The sidecap package

Rolf Niepraschk (niepraschk@ptb.de) Hubert Gäßlein

v1.7a – 2023/01/24

Abstract

This package defines the new environments `SCfigure` and `SCtable`, analogous to `figure` and `table`, which make it easy to typeset captions sideways.

Additionally, a `wide` environment is defined; it allows to use the margin area, e. g., for figures wider than `\textwidth`.

1 Introduction

In some cases it may be useful to typeset the caption aside the figure or the table. For this purpose the package sidecap defines the new environments `SCfigure` and `SCtable`. The figure or the table and the caption are put into two minipages that are positioned side by side and centered as a whole. The space between the minipages is `\sidecaptionsep`. The correct positioning with respect to left and right pages requires at least two compilation runs.

2 Usage

`\usepackage[`⟨*option*⟩`]{sidecap}`

`\begin{SCtable}` `[`⟨*relwidth*⟩`][`⟨*float*⟩`]` ... `\end{SCtable}`
`\begin{SCfigure}` `[`⟨*relwidth*⟩`][`⟨*float*⟩`]` ... `\end{SCfigure}`
`\begin{SCtable*}` `[`⟨*relwidth*⟩`][`⟨*float*⟩`]` ... `\end{SCtable*}`
`\begin{SCfigure*}`[⟨*relwidth*⟩][⟨*float*⟩] ... `\end{SCfigure*}`

⟨*option*⟩ – `outercaption` (default): Caption appears left on left pages and right on right pages.

`innercaption`: Caption appears right on left pages and left on right pages.

`leftcaption`, `rightcaption`: Caption is always on the left or right, respectively.

`wide`: The floating objects may extend into the margin area.

`raggedright`, `raggedleft`, `ragged`: Better justification for small captions. The ragged2e package is used if available.

⟨*relwidth*⟩ – optional; caption width relative to the width of the figure or table. A large value (e.g., 50) reserves the maximum width that is possible. Default is `\sidecaptionrelwidth` (which is initialized to 1.0).

⟨*float*⟩ – optional; like the floating position parameter of the original table/figure environments. Default is `tbp`.

`\begin{wide} ... \end{wide}`
The `wide` environment may be used inside `figure` and `table` environments as well as in the normal text.

3 Required packages

This package requires the standard LaTeX package ifthen.

4 Supported packages

This package is compatible with LaTeX package hyperref (tested with version 6.71v as of 2001/11/12). If the `raggedright` or `raggedleft` or `ragged` package option has been given, then captions will be set with ragged margin. The ragged2e package will be used if it can be found.

5 The implementation

5.1 Register allocation and auxiliary macros

```
1 ⟨*package⟩
2
3 \RequirePackage{ifthen}
4
5 \@ifdefinable{\SC@BOXWD}{\newdimen\SC@BOXWD}
6 \@ifdefinable{\SC@CAPWD}{\newdimen\SC@CAPWD}
7 % 2 noetig ???
8 \@ifdefinable{\SC@tempdima}{\newdimen\SC@tempdima}
9 \@ifdefinable{\SC@tempdimb}{\newdimen\SC@tempdimb}
10 \newcounter{SC@C} \newsavebox{\SC@BOX} \newcommand*{\SC@FLOAT}{}
11 \newcommand*{\SC@IDENT}{SC@\number\value{SC@C}}
12 \newcommand*\SC@label[1]{\gdef\SC@LABtext{#1}\ignorespaces}
13 \newcommand*\SC@orig@caption{} \newcommand*\SC@orig@label{}
14 \newcommand*\sidecaptionsep{}
15 \edef\sidecaptionsep{\the\marginparsep}
16 \newcommand*\sidecaptionrelwidth{1.0}
17 \newcommand*\sidecaptionvpos[2]{% 2002/08/31
18   \@namedef{SC@#1@vpos}{#2}%
19 }
20
21 \newboolean{SC@wide}
22 \def\SC@widefalse{\global\let\ifSC@wide\iffalse}
23 \def\SC@widetrue {\global\let\ifSC@wide\iftrue}
24
25   % Kann man auf dieses Laengenregister verzichten? Nein! (RN) <****>
26
27 \newcommand*\SC@calcwidths[1]{%
28   \SC@tempdima#1\relax
29   \SC@BOXWD\wd\SC@BOX
30   \SC@CAPWD\SC@fraction\SC@BOXWD
31   \@tempdima\SC@BOXWD
32   \advance\@tempdima\SC@CAPWD
```

```
  \advance\@tempdima\sidecaptionsep
  \ifdim\@tempdima>\SC@tempdima
    \advance\SC@CAPWD-\@tempdima
    \advance\SC@CAPWD\SC@tempdima
  \fi
  \SC@tempdimb\@tempdima
}

%\newcommand*{\isSC@WIDEi}[2]{#2}  %default: false
%\newcommand*{\isSC@WIDEii}[2]{#2} %default: false
\newcommand*{\isSC@WIDEi}[2]{\ifSC@wide#1\else#2\fi}  %default: false
\newcommand*{\isSC@WIDEii}[2]{\ifSC@wide#1\else#2\fi} %default: false

%\newcommand*{\isSC@First}[2]{#1} %default: true

\newcommand*\SC@justify{}
\newcommand*\SC@justify@caption{}
\newcommand*\SC@justify@body{}

\newboolean{SC@origRagged} \setboolean{SC@origRagged}{false} %% ???
```

\SC@SaveCommands

```
\newcommand*{\SC@SaveCommands}{%
    \let\SC@orig@caption=\caption \let\SC@orig@label=\label%
}

\SC@SaveCommands % necessary? or even wrong?
```

5.2 Package **hyperref** compatibility

Note: The \pageref doesn't work with hyperref . . . :-(

```
\if@twoside
  \newcommand*{\isSC@ODD}[2]{%
    \ifthenelse{\isodd{\pageref{\SC@IDENT}}}{#1}{#2}}
\else
  \newcommand*{\isSC@ODD}[2]{#1}%
\fi
```

5.3 Option processing

```
\DeclareOption{innercaption}{%
  \renewcommand*{\SC@FLOAT}[2]{%
    \isSC@ODD{{#1}\hspace{\sidecaptionsep}{#2}} {{#2}\hspace{\sidecaptionsep}{#1}}}
  \renewcommand*\SC@justify@caption{%
    \isSC@ODD{\RaggedLeft}{\RaggedRight}}
}

\DeclareOption{outercaption}{%
  \renewcommand*{\SC@FLOAT}[2]{%
    \isSC@ODD{{#2}\hspace{\sidecaptionsep}{#1}} {{#1}\hspace{\sidecaptionsep}{#2}}}
  \renewcommand*\SC@justify@caption{%
    \isSC@ODD{\RaggedRight}{\RaggedLeft}}
}

```

```
\DeclareOption{rightcaption}{%
  \renewcommand*{\SC@FLOAT}[2]{{#2}\hspace{\sidecaptionsep}{#1}}
  \def\SC@justify@caption{\RaggedRight}%
}

\DeclareOption{leftcaption}{%
  \renewcommand*{\SC@FLOAT}[2]{{#1}\hspace{\sidecaptionsep}{#2}}
  \def\SC@justify@caption{\RaggedLeft}%
}

\DeclareOption{wide}{%
   \renewcommand*{\isSC@WIDEi}[2]{\if@twocolumn #2\else #1\fi}% ???
   \renewcommand*{\isSC@WIDEii}[2]{#1}}
```

These 'ragged...' options should stay here, behind the '...caption' options!

```
\DeclareOption{raggedright}{%
   \def\SC@justify{\RaggedRight}}

\DeclareOption{raggedleft}{%
   \def\SC@justify{\RaggedLeft}}

\DeclareOption{ragged}{%
   \def\SC@justify{\SC@justify@caption}%   \let ?
}

\DeclareOption{origragged}{\setboolean{SC@origRagged}{true}}

\DeclareOption{margincaption}{%
  \ds@outercaption\ds@outerbody\ds@wide
  \let\ds@innercaption\@empty
  \let\ds@leftcaption\@empty
  \let\ds@rightcaption\@empty
  \def\SC@calcwidths#1{%
    \SC@tempdimb#1\relax
    \SC@BOXWD\SC@tempdimb  % *-form !!! ???
    \advance\SC@BOXWD-\marginparwidth
    \advance\SC@BOXWD-\marginparsep
    \SC@CAPWD\marginparwidth
  }%
}

\DeclareOption{innerbody}{%
  \def\SC@justify@body{%
    \isSC@ODD{\RaggedRight}{\RaggedLeft}}
}

\DeclareOption{outerbody}{%
  \def\SC@justify@body{%
    \isSC@ODD{\RaggedLeft}{\RaggedRight}}
}

\DeclareOption{centerbody}{%
  \def\SC@justify@body{\Centering}
}

\DeclareOption{rightbody}{%
  \def\SC@justify@body{\RaggedLeft}%
}
```

```
127
128 \DeclareOption{leftbody}{%
129   \def\SC@justify@body{\RaggedRight}%
130 }
131 \ExecuteOptions{outercaption}
132 %%\InputIfFileExists{sidecap.cfg}{}{} % hier ? % 2002/06/14
133 \ProcessOptions\relax
134 \AtBeginDocument{%
135   \ifSC@origRagged\else
136     \@ifpackageloaded{ragged2e}{}%
137     {%
138       \IfFileExists{ragged2e.sty}{%
139    \RequirePackage[NewParameters]{ragged2e}}%
140       {%
141         \PackageWarningNoLine{sidecap}{%
142       Package `ragged2e' not found -- expect bad formatting}%
143       }%
144     }
145   \fi
146   \@ifundefined{RaggedRight}%
147   {%
148     \let\RaggedLeft\raggedleft
149     \let\RaggedRight\raggedright
150     \let\Centering\centering
151   }{}
152 }
```

5.4 User-level macros (environments)

`SCfigure` (*env.*) Simply passes the first (optional) parameter and the required one, in this case 'figure', to `SC@float`. The figure caption should be bottom aligned.

```
153
154 %\newcommand*\SCfigureVpos{b}% 2002/06/14
155 \@ifdefinable\SC@figure@vpos{\def\SC@figure@vpos{b}}
156 \newenvironment{SCfigure}{\SC@float[\SC@figure@vpos]{figure}}{\endSC@float}
157 \newenvironment{SCfigure*}{\SC@dblfloat[\SC@figure@vpos]{figure}}{\endSC@dblfloat}
158
```

`SCtable` (*env.*) Simply passes the first (optional) parameter and the required one, in this case 'table', to `SC@float`. The table caption should be top aligned.

```
159
160 %\newcommand*\SCtableVpos{t}% 2002/06/14
161 \@ifdefinable\SC@table@vpos{\def\SC@table@vpos{t}}
162 \newenvironment{SCtable}{\SC@float[\SC@table@vpos]{table}}{\endSC@float}
163 \newenvironment{SCtable*}{\SC@dblfloat[\SC@table@vpos]{table}}{\endSC@dblfloat}
164
```

`wide` (*env.*) This is an environment that allows to extend the width of the text body (or of a floating environment) by using the margin space.
It shouldn't be used in twocolumn text.

```
165 %
166 \newenvironment{wide}%
167 {%
```

```
  \@parboxrestore% 2002/06/14 -- SW-Arch.: wozu? 2002/08/24
  \advance\linewidth\marginparwidth
  \advance\linewidth\marginparsep
  \ifx\label\SC@label
⟨+debug⟩    \typeout{WIDE: inside SCfloat}%
    \SC@widetrue
  \else
⟨+debug⟩    \typeout{WIDE: outside SCfloat}%
    \begin{lrbox}\SC@BOX  % \begin necessary: --> \@noitemerr
       \minipage\linewidth
  \fi
}%
{%
  \ifx\label\SC@label
⟨+debug⟩
\typeout{WIDE: inside SCfloat}%
    %\renewcommand*{\isSC@WIDEi}[2]{\if@twocolumn ##2\else ##1\fi}%
    %\renewcommand*{\isSC@WIDEii}[2]{##1}%
    %\let\isSC@WIDE\isSC@WIDEi
  \else
⟨+debug⟩
\typeout{WIDE: outside SCfloat}%
      \endminipage
    \end{lrbox}%  \end necessary: --> \@noitemerr
    \global\advance\c@SC@C\@ne
    \SC@orig@label{\SC@IDENT}% nicht \ref... !!!
    \trivlist \item\relax
    \isSC@ODD{\def\SC@hpos{l}}{\def\SC@hpos{r}}%
⟨+debug⟩
\typeout{WIDE: \string\linewidth=\the\linewidth}%
    \noindent\makebox[\textwidth][\SC@hpos]{\copy\SC@BOX}% \usebox{\SC@BOX}
    \endtrivlist
  \fi
}
```

5.5 Internal macros

5.5.1 Collecting arguments

The new internal float environment, similar/analogous to LaTeX's `@float` environment.

Syntax: `\SC@float[`⟨*vpos*⟩`]{`⟨*name*⟩`}[`⟨*relwidth*⟩`][`⟨*fps*⟩`]`

- Parameter ⟨*vpos*⟩ (optional) is the vertical positioning of the caption.
- Parameter ⟨*name*⟩ (required) is the name of the 'original' LaTeX floating environment (e.g., 'figure' or 'table').
- Parameter ⟨*relwidth*⟩ (optional) is the desired relative width of the caption.
- Parameter ⟨*fps*⟩ (optional) is the usual LaTeX float positioning specifier.

The usual 'cascading' programming style is applied (cf. LaTeX's `\@float`).

\SC@float Initially, the first optional parameter is checked for.

```
\def\SC@float{\SC@SaveCommands\@ifnextchar[\SC@xfloat{\SC@xfloat[c]}}
```

\SC@xfloat Then the first and second parameters are consumed and the third one is checked for.

```
\def\SC@xfloat[#1]#2{\@ifnextchar[%
  {\SC@yfloat{#1}{#2}}%
  {\SC@zfloat{#1}{#2}{\sidecaptionrelwidth}[\@nameuse{fps@#2}]}}
```

\SC@yfloat Again, the fourth (and last) parameter is checked for.

```
\def\SC@yfloat#1#2[#3]{\@ifnextchar[%
  {\SC@zfloat{#1}{#2}{#3}}%
  {\SC@zfloat{#1}{#2}{#3}[\@nameuse{fps@#2}]}}
```

5.5.2 Capturing the float's contents

\SC@zfloat Finally, here is the macro that does all the work.

```
\def\SC@zfloat#1#2#3[#4]{%
  \def\SC@vpos{#1}%
  \expandafter\edef\csname fps@#2\endcsname{#4}%
  \def\SC@captype{#2}%
  \ifx#3\@empty\def\SC@fraction{1}\else\def\SC@fraction{#3}\fi%
```

The \caption and \label commands must be redefined.

```
  \let\SC@CAPtext\@empty \let\SC@OPTCAPtext\@empty \let\SC@LABtext\@empty%
  \renewcommand\caption[2][]{\gdef\SC@OPTCAPtext{##1}%
  \gdef\SC@CAPtext{##2}\ignorespaces}% \SC@justify
  \let\label=\SC@label
  \@namedef{fnum@#2}{\mbox{\@nameuse{#2name}~\@nameuse{the#2}}}% caption2 ?
```

Save the figure or table (or whatever) in a box.

```
  \SC@widefalse
  \begin{lrbox}\SC@BOX  % \begin necessary: --> \@noitemerr
  %%% \strut
}%

```

\SC@dblfloat Analogous to LaTeX's \@dblfloat.

```
\def\SC@dblfloat{%
  \SC@SaveCommands%
  \if@twocolumn\let\reserved@a\SC@dbflt\else\let\reserved@a\SC@float\fi
  \reserved@a}
\def\SC@dbflt{\SC@float}
```

5.5.3 Output the float's contents

\endSC@float Outputs the figure or table (or whatever) and the caption.

```
\def\endSC@float{%
  %%% \strut
  \end{lrbox}%  \end necessary: --> \@noitemerr
⟨+debug⟩  \typeout{onecolumn}%
  \let\isSC@WIDE\isSC@WIDEi%
  \def\@FLOAT{\@float}\def\end@FLOAT{\end@float}%
```

```
  \isSC@WIDE%
    {\@tempdima\textwidth % \columnwidth ???
     \advance\@tempdima\marginparwidth
     \advance\@tempdima\marginparsep}%
    {\@tempdima\columnwidth}%
⟨+debug⟩ \typeout{endSC@float: \string\@tempdima=\the\@tempdima}%
  \endSC@FLOAT\@tempdima}%
```

`\endSC@dblfloat` Ditto for `*`-forms of floats.

```
\def\endSC@dblfloat{%
  \end{lrbox}%  \end necessary: --> \@noitemerr
⟨+debug⟩  \typeout{twocolumn}%
  \let\isSC@WIDE\isSC@WIDEii%
  \def\@FLOAT{\@dblfloat}\def\end@FLOAT{\end@dblfloat}%
  \isSC@WIDE%
    {\@tempdima\textwidth%
     \advance\@tempdima\marginparwidth
     \advance\@tempdima\marginparsep}%
    {\@tempdima\textwidth}%
⟨+debug⟩
\typeout{endSC@dblfloat: \string\@tempdima=\the\@tempdima}%
  \endSC@FLOAT\@tempdima}%
```

`\endSC@FLOAT` Sets the caption width. If caption width plus figure/table width (plus separation space) is too large then the caption width is set equal to the remaining width.

```
\def\endSC@FLOAT#1{%
  \SC@calcwidths{#1}%
```

`\@FLOAT` Calls the LaTeX float command with the two minipages inside a main minipage.

```
  \@FLOAT{\SC@captype}%
    \abovecaptionskip\z@skip
    \belowcaptionskip\z@skip
```

Creates a label for each figure or table (etc.) for later determination if the page is odd or even. The counter `SC@C` must be incremented before.

```
    \global\advance\c@SC@C\@ne
    \SC@orig@label{\SC@IDENT}% nicht \ref... !!!
```

`\isSC@WIDE`

```
    \isSC@WIDE%
      {\ifdim\SC@tempdimb>\textwidth % ???
         \isSC@ODD{\def\SC@hpos{l}}{\def\SC@hpos{r}}%
       \else%
         \def\SC@hpos{c}%
       \fi
       \@tempdimc\textwidth}% ???
      {\@tempdimc\SC@tempdima\def\SC@hpos{c}}%
    \makebox[\@tempdimc][\SC@hpos]{%
```

`\SC@FLOAT` Has two parameters. The first parameter is the minipage with the caption text inside and the last parameter is the minipage with the body of the figure or table inside.

```
      \SC@FLOAT%
        {\minipage[\SC@vpos]{\SC@CAPWD}%
           %% \nointerlineskip %\offinterlineskip%
           %% \kern0pt\relax
           \hrule \@height\z@\@depth\z@  %%% Testen, Doku. (Implementierung) !!!
           %\show\SC@justify@caption
%%%           \tracingall\tracingonline=0
     \SC@justify
     \let\@arrayparboxrestore\relax % sonst Blocksatz! (\@parboxrestore)
     \let\usecaptionmargin\relax % sonst Pseudo-Blocksatz, nicht-alternierend!
%     \isSC@ODD {\RaggedRight }{\RaggedLeft }
%%%     A B C
%   \isSC@ODD{\typeout{xxxxx ungerade}}{\typeout{xxxxx gerade}}%
%   \def\@makecaption##1##2{##1: ##2\endgraf}%
% \@parboxrestore
% \@setminipage
\iffalse
% DEBUG:
          \endgraf
          \@makecaption{\csname fnum@\SC@captype\endcsname}%
              {\ignorespaces \SC@CAPtext}%
          \endgraf
\else
          % \ifthenelse{\equal{\SC@OPTCAPtext}{\@empty}}%
           \ifx\SC@OPTCAPtext\@empty% Noetig?
% FM: strut needed twice for multiline captions
             \SC@orig@caption[\SC@CAPtext]%
                  {\strut\ignorespaces\SC@CAPtext\unskip\strut}% \expandafter\protect
           \else
% FM: strut needed twice for multiline captions
% FM: if \SC@CAPtext can contain trailing banks then we also need \unskip
% FM: (not checked)
             \SC@orig@caption[\SC@OPTCAPtext]% \expandafter\protect
                  {\strut\ignorespaces\SC@CAPtext\unskip\strut}% \expandafter\protect
           \fi
           \unskip% ??? Dok. !!!
           \ifx\SC@LABtext\@empty\else
             \SC@orig@label{\SC@LABtext}\fi%  \expandafter\protect
\fi
%%%     X Y Z
           \hrule \@height\z@\@depth\z@
         \endminipage}%
        {\minipage[\SC@vpos]{\SC@BOXWD}%
           %% \nointerlineskip %\offinterlineskip%
           %% \kern0pt\relax
           \hrule \@height\z@\@depth\z@
           \SC@justify@body
           \unhbox\SC@BOX % \strut \unhbox/\usebox{\SC@BOX}%  TESTEN!
           % \kern0pt\relax
           \hrule \@height\z@\@depth\z@
         \endminipage}%
    }%
  \end@FLOAT%
}
```

```
\InputIfFileExists{sidecap.cfg}{}{} % hier ? % 2002/06/14
⟨/package⟩
```

Change History

Index

Numbers written in *italic* refer to the page where the corresponding entry is described, the ones underlined to the definition, the rest to the places where the entry is used.