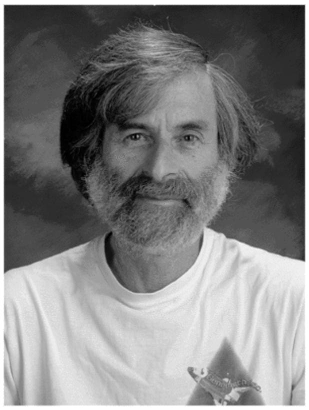

How (LA)TEX changed the face of Mathematics

An E-interview with Leslie Lamport, the author of LATEX

Sehr viel Mathematik, und auch diese Zeitschrift, werden mit TEX oder LATEX gesetzt, und dies hat das Gesicht der (publizierten) Mathematik nachhaltig verändert, und auch das Mathematische Publizieren revolutioniert. Viele Mathematiker setzen inzwischen mit diesen Systemen Ihre Mathematik selbst, und dies hat auch mathematisches Denken verändert, auch wenn noch nicht ernsthaft angefangen wurde, an der Tafel `\sqrt{2}` *statt* $\sqrt{2}$ *zu schreiben … Wir nehmen „ca. 20 Jahre TEX“ als Anlaß, Fragen an Leslie Lamport zu stellen, den Autor von LATEX.* (GMZ)

How were your own first papers produced? Did you start out on a typewriter? On roff/troff/nroff?

Typically, when writing a paper, I would write a first draft in pen, then go to typewritten drafts. I would edit each typed draft with pencil or pen until it became unreadable, and would then type the next draft. I think I usually had two typewritten drafts. I would then have a secretary produce a nicely typed "final" version, which would usually be subject only to minor changes. I went on-line around 1977, using TV edit and a primitive text-formatting system that I believe was called Pub. I switched to Scribe when it became available (maybe 1978?) and switched to TEX perhaps a year later. I first used Unix when I moved to DEC in 1985, so I was never a *roff user.

Could you tell us about the pre-history: Don Knuth wrote TEX in the seventies. It was working but hard to use. People tried, some wrote macros, … What was the situation when you "got started"?

When Don was creating TEX80(?), the second version of TEX, the popular macro package at the time was one written by Max Diaz – I've forgotten its name. I was in the process of starting to write a book, and I found Diaz's macros inadequate. So, I needed to write a set of macros for the book. I figured that, with a little extra effort, I could make a macro package that could be used by other people as well. That was the origin of LATEX.

Was this always meant to be "free software"? Did you ever try to "get rich" with it? Do you regret that you didn't?

At the time, it never really occurred to me that people would pay money for software. I certainly didn't think that people would pay money for a book about software. Fortunately, Peter Gordon at Addison-Wesley convinced me to turn the LATEX manual into a book. In retrospect, I think I made more money by giving the software away and selling the book than I would have by trying to sell the software. I don't think TEX and LATEX would have become popular had they not been free. Indeed, I think most users would have been happier with Scribe. Had Scribe been free and had it continued to be supported, I suspect it would have won out over TeX. On the other hand, I think it would have been supplanted more quickly by Word than TEX has been.

Tell us about your "comic/tragic experiences trying to get computer scientists and computer science journals to enter the computer age"

People will go to great lengths to avoid having to change what they do. In the early days of LATEX, my colleagues at SRI would always tell me that they would write their next paper in LATEX. A few years ago I got fed up with the fact that computer science journals were still sending around paper manuscripts for review. I circulated a message saying that computer scientists should refuse to review paper manuscripts – except in unusual circumstances, such as submissions from third-world countries. One editor complained that she was handling so many papers that the cost of disk storage for all of them would have been prohibitive. A simple calculation showed that, with disk prices at the time, the storage would have cost about $250 – less than the cost of the filing cabinet she was then using. (Now, of course, it would be about $2.50.)

In the late 80's, I proposed to the ACM that they should create standard document styles or macro packages for what were then the three major formatting programs, TEX/LATEX, troff, and Scribe. While their journals would accept paper submissions as usual, authors who submitted papers electronically in one of those styles would have the benefit of electronic transmission speeds. An editor at ACM dismissed the idea because it was unfair to force people who didn't have access to computers to submit their papers electronically. (I can assure you that I'm not making this up; my imagination isn't that fertile.)

People will switch to something new only if they're forced to by circumstances. People started using TEX because pencil and paper became untenable as a way to produce mathematical documents. Journals started accepting electronic submissions when it became impossible to ignore the Internet any longer.

I'm pessimistic about software in general.

Is LATEX hard to use?

It's easy to use–if you're one of the 2 % of the population who thinks logically and can read an instruction manual. The other 98 % of the population would find it very hard or impossible to use.

Why is there no high/same-quality WYSIWYG system available?

The entrance barrier is too high. To have any chance of success, a system would have to do everything that TEX does. That makes it too much work for any individual. A company like Microsoft could do it; I presume they don't because the market is too small. I occasionally think of going over to the Dark Side and proposing to Microsoft that they hire me and put me in charge of a group to develop such a system. Fortunately, I have other things to do that keep me out of trouble.

The speed of modern computers has removed some of the allure of WYSIWYG. TEX can process a 10-page paper in a couple of seconds. I have a simple Emacs macro that, with a single keystroke, processes and redisplays the paper I'm working on. So, when I'm writing a paper, I just have to type TEX source, I don't have to read it.

It's nearly frightening to what extent LATEX has now "solved all the problems" and seems to be without any (?) competitors?

It doesn't have any competitors in the technical sense of competition – that is, there's no other system that can do what it does. In the Darwinian sense, its competition is much too strong for it to survive. Kids these days use Word. As I already said, people are extremely reluctant to learn something new. When those kids grow up, they're not going to want to learn a new, arcane system. So, I expect the use of TEX and LATEX to die out. However, a mathematician just assured me that there is no alternative for math and physics, and he expects TEX to survive the 100 years that Don predicted. We'll see.

You say that people/kids won't "want to learn a new, arcane system." Couldn't it be fun (!) to learn that certain things don't work, exactly because one had made a logical error? LATEX as a computer game?

It's naive to expect something like LATEX, that's at best going to be used only by professional mathematicians and scientists, to filter down to the grade-school level. Even if there were some point to teaching kids such an esoteric system, it couldn't be done for the same reason that it's been impossible to raise the level of math and science education in this country – namely, kids can't learn from teachers who don't know the subject well, and people who are good in math and science don't become grade-school teachers.

Here is a recent email dialogue I had with a colleague in Toronto:

```
>
> "Guenter M. Ziegler" wrote:
>
> >
> > Charming: people [CS professors!!]
> > still use troff! Weren't they forced by
> > law at some point to adopt TeX?
>
> Can't help it, I prefer to type,   .NH 3 than
> /subsubsection etc.
> But then I love unix's two letter commands also!
>
> By the way I can type , eg., .NH 6, does latex
> use /subsubsubsubsubsection ?
>
```

Please comment.

The use of `\subsubsection` instead of `\sss` was a deliberate choice – inspired by Scribe – to make command names understandable instead of short. I think that was a good choice. The user who hates to type can always define `\sss` to mean `\subsubsection`. However, a technical writer typically spends many hours per page writing a document, and the time spent actually typing text is a negligible part of the work. That's probably why neither I nor anyone I know bothers defining shorter synonyms for commands.

One can argue that the use of `\subsubsection` etc. instead of `\section{3}` was a mistake. However, rather than `\section{3}`, a more logical approach would be a `\heading` command that creates a section heading at the current level, and commands to increase and decrease the current heading level. My feeling now is that the intuitive simplicity of the current system outweighs the advantages of the logical approach; but others might certainly disagree.

One thing along those lines that definitely was a mistake was the use of `\small`, `\large`, etc. instead of a `\size{n}` command along with commands to increase or decrease the size. I'm afraid I just copied the size-changing commands from Scribe without thinking.

Any regrets about things you should have done better when you "did it"? Lessons to be learned from that? (Knuth has published parts of his log books ...)

There are lots of mistakes that I made – such as the size-changing commands. But those are inevitable. You can find many of them by looking at the differences between LaTeX2.09 and LaTeX 2ε. But the biggest mistake I made was not in how I designed LaTeX, but in how I didn't design TeX. When Don was writing TeX80, he announced that it would be a reimplementation of TeX78, but he was not going to add new features. I took him seriously and asked for almost no changes to TeX itself. The only change I can remember strongly urging involved page breaking. People who used TeX78 will remember that, when TeX couldn't find a good page break, it would very often produce a horrible one – a page containing two or three lines. I felt that this would be a real show-stopper – much worse than words extending to the right of the margin – so I lobbied hard for the change. However, there were many other improvements that I could have suggested but didn't. In the end, Don wound up making very big changes to TeX78. But they were all incremental, and there was never a point where he admitted that he was willing to make major changes. Had I known at the beginning how many changes he would be making, I would have tried to participate in the redesign. Don had a small group of helpers – mostly students – with whom he met regularly. I could have joined that group and perhaps have had some influence on the design. Who knows, maybe I could have persuaded him to replace TeX's macro-expansion language with something better. A macro-expansion language is good for a quick-and-dirty solution, so it was appropriate for TeX78. But it's not good for serious programming because you always have to fight to get things expanded at the right time.

Three LaTeX mistakes that people should stop making?

1. Worrying too much about formatting and not enough about content. 2. Worrying too much about formatting and not enough about content. 3. Worrying too much about formatting and not enough about content.

What's your view on mathematical typesetting in the future? Quantum leaps ahead?

I'm pessimistic about software in general. When computers were the province of the technically sophisticated, people wrote software for technically sophisticated users. Now, technically sophisticated users are an insignificant niche market. Standards are being driven by the marketplace, which cares only about the masses. So, mathematicians have no place in the brave new world of computing. They will have to make do with the same flashy but technically impoverished tools that the little old lady in Peoria uses. So, you can display video animations on the web, but there's still no good way to display a mathematical equation.

Mathematicians have no place in the brave new world of computing.

The future of technical communication is the World Wide Web and the CD-ROM. There may soon be a window of opportunity for two products: one for "typesetting" math for the Web, and the other for creating CD-ROM textbooks. The proposed standard for adding math features to html will, if adopted, make it possible to produce poorly formatted but readable math html documents.

Computers make possible all sorts of new forms of communication. For example, one can have a sort of permanent workshop which consists of a set of technical presentations combined with a chat room. Based on the chat-room discussions, participants can continually refine the technical presentations. It could be something like a "living Bourbaki" for a subject.

However, mathematicians, like all people, are extremely conservative. For example, they still write proofs essentially the same way they've been doing it for centuries. I believe I've demonstrated in

```
AUTHOR  = "Leslie Lamport",
TITLE   = "How to Write a Proof",
JOURNAL = "American Mathematical Monthly",
VOLUME  = 102,
NUMBER  = 7,
YEAR    = 1995,
Month   = "August-September",
Pages   = "600--608"
```

that there's a better way. But they are just as reluctant to try it as they are to try anything new. Their excuses make no more sense than the ones I heard 15 years ago to explain why they weren't switching to (LA)TeX.