



**Do it
better
or easier...**

Karel Horák

TeXperience 2010
Břežlov, September 17

Phil Taylor's optimization of parameters

```
\emergencystretch = 0pt
\pretolerance = 150
\tolerance = 250
\hbadness = 150
\hfuzz = 0pt
\tolerance 9999 %%% we look for the worst underfull box
% \tolerance 318 \hbadness 317
```

Phil Taylor's optimization of parameters

```
\emergencystretch = 0pt
\pretolerance = 150
\tolerance = 250
\hbadness = 150
\hfuzz = 0pt
\tolerance 9999 %%% we look for the worst underfull box
% \tolerance 318 \hbadness 317
```

The following definitions also could be useful:

```
\def\em#1 {\bgroup\advance\emergencystretch#1 \par\egroup}
\def\hf#1 {\bgroup\hfuzz#1 \par\egroup}
\def\wpar{{\widowpenalty=0 \parfillskip 0pt \endgraf}}

\def\settight{\begingroup\spaceskip.23em plus.13em minus.1em
  \def\par{\endgraf\endgroup}}
```

Playing with \spaceskip parameters

```
\def\setttight{\begingroup\ifproof\ifhmode\else
    \indent\fi\llap{:\kern1.5\parindent}\fi
\spaceskip.8\fontdimen2\font plus
    .9\fontdimen3\font minus
    1.2\fontdimen4\font
\def\par{\endgraf\endgroup}}
```

Playing with \spaceskip parameters

```
\def\settight{\begingroup\ifproof\ifhmode\else
    \indent\fi\llap{:\kern1.5\parindent}\fi
\spaceskip.8\fontdimen2\font plus
    .9\fontdimen3\font minus
    1.2\fontdimen4\font
\def\par{\endgraf\endgroup}}
```

My experience with commercial fonts is that going the classical way (*afm2tfm*) often doesn't give the best spacing parameters, so I took some lessons from *fontinst*, which is rather sophisticated.

```
\def\fontdims#1:#2:#3:#4  {\%#1=\fontname
    #1 \fontdimen2#1=.#2em
    \fontdimen3#1=.#3\fontdimen2#1
    \fontdimen4#1=.#4\fontdimen2#1 }}
\fontdims\tenrm:28:60:24
```

One should know that \TeX sets these font parameters global!

Other Phil Taylor's very useful macros

```
\def\ignorewhitespace{\begingroup
  \chardef\ignored=9
  \catcode'\ =\ignored % space
  \catcode'\^^I =\ignored % tab
  \catcode'\^^L =\ignored % f/feed
  \catcode'\^^M =\ignored % e-o-l
  \futurelet\ignorewhitespace
\endgroup}
```

Other Phil Taylor's very useful macros

```
\def\ignorewhitespace{\begingroup
  \chardef\ignored=9
  \catcode'\ =\ignored % space
  \catcode'\^^I =\ignored % tab
  \catcode'\^^L =\ignored % f/feed
  \catcode'\^^M =\ignored % e-o-l
  \futurelet\ignorewhitespace
\endgroup}
```

The following macro for making titles should be generally known: it was published in TUGboat and even in Czech translation in $\mathcal{C}_S\text{TeX}$ bulletin. But looking in TUGboat it seems that nobody has read Phil's article, all the titles there are bad positioned making the lines of neighbouring columns inconsistent...

```

\newbox\headerbox \newdimen\headerheight \newdimen\headerdepth
\def \header #{\afterassignment\afterheader\setbox\headerbox=\vtop}
\def \afterheader{\noindent\aftergroup\reallyafterheader}
\def \reallyafterheader
  {\headerheight=\ht\headerbox
  \headerdepth =\dp\headerbox
  \advance\headerheight by\headerdepth
   \headerdepth=\headerheight
  \ht\headerbox=0pt
  \dp\headerbox=0pt
  \advance\headerheight by .5\baselineskip
  \divide \headerheight by \baselineskip
  \multiply\headerheight by \baselineskip
  \ifdim \headerheight < \headerdepth
    \advance \headerheight by \baselineskip
  \fi
  \box\headerbox
  \vskip \headerheight
  \noindent\ignorespaces}

```


Some trivial visualizations

```
\setbox\strutbox=\hbox{\vrule height.70833\baselineskip
                        depth .29167\baselineskip width\z@}
% kdyz vyjde do zlomu strany:
% \maxdepth\dp\strutbox
\let\l@@seness\looseness
\def\looseness#1
    {\kern0pt\strut
     \setbox0\rightline{\rlap{\enspace$\scriptstyle#1$}}%
     \ht0=-\dp\strutbox \dp0=\dp\strutbox
     \vadjust{\box0}\l@@seness#1 }
```

Sometimes it is rather helpful to use `\everypar{\looseness1}` or `\everypar{\looseness-1}` in the bigger part of text (section, chapter) to find the acceptable places... (believe me, surprises are not excluded).

Calculating `\baselineskip`

```
\newcount\numberoflines
\def\calculatebaselineskip{\baselineskip\vsiz
\advance\baselineskip-\topskip
\divide\baselineskip \numberoflines
\normalbaselineskip\baselineskip
\vsiz \numberoflines\baselineskip \advance\vsiz\topskip}
\numberoflines=46 \calculatebaselineskip
```

Calculating `\baselineskip`

```
\newcount\numberoflines
\def\calculatebaselineskip{\baselineskip\vsiz
\advance\baselineskip-\topskip
\divide\baselineskip \numberoflines
\normalbaselineskip\baselineskip
\vsiz \numberoflines\baselineskip \advance\vsiz\topskip}
\numberoflines=46 \calculatebaselineskip
```

```
\newdimen\givenVsize \givenVsize\vsiz
\def\setpage #1 {\baselineskip\givenVsize
\advance\baselineskip-\topskip
\divide\baselineskip #1
\normalbaselineskip\baselineskip
\vsiz #1\baselineskip
\advance\vsiz\topskip}
\setpage 46
```

```
\newcount\filcnt
\def\filskip #1 {\par\filcnt=0
  \edef\lastdepth{\ifdim\prevdepth>-1000pt
    \ifdim\prevdepth>\maxdepth 0pt\else\the\prevdepth\fi
    \else 0pt\fi}
\vskip-\lastdepth
\loop\ifnum\filcnt<#1\advance\filcnt 1
\vskip \baselineskip\penalty-100
\repeat \vskip-#1\baselineskip
\vskip\lastdepth}
```

```

\newcount\filcnt
\def\filskip #1 {\par\filcnt=0
  \edef\lastdepth{\ifdim\prevdepth>-1000pt
    \ifdim\prevdepth>\maxdepth 0pt\else\the\prevdepth\fi
    \else 0pt\fi}
\vskip-\lastdepth
\loop\ifnum\filcnt<#1\advance\filcnt 1
\vskip \baselineskip\penalty-100
\repeat \vskip-#1\baselineskip
\vskip\lastdepth}

```

As the usual practice of people typing e.g. pictures dimensions in a catalogue is to use x like 20x30 mm. Hence, the best solution would be to use some script (e.g. in perl) or regular expression module of editor to get 20\times30 mm. But it is also possible to leave x in the expression and use

```

\newcount\filcnt
\def\filskip #1 {\par\filcnt=0
  \edef\lastdepth{\ifdim\prevdepth>-1000pt
    \ifdim\prevdepth>\maxdepth 0pt\else\the\prevdepth\fi
    \else 0pt\fi}
\vskip-\lastdepth
\loop\ifnum\filcnt<#1\advance\filcnt 1
\vskip \baselineskip\penalty-100
\repeat \vskip-#1\baselineskip
\vskip\lastdepth}

```

As the usual practice of people typing e.g. pictures dimensions in a catalogue is to use x like 20x30 mm. Hence, the best solution would be to use some script (e.g. in perl) or regular expression module of editor to get 20\times30 mm. But it is also possible to leave x in the expression and use

```

\mathcode '\x="8000
{\catcode '\x\active\gdefx{\times}}

```

Measuring the length of the last line

The following macro should be well known (TBN). It was discussed in our bulletin many years ago in connection with special paragraph typesetting.

Recently I realized that it could be found useful for typesetting ornament at the end of paragraph:

```
\newdimen\lastlinewidth % the last line width
\def\testlastline{%\vadjust{\vskip\lskip}
  $$ \advance\preplaysize by-2em
  \global\lastlinewidth=\preplaysize
  \preplaysize=\maxdimen
  \abovedisplayskip=-1\baselineskip
  \belowdisplayskip=0pt %-2pt minus 4pt
  $$\endgraf}
\displaywidowpenalty 10000
\def\par{\futurelet\next\teststar}
\let\stars\relax
```

```
\def\teststar{\ifx\next\stars\let\nxt\st@rs
  \else\let\nxt\endgraf\fi\nxt}
\def\st@rs#1{\testlastline\nobreak
  \ifdim\lastlinewidth<.43\hsize\vskip-\baselineskip\fi
  \vbox to 0pt{\vbox to 1.5\baselineskip{\vss
    \centerline{\bars}
    \vss}
  \vss}\vskip 2\baselineskip\filskip1
  \noindent\ignorewhitespace}
```

blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla
blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla
blabla blabla blabla blabla



blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla
blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla blabla



Sergeant Pepper's Lonely Hearts Club Band

It was twenty years ago today
That Sergeant Pepper taught the band to play.
They've been going in and out of style
But they're guaranteed to raise a smile.

So may I introduce to you
The act you've known for all these years -
Sergeant Pepper's Lonely hearts Club Band!

Sergeant Pepper's Lonely Hearts Club Band

It was twenty years ago today
That Sergeant Pepper taught the band to play.
They've been going in and out of style
But they're guaranteed to raise a smile.

So may I introduce to you
The act you've known for all these years -
Sergeant Pepper's Lonely hearts Club Band!

```
\def\basen{\bigskip\beginngroup\linescnt=0
  \def\par{\ifhmode\endgraf\advance\linescnt1 \secondparfalse\else
    \ifsecondpar\endgraf\removelastskip\penalty-500
    \bigskip\endgroup
    \else\endgraf\secondpartrue\specialbreak\bigskip\fi\fi}
\obeylines\let\nxxt }
\def\specialbreak{\nobreak\ifnum\linescnt>1 \penalty-200 \fi}
```

If one needs more, e.g. to leave hanging the very long lines, he has to add other parameters:

It was twenty years ago today
That Sergeant Pepper taught the band
to play.
They've been going in and out of
style
But they're guaranteed to raise a
smile.

```
\def\verse{\bigskip\begingroup \parindent Opt  
  \advance\leftskip Opt plus1fil \rightskip Opt plus-1fil  
  \parfillskip Opt plus.9999fil  
  \everypar{\hskip Opt plus-1fil\relax}  
  \def\par{\ifhmode\endgraf\penalty100 \else\endgraf  
    \ifdim\lastskip<\bigskipamount\removelastskip  
    \penalty 0 \bigskip\endgroup\fi\fi}  
\obeylines\let\nxxt }
```

At this place it also worth to mention perl script *findhyph* of *Martin Budaj* which I have rather recently discovered (only some months before its publication in our $\text{T}_{\text{E}}\text{X}$ bulletin 1-2/2010 (its origin goes back to 2000). It will provide list of all the hyphenations in typeset text and proves to save much time when looking for badly hyphenated words.

At this place it also worth to mention perl script *findhyph* of *Martin Budaj* which I have rather recently discovered (only some months before its publication in our $\text{T}_{\text{E}}\text{X}$ bulletin 1-2/2010 (its origin goes back to 2000). It will provide list of all the hyphenations in typeset text and proves to save much time when looking for badly hyphenated words.

symbo-licky

[1]

rov-noběžné.

průsečí-ků.

[3]

polo-přímkou.)

přípa-dě,

hra-niční

přím-kami

[4]

Some workarounds with pictures

It would be nice if [pdf,lua]T_EX would be able to recognize that file was changed only locally and that a big portion of resulting pdf could be used without changes.

In several cases the following notes will concern PostScript. I am far from to be an expert in PostScript.

tiff2ps is a batch file which uses Ghostscript to read the tiff file and write down the so called *header file*. All is done on PostScript level.

Written in BOP s.c., Gdańsk, Poland (*P. Strzelczyk*)

Some workarounds with pictures

It would be nice if [pdf, lua]T_EX would be able to recognize that file was changed only locally and that a big portion of resulting pdf could be used without changes.

In several cases the following notes will concern PostScript. I am far from to be an expert in PostScript.

tiff2ps is a batch file which uses Ghostscript to read the tiff file and write down the so called *header file*. All is done on PostScript level. Written in BOP s.c., Gdańsk, Poland (*P. Strzelczyk*)

```
%GSC% -dNODISPLAY -q -- tif2eps.ps %1 %2 %3 %4 %5 %6 %7 %8 %9  
tiff2ps -h hermes
```

Some workarounds with pictures

It would be nice if [pdf,lua]T_EX would be able to recognize that file was changed only locally and that a big portion of resulting pdf could be used without changes.

In several cases the following notes will concern PostScript. I am far from to be an expert in PostScript.

tiff2ps is a batch file which uses Ghostscript to read the tiff file and write down the so called *header file*. All is done on PostScript level. Written in BOP s.c., Gdańsk, Poland (*P. Strzelczyk*)

```
%GSC% -dNODISPLAY -q -- tiff2eps.ps %1 %2 %3 %4 %5 %6 %7 %8 %9
tiff2ps -h hermes
```

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 1134.0 756.0
%%Creator: TIF2EPS ver. 0.79 / GPL Ghostscript 856
%%DocumentNeededResources: file b10-1.tif
```



```
/OWOI 1134 def
/OHOI 756 def
...
% End of preparing TIFF data source
0 0 translate
1134.0 756.0 scale
/DeviceGray setcolorspace
<<% ImageDictionary
  /ImageType 1
  /Width 1134
  /Height 756
  /ImageMatrix [ 1134 0 0 -756 0 756 ]
  /DataSource currentfile pop filteredfile
  /BitsPerComponent 8
  /Decode [ 0 1 ]
>>
image
```

The output is not as easy readable as that given before 2005 which inspired me to use the same method for other huge eps files:

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 1194.75 1100
%%DocumentNeededResources: file ts019.eps
-10 -50 translate
1.07 dup scale
(ts019.eps) run
```

Only this small header file is read by `dvips` which is done almost immediately.

The output is not as easy readable as that given before 2005 which inspired me to use the same method for other huge eps files:

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 1194.75 1100
%%DocumentNeededResources: file ts019.eps
-10 -50 translate
1.07 dup scale
(ts019.eps) run
```

Only this small header file is read by `dvips` which is done almost immediately.

For such kind of pictures (black&white, well scanned) another possibility exists:

```
|potrace -c herm.bmp|
```

`-c` to get readable, not compressed PostScript.

```
%!PS-Adobe-3.0 EPSF-3.0
%%Creator: potrace 1.8, written by Peter Selinger 2001-2007
%%LanguageLevel: 2
%%BoundingBox: 0 0 468 473
%%Pages: 1
%%EndComments
%%Page: 1 1
...
/z{closepath}D
/b{0 setgray z fill}D
/w{1 setgray z fill}D
```

```
%!PS-Adobe-3.0 EPSF-3.0
%%Creator: potrace 1.8, written by Peter Selinger 2001-2007
%%LanguageLevel: 2
%%BoundingBox: 0 0 468 473
%%Pages: 1
%%EndComments
%%Page: 1 1
...
/z{closepath}D
/b{0 setgray z fill}D
/w{1 setgray z fill}D
```

When looking into uncompressed *potrace* output, one would quickly discover the last two definitions of keys `/b` and `/w`, which are easy to be changed into any `rgb` or `cmyk` colour.

Potrace is very good in tracing (I have an old experience with Coreltrace and Adobe's streamline), it is incorporated into other great programs like *inkscape* and *fontforge*. Now, take a look at Type1 version of METAFONT based Hebrew font `hclassic` which is to be found at CTAN.

Potrace is very good in tracing (I have an old experience with Coreltrace and Adobe's streamline), it is incorporated into other great programs like *inkscape* and *fontforge*. Now, take a look at Type1 version of METAFONT based Hebrew font `hclassic` which is to be found at CTAN.



Potrace is very good in tracing (I have an old experience with Coreltrace and Adobe's streamline), it is incorporated into other great programs like *inkscape* and *fontforge*. Now, take a look at Type1 version of METAFONT based Hebrew font `hclassic` which is to be found at CTAN.



Potrace is very good in tracing (I have an old experience with Coreltrace and Adobe's streamline), it is incorporated into other great programs like *inkscape* and *fontforge*. Now, take a look at Type1 version of METAFONT based Hebrew font *hclassic* which is to be found at CTAN.



From the font header

```
%!PS-AdobeFont-1.0: TeX-hclassic 001.001  
% Filtered by type1fix.pl 0.05
```

I guess that this Type1 version of *hclassic* was created with $\text{T}_{\text{E}}\text{X}$ trace? As one can see the tracing with *potrace* in *fontforge* gives clearly much better result...

Sometime it is useful to have pictures which colour could be changed “dynamically”. With `dvips` it could be done with `\special` command:

Sometime it is useful to have pictures which colour could be changed “dynamically”. With dvips it could be done with `\special` command:

```
/setcmykcolor where {pop}
  {/setcmykcolor {1 sub /mf exch def
  3 {mf add neg dup 0 lt {pop 0} if 3 1 roll} repeat
  setrgbcolor} def} ifelse
/q {gsave} def
/Q {grestore} def
%%EndProlog
%%BeginSetup
%%EndSetup
*u
logocolor
...
```

where `logocolor` should be defined in ones' `TEX` file. Clearly, such an `eps` file cannot work standalone.

Only when preparing these slides I had recall my old temptation to "improve" some often used eps files with logos, which should be coloured differently or even greyed.

Only when preparing these slides I had recall my old temptation to "improve" some often used eps files with logos, which should be coloured differently or even greyed.

I got inspired by T_EX's `\ifundefined`. I had looked into famous Red Book, read something about known operator (without any progress) and suddenly notice the two lines in previous program's key definitions

```
/setcmykcolor where {pop}  
  {/setcmykcolor {1 sub /mf exch def  
  3 {mf add neg dup 0 lt {pop 0} if 3 1 roll} repeat  
  setrgbcolor} def} ifelse
```

Only when preparing these slides I had recall my old temptation to "improve" some often used eps files with logos, which should be coloured differently or even greyed.

I got inspired by T_EX's `\ifundefined`. I had looked into famous Red Book, read something about known operator (without any progress) and suddenly notice the two lines in previous program's key definitions

```
/setcmykcolor where {pop}  
  {/setcmykcolor {1 sub /mf exch def  
  3 {mf add neg dup 0 lt {pop 0} if 3 1 roll} repeat  
  setrgbcolor} def} ifelse
```

and tried to add the following definition

```
/logocolor where {pop} {/logocolor {1 0 1 .4 k} def} ifelse
```

BUT one should keep in mind that using `dvips` all such the specials are collected at the beginning of the PostScript file. It means that the following example would not result in two differently coloured pictures:



Surprisingly also `currentcmykcolor` exists! Discovering this I was happy enough to rewrite `pst-grad.tex` macros to get the possibility of easy manipulation of `cmyk` gradients.

My first gradient (it was long time before free release of John Hobby's METAPOST!) was prepared as `eps` file output by some old version of Corel.

When I got learned some basics from `pstricks` macros, I found also the way how to produce nice gradients

```
gsave \pst@usecolor\psgradbegin currentcmykcolor grestore
gsave \pst@usecolor\psgradend currentcmykcolor grestore
\psgradlines
```

Of course, the definitions in `pst-grad.pro` should be changed appropriately.

One of the last clever programs related to PostScript is the following code from `rules.tex` also supplied by dear Polish colleagues

```
\def\RULES{\special{ps:
  /v {/ruley exch def /rulex exch def V} def
  /V {gsave newpath transform round exch round exch itransform
    moveto rulex 0 rlineto 0 ruley neg rlineto rulex neg 0 rlineto
    closepath fill grestore} def
}}\RULES
```

which changes the default behaviour of `dvips` when interpreting $\text{T}_{\text{E}}\text{X}$'s rules. Sometimes it is helpful when the PostScript rip produces (almost?) invisible lines which I encountered some years ago preparing rather heavy physics textbook.

`\colormap`

There is another great utility from Polish BOP which enables to change colour of grey images. One will find these macros in `colormap.tex`.

`\colormap`

There is another great utility from Polish BOP which enables to change colour of grey images. One will find these macros in `colormap.tex`.



```
\lincmymap[0 .3 0 0 : 0.5 .5 0 0]{\epsfysize 3in  
  \epsfbox{hermes.eps}}
```

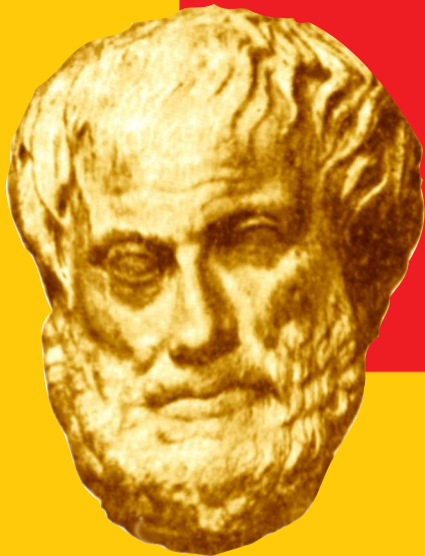
`\colormap`

There is another great utility from Polish BOP which enables to change colour of grey images. One will find these macros in `colormap.tex`.



```
\lincmymap[0 .3 0 0 : 0.5 .5 0 0]{\epsfysize 3in  
  \epsfbox{hermes.eps}}
```

The other picture was generated by *potrace*.



To get clipped picture one can easily in any bitmap program produce black filled contour, which could be traced and then in the resulting file it is sufficient to remove stroke command (or some of its synonym, it depends on the tracing program):

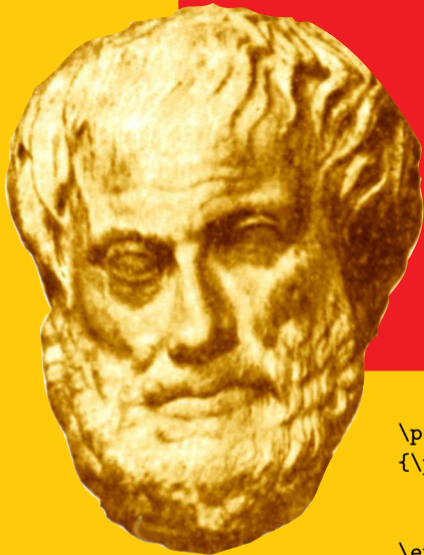
```
% stroke  
closepath
```



To get clipped picture one can easily in any bitmap program produce black filled contour, which could be traced and then in the resulting file it is sufficient to remove stroke command (or some of its synonym, it depends on the tracing program):

```
% stroke  
closepath
```





To get clipped picture one can easily in any bitmap program produce black filled contour, which could be traced and then in the resulting file it is sufficient to remove stroke command (or some of its synonym, it depends on the tracing program):

```
% stroke  
closepath
```



```
\psclip  
{\pscustom[linestyle=none]%  
  {\file{a.ai}}}%  
  \epsfbox{aristote.eps}  
\endpsclip
```

ELIAS LEVI
STVI SMRTI A. DESBAROLLES
IM ZŘETEM KE CHROMOLES
GICKÁ PŘEDHISTORIE
ZASVĚCENÍ V HERMETISMU
DRAKOKAMŮ V HERMETISMU
POSISMIUS PIERRE DE LA
FRANTISEK KABELÁK
FRANTISEK KABELÁK
SIDERICKÉ KVADRO
ALCHYMIE JAN KEER
VÝZNAMU KOVŮ V DĚJINÁCH
ČÍNSKÁ ALCHYMIE HERMETICKÉ
PŘEDKOLUMBOVSKÉ AMERICE
EGYPTSKÁ KNIHA MRTVÝCH PO
RUKY PIERRE DE LA SÉNIE
PARACELSUS O PRODLOUŽENÍ

