

The `swfigure` package

Managing large and spread wide figures

Claudio Beccari*

Version v.0.9.08 — Last revised 2020-10-13

Contents

| | | | |
|-------------------------------------|----------|---|----|
| 1 Introduction | 1 | 6.1 The <code>\SWfigure</code> macro for Spread Wide images . | 8 |
| 2 Installation | 2 | 6.2 The <code>\NFfigure</code> macro for normal full page figures | 10 |
| 3 Usage | 3 | 6.3 The <code>\RFfigure</code> macro for rotated figures | 10 |
| 4 Display mode peculiarities | 4 | 6.4 The <code>\VSfigure</code> macro for tall and slim figures . | 11 |
| 5 Acknowledgements | 5 | 6.5 The <code>\HSfigure</code> macro for spread wide slim fig- ures | 12 |
| 6 The code | 6 | | |

Abstract

This package defines a single command that with different options can insert large images into the document; those that occupy an entire spread s get split into two halves that are inserted on two facing pages in such a way that they merge when the document is read on the screen (set to double page view) or is printed in a well sewn book so that facing pages join correctly at the spine. This documentation is accompanied by another file `swfigure-examples.pdf` containing examples of this package several uses.

1 Introduction

Sometimes it is necessary to insert in a given document very large images; either they are larger than a spread, or they do not exceed the width of a spread. In the first case it is necessary to use to large sheets of paper folded in the proper way and inserted in the printed document as special inserts; or for documents to be read on the screen such large sheets have to be attached to the document as separate files to be viewed in windows different from that where the document is being read.

In the other case it is possible to manage such large figures in different ways; we define a single command that, according to different settings, can insert such figures in five different display modes.

*E-mail: `claudio dot beccari at gmail dot com`

1. The normal L^AT_EX kernel full page display mode, but using the `figure` environment with the `[p]` placing option. Apparently there is no need to define a new command for this display mode, except the advantage of using the same command as the other modes and a few small further functionalities that the new command has available.
2. The landscape display mode available with the `landscape` package and other similar ones. Again this display mode appears superfluous, but the advantage is to use the same command as with the other display modes
3. The screen-wide display mode, where two facing pages display a large figure divided in two halves, each one set in its page shifted to the spine so as to occupy on each page the text width plus the internal margin width; with suitable aspect ratio, both facing pages are practically full. No other text appears in this display mode, except the caption in the right page external margin.
4. The slim screen-wide image is similar to the previous mode, but since the figure is slim, it is set at the top of two facing pages, with text underneath; the caption is under the right figure half in the right page, and the left text block height is set so that the first lines of both text blocks are aligned.
5. The slim tall image is set to the side of the text block that wraps it. Recurse is made to the `wrapfig` package, therefore the software might sometimes hick-up a little bit, because of the idiosyncrasies of `wrapfig` that performs very well in most but not all circumstances; see that package documentation for such `wrapfig` limitations. With this `wsfigure` package we limit the use of this display mode to images that have a “height over width” ratio not lower than 2; we provide also some option arguments so as to correct small imperfections in the sizing of the text indentation

In any case, depending on the page geometry and the image aspect ratio, it is very handy to have available a single command that changes the display mode by just changing a single input optional argument. In facts the user might start with one of the five described display modes; after examining the document draft the user might chose another display mode, and it suffices to change the optional argument, without changing the whole code.

WARNING This package performs well with documents typeset in `twoside` mode, and with a page design where the internal margins of both the odd and even pages are equal (symmetrical page design). For example, it does not work with this document designed to work in `oneside` mode and where the the left page margin is always larger than the right one of each page; this page design is functional to the documentation of the T_EX system software.

2 Installation

This package is already installed with any complete and up to date T_EX system distribution, T_EX Live or MiK_TE_X.

Should the user have available a basic or a partial T_EX system installation, the simplest way to install this package is to download the `swfigure.zip` file from the Comprehensive T_EX Archive Network (CTAN), and decompress it either in the very folder where there is the document main or single document file or, for a general use, in the user personal `texmf` tree; it is possible that the user should

directly create this personal tree; how to do it is described in the documentation of the user \TeX system distribution.

The same holds true if the user employs a vintage \TeX system distribution; this package requires the \LaTeX 3 modern language functionalities, therefore a \LaTeX kernel with a date after 2019-01-01. Lacking this modern kernel, the package is strongly limited, because packages `xparse` and `xfp` are used for its internal workings.

3 Usage

This package defines a single user command, `\DFimage` to be used with the following syntax:

```
\DFimage[⟨display mode⟩]{⟨image file name⟩}[⟨lof
entry⟩]{⟨caption⟩}[⟨label⟩]%
    (⟨height correction⟩)<⟨line correction⟩>\meta{width test}
```

where:

- ⟨*display mode*⟩ is one of the following uppercase acronyms: **NF** (Normal Figure), **RF** (Rotated Figure), **SW** (Spread Wide image), **HS** (Horizontal Spread-wide image), and **VS** (Vertical Slim image).
- ⟨*image file name*⟩ Is the name of the image graphic file; remember that the \LaTeX -based \TeX system typesetting programs accept graphic files in the formats with extensions `.pdf`, `.eps`, `.jpg`, `.png`, and few other less common ones. It is not necessary to specify the extension, but it is not forbidden.
- ⟨*lof entry*⟩ is the optional entry to the List Of Figures; it defaults to the ⟨*caption*⟩ text, but if the latter is sort of lengthy, it is better to enter a shorter text in that list.
- ⟨*caption*⟩ is the caption text.
- ⟨*label*⟩ this optional argument is the string that forms the argument of the `\label` command; of course, if this argument is not specified, the figure number and its page cannot be referenced with `\ref`, `\pageref` and other similar commands.
- ⟨*height correction*⟩ This normal parenthesis delimited optional argument is a fractional number lower than 1 (default 0.8) with which to further scale the scaled image height to be included; it is used by some display modes, and it is ignored by others.
- ⟨*line correction*⟩ is an angle bracket delimited optional argument preset to zero. It is relevant only with the slim vertical image display mode. Sometimes the `wrapfig` package indents the wrapping text in such a way as to leave too much or too little vertical space around the image and its caption; by examining the document drafts it is possible to increase or decrease the vertical space by any (integer) number of lines.
- ⟨*width test*⟩ this further vertical bars delimited argument is used only when dealing with vertical slim images; when they are scaled down in order to fit in the available space, their width may become too small to allow a decently typeset caption underneath the image. A test is made to compare the scaled image width with a fraction of the text block width in order to skip the image insertion if the image width becomes too small; the default value for

this *⟨width test⟩* fraction is 0.25, but the user can specify a different value, even zero; with the zero value the width test is skipped.

4 Display mode peculiarities

Each of the listed display modes has its own pros and cons. In the following the phrase “aspect ratio” refers to the “height over width” ratio either of the image or of the available space where to put the image and possibly its caption.

- NF This display mode is convenient when the image aspect ratio is close to the text block one. Of course the user does not know in advance the vertical space occupied by the caption, therefore the optional *⟨height correction⟩* comes handy for small adjustments; this display mode is fully and freely floating, although its positioning is fixed to [p].
- RF This display mode is convenient when the image aspect ratio is close to the reciprocal of the text block one. This mode accepts the *⟨height correction⟩* in order to leave space to the caption. It is a fully and freely floating object with the same pros and cons as the NF display mode.
- SW This display mode is convenient when a really large image requires two pages to display all its details; its aspect ratio should be close to the two page aspect ratio of the total space available for the two page spread; depending on the page design this total space approximate value probably is close to 0.5. The object to be included in the document should start on an even page, therefore it moves to the first even page available; this implies that the last document page before the spread wide image may be blank or may be shorter than the other text pages; the user then should help the automatic positioning by choosing very carefully where to insert the `\DFfigure` command in the source text; it can even set it within a paragraph and the command provides a new page just after the line that contains the `\DFfigure` command; therefore if the paragraph is close to the end of an odd page and the position of `\DFfigure` command is properly positioned within this paragraph, the spread wide insertion would not disturb the quality of the typeset document.
- HS This display mode is convenient when the initial image aspect ratio is very small; if it is smaller than 0.3 its insertion leaves enough space beneath the two image halves (plus caption) so that normal text can fill the space under such spread wide slim figure. The constraints described for display mode SW apply also to this mode.
- VS This mode is convenient for tall and slim figures with aspect ratio not lower than 2; but for obvious reasons, it should not be too large, let’s say not higher than about 3. As always this depends on the page design and the caption size. The limitations of the underlying of the `wrapfig` package forbids its usage too close to section headings, explicit lists and texts typeset in special modes with different measure than normal text. Again it is up to the user to chose where to insert the `\DFfigure` command. If the aspect ratio of the image to include is lower than 2, this message is printed in the document where the image should appear (of course with `#1` replaced by the actual image file name):

```

=====
Figure #1 is not tall enough.
Consider using a display mode different
from VS; may be NF or RF are better suited.
=====
Nothing done!
=====

```

A similar action and a similar message is output if the scaled image width becomes too small.

5 Acknowledgements

This package would not exist if Heinrich Flech did not need to insert several types of large images in his documents; he asked me to create suitable commands and/or environments and he tested all the series of progressive developments of this package; believing that this package might be useful to other T_EX users, he invited me to submit the result to the T_EX community, and here it is. Heinrich did not want his name as an author, because he insists that he was “just” a tester. He was precious and I warmly thank him.

Thanks also to Benedict Wilde who spotted some errors and suggested the use of a specific setting for the PDF viewer.

References

- [1] The L^AT_EX3 Project, *The **xparse** package* — Document command parser. Release date 2020-05-15. PDF document readable with `texdoc xparse`.
- [2] The L^AT_EX3 Project, *The **xfp** package* — Floating Point Unit. Release date 2020-05-15. PDF document readable with `texdoc xfp`.
- [3] The L^AT_EX3 Project, *The L^AT_EX3 interfaces*. Release date 2020-09-24. PDF Document readable with `texdoc interface3`.
- [4] D.P. Carlisle and The L^AT_EX3 Project, *Packages in the ‘graphics’ bundle*. Release date 2020-08-21. PDF document readable with `texdoc grfguide`.
- [5] P. Lehman and J Wright, *the **etoolbox** Package* — An e-TeX Toolbox for Class and Package Authors. version 2.5j, released on 2020-08-24. PDF document readable with `texdoc etoolbox`.
- [6] D.P. Carlisle, *The **lscape** package*. Version 3.02 released on 2020-05-28. PDF document readable with `texdoc lscape`.
- [7] D.P. Carlisle, *The **afterpage** package*. Version 1.08 released on 2014-10-28. PDF document readable with `texdoc afterpage`.
- [8] D. Arseneau, *The **wrapfig** package*. Version 3.6 released on 2003-01-31.
- [9] P. Wilson, *The Memoir Class for Configurable Typesetting* —User guide. Version 3.7m released 2020-09-10. PDF document readable with `texdoc memman`.

6 The code

The required TeX format and its date, and the identification of this package have already been inserted by the initial commands of this documented file.

Here we call the initial required packages and, since the necessary software to define some L^AT_EX 3 (L³) language “functions” are already part of the L^AT_EX kernel, we just use them in order to define a robust L^AT_EX 2_ε/L³ interface to create a testing macro that accepts many logical operators that act on items that, besides being boolean variables, are also numerical expressions connected with relation operators; for further details it suffices to examine the `interface3.pdf` file, that is integral part of any recent TeX system distribution.

The syntax of the new testing macro is the following:

`\fptest{<test>}{<true>}{<false>}`

The logical operators that are usable in the `<text>` phrase are `||` (OR), `&&` (and), and `!` (NOT), plus other less common ones; the `!` negates the status of boolean variables, but also the numerical relation operators; these, on turn, may be paired so that, for example `>=` is the same as `!<`.

The `trace` package is still present in this beta version of the package; it will not be present in the next stable versions. Packages `graphicx`, `afterpage`, and `wrapfig` are functional for this package. In order to avoid “Option Clash” messages, such packages are loaded without any option; should the user load some or all of them with options, this operation must be done *before* loading this package.

```
1 \RequirePackage{etoolbox}
2 \RequirePackage{xfp}
3
4 \ExplSyntaxOn
5 \AtBeginDocument{%
6 \ProvideExpandableDocumentCommand\fptest{m m m}{%
7   \fp_compare:nTF{#1}{#2}{#3}}
8 }
9 \ExplSyntaxOff
10
11 \RequirePackage{trace}
12
13 \RequirePackage{graphicx}
14 \RequirePackage{afterpage}
15 \RequirePackage{wrapfig}
16
```

We define some dimension and some counter registers; to some dimensions we assign parameters relating to the page design. We also compute the dimensions of the available spread space so as to have available the elements to compare the aspect ratios of the images and of the available spaces. The `\if@SWtmode` switch is used only by the `SW` display mode in order to save the typesetting mode that was active when the `\DFimage` command was used in the source file; if it was in text mode, the switch is set to `true` and no spaces should be present after the last argument of that macro; the result is that the `\SWfigure` service macro shifts to vertical mode, but as soon as it is completed the switch test restores the text mode.

```
17 \newdimen\internalmargin
```

```

18 \internalmargin=\dimexpr\oddsidemargin+1in+1bp\relax
19 \newdimen\spreadwidth
20 \spreadwidth=\dimexpr 2\textwidth+2\internalmargin\relax
21 %
22 \newdimen\DFwidth
23 \newdimen\DFheight
24 \newdimen\DFhalfwidth
25 \newdimen\DFheight
26 \newdimen\FigSpace
27 \newsavebox\DFtotalimage
28 \newsavebox\DFimageI
29 \newsavebox\DFimageII
30 \newsavebox\RFbox
31 \newdimen\VS@textwidth
32 \newcount\VS@lines
33 \newif\if@SWmode
34

```

The following caption-like command recompiles the captions within a zero height vertical box and deals with the mandatory and optional arguments according to the following syntax:

`\DFcaption[$\langle lof entry \rangle$]{ $\langle caption \rangle$ }[$\langle label \rangle$]`

Notice that the `]lof` entry defaults to the full caption text if a different text is not explicitly entered. This command is specific those those display modes that compose the caption in vertical mode, therefore the measure is the `\textheight`.

```

35 \NewDocumentCommand\DFcaption{0{#2} m o}{\refstepcounter{figure}%
36 \vtop to 0pt{\hsize=\textheight\parindent=0pt\leavevmode
37 Figure \thefigure\quad #2\vss}%
38 \addcontentsline{lof}{figure}{\protect\numberline{\thefigure}#1}
39 \IfValueT{#3}{\label{#3}}\relax%
40 }
41

```

The `\cleartoevenpage` command may already exist in the document class. We prefer to make an absolute redefinition so as to be sure that it performs as we need in this package. We tested with the `memoir` class that already has such a command, and no problems were found

```

42 \DeclareRobustCommand\cleartoevenpage{%
43 \clearpage\ifodd\value{page}\null\clearpage\fi}

```

Now we define the “real” user macro necessary to chose the $\langle display mode \rangle$ and the various parameters necessary for the desired one. See section 3 on page 3 for its syntax and the other necessary arguments. Notice that only two arguments are mandatory: the $\langle image file name \rangle$ and the $\langle caption \rangle$ text. All the other five optional parameters are delimited by various delimiters; the first one, $\langle display mode \rangle$ has as default setting the acronym `SW` for a “Spread-Wide” display mode; this mode was the one this package was initially conceived for, and its initials recall this priority. All other modes came afterwards. the other optional parameters may have a default value, but they are ignored by certain modes that don use those parameters. As for the previous macro `\DFimage`, here the $\langle lof entry \rangle$ defaults to the full $\langle caption \rangle$ text.

Actually `\DFimage` just examines the $\langle display mode \rangle$, and accordingly passes the necessary parameters to the actual macros that implement the various mode displays. It outputs an error message if the $\langle display mode \rangle$ acronym was misspelt.

```

44 \NewDocumentEnvironment{DFimage}{%
45     O{SW} m O{#4} m o D(){0.8} D<>{0} D||{0.25}}{%
46 \ifstrequal{#1}{SW}%
47     {%
48         \SWfigure{#2}[#3]{#4}[#5]%
49     }%
50 {\ifstrequal{#1}{NF}%
51     {%
52         \NFfigure{#2}[#3]{#4}[#5](#6)%
53     }%
54 {\ifstrequal{#1}{RF}% OK
55     {%
56         \RFfigure{#2}[#3]{#4}[#5]%
57     }%
58 {\ifstrequal{#1}{VS}%
59     {%
60         \VSfigure{#2}[#3]{#4}[#5](#6)<#7>|#8|}%
61     }%
62 {\ifstrequal{#1}{HS}
63     {%
64         \HSfigure{#2}[#3]{#4}[#5](#6)%
65     }%
66     {\PackageWarning{swfigure}%
67         {%
68             *****\MessageBreak
69             Option #1\space is not valid      \MessageBreak
70             Nothing done                      \MessageBreak
71             Image #2\space was not processed \MessageBreak
72             *****\MessageBreak}%
73     }%
74 }%
75 }%
76 }%
77 }%
78 }%
79

```

In the following subsection the five specific display mode macros are examined and commented.

6.1 The `\SWfigure` macro for Spread Wide images

The specific macro for spread-wide images has the following syntax:

```
\SWfigure{ $\langle image file name \rangle$ }[ $\langle lof entry \rangle$ ]{ $\langle caption \rangle$ }[ $\langle label \rangle$ ]
```

The various arguments have already been described; as you see the $\langle display mode \rangle$ argument is not there anymore, as well as the last two optional arguments of `\DFfigure`.

The working strategy is the following: first the typesetting mode is tested and, in case, a filling of the current line is performed and, by means of a `\vadjust` \TeX

native macro, a `\newpage` command is executed; in any case a `\cleartoevenpage` is executed so as to be sure to start any action while an even page is being processed. Then the initial image to be divided in two halves is stored in a box; this box is examined in order to find its total width and height, so as to be able to compute the aspect ratio. All other dimensions are examined to have available all the necessary data to display the image halves in the requested way.

Notice that the `trim` option to `\includegraphics`, used to divide the initial image in two exact halves, (apparently) must be used without specifying other options; but, most important, it requires real numbers, not macros; therefore we have to make use of the usual dirty trick of defining an expanded macro `\x` within a group that the macro itself closes upon its expansion; in this way the principal part of the `\includegraphics` macro contains only keywords and numbers and may be executed without errors.

Let W be the total spread width and w be the initial image width; let H be the text height and h the initial image height; compute the A_W to be the aspect ratio of the spread available space H/W . Similarly let A_{fig} aspect ratio of the initial image; we have to scale the image with a scaling factor such that the image keeps its aspect ratio, but its scaled height does not exceed the text block height, and its scaled width does not exceed the spread width; of course the aim is to maintain the scaled image as large as possible therefore we have to choose the most suited of the two scaling factors that can be computed from the given data and the inequalities described above. In mathematical terms we have to choose the initial image scaling factor S such that:

$$S = \begin{cases} w/W & \text{if } A_{\text{fig}} \leq A_W \\ h/H & \text{if } A_{\text{fig}} > A_W \end{cases} \quad (1)$$

This choice is made by testing the various ratios by means of the L3 `\fpctest` macro, that computes such ratios and compares them. The commented line below shows exactly which lengths are used to compute the ratios, and which comparison is executed. Once the scaling factors are found the boxes containing the half figures are scaled and further on are inserted in floating figure environments; we use another dirty trick of matryoshka dolls; this metaphor describes nested `\makeboxes` used to mask their actual widths, but capable to push the images towards the spread spine. Eventually the caption is set in the right margin of the right spread page, rotated 90° counterclockwise.

```

80 \NewDocumentCommand\SWfigure{m O{#3} m o}{%
81 \ifvmode
82   \@SWtmodefalse
83 \else
84   \@SWtmodetrue
85   \csuse{@bsphack}%
86   \adjust{\vspace{0pt plus1fill}}\csuse{@esphack}\newpage}%
87 \fi
88 \cleartoevenpage
89 \setbox\DFtotalimage=\hbox{\includegraphics{#1}}%
90 \DFwidth=\wd\DFtotalimage \DFhalfwidth=0.5\DFwidth
91 \DFheight=\ht\DFtotalimage
92 \FigSpace=\dimexpr\textwidth+\internalmargin\relax
93 %
94 \setbox\DFimageI\hbox{\bgroup
95 \edef\x{\egroup\noexpand\includegraphics*[%

```

```

96 trim = 0 0 \the\DFhalfwidth\space 0]\x{#1}}%
97 %
98 \setbox\DFimageII\hbox{\bgroup
99 \edef\x{\egroup\noexpand\includegraphics*[%
100 trim = \the\DFhalfwidth\space 0 0 0]\x{#1}}%
101 %
102 %          h/w          >          H/W
103 \fpptest{\DFheight/\DFhalfwidth > \textheight/\FigSpace}%
104 {\edef\DFscalefactor{\fpeval{\textheight/\DFheight}}}%
105 {\edef\DFscalefactor{\fpeval{\FigSpace/\DFhalfwidth}}}%
106 %
107 \setbox\DFimageI=\hbox{\scalebox{\DFscalefactor}{\usebox{\DFimageI}}}%
108 \setbox\DFimageII=\hbox{\scalebox{\DFscalefactor}{\usebox{\DFimageII}}}%
109 %
110 \begin{figure}[p]%
111 \vbox to\textheight{\vss\hsize=\textwidth\
112 \makebox[\hsize][l]{\makebox[\FigSpace][r]{\box\DFimageI}}\vss}%
113 \end{figure}\clearpage
114 %
115 \begin{figure}[p]%
116 \vbox to\textheight{\vss\hsize=\textwidth
117 \makebox[\hsize][r]{\makebox[\FigSpace][l]{\box\DFimageII}}%
118 % Rotated caption in the right page
119 \makebox(0,0)[lb]{\hspace*{1em}\raisebox{0.5\textheight}{%
120 \rotatebox[origin=tc]{90}{\DFcaption[#2]{#3}{#4}}}%
121 %
122 }}}\vss}%
123 \end{figure}\clearpage\if@SWtmode\noindent\fi}
124

```

6.2 The \NFfigure macro for normal full page figures

This code is without surprises, except that it receives from the `DFimage` steering macro five arguments, the last (optional) one being delimited by round parentheses; by default, it contains the value 0.8, but the user can pass to the steering macro a different value in order to reduce the actual image size so as to leave enough space beneath it for the caption; the user, in facts, does not know in advance how many lines would occupy a structured complex caption.

```

125 \NewDocumentCommand\NFfigure{m O{#3} m o d()}{%
126   \begin{figure}[p]
127     \includegraphics[width=\linewidth,height=#5\textheight,
128                     keepaspectratio]{#1}%
129     \caption[#2]{#3}%
130     \IfValueT{#4}{\label{#4}}\relax
131   \end{figure}
132 }
133

```

6.3 The \RFfigure macro for rotated figures

There are no new comments to make about this almost standard way of displaying a large figure, except, perhaps, the fact the instead of invoking the `lscap` package, the whole figure is rotated by means of rotating the very box that was used to

contain the scaled image; the steering macro does not pass any *<height correction>* argument to this macro, because it automatically scales the box containing the image and the caption so as to fill up the height or the width of the text block.

```

134 %\NewDocumentCommand\RFfigure{m o m o}{%
135 \NewDocumentCommand\RFfigure{m O{#3} m o}{%
136     \dimen8=\textwidth\dimen10=\textheight
137     \figure[p]\setbox\RFbox=\hbox{%
138     \rotatebox[origin=cc]{90}{\parbox[b][\dimen8][c]{\dimen10}%
139     {\centering\includegraphics[width=\dimen10, height=\dimen8,
140     keepaspectratio]{#1}%
141     \caption[#2]{#3}%
142     \IfValueT{#4}{\label{#4}}\relax%
143     }}}%
144     \edef\RFx{\fpeval{\ht\RFbox/\textheight}}}%
145     \edef\RFy{\fpeval{\wd\RFbox/\textwidth}}}%
146     \fptest{\RFx > \RFy}%
147     {\scalebox{\RFx}{\box\RFbox}}}%
148     {\scalebox{\RFy}{\box\RFbox}}}%
149     \endfigure}
150

```

6.4 The \VSfigure macro for tall and slim figures

This is the most critical macro (because it depends on the `wrapfig` package) in order to set the image with its caption on the external side of the text block, with the text flowing around it. Package `wrapfig` has several limitations concerning the text that flows around the image. Nevertheless with a “normal” textual contents of the text block the result may be very good.

The user should help a little bit the correct performance of this macro; for this reason, besides the *<height correction>* factor, the steering macro may pass to it also the *<line correction>* in order to correct the amount of lines the surrounding text should be indented; by playing with both corrections good results may be obtained.

Notice that only with this macro a test on the initial image aspect ratio is performed; should this image have an aspect ratio smaller than 2, the process is aborted and a message is printed in the output document that informs the user and recommends to use another *<display mode>* acronym.

Furthermore, should the image be very slim and tall, the *<height correction>* may conveniently reduce its height, but, in order to avoid changing its aspect ratio, it reduces also its width; eventually the width is reduced to the point that beneath the image there would not be enough space for a decently typeset caption. A further optional delimited argument is available: its delimiting tokens are two vertical bars, | |, and its default value is 0.25. This means that if the scaled image becomes narrower than this coefficient times the text width, an other message is typeset in place of the figure. Depending of the various elements that influence these tests, the user can try different values for this coefficient, even reaching the value zero, that eliminates such test on the scaled image width.

```

151 \NewDocumentCommand\VSfigure{m O{#3} m o d() d<> d||}{%
152     \setbox\DFtotalimage=\hbox{\includegraphics{#1}}
153     \DFwidth=\wd\DFtotalimage \DFheight=\ht\DFtotalimage
154     \edef\VS@aspectratio{\fpeval{\DFheight/\DFwidth}}

```

```

155 \fptest{ \VS@aspectratio < 2 }%
156 {\begin{center}\ttfamily
157 =====\\
158 Image \textit{#1} is not tall enough.\\
159 Consider using a display mode different\\
160 from VS; may be NF or RF are better suited.\\
161 =====\\
162 Nothing done!\\
163 =====
164 \end{center}
165 }%
166 {\edef\VS@factor{\fpeval{#5\textheight/\DFheight}}%
167 \fptest{\VS@factor>1 || \VS@factor\DFwidth < #7\textwidth}%
168 {%
169 \begin{center}\ttfamily
170 =====\\
171 The scaled image \textit{#1} is too slim. \\
172 Maybe directly using the wrapfig package might \\
173 solve this problem. \\
174 =====\\
175 Nothing done!\\
176 =====
177 \end{center}
178 }%
179 {\setbox\DFtotalimage=
180 \hbox{\scalebox{\VS@factor}{\box\DFtotalimage}}%
181 \edef\VS@width{\fpeval{\VS@factor*\DFwidth}\p}%
182 \setbox\DFtotalimage=\vbox{\hsize=\VS@width%
183 \def\@cuptype{figure}\box\DFtotalimage
184 \caption[#2]{#3}%
185 \IfValueT{#4}{\label{#4}}\relax}
186 \VS@lines=
187 \fpeval{round(\ht\DFtotalimage/\baselineskip,0)+#6}%
188 \begin{wrapfigure}[\VS@lines]{0}[0pt]{\VS@width}%
189 \box\DFtotalimage
190 \end{wrapfigure}%
191 }%
192 }%
193 }%
194

```

6.5 The \HSfigure macro for spread wide slim figures

The beginning of the macro is not so different from that used for the `SWfigure` macro. But this time there is no alternative to scale the image taking into account only its width and the spread width. The new little problem, now, is to match the total height of the two halves of the original wide and slim image, so that the texts of the two facing pages have their respective first lines perfectly aligned.

Since the right half of the figure contains also the caption it is necessary to box again the left page box so as to have the same height and depth as the right page one.

```

195 \NewDocumentCommand\HSfigure{m o m o d()}{%
196 \unless\ifvmode

```

```

197 \csuse{@bsphack}%
198 \vadjust{\vspace{0pt plus1fill}}\csuse{@esphack}\newpage}%
199 \fi
200 \cleartoevenpage
201 \setbox\DFtotalimage=\hbox{\includegraphics{#1}}
202 \DFwidth=\wd\DFtotalimage \DFhalfwidth=0.5\DFwidth
203 \FigSpace=0.5\spreadwidth% W
204 %
205 \setbox\DFimageI\hbox{\bgroup
206 \edef\x{\egroup\noexpand\includegraphics*[%
207 trim = 0 0 \the\DFhalfwidth\space 0]}\x{#1}}%
208 %
209 \setbox\DFimageII\hbox{\bgroup
210 \edef\x{\egroup\noexpand\includegraphics*[%
211 trim = \the\DFhalfwidth\space 0 0 0]}\x{#1}}%
212 %
213 \edef\DFscalefactor{\fpeval{\FigSpace/\DFhalfwidth}}%
214 %
215 \setbox\DFimageI=\hbox{%
216 \scalebox{\DFscalefactor}{\usebox{\DFimageI}}}%
217 \setbox\DFimageII=\hbox{%
218 \scalebox{\DFscalefactor}{\usebox{\DFimageII}}}%
219 \setbox\DFimageI=
220 \hbox{\dimen10=\linewidth\dimen8\internalmargin
221 \vbox{\hsize\DFhalfwidth\parindent\z@
222 \box\DFimageII\par
223 \leavevmode\hspace*\dimen8}%
224 \vtop{\hsize\dimen10\parindent\z@
225 \textwidth=\hsize
226 \DFcaption[#2]{#3}{#4}%
227 }%
228 }%
229 }%
230 \setbox\DFimageI=\vbox to\ht\DFimageII{\box\DFimageI\vss}%
231 %
232 \begin{figure}[t]
233 \makebox[\textwidth][l]{%
234 \makebox[\DFhalfwidth][l]{\box\DFimageI}}%
235 \end{figure}
236 \afterpage{%
237 \begin{figure}[t]
238 \makebox[\textwidth][r]{%
239 \makebox[\DFhalfwidth][r]{\box\DFimageII}}%
240 \end{figure}%
241 }
242 }

```

HAPPY L_TE_X-ING