Graphics and \TeX—a reappraisal of METAFONT/MetaPost

Kees van der Laan

Abstract

It is all about the author’s first steps in METAFONT, for creating graphics to be included in \TeX documents, with a wink to MetaPost. The graphics comprises 2D pictures and 2.5D images of 3D objects via projection techniques. Learning METAFONT was much easier than learning \TeX. Included examples are: cat, Hilbert curve, and Gabo’s linear construction in space no 2. A few highlights on macro writing in METAFONT have been selected. The appendix contains the table of contents of my anthology.mf file of examples.

1 Introduction

The handling of graphics in \TeX scripts has a history. There are three approaches from the document preparation point of view:
- \TeX alone
- \TeX and METAFONT
- Use of ‘third party’ graphics tools.

Everybody uses epsf—encapsulated POSTSCRIPT—in the last two cases as medium to merge the graphics with the \TeX script in order to get the results out.

1.1 \TeX alone

\LaTeX’s picture environment is the common example for this class, though plain \TeX Xies might use the macros from Graham/Knuth/Pastashnik, which reflect a subset of \LaTeX’s picture functionality in plain \TeX X. In BLUe’s format system I also introduced the use of ‘Turtle graphics.’ I have used this in PWT—Publishing with \TeX X user’s guide—for simple fractals.

Interesting too is Gurari’s approach.

In scientific circles a problem is how to paste up mathematical graphs electronically. An approach is to calculate the graphs via Pascal or so and let Pascal generate the \TeX X code for the graph. A few years ago I shuffled and typeset bridge hands via this method.

These methods will not be dealt with in this paper.

1.2 \TeX and METAFONT

John Hobby recognized the power of METAFONT for the design of (systematic) graphics and married POSTSCRIPT’s outlines to METAFONT in his MetaPost, banning the bitmap approach. This is the path I’m on to emulate Naum Gabo’s constructive art.

Other approaches are for example:
- Jackowski’s mftoeps package to transform METAFONT files into POSTSCRIPT and vice versa
- Leathrum/Tobin’s mfpic which applies METAFONT’s character handling technique to export graphics in general.

Graphics via METAFONT is the main subject of this paper.

1.3 Use of third party tools

Of late more and more sophisticated graphics and multimedia software emerges, which allow interactive graphics among other things. Happily, import and export of POSTSCRIPT files is possible, and therefore the software—Adobe Illustrator, Photoshop, Coreldraw, etc.—can cooperate with \TeX X and METAFONT.

Of course one could use POSTSCRIPT throughout.

All of the approaches have their pros-and-cons. What to use—and when—depends as usual on your circumstances. The third party tools will not be dealt with in this paper.

2 METAFONT

Learning METAFONT was easier for me than to learn \TeX X. I picked up the flavour from Knuth’s (first) book in the field ‘\TeX X and METAFONT—new directions in typesetting.’ Next, I read METAFONT Book to absorb the ideas, possibilities and details. Finally, and inevitably, I exercised graphics examples borrowed from literature.

---

1 In the old ‘mainframe’ days documents were prepared with space left open for graphics and tables, prepared by other tools, to be pasted in.
2 Alas, there is no standard as yet for the use of \texttt{\special}s. I hope Rockiki will succeed with his proposed standard. The inclusion of epsf in (La)\TeX X documents will not be treated in this paper.
3 Used for typesetting their book Concrete Mathematics.
4 In this note the Hilbert curves of order 1 and 2 have been handled via ‘Turtle graphics.’
5 Designed under UNIX but also ported to DOS. AT&T has released MetaPost and add-ons in the public domain. Thank you.
6 And friends.
7 As opposed to systematical, reproducible, declarative graphics.
8 This resulted in a file called anthology.mf, which I can easily walk through on my Mac with BLUe Sky’s METAfont. It is not (yet) a database to load selectively from.
A superb survey of the language and what you can achieve with it, is given by Hobby in his ‘A user manual for MetaPost, AT&T, CSTR 162.’

Next to that there are the introductions: Tobin’s ‘METAFONT for Beginners’ and Jackowski’s GUST tutorial. The latter, alas, is in Polish.

During my building up of ’A graphics anthology in METAFONT,’ I found the path data structure an eye-opener. It shed new light on algorithms for the drawing of Hilbert curves, Sierpiński curves, Pythagorean trees, and ilks, which are formulated usually recursively.

However, use of the path data structure in combination with META FONT’s operation on pictures, like addto ... , yields elegant, concise and fast non-recursive programs.

### 2.1 MetaPost and extensions

MetaPost is (nearly) upward compatible with META FONT, and concentrates on graphics. It has banned the bitmap approach and combines the goodies of POSTSCRIPT with META FONT. MetaPost also provides for integration of text and graphics, next to suitable I/O.

Hobby’s graph extension has been treated in AT&T CSTR 164. It is all about typesetting scientific graphs, the functionality of troff’s grap recasted in MetaPost. The following functionalities are provided:

- automatic scaling
- automatic generation and labelling of tick marks or grid lines
- multiple coordinate systems
- linear and logarithmic scales
- separate data files
- ability to handle numbers outside the usual range
- arbitrary plotting symbols
- drawing, filling and labelling commands for graphs.

### 3 Why?

I have a keen interest in the works of Naum Gabo. When I heard of META FONT it occurred to me that I could emulate his works. To put it in another way. I was curious whether META FONT could be used conveniently as a design tool for 3D objects too. From a computer science point of view Gabo’s sculptures are very interesting, especially those composed of regular surfaces.

4D impression can be obtained from 1D information via the use of regular surfaces, projection techniques, and varying viewpoint angles.

I guess, if Gabo would have lived now, he would have exploited the use of computers, because programming a computer—professionally known as software engineering—is a constructivistic activity.

### 3.1 POSTSCRIPT straightaway

I have no convincing arguments against nor in favour. Perhaps, it is a matter of taste, biased by the stability and high-levelness of META FONT, next to the reputation of Don Knuth. On the other hand POSTSCRIPT is a de facto standard, POSTSCRIPT can be included in (La)TeX documents, and for the moment POSTSCRIPT is intermediate in the chain dvi→ps→PDF, and therefore a nodding knowledge of POSTSCRIPT is beneficial anyhow.

One thing for sure, however. Because of META FONT’s path data structure I uncovered a new coding for the Hilbert and similar curves, next to a systematic derecursion technique.

The question evaporates into thin air if we look at MetaPost as POSTSCRIPT with META FONT as user interface. Hobby has enriched MetaPost by the add-on tool graph, which provides troff’s grap functionality and a little more.

### 4 Examples

In the sequel a few excerpts from my META FONT anthology.

The included codes work as such on a Mac with BLUe Sky’s META FONT. How to code the pictures, unblurred by the ‘shipit’ details, was the purpose.

When the META FONT shipping out of characters is used the code must be adapted, such as enclosing it by \begin{char} and \end{char} and providing \begin{char} with the appropriate arguments as treated in META FONT Book.

For using MetaPost a few adaptations are needed such as deleting the bitmap operations \cullit, \screenstrokes, and ilks, and enclosing the picture by \begin{fig} and \end{fig}, or \begin{graph} and...
endgraph when the graph extension is used. The effect of reverse video via 
addto blackbackground also -blackpicture 
has to be adapted too, for example via the use of (white) colour.

4.1 Cat

This example is all about the use of a pen of varying width. It gives an impression of what can be attained by META-
FONT/MetaPost with respect to classical drawing.

4.1.1 METAFORENT program

The file is 70 lines. Consult anthology.mf.

4.2 Hilbert curve

In Wirth’s ‘Algorithms + Data structures = Programs’ the drawing of Hilbert and Sierpiński curves have been treated 
as essentially recursive. Via the path data structure and 
copying and rotating of pictures built so far, it can be nicely 
derercursi®ed.

A Hilbert curve consists of 4 (rotated) copies of a base ele-
ment connected by 3 straight lines, the 3 edges of a square. 
$H_0$ is a dot. The base element of $H_k$—a Hilbert curve of 
order $k$—is $H_{k-1}$, $k = 1, 2, \ldots$. 

Below $H_1$ and $H_2$ have been drawn.\footnote{Done in \TeX by means of BLUe’s format Turtle Graphics.}

\begin{figure}[h]
\centering
\includegraphics[width=0.2\textwidth]{hilbert_curve.png}
\caption{Hilbert curve}
\end{figure}

The above picture has been obtained in the spirit of graphics 
via \TeX alone as follows.

\begin{verbatim}
\unitlength5ex
\hbox{\vss\W1\S1\E1\vss\vss}
\kern25ex
\vbox to3\unitlength{\offinterlineskip
\vss\W1\N1\S1\E1
\vss\vss}
\endverbatim

4.2.1 METAFORENT program

message "Hilbert curve, variant; %
the path is built up and drawn at 'one stroke'";
%METAfont experiments, code builds upon
%Wirth’s A+DS=P, pp130-133 (more concise,
% uses path data structure; no redoing
% of already constructed paths.)
%The code has been adapted to build up
%the "path", and is non-recursive.
%December 1995, cgl@rc.service.rug.nl.
%Kees van der Laan, Hunzeweg 57, 9893PB,
%Garnwerd, Holland
tracingstats:=1;
proofing:=1;screenstrokes:autoarounding:=0;
pickup pencircle scaled 0.2pt;
def openit = openwindow currentwindow
from origin to (screen_rows,screen_cols)
at (-40s,15s)enddef;
%
path p; s=10;
sz=0;p:=origin;%H_0 size and path
n=5; %Order of H-curve
for k=1 upto n:%H_1,...H_n consecutively
p:= p transformed (identity rotated 90
reflectedabout (origin,up))--
p shifted ((-sz-1)*s,0)--)--
p transformed (identity rotated -90
reflectedabout (origin,up)
shifted (-sz*s,(-2sz-1)*s));
sz:=2sz+1;
clearit;draw p; showit;
endfor
end

Remarks. Order 4 overflows the ‘rounding table’ limit 
(300) with the default autoarounding. The connecting 
straight lines are implicit via --.

4.2.2 POSTSCRIPT program

Joseph Romanovsky has transliterated my (recursive) 
METAFORENT code into the following concise POST-
SCRIPT program.

\begin{verbatim}
/S{0 R rlineto currentpoint stroke moveto}def
/T(90 rotate)def /TM{T 1 -1 scale}def
/H{TM dup \begin{verbatim}
0g t{ 1s ubHST MHSHTS
-1 1 scale H 180 rotate 1 add} if
TM}def
/R 8 def
100 100 moveto
6 H pop
showpage
\end{verbatim}
\end{verbatim}

\end{verbatim}

\end{verbatim}

15Done in \TeX by means of BLUe’s format Turtle Graphics.
4.3 Linear construction in space II

Thirty years ago I was caught by Gabo, by his constructive art, especially by his ‘Linear construction in space’-like objects.

From the METAFOnt viewpoint this example is all about how to handle a 3D object, that is how to describe, project and draw Gabo’s 3D constructive art.

How to do this? How to develop a general technique?

4.3.1 Design

It is needed to view the object from different angles. This entailed the use of projection techniques.16

While programming the object in METAFOnt I had to solve

| a | how to transform a curve in space? |
| b | how to preserve shape under projection of (a discretisation of) the curve while joining the projected points by METAFOnt’s splines? |
| c | how to create equidistant points along a curve? |
| d | how to avoid blurring lines? |
| e | how to emulate the used (perspex) material? |

4.3.2 Coding

I solved b and c by first creating the basic shape of a boundary in 2D, and then replace the curve by a set of (nearly) equidistant points along the curve. The coding of c reads essentially as follows, where p10 takes over from p1.17

\[
p10:=\text{for } t:=1 \text{ upto } 19: \quad \text{point} \ .05t \text{ of } p1. \text{endfor origin};
\]

The rotation of a boundary curve, for example from the yz-plane into the xz-plane, is coded simply as follows, where p100 takes over from p10 etc.

\[
\%\text{in yz-plane (the screen)}\\p100:=\text{for } k=0 \text{ upto } n-1:\quad \text{pointtopair}(0,\text{xpart(point } k \text{ of p10}), \quad \text{ypart(point } k \text{ of p10))).endfor pointtopair(0,0,0);\]

\[
\%\text{in xz-plane}\\p200:=\text{for } k=0 \text{ upto } n-1:\quad \text{pointtopair}(\text{xpart(point } k \text{ of p10), } 0, \quad \text{ypart(point } k \text{ of p10))).endfor pointtopair(0,0,0);\]

The projection is done via pointtopair, which in its simplest version reads as follows.

\[
de\text{f pointtopair(expr x,y,z)=}\\\%\text{Purpose: The projection of a 3D point}\\\%\text{into a pair in the projection plane.}\\\%\text{Arguments: } x,y,z \text{ coordinates of a point}\\\quad (-.6x+.8y,-4/13x-3/13y+12/13z)\quad \text{enddef;}
\]

Some of the regular surfaces blur the picture.18 To clear this up I removed the ‘hidden’ line parts. To erase what is hidden is determined by the boundary of what is in front.19

\[
\text{erase fill p100..reverse p200..cycle;}
\]

The regular surface which is full-blown in sight has been drawn simply via

\[
\text{for } k=0 \text{ step } 1 \text{ until } n:\quad \text{draw point } k \text{ of p100..point } n-\text{k of p200; endfor}
\]

To emulate the ‘light’ caused by the material perspex I used reverse video as explained in METAFOnt Book 115, 118 for the ‘dangerous bend.’

4.3.3 METAFOnt program

The file is 120 odd lines. Consult anthology.mf.

5 Macro facilities

Macro writing in METAFOnt is completely different from macro writing in \TeX{}. This note is not aimed as a tutorial on macros, it provides a few highlights. An appetizer.

16 See Lauwerier’s ‘Meetkunde met de micro computer,’ for an introduction. It contains many examples in .BASIC.

17 Note that paths of dynamical length—that is determined at runtime—are created. I fine-tuned this by creating really equidistant points along a curve via the use of METAFOnt’s solve and Hobby’s arclength.

18 In reality Gabo’s aim was that from every viewpoint the object could be seen completely. There are no ‘hidden lines’ in his art. He achieved this by using perspex and nylon.

19 Knuth uses overdraw—see for example ex13.11—which is very nice. His watchband logo has not been made of perspex apparently.
5.1 FIFO
My favourite FIFO paradigm—first-in-first-out—is implicit in the (var)def parameter handling. For example the macro \( \text{max} \) allows as argument a list of undetermined length.

\[ \text{max}(a) \text{ max}(a,b,c) \]

Remark. A variable number of arguments is common in METAFONT, for example \( \text{definepixels} \) and ilks can be invoked similarly. This is a consequence of (the abstract) \( \text{text as parameter `type.'} \) MetaPost's \( \text{buildcycle} \) macro makes use of this feature too in allowing a list of paths, of undetermined length, as argument. Useful it is.

5.2 Generic macros
For the \( \text{max} \) macro, for example, the type of the arguments can be either numeric, pair or string. This is possible because METAFONT allows for testing for the type of an argument. Powerful this generic—the same macro for all relevant types—feature.

Neat, that abstraction of type and number of arguments, and definitely in agreement with Knuth's aim 21

‘The rules are intended to work the way you expect them.’

5.3 Gobbling
Another unusual feature is the infix primary \( \text{gobbled} \), which not only absorbs the argument after but also before.\(^22\) Infix operators can be defined with primary, secondary or tertiary level of precedence.

5.4 Clipping boundary
Clipping is not provided as such by METAFONT. However, with \( \text{cullit} \) and \( \text{cull} \) a picture can be clipped. Below the current picture is confined to a (scaled) square, and provided with a fret.

\[ \ldots \text{picture so far} \]
\[ \%\text{reduce all pixels to 0 or 1} \]
\[ \text{cullit;} \]
\[ \%\text{make pixel value of picture 2} \]
\[ \%\text{within the square} \]
\[ \text{fill unitsquare scaled 100;} \]
\[ \%\text{retain picture within the square} \]
\[ \text{currentpicture keeping (2,2);} \]
\[ \%\text{draw the boundary} \]
\[ \text{draw unitsquare scaled 100;} \]

The clipping by a square boundary is just an example to convey the idea. The approach can be applied to all kinds of shapes, for example to a ring as done by Jackowski in his Euro\TeX\ 96 paper.

5.5 Length of a curve
The macro length can be applied to a path with as result the maximum ‘time,’ not the arc length. Hobby applied Simpson’s quadrature rule, which results in the following concise approximation, because we know the formula of the Bezier spline.\(^23\)

\[ \int_0^1 \| \mathbf{B}' \| \, dt \approx \frac{1}{5} (\| z_0 \| + \| z_0 + 2z_1 + z_2 \| + \| z_2 \|) \]

\[ \text{vardef lengthpath expr p=save dz; pair dz[];} \]
\[ \text{dz0=point 1 of p - point 0 of p;} \]
\[ \text{dz1=point 2 of p - point 1 of p;} \]
\[ \text{dz2=point 3 of p - point 2 of p;} \]
\[ \text{.5(length(dz0)+length(dz0+2dz1+dz2)+} \]
\[ \text{length(dz2))} \]
\[ \text{enddef;} \]

If the path’s name is \( p \) an invoke might read \( \text{lengthpath} \)
\[ p \text{, or for a subpath lengthpath(subpath (3,6) of p).} \]

5.6 Selective loading
In BLUe’s format system the mechanism of selective loading has been used to build a database of tools, pictures and ilks. This functionality can be implemented in METAFONT too. The file to load selectively from consists again of triples: list element tag, a symbolic token, and text enclosed by parentheses and ended by a semicolon. The list element tag macro has 2 arguments: the implicit suffix—which is compared as string with the name of what we want to select—and the text enclosed by parentheses. If the suffix agrees with the required name, a macro of this name is defined with the text as replacement text. For example the definition of the list element tag might read as follows.

\[ \text{vardef lst@#(text t)=} \]
\[ \text{if str @#= s:%s=selection key} \]
\[ \text{def @#= t enddef} \]
\[ \text{fi enddef;} \]
\[ \text{input macro.lst} \]

The (toy) file \( \text{macro.lst} \) might read as follows

\[ \text{lst na (draw unitsquare scaled size;);} \]
\[ \text{lst ns (rt);} \]
\[ \text{The result with s:="na";} \]
\[ \text{reads} \]
\[ \text{def na=draw unitsquare scaled size;} \]

6 Pitfall in compatibility
The METAFONT program for my cat processed by Meta- post does not yield the correct result. The moustache has disappeared?! If we move the moustache code to the end then the correct result is obtained.

What has happened?
In my opinion this is a consequence of that pixels have values, to quote METAFONT Book 109
Pixels aren’t simply ‘on’ or ‘off’ when METAFONT is working on a picture; they can be ‘doubly on’ or ‘triply off.’ Each pixel contains a small integer value, ...

Metapost’s path is apparently just on or off. A draw <path> followed by an overlapping fill and unfill makes that the path disappears. It is better to let the draw <path> follow the fill and unfill.

Example (Ring with center)

The compatible way to program this in METAFONT reads essentially as follows.

fill fullcircle scaled 10;
unfill fullcircle scaled 7;
drawdot origin;

7 Summarizing my experience

MetaPost combines the best of both worlds: METAFONT’s language features are enriched with contours and epsf output. Moreover, it allows access to POSTSCRIPT’s wealth.

MetaPost can be looked upon as POSTSCRIPT with a METAFONT user interface to create pictures to be included in (La)TeX or troff documents.

Maybe METAFONT/MetaPost will find their niche in history as convenient tools to describe pictures concisely and at a high level.

7.1 What I like of METAFONT

First of all I like very much the quality, stability and its being for free. Next it is available on nearly every platform. Finally, there are the following pleasing details.

- declarative nature of the language
- the meta-ness and the generic aspects
- the generalization of variable, subscripted variable or record field variable into ⟨tag⟩⟨suffix⟩
- just 3 kinds of arguments: expr (independent of type), suffix, and text
- the operator definitions with built in priorities (primarydef etc.)
- pen, path and picture data structures and operations
- filling and erasing operations
- operations for intersection points
- (nonlinear) interpolation between curves
- various handy ‘syntactic sugars,’ and
- rich tracing facilities.

If Descartes’ Analytic Geometry is still taught today, it might benefit from METAFONT in visualizing.

7.2 What I missed in METAFONT

In the list of missing items below those denoted by + are provided in MetaPost.

+ outlines or epsf output (it is all about bitmaps of fonts)
+ mixing of text and pictures
+ general file I/O (writing and reading of pictures)
+ a suitable number range (restricted to the half-open interval \([1/256/256\rightarrow 4096])
+ generality of rotating pictures (restricted to a multiple of 90°)
+ dashed/dotted kind of lines
+ arrows
+ clipping
+ shading and greyscales
+ colour
+ to invoke \TeX
+ to make use of POSTSCRIPT facilities
- triple datatype—point in 3D as analogon of pair
- tree datastructure (pointer/handle).

Then there is Hobby’s ‘graph’ extension with functionalities I did not miss yet, but which I do need for sure when typesetting scientific graphs, gracefully.

7.3 METAFONT/MetaPost as production tool?

History has it that \TeX has been used mainly by scientist with substantial complex copy full of mathematics, tables, or graphics, who wish to publish via the electronic networks, via internet, not in the least in the creative off-off self-publishing world. METAFONT has been used mainly for font production by off-off linguists for non-Latin alphabets.

I don’t know of better tools for document production which are so reliable, portable, ubiquitous, open, completely documented, stable, and cooperative towards other tools. With respect to the latter one can think of the various printer and screen drivers, POSTSCRIPT and PDF, and the new hype HTML—HyperText Markup Language. Moreover, the twins \TeX&METAfont (and descendants) are in the public domain and ported to every platform.

However, one has to learn the systems, to know what is under the hood. \TeX&METAfont are not of the push-the-button type tools, like washing machines or cars. Therefore, education is paramount.

The METAFONT/POSTSCRIPT experts of the Polish \TeX User Group GUST have reported that the necessary condition for METAFONT/MetaPost to become a production tool is that epsf can function as medium to ease the cooperation with third party tools. As far as I know BoP s.c. is the only company with METAFONT in production.

\footnote{It is nearly upwards compatible. \texttt{cullit} and other bitmap operations have to replaced.}

\footnote{Agreed, a necessary evil.}

\footnote{Not to mention Hoenig’s typesetting along curved paths.}

\footnote{Enumerated earlier.}

\footnote{Bogusław Jackowski and Piotr Pianowski.}
### 7.4 The future

Maybe we should no longer paste up for figures. What about creating hyperlinks to a picture database? If we want to see the picture we can just click and there it is. This is similar to when we like to hear the music when we read about a composer or read the music notation.

Whatever these new ways will bring, I for one like the complete document on paper.

### 8 Acknowledgments

Thank you Joseph Romanovsky, Geoffrey Tobin and John Hobby for your friendly support. Piet van Oostrom processed ‘linear construction no. 2’ via MetaPost. The picture came out for sure, at the expense of a large POSTSCRIPT file.

Jos Winnink transcribed the METAfont pictures into MetaPost. He also lend the usual helping hand in transforming the BLUE’s scripts into MAPS submissions.

For the details of the references search the file lit.dat which comes with BLUE’s format system.

Have fun, and all the best.

### 9 Appendix: ToC METAfont anthology

I hope that the codings mentioned below will contribute to the METAfont ‘literature.’

% Anthology of METAfont endeavours,
% Started November 1995, last change March 1996
% Version 0, March 1996.
% Author/Composer: Kees van der Laan,
% Hunzeweg 57, 9883PB, Garnwerd, Holland
% cgl@rc.service.rug.nl.
% This file works under BLUE’s METAfont
% (Public Domain) on the Macintosh.
% How to scroll through these examples on
% other systems I don’t know.
% Just copy what is of interest for you and
% emulate...
% towards a discipline of METAfont/MetaPost.
% For other systems copy the example(s) you
% are interested in build a character
% from it and follow the usual procedures to
% include the font in your (La)TeX,
% or use for example the Jackowski/Rycko
%mftoepsf package to produce epsf.
% For use in MetaPost enclose the elements
% by beginfig ...endfig, I presume.
%
% Disclaimer borrowed from
% Preface METAfont book:
% ‘That’s the way how it is with any
% powerful tool:
% There is always more to learn, and
% there are always better ways to do
% what you have done before.’

% Table of Contents
% Preliminary, file: ant.mac
% Macros used
% (default) openit
% pointtopair projection
% dash
% turtle movements: east, south, west, north
% Defaults
% ToC proper

% % Anthology of METAfont endeavours,
% Started November 1995, last change March 1996
% Version 0, March 1996.
% Author/Composer: Kees van der Laan,
% Hunzeweg 57, 9883PB, Garnwerd, Holland
% cgl@rc.service.rug.nl.
% This file works under BLUE’s METAfont
% (Public Domain) on the Macintosh.
% How to scroll through these examples on
% other systems I don’t know.
% Just copy what is of interest for you and
% emulate...
% towards a discipline of METAfont/MetaPost.
% For other systems copy the example(s) you
% are interested in build a character
% from it and follow the usual procedures to
% include the font in your (La)TeX,
% or use for example the Jackowski/Rycko
%mftoepsf package to produce epsf.
% For use in MetaPost enclose the elements
% by beginfig ...endfig, I presume.
%
% Disclaimer borrowed from
% Preface METAfont book:
% ‘That’s the way how it is with any
% powerful tool:
% There is always more to learn, and
% there are always better ways to do
% what you have done before.’

% Table of Contents
% Preliminary, file: ant.mac
% Macros used
% (default) openit
% pointtopair projection
% dash
% turtle movements: east, south, west, north
% Defaults
% ToC proper
%-Pythagorean tree, non-recursive
%-Sierpinski square (Courtesy B Jackowski)
%-Sierpinski triangle (recursive
% with use of save)
%-Sierpinski triangle (without save)
%-Sierpinski carpet (via gambling)
%
%%%Various, file ant.var
%-Haralambous’ deformations of a ‘circle’
%-Calculation of (arc)length of a Bezier
% segment via quadrature
%-Exercise from MB: p.176 (solve use)
%-Exercise from MB: 13.8 (star)
%-Exercise from MB: 13.10 (S-figure)
%-Exercise from MB: 13.10 (S-figure
% contour via ‘expanded stroke’)
%-Exercise from MB: 13.11 (Moebius band)
%-Exercise from MB: 13.11 (variant
% 2.5D Moebius band)
%-Exercise from MB 14 p134 (Interpath use:
% heart interpolation)
%-Exercise from MB: 15.6 (variant,
% via paths)
%-Exercise from MB: 20.5 (variant,
% parameter separator TeXniue?)
%-n faculty (exercise recursion and number range)
%-n asterisks(exercise recursion)
%-quicksort (exercise recursion)
%%%end ToC