to increase the height of all rows, but it has a global effect, whereas `\vstrut` allows to adjust the height of each row individually, as in the following table:

bad		good	
$\lim_{\substack{x \rightarrow 1 \\ x > 1}} \ln \left(\frac{x^2}{x-1} \right)$	$\frac{a}{b}$	$\lim_{\substack{x \rightarrow 1 \\ x > 1}} \ln \left(\frac{x^2}{x-1} \right)$	obtained with <code>\vstrut{3.8ex}</code>
$\frac{a}{b}$	$\int_1^x \frac{1}{t} dt$	$\frac{a}{b}$	<code>\vstrut[2ex]{3ex}</code>
$\int_1^x \frac{1}{t} dt$		$\int_1^x \frac{1}{t} dt$	<code>\vstrut[2.5ex]{4.2ex}</code>

However, for tables, we have the `arraycols` package [5], which is based on `cellspace` [9] and it allows to automatically adjust row heights. Nevertheless, `\vstrut` can be useful for fine adjustments.

In a text line `\vstrut` can be used to increase line space, and is better than `\vspace`.

`\indent` The command `\setlength{\parindent}{0cm}` eliminates indentation at the beginning of every paragraph. However, in this case, the `\indent` command does no longer works if you want to apply exceptional indentation to a specific paragraph. To address this issue, the initial length of `\parindent` is saved in `\parindentlength`, and the command `\indent` is redefined to allow indentation of the length `\parindentlength`.

`indentblock` The `indentblock` environment facilitates the indentation of an entire block of lines. It has an optional argument which sets the length of indentation (by default it uses `\parindentlength`). The lyrics below have been indented (and typeset in italic shape) using `\begin{indentblock}\itshape` and stanzas 2 and 4 have undergone additional indentation using `\begin{indentblock}[3em]`.

*Overhead the albatross hangs motionless upon the air
And deep beneath the rolling waves in labyrinths of coral caves
The echo of a distant time comes willowing across the sand
And everything is green and submarine*

*And no one showed us to the land
And no one knows the where's or why's
But something stirs and something tries
Starts to climb towards the light*

*Strangers passing in the street
By chance two separate glances meet
And I am you and what I see is me
And do I take you by the hand
And lead you through the land
And help me understand the best I can?*

*And no one calls us to move on
And no one forces down our eyes
No one speaks and no one tries
No one flies around the sun*

`compactlist` As the name suggests, the `compactlist` environment allows you to create a “compact” list, meaning it has no vertical space above or between items. Similar to regular lists in \LaTeX , items are generated using the `\item` command. The environment also accepts an optional argument: `\begin{compactlist}[\langle symbol \rangle]`.

`\bul` The default item symbol in the `compactlist` environment is `\textbullet`, but it can
`\dash` be changed. We have provided aliases for several symbols commonly used in lists:
`\ddash` `\bul •` (alias for `\textbullet`), `\dash –` (alias for `\textendash`), `\ddash —` (alias
`\aster` for `\textemdash`), `\aster *` (alias for `\textasteriskcentered`).

`[pifont]` Some other common symbols can be found in the `pifont` package [10]. Therefore
`\hand` the `spacingtricks` package includes a `pifont` option that loads this package and creates
`\checksymb` the appropriate aliases: `\hand ☞` for `\ding{43}`, `\checksymb ✓` for `\ding{51}` and
`\arrowsymb` `\arrowsymb ➤` for `\ding{226}`.

The following example is produced by `\begin{compactlist}[\checksymb]`:

- ✓ First item.
- ✓ Second item.
- ✓ Third item.

☞ These aliases can also be used directly in some text. The symbol is followed by a space if there is no punctuation character immediately after it (thanks to the macro `xspace` from the `xspace` package [2]). Use `\langle symbol \rangle\unskip` if you want to remove this space.

`\compactlistindent` Indentation length is set by `\compactlistindent` (fixed at 0.5 em by default) and can be modified with `\setlength`. Notice that there are several other ways to create a compact list, particularly by using the `noitemsep` key of the `enumitem` package [11].

`dinglist` The `pifont` package has also a list environment `\begin{dinglist}\langle number \rangle` to typeset a list with normal spacing, using the `\ding{\langle number \rangle}` symbol for each `\item` (instead of using `\item[\ding{\langle number \rangle}]`). So `\begin{dinglist}\{226\}` yields:

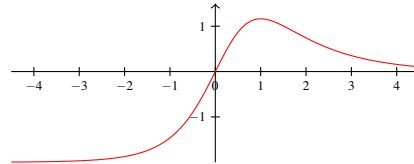
- ✓ The first item in the list.
- ✓ The second item in the list.
- ✓ The third item in the list.

`\ie` In English, at the end of a sentence, the period is followed by an em space which is
`\eg` larger than an inter-word space. To achieve correct spacing after the abbreviations ‘i.e.’ (*id est*) and ‘e.g.’ (*exempli gratia*), e.g. here, we provide the `\ie` and `\eg` macros, suggested in The \LaTeX Companion [18]. In American typography, a comma is often placed after these abbreviations, which can be achieved using `\ie`, on the other hand, some authors prefer to typeset *i.e.* in italic shape, which is always possible with `\textit{\ie}`.

`\dualboxes` Several packages aim to typeset text around a figure or a table, but some of them require providing the width of the box containing the figure or the table. Here we provide another approach with the macro `\dualboxes[\langle pos \rangle]\langle left \rangle\langle right \rangle`, which places two boxes, `\langle left \rangle` and `\langle right \rangle`, side by side. These boxes can contain figures, tables, a small line of text, `parbox` text, `minipage` environments (for several paragraphs and lists), etc. The optional `\langle pos \rangle` parameter sets the vertical level on which the boxes are aligned, and it takes a value between 0 (bottom) and 1 (top, default value).

Here is a first example in which we have adjusted the position of the boxes vertically using `\dualboxes[0.6]`.

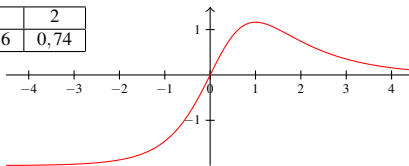
x	-2	-1	0	1	2
$f(x)$	-1,87	-1,46	0	1,16	0,74



The horizontal space is equally shared between left margin, inter-box space, and right margin.

In the following example, boxes are aligned by top, and the right box has been shifted back (left) from 3 cm to create an overlapping effect with the left one, which cannot be achieved using `picins`.

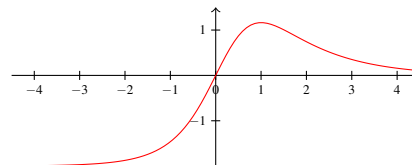
x	-2	-1	0	1	2
$f(x)$	-1,87	-1,46	0	1,16	0,74



```
\dualboxes{\scriptsize
  $\begin{array}{|*{6}{c}|} \hline
    x & -2 & -1 & 0 & 1 & 2 \\ \hline
    f(x) & -1,87 & -1,46 & 0 & 1,16 & 0,74 \\ \hline
  \end{array}$
}{\mbox{} \hspace{-3cm}
\begin{tikzpicture}[scale=0.6]\tiny
  \draw[->] (-4.5,0) -- (4.5,0);
  \draw[->] (0,-2) -- (0,1.5);
  \foreach \x in {-4,...,4} {
    \draw (\x,0.1cm) -- (\x,-0.1cm) node[below] {$\x$};}
  \foreach \y in {-1,1} {
    \draw (0.1cm,\y) -- (-0.1cm,\y) node[left] {$\y$};}
  \draw[domain=-4.5:4.5,samples=100,color=red]
    plot ({\x},{2*\x/(exp(\x)-\x)});
\end{tikzpicture}
}
```

`\dualboxes*` Indeed this command has a starred version that eliminates space before the first box and after the second one, but not between them: `\dualboxes* [⟨pos⟩]{⟨left⟩}{⟨right⟩}`.

In practice, the `\dualboxes` macro is suitable for boxes in which you put figures, tables, or small one-line text. However, for longer text, it must be nested in a paragraph box, e.g. here we used `\parbox{6.5cm}{...}`. If you need to include several paragraphs, a list, or a mathematical displayed formula, you should use a `minipage` environment.



Let us mention that, unfortunately, it is not possible to use `verbatim` environments (nor the in-line `\verb` command) within the `\dualboxes` arguments. Additionally, using footnotes or margin notes within `\dualboxes` is also not possible.

For arranging two figures side by side, or a table and a figure, `\dualboxes` is perfect. However, for wrapping text around a figure, some other package may be more efficient. Let us first mention `wrapfig` [12], one of the easiest, but it requires the text to be contained in a single paragraph (like above), without list. Otherwise you have to place it in a `minipage` environment.

More sophisticated is the `picins` package [13], cited in *The L^AT_EX Companion* [18]. Unlike most others, it can be used with lists; however vertical positioning can be tricky, and it is not included in common distributions `TEX Live` or `MiKTEX`. Another package worth mentioning is `picinpar` [14], in which text can begin after the first line and flow to both sides of the figure. It allows several paragraphs, but it does not support lists or verbatim text.

`juxtapose`
`\juxtopskip`
`\juxbottomskip`
`\juxsepspace`

As a very interesting alternative, we present the powerful `juxtapose` environment, originally written by Christian Obrecht in the `fiche.cls` class file, which is part of the `mafr` distribution [4]. We have made small improvements, including ensuring proper functionality and allowing customization of certain lengths. By default, the top and bottom lengths are set to `\smallskipamount`, and the separation space is set to 1.5 em. If the default lengths do not fit well for spacing above or below the environment, you can add a `\bigskip` or a `\vspace{⟨dimen⟩}` for example, or, as a last resort, you can redefine the lengths `\juxtopskip` and `\juxbottomskip`.

In this example the `juxtapose` environment begins here. It can handle various elements, including figures, tables, as well as:

- verbatim text: `\mytestmacro`,
- text with several paragraphs,
- lists, like here (a compact list),
- displayed mathematics as shown on the right side,

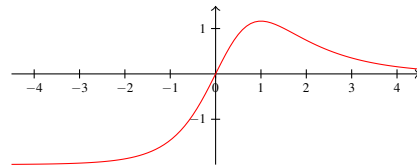
without the need for `multpage` environments.

The two boxes are aligned by their center, but unlike `\dualboxes`, you cannot adjust vertical positioning directly. However, you can use `\vspace` inside a box to alter vertical positioning and achieve the desired layout.

Here begins the following text. The syntax of `juxtapose` is simple:

```
\begin{juxtapose} ... \otherside ... \end{juxtapose}
```

It has an optional argument `\begin{juxtapose}[⟨lefratio⟩]` where `⟨lefratio⟩` is the width ratio of the left side relative to `\textwidth` (the default value is 0.5 which means `0.5\textwidth`).



$$f(x) = \frac{2x}{e^x - x}$$

This is the last line of the environment.

Paragraphs are indented inside each side, except the first one. For a figure, there's no need to shift it to the right (from indent length) to prevent users from forgetting to use `\noindent` before the figure. If you still want to indent the first paragraph, you can use `\indent`. For horizontal alignment inside a box, you can use `\centering` or `\flushright`, particularly useful for figures. For a global indentation of the first (or the second) box, you enclose it in an `indentblock` environment, and will work perfectly.

One may ask why not using multi-column environments instead. In the classic multicol environment, you cannot choose horizontally unbalanced columns, and you cannot control where to transition to the other column.

Some packages improve multicol. For instance, `vwcol` [15] allows fixed column widths. However, it still has limitations, such as not accepting verbatim text and not handling lists and figures correctly (as `juxtapose` did it above).

Let's mention `paracol` [16] and `reledpar` [17], both designed for typesetting two columns of text in parallel, making them useful for translation or critical editions of text. They work fine with lists, figures, verbatim text, footnotes and margin notes. However, achieving precise vertical positioning of two boxes is not straightforward. While they offer synchronization mechanisms, convenient for text, they may not be ideal for placing two figures side by side. Additionally, they are a bit over-sized for our specific need. For our purposes, `juxtapose` or `\dualboxes` may be more suitable options.

3 Implementation

```

1 \newif\ifspacingtricks@pifont
2 \DeclareOption{pifont}{\spacingtricks@pifonttrue}
3 \ProcessOptions \relax
4
5 \RequirePackage{ifthen}
6 \RequirePackage{calc}
7 \RequirePackage{setspace}
8 \RequirePackage{xspace}
9 \RequirePackage{centeredline}
10 \ifspacingtricks@pifont \RequirePackage{pifont} \fi
11
12 \newcommand*{\centered}[1]{\setlength{\parskip}{0pt}\par\noindent\hfill
13     #1\hfill\mbox{}}

```

The double braces are necessary here to ensure that the `\parskip` modification applies locally within the command and not globally to the rest of the document.

```

14 \newcommand{\footnotespace}{}
15 \newcommand{\footnoteindent}{}
16 \let\footnt\footnote
17 \renewcommand{\footnote}[2][\unskip\footnotespace%
18     \ifthenelse{equal{#1}}{\footnoteindent\ignorespaces #2}
19     ]{\unskip\footnt[#1]{\footnoteindent\ignorespaces #2}
20     }{\unskip
21     }
22

```



```

23 }
\unskip eliminates undesirable spaces before and \ignorespaces after.
24 \newlength{\strutheight}
25 \newcommand*\vstrut}[2][0pt]{%
26   \setlength{\strutheight}{#2}%
27   \addtolength{\strutheight}{#1}%
28   \unskip
29   \ensuremath{\rule[-#1]{0pt}{\strutheight}}%
30   \ignorespaces%
31 }
32
33 \newlength{\parindentlength}
34 \setlength{\parindentlength}{\parindent}
35 \renewcommand{\indent}{\hspace{\parindentlength}}
36
37 \newenvironment*{indentblock}[1][\parindentlength]{
38   \begin{list}{}{
39     \setlength{\leftmargin}{#1}
40     \setlength{\itemsep}{0pt}
41     \setlength{\topsep}{0pt} % previously 1ex
42     \setlength{\partopsep}{0pt}
43   }
44   \item[]
45 }\end{list}}
46
47 \newlength{\compactlistindent}
48 \setlength{\compactlistindent}{0.5em}
49 \newenvironment*{compactlist}[1][\textbullet]{
50   \par % sometimes necessary
51   \begin{list}{#1\unskip}{% \unskip gobbles space created by \xspace
52     \setlength{\itemsep}{0pt}
53     \setlength{\parsep}{0pt}
54     \setlength{\topsep}{0ex}
55     \setlength{\partopsep}{0pt}
56     \setlength{\labelwidth}{1em}
57     \setlength{\leftmargin}{\labelwidth}
58     \addtolength{\leftmargin}{\labelsep}
59     \addtolength{\leftmargin}{\compactlistindent}
60   }
61 }\end{list}}
62
63 \providecommand{\bul}{\textbullet\xspace}
64 \providecommand{\dash}{\textendash\xspace}
65 \providecommand{\ddash}{\textemdash\xspace}
66 \providecommand{\aster}{\textasteriskcentered\xspace}
The command \asterisk already exists in the mathabx package (for math mode only).
The following macros require the pifont package.
67 \ifspacingtricks@pifont
68   \providecommand{\hand}{\ding{43}\xspace}

```

```

69 \providecommand{\checksymb}{\ding{51}\xspace}
70 \providecommand{\arrowsymb}{\ding{226}\xspace}
71 \fi
72
73 \providecommand{\ie}{i. e. \@xspace}
74 \providecommand{\eg}{e. g. \@xspace}
75
76 \newcommand{\@@dualboxes}[3][1]{
77   \par\noindent
78   \raisebox{\depth-#1\totalheight}{#2} \hfill % needs calc
79   \raisebox{\depth-#1\totalheight}{#3} \smallskip
80 }
81 \newcommand{\@dualboxes}[3][1]{
82   \par\noindent \hfill
83   \raisebox{\depth-#1\totalheight}{#2} \hfill
84   \raisebox{\depth-#1\totalheight}{#3} \hfill\mbox{} \smallskip
85 }
86 \newcommand{\dualboxes}{\@ifstar{\@@dualboxes}{\@dualboxes}}
87

```

The following macro was found in the `fiche.cls` file, distributed with the `mafr` package by Christian Obrecht [4]. It had some minor bugs in spacing. I made small changes, including the addition of lengths parameters and the `\linewidth` instruction (without which, lists in the first side would send the second side out of the page).

```

88 \newlength\juxtopskip
89 \setlength\juxtopskip{\smallskipamount}
90 \newlength\juxbottomskip
91 \setlength\juxbottomskip{\smallskipamount}
92 \newlength\juxsepspace
93 \setlength\juxsepspace{1.5em}
94 \newlength\side@width
95
96 \newenvironment{juxtapose}[1][0.5]{% left side ratio
97   \def\otherside{\egroup % the internal function remembers \side@width
98     \@tempdima=\side@width
99     \side@width=\textwidth
100    \advance\side@width by -\@tempdima
101    \advance\side@width by -\juxsepspace
102    \hskip\juxsepspace
103    \linewidth=\side@width % for lists environments
104    \vcenter\bgroup\hsize=\side@width\noindent % \ignorespaces useless
105   }
106   \side@width=#1\textwidth
107   \advance\side@width by -0.5\juxsepspace
108   \linewidth=\side@width % for lists environments
109   \par\vskip\juxtopskip%
110   \hbox\bgroup\begin{math}
111   \vcenter\bgroup\hsize=\side@width\noindent\ignorespaces
112   }\{\egroup\end{math}\egroup\vskip\juxbottomskip}
113

```

References

- [1] *The setspace Package*, Robin Fairbairns, Geoffrey Tobin, CTAN, v9.7b 2022/12/04.
- [2] *The xspace package*, David Carlisle, Morten Høgholm, CTAN v1.13 2014/10/28.
- [3] *centeredline – A macro for centering lines*, Jean-François Burnol, CTAN, v1.2 2022/10/10.
- [4] *La distribution mafr*, Christian Obrecht, CTAN, v1.0 17/09/2006.
- [5] *The arraycols package*, Antoine Missier, CTAN, v1.2 2021/09/20.
- [6] *mismath – Miscellaneous mathematical macros*, Antoine Missier, CTAN, v2.5 2023/02/24.
- [7] *The ‘fancyvrb’ package – Fancy Verbatims in \LaTeX* , Timothy Van Zandt, CTAN, v4.5a 2023/01/19.
- [8] *The makecell package*, Olga Lapko, CTAN, v0.1e 2009/08/03.
- [9] *The cellspace package*, Josselin Noirel, CTAN, v1.9.0 2022/01/04.
- [10] *pifont – Using common PostScript fonts with \LaTeX* , Walter Schmidt, CTAN, v9.3 2020/03/25.
- [11] *Customizing lists with the enumitem package*, Javier Bezos, CTAN, v3.9 2019/06/20.
- [12] *The wrapfig package*, Donald Arseneau, CTAN, v3.6 2003/01/31.
- [13] *Bilder in \LaTeX -Dokumenten – PicIns-Benutzerhandbuch*, Joachim Bleser, Edmund Lang, CTAN, v3.0 sept. 1992.
- [14] *The picinpar package*, Friedhelm Sowa, Stefan Blochwitz, CTAN, v1.3 2022/11/25.
- [15] *The vwcol package*, Will Robertson, CTAN, v0.2 2015/02/10.
- [16] *Package paracol: Yet Another Multi-Column Package to Typeset Columns in Parallel*, Hiroshi Nakashima, CTAN, v1.35 2018/12/31.
- [17] *Parallel typesetting for critical editions: the reledpar package*, Maïeul Rouquette, Peter Wilson, Herries Press, CTAN, 2.25.3 2022/04/22.
- [18] *The \LaTeX Companion*. Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley, 2nd edition, Pearson Education, 2004.