

anyone working in a production environment. Your participation is encouraged.

### **TUG: the present**

TUG is going through some changes.

Sometime very soon after I write this, the TUG office will be moving from Providence to Santa Barbara. A new Executive Director, Pat Monohon, will be taking charge. For much of the past year, Pat has been in charge of a group of volunteers who have been copying and distributing the public domain  $\TeX$  packages that are available from TUG. So she has already begun to become familiar with some of the functions that are part of the TUG office duties. We wish her well.

I would also like to take this opportunity to recognize everyone who has worked so diligently in the Providence TUG office: Karen Butler, Cliff Alper, Teresa Pires, Kathy Sheely. They have been unstintingly helpful whenever I've had questions, and I shall miss working with them. And I'm not forgetting Ron Whitney—his contribution to making my job easier has been greater than I can say; in addition to managing the office, he has continued to respond to my requests for assistance with the *TUGboat* styles, and the credit for their reliability and ease of use in production is mainly his. Thank you all!

Not only is the office undergoing a transition—the elected management will be changing too. As already announced in *TeX and TUG News*, no one stepped forward to stand for election as TUG president, so the Executive Committee has studied the Bylaws for guidance in this situation. There will be a new president for a term beginning on January 1; the specifics will be announced in the next issue of *TTN*.

Please remember that this is your organization, and its success depends on you. Let's all of us, every TUG member, pull together and give our new representatives the support they deserve.

### **New ideas in typography**

Earlier this fall I attended the Goudy Award Symposium at the Rochester Institute of Technology. This annual event, named for the esteemed American typographer, honors an outstanding type practitioner with an award and a program of talks about type and typographers.

One of the speakers this year was Peter Karow, of URW, Hamburg. His talk described a new typesetting program—the *hz-Programm* (named for Hermann Zapf, who designed the fonts on which

it depends)—which optimizes text in such a way as to achieve nearly equal word spacing throughout each paragraph, using several interesting techniques that I believe are of interest to readers of *TUGboat*.

The first technique should be familiar already: paragraph-wide line breaking; this is, of course, the technique used in  $\TeX$ , and though Karow didn't mention it during his talk, when I asked if the source were Knuth's algorithm his answer was "Of course!"

The second technique depends on specially designed fonts to lengthen or shorten lines to approach the target measure. For selected letters multiple shapes of differing widths are provided; typically these are letters that occur relatively frequently (e.g. "e") or have shapes that can by a small change have a significant effect on line length (e.g. "m").

A third technique can be called "intelligent kerning"; this consists in applying only positive kerning to lines that are shorter than the target length, and only negative kerning to lines that are longer.

There are two more components to the package. The typographic quality of the resulting text is most impressive. The paper will be published elsewhere, but I have asked, and expect to receive permission, to reprint it here in a future issue.

◇ Barbara Beeton  
American Mathematical Society  
P. O. Box 6248  
Providence, RI 02940  
USA  
[bnb@Math.AMS.com](mailto:bnb@Math.AMS.com)

---

### **An Interview with Donald Knuth**

In November 1991, Donald Knuth was one of a select group honored by appointments to Honorary Doctorates by the Royal Institute of Technology (KTH), Stockholm (see *TUGboat* 13, no. 2, p. 134). After his installation, he participated in a meeting of the Nordic  $\TeX$  Users Group, where he responded to questions from the audience. He also spent some time talking informally with Roswitha Graham, the president of the Nordic group. These discussions were recorded, and edited transcriptions appear here with the permission of the participants.

Don also spent some time relaxing at the Gramhams' country house on an island in the archipelago outside the Stockholm harbor. A helpful spider provided an appropriate setting for the author of WEB.



#### Question and answer session at the Nordic group meeting

**Donald Knuth (DEK):** I would like to say a big “Thank you” to Roswitha, and to the School of Computer Science and Department of Mathematics, for making my visit here possible and arranging everything. Also, I’m glad to be back here—I wanted to say that T<sub>E</sub>X owes a lot to the Nordic countries and to Sweden in particular.

I guess I never mentioned this in print, but when I designed T<sub>E</sub>X, I chose three examples of mathematical typesetting that I considered as standards of excellence, and I studied those three very carefully. I scanned them digitally—in those days we used a TV camera—and made a lot of careful measurements. One of the three was a volume of *Acta Mathematica*, printed in Stockholm about 1910. The second, in case you’re wondering, was from the Netherlands in 1950, the mathematical section of their Academy of Science proceedings; the third was Addison-Wesley’s house style as used in *The Art of Computer Programming*, when math composition was still done by hand.

Also, here at KTH you had one of the first four Alphatype machines running my original software. I received a copy of an early publication—I don’t

remember the name of it now, but I remember it had a green cover and it was a nice booklet about 40 pages long—a mathematical report. [It was *Non-linear Inverse Problems* by Gerd Ericksson (1983), 46 pp.; see “T<sub>E</sub>X incunabula,” *TUGboat* 5, no. 1 (May 1984), 4–11.—ed.] So some of the very first extensive uses of T<sub>E</sub>X happened right at this institution. That makes me especially pleased to be here.

At the end of my courses at Stanford, I usually reserve the last day of class for a session we call “All questions answered,” and I volunteer to answer questions on any subject whatsoever except religion or politics. And in recent years I’ve also excluded questions about Volume 4 of *The Art of Computer Programming*. [laughter] But today this looks like a very friendly audience, and I don’t even mind if you have questions about Volume 4 of *The Art of Computer Programming*. So ask any questions whatsoever, go ahead. (Including politics.)

**Question:** I have a question about Chinese characters used with T<sub>E</sub>X.

**DEK:** Well, I don’t know too much, but I’ll tell you what I do know. One of the first scientific visitors from the People’s Republic of China to the United States was Dong YunMei, who came to work with me and designed a system called LCCD, Language for Chinese Character Definition; he wrote a Stanford Computer Science report about that work. It was inspired by METAFONT, but he had special graphical primitives in there so that LCCD could do things that were especially important in Chinese for positioning radicals and so on. It was a language that wasn’t based on, but was similar to METAFONT; he implemented it entirely himself. In the second edition of Volume 2 of *The Art of Computer Programming* (which was published in 1981, and was the first real use of T<sub>E</sub>X78), we used LCCD to typeset the Chinese names in the index. Then Dong went back to China, and now he heads a group at the Institute for Software Technology in Beijing. That group has, of course, done a lot toward extending his work since those early days. They now have a language called SP for typesetting Chinese and arbitrary Western texts; it was mentioned in the proceedings of the IFIP Congress, 1989, held in San Francisco; there’s a brief, 5–6 page report about what is going on in computer research in China, and this is one of the projects mentioned. When he was at Stanford, he spelled his name “Tung”; now he spells it “Dong”, but it’s still pronounced “Doong”. He’s a very fine man. The most exciting thing for me is

that he's now directing a big project to use literate programming for Chinese, so they have a good way to document their programs, like the WEB system for documentation. This system, called CDP, combines the C programming language with the SP typesetting language. They also have an ongoing project for drawing Chinese characters electronically with a kind of metalanguage. However, I don't think they're anywhere near doing all the characters that way. That part of his group's activity seems to be more of a research project than a real production project.

There was another visitor from China at the same time, 1980—Mr. Ma, who worked at the Mathematics Department of the University of Beijing. He had some pretty good Chinese fonts at that time, which we could use on our laser printers. But that's all old stuff.

Just three weeks ago, I had a chance to meet Mr. Gu Guoan again. He's from Shanghai and now working for Ikarus, so he is affiliated with Peter Karow in Hamburg. Gu is the head of the Ikarus bureau in Shanghai. He originally came to Stanford from the Shanghai Printing Company, which is the largest printing company in China, shortly after Dong and Ma had returned to China. Gu wrote a paper with John Hobby that is mentioned in *The METAFONTbook*, describing about a METAFONT system for Chinese characters. That paper [*TUGboat* 5, no. 2 (1984), 119–136—ed.] was published as a Stanford Report and has lots of illustrations. They designed a system that did about a hundred test characters, all programmed in terms of 19 basic strokes. One of the basic strokes, for instance, is [draws on blackboard]. But they add a lot of parameters to it, so that it can appear in many different proportions. Then they have another basic stroke that looks something like this [more drawing]; you can have it tipped in different ways, and you can control exactly how much of a teardrop you have, and so on. So you have nineteen of these basic strokes, all done with parameters. Then you design the hundred Chinese characters by saying “do stroke number 1 in this position, and stroke number 2 in another position.” The almost incredible thing about their system was that you could redesign the nineteen primitive subroutines and get completely different styles, still looking like real typefaces; so you could get Chinese in bold style, and you could get Chinese in the Ming Dynasty style, from the exact same descriptions of the hundred characters, just by changing the nineteen subroutines. Their article presented a kind of sans serif as well as a seriffed version, and they

also had what they call a “Long” style or something like that. So they were getting three styles, and the design was a true meta-font in that sense. Their work was quite exciting to me, but as far as I know no one has pursued it; it was done by university researchers, who assume that when they're done, industry will march in and take over, but that didn't happen. I don't think there were any flaws in the research; I think it's just a matter of somebody picking up the idea and matched it with the right group.

On the other hand, Gu is now working for Ikarus, and is about to finish a massive conversion of Chinese characters into Ikarus format, then into PostScript with “hints”; this represents the entire set of all known Chinese characters—more than 50,000 but less than 60,000. He was visiting Adobe, so I happened to meet him a month ago when he came to California. His Ikarus descriptions won't be a meta-font, since they represent only one style. But he has access to the very best fonts, because the Shanghai Printing Company has these in the form of photographic masters.

— \* —

### Conversation with Roswitha Graham

**Roswitha Graham (RG):** Was anything new brought to you at the Nordic meeting, anything that you didn't expect to be there, as you listened to Frank [Mittelbach] and Yannis [Haralambous]?

**DEK:** As I said earlier, I was very impressed by the quality of the work that they're doing. I knew Frank's work before, because I'd talked to him at length when he came to Stanford. He had very penetrating insights into the whole program and has found many of the most subtle bugs in T<sub>E</sub>X. Also I know that he has this great commitment to quality publishing where he really wants to go a level beyond what I had ever conceived that I could do. So the proposals that he's made for going into even finer-than-fine quality printing, I know he comes from the right spirit. He might discover, as I did, that when you make some things better, other things get worse, and so finally you have to find a balance. But he might find a way to make everything better. So it's very inspiring to me to see the time that he puts into this and the vision that he has for making further advances. Also, an international perspective is important—people with different backgrounds from my own can obviously contribute things to T<sub>E</sub>X that I couldn't do myself; I'm only one person.

Then I met Yannis for the first time. His work... that was the biggest surprise for me, to see how much he has achieved singlehandedly, doing things that I thought would need half a dozen or ten people to do, because of his background in many languages. Again, he has this great love for quality and love for preserving the beauty of "the good old days" in printing (where sometimes modern progress has taken steps backward). It was certainly exciting for me to see the care that he is taking to make keyboarding of so many kinds of text reasonable. When I was working on T<sub>E</sub>X, I wouldn't have dared to dream that it would become so easy to use the system in connection with Hebrew with vowel points, as well as Greek with all the breathings, and Ethiopian and Urdu and Arabic and many of these other things.

**RG:** Will you try this when you come home?

**DEK:** Well no, my life is too short. There are so many things that I could love very much spending my life doing—I could spend a lifetime doing fonts; I could spend a lifetime playing with color; I could spend a lifetime doing many languages. But I've already decided that the rest of my life is going to be *The Art of Computer Programming*. That's at least a twenty-year job, so I can't take any time off to do anything else from now on.

**RG:** I meant actually if you are going to try, not if you are going to work with it... just play with it?

**DEK:** I usually allow myself one or two days a year to play with some toy, so it's quite possible that I'll do that, one of these years.

**RG:** Did you get some mail from Beijing? We just got mail from Beijing, what this Chinese fellow had done. He had done some preprocessing.

**DEK:** Yes, that's thrilling for me too. People all around the world feel the same need for quality publishing that I had for mathematics publishing. So when I see people putting a lot of spare time into the work, I know that this is rewarding to them, and to many people afterwards having the fruit of their work. That's why it's exciting to me, of course, to have helped make this possible. I now see many publications where the authors were just inspired to create a fine book that they would never have begun if they hadn't had the power to do it themselves, and they needed something like T<sub>E</sub>X or METAFONT in order to be able to do it at all.

**RG:** Could T<sub>E</sub>X be an ideal program to communicate between different languages? Do you think it could be used in that way as well?

**DEK:** Well, as far as I know it's the best existing system to control things like paragraphing while dealing with many languages with respect to hyphenation and so on. But certainly I never thought that T<sub>E</sub>X would be the last word. In fact, one thing hasn't happened that I was expecting: When people were faced with special kinds of publishing projects, I was assuming that they would just make their own version of T<sub>E</sub>X instead of trying to use T<sub>E</sub>X as it is. For example, if someone wanted to come up with a new edition of the Bible, I was really expecting them to use some WEB change files, hiring a programmer who would be able to give them new features adapted to their project. You know, if you are going to produce an Arabic/English dictionary, or something like that, you'll want special capabilities. I never expected T<sub>E</sub>X itself to be a general tool for all these purposes. Now I find that people are close enough to being able to do most of these things, that hardly anybody is actually changing the WEB code. Instead, they've found that they could do almost everything they want with standard T<sub>E</sub>X. Yannis found a way... I said, are you worried that I have only 256 characters in the font when you're dealing with Greek? (The first fonts of Greek that were developed in Europe, in the Netherlands, had some six hundred characters including a huge number of ligatures.) But he assured me that he's not at all worried about this restriction. He doesn't find the existing system problematical.

Now I see here the report from Chinese T<sub>E</sub>X, and it's got chemical formulas as well as Chinese and mathematics, done with a preprocessor. Of course, to do some of these things you can't expect macros would handle everything the easiest way. My goal was to build an engine that would be versatile, that would be very good at balancing texts and able to do hyphenation in a variety of languages, and so on, and I was hoping that it would facilitate printing the languages of people all over the world—and now here it is!

Since Professor Romanovskii is with us today, I'm reminded also of recent developments in Russia. We all know that when people do some work they're proud of, they also want to make it look like something they're proud of. In America, I couldn't write a book that I thought was going to be produced by a printer who wouldn't do it well. In the Soviet Union, when it was government policy not to have easy access to Xerox machines or laser printers, the policy was holding everybody back from being able to communicate their best work. So it's also a big hope of mine that a new

... freedom isn't the right word for it, I guess I mean an ability to communicate to the world community in mathematics ... will lead to a great flourishing of new Russian books. So much Soviet mathematics I've seen had to be condensed into a few pages. I'm sure the authors were just as proud of their results as we are—deservedly even more proud—but just had no way to let it out. It's the same with computer programs as with mathematics. Computer programs need to be communicated, to be read by people.

The communication factor is the real, to me the most important reason why so many changes began to happen in the Soviet Union. The rest of the world was improving its communications rapidly, while in the Soviet Union, computers were used mostly for numerical calculations and couldn't be used extensively for word processing; it was too much against the policy of the government. So this need for communication, also for beautiful communication, is something that I had long wished for in every part of the world, but especially in the Soviet Union. Jill can tell you about times when I would play Tchaikovsky on the piano, with tears in my eyes because I was thinking how beautiful this music is, but how I wished that my friends in Russia could be doing some of the other things that I was doing.

Communication between machines has become the real reason for computers in most of the world. China faces the same problem; they can never reach modern levels of technology until all limitations are removed from their communication networks.

**RG:** What difference is there between the Japanese way of doing T<sub>E</sub>X and the Chinese way?

**DEK:** I'm not familiar with it, but in both cases they seem to be getting quality. So I'm happy, as long as the output looks great.

I get completely unexpected mail every once in a while. Did I tell you the story about Italy? I guess not. *The T<sub>E</sub>Xbook* has been translated not only into Japanese, but also into Italian (though not yet published in Italian). The people who did the translation are actually prisoners, political prisoners thrown into jail in the '70's because they were liberal activists. But they're in a minimum security prison, so they're supposed to work for their living. This one group decided that their job was going to be to typeset mathematics with T<sub>E</sub>X for the Italian Mathematical Society; so they did this. They sent me a letter explaining how they're prisoners, and they enclosed a laser-printed copy of the entire *T<sub>E</sub>Xbook* translated into Italian, 500

pages of it, a beautiful job. They claim they are enjoying their prison life because of T<sub>E</sub>X. What could be nicer?

**RG:** That's a point of view I never thought of, actually. I find it great that you have given to the world something that is free of charge and it's a challenge for everyone who wants to communicate. If the possibility is given, I think it's wonderful there is something like this. I find it amazing that so many people have put in so many working hours without ever asking any labor union about it.

**DEK:** It was part of my original thinking that T<sub>E</sub>X should not be competing with a labor union, or even with other people's egos. In other words, I knew that if I had been working at Bell Laboratories, for example, I could never do anything that would compete with troff, because that would be not respectful to the other people there who had put years and years into it. To me, troff had proved that the whole idea of something like T<sub>E</sub>X was possible, and therefore it was a good idea to start over from scratch and think over how to do it if you were starting over. Well, it's impossible to do such things in an existing organization. Similarly, if I hadn't done T<sub>E</sub>X for mathematics, if I had had as my first goal to do newspapers, or the yellow pages, or something that counts for the majority of publications in the world, then (if successful) I would have been putting a lot of other people out of work and making them angry at me. I didn't want to do that. So I was happy to be creating a tool for mathematics; it seemed to me that nobody in the world enjoyed typesetting mathematics. Good—I wasn't going to offend anybody. This was a strong component of my thinking as I wrote T<sub>E</sub>X. Well, it turned out there was one person who was offended, and he complained at one of the TUG meetings; this man from Science Typographers had spent a lot of his time making a commercial system, which is still used. He keeps improving it, and it's excellent. There's absolutely no reason to question the quality of the journals his company typesets, in any way. But he did complain, and that made me feel bad, because I didn't think I was going to make anybody unhappy by doing T<sub>E</sub>X.

Another thrill at this Nordic meeting was to learn that T<sub>E</sub>X will soon be used to typeset the journal *Acta Mathematica*. The main reason it's exciting to me is that a circle is being closed: *Acta Mathematica* was one of my first models for quality when I designed T<sub>E</sub>X. As I said at the meeting, I had selected three standards of excellence that I did a lot of measurements on, hoping to be able to

match their quality. *Acta Mathematica*, in 1910, was the mathematics journal that had the best budget in the world at the time, and they did a very good job; so I had looked at it very closely in 1977. Now comes the surprise: Right after the T<sub>E</sub>X meeting, I ran into Jan Michael [Rynning] at the Institute talking to people about how to do *Acta Mathematica* with T<sub>E</sub>X! The decision has apparently been made now. Not only that, but Leif Andersson showed at the Nordic TUG meeting that he understood all the intricacies of making new mathematics fonts for T<sub>E</sub>X. So it should be possible to do *Acta Mathematica* in the fonts that you had in 1910 as well.

**RG:** Did they agree to?

**DEK:** Who knows, but he could certainly do it. The expertise exists in Sweden to make this possible.

I tried to design T<sub>E</sub>X so that people who would be spending a lot of time working with it would still find their job pleasurable. Suppose they have to deal with T<sub>E</sub>X many hours of every day for years and years. I tried to make it so that they would still find this a pleasant job, because they could keep discovering new constructions that would be somehow fun or beautiful or satisfying, in amongst all the other things they have to fight in order to cope with unusual constructions that always come up in printing. My hope was that people could continue to find yet more elegant ways to solve certain problems, and enjoy the process. The number of people around the world who are putting in this extra effort seems to indicate that the language is working in this respect as I had hoped.

**RG:** Just being interested in people communicating . . . , because I am visualizing that it will be possible to have this text in a mail and it could even be translated by the language.

**DEK:** Oh yes, we could build it into fax machines and so on. I hope that in January I will see the absolute last bug in T<sub>E</sub>X, and somebody can make a chip so that it will be easy to have T<sub>E</sub>X inside any machine!

The American Math Society is making T<sub>E</sub>X source available now; the *Math Reviews* are available in T<sub>E</sub>X form going back fifteen years or so, and they continue to extend this so that mathematicians can look for an article on a certain topic and can see it typeset on the screen. And this is something that could certainly work for all sorts of complicated applications.

But you still need, if you are to get the highest quality, to have copy editors review everything and make sure you have the right style of quotation marks, for example. Every academic discipline has its own problems and every standard reference work can use typography in ways that make the reference work more effective but also is specific to that book. So every book is a new challenge. The advantage of T<sub>E</sub>X is that it's able to be adapted to these different challenges; it doesn't put everything into a single format. On the other hand you could also consider that a disadvantage.

Suppose you were allowed to rewrite all the world's literature; should you try to put it all into the same format? I doubt it. I tend to think such unification is a dream that's not going to work.

Authors have adapted themselves to the medium that they are using at the time, and if we're going to understand the authors' intentions, we should see something close to that medium. The medium of the future may be some other sort of standard for international communication; whatever the future brings, people can use that method. Authors will express themselves best in the medium they are using for creation. Authors using T<sub>E</sub>X now can, for example, write a different kind of mathematical paper than they did ten years ago, because they know they can choose a notation that isn't going to be misunderstood by a typesetter who knows no mathematics. So I can write a more effective technical paper if I know that I can make the diagrams correct—I can put in different kinds of diagrams and tables than I would have dared to if I was going through other people who didn't understand my intentions. Things that were written 20 years ago were written in a style that was optimized for older technology. Movies are analogous: People who made silent films chose different scenes than they would in a sound film, but their work was effective as a silent movie. And you can't just take a movie that's black and white and colorize it and get the same effect. It's the same thing with older texts as well as texts for special purposes and other languages. So this idea of a standard format is only something for the future, not for the past.

**RG:** You can't leave the field! You'll follow up in some way what's happening with T<sub>E</sub>X, even if you now are going to devote your time to something else, you cannot let it go. You have spent so much time. . . Is it a hate/love, or is it only love?

**DEK:** Well, it's hate when I learn that I made yet another error. I just got a message in the mail saying that if somebody sets the math unit to a

negative value—which nobody in their right mind would ever do, but if they do—apparently  $\TeX$  goes crazy. So I have to fix that. Ugh.

What I love is when excellent new publications come out that I know wouldn't have been done at all without  $\TeX$ , and also when I see people—as I said, Frank and Yannis—spending a considerable part of their lives doing work that has very high quality. They are excited just by the chance of improving the quality of publication. Those are the things that make me happy.

**RG:** Thank you very much.

## Dreamboat

### Moving a Fixed Point

Richard Palais

#### Abstract

In the past few years there has been increasing discussion of the question “Has the time has come to make basic changes to the inner workings of  $\TeX$ ?”. In late May of 1992, Rainer Schoepf set up a mailing list on the Internet, called “NTS-L”, to discuss the matter. I started out being completely opposed to the idea of even the slightest changes to the  $\TeX$  code, feeling that whatever failings  $\TeX$  might have, they are best approached by pre and post processing (“front and back ends”), and anyway are negligible compared to the danger of losing the remarkable coherence and interchangeability of  $\TeX$  software, everywhere and on all platforms that is enforced by the discipline of having a single, universally accepted underlying piece of software (INITEX). However, after following the discussion carefully for nearly two months, I was convinced by evidence that, for certain purposes,  $\TeX$  was no longer fulfilling its promise of providing typesetting of uncompromising high quality, and probably only careful and limited changes and additions to  $\TeX$  primitives could correct this. What follows is a long message I posted to NTS-L, outlining a minimalist approach to changing  $\TeX$ , and also a suggested method for implementing changes to  $\TeX$  code that would insure documents written for standard  $\TeX$  could still run under the new system. A number of replies to my message were posted to NTS-L and others were addressed to me personally via email. Rather than incorporate these comments by making appropriate changes to the version I posted, I have

decided to append a short addendum, mentioning a few of the more important points made in these replies.

-- \* --

This is going to be more a “position paper” than a simple message. I have been following the NTS-L mail list discussion with considerable interest and finally felt that there were so many issues that I wanted to address and remarks that I wanted either to agree with or to dispute, that only a fairly extensive reply would do. Here is a table of contents:

- Section 1: Introduction
- Section 2: The Many Faces of  $\TeX$
- Section 3: A “Standards” Approach to Solving  $\TeX$  Portability Problems
- Section 4: The Matter of Compatibility
- Section 5:  $\TeX$  as a Front End
- Section 6:  $\TeX$  as a Programming Language
- Section 7: Changing the Fixed Point
- Section 8: Summary
- Section 9: Postscript

### Introduction

First a short personal introduction. The oldtimers of the  $\TeX$  world will perhaps remember me—I was the founding chairman of TUG, worked closely with Don Knuth during the early years of  $\TeX$ , and I wrote a column on mathematical typesetting in the *Notices of the AMS* for three years, with the goal of easing the transition in the mathematical community from the typewriter, along WYSIWYG road, and into the bright new Promised Land of  $\TeX$ . But my name may well be unfamiliar to more recent arrivals in the  $\TeX$  world, for lately I have been only a “lurker” on comp.text.tex, and while I read *TUGboat* and use  $\TeX$  daily for writing my letters, papers, and books, and in connection with my duties as an editor of the *Bulletin of the AMS*, I have not recently been contributing either to the development or to the public discussion of  $\TeX$ .

Next a disclaimer. While I know my way around in *The  $\TeX$ book* and have been writing my own macros and formats since 1978, I consider myself an amateur, not at all in the same league with  $\TeX$ perts like Barbara Beeton, Michael Downes, Victor Eijkhout, Karl Berry, Larry Siebenmann, Tim Murphy, and others who have been contributing to this discussion. So I will happily defer to them on technical matters and hope that they will correct any of my misstatements. What I would like to do is take the point of view of a devoted  $\TeX$