Font News

Dominik Wujastyk

Concrete Roman and Italic

The new book *Concrete Mathematics* by Ronald L. Graham, Donald E. Knuth and Oren Patashnik* is naturally typeset using TEX, and also uses new typefaces. The maths is set in AMS Euler, a typeface designed by Hermann Zapf for the AMS. The text is set in special versions of Knuth's CM family roman and italic, with weights designed to blend with AMS Euler. This has been named Concrete Roman and Italic.

Zapf's design for AMS Euler is intended to suggest the look of mathematics as written on blackboards. This is how maths has chiefly been written by generations of maths teachers and researchers and is the medium in which most mathematics has always been seen by most mathematicians. The face is distinctly calligraphic, as opposed to italic, and in my view achieves the effect it seeks. But it faces the same difficulty as any striking and original new type design: it initially distracts the reader from the underlying text. It would be interesting to hear from anyone who reads *Concrete Mathematics* right through how the typefaces fare after protracted reading.

The Concrete roman face appears to have features in common with the CM typewriter font, although at the time of writing I have not seen the parameter files. It is a face somewhat in the genre of Bigelow's Lucida or Carter's Bitstream Charter, though different from these, of course.

For an example of the Concrete and Euler fonts, see Knuth's article "Typesetting Concrete", page 31 in this issue.

Lucida

In December 1988 Chuck Bigelow informed me that:

> Atari is soon (January 1989) bundling Lucida text fonts with its PostScript clone upgrade for its laser printer, the SLM 804. The Lucida fonts include the TEX text character set. The Lucida math fonts will also be available for Atari systems, but from the Imagen Corp., later in 1989. Also, QMS-Imagen are bundling Lucida fonts in the same character set with a software PostScript clone "UltraScript PC" for IBM PCs and various printers. The Lucida TEX math fonts will also be available from Imagen for that system.

* Reading, Mass.: Addison-Wesley, 1989.

Graphics

Computer Graphics and TEX — A Challenge

David F. Rogers
Aerospace Engineering Department
United States Naval Academy

Of late there has been considerable interest in the inclusion of graphics output within a TEX document. Programs such as PICTEX, gnuTEX, Fig, and TransFig seek to provide a mechanism for the inclusion of graphics within a TEX document. Each of these programs attempts to provide a nearly complete environment for the design and generation of line art or halftones for inclusion in TEX. All are worthwhile efforts. However, each suffers from a serious problem — device dependence. For example, the PICTEX macro package is too large to run on a microcomputer or in fact many workstations — it really requires a special large implementation of TEX; Fig and TransFig are graphics device dependent (Sun workstations); gnuTEX only generates LATEX compatible output; etc. Yet one of the strong attractions of TEX is its device independence. TEX itself runs on literally dozens of different machines from Crays, through the latest Silicon Graphics, Ardent, and Stellar superminicomputer engineering/scientific workstations, to a lowly PC XT running MS-DOS.

Systems such as PICTEX, Fig, Transfig and gnuTEX basically require the user to *recreate graphical output* or to generate it *ab initio*. This is rather inefficient. There are literally dozens of graphics programs that produce better graphical output, more efficiently than any of these systems.

An alternate technique for importing graphics into TEX is to use the `\special` command. Unfortunately, this requires giving up device independence. Further, not all dvi drivers support all `\special` commands.

A Suggested Minimal Set of TEX Graphics Macros

Graphics can be incorporated into TEX documents most efficiently by importing the output of graphics programs directly into the TEX document in the form of Plain TEX commands. The important question is how to do this easily and efficiently. Fortunately, only passive graphics is contained in a publishable document. Consequently, the functional requirements for graphics is quite limited (see Refs. 1 and 2). Specifically, these are the ability to move the