The *supertabular* environment*

Johannes Braams and Theo Jurriens

2004/02/20

1 Introduction

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

With the original *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 *Overfull vbox* message.

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

2 User interface

The package *supertabular* has three options, they control the amount of information that is written to the `.log` file.

1. The option *errorshow* (the default) doesn't write any extra information.

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

3. The option *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.

\tablefirsthead The command `\tablefirsthead` takes one argument, it defines the contents of the first occurrence of the tabular head.

The use of this command is optional. Don’t forget to close the head by a `\`.

\tablehead The command `\tablehead` takes one argument, it defines the contents of all subsequent occurrences of the tabular head.

Don’t forget to close the head by a `\`

\tabletail The command `\tabletail` takes one argument, it defines something which should be inserted before each `\end{tabular}`, except the last.

\tablelasttail The command `\tablelasttail` takes one argument, it defines something

*This file has version number v4.1e, last revised 2004/02/20.*
which should be inserted before the last `\end{tabular}`.
The use of this command is optional.

\begin{tabular}{|ccc|}
\hline
<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>1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>81</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>256</td>
<td>24</td>
</tr>
</tbody>
</table>
\hline
\end{tabular}

\begin{flushleft}
\textit{continued on next page}
\end{flushleft}
<table>
<thead>
<tr>
<th>Number</th>
<th>Number$^2$</th>
<th>Number$^4$</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>3991680</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{4}{|l|}{\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}
\caption{This table is split across pages}
\end{center}
\begin{supertabular}{|r@{\hspace{6.5mm}}|r@{\hspace{5.5mm}}|r|r|}
\hline
1 & 1 & 1 & 1 \\
2 & 4 & 16 & 2 \\
3 & 9 & 81 & 6 \\
4 & 16 & 256 & 24 \\
\vdots \\
19 & 361 & 130321 & 1.21645100E+17 \\
20 & 400 & 160000 & 2.43290200E+18 \\
\hline
\end{supertabular}
\end{center}

Here is another example whith a p column-definition. The tablehead is the same as above. The tabletail is a double \hline; \arraystretch is set to 1.5 and the font size is \small.

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

<table>
<thead>
<tr>
<th>Number</th>
<th>Number²</th>
<th>Number⁴</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>

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>Number²</th>
<th>Number⁴</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>Number²</th>
<th>Number⁴</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*
18 and here is a long entry, where line breaks and line breaks and line breaks have to occur

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 \textit{inside} a floating-environment such as \texttt{table} as this will result in \TeX{} trying to put the whole supertabular on \textit{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 an tail.

6 The Implementation

\begin{verbatim}
\topcaption
\bottomcaption
The user-commands \texttt{\topcaption} and \texttt{\bottomcaption} set the flag \texttt{@topcaption} to determine where to put the tablecaption. The default is to put the caption on the top of the table
\end{verbatim}

\begin{verbatim}
\begin{verbatim}
\topcaption
\bottomcaption
\tablecaption
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.
\end{verbatim}
\end{verbatim}

\begin{verbatim}
\ifST@star
This switch is used in the internal macros to remember which kind of environment was started.
\end{verbatim}

\ifST@mp  This switch is used in the internal macros to remember if the tabular should be
put into a minipage.
16 \newif\ifST@mp
\ST@wd  For the supertabular* environment it is necessary to store the intended width of
the tabular.
17 \newdimen\ST@wd
\ST@rightskip\ST@leftskip\ST@parfillskip  For the mpsupertabular environments we need special versions of \leftskip,\rightskip and \parfillskip.
18 \newskip\ST@rightskip
19 \newskip\ST@leftskip
20 \newskip\ST@parfillskip
\ST@caption  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.
21 \longdef\ST@caption#1[#2]{%#3\par% 22 \addcontentsline{\csname ext@#1\endcsname}{#1}{%#1}% 23 \protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}
24 \begingroup
25 \@parboxrestore
26 \normalsize
27 \if@topcaption \vskip -10\p@ \fi
28 \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
29 \if@topcaption \vskip 10\p@ \fi
30 \endgroup}
\tablehead\tablefirsthead\tablelasttail\tabletail  \tablehead activates the new tabular \cr commands.
32 \newcommand\tablehead[1]{% 33 \gdef\@tablehead{% 34 \noalign{\global\let@savcr=\% 35 \global\let\%=\org@tabularcr}% 36 \noalign{\global\let\%=\org@tabularcr}}
37 \tablehead% 38 \newcommand\tablefirsthead[1]{% 39 \gdef\@table@first@head{#1}}
\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\% 41 \newcommand\tabletail[1]{% 42 \gdef\@tabletail{% 43 \noalign{\global\let@savcr=\%
44 \global\let\%=\org@tabularcr}% 45 \noalign{\global\let\%=\org@tabularcr}}
46 \#1%
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 \texttt{docstrip} to a .sty file.

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

\texttt{\textbackslash ST@trace} A macro that gets the trace message as its argument
\newcommand*{\ST@trace}[2]{%\begin{verbatim}
\ifnum\c@tracingst>#1\relax
\GenericWarning{(supertabular)\@spaces\@spaces}
{{(supertabular)\@spaces\@spaces}
{Package supertabular: #2}\%}
\fi}
\end{verbatim}%}

\texttt{\textbackslash ST@pageleft} 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.
\newdimen\ST@pageleft

\texttt{\textbackslash shrinkheight} A command to diminish the value of \texttt{\textbackslash ST@pageleft} if necessary.
\newcommand*{\shrinkheight}[1]{%\begin{verbatim}
\noalign{\global\advance\ST@pageleft-#1\relax}}
\end{verbatim}%

\texttt{\textbackslash setSTheight} A command to set the value of \texttt{\textbackslash ST@pageleft} if necessary.
\newcommand*{\setSTheight}[1]{%\begin{verbatim}
\noalign{\global\ST@pageleft=#1\relax}}
\end{verbatim}%

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

\texttt{\textbackslash ST@pagesofar} The register \texttt{\textbackslash ST@pagesofar} is used to store the estimate of the amount of page already filled up.
\newdimen\ST@pagesofar

\texttt{\textbackslash ST@pboxht} The measured (total) height of a parbox-argument
\newdimen\ST@pboxht

\texttt{\textbackslash ST@lineht} The estimated height of a normal line is stored in \texttt{\textbackslash ST@lineht}. The dimension register \texttt{\textbackslash ST@stretchht} is used to store the difference between the ‘normal’ line height and the line height when \texttt{\textbackslash arraystretch} has a non-standard value. This is used in the case where p-box entries are added to the tabular. The dimension register \texttt{\textbackslash ST@prevht} is used 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.
\newdimen\ST@lineht
\newdimen\ST@stretchht
\newdimen\ST@prevht
\ST@toadd  When a tabular row is ended with `\[\ldots\]` we need to temporarily store the optional argument in `\ST@toadd`.

\newdimen\ST@toadd

\ST@dimen  A private scratch dimension register.

\newdimen\ST@dimen

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

\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`.

```latex
\def\ST@tabularcr{%  
\ifnum0='\fi
\@ifstar{\ST@xtabularcr}{\ST@xtabularcr}}
\def\ST@xtabularcr{%  \@ifnextchar[\]%  \ST@argtabularcr\cr\ST@cr}}
\def\ST@argtabularcr[#1]{%  \ifnum0='\fi%  \ifdim #1>\z@\unskip\ST@xargarraycr{#1}%  \else\ST@yargarraycr{#1}%  \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`.

```latex
\def\ST@xargarraycr#1{%  \@tempdima #1\advance\@tempdima \dp@arstrutbox  \vrule@height\z@\@depth\@tempdima\@width\z@\cr
\noalign{\global\ST@toadd=#1}\ST@cr}
```

\ST@yargarraycr  Here we need to insert `\ST@cr`.

```latex
\def\ST@yargarraycr#1{%  \cr
\noalign{\vskip\global\ST@toadd=#1}\ST@cr}
```

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

```latex
\def\ST@startpbox#1{%  \bgroup\hsize#1\setbox\ST@pbox\vtop\bgroup\hsize#1\@arrayparboxrestore}
```

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

```latex
\def\ST@astartpbox#1{%  \bgroup\hsize#1\setbox\ST@pbox\vtop\bgroup\hsize#1\@arrayparboxrestore}
```
\ST@endpbox Our version of \endpbox and \aendpbox.

99 \def\ST@endpbox{%
100 \@finalstrut\arstrutbox\par\egroup
101 \ST@dimen=\ht\ST@pbox
102 \advance\ST@dimen by \dp\ST@pbox
103 \ifnum\ST@pboxht<\ST@dimen
104 \global\ST@pboxht=\ST@dimen
105 \fi
106 \ST@dimen=\z@
107 \box\ST@pbox \hfil}

108 \def\ST@aendpbox{%
109 \@finalstrut\arstrutbox\par\egroup
110 \ST@dimen=\ht\ST@pbox
111 \advance\ST@dimen by \dp\ST@pbox
112 \ifnum\ST@pboxht<\ST@dimen
113 \global\ST@pboxht=\ST@dimen
114 \fi
115 \ST@dimen=\z@
116 \unvbox\ST@pbox \egroup \hfil}

\est@lineht Estimates the height of normal line taking \arraystretch into account. Also computes the difference between a normal line and a ‘stretched’ one.

117 \def\est@lineht{%
118 \ST@lineht=\arraystretch \baselineskp
119 \global\advance\ST@lineht by 1\p@
120 \ST@stretchht=\ST@lineht\advance\ST@stretchht-\baselineskp
121 \ifdim\ST@stretchht<\z@ \ST@stretchht=\z@ \fi
122 \ST@trace\tw@{Average line height: \the\ST@lineht}
123 \ST@trace\tw@{Stretched line height: \the\ST@stretchht}
124 \}

@calfirstpageht Estimates the space left on the current page and decides whether the tabular can be started on this page or on a new page.

125 \def@calfirstpageht{%
126 \ST@trace\tw@{Calculating height of tabular on first page}%

The \TeX register \pagetotal contains the height of the page so far, the \LaTeX register \@colroom contains the height of the column.

127 \global\ST@pagesofar=\pagetotal
128 \global\ST@pageleft=\@colroom
129 \ST@trace\tw@{Height of text = \the\pagetotal; MessageBreak
130 Height of page = \the\ST@pageleft}%

When we are in twocolumn mode \TeX 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.

131 \if@twocolumn
132 \ST@trace\tw@{two column mode}%
133 \if@firstcolumn
134 \ST@trace\tw@{First column}%
135 \ifnum\ST@pagesofar > \ST@pageleft
136 \global\ST@pageleft=2\ST@pageleft
137 \ifnum\ST@pagesofar > \ST@pageleft
138 \newpage@calfirstpageht
In this case we’re in the second column, so we have to compensate for the material in the first column.

When \ST@pagesofar is smaller than \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 we end up here, \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.

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.

\begin{verbatim}
139  \ST@trace{starting new page}\\
140  \else\\
141  \ST@trace{Second column}\\
142  \global\advance\ST@pageleft -\ST@pagesofar\\
143  \global\advance\ST@pageleft -\@colroom\\
144  \fi\\
145  \ifnum\ST@pagesofar > \ST@pageleft\\
146  \ST@trace{starting new page}\\
147  \newpage\@calnextpageht\\
148  \else\\
149  \global\advance\ST@pageleft by -\ST@pagesofar\\
150  \global\ST@pagesofar\z@
151  \fi\\
152  \fi\\
153  \ifnum\ST@pagesofar > \ST@pageleft\\
154  \ST@trace{starting new page}\\
155  \newpage\@calnextpageht\\
156  \else\\
157  \global\advance\ST@pageleft by -\ST@pagesofar\\
158  \global\ST@pagesofar\z@
159  \fi\\
160  \else\\
161  \ifnum\ST@pagesofar > \ST@pageleft\\
162  \ST@trace{starting new page}\\
163  \newpage\@calnextpageht\\
164  \else\\
165  \global\advance\ST@pageleft by -\ST@pagesofar\\
166  \global\ST@pagesofar\z@
167  \fi\\
168  \fi\\
169  \ST@trace{Available height: \the\ST@pageleft}\\
\end{verbatim}
To decide when to start a new page, we need to know the vertical size of the tail of the table.

We add the average height of a line to this because when we decide to continue the tabular we need to have enough space left for one line and the tail.

Now we 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.

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

\x@supertabular 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}
\newcommand{\org@tabular*}{\expandafter\csname tabular*\endcsname}
\newcommand{\tabular*}{\expandafter\csname inner@tabular*\endcsname}
\end{verbatim}
If the caption should come at the top we insert it here.
\begin{verbatim}
\if@topcaption \process@tablecaption \fi
\end{verbatim}
Save the original definition of \texttt{\}.\texttt{\}
\begin{verbatim}
\global\let\@oldcr=\}
\end{verbatim}
Save the current value of \texttt{\baselineskip}, as we need it in the calculation of the average height of a line.
\begin{verbatim}
\def\baslineskp{\baselineskip}\
\end{verbatim}
We have to check whether \texttt{array.sty} was loaded, because some of the internal macros have different names.
\begin{verbatim}
\ifx\undefined\@classix
\let\org@tabularcr=\@tabularcr
\else
\let\org@tabularcr=\@arraycr
\fi
\end{verbatim}
When \texttt{array.sty} was loaded things are a bit different.
\begin{verbatim}
\let\org@startpbox=\@startpbox
\let\org@endpbox=\@endpbox
\let\@startpbox=\ST@startpbox
\let\@endpbox=\ST@aendpbox
\end{verbatim}
Check if the head of the table should be different for the first and subsequent pages.
\begin{verbatim}
\ifx\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 after all. Subsequent parts can not be moved; therefore we will have to switch the definition of \texttt{\ST@skippart} around.
\begin{verbatim}
\let\ST@skippage=\ST@skipfirstpart
\end{verbatim}
Now we can estimate the average line height and the height of the first page of
the \texttt{supertabular}.

\begin{verbatim}
estimation\lineht
@callfirstpageht
\noindent
\}
\end{verbatim}

\texttt{\textbackslash 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...

\begin{verbatim}
def\supertabular{% 
\ifnextchar[{{\@supertabular}}{}

\end{verbatim}

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

\begin{verbatim}
def\@supertabular[#1]{% 
def\ST@tableformat{#2}% \ST@trace{Starting a new supertabular}%

\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.

\begin{verbatim}
def\x@supertabular

Finally start a normal tabular environment.

\begin{verbatim}
expandafter\org@tabular\expandafter{\ST@tableformat}%
\tablehead}%
\end{verbatim}

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

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

\end{verbatim}

We start by saving the intended width and the preamble of the tabular*.

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

Now we can call the common code for both environments.

\begin{verbatim}
x@supertabular

And we can start a normal tabular* environment.

\begin{verbatim}
expandafter\csname org@tabular*\endcsname
expandafter{\ST@tableformat}%
\tablehead}%
\end{verbatim}
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...

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

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

\def\@mpsupertabular[#1]#2{% 
  \def\ST@tableformat{#2}\
  \ST@trace\tw@{Starting a new mpsupertabular}\% 
  Then remember that this is not a mpsupertabular* environment.

  \global\ST@starfalse
  And remember to close the minipage later.
  \global\ST@mptrue
  Since we are about to start a minipage of \columnwidth the horizontal alignment will no longer work. We have to remember the values and restore them inside the minipage.

  \ST@rightskip \rightskip
  \ST@leftskip \leftskip
  \ST@parfillskip \parfillskip
  Most of the following code is shared between the mpsupertabular and mpsupertabular* environments. So to avoid duplication it is stored in a macro.

  \x@supertabular
  Finally start a normal tabular environment.

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

We start by looking for the optional argument of the tabular environment.

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

Now we can save the intended width and the preamble of the tabular*.

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

Then we can call the common code for both environments.

\begin{macrocode}
\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}
\endsupertabular*\endsupertabular
\endmacrocode

This closes the environments \texttt{supertabular} and \texttt{supertabular*}.

\def\endsupertabular{\ifx\@table@last@tail\undefined\@tabletail\else\@table@last@tail\fi\csname endtabular\ifST@star*\fi\endcsname\ST@restore}

Check if we have to insert a caption and restore to default behaviour of putting captions at the top.
\if@topcaption\else\@process@tablecaption\@topcaptiontrue\fi

Restore the meaning of \verb|\| to the one it had before the start of this environment.
Also re-initialize some control-sequences
\global\let\@oldcr\relax\ST@trace\tw@{Ended a supertabular\ifST@star*\fi}\

The definition of the ending of the \texttt{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\endminipage
\ST@restore}
\endmacrocode

This closes the environments \texttt{mpsupertabular} and \texttt{mpsupertabular*}.

\def\endmpsupertabular{\ifx\@table@last@tail\undefined\@tabletail\else\@table@last@tail\fi\csname endtabular\ifST@star*\fi\endcsname\endminipage
\ST@restore}
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

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  }

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}

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.
371      \global\advance\ST@pageleft -\ST@lineht
372      \global\advance\ST@pageleft -\ST@lineht
373  }
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).
\global\ST@lineht\ST@pboxht
\else
\fi
\ST@toadd is the value of the optional argument of \.
\global\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 \fi\else\fi-clause. For this purpose \ST@nxt is used by \ST@newpage. But we need to make sure that \ST@nxt is not undefined when \ST@newpage is not called.
In the middle of tableprocessing it should be an *empty* macro (*not* \relax).
(15.2.91)
\noalign{\global\let\ST@nxt@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@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 substracted (\@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 substracted. 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@pagesofar0
\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.

  \ifST@star\csname endtabular*\endcsname\else\endtabular\fi
  \ifST@mp\endminipage\fi
  \ST@trace tw@{writing head}%

  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
\noindent\ifST@star
\expandafter\csname org@tabular\expandafter\endcsname
\expandafter{\expandafter\ST@wd\expandafter}\%
\expandafter{\ST@tableformat}\%
\else
\expandafter\org@tabular\expandafter{\ST@tableformat}\%
\fi
\ (/package)