Managing large figures

Claudio Beccari and Heinrich Fleck

Abstract Sometimes a document may require the insertion of very large images, such that the best solution might be to insert them so as to occupy a full two page spread without leaving any internal margin. Depending on the image aspect ratio it might be preferable to look for other solutions. It looks a simple problem, but the aspect ratio of the original image may set forth some problems that require special treatment. With package swfigure described in this paper we try to create a single command or environment that can handle several display modes.

Sommario In certe circostanze è necessario inserire in un documento delle immagini molto grandi, tali che la soluzione migliore sarebbe di inserirle in modo da occupare due pagine consecutive affacciate così da usare completamente anche il margine interno. Ma, a seconda del rapporto di forma potrebbe essere necessario ricorrere ad altre soluzioni. La cosa sembra semplice, ma il coefficiente di forma dell'immagine originale può dare luogo a problemi che richiedono un trattamento speciale. Con il pacchetto swfigure descritto in questo articolo si crea un singolo comando o ambiente che può gestire diversi modi di presentazione.

1. Introduction

Author H. Fleck already published on ArsTEXnica (Beccari e Fleck, 2012) a class style suitable for typesetting dictionaries; his main use at that time was to publish a dictionary on astronomy. He worked also on other such documents, some of which are in the public domain; he created also another dictionary on seamanship. In both dictionaries he needed to insert large figures, such as figure 1 or figure 2. Even if they are typeset at full page width, they do not show enough details; it would be helpful if this kind of images could be printed in a spread wide fashion. This package, swfigure, can do it and offers more functionalities; it is already installed with the TEX system distributions; the documentation of its code is in (Beccari, 2020a) and a small gallery of examples is in (Beccari, 2020b).

If the document is read on screen, there are around several PDF viewing programs that can display two PDF pages at a time, with the even numbered page on the left and the odd numbered one on the right; it is even possible to annihilate the little gap between the visible pages, so that the screen view displays exactly as a single page containing the spread. With package pdfpages (Matthias, 2020) it is even possible to reformat the output PDF file, so that each page is a complete spread of the original document.

If the document gets printed and bound in a proper way with sewn signatures, each spread of the document displays the adjacent half figures, so that the spine is almost invisible, and the two half pictures appear as a single one.

This is the problem: if we go into the details, apparently the whole procedure consists in splitting the original image into two halves, and to scale them so that their heights do not exceed the text body height, and their widths do not exceed the text body plus the internal

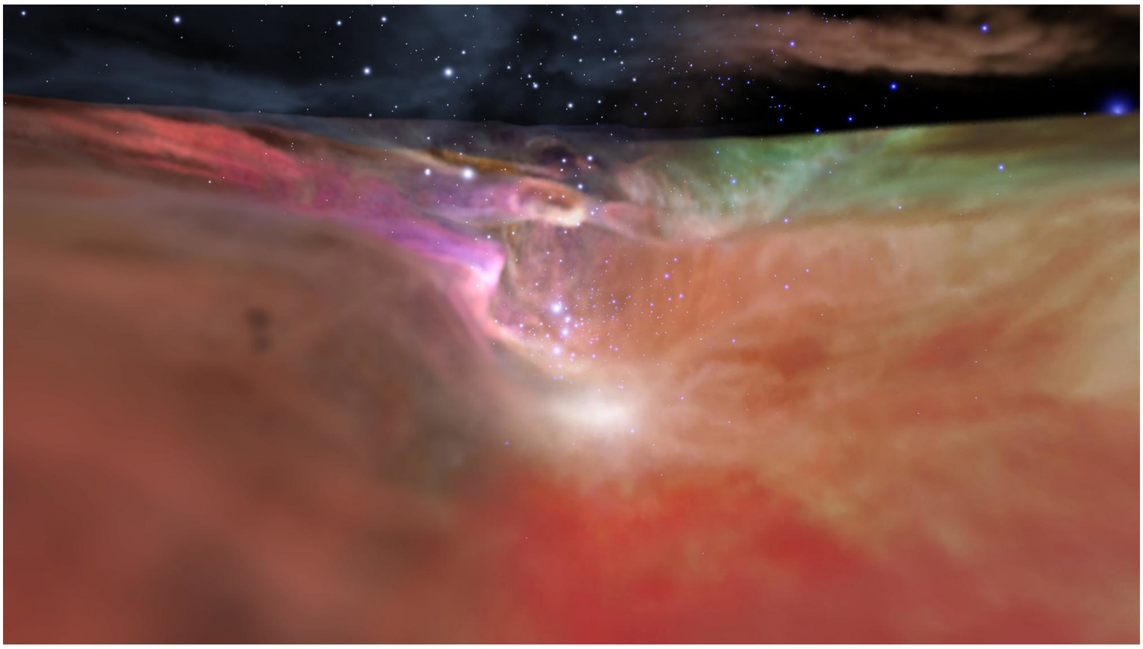


Figure 1. The Orion Valley (Hubble Space Telescope)

Figure 2. A reconstruction of a Greek trireme battle ship (Hellenic Navy Museum)

margin widths. Then such scaled images should be input so that the right or respectively the left sides touch the spine.

The possible scaling factors to reach the above goal are in general different, and in any case one of the two equals its upper limit, while the other one is smaller.

Experience shows that the width constraints are more stringent than the height ones; but the aspect ratio of the original picture might require special treatment.

Sometimes the aspect ratio of the original image is such that the above described display mode is not usable; it is evident that the spread-wide display mode is suited for images that have a height to width ratio (the aspect ratio) lower than 1. If this aspect ratio is slightly smaller than 1, it might be better to use a different display mode, for example a landscape rotated figure. If the aspect ratio is slightly higher than 1 the best display mode might be the standard full page mode. If the original image has an aspect ratio definitely larger or smaller than 1, and looks like a horizontal or vertical ribbon, other display modes might be preferable; if the image is already available in a separate A3 sized (PDF) page, yet another display mode is available to completely fill up the spread pages without leaving around any margin and omitting any resizing of the image;[1]; sometimes for certain slim figures a full paper height display might be preferable, while certain landscape images a full paper width display mode would be more suited.

The aim of the following section is precisely to describe eight different display modes that handle the above described situations, and to create a single command or environment that changes display mode according to an optional argument. In this way the user can choose a different mode without using different commands for each mode; the user can select the default spread-wide display mode and, according to the result, he can change the optional mode-argument and find out which is the best for that particular image.

2. Possible approaches

Here we give a detailed description to the *spread-wide solution* because it is quite tricky. Further on, we describe other possible solutions to the problem of managing large images with different aspect ratios; each of these solutions has advantages and disadvantages depending on the image aspect ratio. We examine several possible situations:

1. The spread-wide situation which forms the core of this paper; it occupies two facing pages with the image and its caption but without any text; the outside margins remain available. so that the spread facing pages may contain header, footer and, in principle, margin notes.
2. The normal full page figure that is defined by the LaTeX kernel.
3. The rotated full page rendering, that might be achieved also with the lscape package.
4. The text-wrapped rendering obtainable with the wrapfig package.
5. The spread-wide situation for pages containing a very long horizontal ribbon-like image at a two page spread top, and filled up with text in the remaining part of the pages.

[1] We prefer to deal with ISO standard paper sizes, but the macros we develop are usable also with US paper sizes, although with less accuracy in filling the margins.

6. The total height situation where the image reaches the top and bottom margins of the page, but leaves some space for a lateral caption.
7. The total width situation where the image and its caption reach the lateral page borders, but some space remains underneath for some text.
8. The full spread situation that completely covers the spread pages, possibly without leaving any margin around.

Some of these situations allow freely floating objects; others allow to float them only to a certain extent and require some manual intervention by the user.

The whole process is performed by a single user command that accepts options; depending on the mode related option, it inserts the image with its caption according to one of the above described situations.

It goes by itself that in order to have a smooth output of such large figures, the figure stack should be empty and should remain empty after each one of such large images is shipped out to the output file. Therefore another figure may be declared several paragraphs after any such large image.

3. Spread-wide strategy

The strategy to solve the described spread-wide display mode is the following.

Let us refer to figure 3. The spread has two facing pages containing two text blocks separated by a distance equal to twice the internal margin; therefore the total space for the spread-wide image is a rectangle with an height equal to the text block height H, while the width W is as wide as twice the width of the text block T plus the internal margin width M. Its aspect ratio is therefore $A_1 = HW$.

The initial image dimensions are w and h with aspect ratio equal to $A_2 = hw$. In order to fit the image within the available spread area it is necessary to scale the image so that its scaled width w_s does not exceed W, *and* its scaled height h_s does not exceed H; at the same time the image should be as large as possible while satisfying the above constraints.

Therefore the scaling factor s_f should satisfy these constraints:

$$s_\mathrm{f} = \begin{cases} wW & \text{if } A_2 \leq A_1 \\ hH & \text{if } A_2 > A_1 \end{cases} \tag{1}$$

4. Problems

The reader should consider the aspect ratio $A_2 = hw$ of the original image and the aspect ratio $A_1 = HW = H2T\ \ M$ of the available space; the former, in principle, might be of any value; the latter is almost certainly smaller than one, depending on the page design and the paper size. Another element to consider is the text block aspect ratio $A_3 = HT$.

If A_2 is significantly larger than 1, the original figure is so slim that a spread wide rendering would not be suitable compared to the same figure in a single page. So avoid using this procedure with images with an aspect ratio $A_2 > A_3$, but use the sigle page rendering or the

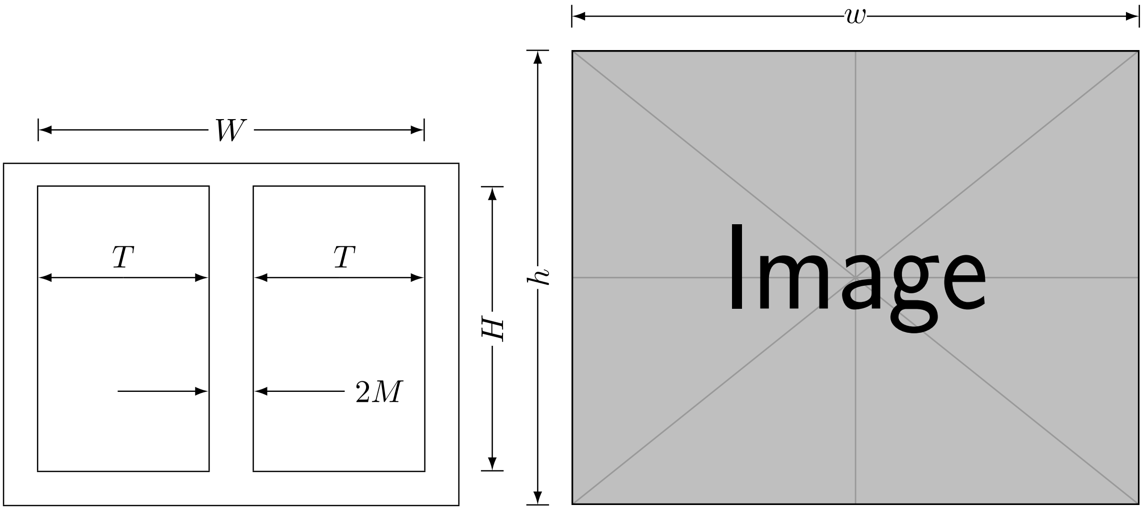


Figure 3. Schematics of the spread page layout and of a dummy image (from package mwe (Scharrer, 2018))

wrapped vertical rendering. If the A_2 is larger than, say, 2, then it would be preferable to use a text wrapped display mode: if A_2 is definitely smaller than A_1 the ribbon like image may find its place on top of a spread, with the rest of the page filled up with text. The display mode, therefore, must be a user choice.

On the opposite, if A_2 is slightly lower than 1, the original image is squarish; dividing it in two halves to set each half on the facing pages of a spread might yield poor results; may be a 90° rotated figure in a single page might be a better solution.

The spread wide solution is preferable when its A_2 aspect ratio is pretty close to A_1. The scaling will produce the best effects, and the spread wide rendering would be a valid solution; it is difficult to specify how close the aspect ratios should be: it depends on the figures themselves and on the inequalities of equation (1).

Another possible problematic situation is when A_2 is rather small, say, smaller than 0.5; the scaling would produce a narrow image strip across the spread, with a lot of white space around; probably this strip might look better if if was at the top of the page, and the rest of the page would be occupied by by the document text. There has been a paper on ArsTeXnica (De Marco e Dominici, 2008) where a similar problem has been solved with another approach; it actually does not address ribbon like images, therefore a different solution must be created.

A final situation is possible when the user needs to include a complex page that is fully and easily readable only if it is printed, or is directly compiled, in A3 format. Of course in this case it would be possible to find other solutions; for example the A3 sheet is printed by itself and then it is joined to the A4 sized document by simply gluing it to the bound document; or, for electronic ones, simply hyperlinking it from the main document. But for screen reading, the insertion as a full spread image, covering also the external/lateral margins would be a solution similar to that for a regular spread-wide one, except for the margins.

The conclusion of this section is simple: the swfigure package can solve the eight described display modes. In few words they are the following.

1. A_2 is very small, say, smaller than 0.5: use a spread-wide horizontal strip.
2. $A_2 \approx A_1$: use a full two page spread-wide rendering.
3. $A_1 < A_2 < A_3$: use a rotated figure in vertical landscape mode.
4. $A_2 \approx A_3$: use a normal full page figure.
5. A_2 is large, say, larger than 2: use a wrapped figure.
6. $A_2 \approx \sqrt{2}$ or $A_2 \approx \sqrt{0.5}$: depending on its contents, it might be preferable to use a full spread figure.

Of course the inequalities of the above enumeration should be evaluated with good sense by a human rather than by an algorithm. The swfigure package takes care of all these situations, but sometimes some human intervention is required in order to specify suitable options.

5. The caption

The LaTeX based typesetting programs can do all the above described processing; in a later section we show the details. But there is another delicate question; where do we put the figure caption?

The answer is not so easy; consider some well typeset magazines, such as, for example, the well known National Geographic, where often they publish beautiful pages much larger than the normal paper width; sometimes such pages are folded one or more times. Depending on the contents of the folded page, either they may label the single sub images, or they are forced to put the caption in the previous or following page.

The book by Gabrielli (1974) is an art book the pagination of which differs almost from page to page, depending on the images that have to be shown; some of them are spread wide images that sometimes allow some space for text or a very long descriptive caption; see for example the photograph in figure 4.

Another industrial art book by Sassi Perino e Faraggiana (1995) is peculiar, not only for its unusual contents, that mixes reproductions of the original design drawings by the engineers or architects who designed the bridges of Turin, but also for the pagination that contains several air borne spread wide images of some of those bridges; furthermore the final trimmed page shape is square; see for example the photograph in figure 5

Now the approach we took was to create something that did not need any page folding; on the contrary we maintained intact the outer margin (except in one case), therefore we decided to put the spread wide figure caption in the right margin, therefore always on the odd numbered page; furthermore we decided to typeset it rotated by an angle of 90°; we decided to choose a measure equal to the scaled figure height; since the spread wide is intended for large figures, we assume that, in spite of scaling the figure, the caption measure might turn out to be close to the text block height. In the full spread case, on the opposite, we decided to vertically typeset the caption close to the image right border overprinting the included image and, depending on this image contents, to choose a contrasting color.

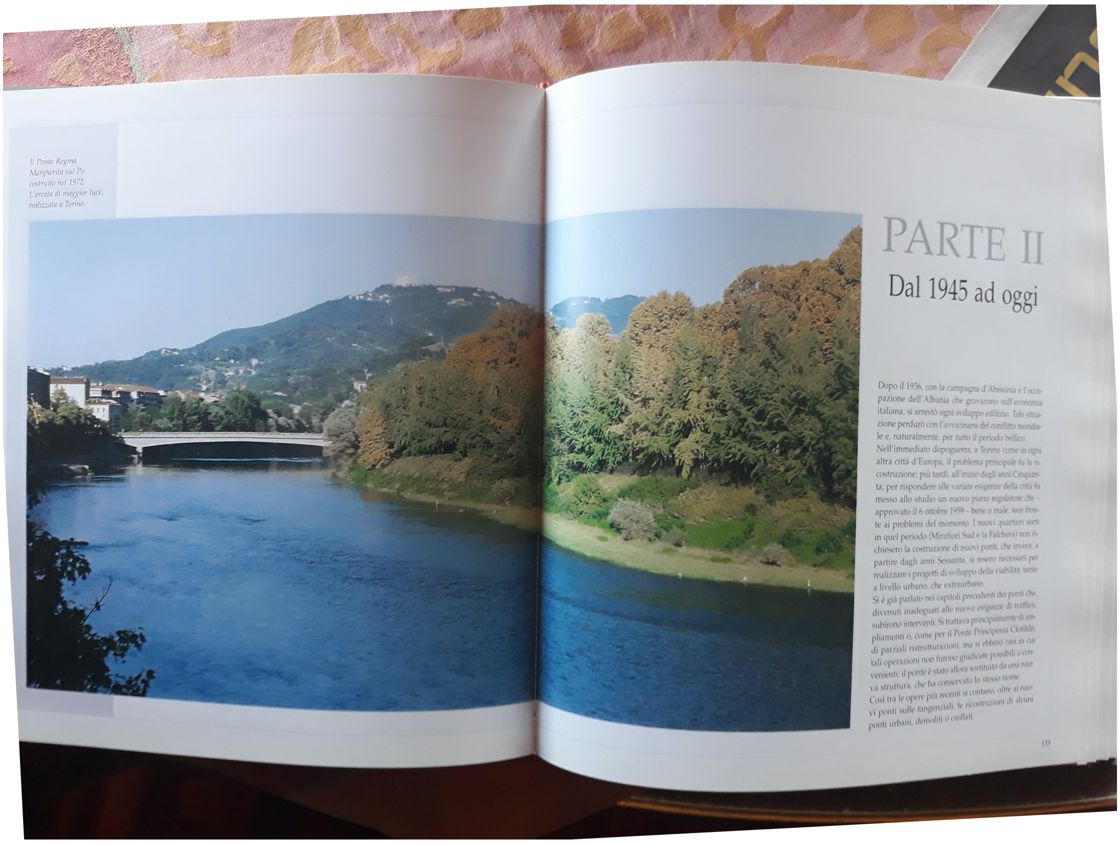
If the caption is not longer than a few lines the result appears to be quite acceptable; of course in future versions we might introduce some modifications or variants in order to produce different results. The sixth, seventh and eighth described modes are already additions to the previous versions functionalities.

Maestro d'Elva. Il marchese Ludovico II e i suoi dignitari. Inginocchiato di profilo ai piedi della Vergine, al collo l'ordine di San Michele Arcangelo conferitogli da Carlo VIII re di Francia, indossa una cappa in broccato d'oro a disegni di gusto quattrocentesco italiano, foderata d'ermellino. Sul prato fiorito il berretto presso il piede della Vergine, calzato da un sandalo che appare dal bordo ripiegato del vestito. Dietro Ludovico II la schiera dei dignitari fra i quali vi dovrebbero essere il Vicario Generale Pietro di Celle e i due fratelli del marchese Gian Giacomo Governatore di Carmagnola e Carlo Domenico, Abate di Staffarda, riconoscibili per il lungo naso appuntito caratteristica dei Saluzzo. Alle loro spalle in primo piano la figura di un giovane, le cui fattezze ricordano quelle di Francesco Cavassa, che a quell'epoca, 1499, godeva già la stima del marchese e ricopriva importanti cariche. Sul fondo il viso scarno di un sacerdote attempato. Si nota una profonda acutezza psicologica nel fermare le caratteristiche fisionomiche dei singoli personaggi. - Saluzzo, Casa Cavassa. Madonna della Misericordia, particolare.

[illegible] d'Elva. Margherita di Foix e le gentildonne del suo seguito. Inginocchiata ai piedi della Madonna della Misericordia, [illegible] con manica a campana tessuto d'oro, con disegni ispirati a motivi vegetali, ha presso di sé il primogenito [illegible] le dame della corte. La giovanissima in abito color rubino ritengo sia Margherita, figlia di Ludovico II [illegible] somiglia in modo impressionante. Bellissimi i tessuti dei costumi: broccati, velluti [illegible] altrettanto belli e giovanili, anche se bene caratterizzati, sono i visi delle donne del seguito della marchesa, [illegible]

257

Figure 4. A painting in an historical palace of Saluzzo, Italy, from (Gabrielli, 1974)



Il Ponte Regina
Margherita sul Po
costruito nel 1972.
L'arcata di maggior luce,
realizzata a Torino.

PARTE II

Dal 1945 ad oggi

Dopo il 1936, con la campagna d'Abissinia e l'occupazione dell'Albania che gravarono sull'economia italiana, si arrestò ogni sviluppo edilizio. Tale situazione perdurò con l'avvicinarsi del conflitto mondiale e, naturalmente, per tutto il periodo bellico.

Nell'immediato dopoguerra, a Torino come in ogni altra città d'Europa, il problema principale fu la ricostruzione; più tardi, all'inizio degli anni Cinquanta, per rispondere alle variate esigenze della città fu messo allo studio un nuovo piano regolatore che – approvato il 6 ottobre 1959 – bene o male, fece fronte ai problemi del momento. I nuovi quartieri sorti in quel periodo (Mirafiori Sud e la Falchera) non richiesero la costruzione di nuovi ponti, che invece, a partire dagli anni Sessanta, si resero necessari per realizzare i progetti di sviluppo della viabilità, tanto a livello urbano, che extraurbano.

Si è già parlato nei capitoli precedenti dei ponti che, divenuti inadeguati alle nuove esigenze di traffico, subirono interventi. Si trattava principalmente di ampliamenti o, come per il Ponte Principessa Clotilde, di parziali ristrutturazioni, ma si ebbero casi in cui tali operazioni non furono giudicate possibili o convenienti; il ponte è stato allora sostituito da una nuova struttura, che ha conservato lo stesso nome.

Così tra le opere più recenti si contano, oltre ai nuovi ponti sulle tangenziali, le ricostruzioni di alcuni ponti urbani, demoliti o crollati.

133

Figure 5. A bridge on the Po river in Turin, Italy, from (Sassi Perino e Faraggiana, 1995)

6. Other problems

It goes by itself that the spread-wide rendering must start on an even numbered page; we cannot float two consecutive full page floating figures without controlling the parity of the page number of the first float.

It would be possible to delay the output of such figures by using the afterpage package by means of its `\afterpage` command (Carlisle, 2014); the parity controls might be done with the command replacement text, and if the next page number is not even, the whole material to be delayed may be postponed again by means of another nested `\afterpage` command. David Carlisle, the author of afterpage, suggested to one of us[2] to proceed this way, but he warned that `\afterpage` is pretty delicate because it modifies the `\shipout` kernel macro; this is the macro that ships out and appends a complete page to the output file; but, if the next page contains a section header, afterpage may mess up things to the point of loosing its contents; it actually happened with nested `\afterpage` commands when we first approached the problems addressed in this paper.

In general, if it is necessary to use `\afterpage`, it is possible to move around the macro with its contents, by advancing or retracting its occurrence in the source file; in our case we thought it might be better to look for a different solution.

We believe that the author is the best judge to find a suitable position for these spread-wide figures. If close to the point where the user would like to output such figure there is a paragraph end, so that after such a paragraph the typesetting program reverts to vertical mode, the macro that produces the spread wide figure can be used freely; the macro starts with a `\clearpage` so that all queued floating objects are output, and a new page is ready to accept new material; but if the page is odd numbered, a blank page is output, and the user might be forced to move the macro somewhere else.

Unfortunately, in spite of using floating objects, the user clever intervention is still necessary.

7. Other approaches

We have already explained that there are other possible solutions to correctly handle large images.

The most simple one is to float the large image by scaling it to a large fraction of the text block while maintaining its aspect ratio into a float-only page; certainly this is the simplest solution; it requires just the standard command set:

```
\begin{figure}[p]
\centering
\includegraphics[width=\linewidth,height=⟨fraction⟩\textheight,%
    keepaspectratio]{⟨file name⟩}
\caption[⟨lof entry⟩]{⟨caption⟩}\label{⟨label⟩}
\end{figure}
```

[2] Private communication.

The advantage is that the image and its caption float where LaTeX finds enough place to output the full page; of course ⟨*fraction*⟩ is a fractional number smaller than one so as to leave some space on the page for the caption and its space above to separate the caption from the image; this might be a good solution for figures that originally are moderately taller than wide. The disadvantage is that a really large image should be scaled to fit the page, and some details might become difficult to distinguish.

Another solution for images that are wider than tall would be to rotate them together with their caption so as to take advantage from the fact that the text block is generally taller than wide. It would be necessary to have available the lscape package (Carlisle, 2000) that allows a simple code such as this one.

```
\begin{figure}[p]
\begin{landscape}
\centering
\includegraphics[width=\linewidth,height=⟨fraction⟩\textheight,%
    keepaspectratio]{⟨file name⟩}
\captionoarglof entry{⟨caption⟩}\label{⟨label⟩}
\end{landscape}
\end{figure}
```

Again, ⟨*fraction*⟩ is a fractional number smaller than one, in order to reduce the scaled original image height so as to leave space for the caption and its label. Actually for these two solutions we preferred to produce the same effects by redefining what is necessary while using the same user command just with a different option. See below in section 10 the code description.

The next two possible solutions are a little trickier and require external packages already input by the swfigure one.

Thin and tall figures would produce a waste of space if inserted with the above methods; is would be more suitable to insert them as wrapped figures; therefore package wrapfig is required. It must be emphasised that wrapfig handles very well the position of a wrapped figure if it is surrounded by normal text, but various limitations exist if the figure falls next to any list or any text that is typeset in a special way. Moreover it does wrong calculations if the surrounding text contains a section header.

It is very difficult the change the page design so as to locally modify the text measure in order to fit a figure wrapped by *any* kind of text. Of course, almost nothing is impossible with LaTeX, but we did not try any other solution different from using the wrapfig package, since we wanted to maintain a single command to handle all such large figures. In section 10 the single user command is described with all its advantages and disadvantages.

Another situation deals with large width and small height figures. These figures require again a spread-wide solution, but done in such a way that the lower part of a page is filled up with the document text. Of course the left and right half images, containing also the caption below the right half one, should have the same dimensions otherwise the first text lines of the facing pages are not aligned.

Eventually there is the situation when a portrait image with an aspect ratio close to $\sqrt{2}$ or a landscape image with an aspect ratio close to $\sqrt{0.5}$ has to be inserted and occupies the full spread without leaving any margin around; in this case the caption must overwrite a small lateral portion of the image; it would not be unusual if the caption had to be typeset in a contrasting color so as to distinguish it from the average background figure color.

8. Examples

A couple of examples with the spread-wide representation are shown on page 11 where we put the screen shots of two spreads. The first one is scaled according to its height, while the second is scaled according to its width; notice that this is done by the software, because it is not the original image that has to be scaled, but each of the two half images that go into different (facing) pages.

In effects, notice that the aspect ratio of the two images is different; therefore the first image is dominated by its height for the proper scaling, while the second image is dominated by its width. This is visible by examining the space between the page headings and the image, and the space between the image right side and the caption on its right. The caption "bottom" is always aligned with the image bottom side. The folios in the respective page foots remain untouched by the image sizes and/or aspect ratios.

They are created with these simple commands:

```
\DFimage{OrionValley}{The Orion Valley}{dfig:ValleDiOrione}

\DFimage{TriremeGreca}[Roman trireme]{A Roman trireme where the beak
   is evidenced together with the three oar opening rows}%
   {dfig:trireme-greca}
```

The command whole syntax is the following:

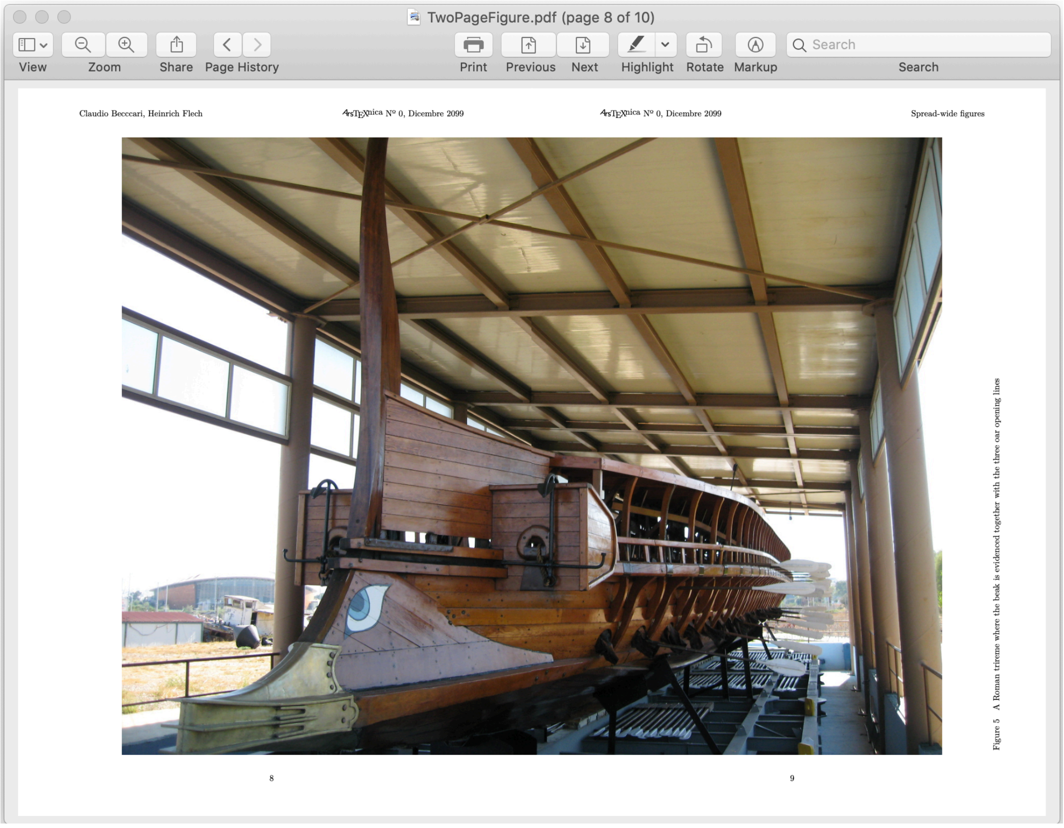
```
\DFimage[⟨option⟩]{⟨file name⟩}[⟨short caption⟩]{⟨caption text⟩}[⟨label⟩]%
     (⟨fraction⟩)<⟨lines⟩>|⟨width test⟩|!⟨precaption⟩!
```

The argument ⟨*fraction*⟩ is optional and is delimited by round parentheses; it is not used by all ⟨*options*⟩; more details in section 10. The angle bracket delimited optional arguments ⟨*lines*⟩ and vertical bar delimited ⟨*width test*⟩ are used only when ⟨*option*⟩ is VS; again see 10. The exclamation point delimited ⟨*precaption*⟩ may contains any declaration that is going to be used to partially format the caption and is optionally used only with the full spread display mode.

Obviously the ⟨*file name*⟩ refers to the file that contains the original image; for spread-wide, horizontal slim, and full spread display-modes the command splits it in two exact halves to be set on each spread page. The ⟨*caption text*⟩ is the full caption and it is recommended to keep it not too long; in any case the optional⟨*short caption*⟩ is useful for the list of figures; the ⟨*label*⟩ is optional; it is just the argument to the `\label` macro; of course, if this argument is not specified the figure cannot be referenced. The ⟨*precaption*⟩ may be used only with the full spread display mode and may be used, for example, to select a caption color contrasting with the image background.

It is worth noting that on an A4 sized page, typeset in one column mode, inserting images in spread-wide fashion appears as an exaggerated effect. But, if these images are printed in a well bound document with sewn signatures, they make a beautiful scene.

If this document, or any other document containing these images, is read on the screen, and the PDF viewer allows to display a spread at a time, the result is even more impressive. If the PDF viewer allows to display the spread without any gap between the spread pages, it is just perfect. In facts, in figure 6 there are the screen shots taken on a Mac computer by using its Preview PDF viewer: it contains the spread pages of this very paper draft; we thought it would

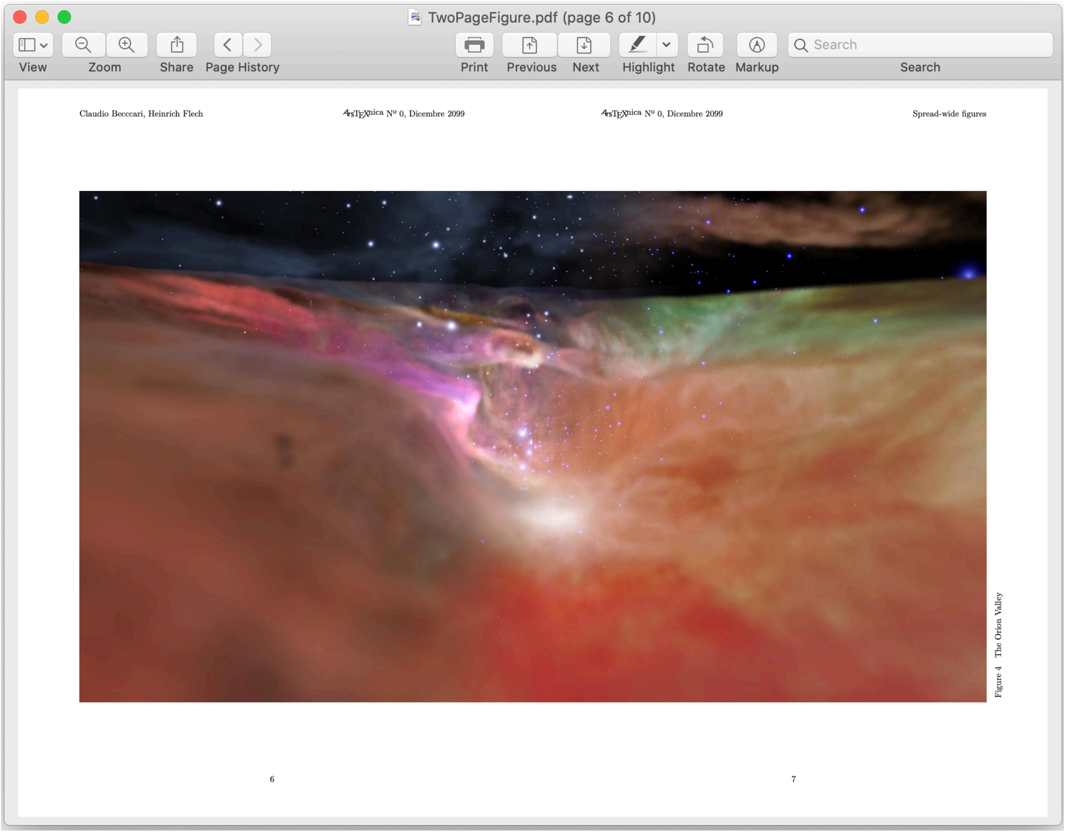


TwoPageFigure.pdf (page 8 of 10)

Claudio Beccari, Heinrich Fleck — ArsTEXnica Nº 0, Dicembre 2099 — ArsTEXnica Nº 0, Dicembre 2099 — Spread-wide figures

Figure 5 A Roman trireme where the beak is evidenced together with the three oar opening lines

8 — 9



TwoPageFigure.pdf (page 6 of 10)

Claudio Beccari, Heinrich Fleck — ArsTEXnica Nº 0, Dicembre 2099 — ArsTEXnica Nº 0, Dicembre 2099 — Spread-wide figures

Figure 4 The Orion Valley

6 — 7

Figure 6. The screen shots of the spreads containing the same images shown in figures 2 and 1

be more sober to display the screen shots rather than the real spreads. These screenshots and the following ones display a full Mac pane obtained with its Preview viewer, because is the only one we found that does not mark the internal page boundary with a gap or a black line.

We have experienced with several PDF viewers on the three common platforms, Mac, Linux, Windows; certainly the Preview.app on Mac can display gapless spreads. The Okular viewer on Linux displays a gapless spread with a very thin black line dividing the halves of the spread. The multi platform viewers Adobe Acrobat Reader DC as well as Adobe Acrobat Pro and PDF Studio Pro display two pages at a time with a very small gap, but start displaying pages from page 1, therefore the even numbered pages fall on the right and the spread-width images get split in two different views. PDF Studio Pro has a particular view setting *Cover Continuous* to display the cover in a single page and then all the following spreads have the even page on the left. Adobe Reader performs the same as PDF Studio Pro, provided that the document source file uses the hyperref package, and *pdfpagelayout=TwoColumnRight* is among its options.

Skim displays spreads correctly[3], but with a small gap between the facing pages. The internal viewers of TeXShop and TeXworks can display spreads in the correct way, but with a small gap between pages. TeX Pro Reader Lite correctly displays spreads wth a gap in between. WPS and Foxit Reader behave just as Adobe Acrobat Reader DC; TeXstudio does not display any spread. On Windows SumatraPDF correctly displays spreads with a little gap between the pages. Xpdf on all platforms displays page spreads as the Adobe programs do.

In spite of the large number of PDF viewers examined by the authors, not all of them behave the same way and, in particular, at least one of them does not display any spread.

The normal result performed by the LaTeX kernel commands, is shown in figures 1 and 2. So we do not insist on this result.

The rotated image and the slim vertical image are shown in the screen shots of figures 7 and 8.

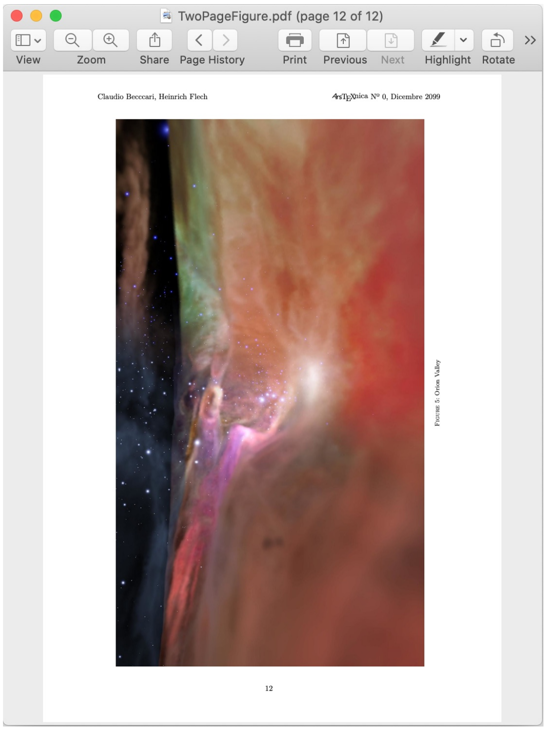
An example of a slim and spread-wide image is shown by means of the screen shot of figure 9. The caption falls under the right side page, so that `\pageref` points to the second page; this feature implies a suitable wording in order to refer to the figure; in any case if the user reads the document by viewing a spread at a time, and the figure numbers are hyperlinked, the screen window displays correctly the whole spread. Notice that the text in the facing pages shown in figure 9 are perfectly aligned at their top.

Figures 11 and 12 display a Total Height display mode, and a Total Width display mode. Both are taken from odd numbered pages, but on even numbered ones the display is the specular one of that shown in these figures. The total width, depending on the figure aspect ratio leaves some space for normal text; while the Total Height one occupies the whole page.

Last but not least, figure 10 displays a screenshot of an A3 sized fake figure, rotated counterclockwise by 90°; it covers the full spread; the figure has its own margins, therefore the overwritten caption falls on its white space at the bottom; there is no need to color the caption with any contrasting color.

We do not go further by displaying other screenshots that prove the functionalities of the main macro we are talking about; we just summarise the eight display modes with the various schemes shown in figure 13; notice that each white framed rectangle shows the full

[3] "Correctly", here and in the following text, means with the even numbered page on the left without any specific set-up as it is necessary with the Adobe viewers and PDF Studio.



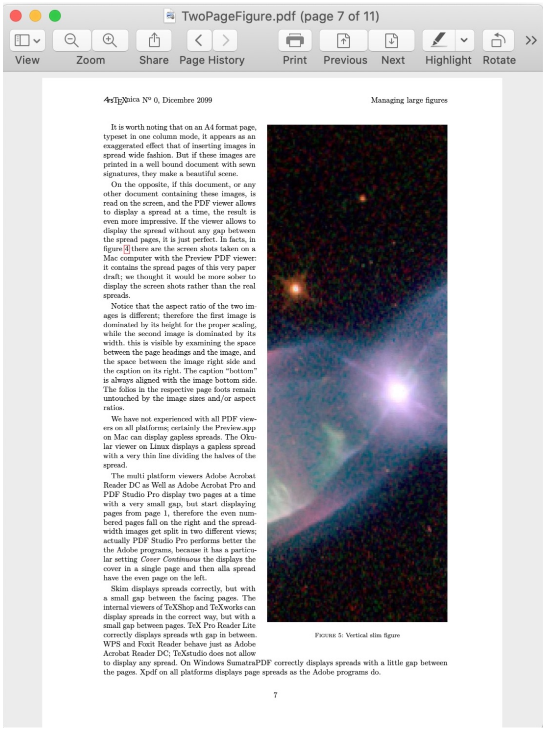
Claudio Beccari, Heinrich Flech

ArsTeXnica Nº 0, Dicembre 2099

Figure 5: Orion Valley

12

Figure 7. Rotated figure



ArsTeXnica Nº 0, Dicembre 2099

Managing large figures

It is worth noting that on an A4 format page, typeset in one column mode, it appears as an exaggerated effect that of inserting images in spread wide fashion. But if these images are printed in a well bound document with sewn signatures, they make a beautiful scene.

On the opposite, if this document, or any other document containing these images, is read on the screen, and the PDF viewer allows to display a spread at a time, the result is even more impressive. If the viewer allows to display the spread without any gap between the spread pages, it is just perfect. In facts, in figure 4 there are the screen shots taken on a Mac computer with the Preview PDF viewer: it contains the spread pages of this very paper draft; we thought it would be more sober to display the screen shots rather than the real spreads.

Notice that the aspect ratio of the two images is different; therefore the first image is dominated by its height for the proper scaling, while the second image is dominated by its width. this is visible by examining the space between the page headings and the image, and the space between the image right side and the caption on its right. The caption "bottom" is always aligned with the image bottom side. The folios in the respective page foots remain untouched by the image sizes and/or aspect ratios.

We have not experienced with all PDF viewers on all platforms; certainly the Preview.app on Mac can display gapless spreads. The Okular viewer on Linux displays a gapless spread with a very thin line dividing the halves of the spread.

The multi platform viewers Adobe Acrobat Reader DC as Well as Adobe Acrobat Pro and PDF Studio Pro display two pages at a time with a very small gap, but start displaying pages from page 1, therefore the even numbered pages fall on the right and the spread-width images get split in two different views; actually PDF Studio Pro performs better the the Adobe programs, because it has a particular setting *Cover Continuous* the displays the cover in a single page and then alla spread have the even page on the left.

Skim displays spreads correctly, but with a small gap between the facing pages. The internal viewers of TeXShop and TeXworks can display spreads in the correct way, but with a small gap between pages. TeX Pro Reader Lite correctly displays spreads wth gap in between. WPS and Foxit Reader behave just as Adobe Acrobat Reader DC; TeXstudio does not allow to display any spread. On Windows SumatraPDF correctly displays spreads with a little gap between the pages. Xpdf on all platforms displays page spreads as the Adobe programs do.

Figure 5: Vertical slim figure

7

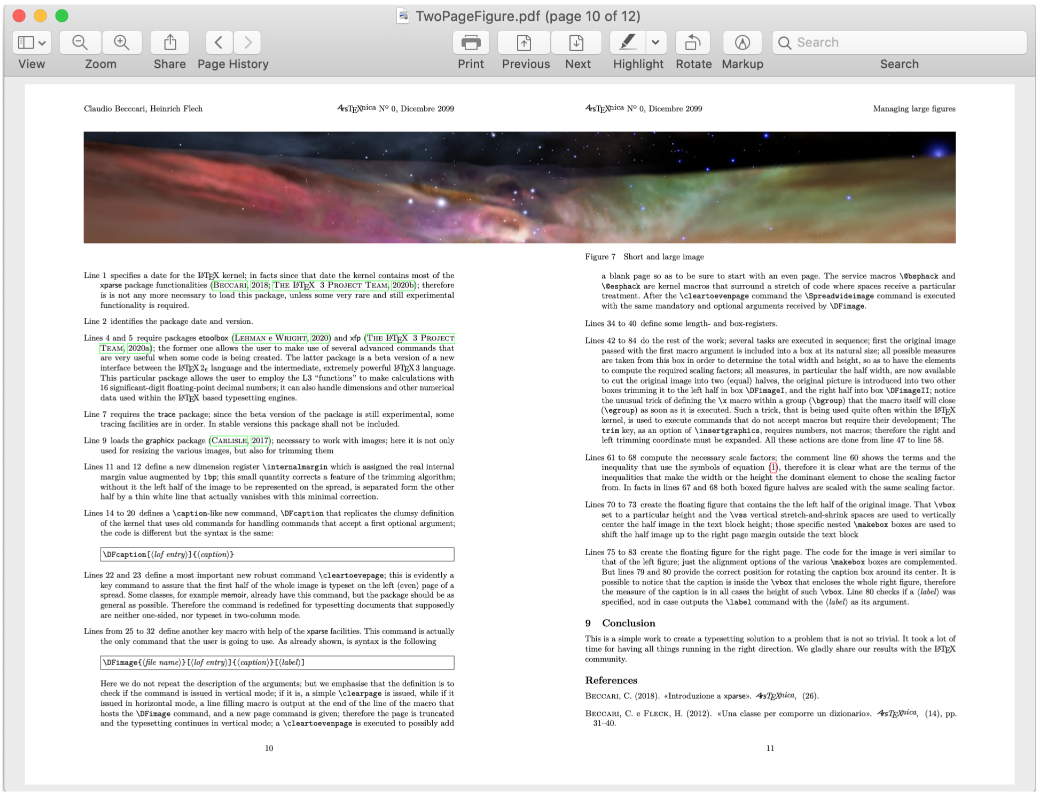
Figure 8. Slim vertical figure

trimmed two pages spread; internally, the white framed rectangles show the text blocks; the grey rectangles display the areas occupied by the images; the red rectangles display the areas occupied by the captions. If the red area is a slim vertical ribbon, the caption is typeset rotated 90° anticlockwise.

9. Command or environment?

As any LaTeX user knows, any command name may be used as an environment name. `\DFimage` is not an exception, but it is a little different from other such environments; in effects,it is prohibited to nest such environments so that it is up to the user to avoid nesting such DFimage environments and/or commands; the package does not provide any test to verify this situation. Moreover, when the command/environment is inserted within a paragraph, it is important that, when its processing is completed, the typesetting mode is restored.

This implies a choice from the user who should ask himself: "When the command is the best choice, and when the environment is more convenient?". We would suggest that the environment is preferable when the large figure appears on the page(s) together with some text; this implies the Vertical Slim and the Horizontal Slim display modes. Certainly it is not convenient with the Normal Figure and the Rotated Figure modes that produce fully floating objects; it is usable with the Spread Wide mode, because it can be used within a paragraph, in order to choose where to start a new page; but it is not really necessary; the Full Spread display mode should be used preferably only at the end of a chapter or at the end of a document, when the command is the best choice.



Claudio Becccari, Heinrich Flech   ArsTeXnica Nº 0, Dicembre 2099   ArsTeXnica Nº 0, Dicembre 2099   Managing large figures

Figure 7   Short and large image

Line 1 specifies a date for the LaTeX kernel; in facts since that date the kernel contains most of the xparse package functionalities (Beccari, 2018; The LaTeX 3 Project Team, 2020b); therefore is is not any more necessary to load this package, unless some very rare and still experimental functionality is required.

Line 2 identifies the package date and version.

Lines 4 and 5 require packages etoolbox (Lehman e Wright, 2020) and xfp (The LaTeX 3 Project Team, 2020a); the former one allows the user to make use of several advanced commands that are very useful when some code is being created. The latter package is a beta version of a new interface between the LaTeX 2$_\epsilon$ language and the intermediate, extremely powerful LaTeX 3 language. This particular package allows the user to employ the L3 "functions" to make calculations with 16 significant-digit floating-point decimal numbers; it can also handle dimensions and other numerical data used within the LaTeX based typesetting engines.

Line 7 requires the trace package; since the beta version of the package is still experimental, some tracing facilities are in order. In stable versions this package shall not be included.

Line 9 loads the graphicx package (Carlisle, 2017); necessary to work with images; here it is not only used for resizing the various images, but also for trimming them

Lines 11 and 12 define a new dimension register `\internalmargin` which is assigned the real internal margin value augmented by `1bp`; this small quantity corrects a feature of the trimming algorithm; without it the left half of the image to be represented on the spread, is separated form the other half by a thin white line that actually vanishes with this minimal correction.

Lines 14 to 20 defines a `\caption`-like new command, `\DFcaption` that replicates the clumsy definition of the kernel that uses old commands for handling commands that accept a first optional argument; the code is different but the syntax is the same:

```
\DFcaption[⟨lof entry⟩]{⟨caption⟩}
```

Lines 22 and 23 define a most important new robust command `\cleartoevepage`; this is evidently a key command to assure that the first half of the whole image is typeset on the left (even) page of a spread. Some classes, for example memoir, already have this command, but the package should be as general as possible. Therefore the command is redefined for typesetting documents that supposedly are neither one-sided, nor typeset in two-column mode.

Lines from 25 to 32 define another key macro with help of the xparse facilities. This command is actually the only command that the user is going to use. As already shown, is syntax is the following

```
\DFimage{⟨file name⟩}[⟨lof entry⟩]{⟨caption⟩}[⟨label⟩]
```

Here we do not repeat the description of the arguments; but we emphasise that the definition is to check if the command is issued in vertical mode; if it is, a simple `\clearpage` is issued, while if it issued in horizontal mode, a line filling macro is output at the end of the line of the macro that hosts the `\DFimage` command, and a new page command is given; therefore the page is truncated and the typesetting continues in vertical mode; a `\cleartoevenpage` is executed to possibly add a blank page so as to be sure to start with an even page. The service macros `\@bsphack` and `\@esphack` are kernel macros that surround a stretch of code where spaces receive a particular treatment. After the `\cleartoevenpage` command the `\Spreadwideimage` command is executed with the same mandatory and optional arguments received by `\DFimage`.

Lines 34 to 40 define some length- and box-registers.

Lines 42 to 84 do the rest of the work; several tasks are executed in sequence; first the original image passed with the first macro argument is included into a box at its natural size; all possible measures are taken from this box in order to determine the total width and height, so as to have the elements to compute the required scaling factors; all measures, in particular the half width, are now available to cut the original image into two (equal) halves, the original picture is introduced into two other boxes trimming it to the left half in box `\DFimageI`, and the right half into box `\DFimageII`; notice the unusual trick of defining the `\x` macro within a group (`\bgroup`) that the macro itself will close (`\egroup`) as soon as it is executed. Such a trick, that is being used quite often within the LaTeX kernel, is used to execute commands that do not accept macros but require their development; The `trim` key, as an option of `\insertgraphics`, requires numbers, not macros; therefore the right and left trimming coordinate must be expanded. All these actions are done from line 47 to line 58.

Lines 61 to 68 compute the necessary scale factors; the comment line 60 shows the terms and the inequality that use the symbols of equation (1), therefore it is clear what are the terms of the inequalities that make the width or the height the dominant element to chose the scaling factor from. In facts in lines 67 and 68 both boxed figure halves are scaled with the same scaling factor.

Lines 70 to 73 create the floating figure that contains the the left half of the original image. That `\vbox` set to a particular height and the `\vss` vertical stretch-and-shrink spaces are used to vertically center the half image in the text block height; those specific nested `\makebox` boxes are used to shift the half image up to the right page margin outside the text block

Lines 75 to 83 create the floating figure for the right page. The code for the image is veri similar to that of the left figure; just the alignment options of the various `\makebox` boxes are complemented. But lines 79 and 80 provide the correct position for rotating the caption box around its center. It is possible to notice that the caption is inside the `\vbox` that encloses the whole right figure, therefore the measure of the caption is in all cases the height of such `\vbox`. Line 80 checks if a ⟨*label*⟩ was specified, and in case outputs the `\label` command with the ⟨*label*⟩ as its argument.

9 Conclusion

This is a simple work to create a typesetting solution to a problem that is not so trivial. It took a lot of time for having all things running in the right direction. We gladly share our results with the LaTeX community.

References

Beccari, C. (2018). «Introduzione a xparse». *ArsTeXnica*, (26).

Beccari, C. e Fleck, H. (2012). «Una classe per comporre un dizionario». *ArsTeXnica*, (14), pp. 31–40.

10   11

Figure 9. A short and large spread-with image

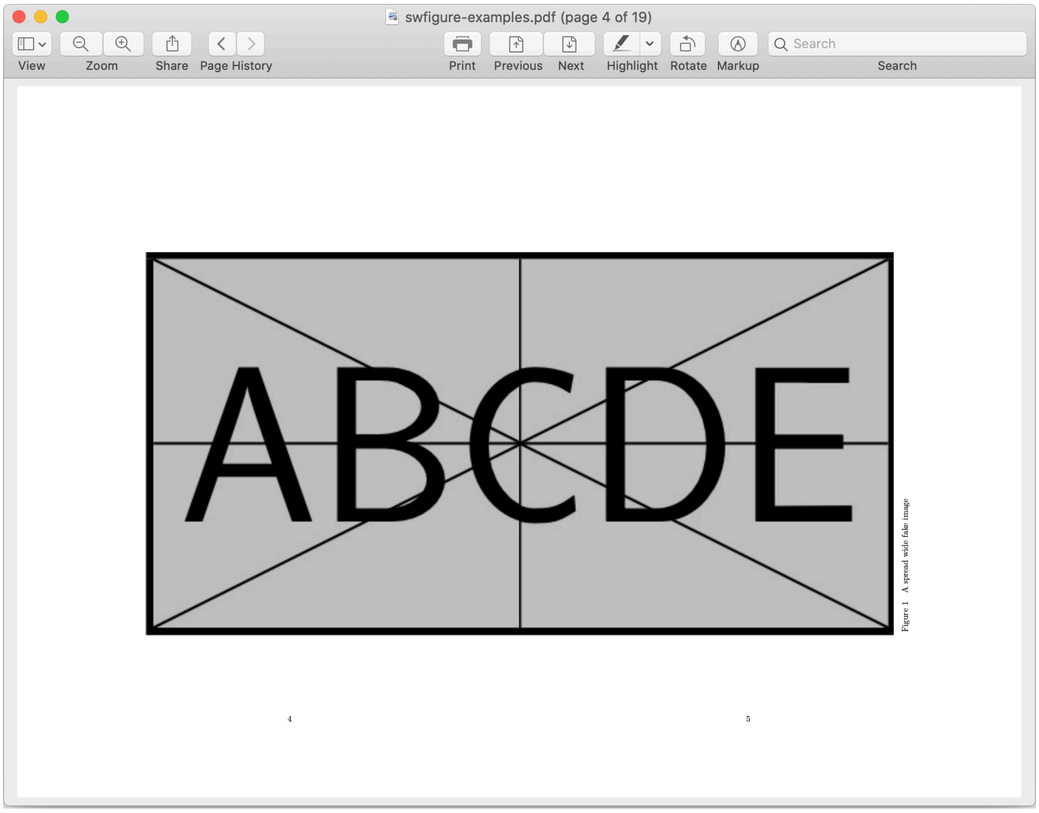


Figure 1   A spread wide fake image

4   5

Figure 10. An A3 sized fake image

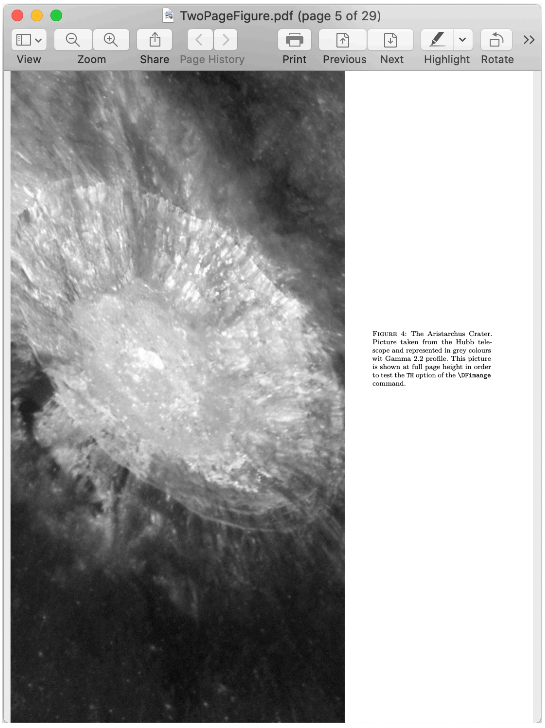


FIGURE 4: The Aristarchus Crater. Picture taken from the Hubb telescope and represented in grey colours wit Gamma 2.2 profile. This picture is shown at full page height in order to test the `TH` option of the `\DFimange` command.

Figure 11. Total Height figure

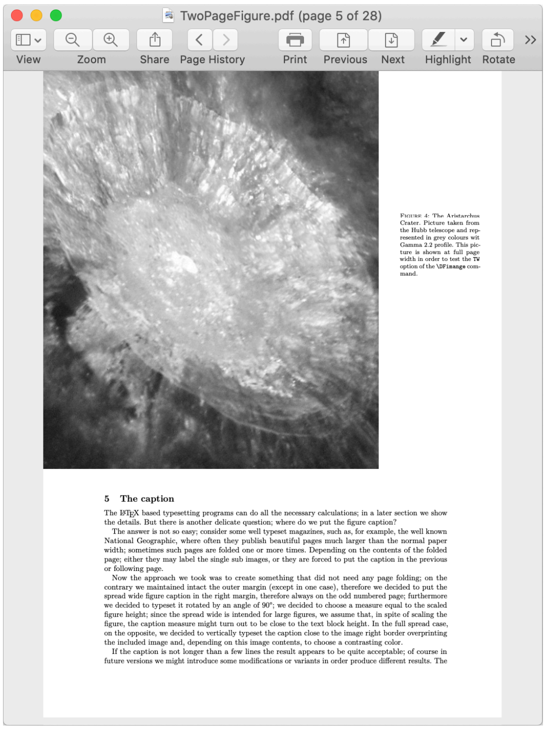


FIGURE 4: The Aristarchus Crater. Picture taken from the Hubb telescope and represented in grey colours wit Gamma 2.2 profile. This picture is shown at full page width in order to test the `TW` option of the `\DFimange` command.

5 The caption

The LaTeX based typesetting programs can do all the necessary calculations; in a later section we show the details. But there is another delicate question; where do we put the figure caption?

The answer is not so easy; consider some well typeset magazines, such as, for example, the well known National Geographic, where often they publish beautiful pages much larger than the normal paper width; sometimes such pages are folded one or more times. Depending on the contents of the folded page; either they may label the single sub images, or they are forced to put the caption in the previous or following page.

Now the approach we took was to create something that did not need any page folding; on the contrary we maintained intact the outer margin (except in one case), therefore we decided to put the spread wide figure caption in the right margin, therefore always on the odd numbered page; furthermore we decided to typeset it rotated by an angle of 90°; we decided to choose a measure equal to the scaled figure height; since the spread wide is intended for large figures, we assume that, in spite of scaling the figure, the caption measure might turn out to be close to the text block height. In the full spread case, on the opposite, we decided to vertically typeset the caption close to the image right border overprinting the included image and, depending on this image contents, to choose a contrasting color.

If the caption is not longer than a few lines the result appears to be quite acceptable; of course in future versions we might introduce some modifications or variants in order produce different results. The

Figure 12. Total Width figure

10. The software

The package swfigure.sty is already available in any updated and complete installation of the TeX system which has available both the code documentation `swfigure.pdf` and a short user manual `swfigure-examples.pdf`; both are readable with the command line `texdoc` followed by either file name. Therefore we redirect the reader to those two documents. Here we describe the peculiar main environment, just to make an example of the type of code we used by employing the xparse facilities and the special string-list analysis LaTeX 3 functions, useful to code the various display mode selections.

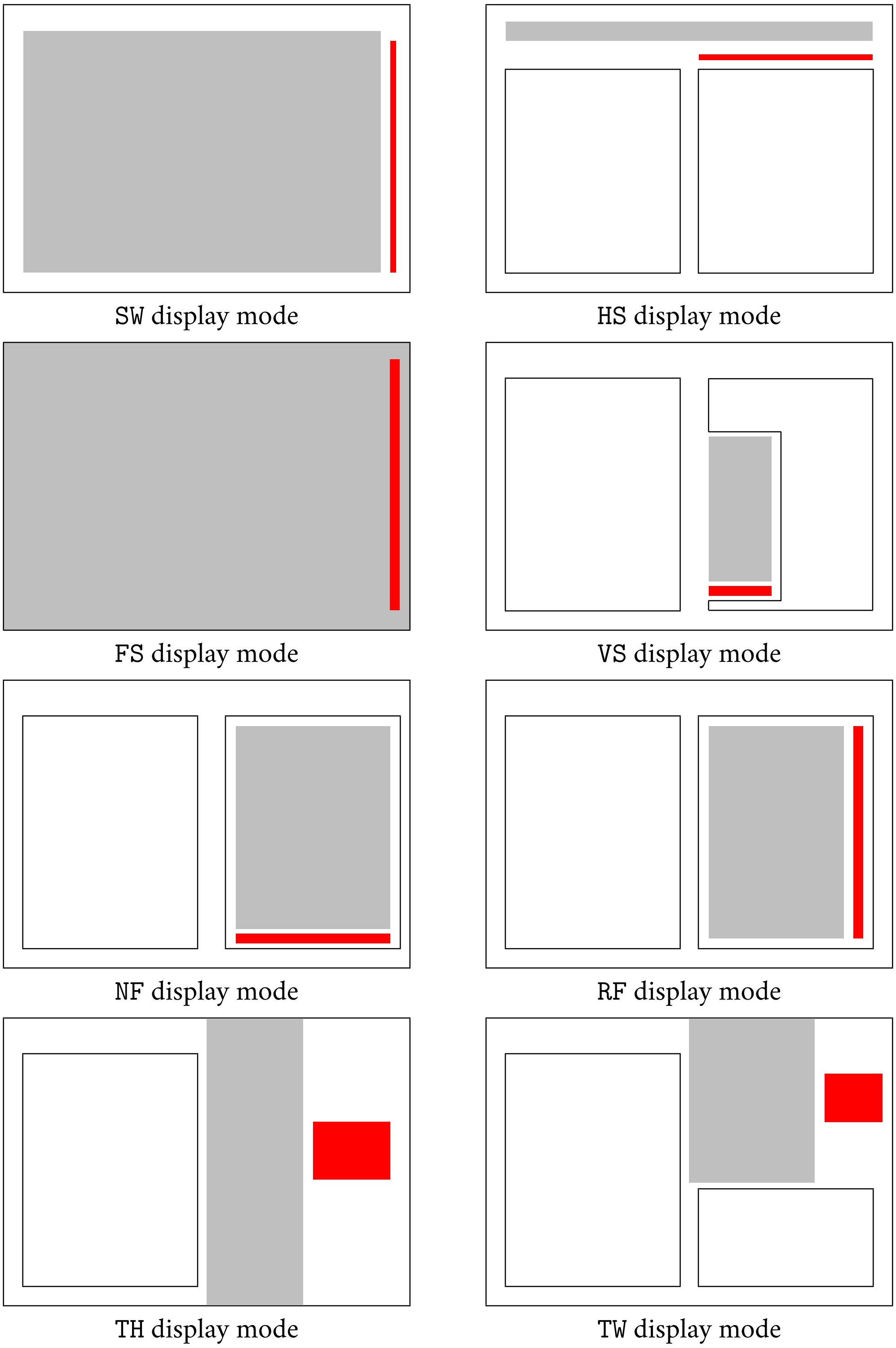
In facts the single `\DFimage` command or DFimage environment receive some mandatory and optional arguments that allow to typeset the eight varieties of each large image, as described in the previous sections. The default display mode is SW, the one for which this code was initially created; it refers to a spread-wide figure on two facing pages that do not contain any document text. By means of the argument ⟨*display mode*⟩ it is possible to create the other display modes:

```
\DFimage[⟨display mode⟩]{⟨image file name⟩}[⟨lof entry⟩]{⟨caption⟩}[⟨label⟩]%
     (⟨fraction⟩)<⟨lines⟩>|⟨width test⟩|!⟨precaption⟩!
```

The various arguments have the following meanings.

⟨*display mode*⟩ This optional argument accepts eight values.

`SW` (Spread-Wide) This is the default display mode to produce a spread that contains only one large image; the caption is vertically set in the right margin; the scaling



SW display mode

HS display mode

FS display mode

VS display mode

NF display mode

RF display mode

TH display mode

TW display mode

Figure 13. Schematic diagrams of the spreads containing the eight display modes: the grey area is the space occupied by the actual figure, while the red one is the space occupied by the caption.

of the half images is computed in base of the full image aspect ratio in such a way that it exceeds neither the text block height nor the two adjacent spread text block widths including the internal margins.

`NF` (Normal Figure) This is the document-class mode to display an image, with the caption underneath. This is apparently useless, but it is handy to have a single command with several modes; if the user does not like a certain mode he can change it by simply changing its mode option. It is similar to the normal behaviour; actually it performs a little differently but the real feature is that it is provided by the same `\DFimage` command used to provide the other display modes; therefore changing mode implies the simple substitution of NF with another display mode option.

`RF` (Rotated Figure) In this mode the figure is rotated counterclockwise by 90° on a page that contains only the figure with its caption; the ⟨*fraction*⟩ argument may be used to scale the original image height so that it does not fill up the whole page width and some space is used for the caption; depending on the caption length the user might choose another larger or smaller value than the default one (preset to 0.8). See below.

`VS` (Vertical Strip) When the image is a high and thin strip, it is inserted as a narrow figure wrapped by the surrounding text. The vertical scaling ⟨*fraction*⟩ lets the user control the total height of the image with its caption underneath. The ⟨*lines*⟩ integer number can be used in order to correct the actual number of indented lines that the `\begin{wrapfigure}` computes for the figure vertical space. If the aspect ratio of the image to be displayed with this display mode is smaller than 2, the image is not displayed, and in its place a notice is printed in the document that explains why, and suggests the user to select a different display mode. For this mode it is convenient to use the environment in order to have the `\end` command after a suitable number of full paragraphs, so as to assure the correct indentations without bothering about other floating bodies. But the limitation of the aspect ratio not smaller than 2 may be ignored if the user directly employs the wrapfigure environment and lets the figure protrude in the outer margin so as to avoid a too short measure for the indented wrapping text.

`HS` (Horizontal Strip) When the original image is very large but relatively thin, then a spread is created with both half images well aligned at the top of the spread facing pages; the text underneath in both pages is also correctly aligned.

`FS` (Full Spread) This mode is suited to import single-page A3-sized documents without any scaling to fit the spread; it fills up the entire page with no margins around; it is an alternative to attach a folded A3 sheet to a well bound printed document; or to link an external PDF file from inside an A4-sized document. It does not require a special class, such as memoir, that can locally change the virtual paper size; neither it requires a special printer capable of changing paper size on demand of the document to be printed. The result with this display mode might be fundamental if the A3 sized PDF file contained a complicate drawing or a very large tabular material that might become unreadable if shrunk to A4 size.

`TH` (Total Height) This mode is suited to display a tall not so slim image that extends from the top page border to the bottom one; if the aspect ratio is just greater than unity, but is not large enough, a warning message is sent to the `.log` file , but compilation continues; the caption is typeset in a box in the space available close to the external paper border. The image and the caption boxes are aligned according to their median line.

`TW` (Total Width) This mode is similar to the preceding one, but the image is scaled to a width that equals the sum of the internal margin width and 80% of the text one. The white space between the image and the caption is used for the caption and the objects are aligned as specified above for the full height image display mode. Under these boxes alignment there is place for some text, therefore the page cannot be flushed to the output by means of a `\clearpage` command.

If a ⟨*display mode*⟩ is misspelt, a warning appears in the log file together with the misspelt string and the image file name which it applied to.

⟨*image file name*⟩ The ⟨*image file name*⟩ can be written without specifying the extension; we recall the the allowed extensions are just `.pdf`, `.eps`, `.jpg`, `.png`, `.mps`, the last one of this list belonging to images created with METAPOST. Other graphic files with other extensions may be used, but they require special settings; see the documentation Carlisle (2017).

⟨*lof entry*⟩ The ⟨*lof entry*⟩ is the short caption used to enter a shortened caption text into the list of figures. It is obviously optional, as it is for the standard `\caption` command; its use is recommended when the full caption is longer than one line.

⟨*caption*⟩ The `\caption` argument is the text of the figure caption; we recommend to maintain it short; in most cases two or three lines should be sufficient; with the Vertical Strip mode the caption is typeset in a narrow text block with a small measure, so that the caption text should be carefully handled and possibly hyphenated by the user. With this mode the ⟨*precaption*⟩ optional argument may be used if ragged right typesetting is desired.

⟨*label*⟩ The ⟨*label*⟩ is optional; if it is used, it amounts to whatever the user would write as the `\label` argument; obviously without this argument is is impossible to make reference to the (numbered) figure with the usual mechanism of `\label`, `\ref`, `\pageref` and similar commands provided by other packages.

⟨*fraction*⟩ This optional argument should be delimited by normal parentheses `( )`; it is a fractional number not greater than one; as specified in some previous description items, it may be used with some display modes, and neglected in other ones; its default value is 0.8; the user can specify a different value in order to fine tune the height of vertical objects: rotated images or vertical strips. For the Full With display mode this parameter may be used to fine tune the lateral caption width; in this case it can never exceed unity and we suggest to use values close to the default value 0.8. This parameter can be used also to fine tune the caption with in the `TW` mode. Its default value in general is adequate, but the user may choose values that are not too different from the default value, provided it does not exceed unity. According with the experience we accumulated with many tests, the useful range is between 0.7 and 0.9.

⟨*lines*⟩ In the Vertical Strip display mode the whole work is done by package wrapfig by means of its environment wrapfigure. This environment opening command accepts several arguments, one of which specifies the total amount of lines to be indented to make space for the wrapped figure. Sometimes the environment computes a smaller or larger value than necessary; it depends from many factors, but the most important of which, probably, is that the native command `\hangindent` handles lines, not vertical spaces, therefore the contents of the wrapping text may cause such “errors”. In order to correct this error, the ⟨*lines*⟩ (angle bracket `< >` delimited) optional argument (preset to zero) may be specified in order to correct the computed number of lines by adding or subtracting an integer number so as to get the wrapped figure correctly spaced from the wrapping text. With the tests we made, we found situations where a correction of -1 lines was necessary, and other ones where the correction amounted to 2 lines.

⟨*width test*⟩ is used only in the Vertical Strip display mode; it is the vertical-bar delimited optional argument (`| |`); its default value is 0.25; it is the fraction of the text block width under which the scaling of the wrapped image width should not go below; in facts, by using the ⟨*fraction*⟩ optional value, the image gets shorter than that ⟨*fraction*⟩ of the text block *height*; but by reducing its height, also its width gets smaller; if such width gets too small, the ⟨*width test*⟩ controls that it is not so small that the caption cannot be typeset in a decent way. The preset width test fraction 0.25 assures that the caption measure does not get smaller that one fourth of the text block width; the user can select a different value not greater than 0.5 but as low as zero. Evidently whatever ⟨*fraction*⟩ is selected, the scaled image width never becomes negative, therefore the zero value always produces a false test, equivalent to ‘no-test’; on the opposite, the larger this ⟨*width test*⟩ value, the most probable becomes the failure of the test, and therefore a notice is printed in place of the image.

⟨*precaption*⟩ is an optional argument delimited by exclamation marks (`! !`). With the Full Spread display mode there is no place for captions, because the whole space of the spread pages is occupied by the imported material; if this material has its own margins, the vertical caption may not need any special treatment, as in figure 10. But if the imported material does not have a white margin, the caption overwrites a small part of the imported “image”, and might become unreadable if typeset with the default black color; this option allows the user to choose a different contrasting color so as to let the caption text emerge form the background; the color should be specified by means of the full declaration `\color{`⟨*color*⟩`}`. Besides the color the user can specify a displacement, a different font, a justification command, and any other declaration that makes sense with a caption.

The actual short code for the environment DFimage is the following:

```
\NewDocumentEnvironment{DFimage}%
  { O{SW} m O{#4} m o D(){0.8} D<>{0} D||{0.25} D!!{} }%
{%
  \SetList{\DisplayModeList}{SW,HS,VS,FS,NF,RF,TH,TW}%
  \TestList{\DisplayModeList}{#1}%
  {\csuse{#1figure}{#2}[#3]{#4}[#5](#6)<#7>|#8|!#9!%
    \fptest{\CompStrings{\@currenv}{DFimage}}{}{\reset@tmode}%
  }%
  {%
    \DFwarning[#1]{#2}[#3]{#4}
  }%
}%
```

```
{%
  \aftergroup\reset@tmode
}%

\NewDocumentCommand\DFwarning{ o m o m }{%
\PackageWarning{swfigure}%
    {********************************************\MessageBreak
      Option #1\space is not valid. Nothing done \MessageBreak
                                                 \MessageBreak
      Image #2 was not processed                 \MessageBreak
     ********************************************\MessageBreak
    }%
}
```

Notice the LaTeX 3 environment signature with a total of nine arguments, two of which are mandatory and seven ad variously delimited optional argument with or without initial default values; all these preset values have been just described above. Note the third argument; it is the `lof` entry corresponding to the particular environment or command; its default value is the same of the fourth argument, the caption text; this greatly simplifies the handling of the optional `lof` entry implemented in the LaTeX kernel `\caption` command.

The `\SetList` command is a user interface to an internal LaTeX 3 function to create a list of strings; here the strings are the eight valid display mode codes.

The `\TestList` command is user interface to another LaTeX 3 function that tests if the specified display mode is in the list; if the test is true the macro obtained by glueing the text mode code to the word “figure” is called with all the macro arguments except the first one already used. Therefore all the mandatory and the optional arguments are passed to the actual macros that implement the particular display mode, even those arguments that shall not be used by the macro. On the opposite, if the specified display mode is not in the list, it was misspelt, and the warning macro `\DFwarning` is executed; this macro, as explained before, does not output any image, and in its place it writes the reason why: the display mode was wrong and the corresponding image has not been processed.

The environment closing commands just execute the “reset the text printing mode” after the possible environment end; actually the eight display mode specific macros do the same; therefore using the command or the environment is almost the same thing. It goes by itself that some switches are set by the specific macros, so that these final environment commands do not produce errors if the command was used in place of the environment or vice versa.

11. Conclusion

The code in the swfigure package contain various special artifices that were necessary to realise the goal of managing large images. This task was not a simple one; this code resulted sufficient to create typesetting solutions to a problem that is not so trivial. It took a lot of time and several code subversions to fine tune every detail so that all display modes produce the desired results. Some little “features” still remain in the code, but substantially this beta version might remain stable at user level. We gladly share our results with the LaTeX community and ask the possible users to address comments, suggestions, and bug notices to the first author of this paper; we did our best, but bugs may still be hiding somewhere in our code.

References

Beccari, Claudio (2020a). «The swfigure package — Managing large and spread wide figures». PDF document. Readable with command `texdoc swfigure`.

— (2020b). «The swfigure package — Usage examples». PDF document. Readable with command `texdoc swfigure-examples`.

Beccari, Claudio e Heinrich Fleck (2012). «Una classe per comporre un dizionario». *ArsTeXnica*, (14), pp. 31–40.

Carlisle, David (2000). «The lscape package». PDF document. Readable with command `texdoc lscape`.

— (2014). «The afterpage package». PDF document. Readable with command `texdoc afterpage`.

Carlisle, D.P. (2017). «Packages in the graphics bundle». PDF document. Readable with command `texdoc graphicx`.

De Marco, Agostino e Massimilano Dominici (2008). «longmedal: un pacchetto per medaglioni divisi su più pagine». *ArsTeXnica*, (6), pp. 86–92.

Gabrielli, Noemi (1974). *L'arte nell'antico marchesato di Saluzzo*. Istituto BAncario San Paolo, Torino.

Matthias, Andreas (2020). «The pdfpages package». PDF document. Readable with command `texdoc pdfpages`.

Sassi Perino, Angia e Giorgio Faraggiana (1995). *I trentasei ponti di Torino*. Edizioni del Capricorno, Torino.

Scharrer, Martin (2018). «The mwe package». PDF document. Readable with command `texdoc mwe`.

Claudio Beccari
claudio.beccari@gmail.com

Heinrich Fleck
heinrich.fleck@yahoo.it