Typesetting tables with \LaTeX

Klaus Höppner

1 Basic tables

\LaTeX{} already has built-in support to typeset tables. For beginners it may be a bit confusing, since \LaTeX{} provides two environments: tabular and table. To typeset material in rows and columns, tabular is needed, while the table environment is a container for floating material similar to figure, into which a tabular environment may be included.

So, let’s have a look how to typeset a simple table:

\begin{tabular}{lcr}
a & b & c \\
aa & ab & ac \\
\end{tabular}

will result in

a b c
aa ab ac

The rows of the table are divided by \LaTeX{}’s usual \textbackslash{} command (in some cases, it may be needed to use \texttt{\textbackslash{}tabularnewline} instead, as we will see later in this article). Columns are separated by \&, the ampersand character.

The required argument of tabular defines the basic layout of the table, especially the alignment of the columns:

\begin{itemize}
  \item \texttt{l} left aligned column
  \item \texttt{c} centered column
  \item \texttt{r} right aligned column
  \item \texttt{p\{\langle width\rangle\}} paragraph-like column of a predefined width (with the baseline of the paragraph’s first line aligned relative to the other cells in the table row)
\end{itemize}

The normal space between columns, which is also added before the first and after the last column, may be overridden by \texttt{\@{\langle sep\rangle}}, where \langle sep\rangle is any \LaTeX{} code, inserted as the separator. For illustration, let’s typeset some flight data:

\begin{tabular}{@{}lr@{--}l}
\text{flight no.} & \text{route} \\
LH 402 & Frankfurt–Newark \\
KL 3171 & Amsterdam–Cork \\
US 1152 & San Diego–Philadelphia
\end{tabular}

Here, the \texttt{@} command is used twice: The space that normally would have been inserted left of the first column is replaced by nothing, thus the table is left aligned with the surrounding text (compare it with the first tabular example in this article, you will see the difference). Additionally, the inter-column space between the points of departure and destination is replaced by a dash. So the code used to produce this table looks as follows (silently introducing \texttt{\multicolumn} to combine cells):

\begin{verbatim}
\begin{tabular}{@{}lr@{--}l}
\text{flight no.} & \text{\multicolumn{2}{c}{route}} \\
LH 402 & Frankfurt & Newark \\
KL 3171 & Amsterdam & Cork \\
US 1152 & San Diego & Philadelphia
\end{tabular}
\end{verbatim}

2 Extra packages for typesetting tables

Beyond \LaTeX{}’s built-in ability to typeset tables, several extra packages exist. Some of them add new effects in typography and layout, others simplify the task of writing the document’s source code. The packages that I will introduce in this article (and more that I won’t) are covered in detail in the \LaTeX{} Companion by Mittelbach et al. (2004).

Here are some important packages for authors who want to typeset tables:
array adds paragraph-like columns \multicolumn{1}{c}{}

2.1 Using array

From my point of view, the most important feature of array written by MITTELBach e CARLISLe is the new possibility of defining commands that are added automatically before or after a column’s cell. It saves a lot of typing, making the source code more concise and more flexible. array is one of the required \LaTeX\ packages, so it must be part of any \LaTeX\ installation.

For a simple example, have a look at the following table:

<table>
<thead>
<tr>
<th>Command</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>\alpha</td>
<td>\alpha</td>
</tr>
<tr>
<td>\beta</td>
<td>\beta</td>
</tr>
<tr>
<td>\gamma</td>
<td>\gamma</td>
</tr>
</tbody>
</table>

The left column displays \LaTeX\ commands, beginning with a backslash and using a typewriter font, while the right columns displays the corresponding math symbol. Using the array package, the source code is pretty straightforward:

\begin{tabular}{|l|c|}
\hline
\multicolumn{1}{|c|}{Command} & \multicolumn{1}{|c|}{Symbol} \\
\hline
\alpha & \alpha \\
\beta & \beta \\
\gamma & \gamma \\
\hline
\end{tabular}

As shown in this code, we can now define a command sequence inside the \texttt{>\{...\}} preceding the column type definition. These commands are executed before each cell of that column. Similarly, using \texttt{<\{...\}} after the column type defines which commands to be executed after typesetting the column’s cells.

In the example above, the first row is different from the others, since it contains the column titles that obviously should not be typeset in typewriter font or math mode. This is handled by ‘abusing’ the \multicolumn command, which prevents the \texttt{>} and \texttt{<} command hooks from being applied for these cells.

Another use of these command hooks is typesetting paragraphs in narrow columns. \LaTeX\ typesets these paragraphs left and right justified by default, but in narrow columns it is often more appropriate to typeset them using \texttt{\raggedright}. So we might think of trying the following code:

\begin{tabular}{|l|}
\hline
\texttt{\raggedright}{3in}
\hline
\end{tabular}

Unfortunately this fails when ending the table rows with the \texttt{\\} command, with rather weird error messages about misplaced aligns. The problem is that \texttt{\raggedright} redefines \texttt{\}, so it can’t be recognized as the end of table rows. There are three solutions for this problem:

1. Use \texttt{\tabularnewline} instead of \texttt{\\}. In fact, it does no harm to always use this, even when you don’t have problems with \texttt{\raggedright}.

2. Restore the original definition of \texttt{\} by using the command \texttt{\arraybackslash}, as follows:

\begin{tabular}{|l|}
\hline
\texttt{\}x{3in}\texttt{\arraybackslash}
\hline
\end{tabular}

3. Use Martin SCHRÖDER\’s \texttt{ragged2e} package. It redefines the command \texttt{\raggedright} to prevent it from redefining \texttt{\}, so the problem disappears without any further change to the original code. Additionally, \texttt{ragged2e} provides the new command \texttt{\RaggedRight} that typesets the paragraph left aligned, but doesn’t disable hyphenation.

2.2 Using tabularx

Besides the normal \texttt{tabular} environment, a rarely used environment \texttt{tabular*} exists. In addition to the column definition, it takes a table width as argument. The resulting table is typeset to this width, but surprisingly by expanding the space between columns.

A more convenient implementation is done by the \texttt{tabularx} (\texttt{CARLISLe}) package (another required \LaTeX\ package present in every \LaTeX\ installation). This introduces a column type \texttt{X} for paragraph-like columns whose width is automatically calculated in order to achieve a table of a desired total width. For example, let’s look at the following:

\begin{tabularx}{\linewidth}{|l|}
\hline
\multicolumn{1}{|c|}{Label & Text} \\
\hline
\texttt{One & This is some text without meaning, just using up some space. It is not intended for reading.}\ \\
\texttt{...} \\
\hline
\end{tabularx}
This produces a table across the full line width, where the right column just uses the space remaining after typesetting the left column:

<table>
<thead>
<tr>
<th>Label</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>This is some text without meaning, just using up some space. It is not intended for reading.</td>
</tr>
<tr>
<td>Two</td>
<td>This is another text without meaning, just using up some space. It’s not intended for reading either.</td>
</tr>
<tr>
<td>Three</td>
<td>This is yet another text without meaning. Guess what? It’s not intended for reading. It is just there.</td>
</tr>
<tr>
<td>Four</td>
<td>How often did I mention that you should not read this text?</td>
</tr>
</tbody>
</table>

It is possible to use more than one X-column. By default, all of them are typeset to the same width, but it is possible to manually adjust how the available space is divided. Here’s our next example:

<table>
<thead>
<tr>
<th>Label</th>
<th>Text</th>
<th>More text</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>This is some text without meaning.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This is another text without meaning, just using up some space.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>It is not meant for reading.</td>
<td></td>
</tr>
</tbody>
</table>

This table was produced with the following code:

```latex
\begin{tabularx}{\linewidth}%
{l>{\setlength\hsize{0.6\hsize}\raggedright}X%
>{\setlength\hsize{1.4\hsize}\raggedright}X}
\hline
Label & Text & More text	abularnewline
\hline
\end{tabularx}
```

When balancing the column widths manually, it is important that the \hsize fractions add up to the number of X-columns, as in the example above, where $0.6 + 1.4 = 2$. To achieve automatic balancing of columns, take a look at the \texttt{tabularx} package.

Be aware that the way \texttt{tabularx} parses the contents of a table limits the possibility of defining new environments based on the \texttt{tabularx}. If you consider doing this, first look at the documentation.

### 3 Using lines in tables

\LaTeX \ provides the possibility of using lines in tables: vertical lines are added by placing a | at the appropriate position in the definition of the column layout, and horizontal lines are added by using the command \texttt{\hline}.

While using lines in tables can help the reader in understanding the contents of a table, it is quite easy to produce really ugly tables like the following:

<table>
<thead>
<tr>
<th>Label</th>
<th>Text</th>
<th>More text</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>This is some text without meaning.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This is another text without meaning, just using up some space.</td>
<td></td>
</tr>
</tbody>
</table>

Though nobody would typeset this particular table in real life, it illustrates a general and common problem: the column titles and the word “another” in the rightmost column touch the horizontal lines above them.

As a first step to improve the spacing between the table rows and the horizontal lines in such cases, set \texttt{\extrarowheight} to a non-zero length, e.g. to 4pt. If this isn’t enough, additional adjustment may be done by adding invisible rules. Here is revised source code for the above example illustrating both these points:

```
\setlength{\extrarowheight}{4pt}
\begin{tabularx}{\linewidth}%
{\setlength\hsize{0.67\hsize}\raggedright}X%
{\setlength\hsize{1.33\hsize}\raggedright}X}
\hline
Label & Text & More text	abularnewline
\hline
\hline
One & This is some text without meaning.                                      |
    & This is another text without meaning, just using up some space.         |
\hline
\rule{0pt}{18pt} & \huge another & text without meaning, just using up some space. \textbackslash\textbackslash |
\hline
\end{tabularx}
```

we get a somewhat better result:

<table>
<thead>
<tr>
<th>Label</th>
<th>Text</th>
<th>More text</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>This is some text without meaning.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This is another text without meaning, just using up some space.</td>
<td></td>
</tr>
</tbody>
</table>

Please notice that the \texttt{\rule} used as an additional spacer was typeset with a horizontal width of 0.4 pt instead of 0 pt (as shown in the code) in order to make its effect and location visible.

Even after this, the layout of the table still looks quite poor, e.g. the broken vertical lines between the double horizontal line. This might be solved with the package \texttt{hhline} \cite{CARLILE}, but for typesetting tables with pretty lines, have a look at the \texttt{booktabs} package by \texttt{FEAR e EL}. It starts by giving users a basic piece of advice, namely to avoid vertical lines, and introduces commands to typeset horizontal lines with appropriate thickness and spacing. Using \texttt{booktabs}, the source code for our weird example now looks as follows:

```
\begin{tabularx}{\linewidth}%
{\setlength\hsize{0.67\hsize}\raggedright}X%
{\setlength\hsize{1.33\hsize}\raggedright}X}
\toprule
Label & Text & More text	abularnewline
\toprule
One & This is some text without meaning.                                      |
\hline
\end{tabularx}
```

```
Using this, the result becomes:

<table>
<thead>
<tr>
<th>Label</th>
<th>Text</th>
<th>More text</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>This is some text without meaning.</td>
<td>This is another text without meaning, just using up some space.</td>
</tr>
</tbody>
</table>

At last, we’ve improved the layout of the table quite a bit. The content with arbitrary changes of font size still looks weird, but that’s something for which the author and not \LaTeX must be blamed.

For a more realistic example of using rules, here I present an example from the booktabs manual:

<table>
<thead>
<tr>
<th>Item</th>
<th>Animal</th>
<th>Description</th>
<th>Price ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gnat</td>
<td>per gram</td>
<td>13.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td>each</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>Gnu</td>
<td>stuffed</td>
<td>92.50</td>
</tr>
<tr>
<td></td>
<td>Emu</td>
<td>stuffed</td>
<td>33.33</td>
</tr>
<tr>
<td></td>
<td>Armadillo</td>
<td>frozen</td>
<td>8.99</td>
</tr>
</tbody>
</table>

As shown in this source code, the longtable environment may contain definitions for headers and footers before the normal content of the table:

- \endfirsthead defines what to typeset at the very first head of the table,
- \endhead defines a table head that is used on continuation pages,
- \endfoot defines what to typeset on the foot of the table if a continuation page follows, and
- \endlastfoot defines the foot at the very end of the table.

I personally prefer longtable simply because there exists yet another package \ltxtable (Carlisle \cite{Carlisle}), which combines longtable and tabularx. It provides the command \LTXTable{⟨width⟩}{⟨file⟩}, which reads a given file containing a longtable environment using X-columns and typesets it to the requested width.

### Riferimenti bibliografici

- Braams, J. e Jurriens, J. «The supertabular package». CTAN: \url{macros/latex/contrib/supertabular}
- Carlisle, D. (a). «The hhline package». CTAN: \url{macros/latex/required/tools}
- (b). «The longtable package». CTAN: \url{macros/latex/required/tools}
- (c). «The \ltxtable package». CTAN: \url{macros/latex/contrib/carlisle}
- (d). «The tabularx package». CTAN: \url{macros/latex/required/tools}
- Fear, S. e Els, D. «The booktabs package». CTAN: \url{macros/latex/contrib/booktabs}
- Mittelbach, F. e Carlisle, D. «The array package». CTAN: \url{macros/latex/required/tools}
- Schröder, M. «The ragged2e package». CTAN: \url{macros/latex/contrib/ms}

\footnotesize
\begin{itemize}
  \item Klaus Höppner
  Haardtring 230a
  64295 Darmstadt
  Germany
  klaus.hoeppner (at) gmx (dot) de
\end{itemize}