lutabulartools

some useful LuaLaTeX-based tabular tools

Kale Ewasiuk (`kalekje@gmail.com`)

2023–07–22

`lutabulartools` is a package that contains a few useful LuaLaTeX-based macros to help with tables. A global `lua` variable `lutabt` is created. This package redefines the `tabular` and `tabular*` environments as well as `\@arraycr` to add functionarlity. The following packages are loaded by this one, so if you have specific settings for these packages, load the `lutabulartools` package after: `booktabs, multirow, makecell, xparse, array, xcolor, colortbl, luacode, penlightplus`.

1 `\settabular`

A key-val interface in the `\settabular{}` command is used to set some tabular settings.
`nopad` automatically adds `@{}` on each end of the column spec.
`tbrule` automatically adds `\toprule` as the first thing in the `tabular(*|x)` environment, and `\bottomrule` as the last. Note that this automatic top/bottom rule adding is disabled in `longtable`. `row/colsep` tweaks the row spacing with `arraystretch` or adjusts the `tabcolsep` length (an integer must be used, the result is multiplied by 6pt). For example:

```
\settabular{nopad,tbrule,rowsep=2,colsep=2} % or
\settabular{nopad=false,tbrule=false} % to set the switches to off
```

2 Debugging

You can toggle log output debugging with `\lttdebugON` and `\lttdebugOFF`. The messages will be printed in a format like so:

```
vvvvv msg1 (lutabulartools)
msg2
^^^^^
```

`\lttdebugprt` can be used pretty-print the `lutabt` module and its attributes—useful for checking the "state" of the package.

3 `\MC` – Magic Cell

`\MC` (magic cell) combines the facilities of `\multirow` and `\multicolumn` from the `multirow` package, and `\makcell` from the titular package. With the help of LuaLaTeX, it takes an easy-to-use cell specification and employs said commands as required. `\MC` will not work properly if your table is only 1 column wide (you probably don't need `MC` in that case anyway...). Here is the usage:

```
\MC * [cell spec] [override multicol] <cell format> {contents}
```

`*` This will wrap the entire command in {}. This is necessary for `siunitx` single-column width columns. However, the `\MC` command attempts to detect this automatically.

`[cell spec]` Any letters placed in this argument are used for cell alignment. You can use one of three: `t`, `m`, `b` for top, middle, bottom (vertical alignment), and/or `l`, `c`, `r` for horizontal alignment, in no particular order. By default, `\MC` will try to autodetect the horizontal alignment based on the current column. If it can't, it will be left-aligned. The default vertical alginment is top. More on this in section 3.1.

This argument can also contain two integers, separated by a comma (if two are used). `C,R`, `C`, or `,R` are a valid inputs, where `R`=rows (int), and `C`=columns, (int). If you want a 1 column wide, multirow cell, you can pass `,R`. These numbers can be negative (positive numbers occupy columns to the right and rows below, and negative numbers occupy columns to the left and rows above). If no spec is passed, (argument empty), `\MC` acts like a `makecell`. Additionally, you can pass `+` in place of C (number of columns wide), and it will make the cell width fill until the end of the current row.

Examples:
`\MC[2,2]` means two columns wide, two rows tall.
`\MC[2,1]` or `\MC[2]` means two columns wide, one row tall.
`\MC[1,2]` or `\MC[,2]` means one column wide, two rows tall.
`\MC[+,2]`, if placed in the first column, occupies the entire row and is two rows tall.
`\MC[+,2]`, if placed in the second column, occupies the second column to the end of the table and is two rows tall.
In any of these examples, you can place the alignment letters anywhere. So, `\MC[1l,2b]` and `\MC[1,2 lb]` are both left-bottom aligned (spaces are ignored).
Put an `_` as the last thing to add a `cmidrule` under, followed by a trim spec (the thing that goes in `cmidrule()`) `\MC[+c_l]` will add a `cmidrule` underneath the cell (which fills until the end of the table), with the left side trimmed.

`[override mc]` You may want to adjust the column specification of a multicolumn cell, `[@{}c@{}]` for example to remove padding between the cell.

`<cell format>` You can place formatting like `\bfseries` here.

3.1 `\MC` Defaults

The `tabular[*]` environment is re-defined to use Lua pattern matching to parse the column specification of the table. This is done to determine how many columns there are, and what the current column type is, even if specifications like `r@{.}l*{3}{r}` are used. If you have defined a column that expands many, you should register it with `\setMCrepl{?}{??}` where `?` is your column and `??` is what it expands to. You can also specify default horizontal and vertical alignments (ie if alignment not passed to `\MC`) for an arbitrary column by `\setMChordef{?}{l|r|c}` and `\setMChordef{?}{t|m|b}`, where `?` is the column. To add a column that should be surrounded by brackets for `siunitx` purposes, do so with `\addMCsicol{?}`. `S` is included by default.

4 `\midrulesat`

If you'd rather specify the location of midrules outside the table, use `\midrulesat{1,2,3}`: a midrule will be placed on rows 1, 2, and 3, for the next table only.

5 Examples

Change the settings for the rest of the doc.

```
\settabular{nopad,tbrule}
```

5.0.1 A good use for headers

```
\begin{tabular}{ l l l }
\MC[+m]<\itshape>{A Decent
                 Example}\\\midrule
    & \MC[2m_]{Heading} \\
\MC[b,-2]{Multi\\Line}  & A & B \\\midrule
   end & & \\
\end{tabular}
```

A Decent Example		
Multi	Heading	
Line	A	B
end		

5.0.2 A small example

```
\midrulesat{1,2,3}
\begin{tabular}{ l l l }
a \\
b \\
c \\
d \\
\end{tabular}
```

a
b
c
d

5.0.3 A small example

```
\begin{tabular}{ l l l }\midrule
 \MC[_]{A} & \MC[mc2,2]{Lttrs}    \\
 \MC[_r]{B} &         \\ \cmidrule↩
    {2-3}
 \MC[_r]{1} & \MC[_r]{A} & \MC[_r]{B}↩
       \\
  \\
\end{tabular}
```

A	Lttrs	
B		
1	A	B

5.0.4 A small bad example

Notice that the top-aligned p-column doesn't play particularly well with the middle aligned `\MC`

```
\begin{tabular}{ p{1cm} l }
   hello\newline world
        & \MC[mr]{11\\2} \\
\end{tabular}
```

hello world	11 2

5.0.5 If you insist on vertical lines

```
\begin{tabular}{|c|c|c|} \hline
1 & 2 & 3\\\hline
4 & \MC[2,2cm][@{}c@{}|]%
    <\ttfamily>{5}\\\cline{1-1}
  & \MC[2][r|]{} \\\hline%hacky fix
6 & 7 & 8\\\hline
\end{tabular}
```

1	2	3
4	5	
6	7	8

5.0.6 A perhaps useful example

```latex
\begin{tabularx}{\linewidth}{S[table-format=2.1,table-alignment=left]X}
% err & ... \\% ERROR, not wrap
  \MC{Error,\%} & Comment \\% MC helps
 3.1 & \MC[,2]{multi-line\\comment}\\
 10.1& \\
 \MC[2c]{... ...} \\
\end{tabularx}
```

Error,%	Comment
3.1	multi-line
10.1	comment
... ...	

5.0.7 A messy example

```latex
\begin{tabular}{| c | c | c | c | c | c |}\toprule
 \MC[2,2cm]<\ttfamily>{2,2cm}   & \MC[2r]<\ttfamily>{2r} & 5 & \MC[,2b]<\ttfamily>{,2b}\\
   &   & 3 & 4 & 5 & \\\midrule
 1 & 2 & \MC[2l][@{}l]<\ttfamily>{2l (\@\{\}l)} & 5 & 6666\\\cmidrule{3-4}
 1 & \MC[+r]<\ttfamily>{+r}  \\
  \\
 1 & 2 & 3 & 4 & 5 & \MC[,-2]<\ttfamily>{,\\-2}\\
\end{tabular}
```

2,2cm		2r		5	
		3	4	5	,2b
1	2	2l ({}l)		5	6666
1					+r
					,
1	2	3	4	5	-2

6 Some additional rules

This package also redefines the `booktabs` midrules.

`\gmidrule` is a full gray midrule.

By taking advantage of knowing how many columns there are (if you chose to redefine `tabular`), you can specify individual column numbers (for a one column wide rule), or reference with respect to the last column (blank, `+1`, `+0`, or `+` means last column, `+2` means second last column, for example) or omit the last number.

`\cmidrule` is a single partial rule, with the above features

`\gcmidrule` is a single partial gray rule, with the above features

You can add multiple `cmidrule`'s with the `(g)cmidrules` command. Separate with a comma. You can apply global trimming of the rules with the `()` optional argument, and then override it for a specific rule by placing `r` or `l` with the span specification.

`\gcmidrules` Can produce multiple, light gray partial rules

`\cmidrules` Can produce multiple black partial rules.

Here's an example:

```
\begin{tabular}{c c c c c c}
 a & 2 & 3 & 4 & 5 & 6\\
    \cmidrule{+1}  % rule on last column
 b & 2 & 3 & 4 & 5 & 6\\
    \cmidrules{1,3-+3,+} % rule on first col, third to third last col, and last col
 c & 2 & 3 & 4 & 5 & 6\\
    \cmidrules{1,3-+3rl,+} % same as above, but trim middle
 d & 2 & 3 & 4 & 5 & 6\\
    \cmidrules(l){1,r3-+3,+1}% trim left for all, but only trim right for middle rule
 e & 2 & 3 & 4 & 5 & 6\\
    \gcmidrule{+1}  % rule on last column
 f & 2 & 3 & 4 & 5 & 6\\
    \gcmidrules{1,3-+3,+} % rule on first col, third to third last col, and last col
 g & 2 & 3 & 4 & 5 & 6\\
    \gcmidrules{1,3-+3rl,+} % same as above, but trim middle
 h & 2 & 3 & 4 & 5 & 6\\
    \gcmidrules(l){1,r3-+3,+1}% trim left for all, but only trim right for middle rule
\end{tabular}
```

a	2	3	4	5	6
b	2	3	4	5	6
c	2	3	4	5	6
d	2	3	4	5	6
e	2	3	4	5	6
f	2	3	4	5	6
g	2	3	4	5	6
h	2	3	4	5	6

7 `\midruleX` - Midrule every Xth row

With this command, you can place a rule every X rows for the next table made (place command outside of table). You can configure the step size and what kind of midrule you prefer with the following key-val syntax, with default values below:
`\midruleX{step=5,rule=midrule,cntr=0,head=0,long=false}`
`step` is the number of rows before applying the rule set by `rule`.

Concerning `longtable`s: If `long` is set to `true` (or the key is present), `\midruleX` will also add a unique `\label{}` on each ro (to detect page changes), and if the row starts on a newpage, resets the row counter.

Use optional parameter `\midruleX*[o|n|f]{}` to control expansion of the key-val settings (`[n]` for not expanded is default). Before you want counting to begin, or anywhere in the table, you could apply `\resetmidruleX[cntr]` to an arbitrary value: `cntr` is normally incremented by 1 each row. You may want to avoid header rows being counted, or rows being underlined near the end of a table, for example. If you want to skip the first `x` rows with `\midruleX`, set `head=x` (which is equivalent to `cntr=-x`). If you want to skip the auto-ruling at that ro `x` (say a gray one) and instead use a `midrule`, use `head*=x`.

Note: Use `\noalign{\resetmiduleX}` if you need place a rule on the same line the reset takes place (ie. in a cell before `\\`).

7.1 `\midruleX` examples

```
\midruleX{step=3,rule=gmidrule,head*=1}
\begin{tabular}{rclc}
                % ^^^ inject midruleX
Num   & . & . & .  \\
1     & & &  \\
2     & & &  \\
3     & & &  \\
4     & \MC[2,2]{Hi\\world}  \\
5     & & &  \\
6     & & &  \\
7     & & &  \\
8     & & &  \\
9     & & &  \\
10    & & &  \\
11    & & &  \\
\resetmidruleX % resest
   % so no bottom gray rule
12    & & &  \\
\end{tabular}
```

Num	.	.	.
1			
2			
3			
4	Hi		
5	world		
6			
7			
8			
9			
10			
11			
12			

Here's an example with long table. Notice the gray rules reliably appear 3 rows after each header.

```
\midruleX{long=true,step=3,rule="\gcmidrules{1r,2-}"}
\def\tblhead{\toprule No & Name & Place & other\\\midrule}
\def\tblcontinued{\MC[+l]{Continued...}\\}
\def\tblcontinues{\MC[+r]{...Continues}\\}
\begin{longtable}{rclc}
  \tblhead\endfirsthead
  \tblcontinued\tblhead\endhead % all the lines above this will be repeated on every pag
  \tblcontinues\endfoot
  \bottomrule End.\endlastfoot
  \resetmidruleX
1       &           &       &  \\
2       &           &       &    \\
3       &           &       &    \\
4       &           &       &    \\
5       &           &       &    \\
6       &           &       &    \\
7       &           &       &    \\
8       &           &       &    \\
9       &           &       &    \\
10      &           &       &    \\
11      &           &       &    \\
12      & \MC[2,-2](){Hi\\world}     \\
13      &           &       &    \\
14      &           &       &    \\
15      &           &       &    \\\newpage
16      &   \MC[2,2]{Hi\\world}     \\
17      &           &       &    \\
18      &           &       &    \\\resetmidruleX
19      &           &       &    \\
\end{longtable}
```

No	Name	Place	other
1			
2	hi		
3			
4			
5			
			...Continues

...

Continued...

No	Name	Place	other
6			
7			
8			
9			
10			
11	Hi		
12	world		
13			
14			
15			

...Continues

...

Continued...

No	Name	Place	other
	Hi		
16	world		
17			
18			
19			

End.