

# The `arraycols` package\*

Antoine Missier  
`antoine.missier@ac-toulouse.fr`

August 9, 2023

## 1 Introduction

This package introduces new predefined column types for tables, alongside the `array` package by Frank Mittelbach and David Carlisle [1]. It also includes a command for wide horizontal rule drawing. Below is a summary of the column types and macro defined by `arraycols`, with further to follow.

Column definitions	
<code>L</code>	Left adjusted column (applicable in LR mode for <code>array</code> environments or math mode for <code>tabular</code> environment)
<code>C</code>	Centered-adjusted column (similar to L but centered)
<code>R</code>	Right-adjusted column (similar to L but right-adjusted)
<code>t{width}</code>	Text column of fixed $\langle width \rangle$ (LR mode), similar to <code>p</code> , but with horizontal and vertical centering
<code>x</code>	Centered column in math mode with adjusted height to avoid touching the horizontal rules
<code>y</code>	Left-aligned column in math mode with adjusted height
<code>z{width}</code>	Centered column in math mode, similar to <code>x</code> , with adjusted height, but with fixed $\langle width \rangle$
<code>T</code>	Centered text column with adjusted width for <code>tabularx</code> environments (calculated like <code>X</code> column)
<code>Z</code>	Centered column for <code>tabularx</code> , similar to <code>T</code> , but in math mode with adjusted height, like <code>x</code> and <code>z</code>
<code>I</code>	Thick vertical rule (1 pt)
<code>V{thickness}</code>	Vertical rule with variable $\langle thickness \rangle$
Horizontal rules	
<code>\whline</code>	Wide horizontal rule (1 pt)

Note that if a column type has been previously defined, using `arraycols` will overwrite it and display a warning message.

---

\*This document corresponds to `arraycols` v1.3, dated 2023/08/09. Thanks to François Bastouil for assistance with the English translation.

In addition to loading the `array` package, `arraycols` also requires the `cellspace` package [2], which is necessary for the `x`, `y`, `z` and `Z` column types. Moreover it relies on `tabularx` [3] for `T` and `Z` column types and loads `makecell` [4] for creation of multilined tabular cells. It's worth mentioning that the `tablestyles` package [6] also defines `L`, `C`, `R` and `Z` column types differently. Nevertheless, `tablestyles` is incompatible with `makecell` and consequently with `arraycols` as well.

Despite its minimal code, this package does not aim to introduce new macros. Its main purpose is to combine and configure features derived from other packages.

## 2 Usage

**L** Referring to an example from the `array` package documentation, the `L`, `C` and `C` **R** columns types, offered by `arraycols`, provide the convenience of reversing the **R** mathematical mode. This allows to achieve centered, left-aligned or right-aligned LR-mode in an `array` environment or an equivalent math-mode in a `tabular` environment. For instance, using the declaration `\begin{tabular}{|l|C|r|}` sets the second column in centered mathematical mode. Similarly, using the declaration `\begin{array}{|L|c|c|}` sets the first column<sup>1</sup> in text mode, left-aligned.

`t{width}` The newly introduced column type definition `t{width}` (text in LR-mode) achieves both horizontal and vertical centering within the column. It is distinct from the traditional `p{width}` (in standard L<sup>A</sup>T<sub>E</sub>X) and `m{width}` (from the `array` package) column types, which are equivalent to `\parbox[t]` or `\parbox`.

**x** In order to guarantee adequate row heights, especially for displaymath mode **y** formulas, the package includes the column types `x` (centered) and `y` (left aligned), based on the `cellspace` package by Josselin Noirel [2]. These column types allow automatic adjustment of row heights to prevent any overlap with horizontal rules in cases where the content is too tall. While `cellspace` is initially designed for `tabular` environments, the new `x` and `y` column types, introduced by `arraycols`, generate mathematically well-adjusted columns. They are applicable in both `tabular` and `array` environments. Examine the following examples created using `\begin{array}{|c|}` and `\begin{array}{|x|}`.

bad	good
$\lim_{\substack{x \rightarrow 1 \\ x > 1}} \ln \left( \frac{x^2}{x-1} \right)$	$\lim_{\substack{x \rightarrow 1 \\ x > 1}} \ln \left( \frac{x^2}{x-1} \right)$
$\frac{a}{\bar{b}}$	$\frac{a}{\bar{b}}$
$\int_1^X \frac{1}{t} dt$	$\int_1^X \frac{1}{t} dt$

<sup>1</sup>The declarations `L`, `C`, `R` do not work in a `tabularx` environment. Additionally, the `tabulary` package by David Carlisle [5] already defines the `L`, `C`, `R` and `J` column types for specific alignments in tables of the same type as `tabularx`. However there is no incompatibility with `arraycols` because these column definitions apply exclusively in `tabulary` environments.

The `cellspace` package is loaded with the `math` option<sup>2</sup> to effectively manage row heights in matrix tables. Another option of `cellspace`, `column=Q` (with `S` being the default in `cellspace`)<sup>3</sup>, was necessary to prevent any compatibility issues with the `siunitx` package (also loaded by `pstricks-add`). The `Q` column type serves as a “modifier” that, when combined with the traditional column types `l`, `c`, `r`, permits the adjustment of cell height.

Notice that another package, `booktabs` [7], also offers excellent row height adjustment. However, regrettably, it doesn’t handle the height of vertical separators `|`. In order to achieve a similar vertical adjustment as `booktabs`, we configure the `cellspace` parameters as follows:

```
\setlength{\cellspacetoplimit}{3pt},
\setlength{\cellspacebottomlimit}{2pt}.
```

Additionally, it’s worth mentioning the `tbls` package by Donald Arneseau [8] that makes a good adjustment of row heights as well, but it is incompatible with the `array` and `numprint` packages.

Finally, manual adjustments can also be achieved using the `\vstrut` command from the `spacingtricks` package [10], or by utilizing `\gape` and `\Gape` from the `makecell` package [4], as well as employing `\bigstrut` from the `bigstrut` package [9].

`z{width}` The `z{width}` column type allows you to define the column width, just like `t{width}`, and it also activates math mode and adjusts the row height, similar to the `x` column type.

`T` The `tabularx` package by David Carlisle [3] introduces the `X` column def-  
`Z` inition, which calculates its width based on the required width for the entire table. It aligns text to the left akin to `p{width}`. For instance, using `\begin{tabularx}{8cm}{|c|X|X|}` adjusts the width of the `X` columns to achieve a total width of 8cm. To complement this, we offer the `T` declaration, which performs a similar function but centers the content horizontally. Additionally the `Z` declaration goes further by activating mathematical mode and adjusting line heights, comparable to `x` or `z`). Here’s an example with `\begin{tabularx}{\linewidth}{|T|y|x|Z|T|}`.

A good job	$\lim_{\substack{x \rightarrow 1 \\ x > 1}} \ln \left( \frac{x^2}{x-1} \right)$	$\frac{a}{b}$	$\frac{a}{b} + \int_1^X \frac{1}{t} dt$	a multi-line piece of text
------------	---	---------------	---	-------------------------------

In order to maintain precise alignment of fraction bars within mathematical formulas, cells are deliberately not vertically centered. Nevertheless, for achieving accurate vertical positioning within the last cell, we have used the powerful `\makecell` command from the `makecell` package by Olga Lapko [4]: `\makecell{a multi-line \\ piece of text}`.

<sup>2</sup>The `math` option loads the `amsmath` package. As mentioned in the `cellspace` package documentation: “the `amsmath` package can be loaded beforehand with other packages (such as `empheq` or `mathtools`), were an incompatibility to arise from one’s loading it later”.

<sup>3</sup>The selection of the letter `Q` to substitute the default column modifier `S` of the `cellspace` package is here arbitrary.

`I`        The column definition `I` is mentioned in The L<sup>A</sup>T<sub>E</sub>X Companion [11] and allows  
`V{thickness}` for drawing a thicker *vertical* line (1 pt thick) compared to the one achieved with  
the standard declaration `|`. For selecting the line thickness, we additionally provide  
the column definition `V{thickness}`<sup>4</sup>.

`\whline`        Similarly, the `\whline` command, suggested in The L<sup>A</sup>T<sub>E</sub>X Companion, al-  
lows for drawing a thicker *horizontal* line (1 pt thick) compared to the line ob-  
tained with `\hline`. Moreover, the `makecell` package provides the command  
`\Xhline{thickness}` enabling the choice of horizontal rule thickness.

The introductory table has been typeset with a column declaration `I` serving  
as a separator between the two text columns. Horizontal rules at the beginning  
and end of the table are accomplished using `\whline`, while a `\Xhline{0.8pt}`  
rule is employed after the legend rows. The formatting of header rows is  
achieved using the `\thead` command from the `makecell` package. By default,  
`arraycols` sets: `\renewcommand\theadfont{\footnotesize\sffamily}` (initially  
`\footnotesize` only, without `\sffamily`). Lastly, following a recommendation of  
the `array` package [1], an additional 1 pt has been added to the standard height of  
each row within this table. This adjustment is implemented with the command  
`\setlength{\extrarowheight}{1pt}`<sup>5</sup>.

### 3 Implementation

```

1 \RequirePackage{array}
2 \RequirePackage[math,column=Q]{cellspace}
3 \RequirePackage{tabularx} % must be loaded after cellspace
4 \RequirePackage{makecell}
5
6 \newcolumntype{C}{>{$}c<{$}}
7 \newcolumntype{L}{>{$}l<{$}}
8 \newcolumntype{R}{>{$}r<{$}}
9 \newcolumntype{t}[1]{>{\centering\arraybackslash}m{#1}}

```

The `cellspace` package offers the `S` modifier (we used `Q` instead), which, when  
placed before a column declaration, allows for the adjustment of cell content  
height to prevent any overlap with horizontal rules. The spacing between the  
content and the rules is governed by the parameters `\cellspacetoplimit` and  
`\cellspacebottomlimit`.

```

10 \newcolumntype{x}{>{$}Qc<{$}}
11 \newcolumntype{y}{>{$}Ql<{$}}
12 \setlength{\cellspacetoplimit}{3pt}
13 \setlength{\cellspacebottomlimit}{2pt}
14 \newcolumntype{z}[1]{>{$}Q{>{\centering\arraybackslash}p{#1}}<{$}}

```

<sup>4</sup>The definition of `V` would have been simplified by utilizing an optional argument for `I`, but  
unfortunately, this approach doesn't function.

<sup>5</sup>As stated in the `array` package documentation: "This is important for tables with horizontal  
lines because those lines normally touch the capital letters".

When defining the `z` column type, we employ the `p` column type instead of `m` (which centers content automatically). This choice ensures proper alignment of mathematical expressions within cells of the same row.

```
15 \newcolumnntype{T}{>{\centering\arraybackslash}X}
16 \newcolumnntype{Z}{>{\$}QT<{\$}}
```

The `T` columns are not centered automatically. Although it's possible to achieve this by using the command `\renewcommand{\tabularxcolumn}[1]{m{#1}}` (with `m` instead of default value `p`), unfortunately, this approach has a global effect on all column declarations based on `X`, including `T` and `Z`. As a result, it could disrupt the alignment of mathematical expressions within cells of the same row.

```
17 \newcolumnntype{I}{!\vrule width 1pt}
18 \newcolumnntype{V}[1]{!\vrule width #1}
19 \newlength\savedwidth
20 \newcommand{\whline}{%
21     \noalign{\global\savedwidth\arrayrulewidth\global\arrayrulewidth 1pt}
22     \hline
23     \noalign{\global\arrayrulewidth\savedwidth}
24 }
25 \renewcommand\theadfont{\footnotesize\sffamily}
```

## References

- [1] *A new implementation of LATEX's tabular and array environment*, Frank Mittelbach, David Carlisle, CTAN, v2.4k revised 2018/12/30.
- [2] *The cellspace package*, Josselin Noirel, CTAN, v1.8.1 2019/03/11.
- [3] *The tabularx package*, David Carlisle, CTAN, v2.11.b 2016/02/03.
- [4] *The makecell package*, Olga Lapko, CTAN, v0.1e 2009/08/03.
- [5] *The tabulary package*, David Carlisle, CTAN, v1.10 2014/06/11.
- [6] *The tablestyles package*, Matthias Pospiech, CTAN, v0.1 2014/06/27.
- [7] *Publication quality tables in LATEX*, package booktabs by Simon Fear, CTAN, v1.618033 2016/04/29.
- [8] *The tabs package*, Donald Arseneau, CTAN, v3.5 2010/02/26.
- [9] *The multirow, bigstrut and bigdelim packages*, Piet van Oostrum, Øystein Bache, Jerry Leichter, CTAN, v2.4 2019/01/01.
- [10] *The spacingtricks package*, Antoine Missier, CTAN, v1.3 2020/11/02.
- [11] *The LATEX Companion*. Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley, 2nd edition, Pearson Education, 2004.