The colortbl package

David Carlisle
2012/02/13

Abstract

This package implements a flexible mechanism for giving colored ‘panels’ behind specified columns in a table. This package requires the array and color packages.

1 Introduction

This package is for colouring tables (i.e., giving coloured panels behind column entries). In that it has many similarities with Timothy Van Zandt’s colortab package. The internal implementation is quite different though, also colortab works with the table constructs of other formats besides \LaTeX. This package requires \LaTeX (and its color and array packages).

First, a standard \texttt{tabular}, for comparison.

\begin{tabular}{|l|c|}
\hline
one & two \\
\hline
three & four \\
\hline
\end{tabular}

2 The \texttt{\textbackslash columncolor} command

The examples below demonstrate various possibilities of the \texttt{\textbackslash columncolor} command introduced by this package. The vertical rules specified by | are kept in all the examples, to make the column positioning clearer, although possibly you would not want coloured panels and vertical rules in practice.

The package supplies a \texttt{\textbackslash columncolor} command, that should (only) be used in the argument of a > column specifier, to add a coloured panel behind the specified column. It can be used in the main ‘preamble’ argument of \texttt{array} or \texttt{tabular}, and also in \texttt{\textbackslash multicolumn} specifiers.

The basic format is:

\begin{verbatim}
\texttt{\textbackslash columncolor\{}\langle color model\rangle\}\{}\langle colour\rangle\} \langle\{left overhang\}\}\langle\{right overhang\}\}
\end{verbatim}

The first argument (or first two if the optional argument is used) are standard color package arguments, as used by \texttt{\textbackslash color}.

∗This file has version number v1.0a, last revised 2012/02/13.
The last two arguments control how far the panel overlaps past the widest entry in the column. If the right overhang argument is omitted then it defaults to left overhang. If they are both omitted they default to \texttt{tabcolsep} (in \texttt{tabular}) or \texttt{arraycolsep} (in \texttt{array}).

If the overhangs are both set to 0pt then the effect is:

\begin{verbatim}
|>{\columncolor[gray]{.8}[0pt]}l|
>{\color{white}\columncolor[gray]{.2}[0pt]}l|
\end{verbatim}

The default overhang of \texttt{tabcolsep} produces:

\begin{verbatim}
|>{\columncolor[gray]{.8}}l|
>{\color{white}\columncolor[gray]{.2}}l|
\end{verbatim}

You might want something between these two extremes. A value of .5\texttt{tabcolsep} produces the following effect:

\begin{verbatim}
|>{\columncolor[gray]{.8}[,5\texttt{tabcolsep}]}l|
>{\color{white}\columncolor[gray]{.2}[,5\texttt{tabcolsep}]}l|
\end{verbatim}

This package should work with most other packages that are compatible with the \texttt{array} package syntax. In particular it works with \texttt{longtable} and \texttt{dcolumn} as the following example shows.

Before starting give a little space:\texttt{\setlength\minrowclearance{2pt}}

\begin{verbatim}
\begin{longtable}{p{1cm}p{1cm}p{1cm}p{1cm}}
\hline
\textbf{A long table example} & \multicolumn{3}{c}{Third column} \\
\textbf{First two columns} & \textbf{p-type} & \textbf{D-type (dcolumn)} \\
\hline
P-column & and another one & 12-34 \\
Total & (wrong) & 100-6 \\
\hline
Some long text in the first column & bbb & 1-2 \\
aaa & and some long text in the second column & 1-345 \\
\hline
Total & (wrong) & 100-6 \\
\hline
aaa & |
Note that the coloured rules in all columns stretch to accommodate large entries in one column. & |
\hline
\end{longtable}
\end{verbatim}

Continued...
## A long table example (continued)

<table>
<thead>
<tr>
<th>First two columns</th>
<th>Third column</th>
</tr>
</thead>
<tbody>
<tr>
<td>p-type</td>
<td>D-type (\texttt{dcolumn})</td>
</tr>
<tr>
<td>bbb</td>
<td>100</td>
</tr>
<tr>
<td>Depending on your driver you may get unsightly gaps or lines where the ‘screens’ used to produce different shapes interact badly. You may want to cause adjacent panels of the same colour by specifying a larger overhang or by adding some negative space (in a \texttt{\noalign} between rows.).</td>
<td>12.4</td>
</tr>
<tr>
<td>bbb</td>
<td>45.3</td>
</tr>
</tbody>
</table>

This example shows rather poor taste but is quite colourful! Inspect the source file, \texttt{colortbl.dtx}, to see the full code for the example, but it uses the following column types.

\begin{verbatim}
\newcolumntype{A}{% >{\color{white}\columncolor{red}[.5\tabcolsep]% \raggedright}p{2cm}}
\newcolumntype{B}{% >{\columncolor{blue}[.5\tabcolsep]% \color{yellow}\raggedright}p{3cm}}
\newcolumntype{C}{% >{\columncolor{yellow}[.5\tabcolsep]% \D{.}{\cdot}{3.3}}} \newcolumntype{E}{% >{\large\bfseries \columncolor{cyan}[.5\tabcolsep]}c} \newcolumntype{F}{% >{\color{white}\columncolor{magenta}[.5\tabcolsep]}c} \newcolumntype{G}{% >{\columncolor{gray}[0.8][.5\tabcolsep][\tabcolsep]}l}
\end{verbatim}
3 Using the ‘overhang’ arguments for \texttt{tabular*}

The above is all very well for \texttt{tabular}, but what about \texttt{tabular*}? Here the problem is rather harder. Although \TeX{}’s \texttt{leader} mechanism which is used by this package to insert the ‘stretchy’ coloured panels is rather like \texttt{glue}, the \texttt{tabskip} glue that is inserted between columns of \texttt{tabular*} (and \texttt{longtable} for that matter) has to be ‘real glue’ and not ‘leaders’.

Within limits the overhang options may be used here. Consider the first table example above. If we use \texttt{tabular*} set to 3 cm with a preamble setting of

\begin{verbatim}
\begin{tabular*}{3cm}{%
@{\extracolsep{\fill}}
>{\columncolor[gray]{.8}[0pt][20mm]}l
>{\columncolor[gray]{.8}[5mm][0pt]}l
@{}}
one\ & two  \
three & four
\end{tabular*}
\end{verbatim}

Changing the specified width to 4 cm works, but don’t push your luck to 5 cm...

\begin{tabular}{cc}
one & two
\end{tabular}

\begin{tabular}{cc}
three & four
\end{tabular}

4 The \texttt{\textbackslash rowcolor} command

As demonstrated above, one may change the colour of specified rows of a table by the use of \texttt{\multicolumn} commands in each entry of the row. However if your table is to be marked principally by \texttt{rows}, you may find this rather inconvenient. For this reason a new mechanism, \texttt{\textbackslash rowcolor}, has been introduced\footnote{At some cost to the internal complexity of this package}.

\texttt{\textbackslash rowcolor} takes the same argument forms as \texttt{\textbackslash columncolor}. It must be used at the \texttt{start} of a row. If the optional overhang arguments are not used the overhangs will default to the overhangs specified in any \texttt{\textbackslash columncolor} commands for that column, or \texttt{\textbackslash tabcolsep (arraycolsep in array).}

If a table entry is in the scope of a \texttt{\textbackslash columncolor} specified in the table preamble, and also a \texttt{\textbackslash rowcolor} at the start of the current row, the colour specified by \texttt{\textbackslash rowcolor} will take effect. A \texttt{\textbackslash multicolumn} command may contain \texttt{\textbackslash rowcolor...} which will override the default colours for both the current row and column.
\begin{tabular}{|l|c|}
\rowcolor{gray}{.9}
one&two\\
\rowcolor{gray}{.5}
three&four
\end{tabular}

5 The \texttt{\cellcolor} command

A background colour can be applied to a single cell of a table by beginning it with \texttt{\multicolumn{1}{>{\rowcolor\cellcolor\columncolor}{. . . , (or \columncolor if no row-colour is in effect))} but this has some deficiencies: 1) It prevents data within the cell from triggering the colouration; 2) The alignment specification must be copied from the top of the tabular, which is prone to errors, especially for p{\} columns; 3) \texttt{\multicolumn{1}} is just silly. Therefore, there is the \texttt{\cellcolor} command, which works like \texttt{\columncolor} and \texttt{\rowcolor}, but over-rides both of them; \texttt{\cellcolor} can be placed anywhere in the tabular cell to which it applies.

6 Colouring rules.

So you want coloured rules as well?

One could do vertical rules without any special commands, just use something like !{\color{green}\vline} where you’d normally use \|. The space between || will normally be left white. If you want to colour that as well, either increase the overhang of the previous column (to \tabcolsep + \arrayrulewidth + \doublerulesep) Or remove the inter rule glue, and replace by a coloured rule of the required thickness. So

!{\color{green}\vline}
@{\color{yellow}\vrule width \doublerulesep}
!{\color{green}\vline}

Should give the same spacing as || but more colour.

However colouring \texttt{\hline} and \texttt{\cline} is a bit more tricky, so extra commands are provided (which then apply to vertical rules as well).

7 \texttt{\arrayrulecolor}

\texttt{\arrayrulecolor} takes the same arguments as \texttt{\color}, and is a global declaration which affects all following horizontal and vertical rules in tables. It may be given outside any table, or at the start of a row, or in a > specification in a table preamble. You should note however that if given mid-table it only affects rules that are specified after this point, any vertical rules specified in the preamble will keep their original colours.
8 \texttt{\doublerulesepcolor}

Having coloured your rules, you’ll probably want something other than white to go in the gaps made by || or \texttt{\hline}. \texttt{\doublerulesepcolor} works just the same way as \texttt{\arrayrulecolor}. The main thing to note that if this command is used, then \texttt{\longtable} will not ‘discard’ the space between \texttt{\hline} at a page break. (\TeX{} has a built-in ability to discard space, but the coloured ‘space’ which is used once \texttt{\doublerulesep} is in effect is really a third rule of a different colour to the two outer rules, and rules are rather harder to discard.)

\begin{verbatim}
\setlength{\arrayrulewidth}{2pt}\arrayrulecolor{blue}
\setlength{\doublerulesep}{2pt}\doublerulesepcolor{yellow}
\begin{tabular}{||l||c||}
\hline\hline one & two \\
three & four \\
\hline\hline
\end{tabular}
\end{verbatim}

9 More fun with \texttt{\hhline}

The above commands work with \texttt{\hhline} from the \texttt{hhline} package, however if \texttt{hhline} is loaded in addition to this package, a new possibility is added. You may use >{\ldots} to add declarations that apply to the following – or = column rule. In particular you may give \texttt{\arrayrulecolor} and \texttt{\doublerulesepcolor} declarations in this argument.

Most manuals of style warn against over use of rules in tables. I hate to think what they would make of the following rainbow example:

\begin{verbatim}
\newcommand\rainbowline[1]{% \\
  >{\arrayrulecolor {red}\doublerulesepcolor[rbg]{.3,.3,1}}\% \\
  >{\arrayrulecolor {orange}\doublerulesepcolor[rbg]{.4,.4,1}}\% \\
  >{\arrayrulecolor {yellow}\doublerulesepcolor[rbg]{.5,.5,1}}\% \\
  >{\arrayrulecolor {green}\doublerulesepcolor[rbg]{.6,.6,1}}\% \\
  >{\arrayrulecolor {blue}\doublerulesepcolor[rbg]{.7,.7,1}}\% \\
}
\end{verbatim}
Richard of York gave battle in vain

\begin{tabular}{||*7{>{\columncolor[.9]c}}||}
\multicolumn{1}{>{\columncolor[.9]c}||}{vain}\end{tabular}

10 Less fun with \texttt{\textbackslash cline}

Lines produced by \texttt{\textbackslash cline} are coloured if you use \texttt{\arrayrulecolor} but you may not notice as they are covered up by any colour panels in the following row. This is a ‘feature’ of \texttt{\textbackslash cline}. If using this package you would probably better using the \texttt{-rule type} in a \texttt{\textbackslash hhline} argument, rather than \texttt{\textbackslash cline}.

11 The \texttt{\textbackslash minrowclearance} command

As this package has to box and measure every entry to figure out how wide to make the rules, I thought I may as well add the following feature. ‘Large’ entries in tables may touch a preceding \texttt{\hline} or the top of a colour panel defined by this style. It is best to increase \texttt{\extrarowsep} or \texttt{\arraystretch} sufficiently to ensure this doesn’t happen, as that will keep the line spacing in the table regular. Sometimes however, you just want to \LaTeX{} to insert a bit of extra space above a large entry. You can set the length \texttt{\minrowclearance} to a small value. (The height of a capital letter plus this value should not be greater than the normal height of table rows, else a very uneven table spacing will result.)

Donald Arseneau’s \texttt{tabs} packages provides a similar \texttt{\tablinesep}. I was going to give this the same name for compatibility with \texttt{tabs}, but that is implemented quite differently and probably has different behaviour. So I’ll keep a new name for now.