

Stephen A. Fulling

KEYNOTE:

TeX and the Web in the Higher Education of the Future: Dreams and Difficulties

Stephen A. Fulling

Mathematics Department,
Texas A&M University,
College Station, Texas,
77843-3368 USA
fulling@math.tamu.edu

Abstract

New technology provides an opportunity to move mathematical and technical education out of the lecture-homework-test mold into modes that do more to develop students' communication skills, teamwork, attention to quality, and overall responsible, mature behavior. In practice, however, severe and presumably unnecessary obstacles are encountered, mostly connected with the difficulty of transmitting mathematical notation electronically.

As I document from personal experience, these obstacles are bad enough for an instructor providing information to students over the Web, and worse when students themselves are expected to communicate mathematically. I describe a partially successful scheme for carrying out peer review of homework papers over e-mail and the Web, resulting in a student-written solutions manual, and I offer a wish list of technical improvements.

In memory of Norman Naugle and James Boone

There are two kinds of people, classified by their reactions to new technology. People of the first type say, "Now we can do new things — things that were impossible before." The other people say, "Now, finally, for the first time, we can do the old things

right!" Tim Berners-Lee, who made this conference possible and necessary by creating the World Wide Web, surely belongs to the first camp. Donald Knuth, with his injunction to go forth and create beautiful books, is perhaps the most successful exemplar of the second mode of thinking.

The advance of computer technology is having profound effects — actual or projected — on undergraduate education. Most of the discussion takes place in the first mode: laboratories with computer algebra systems, pedagogical Java applets, distance education to reach new audiences, greater attention to numerical methods in the content of courses, etc. Personally, I feel more capable of contributing in the second mode, using the computer, the Internet, and the Web, mainly as marvelous communication tools that address the discontents of modern mass higher education. Here I shall describe two efforts in this direction, with emphasis on the frustrations I

encountered because our two most marvelous tools, TeX and the World Wide Web, do not communicate well with each other.

I am not a computer professional. I'm a university teacher of mathematics, trying to make significant contributions to that art while striving to keep a research career alive. This gives me little time to write my own software, and not even enough time or facilities to find and evaluate all the software that already exists. It is a great honor and responsibility to be scheduled as the leadoff speaker of this Annual Meeting. The professionals should regard me as something like Dilbert's Boss (Adams, 1996, for example): I will present you with a list of technical demands; some of them may be ridiculously impractical; the rest are those you must implement to keep academic customers happy in the next decade. Your job is to distinguish between the two categories and to take action on the good half. I am making an effort not to dwell on any existing partial solutions that I may be aware of, because I know that my information is incomplete; it is up to the providers of those solutions to speak for themselves.

What I want for myself yesterday (or at least tomorrow)

Let's start with the obstacles facing an instructor who wishes to make text material with significant mathematical content available to students over the Web.

For two academic years (from 1996 to 1998) I was one of two calculus teachers in an experimental freshman engineering program integrating calculus, physics, English, chemistry, and two introductory engineering courses into a unified curriculum. (This is a product of the Foundation Coalition, a consortium of universities and colleges supported by the National Science Foundation.) To meet the demands of a complete physics sequence in the freshman year, it was necessary to revise the traditional calculus syllabus by moving large quantities of multivariable material from the third semester into the first two. No textbook on the market does that! Therefore, I wrote some textbook fragments covering vectors, multiple integrals, Taylor series, and a few other topics, at the level and from the point of view needed. (See the URLs listed at the end of this article.)

As a long-time user of T_EX, my first inclination was to write everything in T_EX and post the resulting Masterpieces of the Publishing Art onto the Web. But that smelled too much of using the computer monitor as a bulky imitation of a printed page. In particular, hypertext features would not be available. Also, the screens would not have the look and feel to which my Generation X audience is accustomed—possibly a serious tactical error in winning their confidence and engaging their enthusiasm. I went to the opposite extreme: I wrote the documents in HTML, in the hypertext and bulleted-list style. Whenever possible, the mathematical expressions were written in HTML. Whenever I just couldn't stomach the results, or a passage was particularly heavy in formulas, I wrote a patch of T_EX and linked the .dvi file to the higher-level HTML page. (Since not every browser the students were likely to use was configured to handle .dvi files, PDF versions had to be supplied as well.) The results still have a somewhat amateurish appearance because the T_EX (and the graphics) are not properly integrated into the HTML pages. (With considerably more work and self-education I could have done better, but I did not have the time or the expertise.)

But the world shouldn't be like this!

1. NON-NEGOTIABLE DEMAND: It must be possible to place all standard mathematical symbols

on an HTML page, including the Greek letters, integral signs, square roots, built-up fractions, etc., with confidence that they will be displayed properly by every standard browser. ("Standard" may need to be redefined but the new standard must be implemented!)

2. The screen appearance of the mathematical expressions must be of "typeset quality". NOTE: this does not necessarily mandate Computer Modern fonts. Indeed, one of the obstacles to widespread acceptance of T_EX on the Web is that the CM fonts were designed for high-resolution devices, and the more slanted ones, especially, are barely readable on screen except at very large magnifications. Some compromise in the design of the math fonts may be necessary and acceptable.

I think that anyone who has tried to type mathematics in HTML, even when all the necessary characters and features (such as superscripts and subscripts) are available—or who has looked at the syntax of MathML—will agree with the following requirement:

3. It must be possible for the author to create mathematical expressions by typing raw math-mode T_EX code into the HTML (or XML) file (inside some SGML-type wrapper, of course).

You'll note that what I'm calling for here is "encapsulated T_EX" (Murphy, 1991). Indeed, for many purposes I am perfectly happy with the *text* of an HTML document as browsers display it on the screen and PostScript printers print it out. It is the mathematical expressions that are unacceptable at present. Moreover, the new MathML language in its raw form is not writable or readable by human beings—it would be like writing a text document directly in PostScript. A human-oriented language is needed for input to MathML. But we have such a language—it is T_EX! The problem is to get the rest of the world to recognize it as the preexisting standard that it is.

This leads back to a question that I'm sure many of you wanted to ask earlier: Why don't I use a program like `latex2html`? In part, that is a personal prejudice; my natural language is `plain`, with an accumulation of personal macros, and I would need to translate my files into L^AT_EX to use such programs. But I refuse to make that effort because, in any event, I can't accept the results as a permanent solution:

4. The mathematical expressions must resize properly when the user changes the font size in the browser.

5. The mathematics must reside in the HTML file, not in dozens of auxiliary files.
6. The system must not rely on an engine located on a remote third-party machine. (This seems to rule out MINSE, for example.)

So, that is what I want for myself. And I want it yesterday — because the curriculum pilot project is now over. As usually happens, something much less ambitious was mainstreamed by my institution; the course materials produced for the experimental classes remain on the hard disk, as a monument to ... something.

What I want for my students today (or at least next year)

I frequently teach courses for engineering and science majors in their last two undergraduate years, covering topics in mathematics such as linear algebra and Fourier series. I have concluded that, even at this advanced level, our standard teaching practices do very little to develop habits of mature, independent, professional behavior. This is not the place for a harangue on pedagogical philosophy. (See references under “Homework review” below.) Suffice it to say that we stand at a crossroads: *Educators can use computers to restore literacy, or to drive the last nails into its coffin. Freedom from paper is not the same thing as the death of the alphabet.*

One of my attempts to address this situation is a drastic change in the handling of homework assignments. Instead of requiring every student to turn in every exercise on the list, I assign each student only one or two problems per week. But these problems must be taken seriously: The instruction is to produce *a carefully written, complete, formal solution, comparable to a worked-out example in a textbook*. The class, collectively, solves most of the problems, thereby producing a “solutions manual”. This is a class project, their legacy to the students of future semesters. All the papers turned in on one exercise are given to a team of two or three students for peer review. Subject to quality control by the paid grader and myself, they write a brief critique of each paper and choose the best one for “publication”. The criteria for evaluating papers include presentation as well as correct content; in descending order of importance, they are:

- mathematical correctness
- completeness
- organization
- pedagogical effectiveness

- sentence structure and punctuation
- brevity and style
- grammar and spelling
- neatness

After several years of working purely with paper, this system moved to e-mail in 1996. I expected that keeping the reviewers’ reports electronic would cause a huge increase in efficiency, and for me it did; however, I soon faced a revolt from the students, who said they were spending too much time on keyboard busywork instead of learning math, and from the grader, who found the chaotic e-mail files harder to work with than paper. Then a miracle happened: a student in the class, Justin M. Sadowski, showed up in my office with a C++ program to automate the mailing of reports. This program operates on the departmental UNIX systems where our students have accounts. A menu guides the student reviewer to type in the account names of the student authors, and the program spawns sessions of the `pico` editor to write the reviews and a summary report to the instructor into template files. At the end it mails the messages to the authors, grader, and me.

Students are strongly urged to make their homework solutions visible on the Web as well as on paper. The long-range goal is an on-line solutions manual, recreated each semester by a new class. But since these are courses in mathematics, not computer usage, the system must run on the students’ preexisting computer skills, which vary widely. At present I estimate that about:

- 50% of the papers are still hand-written
- 20% come from a word processor but are not Web-displayed
- 20% appear on-line, either in ASCII text or in HTML, the latter usually converted either from Microsoft Word, or from *Maple* (which all our students learn as freshmen)
- 10% are written in $\text{T}_{\text{E}}\text{X}$

These divisions are moving up (in the higher-tech direction) every year.

Why don’t more students use $\text{T}_{\text{E}}\text{X}$? Well, one of my students wrote:

I spend enough time trying to get my programs to compile. I don’t want to have to debug my [homework] papers!!

They are not necessarily encouraged by their senior mentors, either. Although $\text{T}_{\text{E}}\text{X}$ has become the indispensable standard in academic mathematics and physics, the situation is different in engineering, where most of my students come from. Two years ago in a discussion at a conference on engineering

education I suggested that we encourage the use of T_EX, and the reaction of an engineering professor was, “Surely you’re not still using that old thing!” He couldn’t believe that mathematicians hadn’t all switched to Microsoft Word. The fact that most Web users still can’t even read T_EX output is another indication that we have a problem:

7. A plug-in or helper application to display .dvi files should be standard with every Web browser.

A look at the mathematical documents created by my students and, even more, a look at the reports they have written on their peers’ work, shows that serious problems remain. The students face the same obstacles listed earlier in putting mathematical expressions on the Web or into their printed papers, compounded by their inexperience. So, let us extend the list of demands:

8. It must be possible for *students* to produce decent mathematical documents, both in print and on the Web.

Universal participation requires that:

9. The system must be platform-independent. (At the very least, it must exist in both UNIX and Windows versions, which communicate flawlessly with each other.)
10. The cost to students (in required software purchases) must be minimal.
11. The time spent in learning software syntax must be minimized.

These last two problems will be alleviated if the students use the software throughout their undergraduate careers, not just in one course taught by an eccentric professor. Unfortunately, “it is a characteristic of those in the [college teaching] professions to resist change unless it is change that they tried to start.”¹

How can we minimize software learning time? Does it mean giving up T_EX? On the contrary, I return to the notion of encapsulated T_EX. I believe that most of T_EX’s reputation for being hard to learn and to use is related to high-level document formatting. Most people first try out T_EX on a small document, such as a job résumé or a letter, that would be trivial to type as a pure ASCII file where the typist has direct control over the placement of every character: the beginner soon discovers that

heroic measures are necessary to override T_EX’s default behavior of reformatting paragraphs and discarding all the redundant white space. But all that is really necessary for most purposes is the math mode of T_EX.

12. We must create an environment that makes it worthwhile for every student to learn basic T_EX math mode syntax. (Learning to create complete T_EX documents, however, may be left as the student’s option.) Instructional material for T_EX beginners should be reorganized to reflect this priority.

T_EX math syntax is a very efficient and logical way of typing in mathematical expressions, far superior to raw SGML-type languages. Hunt-and-peck menu systems are arguably easier to learn, but in the long run excruciatingly slow to use. The T_EX community simply hasn’t succeeded in getting this message out to the rest of the world.

In those fields where T_EX has become the standard method of producing serious documents, its basic math syntax (“pidgin-T_EX”) has also become a rough-and-ready way of incorporating math into e-mail and other informal pure-ASCII communications. This is another major reason why all students should learn this part of T_EX. My student reviewers are supposed to provide constructive criticism of their peers’ papers and to describe in their top-level reports the most serious and systematic mathematical errors they observed. Very often, what they write is something like “You made a mistake in part (c),” leaving the author and me wondering what the mistake is. A big part of the problem is the difficulty of discussing mathematics with any specificity in an ASCII text message. More generally,

13. All students (from the freshman year on) must be enabled to discuss mathematics easily with their teachers and with fellow students, in e-mail, list servers, chat rooms, etc. Pidgin-T_EX should be promoted as a medium for this.

Of course, a more sophisticated system that made it possible for the students to annotate one another’s documents with electronic sticky-notes would be even better.

Conclusions

From my standpoint as a university teacher, the overwhelming issue right now is how to present decently typeset mathematics on the Web. Presumably, in the near future this means: (1) tools for turning T_EX input into MathML output, and (2)

¹ The original quotation (Childs, et al., 1995, p. 303) refers to the *programming* professions, but one of the authors has assured me that the theorem has broader validity.

assuring universal availability (on college campuses, at least) of browsers that can handle MathML.

Students should be able to create decent mathematical documents of their own and also to discuss mathematics in e-mail, etc. T_EX math mode provides a suitable language for the latter if we promote it adequately.

The academic scientific community laid the golden eggs for the exploding computer and software industries. In that light, the lingering difficulty of including mathematical notation in Web presentations and in electronic mail amounts to a scandal. It is up to the T_EX Users Group to:

- solicit its membership's contributions, at all technical levels, toward solving this problem
- exert pressure on commercial developers (and on agencies such as the W3 Consortium) to steer developments in directions that meet these needs²
- continually keep its membership and the rest of the academic community informed of what options are or will be available
- keep T_EX in the mainstream of electronic communications, so that the rest of the world does not come to regard it as a relic of the '80s

Some links in lieu of references

Mathematics in the Foundation Coalition (FC). A detailed report on the FC freshman calculus program at Texas A&M University is at

<http://www.math.tamu.edu/~fulling/fc/>

A summary of its goals and practices appears as

<http://www.math.tamu.edu/~barrow/aseepaper.html>

The Web pages for students to read in place of textbook sections can be reached directly as

<http://www.math.tamu.edu/~fulling/coalweb/xxx.htm>

where *xxx* = *vectors*, *cenmon*, *diffs*, or *taylor*, for instance; all are accessible from the course syllabus pages within the report.

Homework review. The home pages for my sections of Mathematics 311 are at

<http://calclab.math.tamu.edu/~fulling/m311/>

Instructions for the students are

² I reiterate that I have avoided positive remarks on various products because they would inevitably have been incomplete and unfair.

<http://calclab.math.tamu.edu/~fulling/m311/s99/handout.dvi>

for class procedures, and

<http://calclab.math.tamu.edu/~fulling/m311/s99/instruct.txt>

for operating Sadowski's homework review generator. Samples of the resulting student reports and solutions can be seen at

<http://calclab.math.tamu.edu/~fulling/hwk/311s99/sol2.html>

and similar URLs (all accessible from the course home pages). The rationale for the peer review system is described in

<http://calclab.math.tamu.edu/~fulling/iawpaper.ps>

(written at an early stage when all the communication was still on paper). These ideas were heavily influenced by the collections in Connolly and Vilardi (1989) and Sterrett (1990).

General issues. Primarily to introduce students to T_EX and to the problems of mathematical electronic communication, I have set up three sets of links:

<http://calclab.math.tamu.edu/~fulling/xxx.html>

where *xxx* = *webmath*, *webtex*, and *viewers*.

webmath lists on-line references on the problem of putting mathematical notation into Web pages and e-mail, with links to proposed solutions, including MINSE, TTH, *Techexplorer*, and *latex2html*.

webtex provides information about T_EX, including its derivatives with (allegedly) simpler input interfaces, such as *StarTex* and *Scientific Word*. It has links to CTAN and TUG's home page.

viewers deals with viewers for .dvi files, which people who do not aspire to write T_EX themselves nevertheless need to read T_EX documents placed on-line in the most compact and convenient format.

References

- Adams, S. *The Dilbert Principle*. Harper-Collins, New York, 1996.
- Childs, B., D. Dunn, and W. Lively. Teaching CS/1 courses in a literate manner. *TUGboat*, 16(3), 300–309 (1995).
- Connolly, P. and T. Vilardi, eds. *Writing to Learn Mathematics and Science*. Teachers College Press, New York, 1989.
- Murphy, T. PostScript, **QuickDraw**, T_EX. *TUGboat*, 12(1), 64–65 (1991).
- Sterrett, A., ed. *Using Writing to Teach Mathematics* (MAA Notes No. 16). Mathematical Association of America, Washington, 1990.