

The spacingtricks package*

Antoine Missier
antoine.missier@ac-toulouse.fr

August 2, 2026

1 Introduction

This package offers several macros or environments 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, and `\cstrut` being symmetric around the mathematical axis.
- `\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.
- The `\juxtapose` environment does the same thing but adjusts the horizontal positioning.

Two other “spacing” packages are loaded by `spacingtricks`: `xspace` [1], which adds an inter-word space unless the macro is followed by a punctuation character, useful for abbreviations like *i.e.* and *e.g.*, and `centeredline` [2] which offers an interesting alternative to our `\centered` macro (see further). Since version 1.9, the package `setspace` [3], natively in $\text{\LaTeX 2}_{\mathcal{E}}$, used for setting line spacing in a piece of text, isn’t loaded anymore, because it is rarely used, but it is still documented here. Anyone can load it if necessary.

Furthermore, we have developed the package `arraycols` [4], which allows for effective management of spacing in `tabular` or `array` environments, and `mismath` [5] providing several macros to improve spacing in mathematical formulas.

2 Usage

`\centered` The `\centered{<text>}` command yields a centered line without vertical spacing. It works similarly to `\centerline`, except in lists or tables where its behavior is notably

*This document corresponds to `spacingtricks` v1.9, dated 2026/08/02.

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 [2] 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 (env.)` The `setspace` package [3] provides environments for changing the spacing between
`doublespace (env.)` lines. The current paragraph is nested in a `onehalfspace` environment, as you can see.
`spacing (env.)` The `spacing` environment can be used to achieve other interline spacing, for example `\begin{spacing}{2.5}`. As this feature is rarely used, the `setspace` package is not loaded by `spacingtricks` anymore, but it is worth knowing.

`\footnote` The `\footnote` command present some spacing issues, especially when the `hyperref` package has been loaded. In English typography, there is no space before numbers (or

¹In tables, we can also utilize the powerful `\makecell` command from the `makecell` package [7]. However, neither the `\centerline` nor `\centering` commands work for a single line in a cell.

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.

This is a note <code>\footnote{</code> The old command.} with bad spacing. This is a note^a with bad spacing.	This is a note <code>\footnote{</code> The new command.} with good spacing. This is a note^a with good spacing.
--	--

^aThe old command.

^aThe 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:

$$\begin{aligned} & \backslash[\frac{\sqrt{0.5p}}{10}] & \frac{\sqrt{0.5p}}{10} = \frac{\sqrt{0.5p}}{10} \\ & = \backslashfrac{\sqrt{\vstrut{1.7ex} 0.5p}}{10} \backslash \end{aligned}$$

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

`\cstrut`

The `\cstrut` macro is an alternative of `\vstrut` that adds a symmetric strut in height and depth relative to the mathematical center line, which is halfway the height of an 'x'. It takes a single parameter `\cstrut{⟨height⟩}`. It's useful in `array` environments as it maintains the mathematical axis at the center of the cell with `⟨height⟩` above and under.

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

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 our 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 [6].

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

bad	good	
$\lim_{\substack{x \rightarrow 1 \\ x > 1}} \ln \left(\frac{x^2}{x-1} \right)$	$\lim_{\substack{x \rightarrow 1 \\ x > 1}} \ln \left(\frac{x^2}{x-1} \right)$	obtained with <code>\cstrut{3.8ex}</code>
$\frac{a}{\frac{b}{\int_1^X \frac{1}{t} dt}}$	$\frac{a}{\frac{b}{\int_1^X \frac{1}{t} dt}}$	<code>\cstrut{2.5ex}</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 [4], which is based on `cellspace` [8] and it allows to automatically adjust row heights. Nevertheless, `\cstrut` and `\vstrut` can be useful for fine adjustments.

`\indent` The command `\setlength{\parindent}{0cm}` eliminates indentation at the beginning of every paragraph. However, in this case, the `\indent` command does no longer work 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 (env.)` 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 (env.)` 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 $\text{\LaTeX}$, 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 (opt.)` Some other common symbols can be found in the pifont package [9]. 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 [1]. 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 create
a compact list, particularly by using the `noitemsep` key of the `enumitem` package [10].

`dinglist (env.)` 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}{55}` yields:

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

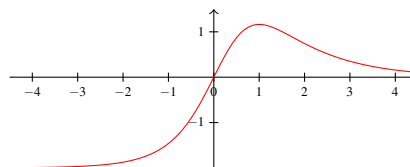
`\ie` In English, at the end of a sentence, the period is followed by an em space which
`\eg` is larger than an inter-word space. To achieve correct spacing after the abbreviations
‘i.e.’ (*id est*) and ‘e.g.’ (*exempli gratia*), e.g. here, we provide the `\ie` and `\eg` macros,
suggested in The L^AT_EX Companion [19]. A comma can be placed after the abbreviations,
which is encountered in American typography. Although it is not mandatory, some prefer
to typeset them in italics, which is always possible with `\textit{\ie}`⁵.

`\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[\<pos>]{\<left>}{\<right>}`, which places
two boxes, `\<left>` and `\<right>`, 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 `\<pos>` parameter sets the vertical level on which the boxes are aligned,
and it takes a value between 0 (bottom) and 1 (top, default value). The macro works also
inside a list item.

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

⁵The interesting foreign package [11] provides also `\ie` and `\eg` and many others like `\etal` for ‘et al.’ and also generic macros to create any chosen abbreviation. Moreover it has macros for main common Latin expressions which must normally be written in italics except when they are considered as standard English because they are very common. If `foreign` is loaded before `spacingtricks` there will be no conflict.

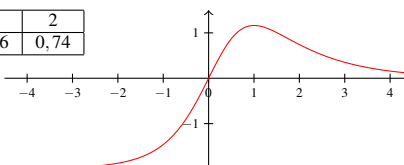
x	-2	-1	0	1	2
$f(x)$	-1,87	-1,46	0	1,16	0,74



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

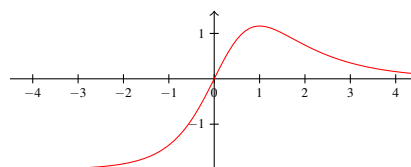
In the following example, boxes are aligned by top, and the right box has been shifted back (left) from 3 cm to create an overlapping effect with the left one.

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*[\langle pos \rangle]{\langle left \rangle}{\langle right \rangle}`. 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 or the following `juxtapose` 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.

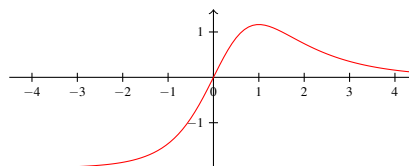
More sophisticated is the `picins` package [13], cited in The $\text{\LaTeX}$ Companion [19]. Unlike most others, it can be used with lists; however vertical positioning can be tricky, and it is not included in common distributions $\text{\TeX}$ Live or $\text{\MiKTeX}$. Another package worth mentioning 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 (env.)`
`\juxtopskip`
`\juxbottomskip`
`\juxsepspace`

An interesting alternative, is provided here with the `juxtapose` environment, which is based on two `minipage` environments (since version 1.9) placed side by side. The environment allows customization of certain lengths: at top, bottom and box separation. 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{\langle dimen \rangle}` for example, or 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,
- footnotes.



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

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

This is the last line of the environment.

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

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

`\langle leftratio \rangle (opt.)` It has an optional argument `\begin{juxtapose}[\langle leftratio \rangle]` where `\langle leftratio \rangle` is the width ratio of the left side relative to `\linewidth` (the default value is 0.5 which means `0.5\linewidth`).

As is the case with `minipage` environments, paragraphs are not indented inside each side, but, unlike `minipage`, you *can* indent paragraph with the `\indent` command. 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. Note that the environment works also perfectly inside a list item.

One may ask why not using multi-column environments instead. In the classic `multicol` environment, you cannot choose horizontally unbalanced columns.

Some packages improve multicolumn. 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{xspace}
8 \RequirePackage{centeredline}
9 \ifspacingtricks@pifont \RequirePackage{pifont} \fi
10
11 \newcommand*{\centered}[1]{\setlength{\parskip}{0pt}\par\noindent\hfill
12     #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.

```

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

```

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

```

23 \newlength{\strutheight}
24 \newcommand*{\vstrut}[2][0pt]{%
25     \setlength{\strutheight}{#2}%
26     \addtolength{\strutheight}{#1}%
27     \unskip%
28     \rule[-#1]{0pt}{\strutheight}%
29     \ignorespaces%
30 }
31
32 \newlength{\strutdepth}
33 \newcommand*{\cstrut}[1]{%
34     \setlength{\strutdepth}{#1}%

```

```

35 \setlength\strutheight{2\strutdepth}% total height
36 \addtolength{\strutdepth}{-0.5ex}%
37 \unskip%
38 \rule[-\strutdepth]{0pt}{\strutheight}%
39 \ignorespaces%
40 }
41
42 \newlength{\parindentlength}
43 \setlength{\parindentlength}{\parindent}
44 \renewcommand{\indent}{\hspace{\parindentlength}}
45
46 \newenvironment*{indentblock}[1][\parindentlength]{
47   \begin{list}{}{%
48     \setlength{\leftmargin}{#1}
49     \setlength{\itemsep}{0pt}
50     \setlength{\topsep}{0pt} % previously 1ex
51     \setlength{\partopsep}{0pt}
52   }
53   \item[]
54 }{\end{list}}
55
56 \newlength{\compactlistindent}
57 \setlength{\compactlistindent}{0.5em}
58 \newenvironment*{compactlist}[1][\textbullet]{
59   \par % sometimes necessary
60   \begin{list}{#1\unskip}{% \unskip gobbles space created by \xspace
61     \setlength{\itemsep}{0pt}
62     \setlength{\parsep}{0pt}
63     \setlength{\topsep}{0ex}
64     \setlength{\partopsep}{0pt}
65     \setlength{\labelwidth}{1em}
66     \setlength{\leftmargin}{\labelwidth}
67     \addtolength{\leftmargin}{\labelsep}
68     \addtolength{\leftmargin}{\compactlistindent}
69   }
70 }{\end{list}}
71
72 \providecommand{\bul}{\textbullet\xspace}
73 \providecommand{\dash}{\textendash\xspace}
74 \providecommand{\ddash}{\textemdash\xspace}
75 \providecommand{\aster}{\textasteriskcentered\xspace}

```

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

The following macros require the `pifont` package.

```

76 \ifspacingtricks@pifont
77   \providecommand{\hand}{\ding{43}\xspace}
78   \providecommand{\checksymb}{\ding{51}\xspace}
79   \providecommand{\arrowsymb}{\ding{226}\xspace}
80 \fi
81
82 \providecommand{\ie}{i.e.\@\xspace}
83 \providecommand{\eg}{e.g.\@\xspace}
84
85 \newcommand{\@@dualboxes}[3][1]{

```

```

86 \par\noindent
87 \raisebox{\depth-#1\totalheight}{#2} \hfill % needs calc
88 \raisebox{\depth-#1\totalheight}{#3} \smallskip
89 }
90 \newcommand{\@dualboxes}[3][1]{
91 \par\noindent \hfill
92 \raisebox{\depth-#1\totalheight}{#2} \hfill
93 \raisebox{\depth-#1\totalheight}{#3} \hfill\mbox{}\smallskip
94 }
95 \newcommand{\dualboxes}{\@ifstar{\@dualboxes}{\@dualboxes}}
96
    Originally inspired from an environment with the same name written by Christian
    Obrecht in the fiche.cls file, part of the mafr distribution [18], which has been im-
    proved in previous versions of spacingtricks. Since version 1.9 we use the minipage
    environment which gives better result in paragraph justifying.
97 \newlength\juxtopskip
98 \setlength\juxtopskip{\smallskipamount}
99 \newlength\juxbottomskip
100 \setlength\juxbottomskip{\smallskipamount}
101 \newlength\juxsepspace
102 \setlength\juxsepspace{1.5em}
103 \newlength\side@width
104 \newlength\otherside@width
105
106 \newenvironment{juxtapose}[1][0.5]{% left side ratio
107 \def\otherside{\end{minipage}}%
108 \hskip\juxsepspace
109 \begin{minipage}{\otherside@width}
110 }
111 \side@width=#1\linewidth
112 \advance\side@width by -0.5\juxsepspace
113 \otherside@width=\linewidth
114 \advance\otherside@width by -\side@width
115 \advance\otherside@width by -\juxsepspace
116 \par\vskip\juxtopskip%
117 \noindent\begin{minipage}{\side@width}
118 }\end{minipage}\vskip\juxbottomskip}
119

```

References

- [1] *The xspace package*, David Carlisle, Morten Høgholm, CTAN v1.13 2014/10/28.
- [2] *centeredline – A macro for centering lines*, Jean-François Burnol, CTAN, v1.2 2022/10/10.
- [3] *The setspace Package*, Robin Fairbairns, Geoffrey Tobin, CTAN, v9.7b 2022/12/04.
- [4] *The arraycols package*, Antoine Missier, CTAN, v1.5 2024/05/04.
- [5] *mismath – Miscellaneous mathematical macros*, Antoine Missier, CTAN, v3.2 2025/10/08.

- [6] *The 'fancyvrb' package – Fancy Verbatims in L^AT_EX*, Timothy Van Zandt, CTAN, v4.5a 2023/01/19.
- [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 L^AT_EX*, 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 foreign package for L^AT_EX2_ε*, Philip G. Ratcliffe, CTAN, v2.7 2012/09/25.
- [12] *The wrapfig package*, Donald Arseneau, CTAN, v3.6 2003/01/31.
- [13] *Bilder in L^AT_EX-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] *La distribution mafr*, Christian Obrecht, CTAN, v1.0 17/09/2006.
- [19] *The L^AT_EX Companion*. Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley, 2nd edition, Pearson Education, 2004.