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) doens'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 occurence 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 ocurrences 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.

`\topcaption`
`\bottomcaption`
`\tablecaption` 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.

supertabular
supertabular* The environments supertabular and supertabular* can be used much like the standard LATEX environments tabular and tabular*.

mpsupertabular
mpsupertabular* 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 maximimum 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`es will be produced at all.

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

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

  A quick but not very elegant solution: shrink the allowed height of the table with the command `\shrinkheight{...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	Number^2	Number^4	Number!
1	1	1	1
2	4	16	2
3	9	81	6
4	16	256	24
continued on next page			

continued from previous page			
Number	Number^2	Number^4	Number!
5	25	625	120
6	36	1296	720
7	49	2401	5040
8	64	4096	40320
9	81	6561	362880
10	100	10000	3628800
11	121	14641	39916800
12	144	20736	479001600
13	169	28561	6.22702080E+9
14	196	38416	8.71782912E+10
15	225	50625	1.30767437E+12
16	256	65536	2.09227899E+13
17	289	83521	3.55687428E+14
18	324	104976	6.40237370E+15
19	361	130321	1.21645100E+17
20	400	160000	2.43290200E+18

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$^2$} &
  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$^2$} &
  Number$^4$ &
  \multicolumn{1}{c|}{Number!} \\
  \hline}
\tabletail{%
  \hline
  \multicolumn{4}{|r|}{\small\sl continued on next page}\\
  \hline}
\tablelasttail{\hline}
\bottomcaption{This table is split across pages}
```

```
\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}
```

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.

Number	Number^2	Number^4	Number!
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

continued from previous page			
Number	Number^2	Number^4	Number!
10	1	1	and here is a long entry, where line breaks and line breaks and line breaks have to occur
11	1	1	and here is a long entry, where line breaks and line breaks and line breaks have to occur
12	1	1	here is a relative short entry
13	1	1	and here is a long entry, where line breaks and line breaks and line breaks have to occur
14	1	1	and here is a long entry, where line breaks and line breaks and line breaks have to occur
15	1	1	and here is a long entry, where line breaks and line breaks and line breaks have to occur
16	1	1	and here is a long entry, where line breaks and line breaks and line breaks have to occur
17	1	1	and here is a long entry, where line breaks and line breaks and line breaks have to occur
18	1	1	and here is a long entry, where line breaks and line breaks and line breaks have to occur

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 accross pages.

Number	Number^2	Number^4	Number!
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

Number	Number^2	Number^4	Number!
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
10	1		1 and here is a long entry, where line breaks and line breaks and line breaks have to occur
11	1		1 and here is a long entry, where line breaks and line breaks and line breaks have to occur
12	1		1 here is a relative short entry
13	1		1 and here is a long entry, where line breaks and line breaks and line breaks have to occur
14	1		1 and here is a long entry, where line breaks and line breaks and line breaks have to occur
15	1		1 and here is a long entry, where line breaks and line breaks and line breaks have to occur
16	1		1 and here is a long entry, where line breaks and line breaks and line breaks have to occur
17	1		1 and here is a long entry, where line breaks and line breaks and line breaks have to occur

continued from previous page			
Number	Number^2	Number^4	Number!
18	1	1 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 *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 an tail.

6 The Implementation

```
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

```
7 \newif\if@topcaption \@topcaptiontrue
8 \def\topcaption{\@topcaptiontrue\tablecaption}
9 \def\bottomcaption{\@topcaptionfalse\tablecaption}
```

\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.

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

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

```
15 \newif\ifST@star
```

\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{%
24                         \csname the#1\endcsname}{\ignorespaces #2}}
25   \begingroup
26     \@parboxrestore
27     \normalsize
28     \if@topcaption \vskip -10\p@ \fi
29     \@makecaption{\csname fnum@#1\endcsname}{\ignorespaces #3}\par
30     \if@topcaption \vskip 10\p@ \fi
31   \endgroup}
```

\tablehead
\tablefirsthead   `\tablehead` activates the new tabular `\cr` commands.

```
32 \newcommand\tablehead[1]{%
33   \gdef\@tablehead{%
34   \noalign{%
35       \global\let\@savcr=\\
36       \global\let\\=\org@tabularcr}%
37     #1%
38     \noalign{\global\let\\=\@savcr}}}
39 \tablehead{}
40 \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 `\\`

```
41 \newcommand\tabletail[1]{%
42   \gdef\@tabletail{%
43     \noalign{%
44       \global\let\@savcr=\\
45       \global\let\\=\org@tabularcr}%
46     #1%
```

```
47      \noalign{\global\let\\=\@savcr}}}
48 \tabletail{}
49 \newcommand\tablelasttail[1]{\gdef\@table@last@tail{#1}}
```

\sttraceon
\sttraceoff

There now is a possiblity to follow the decisions supertabular makes about breaking the tabular. This has to be enabled when converting this file with `docstrip` to a `.sty` file.

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

\ST@trace

A macro that gets the trace message as its argument

```
52 \newcommand\ST@trace[2]{%
53   \ifnum\c@tracingst>#1\relax
54     \GenericWarning
55       {(supertabular)\@spaces\@spaces}
56       {Package supertabular: #2}%
57   \fi
58   }
```

\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.

```
59 \newdimen\ST@pageleft
```

\shrinkheight

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

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

\setSTheight

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

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

\ST@headht
\ST@tailht

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)

```
64 \newdimen\ST@headht
65 \newdimen\ST@tailht
```

\ST@pagesofar

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

```
66 \newdimen\ST@pagesofar
```

\ST@pboxht

The measured (total) height of a parbox-argument

```
67 \newdimen\ST@pboxht
```

\ST@lineht
\ST@stretchht
\ST@prevht

The estimated height of a normal line is stored in \ST@lineht. The dimension register \ST@stretchht is used to store the difference between the 'normal' line height and the line height when \arraystretch has a non-standard value. This is used in the case where p-box entries are added to the tabular. The dimension register \ST@prevht is use to store the height of the previous line to use it as an estimate for the height of the next line. This is needed for a better estimate of when to break the tabular.

```
68 \newdimen\ST@lineht
69 \newdimen\ST@stretchht
70 \newdimen\ST@prevht
```

\ST@toadd When a tabular row is ended with \\[...] we need to temporarily store the optional argument in \ST@toadd.

```
71 \newdimen\ST@toadd
```

\ST@dimen A private scratch dimension register.

```
72 \newdimen\ST@dimen
```

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

```
73 \newbox\ST@pbox
```

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

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

```
74 \def\ST@tabularcr{%
75   {\ifnum0=`}\fi
76   \@ifstar{\ST@xtabularcr}{\ST@xtabularcr}}
77 \def\ST@xtabularcr{%
78   \@ifnextchar[%
79     {\ST@argtabularcr}%
80     {\ifnum0=`{\fi}\cr\ST@cr}}
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}
```

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

```
88 \def\ST@xargarraycr#1{%
89   \@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{\vskip #1\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.

```
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}
```

\ST@astartpbox Our version of \@astartpbox.

```
96 \def\ST@astartpbox#1{%
97   \bgroup\hsize#1%
98   \setbox\ST@pbox\vtop\bgroup\hsize#1\@arrayparboxrestore}
```

`\ST@endpbox`
`\ST@aendpbox`   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}
```

`\estimate@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\estimate@lineht{%
118   \ST@lineht=\arraystretch \baslineskp
119   \global\advance\ST@lineht by 1\p@
120   \ST@stretchht\ST@lineht\advance\ST@stretchht-\baslineskp
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 sofar, 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\@calnextpageht
```

```
139            \ST@trace\tw@{starting new page}%
140         \else
```

In this case we're in the second column, so we have to compensate for the material in the first column.

```
141            \ST@trace\tw@{Second column}%
142            \global\advance\ST@pageleft -\ST@pagesofar
143            \global\advance\ST@pageleft -\@colroom
144         \fi
```

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.

```
145       \else
146         \global\advance\ST@pageleft by -\ST@pagesofar
147         \global\ST@pagesofar\z@
148       \fi
149    \else
```

When we end up here, TEX has already decided it had enough material for the first column and is building the second column.

```
150       \ST@trace\tw@{Second column}
151       \ifnum\ST@pagesofar > \ST@pageleft
152         \ST@trace\tw@{starting new page}%
153         \newpage\@calnextpageht
154       \else
155         \global\advance\ST@pageleft by -\ST@pagesofar
156         \global\ST@pagesofar\z@
157       \fi
158    \fi
159  \else
```

In one column mode there is a simple decision.

```
160       \ST@trace\tw@{one column mode}%
161       \ifnum\ST@pagesofar > \ST@pageleft
162         \ST@trace\tw@{starting new page}%
163         \newpage\@calnextpageht
```

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

```
164       \else
165         \global\advance\ST@pageleft by -\ST@pagesofar
166         \global\ST@pagesofar\z@
167       \fi
168    \fi
169  \ST@trace\tw@{Available height: \the\ST@pageleft}%
```

Now we need to know the height of the head of the table. In order to measure this we typeset it in a normal tabular environment.

```
170  \ifx\@@tablehead\@empty
171    \ST@headht=\z@
172  \else
173    \setbox\@tempboxa=\vbox{\@arrayparboxrestore
174       \ST@restore
175       \expandafter\tabular\expandafter{\ST@tableformat}%
176       \@@tablehead\endtabular}%
```

```
177     \ST@headht=\ht\@tempboxa\advance\ST@headht\dp\@tempboxa
178   \fi
179   \ST@trace\tw@{Height of head: \the\ST@headht}%
```

To decide when to start a new page, we need to know the vertical size of the tail of the table.

```
180   \ifx\@tabletail\@empty
181     \ST@tailht=\z@
182   \else
183     \setbox\@tempboxa=\vbox{\@arrayparboxrestore
184       \ST@restore
185       \expandafter\tabular\expandafter{\ST@tableformat}
186         \@tabletail\endtabular}
187     \ST@tailht=\ht\@tempboxa\advance\ST@tailht\dp\@tempboxa
188   \fi
```

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.

```
189   \advance\ST@tailht by \ST@lineht
190   \ST@trace\tw@{Height of tail: \the\ST@tailht}%
191   \ST@trace\tw@{Maximum height of tabular: \the\ST@pageleft}%
192   \@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.

```
193   \advance\@tempdima\ST@lineht
194   \advance\@tempdima\ST@tailht
195   \ST@trace\tw@{Minimum height of tabular: \the\@tempdima}%
196   \ifnum\@tempdima>\ST@pageleft
197     \ST@trace\tw@{starting new page}%
198     \newpage\@calnextpageht
199   \fi
200 }
```

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

```
201 \def\@calnextpageht{%
202   \ST@trace\tw@{Calculating height of tabular on next page}%
203   \global\ST@pageleft\@colroom
204   \global\ST@pagesofar=\z@
205   \ST@trace\tw@{Maximum height of tabular: \the\ST@pageleft}%
206   }
```

\x@supertabular The body of the beginning of both environments is stored in a single macro as the code is shared.

```
207 \def\x@supertabular{%
```

First save the original definition 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 accross this inner tabular environment.

```
208   \let\org@tabular\tabular
209   \let\tabular\inner@tabular
```

The same needs to be done for the `tabular*` environment. The coding is slightly more verbose.

```
210    \expandafter\let
211      \csname org@tabular*\expandafter\endcsname
212      \csname tabular*\endcsname
213    \expandafter\let\csname tabular*\expandafter\endcsname
214      \csname inner@tabular*\endcsname
```

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

```
215    \if@topcaption \@process@tablecaption \fi
```

Save the original definition of `\\`.

```
216    \global\let\@oldcr=\\
```

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

```
217    \def\baslineskp{\baselineskip}%
```

We have to check whether `array.sty` was loaded, because some of the internal macros have different names.

```
218    \ifx\undefined\@classix
```

Save old `\@tabularcr` and insert the definition of `\@stabularcr`.

```
219      \let\org@tabularcr\@tabularcr
220      \let\@tabularcr\ST@tabularcr
```

Activate the new parbox algorithm.

```
221      \let\org@startpbox=\@startpbox
222      \let\org@endpbox=\@endpbox
223      \let\@@startpbox=\ST@startpbox
224      \let\@@endpbox=\ST@endpbox
225    \else
```

When `array.sty` was loaded things are a bit different.

```
226      \let\org@tabularcr\@arraycr
227      \let\@arraycr\ST@tabularcr
228      \let\org@startpbox=\@startpbox
229      \let\org@endpbox=\@endpbox
230      \let\@startpbox=\ST@astartpbox
231      \let\@endpbox=\ST@aendpbox
232    \fi
```

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

```
233    \ifx\@table@first@head\undefined
234      \let\@@tablehead=\@tablehead
235    \else
236      \let\@@tablehead=\@table@first@head
237    \fi
```

The first part of a supertabular may be moved on to the next page if it doesn't fit on the current page afterall. Subsequent parts can not be moved; therefore we will have to switch the definition of `\ST@skippart` around.

```
238    \let\ST@skippage\ST@skipfirstpart
```

Now we can estimate the average line height and the height of the first page of the supertabular.

```
239    \estimate@lineht
240    \@calfirstpageht
241    \noindent
242    }
```

\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...

```
243 \def\supertabular{%
244    \@ifnextchar[{\@supertabular}%]
245                {\@supertabular[]}}
```

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

```
246 \def\@supertabular[#1]#2{%
247    \def\ST@tableformat{#2}%
248    \ST@trace\tw@{Starting a new supertabular}%
```

Then remember that this is not a supertabular* environment.

```
249    \global\ST@starfalse
```

Don't use minipages.

```
250    \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.

```
251    \x@supertabular
```

Finally start a normal tabular environment.

```
252    \expandafter\org@tabular\expandafter{\ST@tableformat}%
253    \@@tablehead}
```

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

```
254 \@namedef{supertabular*}#1{%
255    \@ifnextchar[{\@nameuse{@supertabular*}{#1}}%
256                {\@nameuse{@supertabular*}{#1}[]}%]
257    }
```

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

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

Now we can call the common code for both environments.

```
264    \x@supertabular
```

And we can start a normal tabular* environment.

```
265    \expandafter\csname org@tabular*\expandafter\endcsname
266    \expandafter{\expandafter\ST@wd\expandafter}%
267    \expandafter{\ST@tableformat}%
268    \@@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...

```
269 \def\mpsupertabular{%
270   \@ifnextchar[{\@mpsupertabular}%]
271                 {\@mpsupertabular[]}}
```

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

```
272 \def\@mpsupertabular[#1]#2{%
273   \def\ST@tableformat{#2}%
274   \ST@trace\tw@{Starting a new mpsupertabular}%
```

Then remember that this is not a mpsupertabular* environment.

```
275   \global\ST@starfalse
```

And remember to close the minipage later.

```
276   \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.

```
277   \ST@rightskip \rightskip
278   \ST@leftskip \leftskip
279   \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.

```
280   \x@supertabular
```

Finally start a normal tabular environment.

```
281   \minipage{\columnwidth}%
282   \parfillskip\ST@parfillskip
283   \rightskip \ST@rightskip
284   \leftskip \ST@leftskip
285   \noindent\expandafter\org@tabular\expandafter{\ST@tableformat}%
286   \@@tablehead}
```

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

```
287 \@namedef{mpsupertabular*}#1{%
288   \@ifnextchar[{\@nameuse{@mpsupertabular*}{#1}}%
289                 {\@nameuse{@mpsupertabular*}{#1}[]}%]
290   }
```

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

```
291 \@namedef{@mpsupertabular*}#1[#2]#3{%
292   \ST@trace\tw@{Starting a new mpsupertabular*}%
293   \def\ST@tableformat{#3}%
294   \ST@wd=#1\relax
295   \global\ST@startrue
296   \global\ST@mptrue
297   \ST@rightskip \rightskip
298   \ST@leftskip \leftskip
299   \ST@parfillskip \parfillskip
```

Then we can call the common code for both environments.

```
300    \x@supertabular
301 %     And we can start a normal \textsf{tabular*} environment.
302 %     \begin{macrocode}
303    \minipage{\columnwidth}%
304    \parfillskip\ST@parfillskip
305    \rightskip \ST@rightskip
306    \leftskip \ST@leftskip
307    \noindent\expandafter\csname org@tabular*\expandafter\endcsname
308    \expandafter{\expandafter\ST@wd\expandafter}%
309    \expandafter{\ST@tableformat}%
310    \@@tablehead}%
```

\endsupertabular   This closes the environments supertabular and supertabular*.
\endsupertabular*
```
311 \def\endsupertabular{%
312    \ifx\@table@last@tail\undefined
313      \@tabletail
314    \else
315      \@table@last@tail
316    \fi
317    \csname endtabular\ifST@star*\fi\endcsname
```
Restore the original definition of \@tabularcr
```
318    \ST@restore
```
Check if we have to insert a caption and restore to default behaviour of putting
captions at the top.
```
319    \if@topcaption
320    \else
321      \@process@tablecaption
322      \@topcaptiontrue
323    \fi
```
Restore the meaning of \\ to the one it had before the start of this environment.
Also re-initialize some control-sequences
```
324    \global\let\\\@oldcr
325    \global\let\@process@tablecaption\relax
326    \ST@trace\tw@{Ended a supertabular\ifST@star*\fi}%
327    }
```
The definition of the ending of the supertabular* environment is simple:
```
328 \expandafter\let\csname endsupertabular*\endcsname\endsupertabular
```

\endmpsupertabular   This closes the environments mpsupertabular and mpsupertabular*.
\endmpsupertabular*
```
329 \def\endmpsupertabular{%
330    \ifx\@table@last@tail\undefined
331      \@tabletail
332    \else
333      \@table@last@tail
334    \fi
335    \csname endtabular\ifST@star*\fi\endcsname
336    \endminipage
```
Restore the original definition of \@tabularcr
```
337    \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
```

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

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

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

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

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

```
367 \def\ST@cr{%
368  \noalign{%
369    \ifnum\ST@pboxht<\ST@lineht
```

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.

```
370      \global\advance\ST@pageleft -\ST@lineht
```

And we store that fact in `\ST@prevht`.

```
371        \global\ST@prevht\ST@lineht
372     \else
```

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

```
373       \ST@trace\thr@@{Added par box with height \the\ST@pboxht}%
374        \global\advance\ST@pageleft -\ST@pboxht
375        \global\advance\ST@pageleft -0.1\ST@pboxht
376        \global\advance\ST@pageleft -\ST@stretchht
377        \global\ST@prevht\ST@pboxht
378        \global\ST@pboxht\z@
379     \fi
```

`\ST@toadd` is the value of the optional argument of `\\`.

```
380     \global\advance\ST@pageleft -\ST@toadd
381     \global\ST@toadd=\z@
382     \ST@trace\thr@@{Space left for tabular: \the\ST@pageleft}%
383   }
```

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 shoud be an *empty* macro (*not* `\relax`). (15.2.91)

```
384   \noalign{\global\let\ST@next\@empty}%
```

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

```
385   \ifnum\ST@pageleft<\z@
386     \ST@skippage
387   \else
```

When there is not enough space left on the current page, we start a new page. To compute the amount of space needed we use the height of the previous line (`\ST@prevht`) as an estimation of the height of the next line. If we are processing a mpsupertabular we need to take the height of the footnotes into account.

```
388     \noalign{\global\@tempdima\ST@tailht
389        \global\advance\@tempdima\ST@prevht
390     \ifST@mp
391       \ifvoid\@mpfootins\else
392         \global\advance\@tempdima\ht\@mpfootins
393         \global\advance\@tempdima 3pt
394       \fi
395     \fi}
396     \ifnum\ST@pageleft<\@tempdima
397       \ST@newpage
398     \fi
399   \fi
400   \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.

```
401 \def\ST@skipfirstpart{%
402   \noalign{%
403     \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:

```
404     \global\advance\ST@pageleft\pagetotal
```

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

```
405     \global\ST@pagesofar\z@
406     \newpage
```

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

```
407     \global\let\ST@skippage\ST@newpage
408   }}
```

`\ST@newpage`   This macro performs the actions necessary to start a new page.

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

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

```
411   \@tabletail
412   \ifST@star
413     \csname endtabular*\endcsname
414   \else
415     \endtabular
416   \fi
417   \ifST@mp
418     \endminipage
419   \fi
```

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

```
420   \global\let\ST@skippage\ST@newpage
421   \newpage\@calnextpageht
422   \let\ST@next\@tablehead
423   \ST@trace\tw@{writing head}%
424   \ifST@mp
425     \noindent\minipage{\columnwidth}%
426     \parfillskip\ST@parfillskip
427     \rightskip \ST@rightskip
428     \leftskip \ST@leftskip
429   \fi
```

```
430    \noindent
431    \ifST@star
432      \expandafter\csname org@tabular*\expandafter\endcsname
433      \expandafter{\expandafter\ST@wd\expandafter}%
434      \expandafter{\ST@tableformat}%
435    \else
436      \expandafter\org@tabular\expandafter{\ST@tableformat}%
437    \fi}

438 ⟨/package⟩
```