The \texttt{supertabular} environment

Johannes Braams and Theo Jurriens

2002/07/19

1 Introduction

The package \texttt{supertabular} offers a new environment, the \texttt{supertabular} environment. As the name indicates it is an extension of the normal \texttt{tabular} environment.

With the original \texttt{tabular} environment a tabular must always fit on one page. If the tabular becomes too large the text overwrites the page’s bottom margin and you get an \texttt{Overfull vbox} message.

The \texttt{supertabular} environment uses the \texttt{tabular} environment internally, but it evaluates the used space every time it gets a \texttt{\}\ command. If the tabular reaches the textheight, it automatically inserts an optional tabletail, an \texttt{\end{tabular}} command, starts a new page, a new \texttt{tabular} environment and inserts the optional tablehead on the new page continuing the tabular.

2 User interface

The package \texttt{supertabular} has three options, they control the amount of information that is written to the \texttt{.log} file.

1. The option \texttt{errorshow} (the default) doesn’t write any extra information.

2. The option \texttt{pageshow} writes information about when and why \texttt{supertabular} decides to break the tabular environment in order to produce a new page.

3. The option \texttt{debugshow} also adds information about each line that is added to the tabular.

Below is a description of the new commands and environments that this package provides.

\begin{itemize}
  \item \texttt{\tablefirsthead} takes one argument, it defines the contents of the first occurrence of the tabular head.
  \item \texttt{\tablehead} takes one argument, it defines the contents of all subsequent occurrences of the tabular head.
  \item \texttt{\tabletail} takes one argument, it defines something which should be inserted before each \texttt{\end{tabular}}, except the last.
  \item \texttt{\tablelasttail} takes one argument, it defines something
\end{itemize}

\footnote{This file has version number v4.1e, last revised 2002/07/19.}
which should be inserted before the last `\end{tabular}`.
The use of this command is optional.

\begin{tabular}{|c|c|c|c|}
\hline
Number & Number$^2$ & Number$^4$ & Number! \\
\hline
1 & 1 & 1 & 1 \\
2 & 4 & 16 & 2 \\
3 & 9 & 81 & 6 \\
4 & 16 & 256 & 24 \\
\hline
\end{tabular}

\textit{continued on next page}

3 \textbf{Weak points}

- When the material of a normal entry (not a p-arg) becomes larger than the estimated $\ST@lineht$, overfull $\vbox$es will be produced at all.

- When the last p-arg on a page gets more than 4 lines (probably even more than 3 lines) it will result in an overfull $\vbox$. Also some combinations of $\baselinestretch$ $\arraystretch$ and a large font may lead to one line too much.

- if accidentally the last line of the tabular produces a newpage, on the next page the tabletail will be written immediately after the tablehead. Depending on the contents this may result in an error message regarding misplaced $\noalign$.

A quick but not very elegant solution: shrink the allowed height of the table with the command $\shrinkheight{\ldots pt}$ after the first `\` of the supertabular.

- The $\mpsupertabular$ environment sometimes has problems with pagesbreaks when footnotes appear in the lower part of the tabular.

4 \textbf{Examples}

Here is an example of a \texttt{supertabular}. You will find the definitions after the table.

\begin{verbatim}
\topcaption{Top caption.}
\bottomcaption{Bottom caption.}
\tablecaption{Table caption.}
\end{verbatim}
<table>
<thead>
<tr>
<th>Number</th>
<th>Number²</th>
<th>Number⁴</th>
<th>Number!</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>25</td>
<td>625</td>
<td>120</td>
</tr>
<tr>
<td>6</td>
<td>36</td>
<td>1296</td>
<td>720</td>
</tr>
<tr>
<td>7</td>
<td>49</td>
<td>2401</td>
<td>5040</td>
</tr>
<tr>
<td>8</td>
<td>64</td>
<td>4096</td>
<td>40320</td>
</tr>
<tr>
<td>9</td>
<td>81</td>
<td>6561</td>
<td>362880</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>10000</td>
<td>3628800</td>
</tr>
<tr>
<td>11</td>
<td>121</td>
<td>14641</td>
<td>39916800</td>
</tr>
<tr>
<td>12</td>
<td>144</td>
<td>20736</td>
<td>479001600</td>
</tr>
<tr>
<td>13</td>
<td>169</td>
<td>28561</td>
<td>6.22702080E+9</td>
</tr>
<tr>
<td>14</td>
<td>196</td>
<td>38416</td>
<td>8.71782912E+10</td>
</tr>
<tr>
<td>15</td>
<td>225</td>
<td>50625</td>
<td>1.30767437E+12</td>
</tr>
<tr>
<td>16</td>
<td>256</td>
<td>65536</td>
<td>2.09227899E+13</td>
</tr>
<tr>
<td>17</td>
<td>289</td>
<td>83521</td>
<td>3.55687428E+14</td>
</tr>
<tr>
<td>18</td>
<td>324</td>
<td>104976</td>
<td>6.40237370E+15</td>
</tr>
<tr>
<td>19</td>
<td>361</td>
<td>130321</td>
<td>1.21645100E+17</td>
</tr>
<tr>
<td>20</td>
<td>400</td>
<td>160000</td>
<td>2.43290200E+18</td>
</tr>
</tbody>
</table>

Table 1: This table is split across pages

And here is (part of) the user input for the table above:

\begin{center}
\tablefirsthead{%
\hline
\multicolumn{1}{|c}{\tbsp Number} &
\multicolumn{1}{|c}{Number$^2$} &
\multicolumn{1}{|c|}{Number$^4$} &
\multicolumn{1}{|c|}{Number!} \\
\hline}
\tablehead{%
\hline
\multicolumn{1}{|c|}{\small\sl continued from previous page} \\
\hline
\multicolumn{1}{|c|}{\tbsp Number} &
\multicolumn{1}{|c}{Number$^2$} &
\multicolumn{1}{|c|}{Number$^4$} &
\multicolumn{1}{|c|}{Number!} \\
\hline}
\tabletail{%
\hline
\multicolumn{4}{|r|}{\small\sl continued on next page} \\
\hline}
\tablelasttail{\hline}
\bottomcaption{This table is split across pages}
\end{center}
Here is another example with a \texttt{p} column-definition. The tablehead is the same as above. The tabletail is a double \texttt{\hline}; \texttt{\arraystretch} is set to 1.5 and the font size is \texttt{\small}.

Table 2: This table should also be split across pages.

<table>
<thead>
<tr>
<th>Number</th>
<th>Number$^2$</th>
<th>Number$^4$</th>
<th>Number!</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>here is a relative short entry</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>here is a relative short entry</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
</tbody>
</table>

\textit{continued on next page}
Here is the same table again, but this time using the `supertabular*` environment and stretching the table to the full width of the text.

Table 3: This table should also be split across pages.

<table>
<thead>
<tr>
<th>Number</th>
<th>Number$^2$</th>
<th>Number$^4$</th>
<th>Number!</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1</td>
<td>here is a relative short entry</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
</tbody>
</table>

continued on next page
<table>
<thead>
<tr>
<th>Number</th>
<th>Number(^2)</th>
<th>Number(^4)</th>
<th>Number!</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1 here is a relative short entry</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>1 here is a relative short entry</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
<td></td>
</tr>
</tbody>
</table>

*continued on next page*
5 Known problems

- When a float occurs on the same page as the start of a supertabular you can expect unexpected results. When the float was defined on the same page you might end up with the first part of the supertabular on a page by its own.

- You should not use the supertabular inside a floating-environment such as \texttt{table} as this will result in \TeX{} trying to put the whole supertabular on one page.

- In some instances you might still end up with overfull \texttt{vbox} messages.

- Sometimes the last page of the supertabular contains just an empty head and tail.

6 The Implementation

\begin{verbatim}
\newcount\c@tracingst
\DeclareOption{errorshow}{\c@tracingst\z@}
\DeclareOption{pageshow}{\c@tracingst\tw@}
\DeclareOption{debugshow}{\c@tracingst\fiv@\relax}
\ProcessOptions

\topcaption \bottomcaption
\tablecaption
\ifST@star
\newif\if@topcaption \@topcaptiontrue
\def\topcaption{\@topcaptiontrue\tablecaption}
\def\bottomcaption{\@topcaptionfalse\tablecaption}
\global\let\@process@tablecaption\relax
\ifST@star
\newif\if@topcaption \@topcaptiontrue
\def\topcaption{\@topcaptiontrue\tablecaption}
\def\bottomcaption{\@topcaptionfalse\tablecaption}
\global\let\@process@tablecaption\relax
\end{verbatim}

This command has to function exactly like \texttt{\caption} does, except it has to store its argument (and the optional argument) for later processing \textit{within} the supertabular environment.

\begin{verbatim}
\long\def\tablecaption{%
  \refstepcounter{table}\dblarg{\@xtablecaption}}
\long\def\@xtablecaption[#1]#2{%
  \long\gdef\@process@tablecaption{\ST@caption{table}[#1]{#2}}
\global\let\@process@tablecaption\relax

\ifST@star
\newif\if@topcaption \@topcaptiontrue
\def\topcaption{\@topcaptiontrue\tablecaption}
\def\bottomcaption{\@topcaptionfalse\tablecaption}
\global\let\@process@tablecaption\relax
\end{verbatim}

This switch is used in the internal macros to remember which kind of environment was started.
This switch is used in the internal macros to remember if the tabular should be put into a minipage.
\newif\ifST@mp

For the supertabular* environment it is necessary to store the intended width of the tabular.
\newdimen\ST@wd

For the mpsupertabular environments we need special versions of \leftskip, \rightskip and \parfillskip.
\newskip\ST@rightskip
\newskip\ST@leftskip
\newskip\ST@parfillskip

This is a redefinition of LaTeX's \@caption, \@makecaption is called within a group so as not to return to \normalsize globally. also a fix is made for the 'feature' of the \makecaption of the document class article and friends that a caption always gets a \vskip 10pt at the top and none at the bottom. If a user wants to precede his table with a caption this results in a collision.
\long\def\ST@caption#1[#2]#3\par{
\addcontentsline{\csname ext@#1\endcsname}{#1}{\numberline{\csname the#1\endcsname}{#2}}
\begingroup
\@parboxrestore
\normalsize
\if@topcaption \vskip -10\p@ \fi
\@makecaption{\csname fnum@#1\endcsname}{#3}\par
\if@topcaption \vskip 10\p@ \fi
\endgroup}

\tablehead
\tablefirsthead
\tabletail
\tablelasttail

If the user uses an extra amount of tabular-data (like \multicolumn) in \tabletail \TeX{} starts looping because of the definition of \ST@cr. So make \\ act just like a \@tabularcr inside this tail to prevent the loop. Save and restore the value of \\.
There now is a possibility to follow the decisions supertabular makes about breaking the tabular. This has to be enabled when converting this file with docstrip to a .sty file.

\newcommand\sttraceon\c@tracingst5\relax
\newcommand\sttraceoff\c@tracingst\z@}

A macro that gets the trace message as its argument

\ifnum\c@tracingst>1\relax
\GenericWarning{(supertabular)@spaces@spaces}
\PackageWarning{supertabular: #2}\i

This register holds the estimate of the amount of space left over on the current page. This is used in the decision when to start a new page.

\shrinkheight A command to diminish the value of \ST@pageleft if necessary.

\setSTheight A command to set the value of \ST@pageleft if necessary.

The register \ST@headht will hold the height of the first head of a supertabular environment; the register \ST@tailht will hold the height of table tail (if any)

The register \ST@pagesofar is used to store the estimate of the amount of page already filled up.

The measured (total) height of a parbox-argument

The estimated height of a normal line is stored in \ST@lineht. The dimension register \ST@stretchht is used to store the difference between the ‘normal’ line height and the line height when \arraystretch has a non-standard value. This is used in the case where p-box entries are added to the tabular. The dimension register \ST@prevht is use to store the height of the previous line to use it as an estimate for the height of the next line. This is needed for a better estimate of when to break the tabular.
When a tabular row is ended with `\[...\]` we need to temporarily store the optional argument in `\ST@toadd`.

A private scratch dimension register.

A box register to temporarily store the contents of a parbox.

These are redefinitions of `\@tabularcr` and `\@xtabularcr`. This is needed to include `\ST@cr` in the definition of `\@xtabularcr`.

All redefined macros have names that are similar to the original names, except with a leading `ST'`

In this case we need to copy the value of the optional argument of `\[` in our private register `\ST@toadd`.

The macros that deal with parbox columns need to be redefined, because we need to know the size of the parbox.

To achieve our goal we need to save the text in box.

Our version of `\astartpbox`.

\begin{verbatim}
\ST@toadd When a tabular row is ended with `\[...\]` we need to temporarily store the optional argument in `\ST@toadd.
71 \newdimen\ST@toadd

\ST@dimen A private scratch dimension register.
72 \newdimen\ST@dimen

\ST@pbox A box register to temporarily store the contents of a parbox.
73 \newbox\ST@pbox

\ST@tabularcr \ST@xtabularcr \ST@argtabularcr These are redefinitions of `\@tabularcr` and `\@xtabularcr`. This is needed to include `\ST@cr` in the definition of `\@xtabularcr`.
All redefined macros have names that are similar to the original names, except with a leading `ST'`

74 \def\ST@tabularcr{%
75 {\ifnum0=`\fi}
76 \@ifstar{\ST@xtabularcr}{\ST@xtabularcr}}
77 \def\ST@xtabularcr{%
78 \@ifnextchar[\ST@argtabularcr
79 {\ifnum0=`\fi}
80 \ifnum\@width=0
81 \unskip\ST@xargarraycr{#1}#1
82 \else
83 \ST@yargarraycr{#1}
84 \else
85 \ST@yargarraycr{#1}
86 \fi
87 \fi

\ST@xargarraycr \ST@yargarraycr In this case we need to copy the value of the optional argument of `\[` in our private register `\ST@toadd`.

88 \def\ST@xargarraycr#1{%
89 {\tempright\textwidth\advance\tempright \ht0 \unskip\ST@xargarraycr{#1}
90 \noalign{\global\ST@toadd=\ht0\ST@cr}
91 \noalign{\global\ST@toadd=\ht0\ST@cr}
92 \noalign{\global\ST@toadd=\ht0\ST@cr}
93 \noalign{\global\ST@toadd=\ht0\ST@cr}

\ST@yargarraycr The macros that deal with parbox columns need to be redefined, because we need to know the size of the parbox.

94 \def\ST@startpbox#1{%
95 \bgroup\hsize#1\hsize\global\ST@toadd=\ht0\ST@cr}
96 \noalign{\global\ST@toadd=\ht0\ST@cr}
97 \noalign{\global\ST@toadd=\ht0\ST@cr}
98 \noalign{\global\ST@toadd=\ht0\ST@cr}

\ST@startpbox Our version of `\astartpbox`.

99 \def\ST@astartpbox#1{%
100 \bgroup\hsize#1\hsize\global\ST@toadd=\ht0\ST@cr}
101 \noalign{\global\ST@toadd=\ht0\ST@cr}
102 \noalign{\global\ST@toadd=\ht0\ST@cr}
103 \noalign{\global\ST@toadd=\ht0\ST@cr}
Our version of \end{pbox} and \aend{pbox}.

\def\ST@endpbox{%  
\ifnum\ST@pboxht<\ST@dimen
\global\ST@pboxht=\ST@dimen
\fi
\ST@dimen=\z@
\box\ST@pbox\hfil}
\def\ST@aendpbox{%  
\ifnum\ST@pboxht<\ST@dimen
\global\ST@pboxht=\ST@dimen
\fi
\unvbox\ST@pbox\egroup\hfil}

\def\estimate@lineht{  
\ST@lineht=\arraystretch \baselineskip  
\global\advance\ST@lineht by 1\p@  
\ST@stretchht\ST@lineht\advance\ST@stretchht-\baselineskip  
\ifdim\ST@stretchht<\z@\ST@stretchht=\z@\fi
\ST@trace\tw@{Average line height: \the\ST@lineht}  
\ST@trace\tw@{Stretched line height: \the\ST@stretchht}  
}

\def\@calfirstpageht{%  
\ST@trace\tw@{Calculating height of tabular on first page}  
The \LaTeX{} register \pagetotal{} contains the height of the page so far, the \LaTeX{} register \@colroom{} contains the height of the column.
\global\ST@pagesofar\pagetotal  
\global\ST@pageleft\@colroom  
\ST@trace\tw@{Height of text = \the\pagetotal; \MessageBreak Height of page = \the\ST@pageleft}  
When we are in twocolumn mode \LaTeX{} may still be collecting material for the first column although there seems to be no space left. In this case we have to check against two times \ST@pageleft.  
\if\twocolumn  
\ST@trace\tw@{two column mode}  
\if\firstcolumn  
\ST@trace\tw@{First column}  
\ifnum\ST@pagesofar > \ST@pageleft  
\global\ST@pageleft=2\ST@pageleft  
\fi  
\fi  
\fi  
\else  
\fi
In this case we're in the second column, so we have to compensate for the material in the first column.

When \texttt{\ST@pagesofar} is smaller than \texttt{\ST@pageleft} \TeX{} is still collecting material for the first column, so we can start a new tabular environment like we do on a single column page.

When \texttt{\ST@pagesofar} is greater than \texttt{\ST@pageleft} \TeX{} has already decided it had enough material for the first column and is building the second column.

In one column mode there is a simple decision.

When we are not starting a new page subtract the size of the material already on it from the available space.

Now we need to know the height of the head of the table. In order to measure this we typeset it in a normal \texttt{tabular} environment.
To decide when to start a new page, we need to know the vertical size of the tail of the table.

To decide whether we can continue on the current page or whether we need to start on a new page. We assume that the minimum height of a tabular is the height of the head, the tail and one line of data. If that doesn’t fit a new page is started.

This calculates the maximum height of the tabular on all subsequent pages of the supertabular environment.

The body of the beginning of both environments is stored in a single macro as the code is shared.
The same needs to be done for the \texttt{tabular*} environment. The coding is slightly more verbose.

\begin{verbatim}
\let\csname org@tabular*\endcsname\tabular*
\let\csname inner@tabular*\endcsname\tabular*
\let\@oldcr=\cr
\global\let\baselineskip=\baselineskip
\def\baselineskip{\baselineskip}
\if\undefined\@classix
  \let\org@tabularcr=\@tabularcr
  \let\@tabularcr\ST@tabularcr
  \let\org@startpbox=\@startpbox
  \let\org@endpbox=\@endpbox
  \let\@@startpbox=\ST@startpbox
  \let\@@endpbox=\ST@endpbox
\else
  \let\org@tabularcr=\@arraycr
  \let\@arraycr\ST@tabularcr
  \let\org@startpbox=\@startpbox
  \let\org@endpbox=\@endpbox
  \let\@startpbox=\ST@astartpbox
  \let\@endpbox=\ST@aendpbox
\fi
\if\@table@first@head\undefined
  \let\@@tablehead=\@tablehead
\else
  \let\@@tablehead=\@table@first@head
\fi
\let\ST@skippage\ST@skipfirstpart
\let\ST@skippart=\ST@skipsecondpart
\if\undefined\@table@first@head\undefined
  \let\@@tablehead=\@tablehead
\else
  \let\@@tablehead=\@table@first@head
\fi
\end{verbatim}

The first part of a supertabular may be moved on to the next page if it doesn’t fit on the current page afterall. Subsequent parts can not be moved; therefore we will have to switch the definition of \texttt{\ST@skippart} around.
Now we can estimate the average line height and the height of the first page of the supertabular.

\estimatelineht
\@calfirstpageht
\noindent
}

\supertabular
We start by looking for an optional argument, which will be duly ignored as it seems to make no sense to try to align a multipage table in the middle...

\def\supertabular{\
  \@ifnextchar[{{@supertabular}}{\@supertabular\[]}}

We can now save the preamble of the \texttt{tabular} in a macro.

\def\@supertabular[#1]{\def\ST@tableformat{#1}\ST@trace\tw@{Starting a new supertabular}}

Then remember that this is not a \texttt{supertabular*} environment.

\global\ST@starfalse
Don’t use minipages.

\global\ST@mpfalse

Most of the following code is shared between the \texttt{supertabular} and \texttt{supertabular*} environments. So to avoid duplication it is stored in a macro.

\x@supertabular

Finally start a normal tabular environment.

\expandafter\org@tabular\expandafter{\ST@tableformat}\@@tablehead

\supertabular*
We start by looking for the optional argument of the \texttt{tabular} environment.

\@namedef{supertabular*}{#1}\%\@ifnextchar[{{@nameuse{@supertabular*}{#1}}}{\@nameuse{@supertabular*}{#1}\[]}

We start by saving the intended width and the preamble of the \texttt{tabular*}.

\@namedef{supertabular*}{#1[#2][#3]}\%
\ST@trace\tw@{Starting a new supertabular*}\%
\def\ST@tableformat{#3}\%
\ST@wd=#1\relax
\global\ST@startrue
\global\ST@mpfalse

Now we can call the common code for both environments.

\x@supertabular

And we can start a normal \texttt{tabular*} environment.

\expandafter\csname org@tabular*\expandafter\endcsname
\expandafter{\expandafter\ST@wd\expandafter}\expandafter{\expandafter\ST@tableformat}\%
\global\ST@mpfalse
This version of the supertabular environment puts each tabular into a minipage, thus making footnotes possible. We start by looking for an optional argument, which will be duly ignored as it seems to make no sense to try to align a multipage table in the middle...

```latex
\def\mpsupertabular{% 
  \@ifnextchar[{{\@mpsupertabular}}{}
}
```

We can now save the preamble of the tabular in a macro.

```latex
\def\@mpsupertabular[#1]#2{% 
  \def\ST@tableformat{#2}% 
  \ST@trace\tw@{Starting a new mpsupertabular}% 
  \ST@starfalse 
  \@nameuse{@mpsupertabular*}{#1}{}% 
}
```

Then remember that this is not a mpsupertabular* environment.

```latex
\global\ST@startrue 
```

And remember to close the minipage later.

```latex
\global\ST@mptrue 
```

Since we are about to start a minipage of \texttt{columnwidth} the horizontal alignment will no longer work. We have to remember the values and restore them inside the minipage.

```latex
\ST@rightskip \rightskip 
\ST@leftskip \leftskip 
\ST@parfillskip \parfillskip 
```

Most of the following code is shared between the \texttt{mpsupertabular} and \texttt{mpsupertabular*} environments. So to avoid duplication it is stored in a macro.

```latex
\x@supertabular 
Finally start a normal \texttt{tabular} environment.

```latex
\minipage{\texttt{columnwidth}}% 
\parfillskip\ST@parfillskip 
\rightskip\ST@rightskip 
\leftskip\ST@leftskip 
\noindent\expandafter\org@tabular\expandafter{\ST@tableformat}% 
\@@tablehead} 
```

\texttt{\mpsupertabular*} We start by looking for the optional argument of the tabular environment.

```latex
\@namedef{mpsupertabular*}#1{% 
  \@ifnextchar[{\@nameuse{\mpsupertabular*}{#1}}{% 
  \{\@nameuse{\mpsupertabular*}{#1}[]}% 
}} 
```

Now we can save the intended width and the preamble of the \texttt{tabular*}.

```latex
\@namedef{\mpsupertabular*}#1[#2][#3]{% 
  \ST@trace\tw@{Starting a new mpsupertabular*}% 
  \def\ST@tableformat[#3]{% 
    \ST@wd=#1\relax 
    \global\ST@startrue 
    \global\ST@mptrue 
    \ST@rightskip \rightskip 
    \ST@leftskip \leftskip 
    \ST@parfillskip \parfillskip 
  } 
  \noindent\expandafter\org@tabular\expandafter{\ST@tableformat}% 
  \\@@tablehead} 
```
Then we can call the common code for both environments.

\begin{macrocode}
\begin{minipage}{\columnwidth}\
\parfillskip\ST@parfillskip
\rightskip \ST@rightskip
\leftskip \ST@leftskip
\noindent\expandafter\csname org@tabular*\expandafter\endcsname
\expandafter{\expandafter\ST@wd\expandafter}\expandafter{\ST@tableformat}\@@tablehead
\end{minipage}
\endsupertabular
\endsupertabular*
\end{macrocode}

This closes the environments supertabular and supertabular*.

\begin{macrocode}
\def\endsupertabular*{\
\ifx\@table@last@tail\undefined
\@tabletail
\else
\@table@last@tail
\fi
\csname endtabular\ifST@star*\fi\endcsname
\ST@restore
\if@topcaption
\else
\@process@tablecaption
\@topcaptiontrue
\fi
\global\let\\\@oldcr
\global\let\@process@tablecaption\relax
\ST@trace\tw@{Ended a supertabular*}\
\end{macrocode}

Check if we have to insert a caption and restore to default behaviour of putting captions at the top.

\begin{macrocode}
\if\@topcaption
\else
\@process@tablecaption
\@topcaptiontrue
\fi
\global\let\\\@oldcr
\global\let\@process@tablecaption\relax
\ST@trace\tw@{Ended a supertabular*}\
\end{macrocode}

Restore the meaning of $\backslash$ to the one it had before the start of this environment.

Also re-initialize some control-sequences

\begin{macrocode}
\global\let\\\@oldcr
\global\let\@process@tablecaption\relax
\ST@trace\tw@{Ended a supertabular*}\
\end{macrocode}

The definition of the ending of the supertabular* environment is simple:

\begin{macrocode}
\def\endmpsupertabular*{\
\ifx\@table@last@tail\undefined
\@tabletail
\else
\@table@last@tail
\fi
\csname endtabular\ifST@star*\fi\endcsname
\end{macrocode}

This closes the environments mpsupertabular and mpsupertabular*.

\begin{macrocode}
\def\endmpsupertabular{\
\ifx\@table@last@tail\undefined
\@tabletail
\else
\@table@last@tail
\fi
\csname endtabular\ifST@star*\fi\endcsname
\end{macrocode}

Restore the original definition of $\backslash$tabularcr

\begin{macrocode}
\ST@restore
\end{macrocode}
Check if we have to insert a caption and restore to default behaviour of putting captions at the top.

338 \if@topcaption
339 \else
340 \@process@tablecaption
341 \@topcaptiontrue
342 \fi

Restore the meaning of \ to the one it had before the start of this environment. Also re-initialize some control-sequences

343 \global\let\\@oldcr
344 \global\let\@process@tablecaption\relax
345 \ST@trace\tw@{Ended a mpsupertabular\ifST@star*\fi}\
346 }

The definition of the ending of the supertabular* environment is simple:

347 \expandafter\let\csname endmpsupertabular\endcsname\endmpsupertabular

\ST@restore

This macro restores the original definitions of the macros that handle parbox entries and the macros that handle the end of the row.

348 \def\ST@restore{%
349 \ifx\undefined\@classix
350 \let\@tabularcr\org@tabularcr
351 \else
352 \let\@arraycr\org@tabularcr
353 \fi
354 \let\@startpbox\org@startpbox
355 \let\@endpbox\org@endpbox
356 }

\inner@tabular\inner@tabular*

In order to facilitate complete tabular environments to be in a cell of a supertabular environment we need to adapt the definition of the original environments somewhat. For the inner tabular a number of definitions needs to be restored.

357 \def\inner@tabular{%
358 \ST@restore
359 \let\\@oldcr
360 \noindent
361 \org@tabular}
362 \@namedef{inner@tabular*}%
363 \ST@restore
364 \let\\@oldcr
365 \noindent
366 \csname org@tabular*\endcsname}

\ST@cr

This macro is called by each \ inside the tabular environment. It updates the estimate of the amount of space left on the current page and starts a new page if necessary.

367 \def\ST@cr{%
368 \noalign{%
369 \ifnum\ST@pboxht<\ST@lineht
370 If there is a non-empty line, but an empty parbox, then \ST@pboxht might be non-zero, but too small thereby breaking the algorithm. Therefore we estimate the height of the line to be \ST@lineht in this case.
And we store that fact in \ST@prevht.

\global\ST@prevht\ST@lineht

\else

When the parbox was not empty we take into account its height (plus a bit extra).
\ST@trace\thr@@{Added par box with height \the\ST@pboxht}%%
\global\advance\ST@pageleft -\ST@pboxht
\global\advance\ST@pageleft -0.1\ST@pboxht
\global\advance\ST@pageleft -\ST@stretchht
\global\ST@prevht\ST@pboxht
\global\ST@pboxht\z@
\fi
\ST@toadd is the value of the optional argument of \.
\global\advance\ST@pageleft -\ST@toadd
\global\ST@toadd=\z@
\ST@trace\thr@@{Space left for tabular: \the\ST@pageleft}%%

This line is necessary because the tablehead has to be inserted *after* the following \if\else\fi-clause. For this purpose \ST@next is used by \ST@newpage. But we need to make sure that \ST@next is not undefined when \ST@newpage is not called. In the middle of tableprocessing it should be an *empty* macro (*not* \relax).
\noalign{\global\let\ST@next\@empty}%%

When the \ST@pageleft has become negative, the last row was so high that the supertabular doesn’t fit on the current page after all. In this case we will skip the current page and start at the top of the next one; otherwise TEX will move this part of the table to a new page anyway, probably with a message about an overfull \vbox.
\ifnum\ST@pageleft<\z@
\ST@skippage
\else

When there is not enough space left on the current page, we start a new page. To compute the amount of space needed we use the height of the previous line (\ST@prevht) as an estimation of the height of the next line. If we are processing a mpsupertabular we need to take the height of the footnotes into account.
\noalign{\global\@tempdima\ST@tailht
\global\advance\@tempdima\ST@prevht
\if\ST@mp
  \ifvoid\@mpfootins\else
    \global\advance\@tempdima\ht\@mpfootins
  \global\advance\@tempdima 3pt
  \fi
  \fi}
\ifnum\ST@pageleft<\@tempdima
\ST@newpage
\fi
\fi
\fi
\ST@skipfirstpart This macro skips the current page and moves the entire supertabular that has been built up sofar to the next page.
In order for this to work properly we need to adapt the value of \ST@pageleft. When this macro is called it has a negative value. We should add the height of the next page to that (\@colroom). From the result the ‘normal’ height of the supertabular should be subtracted (\@colroom - \pagetotal). This could be coded as follows:

\ST@dimen\@colroom
\advance\ST@dimen-\pagetotal
\global\advance\ST@pageleft\@colroom
\global\advance\ST@pageleft-\ST@dimen

When you examine the code you will note that \@colroom is added and subtracted. Therefore the code above can be simplified to:

\global\advance\ST@pageleft\pagetotal

Then we can set \ST@pagesofar to 0 and start the new page.

\global\ST@pagesofar\z@\newpage

Finally we make sure that this macro can only be executed once for each supertabular by changing the definition of \ST@skippage.

\global\let\ST@skippage\ST@newpage

\ST@newpage

This macro performs the actions necessary to start a new page.

\def\ST@newpage{%
  \noalign{\ST@trace\tw@{Starting new page, writing tail}}%

  Output \tabletail, close the tabular environment, close a \minipage if necessary, output all material and start a fresh new page.

  \@tabletail
  \ifST@star
    \csname endtabular*\endcsname
  \else
    \endtabular
  \fi
  \ifST@mp
    \endminipage
  \fi
  \fi

  Then we make sure that the macro \ST@skippage can no longer be executed for this supertabular by changing the definition of it.

  \global\let\ST@skippage\ST@newpage
  \newpage\calnextpageht
  \let\ST@next\@tablehead
  \ST@trace\tw@{writing head}%
  \ifST@mp
    \noindent\minipage{\columnwidth}%
    \parfillskip\ST@parfillskip
    \rightskip \ST@rightskip
    \leftskip \ST@leftskip
  \fi

  \fi

20
\noindent
\ifST@star
\expandafter\csname org@tabular*\expandafter\endcsname
\expandafter\{\expandafter\ST@wd\expandafter\expandafter\}
\expandafter\{\expandafter\ST@tableformat\}
\else
\expandafter\org@tabular\expandafter\{\expandafter\ST@tableformat\}
\fi

\package