TIA-simpler-WTDI

Abstract
Recreational use of TEX&Co in my work is enumerated and elucidated. Examples from MetaFun, from Lancaster's Fonts for Free, from Jackowski&Ryćko metafont logo, and from Word have been borrowed. PostScript and let TEX insert mark-up, will be the main subjects of discussion. PostScript is not sufficient for graphics. Now and then MetaPost is used to specify a problem in a declarative way, or at the end Photoshop is used to enrich the graphics interactively by colour gradients. Moreover, for drawing emulations of 3D objects, projection techniques are indispensable. Emulations of Escher's impossible cube and of Gabo's objects are included as 3D-examples. All my pictures have a recreational flavour because none has been triggered by external practical need. Interesting is the combined use of Turtle Graphics and recursion. TEX codes and PostScript codes are compared, although they are like apples and pears intrinsically incomparable, but ... have been used for the same purpose. The most astonishing is that so much from `BLUe.tex` passed by unnoticed. `Pic.dat` for TEX-alone pictures has received its cousin library, `PSlib.eps`, for PostScript pictures. The TEX-MF-flow picture has been updated and included, next to a screen-shot of a nowadays IDE TEXworks.
In this note I'll try to draw your interest, to persuade you, kind reader, to look at the contents, the paradigms, and the kernel and modules set-up of `BLUe.tex`. My sincere hope is that `BLUe.tex` will be saved from oblivion, that the paradigms used will be adhered.
The serious undertone in TEX is about minimal mark-up or better still the absence of user mark-up, where TEX will insert the mark-up. The serious undertone in PostScript is about printing along paths, especially for the special cases where the paths are implicit. Handy and convenient is the extended `PSlib.eps` to over 300 pictures.
Critics on TEX&Co and pdfTEX have been included, next to my wishes.
After the presentation `PSTricks` was shown to me, and my comment on it is included.

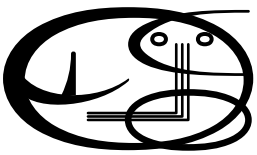
Keywords
Acrobat Pro, Adobe, art, automatic mark-up, backtracking, BLUe, Blue Sky research, bridge, Caroll, chess, Cohen, ConTEXt, crosswords, dancing text, Deubert, Ensor, EPSF, Escher, FIFO, font charts, function-grapher previewer, Gabo, Hagen, Henderson, IDE (Integrated Development Environment), impossible figures, Jackowski, Lancaster, Lauwerier, LIFO, Lindenmayer, magic square, Malevich, Margritte, MetaFun, MetaPost, Metafont, MetaType1, minimal encapsulated PostScript, minimal mark-up, minimal plain TeX, Mondriaan, Monte Carlo, musiX.tex, mppreviewer, Nolde, Photoshop, π-decimals, projection, PSlib, PSTricks, PSView, Pythagoras tree, Ryćko, Schrofer, Soto, Taupin, TEXworks, tic-tac-toe, Vasarely.

Introduction

In the late 80s I became aware of TEX and immediately realized the relevance for a university community. I started the 'Publiceren met LATEX' project, which resulted in the CWI-syllabus 19. We organized a LATEX course at Utrecht. University users found their way in how to use LATEX. I became 1st president of NTG.

In order to learn macro-writing I developed my bridge macros, which marks my start of Recreational use of TEX. My learning of macro-writing culminated in `BLUe.tex` and my 'Publishing with TEX' guide,[1] which concentrates on what can be done by TEX alone, without incorporating the results of graphics software.

My next project was typesetting tables by TEX, where I en-passant looked for a taxonomy of tables. The conclusion of this work was that tables are too varied, but one could discern a broad class of tables which have a border and in there the proper table data. This lead to my table macros, which I presented at the EuroTEX92 at Prague.[2] My Recreational use of TEX in the table area are amongst others bridge layouts, the crosswords table, a magic square, and the PASCAL's triangle of binomial coefficients.



At the Prague conference I was impressed by Karel Horák's graphical work in Metafont. At home I started to use Metafont for graphics, mainly recreational, which resulted in my 'cat'.[3] The incorporation of Metafont graphics in TEX via symbols of a font I experienced as inconvenient. Later I learned about `\psfig`, which easily lets you include PostScript pictures in TEX-documents. The use of `\psfig` marks my beginning of TEX&PostScript. My viewer was the Apple Laserwriter, and not `PSView` which was not available on my PowerMac. My use of PostScript as part of TEX&Co has a strong recreational flavour. Many pictures have been inspired by work of artists, such as Escher, Gabo, Malevich, Mondriaan, Soto, Schrofer, Vasarely, ... as can be witnessed in this paper. Pictures have been improved:

- PostScript pictures resulting from MetaPost with at least better BoundingBox values,
- gkp-pictures were done anew in PostScript now and then, and included in `PSlib.eps`.

All pictures come with better explanations. Inclusion in my pdfTEX-documents goes by the macros `\pdfximage... \pdfrefximage\pdflastximage`, or my less-verbose macros `\insertjpg`, c.q. `\insertpdf`.

My use of graphics with TEX marks five periods: 1st by LATEX's picture-environment, 2nd by TEX's `gkp`-macros, with the same functionality as the picture-environment, 3rd by Metafont, supported by projection techniques, 4th by MetaPost, 5th by PostScript, supported by Photoshop as post-processor, mainly for colour gradients.

This paper consists of examples from earlier MAPS papers, from Hagen's MetaFun, from Lancaster's Fonts for Free, from the 3D Jackowski&Ryćko metafont logo, from Word and from literature. The 1st appendix contains my balanced binary tree macros in TEX of old next to my superior PostScript variant, on occasion of the EuroTEX-ConTEXt2012. Another appendix contains determination of the BoundingBox values in 1-pass, on-the-fly. The 3rd appendix contains a LMR font table.

Select what you are interested in. If only you enjoy one picture, kind reader, I'm happy already ■.

1. Available from CTAN.
2. Knuth considers bordered matrices in the TEXbook but does not mention bordered tables.
3. Much later the cat was adapted to MP and the resulting data resulted in an EPSF.

One by one the guests arrive, MAPS 96.2 1996

This 1-page paper is best read with Cohen's song in the background.[4] It is a plea for serious — non-recreational — use of TEX.

> "*...This plea, this shout, hopes to awaken the notion that we are all better off if we write macros in the lowest common set of all TEX-flavours, i.e. plain TEX. At least it might initiate a discussion whether to do so or otherwise, because I'm realistic enough that not all share my views ...*"

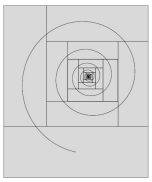
A little later the song continues

> "*And no one knows where the night is going*
> *And no one knows why the wine is flowing*
> *O love, I need you, I need you, I need you*
> *I need you now ...*"

The point I'm trying to make is that we are all better off when complex fundamental parts will be programmed in plain TEX, perhaps after it has proven to be worth it.[5]

To end Cohen's song

> "*The guests are coming through*
> *The open-hearted many*
> *The broken-hearted few*"



Looking back the TEX-community decided otherwise: LATEX-packages are contributed to CTAN; ConTEXt and LuaTEX were developed; the new Latin Modern Roman fonts are Adobe Type 1 . The GUST e-foundry TEX-Gyre OTF-project is under way, funded by several LUGs and TUG. But ... nevertheless I keep saying it.

Macros from `BLue.tex` and pictures from `PSlib.eps` can be reused even by ConTEXt, LaTEX, ...respectively MetaPost users, because they are written in the common plain TEX subset, respectively the underlying PostScript. But there is more than reuse ...MetaType 1 fonts are in Adobe Type 1, however ... Adobe has declared Adobe Type 1 obsolete, see Ludwichowski this proceedings.

The Life-cycle diagram of publications is one of my favourites. The invoke of `\halign` is straight-forward.

Produce → Distribute → Consume
↑ ↑ ↓
reuse ← retrieve ← store

```
\halign{&\enspace\hfil#\hfil\cr
   Produce&$\rightarrow$&Distribute&$\rightarrow$&Consume\cr
$\uparrow$&            &$\uparrow$&            &$\downarrow$\cr
    reuse&$\leftarrow$ &retrieve  &$\leftarrow $&store\cr}
```

In principle the above life-cycle is OK, but ... in practice the reuse aspect is hampered by changes, such as a different IDE

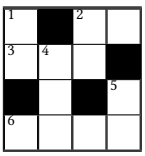
- ☐ or a new TEX engine, such as pdfTEX, which no longer supports for example `\psfig`
- ☐ or programs have become obsolete such as the picture environment
- ☐ or the gkp-macros have become outdated, as happened with the Happy Birthday cake picture.

Moreover, it is hard to maintain original data over time, over computer renewals. Nevertheless ...

4. During the presentation the tune was played by just pushing a button in my slide, multi-media, aha.
5. The same holds for pictures: we should create and adhere a library of PS pictures. Why not start with `PSlib.eps`?

Typesetting Crosswords via TEX, MAPS 8, 1992

The typesetting crosswords tool, as one of the tools in `tools.dat`, comes with `BLUe.tex`. The environment is `\begincrosswords` ... `\endcrosswords`. The example has been borrowed from the table chapter of PWT. The crosswords tool has been copied from BLUe's `tools.dat` and used stand-alone in this paper.



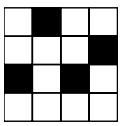
Across
2 Switch mode
3 Knuth
6 Prior to TEX

Down
1 Public domain
2 All right
4 All comes to it
5 Atari type

```
\begincrosswords
$\bdata
P*On
DEk*
*n*S
Edit
\edata
\crw
<Clues in 2 \vtop's, v-centered>
\sol
\endcrosswords
```

Interesting is the near-WYSIWYG-data specification of the puzzle. Minimal mark-up has been strived after, no `\cr`-s nor `&`-s have to be inserted by the user, TEX will do it for you. Mean-and-lean is that the solution or the puzzle can be toggled by `\sol` respectively `\crw`. Note the use of capitals and lower case. The capitals mark where a number for the clues has to be inserted, automagically. Paradigm: let TEX insert mark-up.

A variant via PostScript inspired by David Byram–Wigfield,[6] who created a special font `QuadFont`, interesting in itself, for the black and white squares. But ... without numbers for the clues and no toggling of solution and puzzle. In my PS version I simplified, without creating `QuadFont`. TEX's version is superior.



```
/x{0 0 20 20 rectstroke 20 0 translate} def
/X{0 0 20 20 rectstroke 0 0 20 20 rectfill 20 0 translate} def
/crl{-80 -20 translate 0 0 moveto} def
x X x x crl              %X denotes black and x denotes white
x x x X crl
X x X x crl
x x x x crl
```

Typesetting Bridge via TEX, MAPS 7, 1991

My recreational use of (La)TEX started with writing LaTEX bridge macros in 1990.[7] In 1995 as part of `BLUe.tex` the plain TEX variants became a tool in BLUe's toolbox. The macros are available in `BLUe.tex`'s `tools.dat`. The environment `\beginbridge` ...`\endbridge` selectively loads, behind the scenes and OS-independently, the macros from the `tools.dat` into your BLUe job. They can also be copied from the toolbox, manually, and used as a independent part, without BLUe, as I did for this paper.

6. For a wealth of examples see Practical PostScript—A guide to Digital Typesetting. David Byram–Wigfield. `http://www.cappella.demon.co.uk`, or John Deubert's `http://www.acumentraining.com/acumenjournal.html`.

7. Bridge is a card game and played with 52 cards: A K Q J T 9 ...3 2, each in the suits: , , and . There are 4 players around a table called North, East, South and West. N and S form a team, so do E and W. The cards are dealt, each receives 13 cards and then the auction starts. After the auction the playing of the cards begins. A game takes 5-7min.

Puzzle KQ76 6NT
J98 by East
J942
65

AJ3 T9
K653 A2
AK3 T5
AQT KJ9xxxx

W N E
S

8542
QT74
Q876
2

```
\beginbridge%loads bridge nacros
\def\LFTINF{Puzzle}
\def\RGTINF{\vtop{\hbox{6NT}
                  \hbox{by East}}}
\Ns={KQ76}\Es={T9}     \Ss={8542}\Ws={AJ3}
\Nh={J98} \Eh={A2}     \Sh={QT74}\Wh={K653}
\Nd={J942}\Ed={T5}     \Sd={Q876}\Wd={AK3}
\Nc={65}  \Ec={KJ9xxxx}\Sc={2}   \Wc={AQT}

\showgame
```

Trick					NS	EW
1	4!	K	8	2	–	1
2	A	5	x	2	–	2
3	Q	6	x	2	–	3
4	T	9	K	4	–	4
5	J	5	3	6	–	5
6	9	8	5	7	–	6
7	x	6	J	2	–	7

```
\LEADS\bplay
h4! & hK & h8 & h2 & --& 1\LEADW\cr
cA  & c5 & cx & c2 & --& 2\cr
cQ  & c6 & cx & s2 & --& 3\cr
cT  & h9 & cK & s4 & --& 4\LEADE\cr
cJ  & s5 & s3 & s6 & --& 5\cr
c9  & s8 & h5 & s7 & --& 6\cr
cx  & d6 & sJ & d2 & --& 7\cr
```

Puzzle KQ NS squeezed on
J continuation?
J94
–

A T9
63 A
AK3 T5
– x

W N E
S

–
QT7
Q87
–

8 x 7 6 J – 8
et cetera

```
\bintermezzo
\def\RGTINF{\vtop{\hbox to 0pt{NS squeezed on\hss}
            \hbox to 0pt{\cl\ continuation?\hss}
            \hbox to 7ex{\hss}}} %phantom for alignment
\Ec={{\ooalign{\hfil\raise.07ex%
    \hbox{x}\hfil\crcr\mathhexbox20D}}%mark lead
\showgame  \Ec={x}%restore E club
\eintermezzo
cx & h7 & h6 & hJ & --& 8\cr
\omit et cetera \hidewidth \cr
\eplay\endbridge
```

The `\bplay` ...`\eplay` table is interrupted by showing the status of the play, the remaining cards, between trick 7 and 8. Interesting is that data integrity has been strived after, because played cards have been removed from memory. Note the minimal mark-up: h just means hearts. At the time I did not know how TEX could include the `&`-s and `\cr`-s.

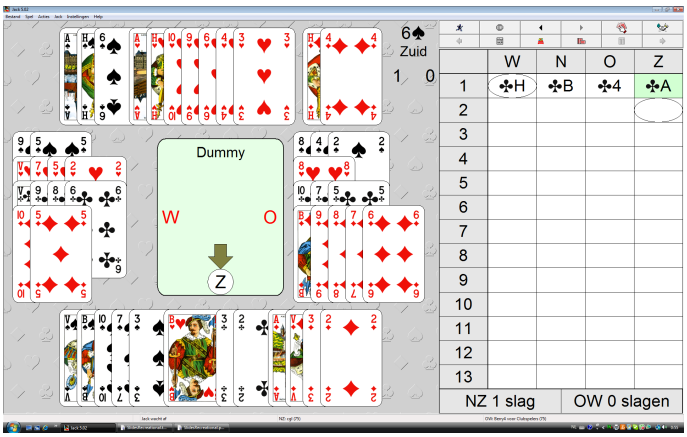
There is also an auction-environment. The example is borrowed from the table chapter of Publishing with TEX, PWT for short. In the LATEX Graphics Companion the LATEX bridge macros are mentioned, and some results have been shown.

From the macro-writing point of view, the dynamically declaration of token variables, as shown below, is interesting; a paradigm. The cards as sets and TEX-operations on sets is a paradigm too.

```
%\NT is alias of \newtoks without restricted use; \ea means \expandafter
\ea\let\ea\NT\csname newtoks\endcsname
\NT\Ns\NT\Es\NT\Ss\NT\Ws \NT\Nh\NT\Eh\NT\Sh\NT\Wh
\NT\Nd\NT\Ed\NT\Sd\NT\Wd \NT\Nc\NT\Ec\NT\Sc\NT\Wc \NT\hnd
```

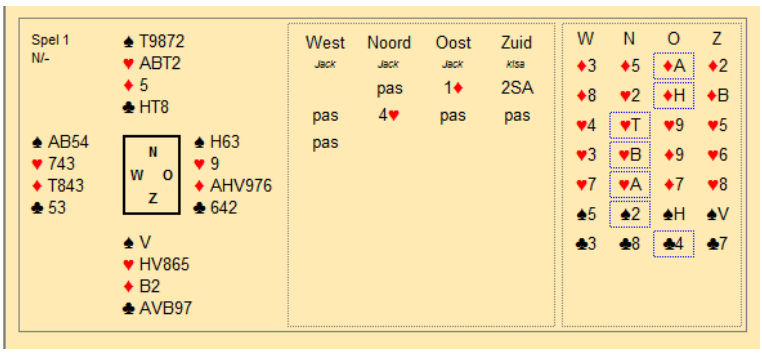
Computers and Bridge

In the past 20 years we have witnessed an enormous development, and increase in the use, of computers. In Bridge this has resulted in Bridge playing software such as the Dutch multiple Computer Bridge World-champion Jack.



Jack plays bridge

Characteristics: Data integrity, WYSIWYG input, Portable Bridge Notation standard, HTML export.



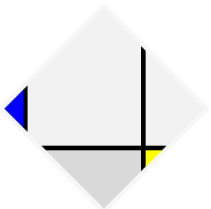
Jack bridge reporting

Do notice that typesetting is an aside for the developers of Jack.

The (recreational) play is assisted by Bridgemate (chipped-)boxes, which are used for the registration of the scores and are Wi-Fi coupled to the tournament directors computer for calculating the ranking. The results are put on the club's WWW page, made possible by the Nederlandse Bridge Bond, where the club members may find the scores and the ranking. All the hands played are usually also available, on the WWW and on an A4-sized paper.

Mondriaan inspired invitation

The Mondriaan (background) lozenge has been emulated in PostScript. The complete invitation after merging of the photograph and adding text, has been done in Photoshop. Emulated Mondriaan →

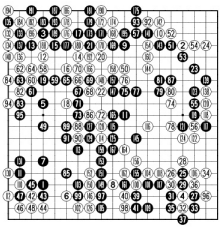


Other Games and TEX

Chapter 10 of the LaTEX Graphics Companion is devoted to Playing Games: Chess, Chinese Chess, Go, Backgammon, Card Games, Crosswords in various forms, and Sudokus.[8] In the sequel I'll mention what has been published in MAPS on the issue.

Hanna Kołodzieska has published in MAPS 7, p63–68, 1991, her 'Go diagrams with TEX.' She was inspired by Zalman Rubinstein, 'Chess printing via MetaFont and TEX,' TUGboat, 10, 2.
Piet Tutelaers has published in the (same) MAPS issue on the occasion of the NTG meeting about Games&TEX, 'A font and Style for Typesetting Chess using (La)TEX,' MAPS 7, p41-46.

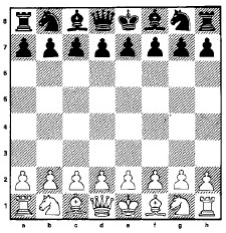
Go diagrams with TEX

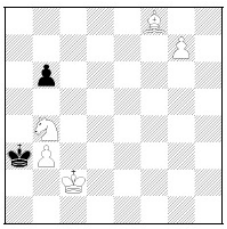


←Go situation

Chess board with pieces →

Chess position →→

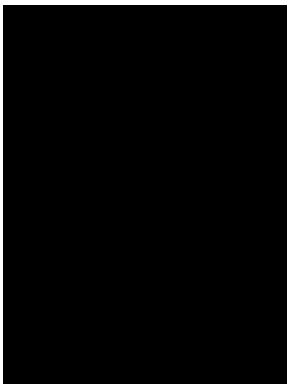




```
\board{ * * B *}
      { * * * P }
      { p * * *}
      { * * * * }
      { N * * *}
      {kP* * * }
      { *K* * *}
      { * * * * }
```

Computer chess is computer architecture encompassing hardware and software capable of playing chess autonomously without human guidance. Computer chess acts as solo entertainment (allowing players to practice and to better themselves when no human opponents are available), as aids to chess analysis, for computer chess competitions, and as research to provide insights into human cognition.

Chess-playing computers are now accessible to the average consumer. From the mid-70's to the present day, dedicated chess computers have been available for purchase. There are many chess engines such as Crafty, Fruit and GNU Chess that can be downloaded from the Internet for free. These engines are able to play a game that, when run on an up-to-date personal computer, can defeat most master players under tournament conditions. Top programs such as the Proprietary software programs Shredder or Fritz or the open source program Stockfish have surpassed even world champion calibre players at blitz and short time controls. In October 2008 Rybka was rated top in various rating lists and has won many recent official computer chess tournaments such as CCT 8 and 9, the 2006 Dutch Open Computer Championship, the 16th IPCCC, and the 15th World Computer Chess Championship. As of August 2012, Houdini is the top rated chess program on the IPON rating list with Rybka in 5th place.
Courtesy `http://en.wikipedia.org/wiki/Computer_Chess`.

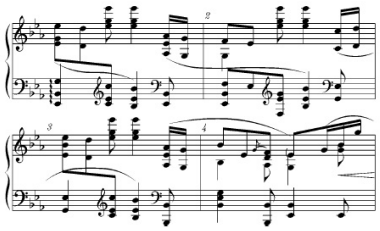


IPad Chess

8. Curiously Draughts is missing.

TEX and Music

Chapter 9 of the LaTEX Graphics Companion is devoted to Preparing Music Scores, and consists of 76p. It is hardly for fun. MusiXTEX of the late Daniel Taupin is leading. Too advanced and too difficult to be treated here, as can be witnessed from the various pre-processors to simplify the use.

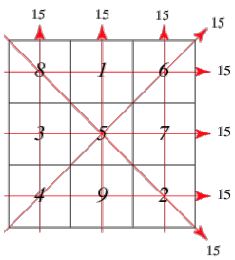


Graphics in Publishing with TEX, 1995

The graphics in PWT is limited, because the graphics is obtained by TEX alone. The gkp-macros have been used for PWT. These macros are limited due to the few discrete orientations of lines and there is no colouring.

Magic Squares recreational Math

A magic square of order n, is a square array of numbers consisting of the distinct positive integers 1, 2, ... n, arranged such that the sum of the numbers in any horizontal, vertical, or main diagonal line is always the same number, known as the magic constant $M_n = .5n(n^2+1)$.
Proof. Sum of all elements $\sum_{k=1}^{N} k = .5N(N+1),\ N = n^2$. One column, or row, sums up to $.5N(N+1)/n = .5n(n^2+1)$, the magic constant M_n.
In `http://en.wikipedia.org/wiki/Magic_square` curious, recreational Math algorithms are mentioned for squares of (double) even and odd n.



A 4x4 magic square puzzle is available at `http://www.dubster.com/math/`, where one can drag-and-drop the pieces; the magic constant is 30.

For odd squares the fun algorithm reads ... Starting from the central column of the first row with the number 1, the fundamental movement for filling the squares is diagonally up and right, one step at a time. If a filled square is encountered, one moves vertically down one square instead, then continuing as before. When a move would leave the square, it is wrapped around to the last row or first column, respectively.

```
- 1 -     - 1 -     - 1 -     - 1 -     - 1 -     - 1 6     - 1 6     8 1 6     8 1 6
- - - →   - - - →   3 - - →   3 - - →   3 5 - →   3 5 - →   3 5 7 →   3 5 7 →   3 5 7
- - -     - - 2     - - 2     4 - 2     4 - -     4 - 2     4 - 2     4 - 2     4 9 2
```

The permutation array algorithm, as mentioned above, is implemented as a PS-snippet for `ms3x3` as follows[9]

```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Magic Square of order 3. %Permutation array algorithm as given in Wikipedia
%%Creator: Kees van der Laan, okt 2012
%%BoundingBox: 0 0 24 36
%%BeginSetup
%%EndSetup
/Times-Roman 12 selectfont
/p [0 8 1 6
      3 5 7
```

9. `ms3x3`, `ms4x4` and `ms5x5` are included in `PSlib.eps`.

```
      4 9 2] def%0th entry dummy
0 1 2{/i exch def 0 25 i 12 mul sub moveto
      1 1 3{/j exch def p i 3 mul j add get ( ) cvs show 2 0 rmoveto}for
     }for showpage
%%EOF
```

In PSlib I have included a branch-and-bound, backtracking algorithm for order 3.

8 1 6
3 5 7
4 9 2

The finding of symmetrical copies in the branch-and-bound algorithm is suppressed by fixing the middle above element on 1 and restricting the loop variables. The unrestricted code took 38sec and restricted 8sec in Acrobat Pro. PSview took 3sec for the restricted version. The number of magic squares for n=1 is 1, for n=2 there is no magic square and for n=3, 4 and 5 see the accompanying table. Programming the Magic square is as instructive as programming the 8-Queens problem. For the latter see Wirth, N(1976): Algorithms + Datastructures = Programs, p143. Programming Magic squares yields extra Math insight.

n	M_n	Number
3	15	1
4	34	880
5	65	275,305,224

For double-even squares the fun algorithm reads ... All the numbers are written in order from left to right across each row in turn, starting from the top left corner. Numbers are then either retained in the same place or interchanged with their diametrically opposite numbers. In the magic square of order four, the numbers in the four central squares and one square at each corner are retained in the same place and the others are interchanged with their diametrically opposite numbers.

$$\begin{array}{cccc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array} \rightarrow \begin{array}{cccc} 1 & 15 & 14 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 3 & 2 & 16 \end{array} \rightarrow \begin{array}{cccc} 1 & 15 & 14 & 4 \\ 12 & 6 & 7 & 9 \\ 8 & 10 & 11 & 5 \\ 13 & 3 & 2 & 16 \end{array}$$

The Magic square of Dürer shows more than the usual properties: also the four quadrants add up to the magical constant 34. By adding up 2 to each cell the magic constant becomes 42, the answer to the question of 'Life, Universe, and Everything.'[10]

16 3 2 13
5 10 11 8
9 6 7 12
4 15 14 1

```
\oldstyle
\halign{\quad\hfil#\hfil&&\quad\hfil#\hfil\cr
        16& 3&  2& 13\cr
        5& 10& 11&  8\cr
        9&  6&  7& 12\cr
        4& 15& 14&  1\cr}
```

In PWT the \btable macro was used with flexibility with respect to the frame and the horizontal and vertical lines. Syntactic sugar?

Frans Goddijn calls \oldstyle numbers 'dartele cijfertjes.'[11] A nice Dutch word, dartel.

Knuth's most beautiful tables

Knuth's useful and most beautifully structured and parametrized mark-up of font tables is worth studying. Knuth's macros have been incorporated in BLUe.tex. In the Metafont book in App H a similar but interactive program testfont.tex is available and when TEXlive has been installed one can just say \inputtestfont.tex.

10. Adams, D(1982): The Hitchhiker's guide to the Galaxy. Pan Books.
11. Goddijn, F(1998): Dartele cijfers: poor man's oldstyle, MAPS 20.

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	"0x
´01x	Φ	Ψ	Ω	ff	fi	fl	ffi	ffl	
´02x	ı	ȷ	`	´	ˇ	˘	¯	°	"1x
´03x	¸	ß	æ	œ	ø	Æ	Œ	Ø	
´04x	‘	!	”	#	$	%	&	’	"2x
´05x	(	)	*	+	,	-	.	/	
´06x	0	1	2	3	4	5	6	7	"3x
´07x	8	9	:	;	¡	=	¿	?	
´10x	@	A	B	C	D	E	F	G	"4x
´11x	H	I	J	K	L	M	N	O	
´12x	P	Q	R	S	T	U	V	W	"5x
´13x	X	Y	Z	[	“	]	^	˙	
´14x	‘	a	b	c	d	e	f	g	"6x
´15x	h	i	j	k	l	m	n	o	
´16x	p	q	r	s	t	u	v	w	"7x
´17x	x	y	z	–	—	”	~	¨	
	"8	"9	"A	"B	"C	"D	"E	"F	

For use with pdfTEX the mark-up reads as given below. LaTEX and ConTEXt users are not aware of this mark-up, I presume, but they might benefit from it.

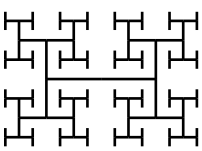
```
\input blue.tex
\beginchart{\postdisplaypenalty=0
\tenrm}
%or \tenit ... \tenlmr?
\normalchart
\endchart
\bye

%or simply
\input testfont.tex
%a prompt for font name
%appears: type cmr10 f.e.
\table
\bye
```

Font tables have been supplied in the TEXbook, Appendix F. In Appendix H of the Metafont book.`testfont.tex` is discussed[12]

I was curious how I could obtain a font table for Latin Modern Roman. Hans Hagen prompted `\starttext\showfont[lmroman10regular][all]\stoptext`, which I processed under Context(LuaTeX) in TEXworks. Pane 1 of the table is supplied in the 3rd Appendix.

H-fractal from PWT

Earlier I remarked that the binary tree, the H-fractal and Adobe's `FractArrow`, Bluebook, p.74, are closely related, one just has to adapt the invoke by the appropriate angle.



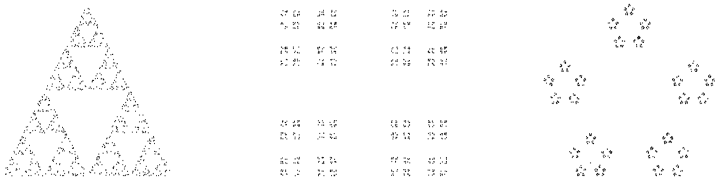
In `BLUe.tex` I implemented the Turtle Graphics approach. The H-fractal was programmed recursively and supplied as exercise 5.3 in PWT. Apart from pictures generated on-the-fly, pictures are provided in `pic.dat`, the picture-base of TEX-alone pictures which comes with `BLUe.tex`.

Compared with programming in PostScript the coding of a TEX-alone picture is cripple, without the possibility to crop the result, to include BoundingBox values for pdfTEX. There is no need to include the H-fractal `gkp`-codes and PS-codes here; they have been included in the EuroTEX-ConTEXt2009 proceedings. I also mentioned there the notches, the absence of appropriate line-endings in TEX. TEX is the wrong tool for graphics, definitely. But ... in cooperation with Metafont artistic effects can be obtained, as was done by Jackowski&Ryćko in the early 90s. For simple, quick-and-dirty, line-drawings TEX might do.

Iterated Function System fractals from PWT

In 1989 I attended the TUG conference where Alan Hoenig showed some iterated function system fractals,[13] which I reproduced in PWT. The idea is that the points within an n-gon are created by: the mean of a random point within the n-gon and one of its corners at random, à la Monte Carlo. A random number generator for plain TEX had to be written. The representation of the corner points is tricky via `\newdimen`-variables, in order to perform the arithmetic. Too much details in order to be presented here.

12. Lueking, D(2010): How to use `fntproof.tex` and `testfont.tex` (from the WWW).
13. His paper has been published in TUGboat 1989. For iterated function systems and fractals, see Peitgen c.s. (2004 2nd ed): Chaos and Fractals. Springer. No sophisticated Math is required for reading the book.



Pascal triangle from PWT

The table chapter of PWT contains the Pascal triangle. The triangle shows the binomial coefficients $\binom{n}{k}$.
If the values of $\binom{n}{k}$ are available, the typesetting is a trifle via the use of `\displaylines`, TEXbook, p362.
The values $\binom{n}{k}$ can be generated on-the-fly by the recursion

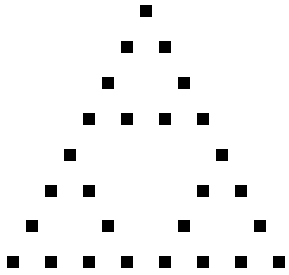
$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}, \quad n = 2, ..., \quad k = 1, ..., n-1, \quad \binom{n}{0} = \binom{n}{n} = 1$$

which has been used in the code as shown in the verbatim text at right below. The intriguing macros make use of recursion for calculating each element in a row. Each row is overwritten in `\1,\2,...` which also entails that each 'row' is extended dynamically. This reminds me of the dynamic array functionality.
Paradigm: a counter-value becomes a control sequence name to denote the position in the row with as value the binomial coefficient.

```
$$\displaylines{1\cr
1 \quad 1\cr
...
1 \quad 9 ... 9 \quad 1\cr}$$
```

```
\newcount\n \newcount\rcnt \newcount\ccnt \newcount\tableentry \newcount\prev
%
\def\pascal#1{\n#1 \def\0{1}                                    %presets
   \ccnt1 \loop\expandafter\xdef\csname\the\ccnt\endcsname{0}
   \ifnum\ccnt<\n \advance\ccnt1
   \repeat  \rcnt0 \ccnt0 \displaylines{\rows}}
%
\def\rows{\global\advance\rcnt1 \ifnum\rcnt>\n \swor\fi \nxtrow\rows}
\def\swor#1\rows{\fi}
%
\def\nxtrow{1 \ccnt1 \prev1
 \loop\ifnum\ccnt<\rcnt \tableentry\prev \prev\csname\the\ccnt\endcsname
 \advance\tableentry\prev
 \expandafter\xdef\csname\the\ccnt\endcsname{\the\tableentry} %the new entry
  \quad\the\tableentry \advance\ccnt1                        %show the entry
 \repeat\cr}
$$\pascal{8}$$
```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1



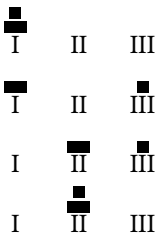
In the picture at right a variant of the PACAL triangle has been shown, where the odd-valued entries are coloured black and the even-valued entries are left blank, which reminds me of the Sierpiński triangle. The macros and pictures have been submitted to GUST's Programming Pearls 2012.

Towers of Hanoi play from PWT

In general BLUe has as top level minimal one-part macros and tries to circumvent the curly braces mania: no curly braces around arguments! Invoke: `\Hanoi\I\II\III\n`,

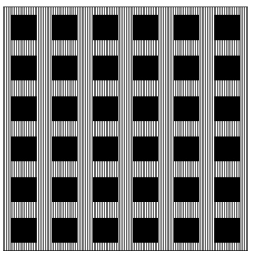
where the capital Roman numerals denote the towers. The one-part macro invokes the two-part macros, the environment. The Hanoi macros are available within the `\beginhanoi...\endhanoi` environment.

The process of replacement of the disks will also be printed by the shortened invoke of the one-part macro `\sethanoi<n>`, `<n>` an integer, the height of the initial tower. The intermediate stages will be shown, no user mark-up is needed.
The Hanoi-tool has been copied from BLUe's `tools.dat` and is used stand-alone in this paper to reproduce the results.
Paradigm: the use of a hidden loop counter. The loop counter is dynamically created in `\preloop`; the user is not bothered by it.



Soto's Op Art from PWT

A verbose version of Soto's Op Art[14] emulation was written originally in Metafont in 1995. For the EuroTEX-ConTEXt2009 the picture was redone in concise plain TEX, with the use of TEX's `\leaders`, `\xleaders` and the reuse of `\setbox`-es. A gkp-macro version appeared earlier in PWT. On occasion of this conference a simpler, mean-and-lean PS-variant has been written.



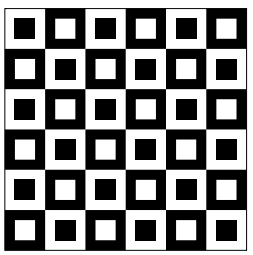
```
\def\boxit#1{\vbox{\hrule
          \hbox{\vrule#1\vrule}\hrule}}
\newbox\cb  \newdimen\ul \ul=6ex
\newdimen\size \size12\ul
\setbox\cb\vbox to2\ul{\vss
    \hbox to2\ul{\hss\vrule height1.2\ul width1.2\ul
                 \hss}%
                      \vss}%
$$\boxit{\vbox{\offinterlineskip
\hbox{\xleaders\hbox to.5ex{\hss\vrule height\size
                           \hss}\hskip\size}%
\kern-\size%setback
\leaders\hbox{\leaders\copy\cb\hskip\size}\vskip\size}}$$
```

```
%!PS-Adobe
...
gsave .25 setlinewidth
57{1 0 translate
   0 0 moveto 0 57 lineto}repeat
stroke
grestore
0 0 57 57 rectstroke
3 3 translate
6{gsave
  6{0 0 6 6 rectfill
    9 0 translate}repeat
  grestore
  0 9 translate}repeat
```

In TEX we strive after efficiency by using `\setbox`-es, such that repetitive material is only typeset once and reused by copying; in PS there is the ucache concept. My PS-graphics is small and fast enough.

Jiggling squares from PWT

An old example, which I did in TEX, see EuroTEX-ConTEXt2009, and on occasion of EuroTEX-ConTEXt2012 in PostScript, in a split-second.



```
%!PS-Adobe
...
gsave .1 setlinewidth 0 0 120 120 rectstroke grestore
/Oc{0 0 moveto 20 0 rlineto 0 20 rlineto -20 0 rlineto closepath}def%outer contour
/Ic{5 5 moveto 0 10 rlineto 10 0 rlineto 0 -10 rlineto closepath}def%inner contour
3{gsave 3{Oc Ic fill 20 0 translate
          Ic    fill 20 0 translate}repeat
 grestore
 gsave 0 20 translate
        3{Ic fill    20 0 translate
          Oc Ic fill 20 0 translate}repeat
 grestore 0 40 translate}repeat
showpage
```

14. Jesús Rafael Soto, 1923–2005, was a Venezuelan Op and Kinetic artist, a sculptor and a painter. Soto has created penetrables, interactive sculptures which consist of square arrays of thin, dangling tubes through which observers can walk. It has been said of Soto's art that it is inseparable from the viewer; it can only stand completed in the illusion perceived by the mind as a result of observing the piece. http://www.wikipedia.org/wiki/Jesus-Rafael_Soto.

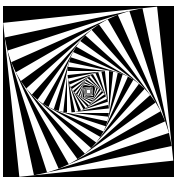
Do realize the use of integers only in the model, otherwise rounding errors spoil the strict regularity. No lines are drawn, except for the border, just `fill`s filled by the non-zero winding rule. Pitfalls for the unwary.
One element of the tile is a square O, programmed by a `0c Ic fill`. How is the O programmed in the Metafont book? From p303

```
beginlogochar("O",15); x1=x4=.5w; top y1=h+o; bot y4=-o; x2=w-x3=good.x(1.5u+s);  y2=y3=barheight;
                       super_half(2,1,3); super_half(2,4,3); labels(1,2,3,4); endchar;
```

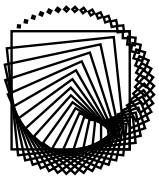
Remarkable is that 2 halves of a superellipse have been used, not just the complete inner and outer contours, apparently for consistency with the upper half of the letter A. On p32 the character O has been drawn by the use of `penstroke`.

Shrinking squares

At last I traced the origin of the left ubiquitous illustration, which is about drawing just squares in transformed user space: Barnsley, M.F(1988): Fractals Everywhere. It is used in Ch. 3.6 to illustrate the idea of a contractive transformation on a compact metric space.



```
%!PS-Adobe-3.0 EPSF-3.0
...
/r 50 def /2r {r r add} def /-r {r neg} def
/alpha 6 def /c 1 alpha cos alpha sin add div def
/square{-r -r 2r 2r rectfill}def
0 1 45{2 mod setgray square
       /r r c mul def alpha rotate}for
```



The right picture is an intriguing variant; nearly the same code.
Paradigm: change of black and white in traversals of the loop.

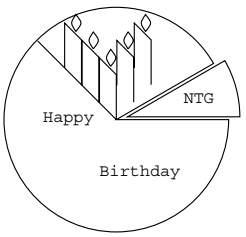
Malevich suppremátism

Malevich Suprematism:
White cross on a White background
Emulation →

I have included a picture, and its emulation, of Malevich's[15] 'White Cross on a white background,' because he is the father of suprematism, which deletes the superfluous, which I associate with Minimal Mark-up. But ... sometimes redundancy is beneficial.

Happy birthday NTG from PWT

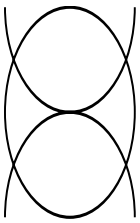
This cake I produced on occasion of the first lustrum of NTG. The original version made use of LATEX's picture environment and does no longer work on my system since I abandoned LATEX. The 2nd version was done with the gkp-macros.[16] At the time I experienced drawing a circle by splines as difficult.[17] The 3rd version of the picture is in PostScript, shown at right, is a trifle and took me a couple of minutes.



15. Kazimir Malevich, 1878–1935. Russian painter, born in Kiev.
16. I should make it priority 1 to get `BLUe.tex` running again as a format in TEXworks.
17. For the solution à la Knuth, see the Metafont book p263, or Appendix 1 in Gabo's Torsion, MAPS 42, 2011.

Logo from PWT and a logo from MetaFun

The logo was created by the gkp-macros. The current version in PostScript took me just a minute. The proportions obey the golden ratio, realized by scaling. At right a logo borrowed from MetaFun.



```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: cgllogo, 2012
%%BoundingBox: -63 -101 63 101
%%BeginSetup
%%EndSetup
/r 100 def /-r r neg def 3 setlinewidth .618 1 scale
r 0 moveto  0 0 r 0 360 arc
2{r -r moveto 0 -r r 0  180 arc 1 -1 scale }repeat
stroke
```

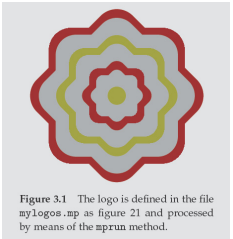
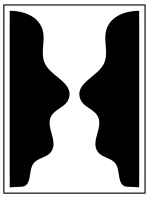


Figure 3.1 The logo is defined in the file mylogos.mp as figure 21 and processed by means of the mprun method.

Profiles or a candle?



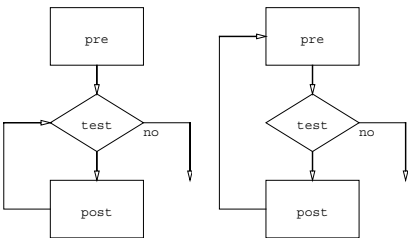
```
size:=210;path p[];
p1=(-1.1size,-1.5size){right}---(-.9size,-1.49size)
    ..(-.7size,-1.1size)..(-.4size,-.9size)..
   (-.4size,-.3size)..(-.1size,0)..(-.5size,.6size)..(-.55size,1.2size)...
   {left}(-1.1size,1.5size)---cycle;
p2= p1 reflectedabout ((0,-size),(0,size));
fill p1; fill p2;
draw (-1.2size,-1.6size)---(1.2size,-1.6size)---(1.2size,1.6size)---
     (-1.2size,1.6size)---cycle;
```

The Metafont code of 1996 was converted into MetaPost, well simply stripped from Metafont's necessities such as screen settings, `cullit`, `show`, ..., and dropped onto Troy Henderson's `mppreviewer` to yield `.png`, on occasion of EuroTEX-ConTEXt2012.[18] Happily, I saved `profile` from disappearance.

Flowchart from PWT

In PWT the flowchart of TEX's loop was made within LATEX's picture-environment and later converted into plain TEX where the picture was created by the gkp-macros. On the occasion of EuroTEX-ConTEXt2009 the flowchart was created in MetaPost by means of Hobby's `boxes` macros, and resulted in a PostScript program created by MP. The right picture mimics TEX's loop.[19] At right my PostScript code on occasion of EuroTEX-ConTEXt2012. No `boxes` macros are needed; the coding is equally simple, or equally difficult, depending on your expertise, as the MetaPost code.
The recent picture, with golden ratio proportions, took me 45min to create in PostScript, which is too long for a production tool. The creation of the previous loop-pictures took me at least as long, if not longer ☹. Use is made of `rectstroke`, `centershow` and Adobe's Bluebook `arrow`. Ça va sans dire that the direct PostScript program is much shorter than the PostScript code which resulted from MetaPost.

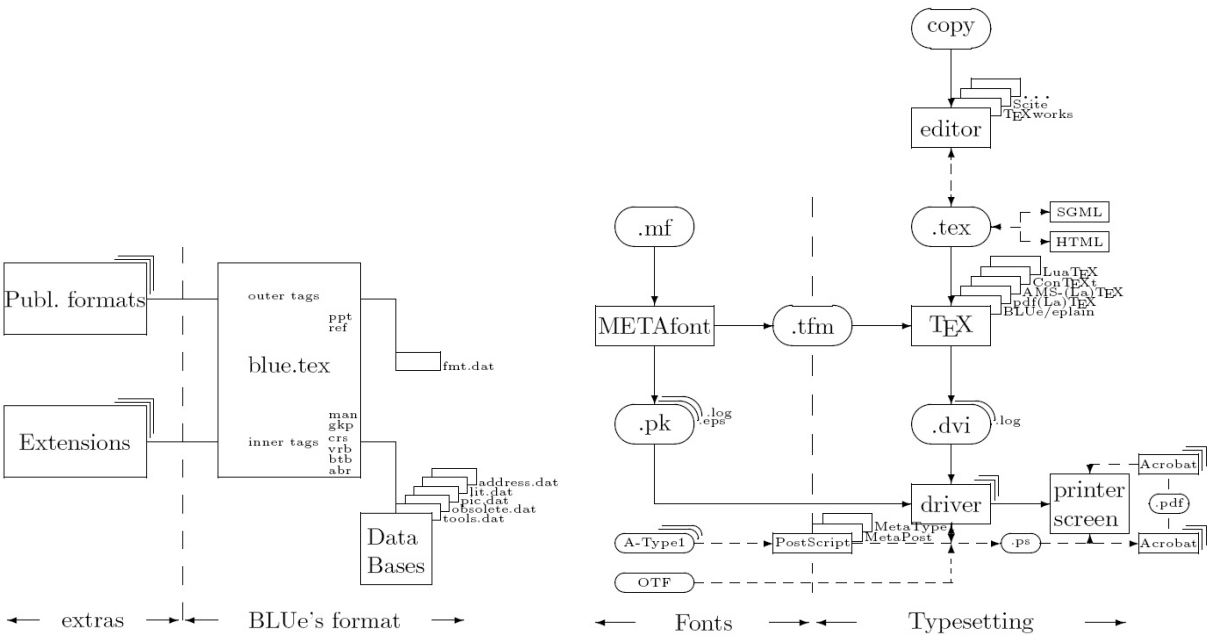
18. I used Hans Hagen's pair `\startuseMPgraphic{dummy}`, `\stopuseMPgraphic` and `\useMPgraphic{dummy}` and got results by ConTEXt(LuaTEX) in TEXworks. I no longer need Henderson's MPpreviewer. Troy has also provided a LaTEX and a function-grapher previewer.
19. In contrast with the usual implementations, `while` respectively `repeat ... until`, where the test is performed at the beginning respectively at the end, the test for termination in TEX's loop can be placed at a place at will within the loop. The same is possible in PostScript within the `loop` procedure, where termination goes via the invoke of `exit` (for the inner loop). I consider the implementation of TEX's loop ingenious.




```
/Courier 14 selectfont
/s 50 def  /t s .61803 mul def %half width and halh height
/2s s s add def/2t t t add def/3t 2t t add def /6t 6 t mul def
/-s s neg def /-t t neg def /-2s 2s neg def /-2t 2t neg def
%bottom rectangle
-s -t 2s 2t rectstroke
0 -7 moveto (post) centershow
-s 0 moveto -2s 0 lineto 0 6t rlineto
-2s 6t -s 6t .5 5 10 arrow stroke                          %top-down arrow
%diamond
0 3t translate
-s 0 moveto 0 t lineto s 0 lineto 0 -t lineto closepath stroke%diamond
0 -7 moveto (test) centershow
0 2t 0 t .5 5 10 arrow stroke                              %top-down arrow
0 -t 0 -2t .5 5 10 arrow stroke                            %arrow on the
left
s 0 moveto 2s 0 lineto 2s 0 2s -2t .5 5 10 arrow  stroke   %arrow on the
right
s -14 moveto (no) show
%top rectangle
0 3t translate -s -t 2s 2t rectstroke 0 -7 moveto (pre) centershow
```

The `flowchartloop def` is included in `PSlib.eps`. Do compare the three generations of code: based on the picture environment, MetaPost (both listed in the EuroTEX-ConTEXt2009 paper), and the PostScript code given here. My conclusion is, and was, that PostScript can equally-well be used directly in a 1-pass job, most of the times.

Below my most complex TEX-made flowcharts of old. The TEX-MF picture has been adapted for this conference.



The above TEX-flow is nowadays practically integrated in simple to use IDE's, such as TEXworks, where the various processing modes, such as pdfTEX, can be chosen from pull-down menus.[20]

TEXworks shows 3 panels: the edit panel with the source `.tex`, the result panel with `.pdf`, and the processing window with the process report and the error messages,

20. Blue Sky provided 20 years ago similar functionalities in its TEXtures for the Macintosh.

eventually. A form of WYSIWYG. Another pull-down menu in the edit panel lets you choose your font and the use of spelling checkers. There is also a script option. In the help menu there is an option for the 'Short manual for TEXworks' by Alan Delmotte, Stefan Löffler, and others.

Font Fun in TEX&Co

Though TEX's CM-fonts are bitmapped and rigged, occasionally recreational effects have been obtained.

Dancing texts by PostScript

Hans Hagen in his `MetaFun` inspired me to think about dancing texts. In PostScript the effect can be obtained by the use of `kshow`, where the procedure as argument of `kshow` takes care of (slightly) rotating user space for each character. `Nrand` delivers a random number $\in [0, 1)$, and `unirand` a random number $\in (-1, 1)$, both from `PSlib.eps`. The colours are composed randomly. In Photoshop dancing-like texts can be obtained by typesetting along a sine-curve. The picture at right is by Emil Nolde.[21]
Avoid in PostScript the trap to create a font variant.
A nice application for children's party invitation cards.

Kees van der Laan

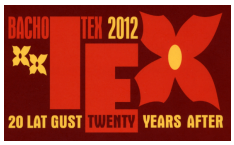
```
07101951 srand
/Helvetica 35 selectfont 0 0 moveto 1 0 0 setrgbcolor
{pop pop unirand 4 mul rotate nrand nrand nrand setrgbcolor}
(Kees van der Laan) kshow %a paradigm
```

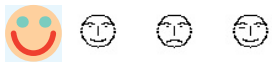
21. Emil Nolde, 7 August 1867 Near Nolde (Denmark) – 13 April 1956 Seebrüke, was a German painter and printmaker. He was one of the first Expressionists, a member of Die Brücke, and is considered to be one of the great oil painting and watercolour painters of the 20th century. He is known for his vigorous brushwork and expressive choice of colours. Golden yellows and deep reds appear frequently in his work, giving a luminous quality to otherwise somber tones. His watercolors include vivid, brooding storm-scapes and brilliant florals. *'There is silver blue, sky blue and thunder blue. Every colour holds within it a soul, which makes me happy or repels me, and which acts as a stimulus. To a person who has no art in him, colours are colours, tones tones...and that is all.'*

Font Fun in TEX

The classical example is the word TEX, with dropped E. Another classic is XƎTEX which can't be done in TEX-alone. With dvips the mirroring can be done at the PS-level, but alas pdfTEX does not allow for PS. NTG's first logo was 'Nederlandse TEX Gebruikersgroep,' which was soon changed, on the way to the TUG meeting at Karlsruhe in discussion with Johannes Braams, into 'Nederlandstalige TEX Gebruikersgroep,' meaning Dutch-language based.

Nederlandstalige TEX Gebruikersgroep



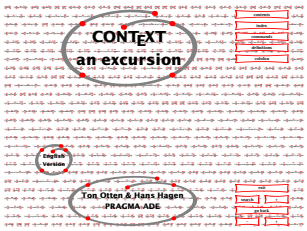


The coloured smiley is in PostScript; the others are done in TEX by dots; very cripple TEX programming.

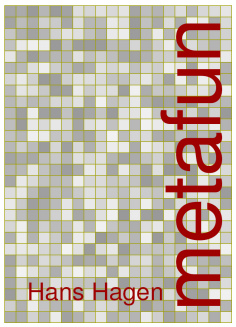
←EuroTEXConTEXt2012 logo

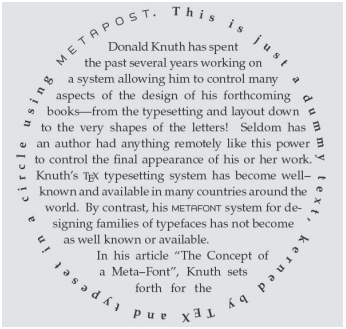
Toruń98 logo, by B. Jackowski→

The TEX-lion and the MF-cat, by Duane Bibby, the running illustrations in the TEXbook and the MetaFontbook, made the books a pleasure to read. The EuroTEX-ConTEXt2012 logo is nice and fun.



Hans Hagen's MetaFun





Don Lancaster's Font Fun

In the late 1970s TEX appeared with rigid, bitmap CM font families. In the mid 1980s Adobe developed scalable, and adaptable PostScript fonts, thanks to the font matrix concept.[22] Lancaster played 20 years ago with PostScript's font variability. In his `pssecrets`[23] he showed various font variations. These modified fonts can be used in PostScript, which is fair enough. Adobe Type 1, with `afm2tfm` for conversion of the metrics, can be used in TEX, although this route becomes more and more outdated in view of that Adobe has Adobe Type 1 declared obsolete and in view of unicode and the TEX-Gyre OTF-project, next to the incorporation of Open Type Fonts in the new TEX-engines LuaTEX, or XeTEX.[24]

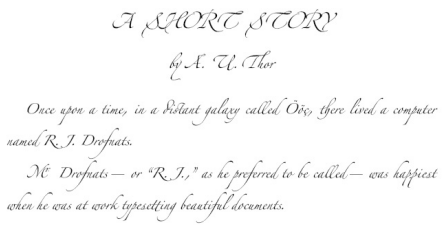
reverse emboss shiny 3D

The tiny PostScript program for the shadowfont is not at all difficult and demonstrates the use of the font matrix, TFM for short.[25] The reverse font is straightforward too with TFM: e.g. `[-40 0 0 40 0 0]`.

rotated boxit shadow

```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Shadow font, Don Lancaster, 1990
%%BoundingBox: -1 -25 180 30
%%BeginSetup
%%EndSetup
.8 setgray  /msg (Free font) def
/Palatino-Bold findfont [40 0 32 -30 0 0] makefont
setfont
 0 0 moveto msg show%shadow
 0 setgray /Palatino-Bold 40 selectfont
 0 0 moveto msg show
```

From the XeTEX showcase



A SHORT STORY

by A. U. Thor

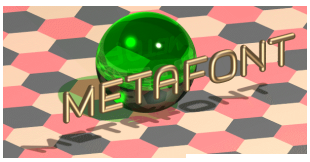
Once upon a time, in a distant galaxy called Ööç, there lived a computer named R. J. Drofnats.

Mr. Drofnats—or "R. J.," as he preferred to be called—was happiest when he was at work typesetting beautiful documents.

```
\font\body="Zapfino" at 10pt \body
\font\title="Zapfino:Stylistic Variants=First variant glyph set" at 12pt
\font\author="Zapfino:Stylistic Variants=Second variant glyph set" at 10pt
\centerline{\title A \ SHORT \ STORY}
\vskip 6pt
\centerline{\author by A. U. Thor}
\vskip .5cm
Once upon a time, in a distant galaxy called Ööç,
there lived a computer named R.~J. Drofnats.

Mr.~Drofnats---or ``R. J.,'' as he preferred to be called---was happiest
when he was at work typesetting beautiful documents.
\bye
```

Metafont&TEX can be used to create beautiful artistic results with fonts as has been shown in the 90s by Bogusław Jackowski and Marek Ryćko. Non-scalability is not relevant for pieces of art.

22. Making outline fonts from TEX's CM fonts is not simple, while outline variants of PS fonts are a trifle. Using font outlines for clipping is fun in PS. In TEX I don't know how to do it.
23. `http://wwwtinaja.com/glib/pssecrets.pdf`. The layout of his tiny programs is horrible. The ones I copied I have simplified.
24. Veith, U, M. Miklavec(2012): Another incarnation of Lucida: Towards Lucide OpenType. BachoTEX2012 proceedings, 5–13. Ludwichowski, this proceedings.
25. The font matrix is specified by 6 digits between square brackets, similar to the general transformation matrix, TFM for short, of PostScript.



Word Art in MS-Word

My Word 7 comes with `Word Art` options and the user can play with the appearance of texts.
My wife Svetlana Morozova tipped me about the font fun in Word. In Photoshop similar effects can be obtained. TEX's bitmap CM fonts are too rigid for fun.


1919 — 9 Avril
Salle Kaufleuten
NON PLUS ULTRA
9. DadA-SOIRÉE
Metteur en scène: W. SERNER
Dompteur des acrobates: TZARA
*) Mouvement Dada, Zurich, Seehof, Schifflande 28.
Non PLUS ultra

Font outlines

In TEX and MetaPost the creation, and the clipping use, of an outline of a glyph is not possible. In PostScript it is part of the orthogonal philosophy: any path, including the character path left by a `charpath` operator, can be used as a clipping outline boundary.[26] The `def o(utline)show`, with on the stack a `(string)`, reads as follows `/oshow{true charpath stroke}def`.
`Clipping` of an outline path may yield interesting effects. The example is borrowed from the Bluebook p103.


```
/Times-Roman 50 selectfont .25 setlinewidth
/rays{120{0 0 moveto 108 0 lineto
      1.5 rotate}repeat stroke}def
0 0  moveto (StarLines) true charpath clip
newpath 100 -15 translate rays
```

Carving

Another nice example in Hans' `MetaFun` is the tallying of data. I imitated his ConTEXt–MetaPost table example. My tallying is done in PostScript, see the code below at right, and the table is set via `\halign`.

System	%	Users
Atari	10.4	𝍸 𝍸 |
MS-DOS	49.1	𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 ||
OS/2	9.4	𝍸 𝍸
MacOS	5.7	𝍸 |
UNIX	51.9	𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸
Windows	64.2	𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 |||

```
/tally{/n exch def 0 0 moveto
1 1 n{dup 5 mod 0 eq{-8 0 rmoveto /d 10 nrand sub def
                      7 d rlineto  4 d neg  rmoveto}
                     {/r unirand 3 mul def
                       r rotate 0 10 rlineto
                       r neg rotate
                       2 -10 rmoveto} ifelse
}bind for
}def
```

Hans' 1-pass job has much in favour. I like, of course, my cooperating tools approach. I don't have to remember the philosophy and details of ConTEXt, MetaPost, nor Metafun; just good old plain TEX and PostScript. The tallying macro and the dancing text were written on occasion of EuroTEX-ConTEXt2012.

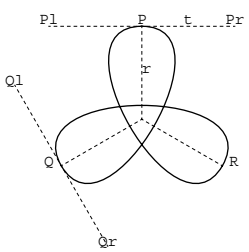
26. This works only for characters which are defined as outlines.

Escher knot

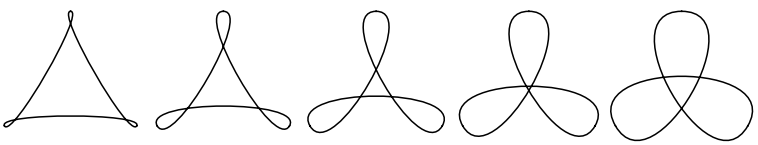
The Escher knot was programmed in Metafont and MetaPost. It marks my beginning of using Metafont/Post as declarative graphical languages. From the latter program the spline data were distilled and inserted in the tiny PostScript program below, with the number of fractional decimals rounded to 2. The gradient colouring has been done in Photoshop by my wife Svetlana Morozova on occasion of the EuroTEX-ConTEXt2009.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -40 -31 40 43
%%Creator: MetaPost and JJW, CGL, June 1996
%%BeginSetup
%%EndSetup
3{-21.65 12.5 moveto
-21.65 27.75 -13.78 42.50  0 42. 50    curveto
 13.78 42.50  21.65 27.75 21.65  12.5  curveto
 21.65 -0.24 16.05 -12.19  6.58 -20.33 curveto
-14.32 15.87 moveto
-12.43 23.59 -7.45 29.75  0   29.75 curveto
  9.65 29.75 15.16 19.42 15.16 8.75 curveto
 15.16 -2.08 9.38 -12.09  0  -17.5  curveto
120 rotate
}repeat stroke
```

While pondering about the Escher knot another solution came to mind for the single knot in PostScript.




```
/Courier 12 selectfont
/r 70 def /t 70 def
/P{0 r}def /Q{P 120 rot}def /R{P -120 rot}def
/Pr{t r}def /Pl{t neg r}def /Qr{Pl 120 rot}def
gsave .1 setlinewidth [2 3] 11 setdash
   3{0 0 moveto P lineto 120 rotate}repeat stroke grestore
P 2 add moveto (P) centershow
Q exch 13 sub exch moveto (Q) show
R exch 5 add exch moveto (R) show
3{P moveto Pr Qr Q curveto 120 rotate} repeat stroke
```



An alike of the third figure in MP reads

```
beginfig(59)
draw (0,u)      {right}   .. tension 4..
     (u*dir-150){dir 120} .. tension 4..
     (u*dir-30) {dir/120} .. tension 4..
     cycle
endfig;
```

The variability by `r` and `t` seems sufficient.[27] The 'tube' version is complicated by hidden lines, which were gracefully handled in MetaPost in the EuroTEX-ConTEXt2009 paper, by use of `cutbefore` and `cutafter`. The single knot version has less graceful curves. The shape can be adapted by changing `r` and/or `t`. It looks like that Metafont's `tension` functionality is not needed. BTW, I much prefer for a curve in Metafont's lingo `z0..controls z1 and z2..z3`, more in accordance with PS's `curveto` and the Math formula $\sum_{i=0}^{3}(1-t)^{3-i}t^i z_i$, without the strange unusual notions `tension` and `curl`.

In PostScript there is no path data-structure and no `def` for calculating the intersection point of 2 B-cubics. It is curious that PostScript does not contain an evaluation procedure for points on a spline. The 'de Casteljau' algorithm for evaluation is nothing more than fixing precedence of operations by parentheses

27. Dennis Roegel has published many articles on MetaPost.

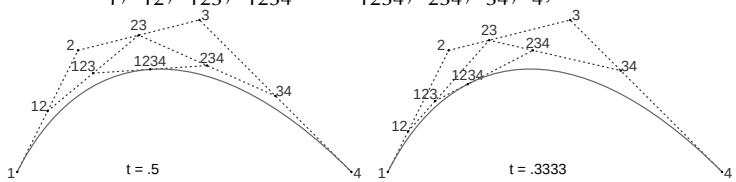
$$z(t)=\sum_{i=0}^{3} z_i(1-t)^{3-i}t^i=(1-t)\Big((1-t)\Big((1-t)z_0+t\,z_1\Big)+t\,\Big((1-t)z_1+t\,z_2\Big)\Big)+$$
$$+t\Big((1-t)\Big((1-t)z_1+t\,z_2\Big)+t\,\Big((1-t)z_2+t\,z_3\Big)\Big).$$

Together with my `solveit` it should not be difficult to write a `def splineintersection`.

Not recreational, pretty serious. Maybe, someday, sometime ...
Knuth in the MetaFont book, p13, gives a graphical representation of the evaluation, mediation algorithm

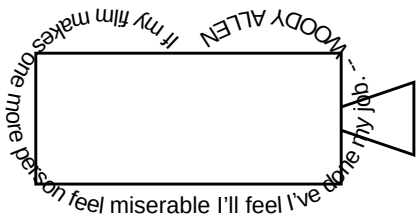
$$\begin{matrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{matrix} \begin{matrix} \\ \rightarrow \\ \\ \end{matrix} \begin{matrix} z_{12}=\frac{1}{2}[z_1,z_2] \\ z_{23}=\frac{1}{2}[z_2,z_3] \\ z_{34}=\frac{1}{2}[z_3,z_4] \end{matrix} \begin{matrix} \\ \rightarrow \\ \\ \end{matrix} \begin{matrix} z_{123}=\frac{1}{2}[z_{12},z_{23}] \\ z_{234}=\frac{1}{2}[z_{23},z_{34}] \end{matrix} \rightarrow z_{1234}=\frac{1}{2}[z_{123},z_{234}]$$

where $\frac{1}{2}[z_1,z_2]$ means the midpoint of the line through z_1 and z_2.
To get the remaining points of the curve determined by z_1,z_2,z_3,z_4 repeat the same construction on $z_1,z_{12},z_{123},z_{1234}$ and $z_{1234},z_{234},z_{34},z_4$, ad infinitum.



The fixing of the precedence of the operators by parentheses is the simplest way to describe the algorithm.

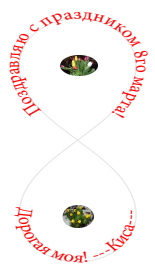
Text along paths, Adobe's pathtext, Bluebook p167




```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Blue Book Program 11, page 171
%%BoundingBox: 126 285 412 435
%%BeginSetup
%%EndSetup
(C:\\PSlib\\Bluebook.eps) run
/Helvetica 16 selectfont 2 setlinewidth
150 310 210 90 rectstroke
360 347 moveto 410 330 lineto 410 380 lineto 360 363 lineto stroke
200 360 70 0 270 arc  200 110 add 360 70 270 180 arc%path for text
(If my film makes one more person feel miserable\
 I'll feel I've done my job.-- WOODY ALLEN) 55 pathtext
```

I would not dream of trying to do this picture in TEX. Adobe treated us on a nice, powerful PostScript `def pathtext`. But ... sometimes we can do without it.

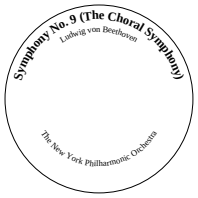
8 March, MAPS 42, 2011



There were several problems which had to be solved in order to achieve the present. First, typesetting along a lemniscate, which was done by Adobe's `pathtext` BLUebook p168. Next the typesetting of Cyrillic in PostScript. This was done by Adobe's `ReEncodeSmall`, Bluebook p207, after I found a font with Cyrillic glyphs. Third, inclusion of `.jpg` photographs in an EPSF program, which was done after the `.jpg` was saved as EPSF in Photoshop.

Powerful `pathtext` is, but ... we can do without `pathtext` when the path is implicit.

CD-DVD lables, MAPS 43, 2012



Adobe's 85 Bluebook

CD label

⟶

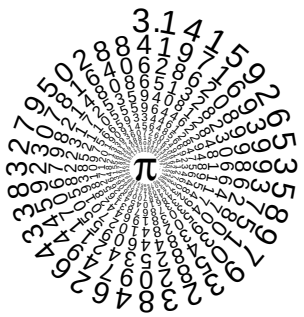
Adobe's 85 Bluebook CD label enriched with notes background

Essentially, it is Adobe's example program from the Bluebook p163, about printing along circular arcs. I have enriched the CD-label by a background, where the `.jpg` picture has been converted into `EPSF`.[28]

π-decimals

A nice printing along an (infinite) implicit spiral is π-decimals. Special is that it has been done without using the page-builder, as Pawel Jackowski used to say. The spiral path is implicit, no explicit path has been built up nor is `pathtext` invoked. It has been published as a GUST programming Pearl 2010. Below a slightly adapted version because `PSView` and `Acrobat` yielded different results on the BachoTEX Pearl version.[29]

The `pop pop` in the procedure are there because `kshow` pushes 2 neighbouring values of the string on the stack each time, which we don't use in the procedure. The backslash allows breaking a long string over lines. The picture was borrowed from the CWI-calendar of 1972.[30]




```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Pi-decimals along a  Spiral, cgl 2010/2012
%%BoundingBox: -80 -100 100 90
%%BeginSetup
%%EndSetup
/Symbol 26 selectfont 1 -18 moveto (p) show%p denotes pi in the symbol font
/Helvetica 20 selectfont
0 70 moveto (3) show 1 0 rmoveto (.) show -2 0 rmoveto%back space
-10 rotate .995 dup scale
{pop pop -10 rotate 3 0 rmoveto .995 dup scale}
(14159265358979323846264338327950288419716939937510582097494459230781640628\
 62089986280348253421170679821480865132823066470938446095505822317253594081\
 28481117450284102701938521105559644622948954930381964428810975665933446128\
 47564823378678316527120190914564856692346034861045432664821339360726024914\
 12737245870066063155881748815209209628292540917153643678925903600113305305\
 48820466521384146951941511609433057270365759591953092186117381932611 79...)
kshow
```

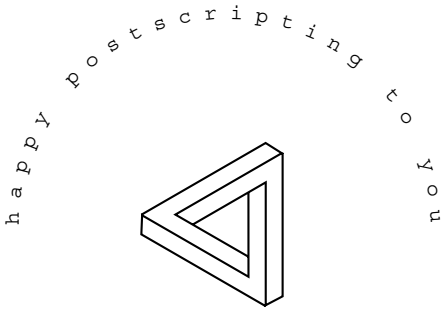
The calculation of the digits of π is a different matter. For a historical survey see paragraph 3.3 in Peitgen c.s.(2004): Chaos and Fractals, or Beukers, F(2000): Pi, de Geschiedenis en de Wiskunde van het getal π. Epsilon. (In Dutch). It is no surprise that millions of digits could only be calculated because of computers.

28. Willi Egger explained how to use ConTEXt's layers to add a picture as background, EuroTEX-ConTEXt2009.
29. `PSView` and `Acrobat Pro` give sometimes different rotated results on pictures where use is made of rotated User Space. Apparently there is some confusion in implementing rotated user space. Moreover, a rotation over 89.9 degrees and a rotation over 90 degrees yielded significantly different results. This is all circumvented in the new version.
30. Frans Goddijn suggested that it would make a nice poster.

Seal: text along circular arc

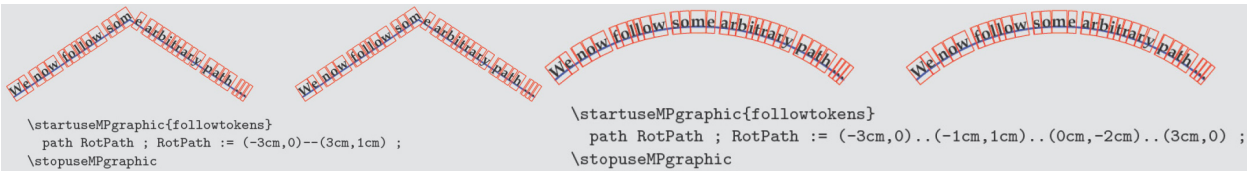
The following seal, or text along a circular arc, illustrates the use of kshow, not pathtext. The circular path is implicit, no explicit path has been built up nor is pathtext invoked.[31]

The included, impossible Escher triangle is intriguing. Once the symmetry has been revealed the programming is a trifle. This time the PostScript def, as included in PSlib.eps, is given in the verbatim below. All 40 pictures of the 'Paradigm: Just a little bit of PostScript'- article have been included in PSlib.eps.




```
/textalongarc{%jalxx
/Courier 10 selectfont
/text (happy postscripting to you) def
/r 100 def
gsave
3{25  34 moveto%3 rotated copies of broken line
  25 -34 lineto
  17 -38.2 lineto
  17  20 lineto
 -17.6 0 lineto
120 rotate
}repeat stroke
  90 rotate  %begin orientation
  0 r moveto %begin point
{pop pop -7.04 rotate 6 0 rmoveto} text kshow
grestore
}def
```

Texts along arbitrary paths in ConTEXt interfaced with MetaPost




```
\startuseMPgraphic{followtokens}
  path RotPath ; RotPath := (-3cm,0)--(3cm,1cm) ;
\stopuseMPgraphic
```

```
\startuseMPgraphic{followtokens}
  path RotPath ; RotPath := (-3cm,0)..(-1cm,1cm)..(0cm,-2cm)..(3cm,0) ;
\stopuseMPgraphic
```

I tried a 1-liner MP-interfacing program from the MetaFun manual in TEXwork's ConTEXt(LuaTeX):

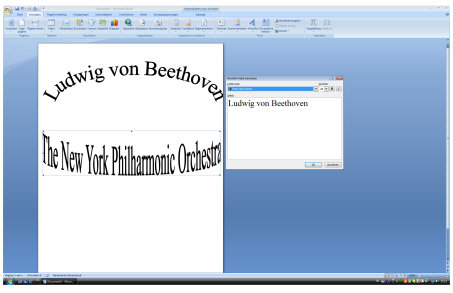
```
\starttext\startuseMPgraphic{dummy} fill fullcircle scaled 5cm withcolor red;\stopuseMPgraphic
\useMPgraphic{dummy} \stoptext
```

My first ConTEXt run! There is still hope in angry days for BLUe ...  .

Professional Circular Text by Photoshop and Word

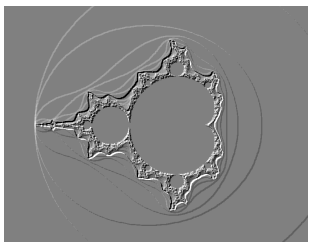
← Photoshop

Word→



31. If you want to do this in TEX&Metafont alone consult Hoenig, A(1989): Circular Reasoning: Typesetting along a circle and related issues, TUGBoat11, or easier consult the digital 24hrs library http://www.tug.org/TUGboat/tb11-2/tb28hoenig.pdf.

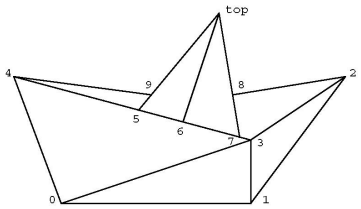
Stars around I — PostScript straight away, MAPS 18, 1997



The stars around notes, I & II, were written after Jacko's Metafont course in Holland, where he taught us among others the OK font. See also Adobe's Bluebook, programs 16–21. In Adobe's Redbook p101, there is an example of a user-defined font of two characters, a filled square and a filled triangle. Another example is given by David Byram–Wigfield who creates a special font `QuadFont` for crosswords. Don Lancaster advocates his Fonts for Free modifications, such as embossed variants.

GUST battleship

The GUST EuroTEX1994 logo — The Battleship — I rewrote at the time in PostScript. In order to obtain the intersection points of 2 straight lines a stack-oriented 2x2 linear equation solver was written in PostScript. In the specification of the points, `\p0` ... `\p9`, `intersect` has been invoked, which delivers the intersection point of 2 lines. The `mean` invoke delivers the midpoint of 2 points. The equation solver in PostScript and the `def`'s `intersect` and `mean` are included in `PSlib.eps`.[32] This is an example where the a priori projection of the drawing and working in 2D throughout is handy. No 3D data.




```
/p0{0 0}def
/p1{3 s mul 0}def
/p2{4.5 s mul 2 s mul}def
/p3{3 s mul s}def
/p4{-.75 s mul 2 s mul}def
/top{2.5 s mul 3 s mul}def
/p5{p0 top p3 p4 intersect}def
/p6{p0 p1 mean top p3 p4 intersect}def
/p7{top p1 p3 p4 intersect}def
/p8{p2 p5 top p1 intersect}def
/p9{p8 dup 0 exch top p0 intersect}def
```

Paradigms: Loops, MAPS 96.2, 1996

Outlines

Borrowed from the TEXbook p65, but rewritten with the use of the FIFO paradigm, and in PostScript, but ... alas, there is no `stringsize` operator in PS. PS' `stringwidth` delivers only the x-size of the string. The kludge of rotating a character and measuring the 'height' did not work. `Pathbb` was needed. A nice example of the use of outlines is GUST's logo.

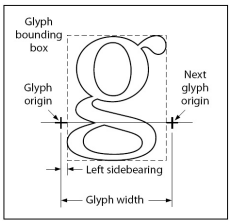
By TEX

By PS (Courier)

```
\leavevmode\fifow Tough exercise. \wofif{ }%{ }=sentinel
%with
\def\fifo#1{\ifx\ofif#1\ofif\fi\process#1\fifo}%
\def\ofif#1\fifo{\fi}%
\def\fifow#1 {\ifx\wofif#1\wofif\fi\processw{#1}\fifow}%
\def\wofif#1\fifow{\fi}%
\def\processw#1{\fifo#1\ofif\ }%
\def\process#1{\boxit#1}%
\def\boxit#1{\setbox0=\hbox{#1}%
  \hbox{\lower\dp0\vbox{\hrule
   \hbox{\vrule\phantom#1\vrule}\hrule}}}
```

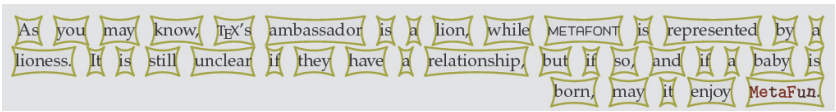
```
/Courier 40 selectfont /str ( ) def
(Tough Exercise.)
{str exch 0 exch put newpath 0 0 moveto
 str false charpath flattenpath pathbbox
 /ury exch def /urx exch def
 /lly exch def /llx exch def
 /w urx llx sub def /h ury lly sub def
 str ( ) ne {llx lly  w h rectstroke}if
 str stringwidth translate
}forall
```

32. Since then a 3x3 linear equation solver has been included in `PSlib.eps`, which (as the 2x2 solver) uses partial pivoting. These are to be preferred above the appealing Metafont/-Post symbolic equation solving functionality when the system is ill-conditioned. For those cases it is best to reformulate the problem into a better conditioned one; next best is to use pivoting strategies. In solving the radical circle problem in my Circle Inversions paper, a sub-problem was to determine the touching point of 2 circles, which is ill-posed, and therefore restated as finding the intersection point of a circle and a nearly-orthogonal line to it.



Paradigm: the use of the nested FIFO-technique,[33] that is, words are scanned and each word is scanned for its characters. A beautiful example of the use of `\phantom`. PS paradigm: walking through a string. In PS a character's BoundingBox has to be determined. The `charbox` width is not the same as `stringwidth` of a character, see picture at right borrowed from the Redbook. The left-side bearing, a kerning(?), is included in the value of `stringwidth`. The PS-code looks simpler with `forall` scanning.

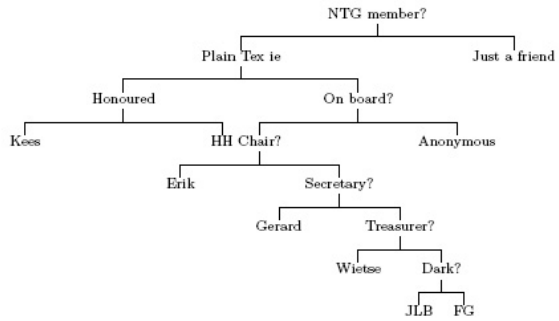
MetaFun's funny-boxed texts



Note the curved, or as Hans calls them squeezed, boxes, which can't be done nicely in TEX alone.

Paradigms: Searching, MAPS 96.2, 1996

After so many years, `BLUe.tex` amazed me by this Searching article. A variant solution of the TEXbook exercise 22.14 has been worked out. In 'Paradigms: searching,' I used a tree structure in TEX for searching. At the end of the article the tree of information was printed as shown below.
I collected my BLUe files: `blue.tex`, `fmt.dat`, `tools.dat`, `lit.dat`, and `pict.dat`, in a map and reproduced the tree by just doing what was stated in the article, and listed in the input verbatim below.[34] Et voilà.
I'm pleased by the results. BLUe surprised even me, by this unbalanced tree and the mean-and-lean data description, after so many years!




```
\input blue.tex
\thisbt{\xoffset{-400}}
\beginbt 1 NTG member?
10 Plain TeX ie?
100 Honoured?
1000 Kees
1001 HH
101 On board?
1010 Chair?
10100 Erik
10101 Secretary?
101010 Gerard
101011 Treasurer?
1010110 Wietse
1010111 Dark?
10101110 Johannes
10101111 Frans
1011 Anonymous
11 Just a friend
16 \endbt%16=checksum sort
of
\bye
```

Below from `bttool` the backtrack macro has been copied. The nodes are binary numbers, and when the `node<binary number>` is undefined the end of the branch has been reached, and backtracking is performed. `\whiteS` draws South and the leaves. Not recreational, pretty advanced macro writing. BLUe in full glory.

```
\beginbt \def\drawbt{\whiteS{120}%
   \ea\ifx\csname\node0\endcsname\relax
         \tbward\fi%Backtrack
   \S{80}\advance\k-125
   {\W{\the\k}\S{80}\edef\node{\node0}%
    \drawbt}%
    \E{\the\k}\S{80}\edef\node{\node1}%
    \drawbt\relax}
\def\tbward#1\relax{\fi}
```

Note the minimal, necessary data specifications: just the binary 'addresses' of the nodes next to their contents. TEX will handle all that is needed.
Contest: How to do this in PS or MP?[35]

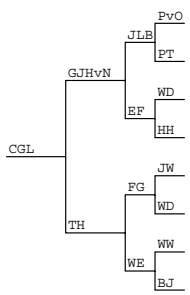
33. FIFO and LIFO sing the blues — Got it?, 1992, 1995(rev), MAPS 9(original). Bernd Raichle likes my `\fifo...\ofif` termination TEXnique.
34. Because it was with the `gkp`-macros I obtained not a picture cropped to the BoundingBox. In order to crop the picture I selected the picture in `Acrobat Pro` and copied it to the clipboard, and created a new cropped `.pdf`, at the expense of sharpness. I should redo it in PostScript, on occasion of EuroTEX-ConTEXt2012. Phil Taylor communicated his recent work on a real genealogy tree in TEX.

A balanced tree in TEX and PostScript

The production rule à la Lindenmayer for the balanced tree reads

$$\mathrm{Bt}_n = \mathrm{E}_n \oplus [\mathrm{N}_{n\div 2}\,\mathrm{Bt}_{n\div 2}] \oplus [\mathrm{S}_{n\div 2}\,\mathrm{Bt}_{n\div 2}], \quad \text{with} \quad n = \ldots 256, 128, \ldots 2,$$

Bt_n the Binary tree of order n,
E_n, N_n, S_n means draw East, North, South with step-size n
$\oplus$ means splice operator, i.e. concatenate properly,
[means store graphics state on the GS-stack and open a new one,
] means remove current graphics state off the GS-stack and recall previous.[36]




```
\def\bintree{\E{\the\n}%
   \ifnum\n=2 \eertnib\fi
   \divide\n2 {\N{\the\n}\bintree}%
              \S{\the\n}\bintree}%
\def\eertnib#1\bintree{\fi}%terminator
\let\Eold\E               %Text at leaves
\def\E#1{\global\advance\k1
  \xytxt{ \csname\the\k\endcsname}%leaf
  \Eold8}}
\def\1{CGL}\def\2{GJHvN}\def\3{JLB}
...
\def\xytxt#1{%place text #1 at x, y
 \xy{\vbox to0pt{\vss
   \hbox to0pt{\strut#1\hss}\kern0pt}}}}
\def\xy#1{%place #1 at x, y
   \vbox to 0pt{\kern-y
     \hbox to 0pt{\kern x#1\hss}\vss}}
```

```
/Bintree{%n on stack
   /n exch 2 div def
   E %draw East and add leaf
   n 8 gt {currentpoint N n Bintree
           moveto      S n Bintree}if
   /n n 2 mul def}def %end Bintree
/N{0 n     rlineto}def /W{n neg 0 rlineto}def
/S{0 n neg rlineto}def
/E{gsave  ntg k get 2 3 rmoveto show grestore
   /k k 1 add def n 16 le {25}{50} ifelse 0 rlineto
  }def
%
/Courier 12 selectfont /k 0 def
/ntg [(CGL) (GJHvN) (JLB) (PvO) (PT) (EF) (WD) (HH)
      (TH) (FG) (JW) (WD) (WE) (WW) (BJ)] def
0 -3 moveto 128 Bintree stroke
```

Intriguing is the use of `currentpoint` in PostScript, which saves the current position values on the stack for use in the other branch. In the TEX-version the placing of the picture on the page is cumbersome. PostScript is simpler for the purpose.
Paradigm: the wind `defs`: `N`, `E` and `W`, which resulted from the Turtle Graphics approach, are used within a recursive environment.

Alice's tale and the mouse's tail, GUST Programming Pearl 2010

This emblematic proza by Lewis Carroll has been first typeset in PostScript, by the use of `forall`, which expects an array, enclosed by [ ], and a procedure, enclosed by { }, on the stack. It is another example of printing text along a path, without an explicit PostScript path, neither is `pathtext` invoked. The array contains a necklace of strings, each enclosed by ( ), the WYSIWYG data. The procedure scales and typesets the lines. No explicit positioning by coordinates on the page nor controlling of the loop is needed. I started with PS' `pathforall`, worked on it for 15-30min, when the direct method popped up.
Paradigm: The `forall` walks through the array and delivers each element of the array on the stack.

In TEX, within a verbatim environment, the same can be achieved with mark-up in a WYSIWYG way; on the other hand one may dawdle with shifted `hbox`-es. Simplest is just to use `\obeyspaces\obeylines` and overrule TEX's default neglecting of superfluous spaces and `e-o-l`s. I was biased by TEX's automatisms and overlooked the simplest solution for quite a while.

35. E-mail solutions to `kisa1@xs4all.nl`.
36. The addition of the graphics state concept to the Lindenmayer production rules is an enrichment.

```
Fury said to
  a mouse, That
                 he met
                   in the
                     house,
                     'Let us
                  both go
              to law:
         I will
    prosecute
  you.
   Come, I'll
       take no
           denial;
                 We must
                          have a
                                  trial:
                                       For
                                  really
                              this
                               morning
                                      I 've
                               nothing
                          to do.'
                   Said the
              mouse to
             the cur,
             'Such a
                   trial,
              dear sir,
       With no
    jury or
  judge,
   would be
         wasting
             our breath.'
                 'I'll be
                 judge
            'I'll be
          jury,'
         Said
             cunning
                 old Fury:
                    'I'll try
                       the whole
                                  cause,
                                  and
                            condemn
                                you
                                    to
                                     death.'
```

```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Alice's tale and the Mouse  tail, cgl feb 2010, 2012
%%BoundingBox: 0 -350 250 115
%%BeginSetup
%%EndSetup
/Courier 10 selectfont
/crlf { .995 dup scale
        currentpoint 10 sub exch pop  LM exch moveto } def
/LM 10 def LM 100 moveto
%array, proc and forall
[(Fury said to)
(  a mouse, That)
(                he met)
(                  in the )
(                    house,)
(                    'Let us)
(                 both go)
(              to law:)
(         I will)
(    prosecute)
(  you.)
(   Come, I'll)
(       take no)
(           denial;)
(                 We must)
(                          have a)
(                                  trial:)
(                                       For)
(                                  really)
(                              this)
(                               morning)
(                                      I 've)
(                               nothing)
(                           to do.')
(                   Said the)
(              mouse to)
(             the cur,)
(             `Such a)
(                   trial,)
(              dear sir,)
(        With no)
(    jury or)
(  judge,)
(   would be)
(         wasting)
(             our breath.')
(                 `I'll be)
(                 judge)
(            `I'll be)
(          jury,')
(         Said)
(             cunning)
(                 old Fury:)
(                     `I'll try)
(                        the whole)
(                                   cause,)
(                                   and)
(                             condemn)
(                                 you)
(                                     to)
(                                      death.')
]{show crlf}forall
```

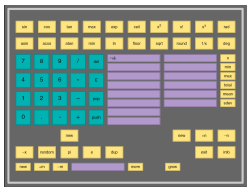
Tic-tac-toe interactivity

A tic-tac-toe application via the log-file, what was called at the time 'in dialogue with TEX,' is discussed in the Searching-article. More elaborated macros are supplied in the article. The advanced macros pay attention to for example a test for inconsistent input, or when a draw situation has arrived to stop automatically and start a new game. At the time when I wrote the macros the dialogue with TEX was via the log-file. At the moment TEXworks opens a different window for supplying the answers to TEX's questions; the questions are shown in the console window.

```
Tic-tac-toe
- - -
- - -
- - -     Supply index for +:
\index=1
+ - -
- - -
- - -     Supply index for o:
\index=5
+ - -
- o -
- - -     Supply index for +:
\index=3
+ - +
- o -
- - -     Supply index for o:
\index=2
+ o +
- o -
- - -     Supply index for +:
etcetera terminated by index 0.
```

```
\def\showboard{\immediate\write0{\1\2\3}
               \immediate\write0{\4\5\6}
               \immediate\write0{\7\8\9}}
\def\initialize{\def\1{-}\def\2{-}\def\3{-}
                \def\4{-}\def\5{-}\def\6{-}
                \def\7{-}\def\8{-}\def\9{-}}
\def\play{\initialize \loop\showboard
   \ifx\mark\markplayer
      \let\mark\markopponent\else
      \let\mark\markplayer\fi
   \immediate\write0{Supply index for \mark:}
   \read0to\index         \expandafter
   \xdef\csname\index\endcsname{\mark}
\ifnum\index>0 \repeat}%end \play
\def\markplayer{+}\def\markopponent{o}
\endlinechar-1 %TB20.18
\play \bye
```

Interactivity: Hans Hagen's calculator in ConTEXt + MetaPost + PDF + ...



Impressive and the summum of interactivity is Hans' calculator. It was said at the time that Knuth was strongly against holding up the processing of TEX, and in the meantime doing something else. Hans has exploited this use of TEX, in for example interfacing of ConTEXt with MetaPost. Fun or serious?

Paradigms: Just a little bit of PostScript, MAPS 19, 1996

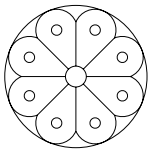
The article will be named JlPS, for short. Previewing was inconvenient via the Apple Laserwriter. `PSView` and `GhostScript` were not available on my PowerMac. All pictures in the article have been included in `PSlib.eps` on the occasion of EuroTEX-ConTEXt21012.

Yin Yang

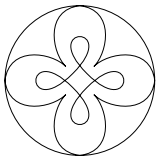
Everybody made his Yin Yang, I presume. Hobby provided it as an example in his MetaPost manual. Included is my matured coding, which is different from the coding in 'Tiling in Metafont and PostScript.'

```
%!PS-Adobe- Yin Yang. cgl July 2009
%%BoundingBox: -8 -8 70 70
/R 25 def     /hR R 2 div def /mR R neg def /mhR hR neg def
/r R 5 div def /mr r neg def
/circle{translate r 0 moveto 0 0 r 0 360 arc}def
0 mR moveto 0   0  R    270  90 arc
            0   hR  hR   90 270 arcn
            0   mhR hR   90 270 arc   fill
R 0 moveto 0 0 R 0 360 arc            stroke
gsave 0 hR circle fill grestore
gsave 0 mhR circle 1 setgray fill grestore
```

Barn and Malbork window



The left window has been done by the use of `arc` and the rotation of user space in PostScript. The right window is an exercise in using splines, the `curveto`, and choosing appropriate control points. The choice of control points I did by trial-and-error. Both are included in `PSlib.eps`.



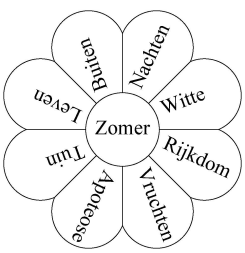
Stylistic flowers

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: -26 -26 26 26
/r 18 def
10 {0 r r 270 360 arc
    r 0 r  90 180 arc
   36 rotate} bind repeat
stroke
```

The black-and-white line-drawing flower has been drawn in PostScript, see verbatim above, where use has been made of the variable user space, such that the drawing of each leaf begins and ends in (0, 0). Subtle are the choices of the circle centres and their sequence. The coding is one of my favourites to demonstrate the use of the variable user space functionality in PostScript. The gradient colouring has been done interactively in Photoshop by my wife Svetlana Morozova on occasion of the EuroTeX-ConTEXt2009. The rotation of the user space can be understood by just paying attention to the rotated coordinate axes. All that follows is drawn with respect to the rotated coordinate axes.[37] At right a circular Julia fractal 'stylistic flower.'

For the bulletin of our gardeners club

The barn window has been reused, enriched by rotated text in PS. Paradigm: rotated texts stored as array.




```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Zomer, vormgedicht, Barnwindow basis. CGL, April 2007
%%BoundingBox: -180 -180 180 180
/l 140 def /Times-Roman 30 selectfont
/m l 22.5 cos mul def /r l 22.5 sin mul def
8{r 0 moveto 22.5 rotate m 0 r -90 90 arc 22.5 rotate}repeat
%inner circle
r 0 moveto 0 0 r 0 360 arc stroke -40 -7 moveto (Zomer) show
/texts[(Witte) (Nachten) (Buiten) (Leven)
           (Tuin)  (Apoteose) (Vruchten) (Rijkdom)]def
0 0 moveto 20 rotate /r r 10 add def
0 1 7{r 0 moveto texts exch get show 45 rotate}for
```

Tiling in PostScript and Metafont — Escher's wink, MAPS 19, 1997

The article will in the sequel be referred to by TPS-MF, for short. All pictures in the article have been included in `PSlib.eps` on the occasion of EuroTEX-ConTEXt2012.

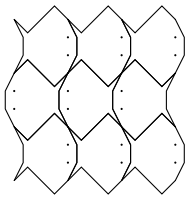
Escher's Sun and Moon

← central part
Dark birds in daylight or white birds at night?
The picture was sampled: the sampled points were provided as spline data.

Zon en maan →

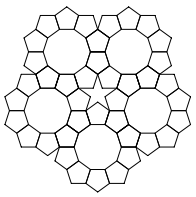
37. See the Bluebook Ch 6 More Graphics, p49, for an enlightening, simple picture.

Escher's fishes and Buddha's

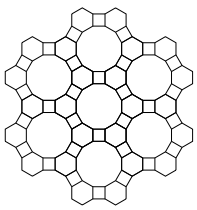


These tiles are examples of Escher's technique where the drawing extends over the boundary of the square tile, such that it matches with the adjacent tiles.
The right picture consists of 4 groups of 4 tiles, where the later are composed of rotated copies.

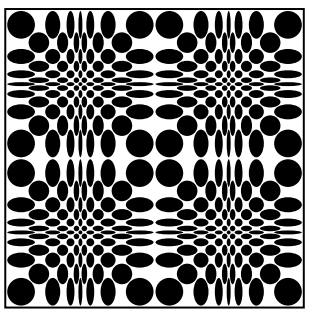
Tilings



The left figure is done by a garland of pentagons. The garland is copied 4 times. The enclosed star is spurious.
The right figure is classified by $\{4, 6, 12\}$, a nice layout for a herb garden.



Schrofer's Op Art

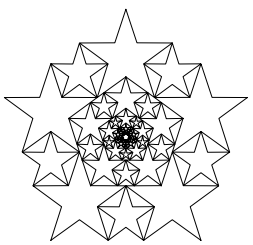


```
/Schrofer{0 begin /flipflop true def /s 5 def
/drawgc{gsave
   r c translate r abs 5 div s add c abs 5 div s add scale
   0 0 1 0 360 arc
   fill grestore}def%end drawgc
/indices [30 21 14 9 5 2 0 -2 -5 -9 -14 -21 -30] def
indices{/r exch s mul def
  gsave indices{/c exch s mul def flipflop{drawgc}if
               /flipflop flipflop not def}forall
  grestore}forall
end}def%end Schrofer
%4 tiles and border
gsave 2{gsave 2{schrofer 375 0 translate}repeat
               grestore   0 375 translate
       }repeat grestore
5 setlinewidth -190 dup 755 dup rectstroke%border
```

A nice picture is Schrofer's Op Art, of which I included a tile of four.[38] Crucial are the row and column indices. The circles and ellipses are scaled copies of the unit circle. All the 80+pictures from the 'Paradigm: Tiling in Metafont and PostScript'-article have been included in `PSlib.eps` on occasion of EuroTEXConTEXt2012.

Tiling by stars

In `PSlib.eps` this stars composition has name `tilxia`; the code is ≈ 40 lines long.



```
/star % n r on stack, n>=5
{/rstar exch def /nstar exch def
 /alfahstar 180 nstar div def
 /rinstar rstar 2 div alfahstar cos mul 1.5 sub def
 0 rstar moveto
 nstar{alfahstar rotate
     0 rinstar lineto
     alfahstar rotate
     0 rstar lineto
  }repeat stroke
}def
```

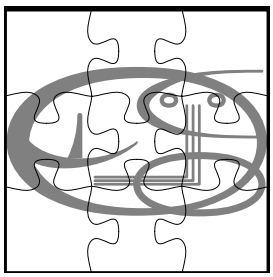
Star impression by Svetlana Morozova →

38. Willem Schrofer, 1898–1968, was a Dutch artist and teacher. In the 30s he painted abstract later figurative.
`http://nl.wikipedia.org/wiki/Willem_Schrofer`.

Puzzle with cat

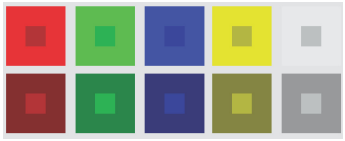
```
%MetaPost variant of \quote{cat} which was adapted from Metafont
beginfig(1); tracingstats:=proofing:=1;
path p[]; sz=25; hsize=17.5sz; vsize=10sz;
%moustache
pickup pencircle scaled .1pt;
draw (.75hsize, .75vsize)--(.75hsize, .2vsize)--
  (.333hsize, .2vsize);
draw (.725hsize, .75vsize)--(.725hsize, .225vsize)--
  (.333hsize, .225vsize);
draw (.7hsize, .75vsize)--(.7hsize, .25vsize)--
  (.333hsize, .25vsize);
z1=(  hsize,.5vsize);  %right
z2=(.5hsize,  vsize);  %top
z3=(      0,.5vsize);  %left
z4=(.5hsize,      0);  %bottom
penpos1(.05vsize,0);penpos2(.09vsize,90);penpos3(.175vsize,180);
penpos4(.075vsize,270);
%Nonlinear interpolation for extra point z25
z25=(z2{left}..{down}z3)intersectionpoint
    ((.2hsize,0)--(.2hsize,vsize));
penpos25(.15vsize,135);
penstroke z1e{up}..z2e{left}..z25e..z3e{down}..
        z4e{right}..{up}z1e;
%mouth
pickup pencircle scaled .2pt;
draw superellipse((hsize, .2vsize),(.75hsize, .4vsize),
                  (.5hsize,.2vsize),(.75hsize,0),.725);
%ear
z5=(0,.5vsize);      penpos5(1.75pt,-90);
z6=(.5hsize,.5vsize);penpos6(.8pt,0 );
p1=z5..controls (.125hsize, .333vsize) and
   (.375hsize,.333vsize)..z6;
z7=point.5 of p1;    penpos7(1.2pt,-30);
z9=point.5 of p1;%(.25hsize,.4vsize);
x9:=x9-.175pt;         penpos9(.75pt,180);
z8=(.25hsize,.75vsize);penpos8(.3pt, 180);
penstroke z6e..z7e..z5e;
penstroke z8e{down}..z9e;
%brow
z10=(hsize,vsize);penpos10(.2pt,90);
z11=(.575hsize,.9vsize);penpos[11](.5pt,135);
z12=(.5hsize, .75vsize);penpos12(.8pt,180);
z13=(.575hsize,.6vsize);penpos13(.4pt,-135);
z14=(hsize,.5vsize);    penpos14(.15pt,-90);
penstroke z10e{left}..z11e..{down}z12e..
          z13e..{right}z14e;
%eyes
p2= superellipse((sz, .375sz),
 (.5sz, .75sz),(0,.375sz),(.5sz,0),.725);
pickup pencircle scaled .1pt;
draw p2 shifted (.6hsize, .75vsize);
draw p2 shifted (.79hsize, .75vsize);
endfig;
end
```

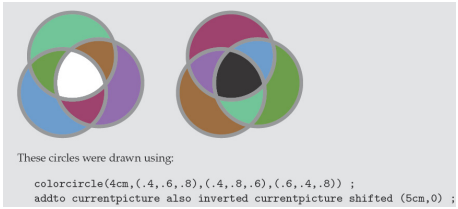
```
%!PS Puzzle of cat, cgl June 97
%%BoundingBox: -4 -39 179 144
%%Creator: MetaPost
%%CreationDate: 1996.05.03:1858
.5 setgray
newpath 178.50075 50 moveto
178.50 66.10 167.61 79.62 153.48 87.70 curveto
133.59 99.07 110.39 102.00 87.5 102.00 curveto
64.61 102.00  41.41  99.07 21.52 87.70 curveto
7.39 79.62 -3.50 66.10 -3.50 50 curveto
-3.50 33.90 7.39 20.38 21.52 12.30 curveto
41.41 0.93 64.61 -2.00 87.5 -2.00 curveto
110.39 -2.00 133.59 0.93 153.48 12.30 curveto
167.61 20.38 178.51 33.90 178.51 50 curveto closepath fill
%
%et cetera, next the puzzle overlay
%
87.5 50 translate 0 setgray .5 setlinewidth
/s 31.5 def /t .6 s mul def
/el{s 0 moveto currentpoint .8 s mul 0
    .4 s mul -.2 s mul  .2 s mul 0 curveto
   currentpoint currentpoint exch
    t t  0 t curveto
}def
%
/side{el reversepath -1 1 scale el -1 1 scale}def
%
/piece{4{0 s translate side 0 s neg translate
  90 rotate}repeat}def
%
/elinv{1 -1 scale el 1 -1 scale}def
/sideinv{elinv reversepath -1 1 scale elinv -1 1 scale}def
/ipiece{4{0 s translate sideinv 0 s neg translate
  90 rotate}repeat}def
/border{3 s mul dup moveto
        4{-3 s mul 3 s mul lineto 90 rotate}repeat
        closepath}def
border clip
ipiece stroke
-2 s mul 4 s mul 2 s mul{/i exch def
-2 s mul 4 s mul 2 s mul{/j exch def
 gsave i j translate piece stroke grestore
}for}for
7.5 setlinewidth border stroke
%%EOF
```

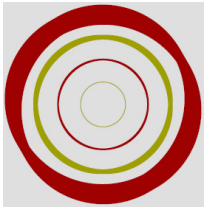


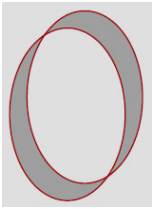
The background picture I made ≈50 years ago by hand. When I started with Metafont in 1995 it was my first graphics example. The PostScript code resulted from the MetaPost adaptation, i.e. deleting Metafont peculiarities. Both included codes are too lengthy to my taste. The PS variant shows how, after we have distilled the PS data from MP, the picture can be further enriched in PS.

MetaFun simple contrasts





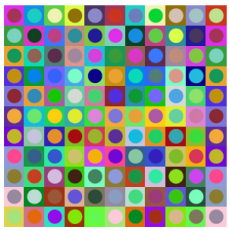


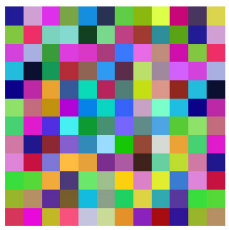


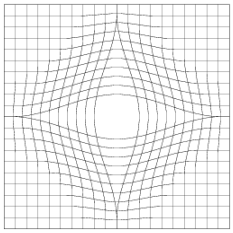
The two pictures at right make use of a varied pen in MetaPost. Calligraphic effects can be obtained.

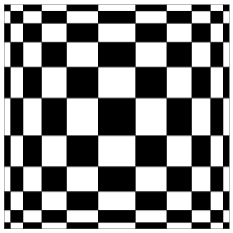
Vasarely

In 1995 I created 8 Vasarely impressions in Metafont; today I could still visualize them in BlueSky's Metafont on my PowerMac of the mid-90s.









The black-and-white pictures were visualized by, and downloaded as `.png` from, Troy Henderson's `mppreviewer`.[39] The Metafont line-picture makes use of the `interpath` functionality, Metafont book p134, which functionality is not available in PostScript, see code below, nor is there a path data-structure, alas.

```
sz=100; path p,q;
p= (-sz,0){right}...(-.9sz,0)...(0,.2sz)...(.9sz,0)...{right}(sz,0);
q= (-sz,sz)--(-.25sz,sz)--(0,sz)--(.25sz,sz)--(sz,sz);
for k= 0 upto 10: pickup pencircle scaled (.02(k+1)*pt) draw interpath(k/10, p, q); endfor
addto currentpicture also currentpicture rotated 180;
addto currentpicture also currentpicture rotated 90;
pickup pencircle scaled .1pt; draw unitsquare scaled 2sz shifted (-sz,-sz);
```

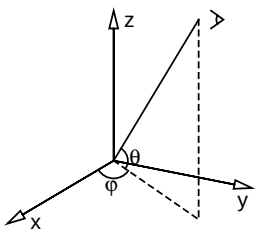
The 3 Vasarely[40] impressions left use PostScript'`rnd`, the pseudo-random number generator. The PS-code for the second picture reads

39. Troy has improved his previewer since 2009, several packages can be used now. `http://www.tlhiv.org/mppreview`. He has also provided a LaTeX previewer and a function-grapher previewer. See his TUGboat 2012 article. All have been done because installing volunteer software by a casual user has become too cumbersome.

40. Victor Vasarely born Vásárhelyi Gyözö, 1906 Pécs – 1997 Paris, was a Hungarian French artist whose work is generally seen as aligned with Op Art. His work entitled Zebra, created by Vasarely in the 1930s, is considered by some to be one of the earliest examples of Op Art. Vasarely developed his style of geometric abstract art, working in various materials but using a minimal number of forms and colours. `http://www.en.wikipedia.org/wiki/Victor_Vasarely`.

```
/r 20 def   /R r 10 mul def
/rd2 r 2 div def /-rd2 rd2 neg def /ri rd2 2 sqrt div def %ri = radius inner circle
/s 1073741823 def %/s 2 31 exp 1 sub def
/square{nrand nrand nrand setrgbcolor -rd2 dup r r rectfill
        nrand nrand nrand setrgbcolor 0 0 ri 0 360 arc fill}def
/nrand{rand s div}def 1951 srand%seed
R neg r R{/i exch def
R neg r R{/j exch def gsave i j  translate square grestore}for}for
```

Projection for emulation of space objects

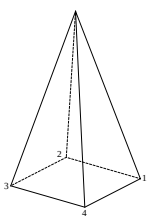


right-screw coordinate system with view direction φ, θ in the main octant

```
/ptp{% point x y z ==> x' y'
% use: /pair { x y z ptp } def
%parameters: phi, theta viewing angles
%coordinate system xyz: x to yuo y right z up, right turning
ptpdict begin /z exch def/y exch def/x exch def
x phi sin mul neg                y phi cos mul add
x phi cos mul theta sin mul neg y phi sin mul theta sin mul sub
                                 z theta cos mul add
end}bind def
/ptpdict 3 dict def
```

The idea in the projection used is that an object is viewed at in plane (computer screen) orthogonal to the view direction. In programming this is translated such that the spacial coordinates are projected onto the projection plane by `ptp`, mnemonics for point-to-pair.[41]

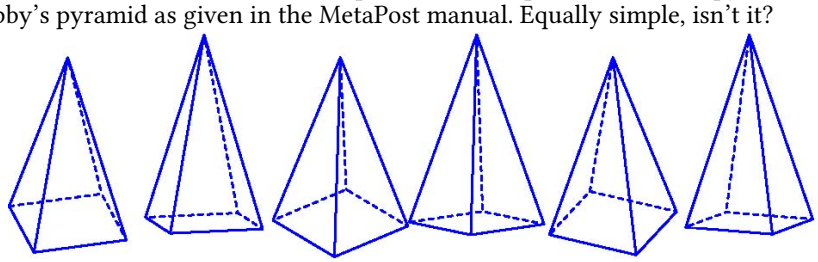
The pyramid illustrative example is the pyramid. Data of the pyramid and the pyramid code have been borrowed from `PSlib.eps`.




```
/z1{r neg r   0 ptp}def
/z2{r neg dup 0 ptp}def
/z3{r r neg   0 ptp}def
/z4{r r       0 ptp}def
/top{0 0 r 4 mul ptp}def
```

```
/pyramid{z1 moveto z2 lineto z3 lineto
   [2]1 setdash stroke
z3 moveto z4 lineto z1 lineto
   []0 setdash stroke
top moveto z1 lineto%  2 -3  rmoveto (1)show
top moveto z3 lineto% -7 -3  rmoveto (3)show
top moveto z4 lineto% -3 -10 rmoveto (4)show
stroke
top moveto z2 lineto
[2]1 setdash stroke% -10 0 rmoveto (2)show
}def  %end pyramid
```

Pyramid viewed from various viewpoints. Do compare the code of pyramid with Hobby's pyramid as given in the MetaPost manual. Equally simple, isn't it?



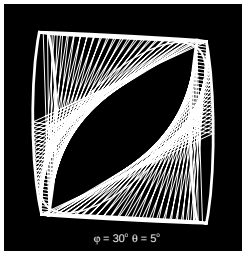
Escher's impossible cubes, TEX Education, EuroTEX-ConTEXt2009
For each corner of the cube there are 8 data-points, specified in 3D. In projection the points of intersection have been calculated as function of the viewing angle. Pretty detailed and tedious code. Too lengthy to be included here. The Metafont book contains a poor man's version, p113, exercise 13.7. The impossible cube in PostScript was written on occasion of the EuroTEX-ConTEXt2009.

41. For more detail see Appendix 0 of Gabo's Torsion, MAPS 42, 2011.

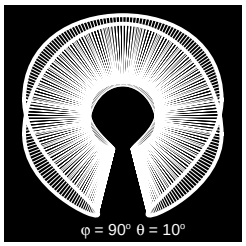
The paper is also available at `http://www.ntg.nl/maps/39/05.pdf`, next to the complete proceedings.

Paradigm: Graphics and TEX — a reappraisal of Metafont, MAPS 16, 1996

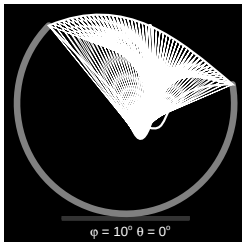
In 1996 I emulated Linear Construction No 2 in Metafont. The picture created by the original Metafont version of the emulation of Linear Construction No 2 is full-page included in LaTeX's Graphics Companion. In the Gabo's Torsion paper an improved and more accurate version in PostScript has appeared, next to some more emulations of Gabo's works. At right an animated simple versions of Linear Construction No 2. In 2011 I rewrote the emulations in PostScript. More use of projection techniques is in Gabo's emulations. The last PostScript version of the Linear Construction No 2 is the most complete and the best. The reverse video suggests the perspex material. Mathematically, the constructions are regular surfaces, meaning the surfaces are suggested by straight lines. It is said that Suspended was Gabo's favourite, because he showed the object at each-and-every exhibition.



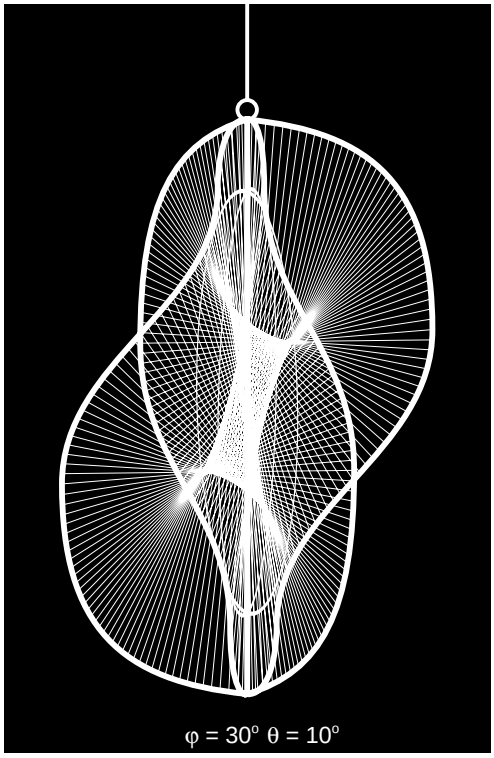
←Linear Construction No 1



←Spheric Theme
Linear Construction No 2→



←Suspended



Invokes of Gabo's emulations from PSlib.eps.

```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Linear Constr. in Space No 1
%%Author: Kees van der Laan, cgl 2011
%%BoundingBox: -130 -135 130 135
%%BeginSetup
%%EndSetup
%%BeginProlog
(C:\\PSlib\\PSlib.eps) run
%%EndProlog
linearconstructionno1 showpage
```

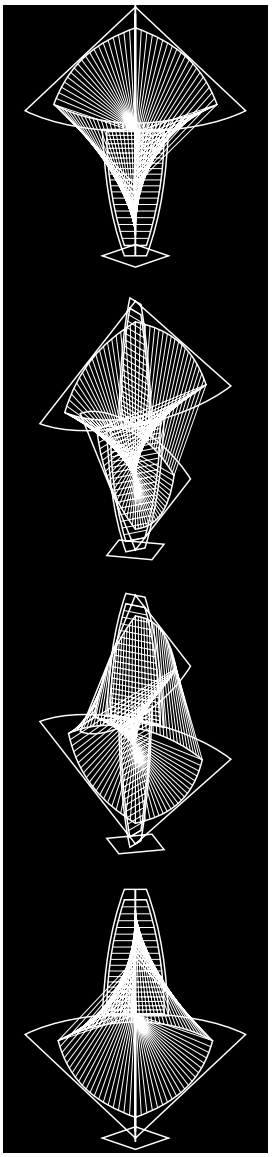
```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Linear Constr. in Space No 2
%%Author: Kees van der Laan, cgl 2011
%%BoundingBox: -125 -30 125 355
%%BeginSetup
%%EndSetup
%%BeginProlog
(C:\\PSlib\\PSlib.eps) run
%%EndProlog
linearconstructionno2 showpage
```

```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Spheric Theme
%%Author: Kees van der Laan, cgl 2011
%%BoundingBox: -90 -95 90 85
%%BeginSetup
%%EndSetup
%%BeginProlog
(C:\\PSlib\\PSlib.eps) run
%%EndProlog
sphericmtheme showpage
```

Warning GhostScript can't be used for previewing with library use because GhostScript does not support the run command for file-inclusion, apparently. Do use PSView, Acrobat Pro or ...

Gabo's Torsion, MAPS 42, 2011

For the Metafont/Post aficionados my Torsion Metafont code of old is included, complete with projection and interactivity, which was not included in MAPS 42.
My emulations of Gabo's[42] objects on paper started in Metafont in 1996,[43] which marks the beginning of my using projection techniques. In the paper a few of Gabo's 3D constructions have been emulated in projection and can be viewed from various viewing angles. Torsion pictures in reverse video have been supplied below.



```
%December 1995, cgl. Gabo's torsion
tracingstats:=proofing:=1;screenstrokes;%tracingmacros:=tracingtitles:=1;
"Torsion from different viewpoints";
string yorn; message "Gabo's Torsion.";
             message "Do you wish simplest variant? (y or n):";
yorn:= readstring; size=50;
def openit = openwindow currentwindow from origin
    to (screen_rows,screen_cols) at (-size,300)enddef;
pickup pencircle scaled .005pt;
pair aux[]; path  po[], pi[];
if yorn="n": d=.1size else: d=0 fi;
r=.2size;
for b= 20:%15step10until45:
for a= 0 step30until90:
currentpicture:=currentpicture shifted (2size,0);
%The following def must be included or a, b, must be supplied as arguments.
def ptp(expr x,y,z)=(-x*cosd a +y*sind a,
                     -x*sind a *sind b -y*cosd a *sind b+ z*cosd b)enddef;
po1:=ptp(0,-size,2d)--ptp(0,0,size+2d)--ptp(0,size,2d)&
     ptp(0, size,2d)..ptp(0,0,0)..ptp(0,-size,2d)..cycle;
aux0:=.5[ptp(0,-size,2d),ptp(0,0,size+2d)];
aux1:=.5[ptp(0,size,2d),ptp(0,0,size+2d)];
pi1:=ptp(0,-size+2.5d,2.5d)..controls aux0..
     ptp(0,0,size-d)..controls aux1..
     ptp(0,size-2.5d,2.5d)...
     ptp(0, size-2.5d,2.5d)...ptp(0,0,d)...ptp(0,-size+2.5d,2.5d)..cycle;
po2:=ptp(-size,0,-2d)--ptp(0,0,-size-2d)--ptp(size,0,-2d)&
     ptp(size,0,-2d)..ptp(0,0,0)..ptp(-size,0,-2d)..cycle;
aux3:=.5[ptp(-size,0,-2d),ptp(0,0,-size-2d)];
aux4:=.5[ptp(size,0,-2d),ptp(0,0,-size-2d)];
pi2:=ptp(-size+2.5d,0,-2.5d)..controls aux3..
    ptp(0,0,-size+d)..controls aux4..
    ptp(size-2.5d,0,-2.5d)...
```

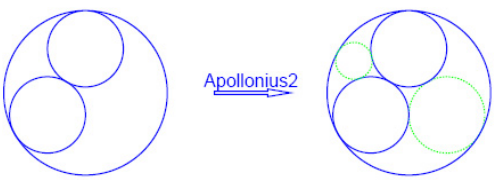
42. Naum Gabo, 1890–1977. Born Naum Borisovich Pevsner. Bryansk. Russian Constructivist. An excellent book about him and his works: Naum Gabo 60 years of Constructivism. Prestel-Verlag 1985, which appeared on the occasion of the retrospective exhibition with the same name at the Dallas Museum of Art, the Art Gallery of Ontario, the Guggenheim Museum NY, the Akademie der Künste Berlin, the Kunstsammlung Nordrhein-Westfalen, the Tate Gallery London. Wikipedia contains a short biography.
43. Graphics and TEX — a reappraisal of Metafont, 1996, MAPS 16.

```
    ptp(size-2.5d,0,-2.5d)...ptp(0,0,-d)...
    ptp(-size+2.5d,0,-2.5d)..cycle;
%Foot
po3:=ptp(r, 0,-size-2d)..ptp( .706r, -.706r,-size-2d)..
     ptp(0,-r,-size-2d)..ptp(-.706r, -.706r,-size-2d)..
     ptp(-r,0,-size-2d)..ptp(-.706r,  .706r,-size-2d)..
     ptp( 0,r,-size-2d)..ptp( .706r,  .706r,-size-2d)..cycle;
%
if yorn="n":fill po1; unfill pi1;fill po2; unfill pi2
           ;draw ptp(0,0,-size+d)--ptp(0,0,-size-2d)
           ;draw ptp(0,0,size-d)--ptp(0,0,size+2d)
       else:draw po1; draw po2; draw pi1; draw pi2 fi;
fill po3;
for k=0 upto 20: draw point .1k of pi1--point 5-.1k of  pi2; endfor
for k=0 upto 20: draw point 3+.1k of pi1--point 2-.1k of  pi2;endfor
showit; endfor endfor end
```

Circle Inversions, MAPS 40, 2010 This paper also introduces `PSlib.eps`.

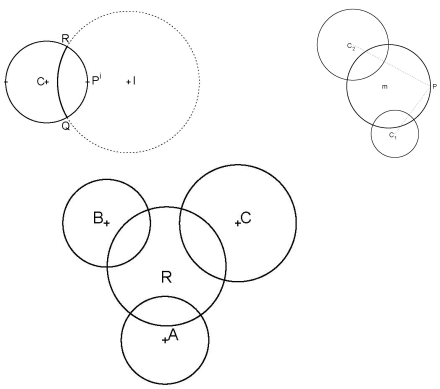
Apollonius problem

The jewel of the Circle Inversions paper is the solution of Apollonius problem: circles touching three circles. Apollonius problem is a classic, which solution I have overlooked for quite a while. The use of the `Apollonius2` PostScript `def` is no more difficult, or easier, depending on your expertise, than using high-level packages.



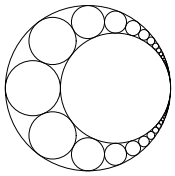

```
0   0  R 0 360 arc stroke% draw the 3 given circles
x1 y1 r1 0 360 arc stroke
x2 y2 r2 0 360 arc stroke
x1 y1 r                  % 3 circle specs for Apollonius2
x2 y2 r                  % pos r means touch external to x2 y2 r circle
0   0 R neg              % neg R means touch inside R-circle
Apollonius2              %delivers touching circles
 /rsnd1 exch def /ysnd1 exch def /xsnd1 exch def%collect data from stack
 /rsnd2 exch def /ysnd2 exch def /xsnd2 exch def
green                    %or a setdash when in b&w
xsnd1 ysnd1 rsnd1 0 360 arc stroke%draw found circles
xsnd2 ysnd2 rsnd2 0 360 arc stroke
```

The radical circle is the circle orthogonal to three circles. The library `def radical` from `PSlib.eps` avoids an ill-posed sub-problem. Below are included pictures of: two circles which intersect orthogonally, a circle through a point p which intersects two circles orthogonally, and the radical circle. How to invoke `radical` from `PSlib.eps` is shown in the PostScript snippet.



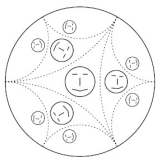

```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Radical circle. CGL april2010
%%BoundingBox: -90 -90 114 105 %-r -r 3r 7r
(C:\\PSlib\\PSlib.eps) run %PS library
H14pt setfont
/r 50 def /mr r neg def
/Ax 0 def  /Ay mr  def          /A {Ax Ay} def  /Ar .75 r mul def
/Bx mr def /By r  def           /B {Bx By} def  /Br .75 r mul def
/Cx 1.25 r mul def  /Cy r def   /C {Cx Cy} def  /Cr     r def
A plus B plus C plus
newpath A Ar 0 360 arc stroke A moveto   2 0 rmoveto (A) show
newpath B Br 0 360 arc stroke B moveto -12 0 rmoveto (B) show
newpath C Cr 0 360 arc stroke C moveto   2 0 rmoveto (C) show
Ax Ay Ar
Bx By Br
Cx Cy Cr radical /radr exch def /rady exch def /radx exch def
newpath radx rady radr 0 360 arc stroke
radx 5 sub rady 14 sub moveto (R) show
```

Circle covered by circles

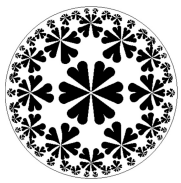


The covering of a circle by small circles I did in 1997 by using a non-linear equation solver in PostScript: `solveit`. Since I rediscovered the solution of Apollonius problem it can be programmed simpler, by the use op the `def Apollonius2`. At right a nice emulated collier of the inversion of the Mandelbrot fractal, borrowed from Lauwerier(1990): Een wereld van Fractals.

Inverted smileys and hearts



Tedious programming for inverted smiley-s. I did only 2 levels of circle inversions.
Inverted hearts was a side-effect when in search for Escher's Circle limits. The PS code was prompted by the Apple Laser Writer.

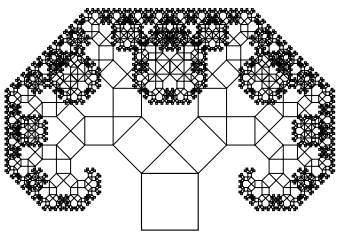


Circle inversion of a rectangular grid

The inversion has been simplified: circular curves have been straightened in the inversion.
At right the realization as a skylight window.

Pythagoras Trees, submitted MAPS, BachoTEX2012 proceedings

The Tree is a collection of scaled and rotated squares placed such that each parent square and its descendants enclose a rectangular triangle. The program is my favourite, non-trivial example of translating and rotating user space in PS. All one has to program is drawing a square and place it scaled and rotated at the right place, repetitively. This can be programmed in PostScript elegantly due to the translation and rotation of User Space functionality. Backtracking and the bookkeeping of auxiliaries is implicit.
The paper contains variants of the Pythagoras tree, such as an oblique tree and the 'X-mas' tree. More realistic trees are mentioned. The Pythagoras Tree has appeared as GUST Programming Pearl in 2011.



```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Pythagoras Tree of squares
%%BoundingBox: -125 -20 175 200
%%BeginSetup              %crops to the prescribed BB
%%EndSetup                %when processed by Acrobat Pro
(C:\\PSlib\\PSlib.eps)run
/s 50 def                 %size of the side of the square
11 pythagorastree  pop    %order 11
%%EOF
```

à la Mondrian, MAPS 41, 2010

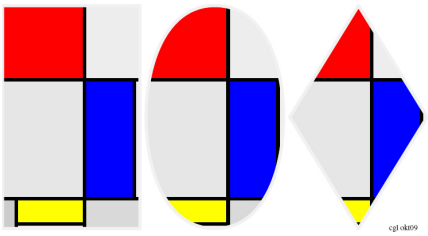
The compositions of coloured line pieces biased by a shade of colour, see accompanying figure,[44] are optionally enclosed in, and clipped by, a square, diamond or oval, to be specified by the user. The program is called `Mondrian` in `PSlib.eps`. In the article a

MetaPost variant was developed for comparison. The PostScript code lends itself for library use. Moreover, the use of a PostScript library `def` is more direct, a one-pass job, then the use of MetaPost, because MetaPost is a preprocessor of PostScript.[45]

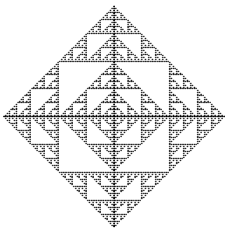
←

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 620 790
(C:\\PSlib\\PSlib.eps) run
22121943 1  0  0  2 Mondrian showpage
07101951 .8 .2 .7 1 Mondrian showpage
%%EOF
```

From De Style period →



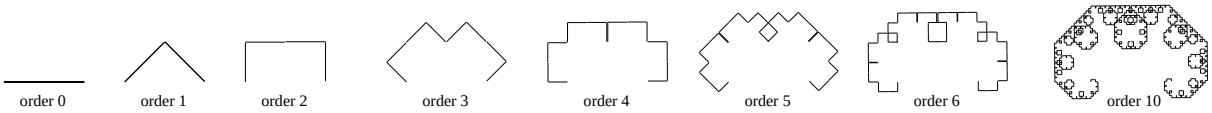
Game of Life Just to show that this game can yield fractal structures.



```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Growth Cell model a la Pickover, simplified
%%Author:  Kees van der Laan, March 2012
%%BoundingBox: -195 -195 195 195
%%BeginSetup
%%EndSetup
(C:\\PSlib\\PSlib.eps) run
64 pickover showpage
```

Classical Math fractals in PS, submitted MAPS, BachoTEX2012 proceedings

Lévy fractal An approximation of the Lévy fractal is also called a C (broken) line of a certain order. The constructive definition of various orders of C lines starts with a straight line, let us call this line C_0. An isosceles triangle with angles 45°, 90° and 45° is built on this line as hypotenuse. The original line is then replaced by the other two sides of this triangle to obtain C_1. Next, the two new lines each form the base for another right-angled isosceles triangle, and are replaced by the other two sides of their respective triangle, to obtain C_2. After two steps, the broken line has taken the appearance of three sides of a rectangle of twice the length of the original line. At each subsequent stage, each segment in the C figure is replaced by the other two sides of a right-angled isosceles triangle built on it. Such a rewriting relates to a Lindenmayer system. Paradigm: Lindenmayer production rule.



Julia fractals in PostScript, EuroTEX-ConTEXt2012

There are many Julia fractals. The one included below is my favourite. The left (incomplete) Julia fractal is obtained by inverse iteration and Monte Carlo, the right by the boundary scan method and enriched by colours by my wife Svetlana Morozova on occasion of EuroTEX-ConTEXt2012.[46] Interesting is the relationship of the various

44. Piet Mondriaan, 1872-1944, was a Dutch painter. He was an important contributor to the De Stijl art movement and group, which was founded by Theo van Doesburg. He evolved a non-representational form which he termed Neo-Plasticism. This consisted of white ground, upon which was painted a grid of vertical and horizontal black lines and the three primary colours.
`http://en.wikipedia.org/wiki/Piet_Mondrian`.

45. MetaPost does **not** interface. For example symbolic names declared in MetaPost can't be accessed in the resulting PostScript.

Julia fractals and the Mandelbrot fractal, which is the map to, and the bifurcation diagram of, the various Julia quadratic fractals.

```
%!PS-Adobe-3.0 EPSF-3.0  $$
%%BoundingBox: -165 -85 165 85
(C:\\PSlib\\PSlib.eps) run
-.8 .15 5000 JULIAMC         %<--Cloud inverse iteration
showpage
-.59 0.34 2.1 1.85 80 JULIABS%Cloud boundary scan -->
showpage
```

The late Mandelbrot[47]advocated that Fractal Geometry is better suited to model clouds and similar repetitive, natural forms than Euclidean geometry.

Conclusions

Portability in time of TEX scripts is hampered when several tools next to TEX have been used. TEX scripts which included PostScript graphics in the past, have to be adapted for use with pdfTEX.

Not only are TEX's CM bitmapped fonts too rigid with respect to font modifications and scaling, but also the philosophy of unbreakable boxes is too rigid in view of page-breaks. TEX's macro language is complicated, verbose and error-prone.

User mark-up can be reduced by letting TEX insert mark-up. Illustrations can be obtained by programming in PostScript, supported by Lindenmayer-like production rules and by projection techniques for emulation of 3D objects. Photoshop can be used as post-processor.

Printing along implicit paths can be done without the use of Adobe's pathtext and alikes.

On occasion of EuroTEX-ConTEXt2012 PSlib.eps has been extended by more than 175 pictures, from JlPS, TPS-MF, and from my fractal geometry work, i.e. translations of Lauwerier's BASIC codes into PostScript, and contains of sept 2012 ≈ 300 defs next to constants and colour names. The defs of Adobe's Bluebook are included and are also available in a separate file. The test programs of the Bluebook are also available in a separate file.

BLUe.tex, fmt.dat, tools.dat, pic.dat for TEX-alone pictures (and the relatively new cousin PSlib.eps, the library for PostScript pictures)[48] next to lit.dat, can be of use for Ben Lee User of the TEXbook fame, even after 17 years. They survived several computer migrations. Pictures of pic.dat can be reused and adapted. I undusted the unbalanced binary tree jewel.

Metafun I (mis)used for viewing old Metafont graphics. Bluesky's Metafont on my old PowerMac is no longer needed.

Not only is the use of TEX&Co recreational, the attendance of (Euro)TEX-meetings is highly recreational and instructive, especially the Polish GUST BachoTEX's with their bonfire and guitars at night. They were like holidays for me.

The mark-up of this paper does not adhere to the promise of the ideal marked-up texts; a lot of adjusting had to be done in order to obey the limitations of the page-size due to the vbox-s with unbreakable verbatim and picture elements next to each other. Letting float these elements would have yielded a mess.

Do realize that typesetting Bridge, Chess, or ... positions is several orders of magnitude less complicated than developing Bridge, c.q. Chess, playing programs.

46. Lauwerier, H.A(1987): FRACTALS — meetkundige figuren in eindeloze herhaling. Aramith. (Contains programs in BASIC. Lauwerier, H.A(1991): Fractals: Endlessly Repeated Geometrical Figures, Translated by Sophia Gill-Hoffstadt, Princeton University Press, Princeton NJ1. ISBN 0-691-08551-X, cloth. ISBN 0-691-02445-6 paperback. 'This book has been written for a wide audience ... ' Includes sample BASIC programs in an appendix. Audience: Instructors, (high-school) students, and the educated layman.)
47. Mandelbrot(1982): The Fractal Geometry of Nature. W.H. Freeman and Co.
48. Introduced in Appendix 1 of Circle Inversions, MAPS40, 2010. http://www.ntg.nl/maps/40/03.pdf.

Developers do the typesetting as an aside. Keep the right balance between form and contents.

'A professional starts where an amateur ends', to quote G.E.Forsythe, my greatest hero. Room for professionals.

> *"It's a myth to believe that each-and-every (La)TEX, ConTEXt, LuaTEX, or ...-user can produce printing-house typographic quality."*

It would be better if users are more modest and strive after preprint results. A preprint is correct with respect to contents and language. To achieve typographic printing-house quality requires another level of non-TEXnical expertise. Typographical corrections should be strictly local and have no global effects, avoiding introducing new typographical errors.

IDE My PC runs 32 bits Vista, with Intel Quad CPU Q8300 2.5GHz assisted by 8GB RAM. I visualize PostScript with `PSView` and convert into `.pdf` via `Acrobat Pro 7`.[49] My cripple PostScript editor is just Windows 'kladblok (notepad), and sometimes I misuse TEXworks for the purpose.' I use Adobe's `EPSF`-feature to crop pictures to their `BoundingBox`. The cropping is necessary for inclusion in documents.

Pictures made by the `gkp`-macros are still viewed in my `BLUe.tex` system of 1995. Metafont pictures are viewed in BlueSky's Metafont which runs on my PowerMac of 1996. No `.eps` or so as result. MetaPost pictures I drop on Henderson's `mppreviewer` and get `.png` in return. Old Metafont I can view in Hans Hagen's MP-interfacing program as well, next to via my BlueSky Metafont on my old PowerMac.

For document production I use TEXworks IDE with the plain TEX engine, pdfTEX, with as few as possible structuring commands borrowed from `BLUe.tex` — adhering minimal TEX mark-up. I use the Terminal font in the edit window with the pleasing effect that comments remain vertically aligned in the `.pdf` window. For checking the English spelling I use the public domain `en_GB` dictionary and hyphenation patterns `en_GB.aff` in TEXworks.

Prior to sending my `PDF`'s by email the files are optimized towards size by Acrobat Pro.[50] The bad news with respect to `.eps` into `.pdf` conversion is, that Acrobat 10 Pro X does not allow for the `run` command for library inclusion.

Errors of TEX ...

It is not told anywhere, but the rigid, bitmapped, unscalable CM-fonts is THE logical error of the twin TEX&Metafont, of which we suffer up till today.

> *"I spend a whole day on trying to create the Metafont-logo example, the Metafont book Appendix E, on my PowerMac of 1996. In vain, without results. I got io.300gf and io.tfm, but lacked the (old) tools to go on.*
> *The other day it took me roughly half an hour to create Adobe's example Type 3 font, Redbook p100. The Adobe's process is less complicated and not lumbered by confusing and complicating bitmap-inheritances from the past. For the purpose of creating Wordart in the spirit of Jackowski&Ryćko the Adobe Type 3 process is good enough."*

Moreover, I experience the boxes-approach as too rigid, little flexible, hampering for example easy page-breaks with floating (misplaced) pictures as result, as well as the impossibility to use footnotes, endnotes or ... from within a box; a 21st century tool unworthy.

Compared with PostScript, TEX's macro language is more complex, as can be seen from the examples in the paper. But ... we have to live with it, in want for something

49. PSView is extremely fast as previewer, allows PS library inclusion via the run command as well, reacts elegantly on errors by showing the results so far and supplies error messages via a pop-up GhostScript window, but ... doesn't provide for `.pdf` output, alas.

50. Courtesy Péter Szabó, EuroTEX-ConTEXt2009.

simpler, better, but also open source and equally-well documented.
Let us not make it more complicated by adding too big and too complex software.

TIA-simpler-WTDI

Errors of pdfTEX

The logical error in pdfTEX is that it does not allow for EPSF inclusion. The use of PostScript via DVIPS had earned its place in the TEXworld: rich, powerful and much used.

Another weak point is the lack of maintenance.[51] To develop a software tool is one thing to maintain it is quite another. A big disadvantage of the volunteer world: lack of follow-up.

Errors of TEXworks

Sometimes lines disappear in the edit-pane, as if printed on 1 line??? Very unhandy, I can't even edit these hidden lines.

Wishes under MS XP, MS Vista or MS System 7

For TEXworks I would like menu options .eps → .pdf and .mp → .pdf.
A decent IDE for MetaPost and PostScript.
Better PDF-viewer in TEXworks.
Accurate BoundingBox values via pathbbox in 1-pass.
BLUe as format in TEXworks.
Maintenance pdfAllTEX, and to allow for PostScript in pdfTEX.

Post-Conference

After my presentation Herbert Voss showed me his PSTricks,[52] which is a continuation and extension of the work of Timothy Van Zandt and Dennis Girou. Impressive, very impressive! Especially his 3D extensions.

PSTricks uses (harnessed) PostScript under the hood. The user-interface strongly reminds me of LaTEX-'s picture-environment. As far as I understand it, Timothy just implemented LaTEX's picture-environment in PostScript, via (one-way) interfacing. This entails that LaTEX users did not have to learn something new and received better value. However, the drawback is that the graphics is not backed up by an imaging model, and nasty things from the picture-environment are inherited.

It can't be used with pdf(La)TEX, because pdf(La)TEX does not allow for PostScript. Undoubtedly, the longer processing path via PostScript can be included as menu item in TEXworks, my TEX-editor.[53]

$$\text{Script} \xrightarrow{\text{pdfTEX}} \text{PDF} \qquad \text{vs} \qquad \text{Script} \xrightarrow{\text{TEX}} \text{DVI} \xrightarrow{\text{DVIPS}} \text{PS} \xrightarrow{\text{Distiller}} \text{PDF}.$$

In principle I favour the 3-steps process, in practice I use the 1-step fast way.

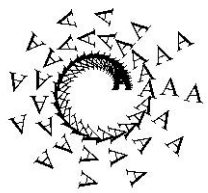
It's a pity that the code for the π-decimals picture, p294, has not been supplied in the book, so I can't compare it with my π-decimals code, as shown earlier and supplied in PSlib.eps.

Next best, I imitated Voss' example of rotated A's, more-or-less, which reminds me of Adobe's rotated word Adobe, Bluebook p98. The picture is also given in the Graphics Companion p357.[54]

In PSTricks' code is too much one has to remember to my taste, too many and too varied braces, {...} , (...), and [...] ... moreover, the data A has to be supplied three times.

51. When working with colours the weird \pdfliteral{1 0 0 0 k} and \pdfliteral{1 0 0 0 K} have to be included?!?
52. PSTricks —Graphics and PostScript for TEX and LaTEX. UIT Cambridge. ISBN 978-1-906860-13-4. The Graphics Companion devotes to PSTricks: Ch 5 Harnessing PostScript inside LaTEX, and Ch 6 The main PSTricks packages, p213–p466, all-in-all 253p.
53. Although I don't know how to do that at the moment.
54. Another example of text along a spiral, explicit, is on p451 of the Graphics Companion, which comes close to typesetting along an implicit spiral.

Personally, I abhor the (curly) braces mania, and favour minimal mark-up; providing the data A three times is not minimal.



```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Herbert Voss, p233
%%BoundingBox: -40 -40 40 40
%%BeginSetup
%%EndSetup
/Times-Roman 10 selectfont 0 0 moveto /kern 5 def
0 .5 33{rotate kern 1 moveto (A) show
        /kern kern 1.029 mul def}for
```

```
\usepackage{pstricks, pst-node, multido}
\begin{pspicture}(4.5, 3.5)
   \cnode*(2,2){4pt}{A}
   \multido{\nA=0+10. \rB=)+0.5}{110}{%
      \nput[rot=\nA, labelsep\rB pt]%
           {\nA}{A}{A}}
\end{pspicture}
```

Acknowledgements

Thank you Adobe for your maintained, adapted to LanguageLevel 3 since 1997, good old, industrial standard PostScript and Acrobat Pro (actually DISTILLER) to view it, Don Knuth for your stable plain TEX, Jonathan Kew for the TEXworks IDE, Hàn Thế Thành for pdf(La)TEX,

Thank you Bogusław Jackowski for supplying me with old artistic material from GUST, and some more.

Thank you Herbert Voss for your comments, that we have met and that we stay on speaking terms.

Thank you Jos Winnink and Henk Jansen for proofing an early draft and the latter also for proofing the final version. MAPS editors for improving my use of English and last but not least Taco Hoekwater for procrusting my plain TEX preprint note into MAPS format.

James Ensor's impression of recreational 'Breskens'

So long and thanks for all the fish.
My case rests, have fun and all the best.

Kees van der Laan
kisa1@xs4all.nl

Appendix: BoundingBox via pathbbox in 1-pass

The verbatim left shows my current trial-and-error cropping, while at right cropping is done on-the-fly in 1-pass, at the expense of providing the path double.

```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Cropping
%%BoundingBox: 0 0 115 23
%%BeginSetup
%%EndSetup
/Times-Roman 30 selectfont
/rays{120{0 0 moveto 108 0 lineto 1.5 rotate
     }repeat stroke}def
0 1  moveto (StarLines) true charpath clip
newpath 50 -15 translate rays
showpage
%%EOF
```

```
%!PS-Adobe-3.0
%%Title: One-pass cropping, LRM 2
/Times-Roman 30 selectfont
0 0 moveto (StarLines) false charpath flattenpath pathbbox
/ury exch def /urx exch def ...
/w urx llx sub cvi def /h ury lly sub cvi def
<</PageSize [w h]>>setpagedevice
newpath
/rays{120{0 0 moveto 108 0 lineto 1.5 rotate
     }repeat stroke}def
0 1  moveto (StarLines) true charpath clip
newpath 50 -15 translate rays
showpage
%%EOF
```

Appendix: Binary tree macros

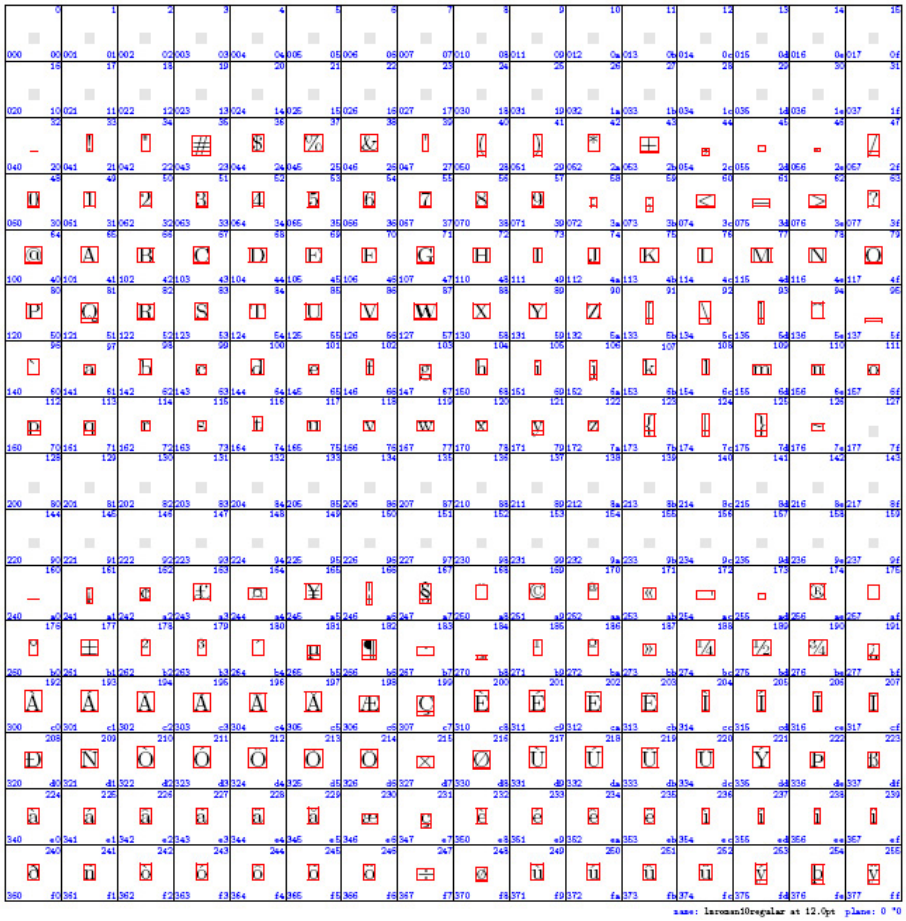
Included below are my stand-alone balanced binary tree TEX macros of old, taken from `tools.dat` with the necessary declarations from `blue.tex` added, next to my tiny, superior, and more clear PostScript variant on occasion of EuroTEX-ConTEXt2012.

```
\newdimen\x   \x0pt \newdimen\y   \y0pt \newcount\n   \newcount\k   \k0
\newdimen\unitlength \unitlength1ex     \newdimen\linethickness \linethickness1pt
\def\xy#1{%Function: place #1 at x, y
   \vbox to0pt{\kern-\y \hbox to0pt{\kern\x#1\hss}\vss}}
\def\xytxt#1{%Function: place text #1 at x, y
   \xy{\vbox to0pt{\vss \hbox to0pt{\strut#1\hss}\kern0pt}}}
\def\N#1{\xy{\kern-.5\linethickness
  \vbox to0pt{\vss \hrule height#1\unitlength width\linethickness}}%
  \advance\y#1\unitlength}
\def\S#1{\advance\y-#1\unitlength {\N{#1}}}
\def\E#1{\xy{\vbox to0pt{\vss
   \hrule width#1\unitlength
          height\linethickness
          depth0pt\vss
  }}\advance\x#1\unitlength}
\def\W#1{\advance\x-#1\unitlength{\E{#1}}}
%
\def\bintree{\E{\the\n}%
   \ifnum\n=2 \eertnib\fi
   \divide\n2 {\N{\the\n}\bintree}%
              \S{\the\n}\bintree%
   \multiply\n2}
\def\eertnib#1\bintree{\fi}

\let\Eold\E
\def\E#1{\global\advance\k1
  \xytxt{\csname\the\k\endcsname}
  \Eold8}

%data
\def\1{CGL}\def\2{GJHvN}\def\3{JLB}\def\4{PvO}
\def\5{PT} \def\6{EF}   \def\7{WD} \def\8{HH}

\offinterlineskip \n=8 \bintree
\bye
```

```
%!PS-Adobe-3.0 EPSF-3.0
%%Title: Binary Tree biased by Lindenmayer production rule
%%BoundingBox: -1 -250 250 250
%%BeginSetup
%%EndSetup
%%BeginProlog
/Bintree{% value of n on stack
  /n exch 2 div def
  E %draw East and add leave
  n 16 gt {currentpoint N n Bintree
          moveto        S n Bintree}if
   /n n 2 mul def}def %end Bintree
/N{0 n     rlineto}def
/S{0 n neg rlineto}def
/E{gsave  ntg k get 2 3 rmoveto show grestore
    /k k 1 add def
     60 0 rlineto
  }def
/Courier 16 selectfont /k 0 def
/ntg [(CGL) (GJHvN) (JLB) (PvO) (PT) (EF) (WD) (HH)
      (TH) (FG) (JW) (WD) (WE) (WW) (BJ)] def
%%EndProlog
%
% Program
%
0 -3 moveto 256 Bintree stroke showpage
```

The unbalanced tree jewel of old in cripple TEX, I should rewrite in PostScript, next to providing for viewing part of a huge tree, by some sort of window on the tree.

Appendix: Latin Modern Roman 16x16 font table (pane 1)



Above is included P0, with the usual glyphs (no euro, however), and some accented characters, AE ligature... Most of the 21 planes are nearly empty. The digits and the alphabet glyphs have the same digital address as in Knuth's 7bit table. Quite another question is how to use Latin Modern and TEX-Gyre.

P1: unusual accents and some double accented characters
P2: accents as such
P3: a few Greek symbols
P5: double embellished characters, 'accents, accents, accents ...'
P6: promille and euro with address 254, as well as pound sterling
P7: TM
P19: contains oldstyle digits beginning with address 060.

BLU does not much profit from all this generality; scientific communication in one language, English, simplifies enormously. Keep simplifying on your mind.