The spacingtricks package*

Antoine Missier
antoine.missier@ac-toulouse.fr

March 5, 2023

1 Introduction

This package offers a few macros to deal with spacing issues. Thus:

- `\centered` yields good horizontal centering without vertical spacing;
- `\footnote` has been redefined to avoid unsuitable spacing;
- `\vstrut` produces a strut with variable height or depth;
- `\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 produces indentation of all its content;
- the `compactlist` environment yields a compact list, without vertical spacing between the items, like here; several aliases are provided to type some list symbols shorter: `\bul`, `\dash`, `\ddash`, `\aster`, `\hand`, `\checksymb`, `\arrowsymb`;
- common abbreviations i.e. and e.g. are typeset by the macros `\ie` and `\eg` with correct spacing;
- the `\dualboxes` command attends to place 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 LaTeX 2ε), for setting line spacing in a piece of text, xspace [2], which adds an inter-word space unless the macro is followed by a punctuation character, and centeredline [3] which gives 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.

Otherwise, we have develop the package arraycols [5], that allows a good management of spacing in `tabular` or `array` environments, and mismath [6] that provides 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 v1.6, dated 2023/03/05.

acts like \centerline except in lists or tables where its behavior is much better (see the following examples). Moreover, the line break before (but not after) the macro is automatic.

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

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

   Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

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

In tables, \centered allows to center a particular line in a cell independently of the column alignment[1], here with \begin{tabular}{|l|r|}:

left aligned column	right aligned column
the second cell fixes the width	centered cell
centered cell	the last cell fixes the width

\centeredline

A limitation of our \centered macro is that it doesn't allow to use \verb commands inside its argument. For this purpose, you get the \centeredline command, from the package centeredline by Jean-François Burnol [3]. This small package is now loaded by spacingtricks. Another 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 with p column declaration.

onehalfspace
doublespace
spacing

The setspace package [1], loaded by spacingtricks, provides environments for changing the spacing between lines. The present paragraph is nested in a onhalfspace environment as you can see. The spacing environment can be used to get other interline spacing, e.g. \begin{spacing}{2.5}.

\footnote

The \footnote command doesn't have a good management of spacing issues, in par-

[1] In tables, we can also use the powerful \makecell command of the makecell package [7]; on the other hand, neither \centerline nor \centering commands do work for a single line in a cell.

ticular when the hyperref package has been loaded. In English tradition, there is no space before numbers (or symbols) of note calls, and likewise at the beginning of footnotes, text begins immediately after the note number[2]. To avoid undesirable spaces, we don't have to put some space before writing \footnote{, for instance:

```
Here is no space\footnote{The old command} before the macro.
```

but sometimes, it is convenient to place the \footnote command on a new line. To achieve this, \footnote has been redefined to completely eliminate unwanted spaces[3].

```
This is a note
\footnote{
The old command.}
with bad spacing.
```

This is a note [a] with bad spacing.

[a] The old command.

```
This is a note
\footnote{
The new command.}
with good spacing.
```

This is a note[a] with good spacing.

[a] The new command.

\footnotespace
\footenoteindent

Like the old one, the new \footnote command can take an optional argument to force the number of the note. Likewise, we have always customization macros \footnotesize, \footnotesep, \footnoterule, but two new macros have been added to manage spacing: \footnotespace produces the space before the note call symbol and \footnoteindent 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

\vstrut[⟨*depth*⟩]{⟨*height*⟩} produces a strut with variable height or depth, in order to increase the line's height (above the base line) or depth (below the base line, optional); this command can be used in a text line, a table, a list, a formula, etc. If the values of ⟨*height*⟩ and ⟨*depth*⟩ are inferior to the height and depth of the current line, the command has no effect. Here are some examples.

```
\[\frac{\sqrt{0.5p}}{10} =
\frac{\sqrt{\vstrut{1.7ex} 0.5p}}{10}\]
```

$$\frac{\sqrt{0.5p}}{10} = \frac{\sqrt{0.5p}}{10}$$

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

The height adjustment is done by trial and error. We could also have used a vertical phantom box; for example in the previous square root, we get a good result with \vphantom{\bar{t}}, but it's not obvious to know what to put in the phantom box, moreover, \vstrut allows a finer tuning.

[2] The typesetting of footnotes and note calls depends on national typographic rules which are, in principle, managed by babel. For instance, by 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.

[3] Probably it would have been enough to recommend the use of the % symbol at the end of line; its effect is to cancel the space produced by a line break, but we do not always think of using it.

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

In a table, `\renewcommand{\arraystretch}{⟨stretch⟩}` allows to increase the height of the rows but this command has a global effect, whereas `\vstrut` allows to adjust properly the height of each row, as in the following table:

bad
$\lim\limits_{\substack{x\to1\\x>1}} \ln\left(\dfrac{x^2}{x-1}\right)$
$\dfrac{a}{b}$
$\displaystyle\int_1^x \frac{1}{t}\,\mathrm{d}t$

good	
$\lim\limits_{\substack{x\to1\\x>1}} \ln\left(\dfrac{x^2}{x-1}\right)$	obtained with `\vstrut{3.8ex}`
$\dfrac{a}{b}$	`\vstrut[2ex]{3ex}`
$\displaystyle\int_1^x \frac{1}{t}\,\mathrm{d}t$	`\vstrut[2.5ex]{4.2ex}`

However, for tables, we have the arraycols package [5], based on cellspace [8], which allows to adjust row heights automatically. Nevertheless, `\vstrut` can be useful for fine adjustments.

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

`\indent` The command `\setlength{\parindent}{0cm}` allow to eliminate any indentation of lines at the beginning of every paragraph. But in this case, the `\indent` command does not work anymore if we want exceptional indentation of a particular paragraph. So, the `\parindentlength` initial length of `\parindent` has been saved in `\parindentlength` and the command `\indent` has been redefined to still allow indentation of length `\parindentlength`.

`indentblock` The `indentblock` environment allows indentation of a whole block of lines. It has an optional argument which is the length of indentation (set by default to `\parindentlength`). The following lyrics have been indented (and typeset in italic shape) with `\begin{indentblock}\itshape` and stanzas 2 and 4 have been affected by an additional indentation with `\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 its name tells it, the `compactlist` environment allows to create a "compact" list, i.e. without vertical space neither above nor between items. As for lists in LaTeX, items are generated by the `\item` command. The environment has an optional argument: `\begin{compactlist}[`⟨*symbol*⟩`]`.

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

`[pifont]` `\hand` `\checksymb` `\arrowsymb` Some other common symbols can be found in the pifont package [9]. Therefore spacingtricks has a `pifont` option, that loads this package and creates the appropriate aliases: `\hand` ☞ for `\ding{43}`, `\checksymb` ✓ for `\ding{51}` and `\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 just after it (thanks to the macro `xspace` from the xspace package [2]). Use `\`⟨*symbol*⟩`\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 construct a compact list in particular with the `noitemsep` key of the enumitem package [10].

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

✓ The first item in the list.

✓ The second item in the list.

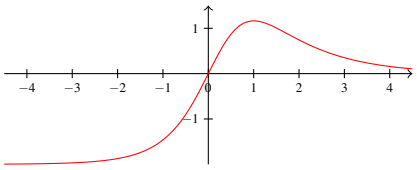
✓ The third item in the list.

`\ie` `\eg` In English, at the end of a sentence, the point is followed by an em space which is larger than an inter-word space. We provide the `\ie` (*id est*) and `\eg` (*exempli gratia*) macros, suggested in The LaTeX Companion [17], to get correct spacing after these abbreviations e.g. here. In American typography, a comma is often placed after these abbreviations, what we can get with `\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 intend to typeset text around a figure or a table, but some of them need to give the width of the box containing the figure or the table. We provide here another approach with the macro `\dualboxes[`⟨*pos*⟩`]{`⟨*left*⟩`}{`⟨*right*⟩`}`, which places two boxes, ⟨*left*⟩ and ⟨*right*⟩, side by side. These boxes can contain figures, tables, one small line text, parbox text, `minipage` environments (for several paragraphs and lists), etc. The optional ⟨*pos*⟩ parameter sets the vertical level on which the boxes are aligned: a number between 0 (bottom) and 1 (top, default value).

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

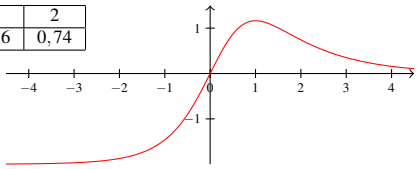
x	-2	-1	0	1	2
$f(x)$	$-1,87$	$-1,46$	0	1,16	0,74



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 make an overlapping with the left one, what picins cannot do.

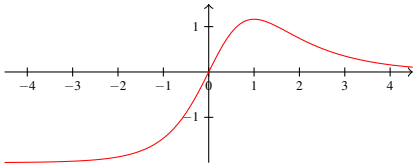
x	-2	-1	0	1	2
$f(x)$	$-1,87$	$-1,46$	0	1,16	0,74




```
\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*` 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, `\dualboxes` is suitable for boxes in which you put figures or tables or small one line text. For longer text, this one must be nested in a paragraph box, e.g. here we used `\parbox{6.5cm}{...}`. If you want several paragraphs, a list or a mathematical displayed formula, then you have to put them in a `minipage` environment.



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

For putting two figures side by side, or a table and a figure, `\dualboxes` is perfect. But for text around a figure, some other package may be more efficient. Les us mention first wrapfig [11], one of the easiest, but text must contain in a single paragraph (like above), without list or you have to place it in a `minipage` environment.

More sophisticated is the picins package [12], cited in The LaTeX Companion [17]. Unlike most others, it can be used with lists, nevertheless vertical positioning can be tricky and it doesn't belong to common distributions TeX Live or MiKTeX. A last one we mention is picinpar [13] in which text can begin after the first line and can flow to the two side of the figure; it allows several paragraphs but neither lists, nor verbatim text.

`juxtapose`

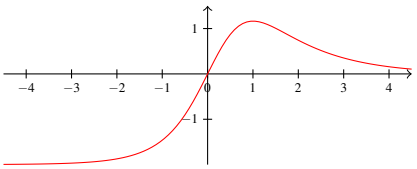
`\juxtopskip`
`\juxbottomskip`
`\juxsepspace`

As a very interesting alternative, we present the powerful `juxtapose` environment, written by Christian Obrecht in the `fiche.cls class` file, which is part of the mafr distribution [4]. We made small improvements, among others to work correctly with lists and get customization of some lengths: top and bottom lengths are set by default to `\smallskipamount` and separation space is set to 1.5 em. For spacing above or below the environment, if the default length doesn't fit well, you can add a `\bigskip` or a `\vspace{`⟨*dimen*⟩`}` for example, or at worst redefine the length `\juxtopskip` and `\juxbottomskip`.

In this example the environment begins here. `juxtapose` can deal with figures, tables, but also:

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

without the need of `multipage` environments.



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

This is the last line of the environment.

The two boxes are aligned by their center, but unlike `\dualboxes`, you cannot adjust vertical positioning, except using `\vspace`.

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

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

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

Paragraphs are indented inside each side, except the first one, because for a figure, there's no need to shift it to the right (from indent length) and otherwise the user may forgot to put `\noindent` before the figure. If you still want to indent the first paragraph use `\indent`. For horizontal alignment inside a box, you can use `\centering` or

`\flushright` (useful for a figure). If you want a global indentation of the first (or the second) box, you can put it in an `indentblock` environment, it works fine!

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

Some nice packages improve multicols, e.g. vwcol [14] in which you can fix on the columns width, but unfortunately it doesn't accept verbatim text, and doesn't manage correctly lists and figures like above.

Let us mention paracol [15] and reledpar [16], which are intended to typeset two column of text in parallel, useful for translation or critical edition of text. They work fine with lists, figures, verbatim text, and also footnotes and margin notes, but it is not immediate to obtain just a chosen vertical positioning of two boxes. They have synchronization mechanisms, convenient for text, but not when you place two figures side by side. Besides they are a bit oversized for our need. They don't have the same purpose. So we can hope `juxtapose` or `\dualboxes` are useful.

3 Implementation

```
\newif\ifspacingtricks@pifont
\DeclareOption{pifont}{\spacingtricks@pifonttrue}
\ProcessOptions \relax

\RequirePackage{ifthen}
\RequirePackage{calc}
\RequirePackage{setspace}
\RequirePackage{xspace}
\RequirePackage{centeredline}
\ifspacingtricks@pifont \RequirePackage{pifont} \fi

\newcommand*{\centered}[1]{{\setlength{\parskip}{0pt}\par\noindent\hfill
        #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.

```
\newcommand{\footnotespace}{}
\newcommand{\footnoteindent}{}
\let\footnt\footnote
\renewcommand{\footnote}[2][]{\unskip\footnotespace%
    \ifthenelse{\equal{#1}{}}{
        \unskip\footnt{\footnoteindent\ignorespaces #2}
    }{
        \unskip\footnt[#1]{\footnoteindent\ignorespaces #2}
    }\unskip
}
```

`\unskip` eliminates undesirable spaces before and `\ignorespaces` after.

```
\newlength{\strutheight}
```

```
\newcommand*{\vstrut}[2][0pt]{%
    \setlength{\strutheight}{#2}%
    \addtolength{\strutheight}{#1}%
    \unskip
    \ensuremath{\rule[-#1]{0pt}{\strutheight}}%
    \ignorespaces%
}

\newlength{\parindentlength}
\setlength{\parindentlength}{\parindent}
\renewcommand{\indent}{\hspace{\parindentlength}}

\newenvironment*{indentblock}[1][\parindentlength]{
    \begin{list}{}{%
        \setlength{\leftmargin}{#1}
        \setlength{\itemsep}{0pt}
        \setlength{\topsep}{0pt} % previously 1ex
        \setlength{\partopsep}{0pt}
    }
    \item[]
}{\end{list}}

\newlength{\compactlistindent}
\setlength{\compactlistindent}{0.5em}
\newenvironment*{compactlist}[1][\textbullet]{
    \par % sometimes necessary
    \begin{list}{#1\unskip}{% \unskip gobbles space created by \xspace
        \setlength{\itemsep}{0pt}
        \setlength{\parsep}{0pt}
        \setlength{\topsep}{0ex}
        \setlength{\partopsep}{0pt}
        \setlength{\labelwidth}{1em}
        \setlength{\leftmargin}{\labelwidth}
        \addtolength{\leftmargin}{\labelsep}
        \addtolength{\leftmargin}{\compactlistindent}
    }
}{\end{list}}

\providecommand{\bul}{\textbullet\xspace}
\providecommand{\dash}{\textendash\xspace}
\providecommand{\ddash}{\textemdash\xspace}
\providecommand{\aster}{\textasteriskcentered\xspace}
```

The command `\asterisk` already exists in the mathabx package (for math mode only). The following macros need the pifont package.

```
\ifspacingtricks@pifont
    \providecommand{\hand}{\ding{43}\xspace}
    \providecommand{\checksymb}{\ding{51}\xspace}
    \providecommand{\arrowsymb}{\ding{226}\xspace}
\fi
```

```
\providecommand{\ie}{i.e.\@\xspace}
\providecommand{\eg}{e.g.\@\xspace}

\newcommand{\@@dualboxes}[3][1]{
    \par\noindent
    \raisebox{\depth-#1\totalheight}{#2} \hfill % needs calc
    \raisebox{\depth-#1\totalheight}{#3} \smallskip
}
\newcommand{\@dualboxes}[3][1]{
    \par\noindent \hfill
    \raisebox{\depth-#1\totalheight}{#2} \hfill
    \raisebox{\depth-#1\totalheight}{#3} \hfill\mbox{}\smallskip
}
\newcommand{\dualboxes}{\@ifstar{\@@dualboxes}{\@dualboxes}}

```

The following macro has been found in the `fiche.cls` file, distributed with the mafr package by Christian Obrecht [4]. It had some tiny bugs in spacing. I just made small changes: adding some lengths parameters and the `\linewidth` instruction (without whom, lists in the first side send the second side out of the page).

```
\newlength\juxtopskip
\setlength\juxtopskip{\smallskipamount}
\newlength\juxbottomskip
\setlength\juxbottomskip{\smallskipamount}
\newlength\juxsepspace
\setlength\juxsepspace{1.5em}
\newlength\side@width

\newenvironment{juxtapose}[1][0.5]{% left side ratio
    \def\otherside{\egroup % internal function remembers \side@width
        \@tempdima=\side@width
        \side@width=\textwidth
        \advance\side@width by -\@tempdima
        \advance\side@width by -\juxsepspace
        \hskip\juxsepspace
        \linewidth=\side@width % for lists environments
        \vcenter\bgroup\hsize=\side@width\noindent % \ignorespaces useless
        }
    \side@width=#1\textwidth
    \advance\side@width by -0.5\juxsepspace
    \linewidth=\side@width % for lists environments
    \par\vskip\juxtopskip%
    \hbox\bgroup\begin{math}
    \vcenter\bgroup\hsize=\side@width\noindent\ignorespaces
    }{\egroup\end{math}\egroup\vskip\juxbottomskip}

```

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 makecell package*, Olga Lapko, CTAN, v0.1e 2009/08/03.

[8] *The cellspace package*, Josselin Noirel, CTAN, v1.9.0 2022/01/04.

[9] *pifont – Using common PostScript fonts with LaTeX*, Walter Schmidt, CTAN, v9.3 2020/03/25.

[10] *Customizing lists with the enumitem package*, Javier Bezos, CTAN, v3.9 2019/06/20.

[11] *The wrapfig package*, Donald Arseneau, CTAN, v3.6 2003/01/31.

[12] *Bilder in LaTeX-Dokumenten – PicIns-Benutzerhandbuch*, Joachim Bleser, Edmund Lang, CTAN, v3.0 sept. 1992.

[13] *The picinpar package*, Friedhelm Sowa, Stefan Blochwitz, CTAN, v1.3 2022/11/25.

[14] *The vwcol package*, Will Robertson, CTAN, v0.2 2015/02/10.

[15] *Package paracol: Yet Another Multi-Column Package to Typeset Columns in Parallel*, Hiroshi Nakashima, CTAN, v1.35 2018/12/31.

[16] *Parallel typesetting for critical editions: the reledpar package*, Maïeul Rouquette, Peter Wilson, Herries Press, CTAN, 2.25.3 2022/04/22.

[17] *The LaTeX Companion*. Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley, 2nd edition, Pearson Education, 2004.