

The `arraycols` package*

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

June 26, 2019

1 Introduction

This package provides new predefined column types to typeset tables in addition to the `array` package by Frank Mittelbach and David Carlisle [1] (loaded by `arraycols`) and also a command to draw wide horizontal rules. Here is a summary of the column types and macro, defined by `arraycols`, which we detail in the next section.

Column definitions	
<code>L</code>	Left adjusted column (LR mode in <code>array</code> environments or math mode in <code>tabular</code> environment)
<code>C</code>	Centered adjusted column (idem)
<code>R</code>	Right adjusted column (idem)
<code>t{<width>}</code>	Text column (LR mode) of fixed <code><width></code> like <code>p</code> , but horizontally and vertically centered
<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 with adjusted height, like <code>x</code> , and fixed <code><width></code>
<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> , like <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 <code><thickness></code>
Horizontal rules	
<code>\whline</code>	Wide horizontal rule (1 pt)

Note that if a column type has already been defined previously, it will be overwritten with a warning message.

*This document corresponds to `arraycols` v1.0, initial version dated 2019/06/26. Thanks to François Bastouil for English translation.

Besides `array`, `arraycols` loads the `cellspace` package, necessary for the `x`, `y`, `z` and `Z` types of columns and `tabularx` [3], necessary for `T` and `Z`, as well as `makecell` [4] for various alignments of multilined table cells.

This package, with a very short code, does not claim to develop new macros. Its main action is to combine and set features coming from other packages.

2 Usage

L Referring to an example from the `array` package documentation, `arraycols` provides
C the `L`, `C`, `R` type of columns which reverse the mathematical mode. Then we can
R use these column types to get centred, left-aligned or right-aligned LR-mode in an `array` or math-mode in a `tabular` environment. For instance, declaration `\begin{tabular}{|l|C|r|}` produces the second column in a centred mathematical mode and declaration `\begin{array}{|L|c|c|}` produces the first column in text mode, left aligned¹.

`t{width}` We add the definition `t{width}` (text in LR-mode) producing a horizontal and a vertical centering in the column unlike the classics `p{width}` and `m{width}` which produce left aligned text.

x To ensure a sufficient height for rows, for instance for formulas in `displaymath`
y mode, we have defined the columntypes `x` (centred) and `y` (left aligned), based on the `cellspace` package by Josselin Noirel [2]. They allows an automatic adjustment of the rows heights to avoid touching the horizontal rules when content is too high. Although `cellspace` is defined *a priori* for `tabular` environments, the new `x` and `y` column types, defined by `arraycols`, produce a column in mathematical mode with the right adjustment, either with a `tabular` or with an `array` environment.

Look at the following examples produced with `\begin{array}{|c|}` and with `\begin{array}{|x|}`.

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

The `cellspace` package is loaded with the `math` option² for a good management of rows heights in matrix tables.

¹The declarations `L`, `C`, `R` do not work in a `tabularx` environment. Note that the `tabulary` package by David Carlisle [5] already defines the `L`, `C`, `R`, `J` column types for particular alignments in tables of same type as `tabularx`, but there is no incompatibility because these column definitions only apply in `tabulary` environments.

²The `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”.

Notice that another package, `booktabs` [6], also provides an excellent adjustment for rows heights, but unfortunately, it doesn't handle the height of vertical separators |. To get the same vertical adjustment as `booktabs`, thus we set the `cellspace` parameters:

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

We should also mention the `tabls` package by Donald Arneseau [7] making a good adjustment of rows heights as well, but it's incompatible with the `array` and `numprint` packages.

At last, it is also possible to make manual adjustments with the `\vstrut` command from the `spacingtricks` package [9], or `\gape` and `\Gape` from the `makecell` package [4], or `\bigstrut` from the `bigstrut` package [8].

`z{<width>}` The column definition `z{<width>}` enables, just as `t{<width>}`, to set the column width, but also activates the math mode and adjusts the row height, as `x`.

`T` The `tabularx` package by David Carlisle [3] provides the `X` column definition
`Z` whose width is calculated according to the required width for the whole table, and with left alignment as for `p{<width>}`. `\begin{tabularx}{8cm}{|c|X|X|}` adjusts the width of the `X` columns for a total width of the table equals to 8cm. As a complement, we propose the `T` declaration, doing the same thing but with horizontal centering and `Z` which furthermore activates the mathematical mode and adjusts the height of the line (as `x` or `z`). Here is 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 multiline piece of text
------------	---	---------------	---	---------------------------

To keep the perfect alignment of fraction bars in mathematical formulas, cells are not vertically centred, however, to get a proper vertical positioning in the last cell, we have used the powerful `\makecell` command of the `makecell` package by Olga Lapko [4]: `\makecell{a multiline \ \ piece of text}`.

`I` The column definition `I` is suggested in The \LaTeX Companion [10] and enables to draw a thick vertical line (1 pt thick) instead of the one obtained with standard declaration |. To choose thickness, we propose further column definition `V{<thickness>}`³.

`\whline` Likewise, the `\whline` command, proposed in The \LaTeX Companion, enables to draw a thick horizontal line (1 pt thick) instead of the one obtained with `\hline` and the `makecell` package provides further command `\Xhline{<thickness>}` enabling to choose the thickness of the horizontal rule.

The introduction table has been typeset with a column declaration `I` as separator between the two columns of text, and with `\whline` for the horizontal rules at

³The definition of `V` would have been simplified by using an optional argument for `I` but this way out is not working.

the begin and at the end of the table, and `\Xhline{0.8pt}` for the one following the legend rows. Formatting header lines has been done with `\thead` command from the `makecell` package. For this command `arraycols` sets by default:

```
\renewcommand\theadfont{\footnotesize\sffamily}
(originally \footnotesize only, without \sffamily). At last, according to a recommendation of the array package [1], 1 pt has been added to the normal height of every row of the table, with the command \setlength{\extrarowheight}{1pt}4.
```

3 Implementation

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

The `cellspace` package provides the `S` modifier enabling, when placed before a column declaration, to adjust the height of the content of the cells to avoid to touch horizontal rules. Spacing between the content and the rules is controlled by the parameters `\cellspacetoplimit` and `\cellspacebottomlimit`.

```
10 \newcolumnntype{x}{>{\$}Sc<{\$}}
11 \newcolumnntype{y}{>{\$}Sl<{\$}}
12 \setlength{\cellspacetoplimit}{3pt}
13 \setlength{\cellspacebottomlimit}{2pt}
14 \newcolumnntype{z}[1]{>{\$}S{>{\centering\arraybackslash}p{#1}}<{\$}}
```

For the `z` definition of column, we use `p` and not `m` (which automatically centers) in order to keep a correct alignment for mathematical expressions in the cells of a same row.

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

The `T` columns are not automatically centred. It would be possible to do it with the command `\renewcommand{\tabularxcolumn}[1]{m{#1}}` (with `m` instead of default value `p`), but unfortunately this has a global effect for all the declarations of columns based on `X`, so `T` but also `Z`, and this would lead to disturb alignment of mathematical expressions in the cells of a 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}}
```

⁴As mentioned in the `array` package documentation: “This is important for tables with horizontal lines because those lines normally touch the capital letters”.

```
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] *Publication quality tables in LATEX*, package `booktabs` by Simon Fear, CTAN, v1.618033 2016/04/29.
- [7] *The tabs package*, Donald Arseneau, CTAN, v3.5 2010/02/26.
- [8] *The multirow, bigstrut and bigdelim packages*, Piet van Oostrum, Øystein Bache, Jerry Leichter, CTAN, v2.4 2019/01/01.
- [9] *The spacingtricks package*, Antoine Missier, CTAN, v1.0 2019/06/26.
- [10] *The LATEX Companion*. Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley, 2nd edition, Pearson Education, 2004.