

Communication of Mathematics with T_EX

Barbara Beeton
Richard Palais

user input:

`\sqrt{b^2}`

T_EX output:

$\sqrt{b^2}$

Abstract

Mathematics publication has changed radically over the past 50 years, for both authors and publishers. What once required a skilled compositor to produce can now be accomplished, with the aid of computers and software, directly by authors. One key component of this change is the T_EXtypesetting program. This software, designed by a mathematically discriminating computer scientist and made freely available, is now in operation on nearly every computer system in common use.

Keywords

open source, composition of mathematics, symbols (math and technical notation), fonts for math and science, mathematical typesetting software, composition software, mathematical symbols in Unicode, TeX, TeXbook, Knuth, amstex, STIX, AMS-TeX, AMS-LaTeX, LaTeX, TUG (TeX Users Group)

few suppliers would take on such work, and mathematical composition was always considered “penalty copy”²

For the first half of the twentieth century, a mathematical work for publication began as a manuscript, either handwritten or partially type-written (the text) with mathematical symbols inserted with pen and ink. A typescript was typically prepared by a secretary: senior faculty had their own personal assistant, junior members relied on departmental staff. Often the secretary primarily responsible for manuscript preparation had a typewriter with special capabilities, greatly reducing the need for manual insertions.

Various mechanical advancements improved the visual quality of manuscripts, and documents intended for limited audiences or quick distribution, such as lecture notes or proceedings of meetings, were often published from such copy. The Varityper and IBM Selectric Composer, two enhanced typewriters with interchangeable type heads (and type styles emulating traditional printing typefaces), in the hands of a skilled typist, were capable of producing quite readable output, with character sets for typical mathematical notation and variant type sizes needed for accurate representation of sub- and superscripts. What they generally lacked was an easy mechanism for justifying lines, an easily recognizable characteristic of typeset copy; justification was possible, but it always required a second pass, which was usually not fully automatic. Nonetheless, as prices increased for hot-metal composition, even some traditional journals began to use this method of preparing copy for the printer.

Investigation into photocomposition began in the late 1940s, with production-capable machines in use in the 1950s. The earliest machines flashed a light through a negative image of a character to produce an image on photographic media. By the mid-1960s tools were in place to convert marked-up copy from codes punched on paper tape into images, at least for ordinary text. But mathematics was still too complicated and mostly beyond the capabilities of this technology. A few machines, manually operated, did have the capability of varying font size and baseline, similar to what was possible with Monotype composition, but their use was not widespread.

More capable imaging devices based on CRT technology provided the necessary flexibility. By the mid-1970s, several commercial systems were available that could produce acceptable mathematics output, but there was nothing remotely available to or usable by an individual mathematician. All required skilled input operators, as the quality of the output was in some cases dependent on input consistency.³

2 Since mathematical composition was so exacting and time consuming, most compositors preferred to take on easier work that was more lucrative; even though mathematical work was charged at a higher price per page, the compositor suffered a penalty for accepting it.

3 According to one anecdotal report, the appearance of the same notation differed in two chapters input by different individuals; the system used for that project was one in which the positioning of symbols in displays was manually adjusted by the person doing the input.

The situation was ripe for improvement when the galleys of the re-set volume 2 of *TAOCP* reached Knuth.

Analysis of the problem

What Knuth did next is described nicely in his lecture on the occasion of his receiving the Kyoto Prize in 1996 [KnK]. Publication of the photoset volume 2 was halted, and Knuth sought out the best examples he could find of the mathematical typesetter’s art. He chose three: Addison-Wesley books, in particular the original *TAOCP*; the Swedish journal *Acta Mathematica*, from about 1910; and the Dutch journal *Indagationes Mathematicae*, from about 1950.

To develop rules for proper spacing in mathematics, he writes

I looked at all of the mathematics formulas closely. I measured them, using the TV cameras at Stanford, to find out how far they dropped the subscripts and raised the superscripts, what styles of type they used, how they balanced fractions, and everything. I made detailed measurements, and I asked myself, “What is the smallest number of rules that I need to do what they were doing?” I learned that I could boil it down into a recursive construction that uses only seven types of objects in the formulas.

[KnQ, pp. 364–365]

Growing pains

The initial implementation of $\text{T}_{\text{E}}\text{X}$ began in October 1977 and was complete in May 1978. This tool was at first intended just for use by Knuth and his secretary to produce future volumes of *TAOCP* of which he could be proud. As a trained mathematician, he designed the input so that it would be meaningful in its raw form to another mathematician, but would also be easy for a secretary to type. Symbols would be input by name, e.g., $\backslash\gamma$, as would the structural components of a document, e.g., $\backslash\text{chapter}$ or $\backslash\text{section}$, as opposed to the prevailing compositor’s approach of marking changes by font and type size. (The latter approach is still evident in the design of many word processing programs, although it’s usually hidden from the person entering the text.) $\text{T}_{\text{E}}\text{X}$ was designed to be used as a batch process, although interactive entry is possible, so the output isn’t seen until the file has been processed; it is decidedly not “WYSIWYG”. It was not contemplated that $\text{T}_{\text{E}}\text{X}$ would become a commercial product; instead, it would be made freely available.⁴

4 $\text{T}_{\text{E}}\text{X}$ is recognized as one of the first major pieces of “open source” software. Only one restriction has been requested: that only the author be allowed to make changes to the original, and that if changes are made, the name $\text{T}_{\text{E}}\text{X}$ not be used, but the derivative renamed. The rationale for renaming is to avoid confusion, so that if, in 50 years, someone processes an old file with $\text{T}_{\text{E}}\text{X}$ the results will be the same as they were when that file was new.

In January 1978, Knuth delivered the Josiah Willard Gibbs lecture to the annual meeting of the American Mathematical Society (AMS). The lecture, entitled "Mathematical Typography" [KnM], began "Mathematical books and journals do not look as beautiful as they used to." Armed with copious examples, both good and bad, and a firm sense of how best to present mathematical notation so that it is intelligible (at least to those who are familiar with its use), Knuth presented a view of how computers can serve to replace the vanishing expertise of traditional compositors and restore the appearance of technical publications to their former glory. In addition to the discussion of proper presentation of mathematical notation, the lecture introduced a companion tool, Metafont, for production of the needed fonts.

The chair of the AMS Board of Trustees, Richard Palais, was in the audience. Since the AMS was one of the publishers suffering from the technological transition, T_EX sounded like the solution to many problems. An arrangement was set up for a group of AMS representatives to spend a month at Stanford and learn T_EX, "bring it back and make it work". This group consisted of one staff member from each of the AMS offices (Barbara Beeton from headquarters and Rilla Thedford from Mathematical Reviews) and three mathematicians: the aforementioned Richard Palais; Robert Morris from the University of Massachusetts, Boston, who had extensive computer experience; and Michael Spivak, who had a proven ability to write cogent textbooks. The charge was to develop methods for dealing with the typical publication cycle and to write an interface and instruction manual for end users as well as production staff.

As one of the AMS representatives, Beeton gathered a number of "good bad examples" that she knew would be encountered in production because they already had. This turned out to be good preparation: several of these examples turned up later in *The T_EXbook* [KnTB] and as new features added to the program itself.⁵

The T_EX program was duly brought back to the Providence office of the AMS, installed, and initial implementation of useful procedures was undertaken.⁶ The first applications were light on mathematical content; polishing of the extended instruction set for use by mathematicians (AMS-T_EX) and writing of its user manual [SpJ] were still underway. Also, in the interim, extensive changes were made in the program to provide features not in the first iteration (known now as T_EX78). These changes included

5 Since Knuth's primary goal was to complete *TAOCP*, he assigned the trademark "T_EX" to the AMS, to keep himself free of legal concerns.

6 In fact, things were rather more complicated. First, a new computer was needed; a DEC-System 20 was chosen to match the hardware Knuth was using at Stanford. Communicating updates, a rather frequent occurrence since T_EX was still under active development, was accomplished via ARPANet file transfer to MIT, where Palais put it on a tape that he drove to Providence.

(1) enhanced manipulation of "boxes" (the containers for printed characters) and surrounding spaces and (2) an increase in the number of fonts that could be used as well as improved methods for manipulating them. The resulting version, known as T_EX82, is the basis for today's program. At the same time, the language in which T_EX was written was changed, from one that was in limited use to one with a solid history of use in teaching programming.⁷ As it had been from day one, the software remained free to use and adapt. Having achieved his goal of a system that met his needs, Knuth returned to his work on *TAOCP*.

Contributing to T_EX's growing popularity was the emergence, starting in the mid-1980s, of personal computer systems and their rapid adoption by technically minded individuals. This was T_EX's natural audience, and implementations of T_EX on these personal machines proliferated.

By the end of the 1980s, a growing user population in Europe was becoming increasingly frustrated with the difficulties in handling non-English texts. T_EX required arcane combinations of characters to represent accented letters rather than the single pre-accented forms provided by European keyboards. Also, the compound input forms could not be properly hyphenated. A persuasive group of German users sat down with Knuth at the 1989 T_EX Users Group meeting to discuss this lack. This meeting resulted in the extension of T_EX to accommodate natively accented letters on input and proper hyphenation in processing.⁸

Communicating mathematics

The basic T_EX system comes with a functional toolkit of typographic functions and one (quite extensive) family of fonts. This is necessary for the typesetting of mathematics and other technical material, but many users did not find it sufficient. Development has occurred in several areas, not all involving T_EX.

Document structuring

While AMS-T_EX formatted complicated math displays admirably using descriptive commands, it lacked the ability to automatically number equations and sections of a document and the means for cross-referencing. Another

7 In the process of upgrading from T_EX78 to T_EX82, Knuth refined the technique that he has called "literate programming". Using this approach to programming, code is interspersed with explanatory text, with the results (more) intelligible to a reader. (Both the T_EX and Metafont programs have been published in this form as part of the series *Computers & Typesetting* [KnCT].) Knuth has said that he considers literate programming to be a more important contribution to software than T_EX.

8 This became version 3. Effective with this version, the version number has been incremented by one decimal digit with every upgrade, converging to the numeric value of π ; Knuth has requested that, at his death, T_EX should not be updated further, and the version frozen as " π ".

user instruction set, L^AT_EX (devised by Leslie Lamport,⁹ a former student of Palais), did provide those features, although it lacked the mathematical refinements of AMS-T_EX. The AMS, responding to pressure from authors, arranged to have the math-formatting facilities of AMS-T_EX rewritten to operate within the L^AT_EX paradigm; the result was called AMS-L^AT_EX, comprising two parts, amsmath and the AMS document classes.¹⁰

Fonts

Font development has been driven by the availability of personal computers and laser printers and the growth of the World Wide Web, as well as by the desire for variation in type styles available for T_EX.

One font family that originated in the need for robust output from low-resolution laser printers is Lucida by Kris Holmes and Charles Bigelow. Bigelow was on the Stanford faculty during part of the T_EX project development, and Lucida has, from the very beginning, included a large complement of math symbols as needed by T_EX users.

Desire to give mathematicians the ability to communicate on the Web was the driving force behind the STIX project.¹¹ In the first phase of this project, a comprehensive list of math symbols was compiled from lists submitted by the STIpub member organizations and submitted for addition to Unicode. The bulk of additions became available with Unicode 4.0 in 2003, comprising several thousand symbols, including several variant alphabets (e.g., Fraktur and script) needed to discriminate between different variables as defined in mathematical contexts.

Version 1 of the STIX fonts (based on Times) was released in 2012, and final polishing of version 2 is underway.

Possibly influenced by the STIX work with Unicode,¹² Microsoft added mathematics support to Word 2007¹³ along with the newly designed

9 Lamport went on to win computer science's prestigious Turing Award in 2014, for reasons not related to L^AT_EX. (Donald Knuth had received the award in 1974.)

10 A document class is a set of macro commands that define the structuring of a document (e.g., a book or article). A class is written in such a way that page size and layout, elements such as chapter and section headings, and the style of bibliographies are easily adapted to conform to the specs for a particular publication. Then all that remains for an author is to invoke the class (`\documentclass{pubname}`) to produce the document in the desired style.

11 Scientific and Technical Information eXchange (<http://stixfonts.org>) is a project sponsored by STIpub, a consortium of five professional societies/technical publishers and a major commercial publisher of technical books and journals. This work is still going on, as new symbols are devised by scientists and symbols previously overlooked are uncovered.

12 One of the Unicode Technical Committee members who helped to shepherd the STIX request through to acceptance was a Microsoft software design engineer, Murray Sargent, who was also a key participant in the implementation of mathematics support in Microsoft products.

13 The design of mathematics support owes a great deal to T_EX. Microsoft engineers met with Knuth in 2003 to study his methods [Sa].

Cambria font [MH]. Cambria is the first OpenType font (OTF) to make use of the OTF Math table. Indeed, the OTF Math table was created specifically for Cambria, and many of its parameters are recognizable as parallel to the T_EX font paradigm.

The Web

XML was developed as a Web-aware application of SGML. Even for SGML, there had been an effort to standardize the names of math symbols as a “public entity set”, and this drew heavily on the names assigned for T_EX and AMS-T_EX. This vocabulary was taken into XML and its technical daughter MathML. Work has continued in this area to maintain parallel naming, insofar as possible, between the two “languages”.

Since MathML is not as easily comprehended by humans as T_EX translation conventions and software have sprung up to allow input using T_EX notation, which is familiar to mathematicians. Another Web presentation tool, MathJax, has emerged to allow in-line math to be delivered natively on-screen (without the use of bitmap inclusions, which are not scalable, or PDF); again, the input notation is essentially T_EX although it is rarely entered directly by a human author.

Non-technical applications

Since T_EX was designed as a hardware-independent batch process, it is capable of being used in repetitive contexts to prepare personalized form letters, invoices, bank statements, train schedules, catalogs, ...; the list goes on and on. The original output format is compact since it contains only the identification of glyphs and their location on the page; thus it can be archived compactly (along with one copy of each needed font and other repetitive content such as logos), an important feature to comply with legal requirements for some documents. Most such uses are “invisible” to those not familiar with the relevant workflow, but they are extensive, especially in Europe.

Remaining limitations

One area that has not yet seen a satisfactory method of presentation is accessibility—the ability to translate T_EX input to an audio output that is readily understandable by a trained mathematician with visual limitations. Part of the problem is that, for best results, an author must think ahead about such use and restrict the way that notation is used; most authors can't be bothered, even if they *are* aware of the problem. Someone may find a credible and easily applied solution, but to date, it's still a quite hard problem.

Conclusion

The most lasting effect of T_EX is separate from the software itself: T_EX's vocabulary has become the *lingua franca* of mathematics. Knuth's design of a linearly coded stream for representing math has withstood the test of time and has been adopted into other software without any substantial redesign. T_EX itself is one of the few pieces of software from that period still in wide use.

Since the input is plain text, it is not affected by (most) upgrades to the processing system, and it is hardware independent; the same input will yield the same output, modulo the availability of identical fonts. Knuth's original goal of creating a system that would enable him to typeset his life's work, *TAOCP*, with the same high quality shown by the first edition of volume 1 and remain consistent regardless of how many years have elapsed has been achieved admirably.

Unless something totally unforeseen materializes that is simpler to use and produces results of equally high quality without the need to unlearn the basics of mathematical discourse itself, the situation is likely to remain very much the same in the coming decades.

References

- [Kn] Knuth, Donald E., *Digital Typography*, CSLI Publications, Stanford, CA, 1999.
- [KnCT] Knuth, Donald E., *Computers & Typesetting* (Five-volume set), Addison-Wesley, Reading, MA, 1984.
- [KnK] Knuth, Donald E., "Digital Typography" (commemorative lecture presented on 11 November 1996 in connection with the 1996 Kyoto Prize for Advanced Technology, awarded by the Inamori Foundation of Kyoto, Japan), in [Kn], pages 1–18, http://www.inamori-f.or.jp/laureates/k12_a_donald/lct_e.pdf
- [KnM] Knuth, Donald E., "Mathematical Typography", *Bulletin of the American Mathematical Society (N.S.)*, Volume 1, No. 2, 1979, 337–372. <http://www.ams.org/journals/bull/1979-01-02/S0273-0979-1979-14598-1>, reprinted in [Kn], pages 19–65.
- [KnQ] Knuth, Donald E., "Questions and Answers with Prof. Donald E. Knuth". CSTUG, Charles University, Prague, March 1996. *TUGboat* 17:4 (1996), 355–367. <http://tug.org/TUGboat/tb17-4/tb53knuc.pdf>, reprinted in [Kn], pages 601–624.
- [KnTB] Knuth, Donald E., *The T_EXbook*, Volume A of [KnCT], Addison-Wesley, Reading, MA, 1984.

- [MH] Mills, Ross and John Hudson (editors), with contributions by Richard Lawrence and Murray Sargent, *Mathematical Typesetting, Mathematical and Scientific Typesetting Solutions from Microsoft*, Microsoft Corporation, 2007. http://tiro.com/Articles/mathematical_typesetting.pdf
- [PhH] Phillips, Arthur, "Computer-aided mathematical composition", in *Handbook of Computer-aided Composition*, Marcel Dekker, 1980, pages 202–206.
- [PhR] Phillips, Arthur, "Setting Mathematics". *The Monotype Recorder*, Volume 40 No. 4, Winter 1956. http://www.metatype.co.uk/downloads/mr/mr_40_4.pdf
- [Sa] Sargent, Murray, "Math in Office", blog post on Microsoft's LineServices, November 14, 2006. <https://blogs.msdn.microsoft.com/murrays/2006/11/14/lineservices/>
- [SpJ] Spivak, Michael D., *The Joy of T_EX*, American Mathematical Society, Providence, RI, 1982.

Authors

Barbara Beeton is a long-time employee of the American Mathematical Society, where she has been involved in technical support of typesetting ever since installation of the first computer. She is a founding member of the T_EXUsers Group (TUG) and editor of their journal, *TUGboat*. She has been a representative to U.S. and international standards working groups with a focus on document processing, and she represented STIpub to the Unicode Technical Committee in the effort to expand Unicode to accommodate mathematical notation.

Richard Palais was the Founding Chair of the T_EXUsers Group. He was a member of the AMS Board of Trustees from 1972 to 1981 and its chair from 1977 to 1979. He is professor of mathematics emeritus at Brandeis University, and since 2004 he has been on the faculty at the University of California, Irvine.