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.

<table>
<thead>
<tr>
<th>Column definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>R</td>
</tr>
<tr>
<td>t{⟨width⟩}</td>
</tr>
<tr>
<td>x</td>
</tr>
<tr>
<td>y</td>
</tr>
<tr>
<td>z{⟨width⟩}</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>Z</td>
</tr>
<tr>
<td>I</td>
</tr>
<tr>
<td>(V{⟨thickness⟩}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Horizontal rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>\whline</td>
</tr>
</tbody>
</table>

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 \texttt{array}, \texttt{arraycols} loads the \texttt{cellspace} package [2], necessary for the \texttt{x}, \texttt{y}, \texttt{z} and \texttt{Z} types of columns and \texttt{tabularx} [3], necessary for \texttt{T} and \texttt{Z}, as well as \texttt{makecell} [4] for various alignments of multilined table cells. The \texttt{tablestyles} package [6] defines also \texttt{L}, \texttt{C}, \texttt{R}, \texttt{Z} column types but in a different way, nevertheless this package is incompatible with \texttt{makecell} and therefore with \texttt{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

\texttt{L} Referring to an example from the \texttt{array} package documentation, \texttt{arraycols} provides the \texttt{L}, \texttt{C}, \texttt{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 \texttt{array} environment or math-mode in a \texttt{tabular} environment. For instance, the declaration \begin{verbatim}
\begin{tabular}{|l|C|r|}
\end{verbatim} sets centered mathematical mode in the second column and declaration \begin{verbatim}
\begin{array}{|L|c|c|}
\end{verbatim} sets text mode, left aligned in the first column\footnote{The declarations \texttt{L}, \texttt{C}, \texttt{R} do not work in a \texttt{tabularx} environment. Note that the \texttt{tabularx} package by David Carlisle [5] already defines the \texttt{L}, \texttt{C}, \texttt{R}, \texttt{J} column types for particular alignments in tables of same type as \texttt{tabularx}, but there is no incompatibility because these column definitions only apply in \texttt{tabularx} environments.}.

\begin{verbatim}
t{\langle width\rangle}
\end{verbatim} The new column type definition \texttt{t}\{\langle width\rangle\} (text in LR-mode) produces horizontal and vertical centering in the column unlike the classics \texttt{p}\{\langle width\rangle\} (in standard \LaTeX) and \texttt{m}\{\langle width\rangle\} (from the \texttt{array} package) which produce left aligned text (visible when the column is wider than the text inside).

\texttt{x} To ensure sufficient height for rows, for instance in displaymath mode formulas, we provide the column types \texttt{x} (centered) and \texttt{y} (left aligned), based on the \texttt{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 \texttt{cellspace} is defined \textit{a priori} for \texttt{tabular} environments, the new \texttt{x} and \texttt{y} column types, defined by \texttt{arraycols}, produce a column in mathematical mode with good adjustment, either in a \texttt{tabular} or in an \texttt{array} environment.

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

\begin{verbatim}
\begin{array}{|c|}
\hline
\text{bad} \\
\hline
\lim_{x \to 1} \ln \left( \frac{x^2}{x - 1} \right) \\
\hline
\frac{a}{b} \\
\hline
\int_1^X \frac{1}{t} \, dt \\
\hline
\end{array}
\end{verbatim}

\begin{verbatim}
\begin{array}{|x|}
\hline
\text{good} \\
\hline
\lim_{x \to 1} \ln \left( \frac{x^2}{x - 1} \right) \\
\hline
\frac{a}{b} \\
\hline
\int_1^X \frac{1}{t} \, dt \\
\hline
\end{array}
\end{verbatim}
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 \(\mid\). 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 Arneeseau [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 \texttt{\vstrut} command from the `spacingtricks` package [10], or \texttt{\gape} and \texttt{\Gape} from the `makecell` package [4], or \texttt{\bigstrut} from the `bigstrut` package [9].

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

The `tabularx` package by David Carlisle [3] provides the \texttt{X} column definition whose width is calculated according to the required width for the whole table, and with left alignment as for \texttt{p{⟨width⟩}}. \begin{tabularx}{8cm}{|c|X|X|} \end{tabularx} adjusts the width of the \texttt{X} columns for a total width of the table equals to 8 cm. As a complement, we propose the \texttt{T} declaration, doing the same thing but with horizontal centering and \texttt{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|} \end{tabularx}.

To keep the perfect alignment 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 powerful \texttt{\makecell} command of the `makecell` package by Olga Lapko [4]: \texttt{\makecell{a multiline \ \ piece of text}}.

The column definition \texttt{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 \texttt{I}. To choose thickness, we propose further column definition \texttt{V{⟨thickness⟩}}\(^3\).

Likewise, the \texttt{\whline} command, proposed in The \LaTeX\ Companion, enables to draw a thick horizontal line (1 pt thick) instead of the one obtained with \texttt{\hline}

\(^2\)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”.

\(^3\)The definition of \texttt{V} would have been simplified by using an optional argument for \texttt{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 \ 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}.

3 Implementation

\begin{verbatim}
1 \RequirePackage{array}
2 \RequirePackage{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}{>{\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 \newcolumntype{x}{>{$}Sc<{$}}
11 \newcolumntype{y}{>{$}Sl<{$}}
12 \setlength{\cellspacetoplimit}{3pt}
13 \setlength{\cellspacebottomlimit}{2pt}
14 \newcolumntype{z}{>{\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 \newcolumntype{T}{>{\centering\arraybackslash}X}
16 \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.
\end{verbatim}

\footnote{As mentioned in the array package documentation: “This is important for tables with horizontal lines because those lines normally touch the capital letters”}
\begin{tabular}{|l|l|}
\hline
\textbf{References} & \\
\hline
[1] & \textit{A new implementation of LATEX’s tabular and array environment}, Frank Mittelbach, David Carlisle, CTAN, v2.4k revised 2018/12/30. \\
[3] & \textit{The tabularx package}, David Carlisle, CTAN, v2.11.b 2016/02/03. \\
[4] & \textit{The makecell package}, Olga Lapko, CTAN, v0.1e 2009/08/03. \\
[6] & \textit{The tablestyles package}, Matthias Pospiech, CTAN, v0.1 2014/06/27. \\
[8] & \textit{The tabs package}, Donald Arseneau, CTAN, v3.5 2010/02/26. \\
[9] & \textit{The multirow, bigstrut and bigdelim packages}, Piet van Oostrum, Oystein Bache, Jerry Leichter, CTAN, v2.4 2019/01/01. \\
\hline
\end{tabular}