Experiments in teaching METAFONT

Richard Southall
Computer Science Department
Stanford University
Stanford, CA 94305, USA

Jacques André
IRISA/INRIA-Rennes
Campus de Beaulieu
F-35042 Rennes Cedex, France

Abstract

In January 1985 a one-week winter course was organized in Rennes, France on Typography and Computer Science. The course was mainly devoted to type design. A set of lectures and practical work was organized on METAFONT *and the design of typefaces.*

METAFONT *differs from other systems of character digitization because a* METAFONT *program tells an output device how to draw the characters it describes, rather than simply what shapes they are. Designers working with it are therefore faced with problems in three areas: the normal problems of type design, the normal problems of font production and the special problems that arise from the nature of* METAFONT.

The practical work on the course was intended to investigate one important problem from each of these areas. Participants were given METAFONT *programs that produced elementary shapes on a printer/plotter. They were asked to alter the programs and make visual assessments of the resulting output.*

The present paper explains why METAFONT *was taught in this way, discusses some of the problems that were encountered, and shows some results of the practical work.*

TEX for Scientific Documentation D. Lucarella, Editor

1 Introduction

Between Gutenberg's time and the beginning of the 1980s, there were really only three truly significant events in the history of the manufacture of printers' type. In the years following 1882 the cutting of steel punches, which was the first stage in the production of metal type, changed from a handicraft to an industrial process. Between 1965 and 1970, the advent of photocomposition changed the product of the type-composing process from a three-dimensional assembly of pieces of metal to a two-dimensional assembly of character images on photographic paper or film. And from about 1977 (though the technique had been in commercial use since 1966) these character images were synthesised in the photocomposing machine by electronic means, rather than being provided as prefabricated master images on a photographically-produced character matrix.

With the arrival of extremely fast microprocessors and high-performance reproducing devices (laser printers and bit-mapped visual displays) at lower and lower cost [Bigelow 85], printers' type has moved out of the printing works into the office systems environment. More and more people, most of them without any experience of letterform design or type manufacture, are becoming involved in the production of digital type: designing systems to produce digitized characters, and designing the characters themselves.

The central problem in industrialized type manufacture has always been the problem of communication between the designer and the producer of the type. This problem is especially important in today's situation, because the producers of type, instead of being people, are now systems of computer programs and the machines on which they run. Among these systems, METAFONT is particularly significant because it is more than a way of describing the shapes of characters: a METAFONT program contains a schematic description of how to produce a series of fonts. Using METAFONT to make technically satisfactory fonts requires of the user both craft knowledge (for deciding on the shapes of the characters in the fonts) and programming skill (for writing the programs that specify the shapes). Using METAFONT well, and teaching it, are therefore both difficult enterprises.

The present paper describes an approach to teaching METAFONT that was used during the winter course on Typography and Computer Science at Rennes in January 1985.

2 The 1985 winter course on typography at Rennes

In 1983 a symposium on Document Preparation Systems was organized in Rennes [André 83]. The participants and lecturers were either typographers and type designers (like Charles Bigelow, Ladislas Mandel and Adrian Frutiger) or computer specialists (like Peter Hibbard, Brian Reid, Patrick Baudelaire and C. Newmann). However, it was noticed that these two categories of people did not mix together very much, and the feeling was that typographers and computer scientists do not speak the same language. It was therefore decided to organize another course to bring computer scientists and typographers (or graphic designers) together a second time. This course was originally planned to take place during the summer of 1985, but it had to be held earlier, during the winter of 1984-85.

The course was organized by the Institut National de Recherche en Informatique et en Automatique (INRIA). It was held at the Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), an INRIA laboratory for computer science located on the University campus at Rennes. The course contained the following parts [André Sallio 85]:

- Lectures on type design (by Ladislas Mandel and Richard Southall), after an introductory survey by Jacques André and some discussions on legibility;
- Lectures on character description techniques and font handling (discussions of spline curves and character digitization algorithms by Gérard Hégron, the PostScript page description language by Patrick Baudelaire, and laser printer interfaces by Roger Hersch);
- Two lectures on commercially important systems for the production of digital type: IKARUS by Peter Karow and the Camex/Bitstream Letter Input Processor (LIP) by Jim Flowers;
- Lectures on copyright questions (Nicole Croix) and the problems of handling non-latin scripts on computers (Jacques Piolle);
- and finally three lectures and four sessions of practical work on type design and METAFONT by Richard Southall.

3 Why METAFONT?

On such a course, it would evidently be valuable to have practical experience of a system for character digitization and font production. But which one? At the time the course was planned there were only a dozen or so systems extant for handling high resolution digital fonts [Ruggles 83], and new ones like IMP [Carter 84] were not yet available.None of the extant systems was really useable in a course context: they were too expensive, or unsuitable for use by a group; and in general they were too thickly veiled in commercial secrecy. Thus special attention was paid to METAFONT [Knuth 79], which, like TEX, belongs more or less to the research community. Alas, the first version of METAFONT (METAFONT79) was written in SAIL, a rather exotic language that runs only on DEC-10 computers and hence was not available at Rennes. Fortunately, however, a new version of METAFONT was being developed that was supposed to be as portable as TEX82. Since the new version of METAFONT was announced in 1984, a few months before the course, we decided to go for this new product.

4 The 1984 Stanford course on METAFONT

A course on 'Topics in digital typography' was organized at Stanford University during the Spring quarter of 1984 [Knuth 84]. The subject of this course was type design in general and the use of the new METAFONT in particular. It was taught by Charles Bigelow, Donald Knuth and Richard Southall. Southall's lectures were on the general topic of 'Designing typefaces'; Bigelow discussed the history and development of letterforms from ancient times to the present day; and Knuth lectured on the new METAFONT, which he was developing while the course was in progress. The first homework assignments were exercises done with cut paper, to illustrate the important differences between 'what you see' and 'what is there'. The first METAFONT assignment was to draw parts of El Palo Alto (the tall tree) for a new version of the Stanford symbol. During the second METAFONT assignment, each student was asked to create two letters of a new typeface, using macros written by Knuth in the new language that drew pen-like strokes and arcs. The last assignment was to design a set of eight characters with METAFONT, that could be used for setting typographic borders. Some results of this work appear in [Knuth 84].

5 Installing and running METAFONT

Once we had decided to try to teach METAFONT at Rennes, it was clear that we had to have practical exercises with it on a computer. Initially, we thought of using METAFONT on Sun workstations, as at Stanford. Although METAFONT is essentially batch-oriented rather than interactive, a high-resolution screen like the Sun's allows the output of a METAFONT run to be displayed next to the program that generated it. Unfortunately, the Sun workstation was not yet available in France at that time; and an interactive version of METAFONT for the Perq was still being developed [Leitch 85]. It was therefore decided to install METAFONT on a VAX running VMS at the IRISA, with a Versatec printer/plotter as output device. However, many problems arose, particularly in the few days before the course began:

- Both the VAX with VMS and the Versatec were new equipments at the IRISA. There was no wizard easily available who had the ability to make a quick installation of the TEX system (parts of which were needed to process METAFONT output for the Versatec). Fortunately a VAX specialist was able to come from

Lausanne to help. In the event, it took more than a week to get TEX running, since we had problems with the distribution tape and very little information on installation procedures.

- We were not able to get a distribution tape of METAFONT. Eventually, thanks to Lynn Ruggles' efforts at Stanford, we obtained a pre-release version. This worked well, except that the 'gray fonts' for making proof-mode output on the Versatec were missing. However, programs for making these were on the tape, and we successfully made both 'gray' and 'black' fonts for the Versatec with METAFONT.
- The VT100 terminals we were waiting for arrived too late: so we had to use old Questar terminals that could not handle the VMS screen editor, and had to use the line editor instead. In terms of running the practical classes, this gave us our greatest difficulties.
- We planned to use at least 15 terminals (one for each pair of participants). Because the VAX had only 8 ports, we had to connect the terminals via the IRISA NetOne network. We needed a special cable for the Versatec, so that it could work at reasonable speed in the room where the course was held: this cable did not arrive until after the course had begun. The VAX had no spooler for the Versatec, so that only one file could be sent to it at a time. This was a major difficulty until Jorge Eggli made us a 'pseudo-spooler', that interrogated the Versatec repeatedly on behalf of each user who was waiting to have a file printed.
- The VAX was being used by IRISA personnel for their own research during the course, and at times it was rather slow. We even had a power failure on the NetOne network.

In absolute terms, we had fewer technical problems at Rennes than there were on the Stanford course [Knuth 84; personal experience (RS)]. However, because of the limited capability of the IRISA equipment, the intensity and short duration of the course and the relative lack of technical support, the problems seemed more acute at Rennes than they did at Stanford.

6 Teaching type design and METAFONT

METAFONT is a programming language for instructing a computer to draw shapes on a bit-mapped output device. It differs from other systems like IKARUS [Karow 83] or LIP [Flowers 84] not only because it is batch-oriented, but also because a METAFONT program tells an output device *how to draw* the characters it describes rather than simply *what shapes* they are. Because METAFONT is primarily intended for the production of new designs rather than the reproduction of existing ones, designers working with it are faced with problems in three areas:

- the normal problems of type design;
- the normal problems of font production;
- the special problems that arise from the nature of METAFONT, where shapes that designers normally make manually and assess visually have to be described in symbolic form.

Richard Southall's lectures at Stanford had been mainly concerned with trying to explain the distinctions between the different tasks involved in what is generally described as 'type design'. The experience of the Stanford course showed (not altogether surprisingly) that there tended to be great differences in habits of thought between computer scientists and graphic designers. We decided to continue at Rennes the attempt to discuss the design process in terms which would be comprehensible to non-designers.

We planned to tackle this task by two means:

1 A series of three lectures about the nature of the type design task [Southall 85b]. These lectures were adapted and condensed from lectures given in the Stanford course. They had the titles *What is a typeface?*, *Quality criteria for typefaces*, and *What does a type designer have to do?*.

2 Practical work with METAFONT, to investigate one important problem from each of the areas of type design, font production and the symbolic description of shapes.

Because of the short duration of the course, we could only devote three afternoons and part of a morning to the lectures and practical work on this subject. For logistical reasons, we had to divide the participants into two groups for part of the time: this meant that each group had a total of about 10 hours for this part of the course.

6.1 Information about METAFONT

Because of the lack of time, it was out of the question to adopt a formal approach to teaching METAFONT, as Knuth had on the Stanford course. We therefore decided to let people learn about METAFONT by osmosis. The participants were given the following texts:

- A draft version of the first five chapters of The METAFONTbook [Knuth 85] as it was available in December 1984;
- A copy of Knuth's paper *The Concept of a Meta-Font* [Knuth 82] in a French translation [Knuth 83];
- A paper on 'METAFONT and the problems of type design' [Southall 85a].The latter two items formed part of the *support de cours* [AndreŠallio 85].

We should say straight away that the course participants did not read these documents soon enough. For this reason, and because of the technical difficulties described earlier, the first session of practical work was rather turbulent. We should have scheduled at least an hour for an introduction to METAFONT, and some additional time to discuss the programs that the participants were given for the practical project.

6.2 The practical work

Once again because of the lack of time, it would have been absurd to ask the course participants to design a complete typeface, and we felt that making individual characters in the absence of a well-understood unifying graphic idea lacked pedagogical value. On the other hand, it seemed relatively easy to investigate some of the basic problems of type design, font production and symbolic description by generating and manipulating elementary shapes by means of METAFONT programs and looking at the results.

The participants worked in pairs. To minimize the difficulties arising from the use of the VMS line editor and computer terminals with no graphic capability, particular attention was paid to mixing participants coming from the world of graphic design with experienced computer users.

Each pair was given a computer account, some files containing METAFONT programs to produce elementary shapes, and listings of these programs and of Knuth's **plain.mf** macro file [Knuth 85].

The brief for the practical project is given in Appendix A. In outline, the objective was to make elementary shapes – solid and hollow rectangles, circles and triangles – by running METAFONT programs; to change the values of parameters in the programs to alter the shapes that were produced; and to assess their appearance visually. Participants were asked to make sets of shapes – first solid, then hollow – that when viewed from about 5 metres away appeared to be equal in size, aligned at top and bottom, evenly spaced and (for the hollow shapes) equal in apparent boldness. Participants were asked to put their output up on the walls in order to assess it visually, and to try to find reasons for the visual effects they observed.

(The final objective described in the project brief, to produce a set of technically satisfactory glyphs, could not have been achieved because we lacked some of the necessary software. In any event, very few of the participants reached that stage.)

7 Some results

Beside the 10 lecturers, 54 people took part in the course. Of these, 22 were connected with typography or graphic design (printers, publishers, teachers in colleges of art). The others came mainly from university research centers or from software development houses.

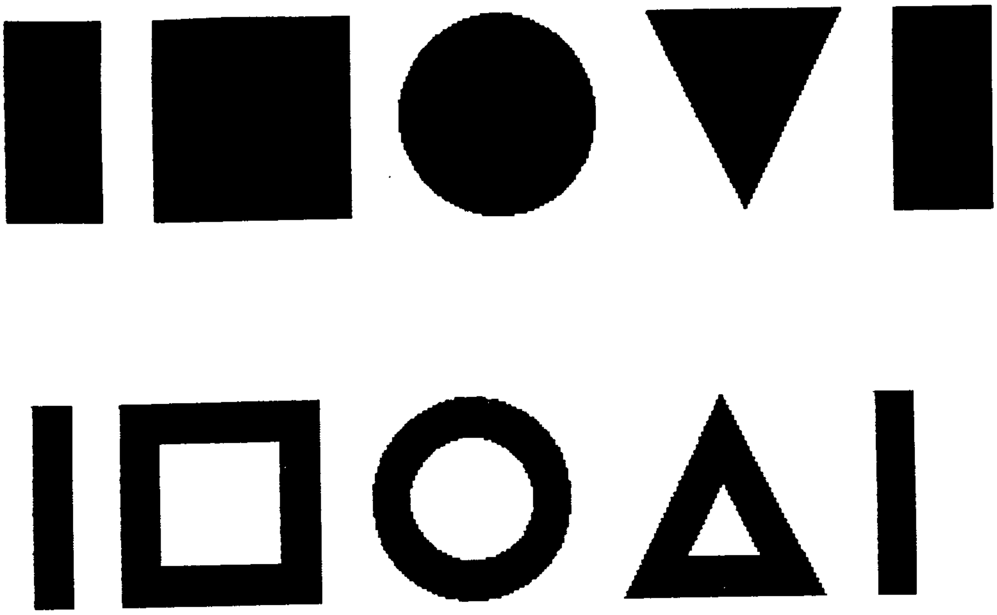

fig.1

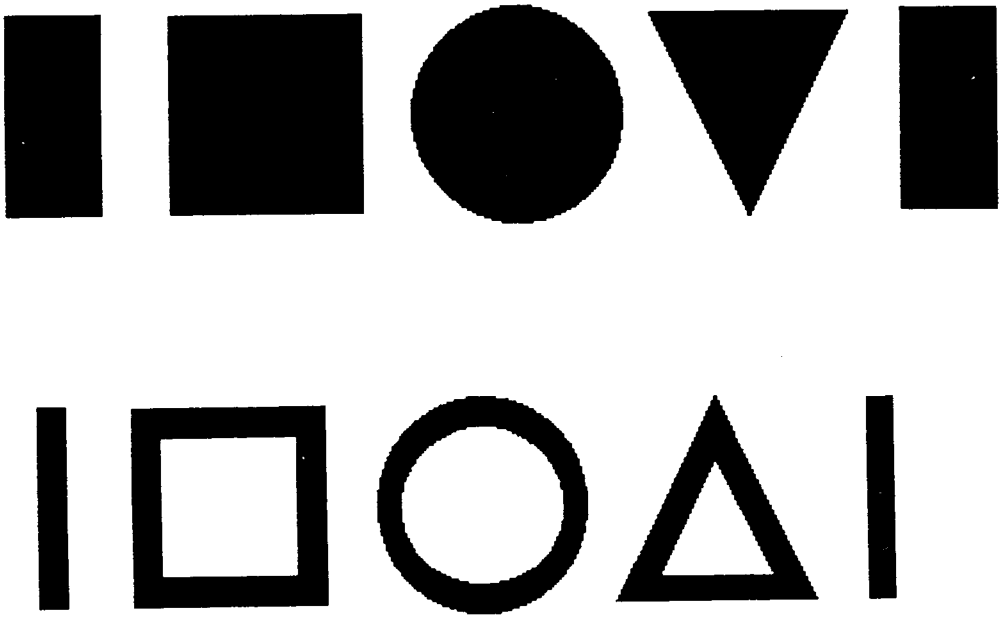

fig.2

The evaluation forms distributed at the end of the session showed that the participants had enjoyed the lectures. In particular, many computer scientists discovered a completely new world.

What about METAFONT? The lectures on type design were successful, and an extra one was added on the last day. The practical work, on the other hand, was not a complete success. Some of the reasons for this are as follows:

- As we said in section 5, the system (particularly the part that sent files to the Versatec for printing) was not working properly in time for the first session: some people got discouraged.
- As we said in section 6.1, many participants did not read the documentation about METAFONT before the first session, and so did not understand what was being asked of them.
- Surprisingly, computer scientists are not good at putting their own work up on the wall and discussing it. In consequence, the 'studio atmosphere' and the spirit of group involvement on which this particular project depends for its complete success never materialized.

In spite of this, many participants made correct modifications to the shapes (Figure 2). Some took the set project as far as they could, and then played with some of the example programs from [Knuth 85] and modified them (figure 3).

From our point of view, the most important thing was that participants generally seemed to understand that type design for computer-based systems is not only a matter of spline drawing and digitization algorithms.

Acknowledgements

Thanks are due to the people who helped us to get METAFONT: Lynn Ruggles, David Fuchs and John Hobby at Stanford; and to the people who made it run on the IRISA machines: MM. Cedelle, Angelini, Portier and Le Bechec. Jorge Egli from EPFL-Lausanne deserves a special mention for his exceptional efforts, without which the course could not have worked as well as it did.

Special thanks are due as well to the participants in the course.

APPENDIX A: Practical Project Brief

1 Choose from the range presented in the terminal room a **nominal drawing size** d and a **boldness** b for the shapes you will make.

1.1 Write Metafont programs to make the following solid shapes:

- A square of side d
- A circle of diameter d
- A triangle of height d and base d
- Two half-squares of height d and width $d/2$

1.2 Assemble the shapes in the sequence *half-square, square, circle, triangle, half-square* with what you feel to be natural amounts of space between them. Adjust the sizes and alignments of the square, circle and triangle so that they appear to be the same size as the half-squares, and aligned with them at top and bottom, when seen from about 5 meters away.

1.3 Modify your Metafont programs to make the square, circle and triangle into hollow shapes, with stroke thicknesses d/b. Instead of the half-square, make rectangles of height d and width d/b.

Make an assembly like the one you made in 1.2. Adjust the sizes, spacing, alignments and stroke-weights of your shapes until, when seen from about 5 m away, they appear to be:

- Equal in size with each other and the rectangles
- Aligned top and bottom with each other and the rectangles
- Evenly spaced
- Equal with each other in "colour" (apparent boldness)

You will now have produced a set of elementary shapes that are technically satisfactory in terms of the criteria that apply to drawings of typeface characters.

2 Now choose a **nominal font size** f from the range presented in the terminal room.

2.1 Modify your Metafont programs to produce an assembly like the one you made in 1.3, but with rectangles of height f rather than d. Repeat the adjustments you made in 1.3, and make other changes to your programs as necessary, until your small shapes have the same evenness of apparrent size, alignment, spacing and colour that your large shapes had. You will now have produced a set of technically satisfactory **glyphs** (components of a font).

2.2 Experiment with changing the values of f and b, to produce differents fonts. Modify your Metafont programs so that they produce technically satisfactory glyphs for all the values of f and b you use.

3 Write down what you think are the reasons for the visual effects you saw at different stages in the project.

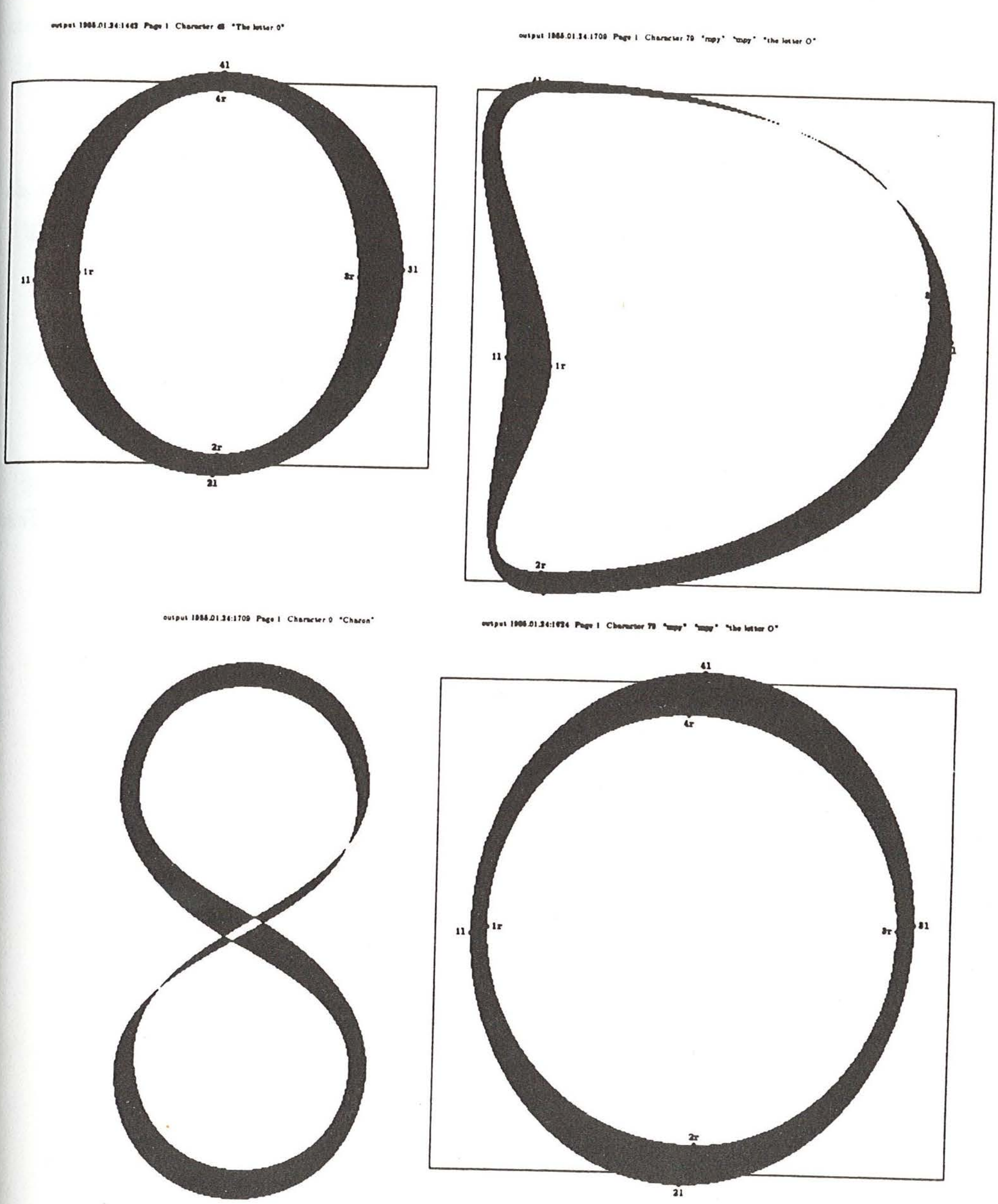
output 1988.01.24:1443 Page 1 Character 48 "The letter O"
41
4r
1l
1r
3r
31
2r
21
output 1988.01.24:1709 Page 1 Character 79 "mpy" "mpy" "the letter O"
1l
1r
2r
output 1988.01.24:1709 Page 1 Character 0 "Charon"
output 1988.01.24:1924 Page 1 Character 79 "mpy" "mpy" "the letter O"
41
4r
1l
1r
3r
31
2r
21

fig.3

References

[ANDRE 83]

Jacques André (ed.), *Actes des Journées sur la Manipulation de Documents (Rennes, 4–6 Mai 1983).*

Rocquencourt: INRIA, 1983.

[ANDRE SALLIO 85]

Jacques André et Patrick Sallio (eds.), *Typographie et Informatique. Support du cours INRIA, Rennes, 21–25 Janvier 1985.*

Rocquencourt: INRIA, 1985.

[BIGELOW 85]

Charles Bigelow, *Font Design for Personal Workstations.*

Byte, vol. 10 no. 1, pp. 255–270 (1985).

[CARTER 84]

K.A. Carter, *IMP — A System for Computer-Aided Typeface Design.*

in T. Miller (ed.), *Proceedings of the First International Conference on Text Processing Systems*, Dublin, 1984, pp. 114–119.

[FLOWERS 84]

Jim Flowers, *Digital Type Manufacture: An Interactive Approach.*

IEEE Computer, May 1984, pp. 40–48.

[KAROW 83]

Peter Karow, *IKARUS for Typefaces in Digital Form.*

Hamburg: URW, 1983.

[KNUTH 79]

Donald E. Knuth, *TEX and Metafont: New Directions in Typesetting.*

Bedford, Mass.: Digital Press, 1979.

[KNUTH 82]

Donald E. Knuth, *The Concept of a Meta-font.*

Visible Language, vol. 16 no. 1, pp. 3–27 (1982).

[KNUTH 83]

Donald E. Knuth, *Le concept de Metafonte.*

Communications et langages, vol. 55, pp. 40–53 (1983).

[KNUTH 84]

Donald E. Knuth, *A Course on* METAFONT *Programming.*

TUGBoat, vol. 5 no. 2, pp. 105–118 (1984).

[KNUTH 85]

Donald E. Knuth, The METAFONT book.
Reading, Mass.: Addison Wesley (in press), 1985.

[LEITCH 85]

S. Leitch, *Implementation of* METAFONT *on ICL PERQ.*
in "TEX for Scientific Documentation" (D. Lucarella ed.), these proceedings, 1985.

[RUGGLES 83]

Lynn Ruggles, *Letterform Design Systems.*
Stanford University Technical Report STAN-CS-83-971 (1983).

[SOUTHALL 85a]

Richard Southall, *Metafont and the problems of type design.*
in [ANDRE SALLIO 85].

[SOUTHALL 85b]

Richard Southall, *Metafont and the design of typefaces.*
Supplement to [SOUTHALL 85a], 1985.