

T_EX: A Branch in Desktop Publishing Evolution, Part 1

Barbara Beeton

American Mathematical Society

Karl Berry

T_EX Users Group

David Walden

Donald Knuth began the development of T_EX in 1977 and had an initial version running in 1978, with the aim of typesetting mathematical documents with the highest quality, and with archival reproducibility far into the future. Its usage spread, and today the T_EX system remains in use by a vibrant user community.

However, the world of T_EX has always been somewhat out of sync with the commercial desktop publishing systems that began to come out in the mid-1980s and are now ubiquitous in the worlds of publishing and printing.

T_EX is a typesetting system that was developed by Donald Knuth, starting in 1977, to “do beautiful typesetting,” particularly of mathematics. At the time, PostScript and PDF did not yet exist, people were in the early stages of mathematically defining digital fonts, and WYSIWYG user interfaces were just being conceived. What we today think of as desktop publishing systems (for example, WYSIWYG, PostScript output) were half a dozen years away, and the computers of the time were slow and small compared with today. Thus, T_EX was implemented using markup of plain text files, as was typical with prior systems such as RUNOFF, to instruct T_EX about document formatting. T_EX also had its own way of outputting page descriptions for print as well as its own companion system, Metafont, for designing and formatting type.

T_EX met a latent need among a significant number of writers of technical papers and quickly became popular for such users, and in time the markup language for T_EX and its derivatives such as L^AT_EX became the lingua franca for an increasingly diverse set of users. Knuth designed T_EX for long-term stability (what works today should continue to work decades from now). He also made all the components of that stable base of the T_EX system, its algorithms, and the code freely available, including ways for people to develop augmentations on top of and within T_EX. A worldwide community of users, developers, and user groups evolved, largely disconnected from the more

conventional desktop publishing world driven by commercial concerns of publishers and desktop publishing system vendors (so very different than Knuth’s concerns for T_EX). This community remains vibrant today, 40 years later, and is an important branch in the development of desktop publishing.

This history of the T_EX typesetting system is being published in two parts. Part 1 is about the creation of T_EX and how it began to spread beyond Stanford University, where it originated. Part 2 is about the evolution of the worldwide community of T_EX users and developers and the impact T_EX has had on the broader world.

To save on the page count for *Annals* publication, a majority of this history’s references and all of its notes are posted on the web (tug.org/pubs/annals-18-19) with back pointers to pages in the published paper. Take a look now or anytime as you are reading the article.

T_EX—THE STANFORD YEARS

Because some readers may not be familiar with how a non-WYSIWYG user interface works, we touch on T_EX markup before getting into the detailed history of T_EX.

T_EX Markup

The general approach of T_EX is to use plain text files containing explicit markup to tell the T_EX processor how to typeset a document. The markup uses commands (called “control sequences”) and special characters; for example, `{\bf now}` causes text to be output in bold face from the `\bf` command to the closing brace (the close brace terminates the effect of the bold face command). T_EX has hundreds of such commands. Several of these commands allow macros to be defined, which themselves can then be called as new commands.

T_EX as users see it (known as “plain T_EX”) is a combination of (hundreds of) primitive T_EX commands and (hundreds of) macros defined using the primitives and other macros. The L^AT_EX version of T_EX (created by Leslie Lamport ca. 1983), which is vastly more popular than plain T_EX, is defined in terms of many more (thousands of) such definitions.

Figure 1 shows small examples of plain T_EX and L^AT_EX documents. When processed, both examples compile approximately into what is shown in Figure 2, appropriately placed on a page numbered 1 (the page number and margins are omitted from the figure). L^AT_EX markup is in terms of the type of document, in this case the L^AT_EX conventions for how an article is formatted (title, author, sections, and so on). The plain T_EX markup uses the system’s typesetting commands to place text and white spaces on a page. Regarding the line of math, some L^AT_EX users prefer L^AT_EX’s `\frac` construction to T_EX’s `\over` primitive; L^AT_EX converts the former to the latter for processing by T_EX.

Users of contemporary What-You-See-Is-What-You-Get word processing and typesetting systems such as Word and InDesign may think that T_EX’s use of markup rather than WYSIWYG is primitive. In a sense it is. T_EX was created at the very beginning of the era of graphical user interfaces; WYSIWYG word processing and its underpinnings were a subject of software and hardware research and development, and there were no widespread commercial WYSIWYG word processors or typesetting systems yet. More importantly, Knuth was trying to typeset documents with T_EX that were highly precise, were archival in terms of reproducibility far into the future, and used complicated symbology such as mathematics. Arguably, a markup approach is more useful for these conditions. One can do things with macro-based markup that are more difficult to do with WYSIWYG. For example, the L^AT_EX markup in Figure 1 is entirely “logical”; the example output in Figure 2 could be altered in nearly any conceivable way (different fonts, different spacing, dif-

```

\font\titlrm = cmr17
\font\subtitlrm = cmr12
\font\sectionbf = cmbx12 scaled 1200
\centerline{\titlrm A Small Example}
\vskip 1.5pc
\centerline{\subtitlrm A.U. Thor}
\vskip 1pc
\centerline{\subtitlrm December 1, 2017}
\vskip 2.25pc

\leftline{\sectionbf 1\quad Introduction}
\vskip .75pc

\noindent In this example, the document type
is declared, the title content is specified,
the document itself is started, the
titlecontent is inserted, there is some content
including some math, and the document ends.

\vskip 1.5pc
\leftline{\sectionbf 2\quad Example
including math}
\vskip .75pc

\noindent The quadratic formula is
$$
-b \pm \sqrt{b^2 - 4ac} \over 2a
$$
\bye

\documentclass{article}
\title{A Small Example}
\author{A.U. Thor}
\date{December 1, 2017}

\begin{document}
\maketitle

\section{Introduction}
In this example, the document type
is declared, the title content is
specified, the document itself is
started, the titlecontent is
inserted, there is some content
including some math, and the
document ends.

\section{Example including math}
The quadratic formula is
\begin{displaymath}
\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\end{displaymath}

\end{document}

```

Figure 1. Examples of plain T_EX markup (left) and L^AT_EX markup (right).

A Small Example

A.U. Thor

December 1, 2017

1 Introduction

In this example, the document type is declared, the title content is specified, the document itself is started, the title content is inserted, there is some content including some math, and the document ends.

2 Example including math

The quadratic formula is

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Figure 2. Output from Figure 1 markup.

ferent ordering of elements, and so on) without changing any of the input except for a different class name.

T_EX's macro capability and how T_EX is extended with macros are explained in more detail in the Appendix, "How T_EX is Extended," following Part 2 of this history (posted at tug.org/pubs/annals-18-19).

Beginnings of the T_EX Project

In 1976, Donald Knuth had revised the second volume of his magnum opus, *The Art of Computer Programming* (TAOCP), and had the galley proofs back from the publisher, Addison-Wesley. However, in the years since the original edition of the volume was published, Addison-Wesley had moved, for cost reasons, from using Monotype typesetting machines operated by expert typesetters of mathematics and other technical material to using computer-based phototypesetting methods. To Knuth's eye, the new galley proofs were so poor that he despaired about continuing his TAOCP book series.¹

Knuth has said, "The genesis of T_EX probably took place on February 1, 1977," when he first saw output from a high-resolution digital typesetting machine that could produce high-quality output. However, it seems he was already thinking about improving computer-based typesetting by the fall semester of 1976, as he gave an assignment in one of his Stanford classes (CS204) that included the problem of breaking paragraphs into lines. By mid-February he had decided to spend his upcoming 1977–1978 sabbatical year working on digital typography. On 5 May he began the design of T_EX and a 13 May memo sketched the system's design. He spent the next 45 days writing software to create digital letterforms and produced the 26 letters of the alphabet. In early July he returned to thinking about T_EX and produced a revised "reasonably complete specification" of the typesetting language, and he then left town for the better part of two months, leaving his students Frank Liang and Michael Plass to do a prototype implementation of T_EX. Plass remembers that they managed to implement enough of T_EX to process "a subset of T_EX input all the way to XGP [Xerox graphics printer] output. . . . Knuth . . . seemed pleased with our efforts, and proceeded to re-code T_EX himself." Knuth later noted that Plass and Liang had prototyped only a fraction of the full T_EX and that it was important that he code the complete program because he learned so much from doing the coding.

In 1978, Knuth also continued and finished typesetting the entire 700-page revised volume 2 of TAOCP in T_EX, thus substantially achieving his original goal: to typeset his opus digitally as beautifully as the prior generation of hot-metal math-specialty human typesetters could, in particular, finding a good way to specify high-quality typesetting of math.

However, that was not the end of Knuth's work with T_EX. Early on, T_EX began to spread beyond Stanford. Meanwhile at Stanford, Knuth had plenty of help; he was in an academic research environment and took advantage of it. What was to come was a 40-year story of contribution and collaboration.^{2,3,4}

The Project Continues

Over the next several years, Knuth greatly extended the T_EX project (with the help of students and other collaborators). He rewrote T_EX for greater portability from the SAIL language of the Stanford AI Laboratory to a subset of standard Pascal; created the Metafont program for designing type fonts and developed the extensive Computer Modern set of digital fonts (described more fully below); and then reimplemented (and improved) T_EX and Metafont, now using a "literate programming"⁵ system called WEB, which he invented for the job. With WEB, Knuth combined capabilities of writing documentation and writing code, with enhancements for both forms.

Student collaborators. Knuth’s funding allowed him to have graduate assistants with whom he could collaborate on T_EX, for instance, the Liang–Plass prototype of T_EX just mentioned. Liang also developed the original T_EX hyphenation algorithm in the summer of 1977 in collaboration with Knuth; Liang’s thesis research was completed in 1983 and included a major improvement to the original hyphenation algorithm, and Knuth adopted Liang’s algorithm for T_EX82. Knuth said that Liang’s experience with the original algorithm “led him [Liang] to discover a much better way, which can adapt to all languages”² that use hyphenation.

Knuth has stated, “One of the most challenging aspects of page layout is the problem of breaking paragraphs of text into individual lines. The fact that computers can do this task better than all but the most dedicated hand-compositors was one of my early motivations for developing the T_EX system.”² He understood this already when he left Liang and Plass to do their T_EX prototyping in the summer of 1977. Michael Plass collaborated with Knuth in studying and documenting T_EX’s line-breaking algorithm.

After finishing his PhD at Stanford in 1978 with Knuth as his advisor, Luis Trabb Pardo stayed on helping Knuth with T_EX in the years 1978–1981. He was the associate director of the T_EX project (Knuth called him “my right-hand man for T_EX78”), mostly working with people who were obtaining T_EX and looking into what could be done to print document proofs on various output devices. He was also a reader for the PhD theses of Michael Plass and Frank Liang (for which Knuth was advisor).

In 1978 David Fuchs, then a graduate student at Stanford, came to work on the T_EX project. He ported the SAIL version of T_EX from the AI lab’s WAITS operating system to DEC’s TOPS-20 operating system; conceived and developed the first version of device-independent (DVI) output for T_EX (more about this shortly); served as Trabb Pardo’s successor as “right-hand man” for T_EX82, the WEB implementation of T_EX; and reported to the T_EX community on T_EX activities at Stanford until 1986, as Knuth was winding down the T_EX project. Knuth dedicated Volume B, *T_EX: The Program, of Computers & Typesetting*⁶ “To David R. Fuchs: Who fixed everything.”

Ignacio Zabala was another of Knuth’s PhD students, whose thesis was also relevant to the T_EX project. In April 1980, Zabala (with Trabb Pardo) completed a conversion of Knuth’s implementation of T_EX in SAIL to Pascal, greatly increasing portability of the T_EX system. This conversion was done with the DOC system (also implemented by Zabala) that was a predecessor of Knuth’s WEB system of literate programming. Zabala’s Pascal implementation continued to track changes to the SAIL implementation until T_EX82 became available.

In other parts of this history, we mention other people at Stanford who helped Knuth develop T_EX.

Funding sources. At a May 2017 desktop publishing pioneers meeting at the Computer History Museum, Knuth discussed funding sources for his work on T_EX and Metafont at Stanford. He explained that he had never been very good at “understanding money,” but things at Stanford did have to be paid for. It was different from starting a business. He had NSF grants and Office of Naval Research contracts to study algorithms, for which he needed to say what he was spending money for under the grant or contract terms. To inform NSF that he was doing (a lot of) work on T_EX, he noted in a report that, “by the way, I have some software that was used to do the formatting to help me prepare these papers” (on his algorithms work). In this way he was able to have graduate assistants to help him with T_EX and Metafont. He also had a couple of tens of thousands of dollars in funding from IBM, which had no restrictions on use. In addition, a private donor gave \$200,000 to help get Charles Bigelow to Stanford.

Eventually the NSF said that, while the T_EX project was doing good work, it was not an appropri-

ate area for NSF to fund, and therefore introduced Knuth to Charles Smith of the System Development Foundation. As Knuth explained it, the Systems Development Corporation had money remaining after its work on the missile shield was done at the end of the 1950s. This money was turned over to the Foundation to be distributed for “good works.” The Foundation had already given a million dollars to Stanford’s music department. Smith asked Knuth what he needed to finish the project; Knuth gave him a number, but Smith said, “you have to ask for more.” (On video 64 of Knuth’s “Web of Stories,”³ Knuth says that System Development Foundation provided \$1 million in funding so he could finish his typography work and get back to TAOCP.)

Thus, all in all, Knuth had money to bring in distinguished visitors to Stanford and to have about a dozen graduate students. Many of these later contributed to desktop publishing beyond the T_EX project.

While Stanford liked to license technology developed there, Knuth was “adamant that T_EX stay in the public domain because [of the] big need in this under-served community”; he “refused to seek any intellectual property.” Knuth believed people were being held back by not having something like T_EX, and he “wanted to help people who were willing to put more into formatting to produce beautiful output.”

TEX SPREADS BEYOND STANFORD

While Knuth’s initial idea for T_EX was to be a tool for Phyllis Winkler, his secretary, and him to use to produce *The Art of Computer Programming*, his work was too polished and too capable even in its initial implementation to remain very long just for personal use. Plus, his inclination was to share the results of what he was doing.

In particular, several other things happened in 1978 that gave T_EX a much wider audience and longer life. (a) In January Knuth gave the American Mathematical Society (AMS) Gibbs lecture at which he demonstrated the necessity for moving beyond the then state-of-the-art of computer-based typesetting, and he explained what T_EX would do.⁷ (b) In August other people began to use T_EX at Stanford. (c) The AMS made a strong move to position itself to use T_EX in its publishing activities.

Making T_EX Portable

Knuth spent much of his 1977–1978 sabbatical year getting T_EX working on the Stanford AI Laboratory PDP-10 computer running the WAITS operating system, and able to drive SAIL’s XGP printer. By the summer of 1978 he had T_EX basically working, was trying to get a manual for T_EX finished,⁸ and was fixing bugs in T_EX but already trying to deflect requests for new features. This original implementation of T_EX had five components: TEXSEM.SAI to handle the semantics; TEXSYN.SAI to handle the syntax; TEXSYS.SAI to do the processing; TEXEXT.SAI for possible extensions; and TEXOUT.SAI to produce XGP output.

There were three technical dimensions for making T_EX portable: getting the code for T_EX to run on other computers, making T_EX able to output to other printers, and making T_EX operate precisely the same when running on different computers.

While Knuth was finishing the initial implementation of T_EX in the summer of 1978, Guy Steele (visiting from MIT) began porting T_EX to the MIT AI Lab PDP-10 computer, running the ITS operating system, and made T_EX at MIT work correctly with the AI Lab’s local printer. (MIT and Stanford were both connected to the ARPANET, permitting file transfer and login to remote computer systems.) Steele also gave Knuth many changes for his draft T_EX manual and convinced

Knuth to add a few new commands to \TeX . Steele made his changes in the `TEXEXT.SAI` module. \TeX was already running at MIT by August, although Steele did a little more work for the MIT printer in September–October when he was back at MIT. Usage of \TeX at MIT spread quickly as the port reached completion, with a user group being formed at MIT almost immediately.

Around 1980 Knuth asked David Fuchs to write a new printer driver (namely, `TEXVER.SAI`) as an alternative to `TEXOUT.SAI` (which drove the XGP) to handle a new Versatec printer they were getting. Fuchs noted that writing a new driver module for each new printer to work with \TeX was a bad idea and instead created a DVI page description language, with its module, `TEXDVI`, being an optional replacement for the `TEXOUT.SAI` module. From then on, drivers for different printers could be written using the DVI output without having to worry about properly linking into the \TeX executable itself.

As Knuth discovered that more and more people saw \TeX as being useful to them, he then set about making it as portable as possible at all levels.

Ignacio Zabala's reimplementations of \TeX in Pascal was called PTEX. As additional changes were made to the SAIL version of \TeX , they were also moved over to PTEX (this included adding a transcription of the `TEXDVI` module from Fuchs). There were two versions of PTEX. The first version merely copied the initial SAIL version of \TeX in using PDP-10 floating point arithmetic. However, different computers used different standards for floating point arithmetic. Thus, the second version of PTEX implemented machine-independent arithmetic within \TeX , so \TeX output would be exactly the same on different computers regardless of what form of floating point arithmetic they used.

At this point, Knuth began redoing \TeX and Metafont using his `WEB` development system, replacing the `DOC` system used by Zabala while retaining Pascal as the base implementation language. Knuth scrupulously avoided any aspect of Pascal that was either machine dependent (for example, floating point, as mentioned above) or too idiosyncratic (for example, pointers); this maximizing of source code portability greatly eased future work on \TeX . The resulting definitive \TeX_{82} version of \TeX included many other improvements based on user experience with \TeX_{78} and feedback.

Starting with PTEX and continuing with the `WEB` version of \TeX , many non-Stanford people were able to join the \TeX porting effort.

AMS Involvement

The AMS had heard rumors about what Knuth was doing with typesetting, and people from there paid a visit to Stanford where Knuth showed them what he was doing, along with some mockups of how \TeX could be used for journals as well as for books. Possibly influenced by that visit but also surely based on his computer science work with the analysis of algorithms (for example, the TAOCP volumes), Knuth was invited to give the prestigious AMS Gibbs lecture in January 1978. Rather than talk about “the glories of computer science” (as he was expected to do),² Knuth used the example of the *Transactions of the American Mathematical Society* to illustrate the poorer quality of typesetting since the AMS had ceased “to use traditional methods of typesetting,” that is, Monotype hot-metal type. He then described the possibilities for improvement using computer-based typesetting and sketched his work with \TeX .⁷

The response was enthusiastic, confirming Knuth's view that there was a need for a \TeX -like system in the math world. In particular, Richard Palais, the chairman of the AMS Board of Trustees, pushed AMS's adoption of \TeX , sending a delegation to Stanford for a month to learn \TeX and to bring it back to the AMS in Providence. Two of that group were Barbara Beeton, who was to

learn T_EX and install it at the AMS, and Michael Spivak. (Beeton believes that the AMS management overestimated how finished T_EX was as a production tool.) In the fall of 1979, David Fuchs visited AMS and set up T_EX on a DECSys_{tem}-20, and the organization started using T_EX for administrative publications (member list, sales catalog).

Michael Spivak, who was known to be a clear writer of mathematics, was commissioned by the AMS to write a tutorial guide to T_EX, which Spivak titled *The Joy of T_EX*. The AMS was also looking to commission someone to create additional macros on top of plain T_EX, to make T_EX even more suitable for writing papers containing advanced math. Spivak also received the commission for this software, which became AMST_EX.⁹ With AMST_EX developed and tested, AMS began accepting T_EX input for its *Proceedings* and *Transactions*, with the first published issue happening in 1984.

With Digital Press, also in 1979, the AMS co-published another book—*T_EX and Metafont: New Directions in Typesetting* by Knuth.¹⁰ Gordon Bell remembers that Digital Equipment Corporation was starting Digital Press around that time. Bell, a legendary computer designer and then a vice president of Digital, had met with Knuth about computer standards for floating point arithmetic, and thinks he heard of the T_EX effort then. Bell passed the news of T_EX to Heidi Mason who contacted Knuth about publishing with them. The 1979 book contains a copy of Knuth's Gibbs lecture and, slightly updated, his Stanford reports on the SAIL versions of T_EX⁸ and Metafont.¹¹ (Bell wrote the foreword to the book.)

In the early 1980s, the AMS commissioned renowned designer Hermann Zapf to create a new typeface called AMS Euler, to be done in collaboration with Donald Knuth using Metafont⁷⁹. The new type design had the goal of being more like the way that mathematicians handwrite math. The project was monitored by an AMS “font committee” reporting to the AMS trustees.

According to Gaudeul's 2003 research paper in *TUGboat*,¹² the AMS support of T_EX was part of a deliberate plan to “become involved in helping to develop a document preparation system, instead of waiting for a commercial system to be provided to them.” They wanted “a system that was compatible with most hardware, simple, flexible, and cheap; it was to run on mainstream computers.” It seems that the AMS was anticipating mathematicians preparing their own papers on the time-shared computers of the time.

The AMS endorsement of T_EX in these and other ways undoubtedly led to faster general acceptance of T_EX and enthusiasm for it. In particular, the AMS was significantly involved in establishing the T_EX Users Group (TUG) and supported the first issues of *TUGboat*, the main publication of the user group.

T_EX Users Group

The formation of TUG was an important step in T_EX's becoming widely popular and in T_EX development activity eventually becoming independent of Stanford. We touch on the early days of TUG here. Part 2 of this history will discuss the later evolution of TUG and expansion of the user community to include “local user groups” for various national and language communities.

In addition to assisting Knuth with T_EX developments (for example, T_EX-in-Pascal and DVI output), Knuth's collaborators at Stanford were handling distribution of T_EX to the people who had learned about it, answering questions as people moved P_TE_X to different computers, and so on. This could not go on forever; a non-Stanford distribution structure was needed.

We are not sure of the steps leading up to it, but the first meeting of the T_EX Users Group was held at Stanford on 22 February 1980. David Fuchs and Richard Palais suspect the core invitation list for this first meeting came out of Stanford. However, inside the back cover of *T_EX and Metafont*:

New Directions in Typesetting were two postage-paid business-reply postcards to be returned to the AMS, “if you have an interest in participating in a T_EX users’ group”; and various informal invitations were doubtless made. About 50 people attended the meeting. People were told how to get the files for T_EX with or without ARPANET access. Knuth reported on the status of T_EX (stability, interchangeability, and a common version were more important than different people trying to add “missing” features) and of Metafont (more than a dozen users); he also answered questions, for instance, T_EX should be applicable to non-math scientific documents, but he discouraged non-math use until the AMS gathered more experience. The coming Pascal version of T_EX was described. There were reports on some macro packages that users had developed and on experiences porting to particular machines. Finally, a steering committee for the group was elected. This was described in the first issue (October 1980) of *TUGboat*, the group’s publication, which also contains introductory material on T_EX and half a column on what TUG would be doing in the near future—being a clearinghouse for information on T_EX and helping get the Pascal version of T_EX into the field.

In its early years, TUG’s (and therefore *TUGboat*’s) efforts involved spreading the word about T_EX, getting T_EX running on many different computers and operating systems (including on the early “small” computers), getting T_EX output supported on a variety of printers (in those pre-PostScript days), and sharing examples of how T_EX was being used and supported in local environments. Also over these years, Knuth and others at Stanford decreased their involvement in T_EX distribution and support, while TUG members and others increased their involvement, with the AMS continuing to help. For instance, in addition to ARPANET file transfers or T_EX on magnetic tapes from Stanford, there was (for several years) a “Unix T_EX” tape organized by Richard Furuta and Pierre MacKay at the University of Washington, a DEC VMS tape (for a shorter time), and distributions on diskettes for Windows (for example, from Jon Radel, and the commercial product PC T_EX).

As TUG evolved, *TUGboat*’s contents also evolved.¹³ *TUGboat* has served as a newsletter about projects, events, and people. It has simultaneously provided tutorial material for all levels of T_EX practitioners, a forum for new ideas to be suggested and experiments to be described, and a place for major new developments in the T_EX world to be permanently documented.

TUGboat has never primarily been a journal of pure academic scholarship. Its articles are always reviewed by knowledgeable people (often the editors themselves) but do not generally undergo anonymous refereeing. Nonetheless, *TUGboat* has served the typical role of a scientific or engineering journal in allowing participants in the field to learn about and build on (or create alternatives to) the work of others. Furthermore, it has played a role beyond T_EX, regularly dealing with non-T_EX issues of typography, design, document preparation, and display. As of the end of 2017, *TUGboat* has published 120 issues in 38 yearly volumes, averaging about 370 pages per year.

Type Design for T_EX

T_EX is more widely known, used, and appreciated than its companion program, Metafont, and in the view of many, T_EX more satisfactorily solved the problem T_EX was meant to take on than is the case for Metafont. Metafont’s approach to type design was and remains unique among type design tools, for better or worse.

Knuth started working on digital letterforms as soon as he had a draft description of T_EX, before he had implemented T_EX. In so doing he began to learn the subtleties and intricacies of type design and began building a computer-based tool to aid in creating digital fonts. He has said it took him seven years to become satisfied with his work on type design (typography practitioners have said that this is a typical length of time for someone to become a competent type designer).

Digital Type

Digital type ultimately involves an array of pixels that represent any character—letters, numbers, and so on. One can thus create a character of digital type by explicitly defining an array of 0s and 1s saying which pixels are off and on; such creation of bitmaps (by humans) was common at the time \TeX and Metafont were developed.

These days, however, the typical approach is to specify the positions of key locations in character shapes (for instance, the top and bottom positions of a letter I, or a few key points along the curve of a letter C), and then have a mathematical function that smoothly specifies the outline of a character or paths among key locations. These mathematical functions then must be rasterized—turned into discrete pixels for a particular device—for the character to be displayed or printed.

Metafont takes this mathematical specification a step further: a character is defined not just by bare points and curves, but by a generalized computer program, written in the Metafont language. This programming approach allows for, among many other things, shared subroutines to draw common elements of characters, such as serifs.

An example program—the Metafont code for a letter A—is shown in Figure 3, along with a so-called “smoke proof” of it, showing the positions of the points defined, the outline of the drawn character, and the box in which the character is typeset (thus showing the space inserted to the left and right of the character itself).

```
beginlogochar("A",15); % width w = 15 units
% x positions of the points:
x1=.5w;
x2=x4=leftstemloc;
x3=x5=w-x2;
% y positions:
top y1=h+o;
y2=y3=barheight;
bot y4=bot y5=-o;
% draw the straight lines and curved top:
draw z4--z2--z3--z5;
super_half(2,1,3);
labels(1,2,3,4,5);
endchar;
```

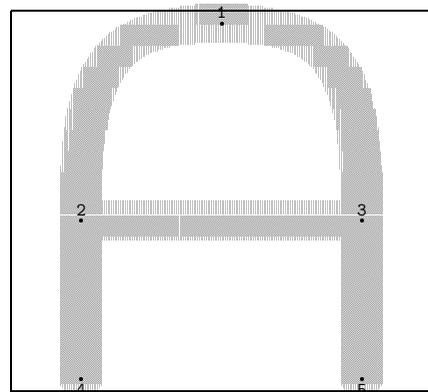


Figure 3. Metafont code to draw the letter A in the typeface used for the METAFONT logo. All code written by Knuth; the comments (starting with %) are editorial. The drawn character is shown graphically to the right, with points labeled. The `leftstemloc` and other variables not defined here are specified globally, ensuring consistency throughout the font.

The Metafont program interprets these character definitions and outputs a bitmap for each; the user must specify the assumed output resolution for this rasterization and, normally, other characteristics of the output device.

Knuth Learns about (Digital) Type Design

Knuth started his type design effort by researching the font used for the last well-printed volume of TAOCP, which was Monotype Modern 8A. This became the basis for his Computer Modern design. His wife, Jill, took 35mm photos of the printed letters, and they blew up the images by

projecting them down a long hallway where Knuth traced them onto a big piece of paper. Quickly reaching the limits of what could be done with the photographic enlargements (distortion from the projector was significant, for instance), he studied books on the history and practice of type design and carefully inspected many typefaces. He then “began to plan for a unified design in which all the shapes would change gradually as the overall specifications of an alphabet were varied.” He worked on that software using the data he had sketched from the wall, and began to produce machine-drawn letters, in June 1977.

Knuth developed T_EX’s basic text processing without math, and then began to develop fonts for T_EX to use. Beginning in the spring of 1977, he wrote a prototype set of subroutines and macros to develop the first iteration of what became his Computer Modern fonts; characters in this proto-Metafont were defined directly in SAIL. After several hundred characters had been designed, Knuth developed a language called Metafont with which to express character shapes more easily and precisely, and had a set for testing by March 1978.

After working on fonts for some months, Knuth began to read about how mathematics was typeset, and from the (conflicting) ideas in the books, he “was able to put together a picture of how a decent job of mathematical typesetting could be done.” After giving the Gibbs lecture in early 1978, he spent four months developing the math processing parts of T_EX and had T_EX running in the fall of 1978.⁸

Then, based on his initial experiences with type design, Knuth reimplemented the Metafont font generation system in 1979, coding it himself in SAIL.¹¹ Knuth completed the first prototype of Computer Modern in January 1980, being largely self taught.

John Hobby was Knuth’s closest collaborator in the development of Metafont84. Hobby used T_EX78 from the time he arrived at Stanford as a PhD candidate in 1980. He became involved with Metafont79 and worked with Gu Guoan (whom Knuth brought from China as a visitor to Stanford) to develop a prototype set of Chinese characters. Metafont’s polynomial pens were an important part of his 1985 thesis, which Knuth advised. Hobby contributed other algorithms to the development of Metafont, including Metafont’s representation of curves as Bézier cubic splines. Hobby recalls that he and Knuth worked together on Metafont in an unusual teacher–student collaboration, in which Hobby mostly designed algorithms and Knuth wrote all the production code. Knuth calls Hobby his “tireless co-conspirator throughout Metafont’s development,” and he dedicated Volume D, *Metafont: The Program, of Computers & Typesetting*⁶ “To John D. Hobby: Who improved everything.” After leaving Stanford, Hobby developed MetaPost, building on many aspects of Metafont while adding many capabilities, most crucially PostScript output. (Knuth is an enthusiastic user of MetaPost for his diagrams in TAOCP and elsewhere.)

Knuth continued to read about font design and the history of letterforms, and he talked to people in the industry, for instance at Linotype. As sketched in the next subsection, Knuth was introduced to or came in contact with experts in type design, often through the digital typography program at Stanford that Knuth initiated with the students aiming for a master’s degree and with various experts spending periods of time at Stanford involved with the program (see the introductory material of volumes C, D, and E of *Computers & Typesetting*). These designers critiqued the fonts in the Computer Modern typeface Knuth was developing, and he improved his designs based on their comments.

By 1981 Knuth had hundreds of specimen sheets that were critiqued by Matthew Carter in England and Charles Bigelow and Kris Holmes in the United States. Richard Southall came to Stanford during April 1982, and he and Knuth made many changes; Southall was especially influential with the sans serif variant of Computer Modern. After these years of continual study and refinement, in 1984 Knuth finished a substantially new Metafont language and system, incorporating ideas especially from Zapf, Bigelow, and Southall.

Perhaps the most significant change in Metafont84 was that it was based on outlines; for instance, the inside and outside edges of a letter O can be specified separately (and the region in between is blackened), whereas in Metafont79 an O would be specified by a single pen stroke moving around the shape (leaving a trail of blackened pixels as it goes). The Computer Modern fonts were improved again for the new Metafont, benefitting from critiques by Southall, Carter, and N.N. Billawala.

In the course of the T_EX project, Knuth obtained an Alphatype CRS typesetting machine, which had extremely high resolution. With the Alphatype, his improved fonts, and T_EX, Knuth was able to generate a good looking set of pages for the revision of Volume 2 of TAOCP, which he sent to Addison-Wesley, where the pages were photographed for offset printing. The book was published in 1981, five years after Knuth had received the proof copy of the book that he considered typographically unacceptable and four years after he started work on T_EX.

Metafont and the Stanford Digital Typography Program

Metafont is both a programming language for fonts (Figure 3) and a processor for that language (a program of some 23,000 lines). The Metafont processor compiles routines written in the Metafont language to produce characters for digital fonts.

As mentioned above, a number of noted calligraphers and font designers collaborated with Knuth at Stanford during the development of T_EX and Metafont and the Computer Modern fonts. Much of this effort took place as part of the Stanford digital typography program.

Charles Bigelow first met Knuth in 1980 when John Seybold (of the *Seybold Report on Publishing*) organized a small workshop at Stanford involving people from several organizations to explore and discuss T_EX and Metafont with the thought “to introduce them to a wider audience and to encourage support.” Seybold and his son Jonathan had previously done consulting work with SRI and “had been introduced to Donald through our SRI contacts. We were fascinated by what Donald was doing and thought it deserved wider attention.”

In 1982 Bigelow moved to Stanford from the Rhode Island School of Design, bringing some of his type design students with him. At Stanford he was an “associate professor (teaching)” of digital typography and was associated with both the studio art department and the computer science department. Bigelow led a Knuth-initiated master’s program in digital typography at Stanford. He was on Stanford’s faculty for 13 years and taught type design, typography, and the history and theory of writing systems.

The content of the following block quote came to us from Charles Bigelow as informal email text, which we have paraphrased and reordered.

Bigelow reports that they “tried to tailor the courses to what students needed and to what we thought digital typography should become. Because it was all new, there was no standard curriculum. We made it up as we went along.” The design students, who did not have a background in computing, took a basic computer science programming course. The students also took some courses in art and design. Mostly they were free to choose what interested them, because the program was “exploring and creating a new field.” The group did not know where new ideas would come from or what would be needed to develop them, apart from a basic grasp of both computer science and typography.

Bigelow taught courses, a notable example being “grammatology,” and he taught a course that covered the basics of type history, letterforms, typography, and type design. His partner, Kris Holmes, taught an occasional evening course on calligraphy.

Later, Bigelow merged several of the subjects into a one-quarter course called Concepts of Text [tug.org/pubs/annals-18-19] that he taught every year and which combined some theory of writing systems with some history of typography with some technology with some linguistics.

There was no typesetting course per se, but Stanford's computer science department had Alto computers with graphical user interfaces, and the students did some typography projects on those and liked the WYSIWYG interface. There were lessons on \TeX and on Metafont; the latter was a major focus because Knuth had gotten \TeX to a stage that he could say was nearly finished and was now actively developing Metafont, "so the students were in the Metafont orbit."

At Stanford, in the early 1980s, Knuth would meet with interested students and colleagues at lunch to discuss a wide range of questions and problems that came out of his research. He called it the "Metafont for lunch bunch." There was recurring discussion about whether there was an optimal class of mathematical curves that would render the best shapes while offering the most natural and intuitive manipulations by designers. Polygonal approximations and circular arcs were rejected, for reasons described below. Knuth preferred cubic polynomials, perhaps as a matter of mathematical aesthetics, and eventually settled on curves based on Bézier cubics, a case of Bernsteïn polynomials. (Vaughan Pratt, a Stanford computer science colleague and former student of Knuth, proposed another approach—generalized conic splines. Pratt described his approach to conic splines to the Metafont group and later published a series of papers on the subject, beginning with "Techniques for conic splines," SIGGRAPH, 1985.)

From 1982 onward, the "Metafont for lunch bunch" met once a week. They included Knuth's graduate students John Hobby, David Fuchs, and Scott Kim, occasionally Howard Trickey, and sometimes others; in 1982–1983, the design students usually attending were Daniel Mills and Carol Twombly, and then in 1983–1984, Cleo Huggins and David Siegel. Lynn Ruggles, a PhD student from UMass Amherst, came in 1982. Knuth was the final arbiter of what would be done with Metafont, but the discussions were wide ranging, from the best mathematical form for curves to the perceptual quality of different curves, to the user interface.

Issues with type design tools were also hot topics. The consensus was that outline fonts were the way to go, that polygonal and circular arc outlines were not good enough perceptually or mathematically, and that general conics proposed by Vaughan Pratt were good and were somewhat more comprehensible to designers than cubic curves; however, cubic curves of the Bézier–Bernsteïn parametric form were Don's eventual mathematical favorites. All the designers wanted a visual graphical interface, but Don liked the keyboarded programming-like interface.

During this time, sometimes as part of the digital typography program, Knuth acknowledged the help and influence of font designer Matthew Carter, though he was at Stanford for only a short time. Knuth has also noted Gerard Unger, who spent February 1985 at Stanford. Richard Southall also worked close by and visited Stanford for several periods of time and influenced Knuth.

In August 1983, Bigelow arranged the first academic conference on digital type design, "The Computer and the Hand in Type Design", held at Stanford. Also, a number of interesting reports came out of the digital typography program. In another important learning step, for several years members of the Stanford digital typography program collaborated with Knuth and Hermann Zapf on a commission to design the Euler typeface for the AMS (summary at tug.org/pubs/annals-18-19).

Ultimately, however, the digital typography program didn't continue. Bigelow notes that although the program had begun under auspices of both the computer science and studio art departments,

the studio art department was not completely on board and wouldn't award MFAs to the students or spend a faculty position on Bigelow. Knuth thus had to obtain outside funding and arranged for the digital typography students to get MS degrees through the computer science department. Bigelow notes, "The difference between an MFA and an MS might have made a difference if they had been interested in teaching careers, but they were all eagerly hired by high-tech firms, especially Adobe, and did very well, most retiring early."

Bigelow continues, "Without the art department connection, the program could not continue to attract visually talented design students, so the program would be [no] more than a minor sideline of computer science. Then Knuth went into early retirement around 1989–1990 to go back to working on TAOCP and fully retiring a few years later, and that ended the program." Bigelow stayed on at Stanford, teaching courses for a few more years, but was mostly consulting to industry by this time.

Bigelow suggests that for a brief time they had not only the best program in digital typography but also the most visionary. "There was not much written *about* the program, as we were too busy doing research, organizing, teaching, advising . . ."

Knuth remembers that the digital typography program produced "more than a dozen students who have a masters degree in typography, basically, and they've been extremely important in the industry since then." In particular, the Metafont class that Knuth, Bigelow, and Southall taught was important to Knuth. He also remembers, "that was a big point in the development of Metafont because I was implementing the features of Metafont one week before they were introduced by the class." The font language finally stabilized in 1984, the manual for Metafont was written, and then he had to do the final revision of the fonts, to make them look acceptable. "Let's say I finished that in 1985, and so finally, after eight years, I was able to bring my typography project to a conclusion. It was supposed to be a one year project for my sabbatical year."

Knuth Ties Up His Work on T_EX

Over the period 1977–1984, Knuth himself wrote two completely different implementations of each of T_EX (called T_EX78, written in SAIL, and T_EX82, written in WEB), Metafont (Metafont79 in SAIL and Metafont84 in WEB), and the Computer Modern fonts ("Almost" Modern and Computer Modern) using the experiences from his first implementation to make the second version as close to the ideal as he could. In each case, the second version is definitive. In 1986, Knuth's five-volume set of books, *Computers & Typesetting*, was published using and (self-)documenting the complete T_EX system,⁶ and he considered his work done. (The five volumes are: A, *The T_EXbook*; B, *T_EX: The Program*; C, *The Metafontbook*; D, *Metafont: The Program*; E, *Computer Modern Typefaces*. Volumes A and C are exceptionally detailed manuals. Volumes B, D, and E are unique examples of computer program documentation: B and D are complete program listings intended to be read as beautiful programs and literature, and E is the complete Metafont source code, with annotations, for the Computer Modern typeface.) In Knuth's view, it was critical to personally do the three major activities of coding, using, and documenting T_EX and Metafont, or the final system would have been markedly inferior.

After 1986 Knuth made only one more major change to T_EX: in 1989 he extended T_EX from using 7-bit characters to allow 8-bit characters as input; he also improved a few other things in passing. Then in 1990 he announced that his work on T_EX and Metafont was done, and that he would "make no further changes except to correct extremely serious bugs."

Knuth put his code in the public domain so that it, and the underlying ideas, could be used by anyone for any purpose. While only Knuth will make changes to canonical T_EX and Metafont and Computer Modern for the rest of his life (and they would remain unchanged after that), anyone

can make modified versions of \TeX or any part thereof, or use his code and algorithms in any way, as long as they called their work something other than \TeX , Metafont, or Computer Modern. He said, “Let us regard these as fixed points, which will give the same results 100 years from now as they do today.” This announcement is worth reading for yourself.¹⁴

True to his word, since 1990 Knuth has only worked on \TeX or Metafont every few years, at ever-increasing intervals, to review bug reports accumulated in the interim. For many years Barbara Beeton was the collector of bug reports (she called it being the “ \TeX entomologist”). She asked knowledgeable people to review each report to discern whether it is in fact a bug, a bug that has already been reported, or not a bug. Every few years, when Knuth was ready to look at possible bugs, Beeton sent her collection to him. Knuth is the sole arbiter of what is a bug and what will be left unchanged. In January 2017, Karl Berry took over the entomologist role.

Knuth’s release number for \TeX at the time he announced he was done was 3.1. He has released seven (slight) updates of \TeX since version 3.1, documenting each time what the changes were. His most recent two \TeX tuneups were in 2008 (version number 3.1415926) and 2014 (version number 3.14159265) (the version number of each update has one more digit of pi). He has stated that at the time of his death, his then-current version should be relabeled “ \TeX , Version π , and any remaining ‘bugs’ will be permanent ‘features.’”

ACKNOWLEDGMENTS, NOTES, AND ADDITIONAL REFERENCES

The acknowledgments for both Part 1 and Part 2 of this article are at the end of Part 2.

See tug.org/pubs/annals-18-19 for notes and additional references.

REFERENCES

1. B. Beeton and R. Palais, “Communication of Mathematics with \TeX ,” *Visible Language*, vol. 50, no. 2, August 2016, pp. 40–51, tug.org/pubs/vislang-16/article.pdf.
2. D. E. Knuth, *Digital Typography*, CSLI Publications, Stanford, CA, 1999.
3. Videos at webofstories.com/play/donald.knuth/50 through webofstories.com/play/donald.knuth/70.
4. N. Beebe, “25 Years of \TeX and Metafont: Looking Back and Looking Forward,” *TUGboat*, vol. 25, no. 1, 2004, pp. 7–30, tug.org/TUGboat/tb25-1/beebe-2003keynote.pdf.
5. D. E. Knuth, *Literate Programming*, CSLI Publications, Stanford, CA, 1992.
6. D. E. Knuth, *Computers & Typesetting*, vols. A through E, Addison-Wesley, 1986.
7. D. E. Knuth, “Mathematical Typography,” *Bulletin of the American Mathematical Society (new series)*, vol. 1, no. 2, March 1979, pp. 337–372, tinyurl.com/knuth-gibbs.
8. D. E. Knuth, *TAU EPSILON CHI: A System for Technical Text*, Stanford Computer Science Department, Report No. STAN-CS-78-675, September 1978.
9. “User’s Guide to AMST \TeX ,” ctan.org/pkg/amstex.
10. D. E. Knuth, *\TeX and Metafont: New Directions in Typesetting*, Digital Press, Bedford, MA, 1979.
11. D. E. Knuth, *Metafont: A System for Alphabet Design*, Stanford Computer Science Department, Report STAN-CS-79-762, 1979.
12. A. Gaudeul, “The (L \mathcal{A}) \TeX project: A case study of open source software,” *TUGboat*, vol. 24, no. 1, 2003, pp. 132–145, tug.org/TUGboat/tb24-1/gaudeul.pdf.

13. D. Walden and K. Berry, “*TUGboat*’s 100 issues—Basic statistics and random gleanings,” *TUGboat*, vol. 32, no. 1, 2011, pp. 17–22, tug.org/TUGboat/tb32-1/tb100walden.pdf.
14. D. E. Knuth, “The Future of \TeX and Metafont,” *TUGboat*, vol. 11, no. 1, 1990, p. 489, tug.org/TUGboat/tb11-4/tb30knut.pdf.

ABOUT THE AUTHORS

Barbara Beeton is employed by the AMS and has been involved with \TeX since 1978. She has been a TUG board member since TUG’s founding in 1980 and has been editor of *TUGboat* since 1983.

Karl Berry has worked with \TeX since 1982, is production editor of *TUGboat*, is a past president of TUG, and has been a key person in a variety of aspects of the \TeX and \TeX community infrastructure.

David Walden studies and writes about computing and digital typography history, including doing oral history interviews (75 from the \TeX world and the others from the more general computing world).

Additional references and notes for \TeX : A branch in desktop publishing evolution, Part 1

In the following, the number at the beginning of a note is a page number; words or a topic from that printed text page then appear indicating the position on the page to which the note or reference relates; then comes the note or reference itself.

- 1 **RUNOFF** J. H. Saltzer, TYPSET and RUNOFF, Memorandum editor and type-out commands, MIT Project MAC, MAC-M-193, November 6, 1964, web.mit.edu/Saltzer/www/publications/CC-244.html
- 2 **close brace** One use of braces in \TeX is to indicate scope in the programming language sense.
- 4 **TAOCP** Knuth's TAOCP website, cs.stanford.edu/~uno/taocp.html
- 4 **Monotype** Fred Williams, The Monotype Story, spring 1984, tinyurl.com/williams-monotype
- 4 **“genesis of \TeX ”** The Errors of \TeX , chapter 10 of Knuth's *Literate Programming* CSLI Publications, Stanford, CA, 1992, tug.org/texlive/devsrc/Master/texmf-dist/doc/generic/knuth
- 4 **“breaking paragraphs into lines”** TUG Interview Corner, interview of Michael Plass, 2009-12-20, tug.org/interviews/plass.html
- 4 **“May 13 memo sketched the system's design”** TEXDR.AFT, Chapter 24 of *Digital Typography* (see note above); or click on “[1,DEK]” at saildart.org/DEK
- 4 **“prototype implementation of \TeX ”** TUG Interview Corner, interview of Franklin Liang, 2009-12-20, tug.org/interviews/liang.html
- 4 **“Plass remembers”** Plass interview (see note above).
- 4 **“Knuth's work with \TeX ”** TEX originally meant “technical text”: Donald E. Knuth, *TAU EPSILON CHI: A System for Technical Text*, Stanford Computer Science Department Report No. STAN-CS-78-675, September 1978. Later TEX became Tau Epsilon Chi with the logo \TeX , as \TeX is an abbreviation of $\tau\epsilon\chi$, Greek for both “art” and “craft”.
- 4 **“forty-year story”** Steve Ditlea, Rewriting the Bible in 0's and 1's, *MIT Technology Review*, September 1, 1999, tinyurl.com/bible0s1s
- 4 **“invented for the job”** Chapter 7, How to Read a WEB, of Knuth's *Literate Programming*, CSLI Publications, Stanford, CA, 1992.
- 4 **“With WEB, Knuth combined capabilities”** For more on WEB, see, for example, a description by Knuth at literateprogramming.com/knuthweb.pdf; and a significant-sized program in WEB, also by Knuth, but not related to \TeX , at literateprogramming.com/adventure.pdf
- 4 **“Liang's thesis research was completed”** Franklin Mark Liang, Word Hy-phen-a-tion by Com-put-er, Stanford University PhD thesis, August 1983, tug.org/docs/liang/
- 5 **“ \TeX 's automatic line-breaking algorithm”** This algorithm is described in a classic paper by Plass and Knuth: Donald E. Knuth and Michael F. Plass, Breaking Paragraphs into Lines. The paper was originally published in 1981 in *Software—Practice and Experience*. It is reprinted in Knuth's *Digital Typography*, pp. 67–155. It also was part of Plass's 1981 PhD thesis: Michael F. Plass, Optimal Pagination Techniques for Automatic Typesetting Systems, Stanford University PhD thesis, June 1981, tug.org/docs/plass/plass-thesis.pdf
- 5 **“reported to the \TeX community”** tug.org/TUGboat/Contents/listauthor.html#Fuchs,David
- 5 **Zabala's Pascal implementation** tug.org/TUGboat/Contents/listauthor.html#Zabala,Ignacio
- 5 **“people at Stanford who helped Knuth”** To the above list of students who helped Knuth, one can also add faculty member

emeritus Arthur Samuel who also found ways to support Knuth in his TeX efforts, as Knuth recounted in 1990 (Arthur Lee Samuel, 1901–1990, *TUGboat*, vol. 11, no. 4, 1990, pp. 497–498, tug.org/TUGboat/tb11-4/tb30knut-samuel.pdf). For instance, see Samuel’s introduction to TeX: i.stanford.edu/pub/cstr/reports/cs/tr/83/985/CS-TR-83-985.pdf; for *TUGboat* writings by Samuel, see tug.org/TUGboat/Contents/listauthor.html#Samuel,Arthur

- 5 **“May 2017 desktop publishing pioneers meeting”** *****We need to include the URL of the transcript of the video when the transcript is available.***** Last DTP meeting video, minutes 26 to 38.
- 5 **System Development Foundation** oac.cdlib.org/findaid/ark:/13030/tf429003m4/
- 6 **“deflect requires for new features”** Email exchange with Guy Steele, June 2017.
- 6 **.SAI files** Click on “[TEX,DEK]” at saildart.org/DEK
- 6 **“Guy Steele... began porting TeX”** Email exchange with Guy Steele, June 2017.
- 6 **Steele port to MIT** In addition to our 2017 email exchange (see note above), Guy Steele provided us with copies of his SAIL emails from July 28 to September 8, 1978, and his MIT emails from September 2 to October 28.
- 6 **Ports to PDP-10s** Other ports of TeX to PDP-10s are noted in Nelson H. F. Beebe, The design of TeX and Metafont: A retrospective, presented at the Practical TeX conference of 2005, tug.org/tugboat/tb26-1/beebe.pdf
- 7 **“TEXDVI, being an optional replacement for”** TUG Interview Corner, Interview of David Fuchs, tug.org/interviews/fuchs.html
- 7 **“could be written using DVI output”** David Fuchs, The Format of TeX’s DVI Files, *TUGboat*, vol. 1, no. 1, pp. 17–19, tug.org/TUGboat/tb01-1/tb01fuchs.pdf
- 7 **“regardless of form of floating point arithmetic”** Nelson H. F. Beebe, Extending TeX and Metafont with floating-point arithmetic, *TUGboat*, vol. 28 (2007), no. 3, pp. 319–328, tug.org/TUGboat/tb28-3/tb90beebe.pdf
- 7 **“based on user experience with TeX”** Metafont went straight from SAIL to WEB.
- 7 **“Rather than talk about”** Richard Palais email, 2017-11-30.
- 7 **“sketched his work with TeX”** webofstories.com/play/donald.knuth/61
- 8 **Joy of TeX** Michael Spivak, *The Joy of TeX*, American Mathematical Society, Providence, RI, 1986.
- 8 **Michael Spivak** In addition to his AMSTeX work, Spivak early on also designed the MathTime professional fonts, based on and for use with the Times font or to replace the Computer Modern math fonts; these were made available via the PCTeX company.
- 8 **“Gordon Bell remembers”** Email of 2017-08-03.
- 8 **“AMS commissioned Hermann Zapf”** Zapf mentions this in his own life story: linotype.com/1494/the-lifestory-of-hermann-zapf.html
- 8 **“goal of being more like how mathematicians handwriting”** *Digital Typography* (see note above), chapter 17—a reprint of a paper co-authored by Knuth and Zapf; see also tug.org/TUGboat/tb22-1-2/tb70zapf.pdf
- 8 **“first meeting of the TeX Users Group”** Palais interview, tug.org/interviews/palais.html; Beeton interview, tug.org/interviews/beeton.html; Fuchs interview, tug.org/interviews/fuchs.html
- 8 **“first issue (October 1980) of TUGboat”** tug.org/TUGboat/Contents/contents1-1.html
- 9 **“getting TeX running on many different computers”** See the categories Output Devices, Site Reports, and “small” TeX at tug.org/TUGboat/Contents/listkeyword.html

- 9 **“developments in the TeX world would be permanently documented”** *TUGboat* was originally subtitled *The TeX Users Group Newsletter*; as of 1988 its subtitle became *The Communications of the TeX Users Group*.
- 9 **“TUGboat has served the typical role”** In parallel with *TUGboat*, TUG published 13 issues of *TeX and TUG News* from 1991–1995 and 20 issues of *The PracTeX Journal* between 2005 and 2012. The news function of the former was merged into *TUGboat*; the latter’s goal was to publish only practical articles, where *TUGboat* has a spectrum of articles.
- 9 **“typography practitioners have said”** For example, Charles Bigelow said this: Note on Typeface Protection, *TUGboat*, vol. 7, no. 3, 1986, pp. 146–151, tug.org/TUGboat/tb07-3/tb16bigelow.pdf
- 10 **“began to produce machine-drawn letters”** Pages 64–72 of Donald E. Knuth, *Companion to the Papers of Donald Knuth*, Center for the Study of Language and Information, Stanford, CA, 2012.
- 11 **METAFONT implementation collaborators** Tomas Rokicki was another implementation contributor to the Metafont project. While at Texas A&M, Tomas Rokicki created the initial version of what became the Web2C system that is used today to compile TeX. Rokicki started working for the TeX project in the summer of 1985, before his first term as a PhD student at Stanford. There he designed and implemented the PK font format and tools, a more compact form for the bitmaps output by Metafont.
- 11 **“part of his 1985 thesis”** John Douglas Hobby, Digitized Brush Trajectories, Stanford dissertation, August, 1985, ect.bell-labs.com/who/hobby/thesis.pdf
- 11 **“Hobby mostly designed the algorithms and Knuth wrote all the code”** TUG Interview Corner, Interview of John Hobby, tug.org/interviews/hobby.html
- 11 **“fonts were improved again”** As described in this and the next subsection, Knuth redid his font design software and improved his set of fonts several times: the first version of the fonts was called Almost Modern; a second, never released, version was called Better Modern.
- 12 **“a program of some 23,000 lines”** The book *Metafont: The Program* shows the Metafont implementation as an approximately 560-page WEB code listing (intermingled Pascal code and documentation). This results from the `mf.web` source file written by Knuth, which is slightly more than 23,000 lines; `mf.web` gets compiled (by the `tangle` program) into super-dense Pascal code (only suitable for computer processing). Translating that Pascal code into C (as is done in the major TeX distributions nowadays, C compilers being much more widely available than Pascal compilers), results in, coincidentally enough, about 23,000 lines of code, more or less as a human programmer might write them (except without any comments). So 23,000 lines is a plausible measure of the size of Metafont either way. Although there are many more refined metrics to measure program complexity, we chose to mention this one for simplicity. (By the way, TeX is very nearly the same size.)
- 12 **“wider audience and to encourage support”** Jonathan Seybold email of 2017-09-03.
- 12 **“thought it deserved wider attention”** Barbara Beeton was also at Stanford in March 1980 (the visit timed so she could also attend the Seybold-organized seminar) “to learn how to program Metafont and to create a prototype Cyrillic font for use in *Math Reviews*.” The Seybold-organized meeting was also where she first met Bigelow.
- 12 **“thought it deserved wider attention”** Bigelow notes (email of 2017-08-31), “The commercial typesetting systems guys [at the seminar] all said that TeX was too complicated and slow to be commercially acceptable. Of course, they are all gone now.”
- 12 **“from Charles Bigelow as informal email text”** Emails of June 6 and August 31, 2017. A parallel version of the story is in his interview in *TUGboat*, tug.org/TUGboat/tb34-2/tb107bigelow-wang.pdf
- 12 **“a notable example being grammarology”** Which was based on an older notion of the study of letters, from the book *A Study of Writing: An introduction to the study of grammarology* by I.J. Gelb.
- 12 **“‘Concepts of Text’ course”** The syllabus for that course is available in the Stanford archives.
- 13 **“METAFONT for lunch bunch”** Before Bigelow got to Stanford, Knuth had a TeX-for-lunch bunch.
- 13 **Bernshtein polynomials** en.wikipedia.org/wiki/Bernstein_polynomial
- 13 **Lynn Ruggles** In 1983, Lynn Ruggles, a graduate student at Stanford, compiled a catalog of different approaches to digital type tools: “Letterform Design Systems”, Stanford Technical Report No. STAN-CS-83-971, i.stanford.edu/

pub/cstr/reports/cs/tr/83/971/CS-TR-83-971.pdf

- 13 **Southall also influenced Knuth** tug.org/TUGboat/tb36-2/tb113southall.pdf, i.stanford.edu/pub/cstr/reports/cs/tr/85/1074/CS-TR-85-1074.pdf
- 13 **“first academic conference”** visiblelanguagejournal.com/issue/73
- 13 **“first academic conference”** At the time, Bigelow was leading the committee on letterform education and research of ATyPI (Association Typographique Internationale).
- 13 **“number of interesting reports came out”** Pijush K. Ghosh on Indian scripts, i.stanford.edu/pub/cstr/reports/cs/tr/83/965/CS-TR-83-965.pdf; John Hobby and Gu Guoan on a Chinese Meta-Font, i.stanford.edu/pub/cstr/reports/cs/tr/83/974/CS-TR-83-974.pdf; A formal approach to letter form design by Ghosh and Bigelow i.stanford.edu/pub/cstr/reports/cs/tr/83/966/CS-TR-83-966.pdf; Letterform Design Systems by Lynn Ruggles (cited above); *The Euler Project at Stanford*, a monograph by David R. Siegel, 1985.
- 13 **“commission to design the Euler typeface”** Stanford library guide to the Euler project archive, pdf.oac.cdlib.org/pdf/stanford/uarc/sc0362.pdf
- 14 **“Knuth remembers that the digital typography program”** A summary of the TeX project, video #70.³
- 14 **“the METAFONT class which Knuth”** Donald E. Knuth, A Course in METAFONT Programming, *TUGboat*, vol. 5 (1984), no. 2, pp. 105–118, tug.org/TUGboat/tb05-2/tb10knut.pdf
- 14 **“he considered his work done”** Donald Knuth, Remarks to Celebrate the Publication of *Computers & Typesetting*, *TUGboat*, vol. 7 (1986), no. 2, pp. 95–98, tug.org/TUGboat/tb07-2/tb15knut.pdf
- 14 **“allow arbitrary 8-bit characters”** Donald E. Knuth, The New Versions of TeX and METAFONT, *TUGboat*, vol. 10 (1990), no. 3, pp. 325–328, tug.org/TUGboat/tb10-3/tb25knut.pdf
- 15 **“review bug reports accumulated in the interim”** See “Errata” at www-cs-faculty.stanford.edu/~knuth/abcde.html.
- 15 **“most recent two TeX tuneups”** Knuth’s so-called “tune-up” reports for his two most recent reviews are readily available: Donald Knuth, The TeX tuneup of 2008, *TUGboat*, vol. 29 (2008), no. 1, pp. 233–238, tug.org/TUGboat/tb29-2/tb92knut.pdf; Donald Knuth, The TeX tuneup of 2014, *TUGboat*, vol. 35 (2014), no. 1, pp. 5–8, tug.org/TUGboat/tb35-1/tb109knut.pdf. They too are worth reading as an example of the care and careful explication Knuth puts into merely fixing a rare bug.