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.

\[\texttt{\textbackslash tablefirsthead}\] The command \texttt{\textbackslash 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 `\`.

\[\texttt{\textbackslash tablehead}\] The command \texttt{\textbackslash tablehead} takes one argument, it defines the contents of all

*This file has version number v4.1g, last revised 2020/02/02.
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 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

\tablecaption

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

supertabular* 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 `\vbox`s 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 pagebreaks when footnotes appear in the lower part of the tabular.
4 Examples

Here is an example of a \texttt{supertabular}. First, here is (part of) the user input for the table below:

\begin{center}
\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 \[5mm\]
... 19 & 361 & 130321 & 1.21645100E+17\ \\ 20 & 400 & 160000 & 2.43290200E+18\end{supertabular}
\end{center}

Then the table should be split across the page boundary:

\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 \[5mm\]
... 19 & 361 & 130321 & 1.21645100E+17\ \\ 20 & 400 & 160000 & 2.43290200E+18\end{supertabular}

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

\textit{continued on next page}
Table 1: This table is split across pages

<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>2.09227890E+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>

Here is another example whith 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 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>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}
<table>
<thead>
<tr>
<th>Number</th>
<th>Number(^2)</th>
<th>Number(^4)</th>
<th>Number(^!)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where 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 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 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 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 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 have to occur</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>and here is a long entry, where 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 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 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 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 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 have to occur</td>
</tr>
</tbody>
</table>

\(continued\ on next page\)

5
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>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>here is a long entry, where line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>here is a long entry, where line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>here is a long entry, where 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>here is a long entry, where line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>here is a long entry, where line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>here is a long entry, where line breaks and line breaks have to occur</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>here is a long entry, where line breaks and line breaks have to occur</td>
</tr>
</tbody>
</table>
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 an tail.

<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 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>
<tr>
<td>18</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>
6 The Implementation

First we define a few options that control the level of tracing output this package delivers. The option \texttt{errorshow} is the default situation.

\begin{verbatim}
\newcount\c@tracingst
\DeclareOption{errorshow}{{c@tracingst\z@}}
\DeclareOption{pageshow}{{c@tracingst\thr@@}}
\DeclareOption{debugshow}{{c@tracingst\f@l@x}}
\ProcessOptions
\end{verbatim}

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

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

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

\begin{verbatim}
\newif\ifST@star\ifST@star
\ifST@mp\ifST@mp
\ST@wd\ST@rightskip\ST@leftskip\ST@parfillskip
\end{verbatim}

For the \texttt{supertabular*} environment it is necessary to store the intended width of the tabular.

\begin{verbatim}
\newdimen\ST@wd
\ST@rightskip\ST@leftskip\ST@parfillskip
\end{verbatim}

For the \texttt{mpsupertabular} environments we need special versions of \texttt{\leftskip}, \texttt{\rightskip} and \texttt{\parfillskip}.

\begin{verbatim}
\ST@captionroom
\end{verbatim}

When a supertabular is preceded by a caption that fact might not yet be visible in the amount of space occupied on the page so far. Therefore we include the possibility to reduce the height of the first part of the supertabular. In order to
this we need a macro that indicates a caption has been put in front of the table. We do this to reduce the risk that the first part of the table is too high after all and is pushed onto the next page due to an overfull \vbox condition.

\def\ST@captionroom{\z@}

\ST@caption This is a redefinition of LaTeX's \caption, \@makecaption is called within a group so as not to return to \texttt{\normalsize} globally. Also a fix is made for the ‘feature’ of the \@makecaption of the document class \texttt{article} and friends that a caption \texttt{always} gets a \texttt{\vskip 10pt} at the top and \texttt{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}{\protect\numberline{\csname the#1\endcsname}{\ignorespaces #2}}\begingroup\@parboxrestore\normalsize\if@topcaption \vskip -10\p@\fi\@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par\if@topcaption \vskip 10\p@\gdef\ST@captionroom{20\p@}\fi\endgroup}

\tablehead \tablefirsthead \tablehead \tablefirsthead activates the new tabular \texttt{\cr} commands.

\newcommand\tablehead[1][]\{%\gdef@tablehead\%\noalign{\global\let\@savcr=\\global\let\=\org@tabularcr}\%\#1\%\noalign{\global\let\=\@savcr}\} \tablehead{}

It’s possible to specify a different tablehead for the first ‘part’ of the table. That only needs to be used once so it ‘undefines’ itself at the end. That way we make sure that it doesn’t accidentally get used for a second supertabular in the document.

\newcommand\tablefirsthead[1][]\{%\gdef@table@first@head\%\noalign{\global\let\@savcr=\\global\let\=\org@tabularcr}\%\#1\%\noalign{\global\let\=\@savcr}\global\let\@table@first@head\undefined\} \tablefirsthead{}

\tabletail \tablelasttail \tabletail If the user uses an extra amount of tabular-data (like \texttt{\multicolumn}) in \tablelasttail \texttt{\TeX} starts looping because of the definition of \texttt{\ST@cr}. So make
It’s possible to specify a different tabletail for the last ‘part’ of the table. That only needs to be used once so it ‘undefines’ itself at the end. That way we make sure that it doesn’t accidentally get used for a second supertabular in the document.

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.

A macro that gets the trace message as its argument

A variant of \texttt{ST@trace} that can be called from within \texttt{\} as that command is looking for an optional argument will end up scanning the next line.

But because this variant is called from within \texttt{\} we need to save the current input linenumber before \LaTeX{} starts scanning for the optional argument. If we don’t, the reported linenumber depends on whether or not the optional argument is present...
Within `\ST@trace@cr` we can then locally modify `\on@line` to use this saved line number.

```latex
\newcommand\ST@trace@cr[2]{%
  \ifnum\c@tracingst>#1\relax
  \begingroup
  \edef\on@line{ on input line \ST@LineNo}\
  \GenericWarning{(supertabular)@spaces@spaces}
  \PackageWarning{supertabular: #2}\
  \endgroup
  \fi
}\
```

\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.

```latex
\newdimen\ST@pageleft
```

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

```latex
\newcommand*{\shrinkheight}[1]{% 
  \noalign{\global\advance\ST@pageleft-#1\relax}}
```

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

```latex
\newcommand*{\setSTheight}[1]{% 
  \noalign{\global\ST@pageleft=#1\relax}}
```

\ST@headht  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)

```latex
\newdimen\ST@headht
\newdimen\ST@tailht
```

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

```latex
\newdimen\ST@pagesofar
```

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

```latex
\newdimen\ST@pboxht
```

\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 \( \text{arraystretch} \) 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.

```latex
\newdimen\ST@lineht
\newdimen\ST@stretchht
\newdimen\ST@prevht
```
When a tabular row is ended with `\[...\]` we need to temporarily store the optional argument in `\ST@toadd`.

`\newdimen\ST@toadd`  
A private scratch dimension register.

`\newdimen\ST@dimen`  
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`.

```latex
\def\ST@tabularcr{\ifnum0='{\fi\ST@save@lineno\@ifstar{\ST@xtabularcr}{\ST@xtabularcr}}
\def\ST@xtabularcr{\@ifnextchar[\ST@argtabularcr}{\ST@argtabularcr{\ifnum0='{\fi}\cr\ST@cr}}
\def\ST@argtabularcr[#1]{\ST@xargarraycr#1}\ST@yargarraycr{#1}\ST@cr}
```

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}
```

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

```latex
\def\ST@yargarraycr[#1]{\cr\noalign{\vskip #1\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.

```latex
\def\ST@startpbox#1{To achieve our goal we need to save the text in box.
\setbox\ST@pbox\vtop{\bgroup\hsize#1\arrayparboxrestore}}
```

12
\begin{verbatim}
\texttt{\textbackslash ST@astartpbox} \texttt{Our version of \textbackslash@astartpbox.}  
\texttt{131 \def\ST@astartpbox#1{\%}  
132 \bgroup\hsize#1\%  
133 \setbox\ST@pbox\vtop\bgroup\hsize#1\@arrayparboxrestore}\texttt{\textbackslash ST@endpbox} \texttt{Our version of \textbackslash@endpbox and \textbackslash@aendpbox.}  
\texttt{134 \def\ST@endpbox{\%}  
135 \@finalstrut\@arstrutbox\par\egroup  
136 \ST@dimen=\ht\ST@pbox  
137 \ifnum\ST@pboxht<\ST@dimen\global\ST@pboxht=\ST@dimen\fi  
138 \ST@dimen=\z@  
139 \box\ST@pbox\hfil}  
\texttt{\ST@aendpbox} \texttt{143 \def\ST@aendpbox{\%}  
144 \@finalstrut\@arstrutbox\par\egroup  
145 \ST@dimen=\ht\ST@pbox  
146 \ifnum\ST@pboxht<\ST@dimen\global\ST@pboxht=\ST@dimen\fi  
147 \ST@dimen=\z@  
148 \unvbox\ST@pbox\egroup\hfil}\texttt{\ST@compute@lineht} \texttt{The height of a line in an array environment can be computed as:}  
\begin{itemize}  
\item the height of the strutbox \texttt{\ht\strutbox} (plus \texttt{\extrarowheight} when the \texttt{array} package is loaded),  
\item multiplied by \texttt{\arraystretch},  
\item plus the depth of the strutbox (\texttt{\dp\strutbox}) multiplied by \texttt{\arraystretch}.  
\end{itemize}  
\texttt{152 \def\ST@compute@lineht{\%}  
153 \ST@lineht=\ht\strutbox  
154 \ifx\extrarowheight\undefined\else  
155 \advance\ST@lineht by \extrarowheight\fi  
156 \if\fi  
157 \ST@lineht=\arraystretch\ST@lineht  
158 \advance\ST@lineht by \arraystretch\dp\strutbox  
159 \ST@trace\tw@{Normal Line height: \the\ST@lineht}  
160 }  
\texttt{\estimate@lineht} \texttt{Estimates the height of normal line taking \texttt{\arraystretch} into account. Also computes the difference between a normal line and a ‘stretched’ one. This macro will be removed in a future release.}  
\texttt{161 \def\estimate@lineht{\%}  
162 \ST@lineht=\arraystretch \baselineskp  
163 \global\advance\ST@lineht by 1\p@  
13}
\end{verbatim}
\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.
\def\calfirstpageht{%
  \ST@trace\tw@{Calculating height of tabular on first page}%
  The $\TeX$ register \pagetotal contains the height of the page sofar, 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 $\TeX$ 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 $\ST@pageleft$.
  \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
        \ifnum\ST@pagesofar > \ST@pageleft
          \newpage\calfirstpageht
        \else
          \global\advance\ST@pageleft by -\ST@pagesofar
          \global\ST@pagesofar\z@
        \fi
      \else
        \global\advance\ST@pageleft by -\ST@pagesofar
        \global\ST@pagesofar\z@
      \fi
    \else
      \ifnum\ST@pagesofar > \ST@pageleft
        \newpage\calfirstpageht
      \else
        \global\advance\ST@pageleft\@colroom
      \fi
    \fi
  \else
    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.
    \else
      \global\advance\ST@pageleft by -\ST@pagesofar
      \global\ST@pagesofar\z@
    \fi
    \else
      When we end up here, $\TeX$ has already decided it had enough material for the
      first column and is building the second column.
      \global\ST@trace\tw@{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.

When a caption precedes the first part of the tabular we need to reduce the available height on the page by \ST@captionroom.

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.

\advance\ST@tailht by \ST@lineht
\ST@trace\tw@{Height of tail: \the\ST@tailht}\
\ST@trace\tw@{Maximum height of tabular: \the\ST@pageleft}
\@tempdima\ST@headht
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.

\advance\@tempdima\ST@lineht
\advance\@tempdima\ST@tailht
\ST@trace\tw@{Minimum height of tabular: \the\@tempdima}
\ifnum\@tempdima>\ST@pageleft
\ST@trace\tw@{starting new page}\
\newpage\@calnextpageht
\fi
Take the height of the table into account, so subtract it from the available height. We need to do it like this because the \ inside the definition of \@tablehead have their normal definition.

\advance\ST@pageleft-\ST@headht
}
\@calnextpageht This calculates the maximum height of the tabular on all subsequent pages of the supertabular environment.
\def\@calnextpageht{%\ST@trace\tw@{Calculating height of tabular on next page}\
\global\ST@pageleft\@colroom
\global\ST@pagesofar=\z@\ST@trace\tw@{Maximum height of tabular: \the\ST@pageleft}\
Take the height of the head into account by subtracting it from the available space.
\advance\ST@pageleft-\ST@headht
}
\x@supertabular The body of the beginning of both environments is stored in a single macro as the code is shared.
\def\x@supertabular{%\ST@trace\tw@{Calculating height of tabular} and then make it point to \inner@tabular. This is done to enable supertabular cells to contain a tabular environment without getting unexpected results when the supertabular would be split across this inner tabular environment.
\let\org@tabular\tabular
\let\tabular\inner@tabular

16
The same needs to be done for the \texttt{tabular*} environment. The coding is slightly more verbose.

257 \expandafter\let
258 \csname org@tabular*\endcsname
259 \csname tabular*\endcsname
260 \expandafter\let\csname tabular*\endcsname
261 \csname inner@tabular*\endcsname

If the caption should come at the top we insert it here.

262 \if@topcaption \@process@tablecaption \fi
Save the original definition of \texttt{\}.  

263 \global\let\@oldcr=\}

Save the current value of \texttt{\baselineskip}, as we need it in the calculation of the average height of a line.

264 \def\baselineskp{\baselineskip}  

We have to check whether \texttt{array.sty} was loaded, because some of the internal macros have different names.

265 \ifx\undefined\@classix  

Save old \texttt{\@tabularcr} and insert the definition of \texttt{ST\@tabularcr}.

266 \let\org@tabularcr\@tabularcr
267 \let\@tabularcr\ST@tabularcr

Activate the new parbox algorithm.

268 \let\org@startpbox=\@startpbox 
269 \let\org@endpbox=\@endpbox 
270 \let\@startpbox=\ST@startpbox 
271 \let\@endpbox=\ST@endpbox 
272 \else

When \texttt{array.sty} was loaded things are a bit different.

273 \let\org@tabularcr\arraycr
274 \let\@tabularcr\ST@tabularcr
275 \let\org@startpbox=\@startpbox 
276 \let\org@endpbox=\@endpbox 
277 \let\@startpbox=\ST@startpbox 
278 \let\@endpbox=\ST@endpbox 
279 \fi

Check if the head of the table should be different for the first and subsequent pages.

280 \ifx\@table@first@head\undefined  
281 \let\@tablehead=\@tablehead
282 \else  
283 \let\@tablehead=\@table@first@head
284 \fi

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; therefor 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 \texttt{supertabular}.

\let\ST@skippage\ST@skipfirstpart

\ST@compute@lineht
\@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}}%}

We can now save the preamble of the tabular in a macro.
\def\@supertabular[#1]#2{%
\def\ST@tableformat{#2}%
\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 \texttt{tabular} environment.
\expandafter\org@tabular\expandafter{\ST@tableformat}%
\@@tablehead}
\supertabular*

We start by looking for the optional argument of the tabular environment.
\namedef{supertabular*}{#1}%
\@ifnextchar[{{\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 tabular* environment.

\expandafter\csname org@tabular*\expandafter\endcsname
\expandafter{\expandafter\ST@wd\expandafter}%
\expandafter{\ST@tableformat}%
\@@tablehead}%
\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...

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

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

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}
\mpsupertabular*
We start by looking for the optional argument of the tabular environment.
\namedef{mpsupertabular*}#1{%
@ifnextchar{{\nameuse{(mpsupertabular*)}[#1]}%}{\nameuse{(mpsupertabular*)}[#1][]}%
}
Now we can save the intended width and the preamble of the \texttt{tabular*}.

\begin{verbatim}
\namedef{\mpsupsupertabular*}{#1}[#2]{% \\
  \ST@trace{tw@}{Starting a new \mpsupsupertabular*}% \\
  \def\ST@tableformat{#3}% \\
  \ST@wd=#1\relax \\
  \global\ST@startrue \\
  \global\ST@mptrue \\
  \ST@rightskip \rightskip \\
  \ST@leftskip \leftskip \\
  \ST@parfillskip \parfillskip \\

  Then we can call the common code for both environments.

  \x@supertabular \\
  \% And we can start a normal \texttt{tabular*} environment.

  \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*
\end{verbatim}

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

\begin{verbatim}
\def\endsupertabular{% \\
  \ifx\@table@last@tail\undefined \\
  \@tabletail \\
  \else \\
  \@table@last@tail \\
  \fi \\
  \csname endtabular\if\ST@star\star\fi\endcsname \\

  Restore the original definition of \texttt{@tabularcr} \\
  \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 \texttt{\textbackslash} to the one it had before the start of this environment.

  Also re-initialize some control-sequences \\
  \global\let\textbackslash@oldcr \\
  \global\let\@process@tablecaption@relax \\
  \ST@trace{tw@}{Ended a supertabular\if\ST@star\star\fi}%
\end{verbatim}

20
The definition of the ending of the \texttt{supertabular*} environment is simple:

```latex
\end{supertabular*}
```

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

```latex
\def\endsupertabular*{\endmpsupertabular*}
```

```
\ifx\@table@last@tail\undefined
  \@tabletail
\else
  \@table@last@tail
\fi
\csname endtabular\ifST@star*\fi\endcsname
\endminipage
```

```
\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.

```latex
\def\ST@restore{\ifx\undefined\@classix
  \let\@tabularcr\org@tabularcr
  \let\@arraycr\org@tabularcr
  \let\@startpbox\org@startpbox
  \let\@endpbox\org@endpbox
  \ST@trace\tw@{Ended a mpsupertabular\ifST@star*\fi}\%
  )
  \ST@restore}
```

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

```latex
\endsupertabular*
```

```
\if\@topcaption\else
  \@process@tablecaption
  \@topcaptiontrue
\fi
```

```
\if\@topcaption\else
  \@process@tablecaption
  \@topcaptiontrue
\fi
```

```
\def\inner@tabular*{\inner@tabular}
```

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

```latex
\def\inner@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.

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`.

When the parbox was not empty we take into account its height (plus a bit extra).

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`).

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.

431 \ifnum\ST@pageleft<\z@

432 \ST@skippage

433 \else

When there is not enough space left on the current page, we start a new page. To compute the amount of space need 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.

434 \noalign{\global\@tempdima\ST@tailht

435 \global\advance\@tempdima\ST@prevht

436 \ifST@mp

437 \ifvoid\@mpfootins\else

438 \global\advance\@tempdima\ht\@mpfootins

439 \global\advance\@tempdima 3pt

440 \fi

441 \fi}

442 \ifnum\ST@pageleft<\tempdim a

443 \ST@newpage

444 \fi

445 \fi

446 \ST@next}

\ST@skipfirstpart This macro skips the current page and moves the entire supertabular that has been built up sofar to the next page.

447 \def\ST@skipfirstpart{%

448 \noalign{\ST@trace\tw@{Tabular too high, moving to next page}%

449 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. Therefor 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.

451 \global\ST@pagesofar\z@

452 \newpage

Finally we make sure that this macro can only be executed once for each supertabular by changing the definition of \ST@skippage.
This macro performs the actions necessary to start a new page.

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

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