The \texttt{supertabular} environment\footnote{This file has version number v4.1d, last revised 1999/08/07.}

Johannes Braams and Theo Jurriens

1999/08/07

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

\topcaption These commands all take the same arguments as \LaTeX’s standard \caption command. They provide a caption for the super-table, either at the top or at the bottom of the table. When \tablecaption is used the caption will be placed at the default location, which is at the top.

\bottomcaption The environments supertabular and supertabular* can be used much like the standard \LaTeX environments tabular and tabular*.

\tablecaption The environments mpsupertabular and mpsupertabular* work like the supertabular and supertabular* environments but put each page into a \minipage first. Thus it is possible to have footnotes inside a mpsupertabular. The footnotetext is printed at the end of each page.

\shrinkheight The allowed maximum height of a part of the supertabular on a page can be adjusted using the command \shrinkheight. It takes one argument, the length with which to shrink (positive value) or grow (negative value) the allowed height.

3 Weak points

- When the material of a normal entry (not a p-arg) becomes larger than the estimated \ST@lineht, overfull \vboxes 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{...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 Examples

Here is an example of a supertabular. You will find the definitions after the table.

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

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>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}{\textbf{Number}} & \multicolumn{1}{|c}{\textbf{Number$^2$}} & \multicolumn{1}{|c}{\textbf{Number$^4$}} & \multicolumn{1}{|c|}{\textbf{Number!}} \\
\hline
\tablehead{\hline
\multicolumn{4}{|l|}{\small\textit{continued from previous page}} \\
\hline
\multicolumn{1}{|c}{\textbf{Number}} & \multicolumn{1}{|c}{\textbf{Number$^2$}} & \multicolumn{1}{|c}{\textbf{Number$^4$}} & \multicolumn{1}{|c|}{\textbf{Number!}} \\
\hline
\tabletail{\hline
\multicolumn{4}{|r|}{\small\textit{continued on next page}} \\
\hline}
\tablelasttail{\hline}
\bottomcaption{This table is split across pages}
\end{center}
Here is another example with a \( p \) 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.

\begin{supertabular}{|r@{\hspace{6.5mm}}|r@{\hspace{5.5mm}}|r|r|}
1 & 1 & 1 & 1 \\
2 & 4 & 16 & 2 \\
3 & 9 & 81 & 6 \\
4 & 16 & 256 & 24 \\
... \\
19 & 361 & 130321 & 1.21645100E+17 \\
20 & 400 & 160000 & 2.43290200E+18 \\
\end{supertabular}

\begin{center}
\end{center}

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|l|}
\hline
Number & Number\(^2\) & Number\(^4\) & Number! \\
\hline
1 & 1 & 1 & here is a relative short entry \\
2 & 1 & 1 & and here is a long entry, where line breaks and line breaks and line breaks have to occur \\
3 & 1 & 1 & and here is a long entry, where line breaks and line breaks and line breaks have to occur \\
4 & 1 & 1 & and here is a long entry, where line breaks and line breaks and line breaks have to occur \\
5 & 1 & 1 & here is a relative short entry \\
6 & 1 & 1 & and here is a long entry, where line breaks and line breaks and line breaks have to occur \\
7 & 1 & 1 & and here is a long entry, where line breaks and line breaks and line breaks have to occur \\
8 & 1 & 1 & and here is a long entry, where line breaks and line breaks and line breaks have to occur \\
9 & 1 & 1 & and here is a long entry, where line breaks and line breaks and line breaks have to occur \\
\hline
\end{tabular}
\end{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>Number²</th>
<th>Number⁴</th>
<th>Number!</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1</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>
</tr>
<tr>
<td>11</td>
<td>1</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>
</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>1 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>1 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>1 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>1 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>1 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>1 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></td>
<td>1 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></td>
<td>1 here is a relative short entry</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td></td>
<td>1 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></td>
<td>1 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></td>
<td>1 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></td>
<td>1 and here is a long entry, where line breaks and line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td></td>
<td>1 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></td>
<td>1 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></td>
<td>1 here is a relative short entry</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td></td>
<td>1 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></td>
<td>1 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></td>
<td>1 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></td>
<td>1 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></td>
<td>1 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
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 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 \vbox messages.
- Sometimes the last page of the supertabular contains just an empty head and tail.

6 The Implementation

\begin{verbatim}
1 ⟨∗package⟩
2 \newcount@c@tracingst
3 \DeclareOption{errorshow}{\c@tracingst\z@}
4 \DeclareOption{pageshow}{\c@tracingst\tw@}
5 \DeclareOption{debugshow}{\c@tracingst5\relax}
6 \ProcessOptions

\topcaption \bottomcaption

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

\begin{verbatim}
7 \newif@if@topcaption \@topcaptiontrue
8 \def\topcaption{\@topcaptiontrue\tablecaption}
9 \def\bottomcaption{\@topcaptionfalse\tablecaption}
\end{verbatim}

\tablecaption

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

\begin{verbatim}
10 \long\def\tablecaption{%
11 \refstepcounter{table}\dblarg{\@tablecaption}}
12 \long\def\@tablecaption[#1]{%'
13 \long\gdef@process@tablecaption{\ST@caption{table}[#1][#2]}
14 \global\let@process@tablecaption\relax
\end{verbatim}

\ifST@star

This switch is used in the internal macros to remember which kind of environment was started.

\begin{verbatim}
15 \newif@ifST@star
\end{verbatim}
\end{verbatim}

\end{verbatim}

\begin{table}
\centering
\begin{tabular}{|l|l|l|l|}
\hline
Number & Number & Number & Number!
\hline
18 & 1 & 1 and here is a long entry, where line breaks and line breaks and line breaks have to occur
\hline
\end{tabular}
\end{table}

\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 \long\def\ST@caption#1[#2]#3{\par%
22 \addcontentsline{\csname ext@#1\endcsname}{#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
\tablehead activates the new tabular \cr commands.
31 \newcommand\tablehead[1]{%  
32 \gdef\@tablehead{%  
33 \noalign{%  
34 \global\let@savcr=\%
35 \global\let=\org@tabularcr\%
36 #1%
37 \noalign{%\global\let=\%\@savcr}}}
38 \tablehead{}  
39 \newcommand\tablefirsthead[1]{%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
\\ act just like a \@tabularcr inside this tail to prevent the loop. Save and
restore the value of \\n
41 \newcommand\tabletail[1]{%  
42 \gdef\@tabletail{%  
43 \noalign{%  
44 \global\let@savcr=\%  
45 \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 .\texttt{sty} file.

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

\ST@trace A macro that gets the trace message as its argument

\begin{verbatim}
\newcommand\ST@trace[2]{%
  \ifnum\c@tracingst#1\relax
  \GenericWarning{(supertabular)@spaces}{Package supertabular: #2}%
  \fi
}\end{verbatim}

\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.

\begin{verbatim}
\newdimen\ST@pageleft
\end{verbatim}

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

\begin{verbatim}
\newcommand*\shrinkheight[1]{%
\noalign{\global\advance\ST@pageleft-#1\relax}}
\end{verbatim}

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

\begin{verbatim}
\newcommand*\setSTheight[1]{%
\noalign{\global\ST@pageleft=#1\relax}}
\end{verbatim}

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

\begin{verbatim}
\newdimen\ST@headht
\newdimen\ST@tailht
\end{verbatim}

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

\begin{verbatim}
\newdimen\ST@pagesofar
\end{verbatim}

\ST@pboxht The measured (total) height of a parbox-argument

\begin{verbatim}
\newdimen\ST@pboxht
\end{verbatim}

\ST@lineht 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 \texttt{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 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.

\begin{verbatim}
\newdimen\ST@lineht
\newdimen\ST@stretchht
\newdimen\ST@prevht
\end{verbatim}
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[\%{
79 \ST@argtabularcr}%
80 \fi}

81 \def\ST@argtabularcr[#1]{% 
82 \ifnum0='{\fi}
83 \ifdim #1>\z@ 
84 \unskip\ST@xargarraycr{#1}
85 \else
86 \ST@yargarraycr{#1}
87 \fi}

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 \@tempdima #1\advance\@tempdima \dp \@arstrutbox 
90 \vrule \@height\z@ \@depth\@tempdima \@width\z@ \cr 
91 \noalign{\global\ST@toadd=#1}\ST@cr}

Here we need to insert `\ST@cr`

92 \def\ST@yargarraycr#1{% 
93 \cr\noalign{\vskip1\global\ST@toadd=#1}\ST@cr}

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

95 \setbox\ST@pbox\vtop\bgroup\hsize#1\@arrayparboxrestore}

Our version of `@astartpbox`.

96 \def\ST@astartpbox#1{% 
97 \bgroup\hsize#1%
98 \setbox\ST@pbox\vtop\bgroup\hsize#1\@arrayparboxrestore}
Our version of \end{pbox} and \end{pbox}.

\begin{pbox}
\def\ST@endpbox{\par\egroup
\ST@dimen=\ht\ST@pbox
\advance\ST@dimen by \dp\ST@pbox
\ifnum\ST@pboxht<\ST@dimen
\global\ST@pboxht=\ST@dimen
\fi
\ST@dimen=\z@ \box\ST@pbox\hfil}
\end{pbox}

\begin{pbox}
\def\ST@aendpbox{\par\egroup
\ST@dimen=\ht\ST@pbox
\advance\ST@dimen by \dp\ST@pbox
\ifnum\ST@pboxht<\ST@dimen
\global\ST@pboxht=\ST@dimen
\fi
\ST@dimen=\z@ \unvbox\ST@pbox\egroup\hfil}
\end{pbox}

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

\begin{pbox}
\def\estimate@lineht{\ST@lineht=\arraystretch \baselineskp
\global\advance\ST@lineht by \z@ \ST@stretchht\ST@lineht\advance\ST@stretchht-\baselineskp
\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}%
}
\end{pbox}

\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.

\begin{pbox}
\def\calfirstpageht{\ST@trace@tw@{Calculating height of tabular on first page}%
\global\ST@lineht\pagetotal \global\@colroom\pagetotal
\ST@trace@tw@{Height of text = \the\pagetotal; \MessageBreak
Height of page = \the\@colroom}%
\if@twocolumn\ST@trace@tw@{two column mode}%
\if@firstcolumn\ST@trace@tw@{First column}%
\ifnum\ST@lineht > \ST@lineht
\newpage\calfirstpageht
\else
\fi
\fi
\fi
\fi
\end{pbox}
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.

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.
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.

\begin{supertabular}
\end{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}
\expandafter\let\csname org@tabular*\expandafter\endcsname\csname tabular*\endcsname
\expandafter\let\csname tabular*\expandafter\endcsname\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{\textbackslash \textbackslash}.

\begin{Verbatim}
\global\let\@oldcr=\textbackslash\textbackslash
\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\baselineskp{\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
\let\@tabularcr\ST@tabularcr
\let\org@startpbox\@startpbox
\let\org@endpbox\@endpbox
\let\@@startpbox\ST@startpbox
\let\@@endpbox\ST@aendpbox
\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
\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{\textbackslash \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 supertabular.

\begin{verbatim}
\estimate@lineht
@calfirstpageht
\noindent
\end{verbatim}

\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}}%}
\@supertabular[#1]#2{%\def\ST@tableformat{#2}\ST@trace\tw@{Starting a new supertabular}\global\ST@starfalse\global\ST@mpfalse
Then remember that this is not a supertabular* environment.
\global\ST@starfalse
Don't use minipages.
\global\ST@mpfalse
Most of the following code is shared between the supertabular and 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}
\end{verbatim}

\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}\]}%
\@supertabular*#1[#2]#3{%\ST@trace\tw@{Starting a new supertabular*}\global\ST@startrue\global\ST@mpfalse
We start by saving the intended width and the preamble of the tabular*.
@namedef{supertabular*}#1[#2]#3{%\ST@trace\tw@{Starting a new supertabular*}\def\ST@tableformat{#3}\ST@wd=#1relax\global\ST@startrue\global\ST@mpfalse
Now we can call the common code for both environments.
\x@supertabular
And we can start a normal tabular* environment.
\expandafter\csname org@tabular*\expandafter\endcsname\expandafter{\expandafter\ST@tableformat}\@@tablehead}
\end{verbatim}
\texttt{\textbackslash mpsupertabular} 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...

\begin{verbatim}
\def\mpsupertabular{%
  \@ifnextchar[{{\@mpsupertabular}{}}
}
\end{verbatim}

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

\begin{verbatim}
\def\@mpsupertabular[#1]#2{%
  \def\ST@tableformat{#2}%
  \ST@trace\tw@{Starting a new mpsupertabular}%
  \global\ST@starfalse
  \global\ST@mptrue
  \ST@rightskip \rightskip
  \ST@leftskip \leftskip
  \ST@parfillskip \parfillskip
  \global\ST@wd=#1\relax
  \global\ST@startrue
  \global\ST@mptrue
  \ST@rightskip \rightskip
  \ST@leftskip \leftskip
  \ST@parfillskip \parfillskip
  \global\ST@starfalse
  \global\ST@mptrue
  \ST@rightskip \rightskip
  \ST@leftskip \leftskip
  \ST@parfillskip \parfillskip
  \noindent\expandafter\org@tabular\expandafter{\ST@tableformat}%
  \@@tablehead}
\end{verbatim}

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

\begin{verbatim}
\x@supertabular
\end{verbatim}

Finally start a normal \texttt{\textbackslash tabular} environment.

\begin{verbatim}
\minipage{\texttt{\textbackslash columnwidth}}%
\parfillskip\ST@parfillskip
\rightskip \ST@rightskip
\leftskip \ST@leftskip
\noindent\expandafter\org@tabular\expandafter{\ST@tableformat}%
\@@tablehead
\end{verbatim}

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

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

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

\begin{verbatim}
\@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
  \global\ST@startrue
  \global\ST@mptrue
  \ST@rightskip \rightskip
  \ST@leftskip \leftskip
  \ST@parfillskip \parfillskip
\end{verbatim}
Then we can call the common code for both environments.

\begin{supertabular}
% And we can start a normal \textsf{tabular*} environment.
\begin{tabular*}{\columnwidth}{p{1\textwidth}}
\end{tabular*}

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

\end{supertabular}
\end{supertabular*}

The definition of the ending of the \texttt{supertabular*} environment is simple:

\begin{endmpsupertabular}
\end{endmpsupertabular*}

This closes the environments \texttt{mpsupertabular} and \texttt{mpsupertabular*}.
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 \ to the one it had before the start of this environment. Also re-initialize some control-sequences
\global\let\@oldcr
\global\let\@process@tablecaption\relax
\ST@trace{tw@{Ended a mpsupertabular\ifST@star*\fi}}%
}

The definition of the ending of the supertabular* environment is simple:
\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.
\def\ST@restore{%
\ifx\undefined\@classix
\let\@tabularcr\org@tabularcr
\else
\let\@arraycr\org@tabularcr
\fi
\let\@startpbox\org@startpbox
\let\@endpbox\org@endpbox
}\ST@cr

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.
\def\inner@tabular{%
\ST@restore
\let\\\@oldcr
\noindent
\org@tabular}
\def\inner@tabular*{%
\ST@restore
\let\\\@oldcr
\noindent
\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.
\def\ST@cr{%
\noalign{%
\ifnum\ST@pboxht<\ST@lineht
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.
\global\advance\ST@pageleft -\ST@lineht
}
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}\
}
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 \texttt{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
\else
\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 \texttt{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
\else
\ST@skippage
\else
This line is necessary because the tablehead has to be inserted *after* the \if\else\fi-clause. For this purpose \ST@next is used. In the middle of tableprocessing it should be an *empty* macro (*not* \relax). (15.2.91)
\noalign{\global\let\ST@next\@empty}\
\fi
\fi
\ST@next}
\ST@skippage
This macro skips the current page and moves the entire supertabular that has been built up sofar to the next page.
\def\ST@skippage{%
403 \noalign{\% \ST@trace\tw@{Tabular too high, moving to 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 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\noindent\minipage{\columnwidth}\parfillskip\ST@parfillskip\rightskip \ST@rightskip\leftskip \ST@leftskip\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

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