The arraycols package*

Antoine Missier
antoine.missier@ac-toulouse.fr

November 23, 2020

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	
`L`	Left adjusted column (LR mode in `array` environments or math mode in `tabular` environment)
`C`	Centered adjusted column (idem)
`R`	Right adjusted column (idem)
`t{`⟨*width*⟩`}`	Text column (LR mode) of fixed ⟨*width*⟩ like `p`, but horizontally and vertically centered
`x`	Centered column in math mode with adjusted height to avoid touching the horizontal rules
`y`	Left aligned column in math mode with adjusted height
`z{`⟨*width*⟩`}`	Centered column in math mode with adjusted height, like `x`, and fixed ⟨*width*⟩
`T`	Centered text column with adjusted width for `tabularx` environments (calculated like `X` column)
`Z`	Centered column for `tabularx`, like `T`, but in math mode with adjusted height, like `x` and `z`
`I`	Thick vertical rule (1 pt)
`V{`⟨*thickness*⟩`}`	Vertical rule with variable ⟨*thickness*⟩
Horizontal rules	
`\whline`	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 v1.1, dated 2020/11/23. Thanks to François Bastouil for help in English translation.

Besides array, arraycols loads the cellspace package [2], 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. The tablestyles package [6] defines also L, C, R, Z column types but in a different way, nevertheless this package is incompatible with makecell and therefore with arraycols too.

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

2 Usage

`L` `C` `R` Referring to an example from the array package documentation, arraycols provides the `L`, `C`, `R` columns types which reverse the mathematical mode. Then we can use them to get centered, left-aligned or right-aligned LR-mode in an `array` environment or math-mode in a `tabular` environment. For instance, the declaration `\begin{tabular}{|l|C|r|}` sets centered mathematical mode in the second column and declaration `\begin{array}{|L|c|c|}` sets text mode, left aligned in the first column[1].

`t{`⟨*width*⟩`}` The new column type definition `t{`⟨*width*⟩`}` (text in LR-mode) produces horizontal and vertical centering in the column unlike the classics `p{`⟨*width*⟩`}` (in standard LaTeX) and `m{`⟨*width*⟩`}` (from the array package) which produce left aligned text (visible when the column is wider than the text inside).

`x` `y` To ensure sufficient height for rows, for instance in displaymath mode formulas, we provide the columntypes `x` (centered) and `y` (left aligned), based on the cellspace package by Josselin Noirel [2]. They allow automatic adjustment of row 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 good adjustment, either in a `tabular` or in an `array` environment.

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

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

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

[1] The 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.

The cellspace package is loaded with the `math` option[2] for a good management of row heights in matrix tables.

Notice that another package, booktabs [7], also provides an excellent adjustment of row heights, but unfortunately, it doesn't handle height of vertical separators `|`. To get the same vertical adjustment as booktabs, we set the cellspace parameters as follows:
`\setlength{\cellspacetoplimit}{3pt}`,
`\setlength{\cellspacebottomlimit}{2pt}`.

We should also mention the tabls 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.

At last, it is also possible to make manual adjustments with the `\vstrut` command from the spacingtricks package [10], or `\gape` and `\Gape` from the makecell package [4], or `\bigstrut` from the bigstrut package [9].

`z{`⟨*width*⟩`}` The column type `z{`⟨*width*⟩`}` enables to set the column width, as `t{`⟨*width*⟩`}`, but also activates the math mode and adjusts the row height, as `x`.

`T` `Z` The tabularx package by David Carlisle [3] provides the `X` column definition 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 witdh of the table equals to 8 cm. 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 line heights (as `x` or `z`). Here is an example with `\begin{tabularx}{\linewidth}{|T|y|x|Z|T|}`.

A good job	$\lim\limits_{\substack{x\to 1\\ x>1}} \ln\left(\dfrac{x^2}{x-1}\right)$	$\dfrac{a}{b}$	$\dfrac{a}{b}+\displaystyle\int_1^X \dfrac{1}{t}\,\mathrm{d}t$	a multiline piece of text

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

`I` `V{`⟨*thickness*⟩`}` The column definition `I` is suggested in The LaTeX Companion [11] 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`

[2] The `math` option loads the amsmath package. As mentionned 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".

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

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 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 this table, with the command `\setlength{\extrarowheight}{1pt}`[4].

3 Implementation

```
\RequirePackage{array}
\RequirePackage[math]{cellspace}
\RequirePackage{tabularx} % must be loaded after cellspace
\RequirePackage{makecell}

\newcolumntype{C}{>{$}c<{$}}
\newcolumntype{L}{>{$}l<{$}}
\newcolumntype{R}{>{$}r<{$}}
\newcolumntype{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`.

```
\newcolumntype{x}{>{$}Sc<{$}}
\newcolumntype{y}{>{$}Sl<{$}}
\setlength{\cellspacetoplimit}{3pt}
\setlength{\cellspacebottomlimit}{2pt}
\newcolumntype{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.

```
\newcolumntype{T}{>{\centering\arraybackslash}X}
\newcolumntype{Z}{>{$}ST<{$}}
```

The `T` columns are not automatically centered. 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.

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

```
\newcolumntype{I}{!{\vrule width 1pt}}
\newcolumntype{V}[1]{!{\vrule width #1}}
\newlength\savedwidth
\newcommand{\whline}{%
  \noalign{\global\savedwidth\arrayrulewidth\global\arrayrulewidth 1pt}
  \hline
  \noalign{\global\arrayrulewidth\savedwidth}
}
\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* tabls *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, v1.0 2019/06/26.

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