

Package `canoniclayout`*

Claudio Beccari[†]

November 6, 2011

Abstract

This package is a small extension that allows to determine the page parameters for a canonical layout where the text block is placed onto the page in such a way as: (a) its diagonal coincides with the page relevant diagonal, and (b) the circle inscribed within the text block is also tangent to the page sides. The text block width to height ratio is kept equal to the page ratio, as well as the inner to outer and the upper to lower margin ratios.

The canonical layout determined by this package is laid down on this very page; of course the actual text block fits the text block grid, because this documentation is typeset by means of the `ltxdoc` document class, that by itself would use other layout parameters, but that is has been arrayed so as to invoke this very package. This allows you to immediately evaluate this traditional layout and see if it fits to your needs.

Contents	3 Usage	3
1 Introduction	4 Font normal size	4
2 The mathematics of the canonical layout	5 Conclusion	5
	6 Documented code	5

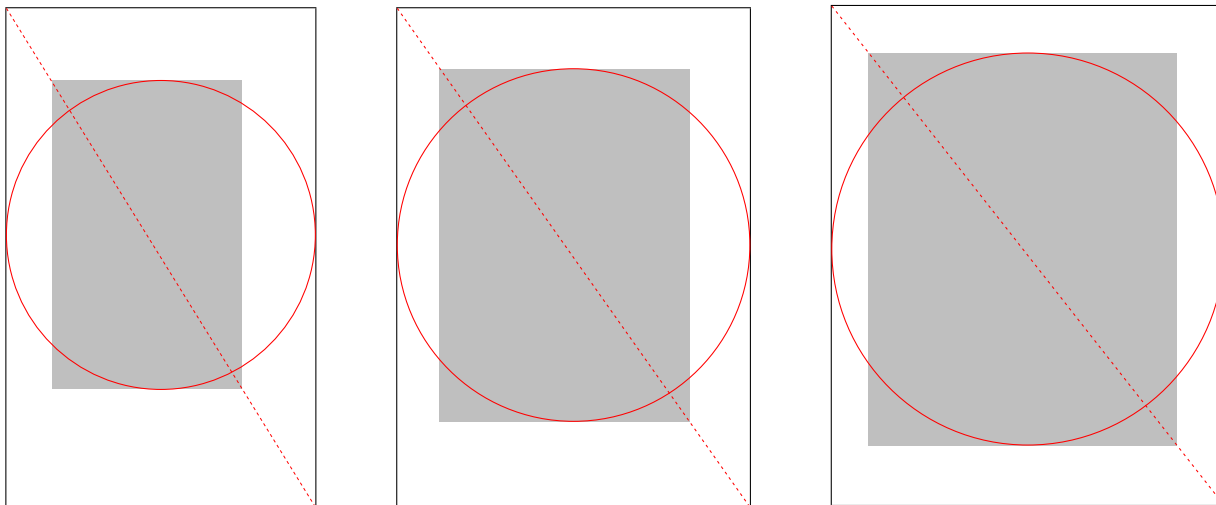
1 Introduction

As any user of the `memoir` class knows well, this class allows to customize the page layout virtually without any limitation; such limitations are connected mainly with the user inconsistent data that miss to specify necessary information or give unnecessary and contrasting information. Also other classes and/or packages allow the customized design of the page layout; some other classes, such as the standard ones, do not have any facility for customizing the page geometry, unless the user invokes suitable packages, such as `geometry` or `tyeparea`, just to male a couple of well known examples.

Many packages exist in order to specify a named layout; this package `canoniclayout` is another one that computes the page parameters to determine the horizontal and vertical dimensions of the margins and the text block. All other details shall be specified by the user, for example the dimensions of the stock paper, the trimming widths, and therefore the position of the page onto the stock paper, and so on. This package deals only with the trimmed page, not with the page and stock paper relationship.

*Version number v.0.3; last revision 2011/11/05.

[†]e-mail: `claudio dot beccari at gmail dot com`



Golden number canonical layout

ISO canonical layout

Letter paper canonical layout

Figure 1: A right page with the relevant diagonal, the text block and the canonical circle. In this figure the important information is the page proportions, not the scale; matter of fact the letter paper is 17.6 mm shorter than the A4 paper, but the drawings to the same height emphasize the relative proportions of the various page parts.

Figure 1 depicts the situation described above; the gray rectangle represents the type block; the dotted page diagonal, as it can be seen, passes through the diagonal vertices of the type block; at the same time the circle is tangent to both the type block and the page contour.

It must also be noticed, in the sketch on the left, that the page rectangle is a golden one; therefore also the type block is a golden rectangle; the inner to the outer margin ratio is the golden section value so as this same ratio exists between the upper and the lower margin. This canonical layout implies that the type block height is equal to the page width.

Therefore this canonical layout is not suited when the page width is too small compared to the page height: with the golden section¹ this ratio is 0.618, with the ISO paper shape, in the sketch on the center, this ratio is 0.707, and so on.

But it is equally unsuitable with squarish pages, such as the sketch on the right where the page proportions conform with the USA letter paper size: the type block leaves little space for the margins, as it is evident from the properties of the canonical circle, which is tangent to the page lateral sides and to the type block horizontal sides; to the limit in a square page the type block is so large that there is no room for the margins.

2 The mathematics of the canonical layout

If one plays a little with the drawing of figure 1 and applies what has been described in the Introduction, may easily derive the following formulas:

$$\text{Page shape ratio:} \quad x = b/h \quad (1)$$

¹Here we denote $\phi = 1,618\dots$ with the name of *golden number*, and its reciprocal $\varphi = 0,618\dots$ with the name of *golden section*.

Inner margin:	$I = bx(1 - x)/(1 + x)$	(2)
External margin:	$E = b(1 - x)/(1 + x)$	(3)
Top margin:	$T = hx(1 - x)/(1 + x)$	(4)
Bottom margin:	$B = h(1 - x)/(1 + x)$	(5)
Text width:	$W = xb$	(6)
Text height:	$H = b$	(7)
Circle center abscissa:	$C_x = 0.5b$	(8)
Circle center ordinate:	$C_y = h(1 - x)/(1 + x) + b/2$	(9)
Circle radius:	$R = 0.5b$	(10)

whose results are summarized in table 1 for three common values of the shape factor and for the letter paper shape ratio.

Actually the formulas 1–10, and their numerical evaluations reported in table 1, are just for general information. The actual computations are performed by this package, either by means of the `memoir` built-in facilities or by explicit computations that exploit the ε -TeX extensions. But it can be easily seen that as the trimmed page becomes more squarish, the page size ratio tends to one, and the margins become smaller and smaller. Therefore it seems that this layout is most suited for intermediate page ratios, such as the ISO one or the common European book size of 170 mm $\times$ 240 mm with a ratio of 0,708.

3 Usage

This package is invoked as usual with:

```
\usepackahe{canoniclayout}
```

No options are necessary; but it is necessary to invoke the package only after the page dimensions are known, be they specified by an option passed to the class, such as, for example, `a4paper`, or by means of a `pagesize` specification, for example `pagebv` for calling the ISO B5 size.

Table 1: Canonical layout ratios referred to the the base (or width, b) of the trimmed page and its height (h).

Quantity	Canon			
	$\phi : 1$	$3 : 2$	$\sqrt{2} : 1$	$11 : 8.5$
Page shape ratio	0,618	0,666	0,707	0,773
Inner margin	$0,146b$	$0,133b$	$0,121b$	$0,099b$
External margin	$0,236b$	$0,200b$	$0,172b$	$0,128b$
Upper margin	$0,146h$	$0,133h$	$0,121h$	$0,099h$
Lower margin	$0,236h$	$0,200h$	$0,172h$	$0,128h$
Text width	$0,618b$	$0,666b$	$0,707b$	$0,773b$
Text height	b	b	b	b
Circle center abscissa	$b/2$	$b/2$	$b/2$	$b/2$
Circle center ordinate	$0,545h$	$0,533h$	$0,526h$	$0,515h$
Circle radius	$b/2$	$b/2$	$b/2$	$b/2$

Table 2: Text line lengths measured in average number of characters in different fonts

Font collection	A4 paper size			A5 paper size			B5 paper size		
	10 pt	11 pt	12 pt	10 pt	11 pt	12 pt	10 pt	11 pt	12 pt
Times eXtended	92	84	77	64	59	54	77	70	64
Latin Modern	86	79	73	60	55	51	72	66	61
Kepler Project	83	76	69	58	53	48	69	63	58
Palatino eXtended	82	75	69	58	53	48	69	63	57

At any convenient moment after calling this package, the trimming margins with respect to the stock paper may be computed and the layout rechecked and fixed for consistency.

4 Font normal size

At the same time if this page layout is chosen, the font normal size should be selected on the actual text block width; it is possible that the golden ratio type block, on a golden rectangle paper obtained by trimming an ISO A4 stock paper, might be suitable with a 10pt normal font size (exactly 66 characters per line with 10pt Latin Modern fonts), while a full ISO A4 paper yields a type block that might be more suited for a 12pt normal font size (86 characters with 10pt Latin Modern fonts). The computations might be left to the `memoir` class commands, but it's better the user is aware of this particular point. For his/her sake this package provides the command `\currentfontletters` (to be used only after `\begin{document}`}, because specific font packages might be loaded after the page layout has been computed; in any case the command is “text producing”, therefore it can't be used while in the preamble!) that types out a sentence such as “The number of characters in one text line with the current font is *nn*”, where the *nn* is the actual number printed out. If such number is much larger than the ideal Bringhurst range of 66 ± 5 , it's wiser to chose a larger normal font size. In any case the alternative command `\charactersperpage` types out the same information, the number of lines and the estimated number of characters per page. These commands are supposed to be useful while setting up the complete preliminary parameters for the page layout.

Just to have a comparison, table 2 displays the number of characters in one line of text for several fonts and for page sizes A4, A5, and B5.

It can be seen that the bold measures satisfy the Bringhurst rule with a number of characters in the range 66 ± 5 ; the measures typeset in medium series type are out of the optimal Bringhurst range, but may be considered acceptable; the measures typeset in italics are completely out of the interval 66 ± 10 and should not be considered acceptable. This means that the large A4 paper size may be best used with intrinsically large fonts such as the Palatino eXtended (package `pxfonts`) and the Kepler Project (package `kpfonts`). With the smaller A5 paper size with 10pt size fonts the best ones are the Times eXtended (package `txfonts`), but substantially all the examined fonts are acceptable, while with 12pt fonts none are acceptable.

With an intermediate sized paper, such as a B5 ISO size, all the considered fonts are acceptable inside or just outside the optimal Bringhurst interval.

Many books are trimmed to a size close the B5 paper size, therefore the page layout produced with this package is most useful with medium and smaller sizes (at least with ISO sizes).

5 Conclusion

I don't think this package should set also the header and footer distances, for example, or should set any specification for the page "decorations", such as underlined headers, or overlined footers, or the size, font family, series and shape of the header and footer contents. These specifications are of course important, but have nothing to do with the relationship between the paper size, the page size and the type block size.

As a final remark it's worth noting that a single pass of *pdflatex* on this `canoniclayout.dtx` file produces at the same time both the `canoniclayout.sty` package file and the documentation `canoniclayout.pdf` file. After running *pdflatex* move `canoniclayout.sty` to a `tex/latex/canoniclayout/` folder and `canoniclayout.pdf` to a `doc/latex/canoniclayout/` one; if such folders don't exist, create them on your personal T_EX tree. After moving these files remember to refresh the file name data base.

6 Documented code

The package preliminaries are already in place. We are going to make some computations in order to determine the page shape factor x ; to this end we exploit the dimensional computing capabilities of ε -T_EX, that are already embedded in the most recent implementations of the program *pdf_{te}x*; in fact it must be remarked that these facilities are being embedded into the *pdf_{te}x* typesetting engine since the T_EX Live distribution of the year 2005. If these facilities are not available, either because the T_EX distribution is a little too old, or because the real *tex* typesetting engine, not *pdf_{te}x*, is being used, then, this entire package contents is skipped and nothing is done, except issuing a suitable warning.

```
1 (*package)
2 \expandafter\ifx\csname eTeXversion\endcsname\relax
3 \PackageWarning{canoniclayout}{%
4 The typesetting engine is not suited\Messagebreak
5 for working in extended mode.\Messagebreak}
6 This package is being skipped}
7 \expandafter\endinput
8 \fi
9 \ifnum\eTeXversion>\z@%
10 \PackageWarning{canoniclayout}{%
11 The typesetting engine is not working in extended mode.\MessageBreak
12 This package is being skipped}
13 \expandafter\endinput
14 \fi
```

In order to have this layout work with the `memoir` class or with any other class we have to test the fact they the package has been loaded within the preamble of a `memoir` styled document; according to the result of this test we can exploit the `memoir` facilities for determining the page geometry, or we have to carry out the complete computations ourselves.

```
15 \ifclassloaded{memoir}{%}% balancing brace
```

Working with the `memoir` facilities, we can determine the page shape factor and we save it into the control sequence `\CLshape`. The trick of multiplying by `p0`, a length of 1 pt, is a simple trick for reducing the shape ratio to a length whose numerical value is exactly the ratio we are interested in. By stripping the 'pt' part from this dimension value we remain with the real number we want to keep in memory. Of course we need also the reciprocal value, therefore we have to perform this calculation a second time with different factors. This "strange" way of computing real ratios derives from the ability of the ε -T_EX engine to scale lengths with a higher computational

precision than “normal” T_EX can do with the standard dimension and integer operations. In facts all the computational operations are performed in the CPU registers and the intermediate first result of the multiplication is kept into a double length register, so that no bits are rounded or truncated; the division operation is performed with a double length numerator and a single length denominator, so that, again, almost no information is lost at the inevitable end of the division algorithm with a non vanishing remainder.

```
16 \dimen@=\dimexpr \p@ * \paperwidth / \paperheight\relax
17 \edef\CLshape{\strip@pt\dimen@}
18 \dimen@=\dimexpr \p@ * \p@ / \dimen@\relax
19 \edef\CLinvshape{\strip@pt\dimen@}
```

We then assign the text height, thanks to the fact that the canonical layout has the block height just as long as the page width, due to the circle property mentioned in the Introduction; in any case the formulas 1–10 and table 1 confirm this property for any shape.

```
20 \textheight=\paperwidth
```

This done, we can use the internal memoir commands to determine the other layout dimensions.

```
21 \settypeblocksize{\textheight}{*}{\CLshape}
22 \setlrmargins{*}{*}{\CLinvshape}
23 \setulmargins{*}{*}{\CLinvshape}
24 \setmarginnotes{7pt}{\dimexpr \foremargin-3\marginparsep}{\onelineskip}
25
```

We now use the `\checkandfixthelayout` so that it adjusts the type block dimensions to an integer number of lines in the normal font and eventually prints the page parameters to the `.log` file.

```
26 \checkandfixthelayout
```

At this point we are finished with the computations if the memoir class is the active one, but we need to perform the complete calculations in case we are not working with the memoir class.

```
27 }{%
```

We start by defining the lengths registers that hold the real inner, outer and upper margins, without the one inch shift and without taking into consideration the headings and their spacings, that must be specified before calling this package.

```
28 \newlength\innermargin
29 \newlength\outermargin
30 \newlength\uppermargin
```

Then we define scratch length registers in order to maintain intermediate values.

```
31 \dimendef\CLpageratio=1256% x=b/h
32 \dimendef\CLsmallratio=1258% z=x*y
33 \dimendef\CLlargeratio=1260% y=(1-x)/(1+x)
```

Now we determine the actual values based on the page width and height; we work on dimensions, and we extract the decimal fractional value by stripping off the ‘pt’ part of their measure; we keep the decimal fractional values into service macros:

```
34 \CLpageratio=\dimexpr \p@*\paperwidth/\paperheight\relax % x
35 \CLlargeratio=\dimexpr \p@*(\p@-\CLpageratio)/(\p@+\CLpageratio)\relax % y
36 \CLsmallratio=\dimexpr \CLpageratio*\CLlargeratio/\p@\relax % z
37 %
38 \edef\CL@pageratio{\strip@pt\CLpageratio}
39 \edef\CL@smallratio{\strip@pt\CLsmallratio}
40 \edef\CL@largeratio{\strip@pt\CLlargeratio}
```

We compute the type block dimensions:

```
41 \textheight\paperwidth% altezza griglia
42 \textwidth=CL@pageratio\textheight
```

We then compute the real margins between the page sides and the type block:

```
43 \innermargin=CL@smallratio\paperwidth
44 \outermargin=CL@largeratio\paperwidth
45 \uppermargin=CL@smallratio\paperheight
```

But the typesetting engine does not use the net margins, due to the historical one inch shift of the original T_EX implementation and, for the upper margin, due to the fact that by default this layout implies the headings and the footers outside the type block. Since we have the real outer margin dimension, we compute the actual default marginal note measure:

```
46 \marginparwidth=\dimexpr0.8\outermargin-\marginparsep\relax
47 \evensidemargin=\dimexpr\outermargin -1in\relax
48 \oddsidemargin=\dimexpr\innermargin-1in\relax
49 \topmargin=\dimexpr\uppermargin-1in-\headheight-\headsep\relax
```

When using different classes from the memoir one, we adjust the type block dimensions so that the type block contains an integer number of lines; we use the normal size `\baselineskip` value and compute the number of lines contained in the initial type block minus the first line stander height equal to `\topskip`; notice that the integer value expression computed for counter 256 is a rounded value, not a truncated value, therefore the actual type block height might get enlarged or reduced a very little bit that will not be noticed with naked eye when one examines a typeset page; eventually we recompute the `\textheight` taking into account the height of the first line.

```
50 \normalfont
51 \count256=\numexpr(\textheight-\topskip)/\baselineskip\relax
52 \textheight=\count256\baselineskip\relax
53 \textheight=\dimexpr\textheight+\topskip\relax
54 \normalbaselineskip\baselineskip
```

and we eventually close the second action depending on the use of the memoir class:

```
55 }
```

Just for testing the number of characters in a line, the following macros are provided. A couple of scratch dimension registers will be used `\dimen@` and `\abcwidth`; `\abcwidth` will hold the current lowercase alphabet length, and `\dimen@` will hold several values in succession. The scratch counter `\count254` will receive the final result to be printed out. ε -T_EX arithmetics will be used throughout.

Another service macro for the user's sake is `\charactersperpage`; this macro typesets the average number of characters per line, the numbers of lines per page, and eventually their product, a good estimate of the number of characters per page.

Many text editors have facilities for counting a “compuscript” number o words and characters; with this further information the user may easily estimate the number of pages of his/her document; of course s/he must allow a certain allowance for the front and back matter pages, so that s/he can evaluate the total amount of signatures of the whole typeset document; of course, by typesetting it with *(pdf)latex* s/he knows exactly the number of pages of the final document, but the typographer wants to know a good estimate before the source T_EX files are finished.

```
56 \newlength\abc@width
57 \newcommand*\currentfontletters{%
58 \settowidth\abc@width{abcdefghijklmnopqrstuvwxyz}\dimen0=26\p@
59 \dimen0=\dimexpr \textwidth * \dimen0 / \abc@width \relax
60 \count254=\numexpr \dimen0 / \p@ \relax
61 \par The number of characters in one text line with the current font is
```

```

62   \the\count254.\par}
63
64 \newcommand*\charactersperpage{%
65 \settowidth\abc@width{abcdefghijklmnopqrstuvwxyz}\dimen0=26\p@
66 \dimen0=\dimexpr \textwidth * \dimen0 / \abc@width \relax
67 \count254=\numexpr \dimen0 / \p@ \relax
68 \count252=\numexpr \textheight / \baselineskip \relax
69 \count250=\numexpr \count252 * \count254\relax
70 \begin{quote} The average number of characters in one text line
71   with the current font is \the\count254;
72   this text block contains \the\count252\ lines;
73   a rough estimate of the number of characters per page is
74   \the\count250.\par
75   This information may be useful to evaluate the total number
76   of pages in a document.
77 \end{quote}}
78
79 \end{package}

```