The subfigure package[*]

Steven Douglas Cochran

Digital Mapping Laboratory, School of Computer Science
Carnegie-Mellon University, 5000 Forbes Avenue
Pittsburgh, PA 15213–3891, USA

sdc+@cs.cmu.edu

1995/03/06

Abstract

This article documents the LaTeX package 'subfigure', which provides support for the inclusion of small, 'sub', figures and tables. It simplifies the positioning, captioning and labeling of such objects within a single figure or table environment. In addition, this package allows such sub-captions to be written to a List-of-Figures or List-of-Tables if desired. The 'subfigure' package also cooperates with the 'caption' package by H.A. Sommerfeldt [1] and should be compatible with all other packages that modify or extend the float environment.

1 Introduction

This package provides support for the manipulation and reference of small or 'sub' figures and tables within a single figure or table environment.[1] It is convenient to use this package when your subfigures are to be separately captioned, referenced, or whose captions are to be included in the List-of-Figures.

If you simply want to center your figure, then use `\centerline` or the `center` environment to do so. If you wrap your figure in a `\parbox` or a `minipage` of a short width, then you can place multiple figures or tables side-by-side. For example, the following will put two images side-by-side in a single figure as shown in Figure 1:

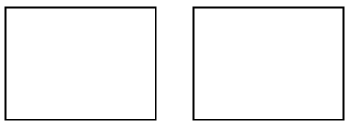


Figure 1: Here are two figures side-by-side.

```
\begin{figure}%
```

[*] This paper documents the subfigure package v2.0, last revised 1995/03/06.
[1] Additional float environments may be easily added as shown in section 4.4.

```
    \begin{center}%
      \parbox{2.5in}{\epsfbox{...}}%
      \hspace{.25in}%
      \parbox{2.5in}{\epsfbox{...}}%
    \end{center}%
    \caption{Here are two figures side-by-side.}%
  \end{figure}
```

2 The user interface

To use this package place

`\usepackage[`⟨*options*⟩`]{subfigure}`

in the preamble of your document. The following options are supported:

Option	Description
`normal`	Provides 'normal' captions, this is the default.
`hang`	Causes the label to be a hanging indentation to the caption paragraph.
`center`	Causes each line of the paragraph to be separately centered.
`centerlast`	Causes the last line only to be centered.
`nooneline`	If a caption fits on one line it will, by default, be centered. This option left-justifies the one line caption.
`scriptsize`, ..., `Large`	Sets the font size of the captions.
`up`, `it`, `sl`, `sc`, `md`, `bf`, `rm`, `sf` or `tt`	Sets the font attributes of the caption labels.

Within a `figure` or `table` environment, you can use the following macros to create a subfigure or subtable "box" with an optional `caption` under a `figure`. The `figure` is centered with `\subfigtopskip` of vertical space added above. If there is a `caption`, then `\subfigcapskip` vertical space is added below the `figure` followed by the `caption`. Finally, `\subfigbottomskip` of vertical space added at the bottom.

`\subfigure[`⟨*caption*⟩`]{`⟨*figure*⟩`}`
`\subtable[`⟨*caption*⟩`]{`⟨*figure*⟩`}`

The resulting "box" is made such that its baseline is at the bottom of the `figure` portion. Therefore, no matter how tall the figures and/or long the captions, adjacent subfigures are aligned at the bottom of their respective figures.

If a `caption` is given (including the null `caption` '`[ ]`') then the subfigure is labeled with a counter formatted by the macro '`\thesubfigure`' which returns, by default, '(a)', '(b)', etc. If desired, this macro may be redefined. The counter used for labeling the subfigures is `subfigure` and is incremented for each subfigure regardless of whether a `caption` was printed. The internals of the `\subtable` macro are symmetric to those of the `\subfigure` macro, described above.

If you wish to reference a specific subfigure or subtable, you can include a `\label` inside the body of either argument to the macro, with the `figure` argument being the preferred. If supplied, the CAPTION is a "moving argument" and, therefore, any "fragile" commands contained within it must be `\protect`'ed.

One final note, these macros are actually identical and it is the environment that defines whether a `subtable` or `subfigure` will be generated and not which macro is used. At the user level, the choice of names is purely cosmetic (and historical).

3 Examples

The easiest way to explain the use of this package is to give some examples. The first example, shown in Figure 2, specifies `\centering` and uses `\\` to control the placement of the subfigures. Note that the alignment of the top two subfigures is along the bottom of the figure portion of each.

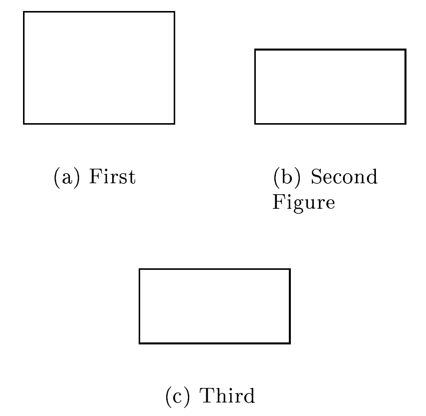


Figure 2: Three subfigures.

Figure 2 contains two top 'subfigures' and Figure 2(c).

```
\newcommand{\goodgap}{%
  \hspace{\subfigtopskip}%
  \hspace{\subfigbottomskip}}
...
\begin{figure}%
  \centering
  \subfigure[First]{...}\goodgap
  \subfigure[Second Figure]{...}\\
  \subfigure[Third]{\label{3figs-c}...}%
  \caption{Three subfigures.}
  \label{3figs}
\end{figure}
...
Figure~\ref{3figs} contains two top ‘subfigures’ and
Figure~\ref{3figs-c}.
```

A second example, shown in Figure 3, is to change the way that the subfigures are labeled and to have the subfigure captions printed in the list-of-figures. Here the alignment of the figures is accomplished by surrounding them in a `center` environment.

3.1: First   3.2: Second

Figure 3: Two subfigures.

See subfigures 3.1 and 3.2.

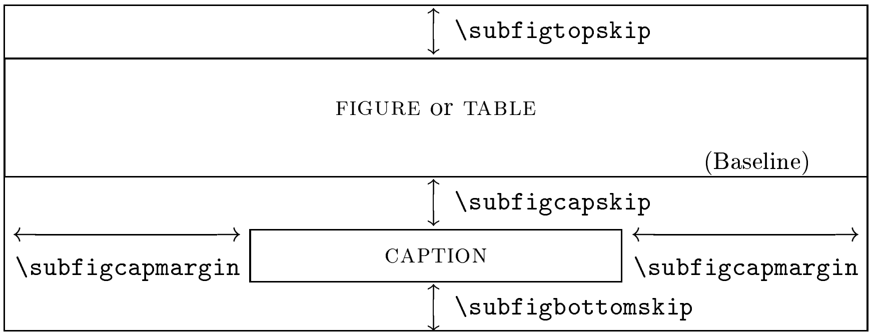
```
\renewcommand{\thesubfigure}{\thefigure.\arabic{subfigure}}
\makeatletter
  \renewcommand{\@thesubfigure}{\thesubfigure:\space}
  \renewcommand{\p@subfigure}{}
\makeatother
...
\setcounter{lofdepth}{2}
\listoffigures
...
\begin{figure}%
  \begin{center}%
    \subfigure[First]{%
      \label{fig:first}%
      ...}%
    \goodgap
    \subfigure[Second]{%
      \label{fig:second}%
      ...}%
  \end{center}
  \caption{Two subfigures.}
\end{figure}
...
See subfigures~\ref{fig:first} and \ref{fig:second}.
```

4 Customization

The following sections describe the internal parameters used by the subfigure package to define the layout of the sub-figures or tables, as well as the labels and captions the accompany them. In addition, adjustments to the entries on a "list-of" page and the addition of new float environments is described.

4.1 Changing the layout

The layout of the `subfigure` or `subtable` is defined by several internal values which may be changed to customize appearance of the object. The following illustration shows the relationship of these values.



4.2 Adjusting the label and caption

The label of the subfigure has two forms. The first is the one that appears in the text when you use the `\ref` macro and the second is the one that appears under the subfigure as part of the caption.

The `\ref` command yields a string composed by concatenating the value of `\p@subfigure` to `\thesubfigure`. By default these are defined by: "`\thefigure`" and "`(\alph{subfigure})`," respectively, which produces a reference of the figure number followed by the subfigure letter in parentheses. The label under the subfigure is generated by `\@thesubfigure`. By default this is "`{\subcaplabelfont\thesubfigure}\space}`." Note the `\subcaplabelfont` value. This starts out as a null value or if one of the font attribute options are given it is set to that value. If you update the `\@subfigure` macro, you should include the `\subcaplabelfont`.

Finally, the text of both the label and caption are prefixed by `\subcapsize`, which defaults to `\footnotesize` and may be changed using the six font size options.

One other way of changing the layout of the lapel and caption is by replacing the `\@makesubfigurecaption` macro which both applies the font size option and the five caption shape options. Each subfloat type may be defined individually.

The subtable label and caption are symmetric to that of the subfigure defined above.

4.3 Modifying the "List-Of" page information

To generate a "list-of" page for a float environment, you need to add a `\listoffigures` or `\listoftables` command where you want the list to appear. This command also causes the appropriate captions and subcaptions to be written to a file with the extensions `lof` or `lot` respectively. If you want the sub-caption text to appear in the "list-of" page, you need to change the value of the counters `lofdepth` or `lotdepth` counters from their default of '1'. For example, to have the `subfigure` sub-captions to appear on the "list-of-figures," add the following to the preamble of your paper:

`\setcounter{`⟨*subfigure*⟩`}{`⟨*2*⟩`}`

If you want to change how the sub-caption appears on the "list-of" pages you can change its format by redefining the `\l@subfigure` macro. Usually you will want to use the `\@dottedxxxline` macro (section 5.7, page 14) to help with the formatting. For instance the default value of `\l@subfigure` is:

```
\newcommand{\l@subfigure}{%
  \@dottedxxxline{\ext@subfigure}{2}{3.9em}{2.3em}}
```

To change the amount of space reserved for the label (if, for instance, you have a lot of figures) you could widen the 2.3em space for the label to 3.1em:

```
\newcommand{\l@subfigure}{%
  \@dottedxxxline{\ext@subfigure}{2}{3.9em}{3.1em}}
```

The arguments of the `\@dottedxxxline` macro are:

1. Type. Here these are, by default, `lof` or `lot`. The internal values `\ext@subfigure` and `\ext@subtable` stand for these extensions.
2. Level. By default this is '2' for the `subfigure` and `subtable`. If the level is greater than `\c@typedepth` (where type is the first argument value), then no line produced.
3. Indent. Total indentation from the left margin.
4. Numwidth. Width of box for the label number if the title has a `\numberline` command. This is also the amount of extra indentation added to second and later lines of a multiple line entry.
5. Title. Contents of entry.
6. Page. Page number.

and the final two arguments, title and page, are automatically appended to the value of `\l@subfigure` (and symmetricly for `\l@subtable`).

4.4 Adding additional float environments

It is easy to add a new float environment. For instance, let us assume we have a new float environment called "map" in which various maps are displayed and for which a list-of-maps is generated in the contents section. If we wanted to have sub-maps, then we could define the following (on the assumption that the definition of `\ext@map` is "`lom`"):

```
\makeatletter
  \newcounter{submap}[map]
  \newcommand{\thesubfigure}{(\alph{submap})}
  \newcommand{\@thesubmap}{{\subcaplabelfont\thesubmap}\space}
  \newcommand{\p@submap}{\themap}
  \newcommand{\ext@submap}{\ext@map}
  \newcommand{\l@submap}{\@dottedxxxline{\ext@submap}{2}{3.9em}{2.3em}}
  \newcounter{lomdepth}
  \setcounter{lomdepth}{1}
  \newcommand{\submap}{\subfigure}
  \newcommand{\@makesubmapcaption}{\@makesubfigurecaption}
\makeatother
```

4.5 Interaction with other parts of LaTeX

In the following sections, the interaction of the subfigure package with other parts of LaTeXis documented. These "other parts" may be either part of the the LaTeX base or contributed packages or classes.

4.5.1 TeX's "mouth"

The most important thing to remember when laying out your figures within a float environment is that spaces take room. If you have an extra space between two figures, then they will be separated by a little bit. If you begin a line with an extra space, then your figures may not be centered correctly.

TeX's state varies as it reads a line of text from a file. It ignores some spaces and carrage-returns and converts others to `\space`'s or `\par`'s. You can use a '`%`' to insure that you only have real spaces where you want them. To understand which spaces are significant, you should read chapter 8 of the TeXbook [2]. However, the main source of unexpected extra spacing is carrage-returns which are turned in to `\space`'s. Take a look at the examples in section 3 for an example. As a general rule: if in doubt, then add a '`%`' immediately after the last significant character of the line.

4.5.2 The float environments

Although the subfigure package was designed to work within a float environment (e.g. `figure`), it can be used outside with the following two caveats:

1. You need to define `\@captype`. This is usually either "`figure`" or "`table`." For example:

   ```
   \makeatletter
     \def\@captype{figure}
   \makeatother
   ```

2. If you want to define references using `\label`, then you need to redefine the LaTeX internal `\@currentlabel`. For example:

   ```
   \makeatletter
     \edef\@currentlabel{\p@subfigure\thesubfigure}
   \makeatother
   ```

4.5.3 The caption package

The only package that interacts with the subfigure package is the caption package by by H.A. Sommerfeldt [1]. If you load the subfigure package **before** the caption package, then it will detect that fact and will change the `\subcapsize` when the options `scriptsize`, ..., `large` are specified (overriding such options used when loading the subfigure package). In addition, it redefines `\@thesubfigure` and `\@thesubtable` to use `\captionlabelfont` rather than `\subcaplabelfont`.

If you **don't** want this behavior, then you must load the subfigure package **after** the caption package.

4.5.4 Creating a subfigure environment

Some people have wanted to use the `verbatim` environnment within the `\subfigure` macro and run into the restriction that the verbatim environment cannot be nested. To include verbatim text in a subfigure, you can define a new environment, in which verbatim text may be enclosed, and which calls the `\subfigure` macro.

```
\newbox\subfigbox
\makeatletter
  \newenvironment{subfloat}
    {\def\caption##1{\gdef\subcapsave{\relax##1}}%
     \let\subcapsave\@empty
     \setbox\subfigbox\hbox
       \bgroup}
      {\egroup
     \subfigure[\subcapsave]{\box\subfigbox}}
\makeatother
```

This environment may be used something like the following (but note that you will need to supply the width of the `verbatim` section using a `minipage`).

```
\begin{figure}
  \begin{center}%
    \begin{subfloat}%
      \begin{minipage}{2.1in}
        \begin{verbatim}
   This text  should be
verbatim.     And   not
 messed with in any way  !
        \end{verbatim}
      \end{minipage}%
      \caption{First subcaption.}%
    \end{subfloat}%
    \goodgap
    \begin{subfloat}%
      \begin{minipage}{2.1in}
        \begin{verbatim}
  This text (also)should be
verbatim.     And   not
 messed with in any way  !
        \end{verbatim}
      \end{minipage}%
      \caption{Second subcaption.}%
    \end{subfloat}
  \end{center}
  \caption{This is an example of verbatim text in a subfigure.}
\end{figure}
```

5 The code

5.1 Identification

```
\ifx\if@compatibility\undefined\else
  \NeedsTeXFormat{LaTeX2e}
  \ProvidesPackage{subfigure}[1995/03/06 v2.0 subfigure package]
  \typeout{Package: subfigure 1995/03/06 v2.0}
\fi
```

5.2 Initialization and Shared constants

`\if@subcaphang` `\if@subcapcenter` `\if@subcapcenterlast` `\if@subcapnooneline`

These four flags control how the style in which the subfloat label and caption are printed.

```
\newif\ifsubcaphang
\newif\ifsubcapcenter
\newif\ifsubcapcenterlast
\newif\ifsubcapnooneline
```

The following table gives the initial (default) values of the internals that are used to control the placement and printing of the subfloats.[2]

Macro	Default	Description
`\subfigtopskip`	10pt	Length from the top of the subfloat box to the begining of the `figure`.
`\subfigbottomskip`	10pt	Length from the bottom of the `caption` to the bottom of the subfloat.
`\subfigcapskip`	10pt	Length from the bottom of the `figure` to the begining of the `caption`.
`\subfigcapmargin`	10pt	Indentation of the caption from the sides of the subfloat box (This should always be positive).
`\subcapsize`	`\footnotesize`	Size for the text portion `caption` font.
`\subcaplabelfont`		Font for the label portion of the caption.

```
\newcommand{\subfigtopskip}{10pt}
\newcommand{\subfigbottomskip}{10pt}
\newcommand{\subfigcapskip}{10pt}
\newcommand{\subfigcapmargin}{10pt}
\newcommand{\subcapsize}{}
\newcommand{\subcaplabelfont}{}
```

5.3 Subfigure constants

`\c@subfigure`

Subfigure counter.

```
\newcounter{subfigure}[figure]
```

`\thesubfigure` `\@thesubfigure` `\p@subfigure`

These define the form that the sub-caption prefix is generated. The `\thesubfigure` macro defines the basic label for text references (prefixed by `\p@subfigure`), while the `\@thesubfigure` macro defines what appears under the subfigure. In the

[2]The `\subcapsize` value is initialized in section 5.5.1 along with the caption flags.

case of a conflict between this package and a prior one over the definition of `\thesubfigure`, this package will win.

```
\def\thesubfigure{(\alph{subfigure})}
\newcommand{\@thesubfigure}{{\subcaplabelfont\thesubfigure}\space}
\let\p@subfigure\thefigure
```

`\ext@subfigure` `\l@subfigure` `\c@lofdepth`

These define how and if the subfigure caption will appear in a list-of-figures file. `\ext@subfigure` defines the default subfigure file extension (which is the same as `\ext@figure` — the list-of-figures file). `\l@subfigure` shows how to print an `lof` subfigure line and defines that line at level two. `\c@lofdepth` is an extension of the table-of-contents depth value and controls the depth to which captions in the file are printed to the actual page. By default, the sub-captions are not printed.

```
\let\ext@subfigure\ext@figure
```

```
\newcommand{\l@subfigure}{%
  \@dottedxxxline{\ext@subfigure}{2}{3.9em}{2.3em}}
```

```
\newcounter{lofdepth}
\setcounter{lofdepth}{1}
```

5.4 Subtable constants

`\c@subtable`

Subtable counter.

```
\newcounter{subtable}[table]
```

`\thesubtable` `\@thesubtable` `\p@subtable`

These define the form that the sub-caption prefix is generated. The `\thesubtable` macro defines the basic label for text references (prefixed by `\p@subtable`), while the `\@thesubtable` macro defines what appears under the subtable. In the case of a conflict between this package and a prior one over the definition of `\thesubtable`, this package will win.

```
\def\thesubtable{(\alph{subtable})}
\newcommand{\@thesubtable}{{\subcaplabelfont\thesubtable}\space}
\let\p@subtable\thetable
```

`\ext@subtable` `\l@subtable` `\c@lotdepth`

These define how and if the subtable caption will appear in a list-of-tables file. `\ext@subtable` defines the default subtable file extension (which is the same as `\ext@table` — the list-of-tables file). `\l@subtable` shows how to print an `lot` subtable line and defines that line at level two. `\c@lotdepth` is an extension of the table-of-contents depth value and controls the depth to which captions in the file are printed to the actual page. By default, the sub-captions are not printed.

```
\let\ext@subtable\ext@table
```

```
\newcommand{\l@subtable}{%
  \@dottedxxxline{\ext@subtable}{2}{3.9em}{2.3em}}
```

```
\newcounter{lotdepth}
\setcounter{lotdepth}{1}
```

5.5 Declaration of options

For backward compatibility, if this file is running under LaTeX2.09 then we just set the defaults and leave it to the user to set any options in the preamble.

```
\ifx\if@compatibility\undefined
  \subcaphangfalse
  \subcapcenterfalse
  \subcapcenterlastfalse
  \def\subcapsize{\footnotesize}
\else
```

The following options allow compatibility with the caption package by H.A. Sommerfeldt [1]. There are five different sub-caption options supported: `normal`, `hang` (or `isu`), `center`, `centerlast` (or `anne`) and `nooneline`. The `hang` sub-caption may be combined with the `center` or `centerlast` options. The `nooneline` may be combined with any of the other options (but it's effect is negated by either of `center` or `centerlast` unless the `hang` option is also used).

```
  \DeclareOption{normal}{%
    \subcaphangfalse
    \subcapcenterfalse
    \subcapcenterlastfalse
    \subcapnoonelinefalse}
  \DeclareOption{hang}{\subcaphangtrue}
  \DeclareOption{center}{\subcapcentertrue}
  \DeclareOption{centerlast}{\subcapcenterlasttrue}
  \DeclareOption{nooneline}{\subcapnoonelinetrue}

  \DeclareOption{isu}{\ExecuteOption{hang}}
  \DeclareOption{anne}{\ExecuteOption{centerlast}}
```

There are options for six different font sizes available.

```
  \DeclareOption{scriptsize}{\renewcommand{\subcapsize}{\scriptsize}}
  \DeclareOption{footnotesize}{\renewcommand{\subcapsize}{\footnotesize}}
  \DeclareOption{small}{\renewcommand{\subcapsize}{\small}}
  \DeclareOption{normalsize}{\renewcommand{\subcapsize}{\normalsize}}
  \DeclareOption{large}{\renewcommand{\subcapsize}{\large}}
  \DeclareOption{Large}{\renewcommand{\subcapsize}{\Large}}
```

There are nine options available to set the font attributes of the sub-caption labels.

```
  \DeclareOption{up}{\renewcommand{\subcaplabelfont}{\upshape}}
  \DeclareOption{it}{\renewcommand{\subcaplabelfont}{\itshape}}
  \DeclareOption{sl}{\renewcommand{\subcaplabelfont}{\slshape}}
  \DeclareOption{sc}{\renewcommand{\subcaplabelfont}{\scshape}}
  \DeclareOption{md}{\renewcommand{\subcaplabelfont}{\mdseries}}
  \DeclareOption{bf}{\renewcommand{\subcaplabelfont}{\bfseries}}
  \DeclareOption{rm}{\renewcommand{\subcaplabelfont}{\rmfamily}}
  \DeclareOption{sf}{\renewcommand{\subcaplabelfont}{\sffamily}}
  \DeclareOption{tt}{\renewcommand{\subcaplabelfont}{\ttfamily}}
```

5.5.1 Execution of options

The `normal` type of sub-caption is preselected and the standard sub-caption size is set to `\footnotesize`.

```
  \ExecuteOptions{normal,footnotesize}
  \ProcessOptions
\fi
```

5.6 The subfigure/subtable body

\subfigure \subtable These macros act as cover functions for the `\@subfloat` macro. They insure that the proper counter is used and has the correct value. Since the caption is usually generated later, we must locally anticipate the future value of its counter by adding one to it within a local group. Upon leaving the `\subfigure` or `\subtable` macro, the old value is restored.

```
\newcommand{\subfigure}{%
  \bgroup
    \advance\csname c@\@captype\endcsname\@ne
    \refstepcounter{sub\@captype}%
    \leavevmode
    \@ifnextchar [%
      {\@subfloat{sub\@captype}}%
      {\@subfloat{sub\@captype}[\@empty]}}
\let\subtable\subfigure
```

\@subfloat This is the common code for setting up the subfloat box and drawing the sub-caption under it. It first determines width of the figure or table by placing it in a box. This box is combined with the top margin (`\subfigtopskip`) and placed as the height portion of the output "`vbox`." The depth of this box is determined by the existence of a sub-caption. If one exists, then the depth is made up of the sub-caption spacing (`\subfigcapskip`), the height+depth of the sub-caption and the bottom margin (`\subfigbottomskip`). If no sub-caption exists, then the depth of the output "`vbox`" is just the bottom margin (see the illustration in section 4.1).

The first argument is the type of object being generated: that is, a `subfigure` or a `subtable`. The second and third are the `caption` and `figure` arguments from the calling `\subfigure` or `\subtable` command.

```
\def\@subfloat#1[#2]#3{%
    \setbox\@tempboxa \hbox{#3}%
    \@tempdima=\wd\@tempboxa
    \vtop{%
      \vbox{
        \vskip\subfigtopskip
        \box\@tempboxa}%
      \ifx \@empty#2\relax \else
        \vskip\subfigcapskip
        \@subcaption{#1}{#2}%
      \fi
      \vskip\subfigbottomskip}%
  \egroup}
```

\@subfigcaptionlist \@subcaption The following series of macros control exactly how the sub-caption is typeset. The `\@subcaption` macro adds the sub-caption to the current list of sub-captions to be added to the "list-of" page as soon as the major caption is declared (see `\@caption` below). (NOTE: Only one list is kept because that seems right; if there is a mix of tables and figures, they will be grouped under the next `\caption`.

Next `\@subcaption` calls the appropriate float-type specific macro to decide how to size and shape the sub-caption text.

```
\newcommand{\@subfigcaptionlist}{}
```

```
\newcommand{\@subcaption}[2]{%
  \begingroup
    \let\label\@gobble
    \def\protect{\string\string\string}%
    \xdef\@subfigcaptionlist{%
      \@subfigcaptionlist,%
      {\protect\numberline {\@currentlabel}%
       \noexpand{\ignorespaces #2}}}%
  \endgroup
  \@nameuse{@make#1caption}{\@nameuse{@the#1}}{#2}}
```

`\@makesubfigurecaption` `\@makesubtablecaption`

By default, the `\@subfigurecaption` and `\@subtablecaption` macros are identical. Unlike the standard `\@makecaption` macro, we assume that the first argument (the label number produced by the `\@thesubfigure` or the `\@thesubtable`) contains any trailing separator characters or spacing (which makes it easier to customize).

The `\@makesubfigurecaption` macro first checks the size of the caption typeset as a single line. If it will fit under the subfigure with the given `\subfigcapmargin`, then we are done. Otherwise, the sub-caption is set with multiple lines either with or without the label hanging.

```
\newcommand{\@makesubfigurecaption}[2]{%
  \setbox\@tempboxa \hbox{%
    \subcapsize
    \ignorespaces #1%
    \ignorespaces #2}%
  \@tempdimb=-\subfigcapmargin
  \multiply\@tempdimb\tw@
  \advance\@tempdimb\@tempdima
    \hbox to\@tempdima{%
      \hfil
      \ifdim \wd\@tempboxa >\@tempdimb
         \subfig@caption{#1}{#2}%
      \else\ifsubcapnooneline
         \subfig@caption{#1}{#2}%
      \else
        \box\@tempboxa
      \fi\fi
      \hfil}}
```

```
\let\@makesubtablecaption\@makesubfigurecaption
```

`\subfig@caption` `\subfig@captionpar`

These macros are called to typeset a multiple-line sub-caption (or a single line when `subcapnooneline` is true). Depending on the `subcapcenter` and `subcapcenterlast` flags, the text will be justified (both false), centered (`subcapcenter` true), or justified with the last line centered (only the flag `subcapcenterlast` set true).

```
\newcommand{\subfig@caption}[2]{%
  \ifsubcaphang
    \sbox{\@tempboxa}{%
      \subcapsize
      \ignorespaces #1}%
    \addtolength{\@tempdimb}{-\wd\@tempboxa}%
    \usebox{\@tempboxa}%
    \subfig@captionpar{\@tempdimb}{#2}%
```

```
  \else
    \subfig@captionpar{\@tempdimb}{#1#2}%
  \fi}
\newcommand{\subfig@captionpar}[2]{%
  \parbox[t]{#1}{%
    \strut
    \ifsubcapcenter
      \setlength{\leftskip}{\@flushglue}%
      \setlength{\rightskip}{\@flushglue}%
      \setlength{\parfillskip}{\z@skip}%
    \else\ifsubcapcenterlast
      \addtolength{\leftskip}{0pt plus 1fil}%
      \addtolength{\rightskip}{0pt plus -1fil}%
      \setlength{\parfillskip}{0pt plus 2fil}%
    \fi\fi
    \subcapsize
    \ignorespaces #2%
    \par}}
```

5.7 Patches to the standard environment

The following adjust the standard environment for the `\subfigure` and `\subtable` macros. They are designed as wrappers to the current definition of the standard commands to minimize any chance of conflict with other packages.

`\@dottedxxxline` This is a generalized wrapper for the `\@dottedtocline` macro. It checks for the level based on the output file (first argument) and not using only `\@tocdepth`. (See section 4.3 for a description of the arguments.)

```
\newcommand{\@dottedxxxline}[6]{%
  \ifnum #2>\@nameuse{c@#1depth}\else
    \@dottedtocline{0}{#3}{#4}{#5}{#6}
  \fi}
```

`\subfig@oldcaption` `\@caption` Next, we redefine the current `\@caption` macro to dump any sub-captions saved. First the 'old' caption macro is called to add the line to the list-of file and then the list of sub-captions, `\@subfigcaptionlist` is written to the same file. Lastly, the `\@subfigcaptionlist` is reinitialized.

```
\let\subfig@oldcaption\@caption
\long\def\@caption#1[#2]#3{%
  \subfig@oldcaption{#1}[{#2}]{#3}%
  \@for \@tempa:=\@subfigcaptionlist \do {%
    \ifx\@empty\@tempa\relax \else
      \addcontentsline
        {\@nameuse{ext@sub#1}}%
        {sub#1}%
        {\@tempa}%
    \fi}%
  \gdef\@subfigcaptionlist{}}
```

References

[1] Harald Axel Sommerfeldt, *The* caption *package*, 1995/01/30.

[2] Donald Ervin Knuth, *The TEXbook*, Addison-Wesley, Reading, Massachusetts, 1986.