TUGboat

Volume 21, Number 2 / June 2000

General Delivery

Font Forum

Software & Tools

Hints & Tricks

LaTeX

News & Announcements

Late-Breaking News

Cartoon

TUG Business

Advertisements

Supplement

TeX Users Group

Memberships and Subscriptions

TUGboat (ISSN 0896-3207) is published quarterly by the TeX Users Group, 1466 NW Naito Parkway, Suite 3141, Portland, OR 97209-2820, U.S.A.

2000 dues for individual members are as follows:

- Ordinary members: $75.
- Students: $45.

Membership in the TeX Users Group is for the calendar year, and includes all issues of *TUGboat* for the year in which membership begins or is renewed. Individual membership is open only to named individuals, and carries with it such rights and responsibilities as voting in TUG elections. A membership form is provided on page 150.

TUGboat subscriptions are available to organizations and others wishing to receive *TUGboat* in a name other than that of an individual. Subscription rates: $85 a year, including air mail delivery.

Periodical-class postage paid at Portland, OR, and additional mailing offices. Postmaster: Send address changes to *TUGboat*, TeX Users Group, 1466 NW Naito Parkway, Suite 3141, Portland, OR 97209-2820, U.S.A.

Institutional Membership

Institutional Membership is a means of showing continuing interest in and support for both TeX and the TeX Users Group. For further information, contact the TUG office (`office@tug.org`).

Printed in U.S.A.

Board of Directors

Addresses

General correspondence,
payments, etc.
TeX Users Group
P. O. Box 2311
Portland, OR 97208-2311
U.S.A.

Delivery services,
parcels, visitors
TeX Users Group
1466 NW Naito Parkway
Suite 3141
Portland, OR 97209-2820
U.S.A.

Telephone

+1 503 223-9994

Fax

+1 503 223-3960

Electronic Mail

(Internet)

General correspondence,
membership, subscriptions:
`office@tug.org`

Submissions to *TUGboat*,
letters to the Editor:
`TUGboat@tug.org`

Technical support for
TeX users:
`support@tug.org`

To contact the
Board of Directors:
`board@tug.org`

World Wide Web

`http://www.tug.org/`
`http://www.tug.org/TUGboat/`

Problems not resolved?

The TUG Board wants to hear from you:
Please email to `board@tug.org`

Sir Benjamin Backbite: ... I think you will like them, when you shall see them on a beautiful quarto page, where a neat rivulet of text shall meander through a meadow of margin.

Richard Brinsley Sheridan
The School for Scandal (1777)

TUGBOAT

COMMUNICATIONS OF THE TEX USERS GROUP

EDITOR BARBARA BEETON

VOLUME 21, NUMBER 2 • JUNE 2000
PORTLAND • OREGON • U.S.A.

TeX Users Group

Memberships and Subscriptions

TUGboat (ISSN 0896-3207) is published quarterly by the TeX Users Group, 1466 NW Naito Parkway, Suite 3141, Portland, OR 97209-2820, U.S.A.

2000 dues for individual members are as follows:

- Ordinary members: $75.
- Students: $45.

Membership in the TeX Users Group is for the calendar year, and includes all issues of *TUGboat* for the year in which membership begins or is renewed. Individual membership is open only to named individuals, and carries with it such rights and responsibilities as voting in TUG elections. A membership form is provided on page 150.

TUGboat subscriptions are available to organizations and others wishing to receive *TUGboat* in a name other than that of an individual. Subscription rates: $85 a year, including air mail delivery.

Periodical-class postage paid at Portland, OR, and additional mailing offices. Postmaster: Send address changes to *TUGboat*, TeX Users Group, 1466 NW Naito Parkway, Suite 3141, Portland, OR 97209-2820, U.S.A.

Institutional Membership

Institutional Membership is a means of showing continuing interest in and support for both TeX and the TeX Users Group. For further information, contact the TUG office (office@tug.org).

Printed in U.S.A.

Addresses

General correspondence,
payments, etc.
TeX Users Group
P. O. Box 2311
Portland, OR 97208-2311
U.S.A.

Delivery services,
parcels, visitors
TeX Users Group
1466 NW Naito Parkway
Suite 3141
Portland, OR 97209-2820
U.S.A.

Telephone

+1 503 223-9994

Fax

+1 503 223-3960

Electronic Mail

(Internet)

General correspondence,
membership, subscriptions:
office@tug.org

Submissions to *TUGboat*,
letters to the Editor:
TUGboat@tug.org

Technical support for
TeX users:
support@tug.org

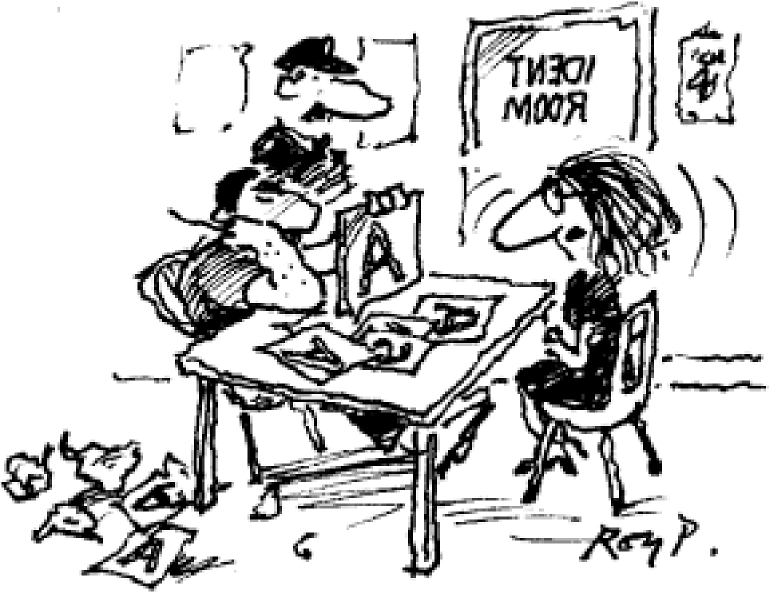
To contact the
Board of Directors:
board@tug.org

World Wide Web

http://www.tug.org/
http://www.tug.org/TUGboat/

Problems not resolved?

The TUG Board wants to hear from you:
Please email to board@tug.org



No, no, no. It had pointy things sticking out at the bottom

General Delivery

From the President

Mimi Jett

Greetings, TUG members!

It is said that anything that becomes routine is, by definition, taken for granted. We see evidence of this throughout our lives; none of us are able to claim that we said an appropriate number of "thank you s" for all of the help and sacrifices people have made on our behalf. Sometimes we are not even aware of the work behind the scenes, to make our life easier or better. As an organization and a community, we owe a debt of thanks to the people who give up time with their families, work, or friends to create the tools and resources we have come to take for granted. For example, the CTAN CDs you are receiving with this issue are the product of literally man-years donated by a number of highly skilled volunteers, including Karl Berry, Robin Fairbairns, Jim Hefferon, Rainer Schöpf, and Reinhard Zierke.

Last quarter you received the TEX Live CD, which is a shining representation of the devotion to quality Sebastian Rahtz and his colleagues put into their work. The website and server administration done for years, again by Karl Berry, and now managed by Kaja Christiansen, are extremely important, yet almost invisible because it "just works". Year after year we see upgrades and new tools to make life a little easier. The publication of this journal is another project that reaches deep into the discretionary time of Barbara Beeton, Mimi Burbank, Christina Thiele, with help for many years from Sebastian, Robin and many others. Please join me in thanking these people, out loud, for their contributions. And please, let us not take any of this for granted.

Speaking of contributions, I want to thank those authors who are considering a new article for *TUGboat*. We are happy to see so much interest in TEX as the move to MathML becomes a stampede. MathML is probably the most important thing to happen to mathematics communication since TEX. With the open standards becoming real, and practically every mathematics software developer writing to MathML, we finally see the interoperability between applications and languages that seemed like a dream not long ago. The overlap between the TEX and MathML communities is quite large, as you would expect. We look forward to many future articles about MathML, XML, XSL, and TEX as the standards evolve.

Another way you might consider joining our cause is to run for a seat on the Board of Directors. Next year is an election year, with five director positions and the presidency up for grabs. Over our 20-year history we have had dozens of directors who contributed to the organization we have today. Several of them keep coming back for yet another term, myself included. I will be the first to admit that working on TUG is *fun*! It is exciting to meet people from all over the world, to work on solving problems and creating something of value together. The sense of accomplishment is nice, but the relationships with this eclectic group of people are what make it fantastic. Please consider joining us on the board, or running for president. We would love to have you.

The election in 2001 will be held electronically, for those members that have access to the Web. To ensure that we reach all our members, we will also send a paper ballot. Please feel free to use either method to cast your vote, but not both. We expect to announce the opening of nominations by December, with voting to begin in February. By April we should have results to announce.

Next year's annual meeting will be held at the University of Delaware, August 11–17, 2001. The call for papers has been issued; please check `tug.org` for current information. The annual meeting for 2002 has two choices of venue: Cork, Ireland, or Kerala, India. We have never had an annual meeting in Asia, although we have a significant number of members on the continent. Due to the diverse opinions about the choice of venue, we are going to survey the members for your opinions. Please watch for a notice from Sue DeMeritt, and let us know what you think.

Thank you for your support of TEX and TUG. And don't forget to thank those who make your life a little better.

◇ Mimi Jett
IBM
T. J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
`jett@us.ibm.com`

Editorial Comments

Barbara Beeton

XΥMTEX posted to CTAN

The package XΥMTEX described in an article by Shinsaku Fujita and Nobuyu Tanaka in the last issue (*TUGboat* **21** (1), pages 7–14) has been posted to CTAN, where it can be found in the area `macros/latex/contrib/supported/xymtex`.

Protection for font names in Germany

While font designs have been subject to registration with the German Patent Office since 1991, older designs could not be protected. However, a recent court case has provided older fonts name protection through trademark as brand names.

The suit was brought by Linotype Library against several vendors of "clones" of common fonts; these companies held that font names such as "Univers", "Helvetica" and "Palatino" are generic, and thus usable by anyone, but the court decided otherwise. The initial case was decided in June 1999 by the Frankfurt district court, and upheld in April 2000 by an appeals court.

CTAN — CDs and catalogue entries

With this issue of *TUGboat* you will find a 3-CD collection, a snapshot of the CTAN contents. This collection supplements the more structured selection from the TEX Live CD shipped with the last issue. We are sure you will find it useful.

We'd like to remind you that finding an item on CTAN is simplified considerably if a catalogue entry exists. If you are the author of a CTAN item, please check CTAN itself to verify such entries. They are located in `help/Catalogue/entries`, where both an `.html` and an `.xml` file should be present for each. If you find that no entry exists, create one, and send it to `ctan@dante.de`, so that it can be added. At a minimum, the information should include

- Name of item and version number
- A one-line description, to be used as a title
- Name of author
- License type, e.g., LPPL, GNU, public domain; or indicate any restrictions, e.g., shareware, free for private use only.
- An abstract describing the main features, names of packages included in the distribution, key words — anything that a potential user might specify in a keyword search
- Location in the CTAN tree
- Location of the canonical source if posted elsewhere on the Web

If you are updating or submitting a new item, don't forget this very important step that will make it easier for potential users to find your contribution.

And if you are a user who discovers that a package contains more than is listed in the catalogue, you might draft an update for the abstract. Just extend the existing ⟨*package*⟩`.xml` file and submit it; you don't even have to know XML to do that. With a little help from thoughtful users, we can all benefit.

TUG Web site moves to Denmark

The host for the TUG Web site has left Boston, where it was managed since its creation by Karl Berry. It now resides in Aarhus, Denmark, where it is in the able hands of Kaja Christiansen at the Aarhus University Department of Computer Science. Kaja was asked by the Board if she could take on this responsibility, as Karl has moved away from Boston; with the support of her department, she accepted.

In addition the the TUG pages, this machine is host to a number of other TEX resources, among them:

- TEX Live information and updates
- The teTEX home page
- The `tex-k` discussion list
- Archives of discussions on `tex-k`, `tex-eplain`, `pdftex`, `fptex`, `twg-tds`, and several other mailing lists
- "Historical" versions of TEX and friends
- *TUGboat* tables of contents and a growing collection of articles

Thanks to Karl Berry for establishing and managing the TUG site and to the University of Massachusetts, Boston, for hosting it up until now. And thanks to Kaja Christiansen and Aarhus University for providing a new home.

Hermann Zapf honored by DANTE

At DANTE 2000, held March 8–11 this year in Clausthal-Zellerfeld, the German TEX users group presented Hermann Zapf with an award for his life work, his service to typography, and in particular, for his contributions to TEX and METAFONT.

Prof. Zapf spoke about his work with Knuth on symbol alphabets and the book *3.16*.

Both the *Laudatio* (presented by Frank Mittelbach) and Prof. Zapf's talk have been published in *Die TEXnische Komödie*. We will try to arrange for a translation to appear in a future issue of *TUGboat*.

GUTenberg publications on the Web

Jacques André has announced that all the publications of the Association GUTenberg are now available on the Web. This includes:

- *La Lettre GUTenberg*, from number 0 (February 1993) to the present (17, April 2000)
- *Cahiers GUTenberg*, from number 0 (April 1988) to the present (35–36, May 2000)

All items are in PDF format, with some papers from the *Cahiers* also in PostScript, when there are no problems of copyrights with fonts. Most papers are in French; however, some are in English, notably papers presented at EuroTeX conferences (*Cahier* 10–11, EuroTeX '91, Paris, and *Cahier* 28–29, EuroTeX '98, Saint-Malo).

These Web pages can be accessed at `http://www.gutenberg.eu.org`, through the link to Publications.

La Lettre almost always uses a different font for each issue, accompanied by a brief introduction to the "font du jour", so this is an excellent collection to examine when looking for new fonts to use for your own TeX documents.

The Romans didn't know about zero

This little item was sent to us by Nelson Beebe. There *is* a certain logic to it; let's just think of it as a feature . . .

```
% tex
This is TeX, Version 3.1415 (C version 6.1)
**\relax

*\romannumeral{0}
! Missing number, treated as zero.
```

Incunabula on-line at the Bavarian State Library

The Bayerische Staatsbibliothek has posted a large collection of page images from incunabula — printed works from before 1500. These can be found at `http://mdz.bib-bvb.de/digbib/inkunabeln/`.

This collection concentrates on book illustrations, but text is shown too, where the pictures do not occupy entire pages. There is a wide variety of subject matter, from biblical and historical tales to illustrations of contemporary trades, scholarly pursuits and everyday life. Most of the text is in German, some in Latin.

◇ Barbara Beeton
American Mathematical Society
P. O. Box 6248
Providence, RI 02940 USA
`bnb@ams.org`

Interview: Donald E. Knuth

Advogato

This week [25 January 2000], Advogato had the pleasure and honor of interviewing Prof. Donald E. Knuth. He is the author of the TeX typesetting system as well as *The Art of Computer Programming* and a number of deep, insightful papers and books. The interview took place by phone on a rainy California winter day. The topics covered the freeness of TeX and its fonts, how TeX's innovations have slowly diffused into commercial systems, some history of math typesetting, the design of TeX from the beginning as an archival system, literate programming in the age of the Web, MMIX and the Transmeta chip, how to avoid generating inscrutable error messages, and taking the TeX ideas to a broader community. Read below to find more about a remarkable person.

Advogato: The first questions that I have are about free software. TeX was one of the first big projects that was released as free software and had a major impact. These days, of course, it's a big deal. But I think when TeX came out it was just something you did, right?

Prof. Knuth: I saw that the whole business of typesetting was being held back by proprietary interests, and I didn't need any claim to fame. I had already been successful with my books and so I didn't have to stake it all on anything. So it didn't matter to me whether or not whether I got anything financial out of it.

Adv: I see.

DEK: There were people who saw that there was a need for such software, but each one thought that they were going to lock everyone into their system. And pretty much there would be no progress. They wouldn't explain to people what they were doing. They would have people using their thing; they couldn't switch to another, and they couldn't get another person to do the typesetting for them. The fonts would be only available for one, and so on.

But I was thinking about FORTRAN actually, the situation in programming in the '50s, when IBM didn't make FORTRAN an IBM-only thing. So it became a lingua franca. It was implemented on all different machines. And I figured this was such a new subject that whatever I came up with probably wouldn't be the best possible solution. It would be more like FORTRAN, which was the first fairly

This interview was conducted by Raph Levien for the Advogato Website, `http://www.advogato.org/person/advogato/`, and is published here with permission.

good solution [chuckle]. But it would be better if it was available to everybody than if there were all kinds of things that people were keeping only on one machine.

So that was part of the thinking. But partly that if I hadn't already been successful with my books, and this was my big thing, I probably would not have said, "well, let's give it away." But since I was doing it really for the love of it and I didn't have a stake in it where I needed it, I was much more concerned with the idea that it should be usable by everybody. It's partly also that I come out of traditional mathematics where we prove things, but we don't charge people for using what we prove.

So this idea of getting paid for something over and over again, well, in books that seems to happen. You write a book and then the more copies you sell the more you get, even though you only have to write the book once. And software was a little bit like that.

Adv: I think that's the model that software publishing generally comes from. There was a quote that you had in the "Mathematical Typography" essay reprinted in *Digital Typography* where you said, "Mathematics belongs to God."

DEK: Yes. When you have something expressed mathematically, I don't see how you can claim... In the context, that was about fonts. That was when I had defined the shape of the letter in terms of numbers. And once I've done that, I don't know how you're going to keep those numbers a secret...

Adv: Proprietary.

DEK: I can conceieve of a number that would be a million digits long and would be extremely expensive to compute, and once somebody knew that number, it would solve all kinds of problems. And I suppose that would make it a little bit harder to say that God already had given us this number, when it's a number that you can only discover by a tremendous amount of sweat.

When I made that quote, I didn't think of such things.

Adv: Fonts seem like a really interesting edge case for that argument, because a font is in some ways a mathematical formula, especially a TEX font, much more so than what came before, but it's also an artwork.

DEK: Absolutely. It absolutely requires great artistry. So the other part of this is that artists are traditionally not paid like scientists. Scientists are supported by the National Science Foundation to discover science, which benefits the human race. Artists, or font designers, are not supported by the National Font Foundation to develop fonts that are going to be beneficial to the human race. Fonts are beneficial to the human race, they just don't traditionally get supported that way. I don't know why. They're both important aspects of our life. It's just that one part has traditionally gotten funded by a royalty type mechanism and the other by public welfare grants for the whole country.

Adv: Perhaps that has something to do with the absolute necessity in science to have open access to the results of others, that if you did science in a closed, proprietary framework, the disadvantages would be so clear.

DEK: With fonts, it was pretty clear to me.

Adv: Ok! That's a question that Federico Mena Quintero suggested. You've gotten a number of free fonts contributed by artists, in some cases very beautiful fonts, to TEX and to the Metafont project. In general, this has been a real struggle for open source development these days, to get free fonts. Do you have any thoughts?

DEK: I think it's still part of this idea of how are the font designers going to get compensated for what they do. If they were like a scientist, then they've got their salary for doing their science. But as font designers, where do they get their salary? And musicians. It's just a matter of tradition as to how these people are getting paid.

Adv: But how did you address those problems with the fonts that got contributed to TEX?

DEK: In my case, I hired research associates and they put their fonts out into the open. Or else, other people learned it and they did it for the love of it. Some of the excellent fonts came about because they were for Armenian and Ethiopian and so on, where there wasn't that much money. It was either them taking time and making the fonts or else their favorite language would be forever backwards, so I made tools by which they could do this. But in every case, the people who did it weren't relying on this for their income.

If we had somebody who would commission fonts and pay the font designer, the font designer wouldn't be upset at all about having it open, as long as the font designer gets some support.

Adv: And you did some of that.

DEK: Yeah. In fact, I worked with some of the absolute best type designers, and they were thrilled by the idea that they could tell what they knew to students and have it published and everything. They weren't interested in closed stuff. They're interested in controlling the quality, that somebody isn't going to spoil it, but we could assure them of that.

Adv: Right. Working with the creator of the software.

DEK: Yeah, if they didn't like the software, I could fix it for them.

Adv: One of the things that struck me when I was reading *Digital Typography* is the intensive study that you did, especially in the area of math typesetting. When I was writing papers, using math formulas in TeX, I just typed in the commands and out came the math and it looked pretty good to me. It shouldn't have been surprising, but it definitely struck me how much attention you paid to the best mathematics typesetting of past centuries.

DEK: I do strongly think that people, when they start throwing computers at something, they think that it's a whole new ballgame, so why should they study the past. I think that is a terrible mistake. But also, I love to read historical source materials, so I couldn't resist. I had a good excuse to study these things, and the more I looked at it, the more interesting it was. But I don't think responsible computer scientists should be unaware of hundreds of years of history that went before us. So that was just a natural thing to approach it that way, for me.

Adv: I noticed, for example, that in the proprietary software market for publishing, that systems are only today acquiring features that have existed in TeX for a long time, for example whole-paragraph optimization. There's a big to-do about Adobe InDesign, which finally...

DEK: They finally implemented the TeX algorithm.

Adv: Did they implement the TeX algorithm?

DEK: Yeah, that's what they said.

Adv: Did you talk to the people?

DEK: I met three of four of them at the ATYPI meeting in Boston in October, but that was after I had heard about it, that some friends had found this in the documentation.

Adv: Another similar issue is TrueType fonts. TrueType fonts have this property of including instructions, computer programs effectively, in the font, to do hinting.

DEK: Well, I never met Elias or whatever.

Adv: Sampo Kaasila?

DEK: I don't know. I know enough about TrueType to know that it's a very intelligent design, that it is similar to Metafont except that it strips out everything that's slow. So the way the hinting is done is by program, certainly. Of course, it came out maybe ten years after Metafont, so probably something got through somehow.

There was the F3 font that Folio was making, if I can remember the name, what the people in industry called it. Some of the people that I had worked with on Metafont went into making font designs that were similar to TrueType, but have not been successful.

Adv: There's a fairly major controversy with TrueType right now, that there a number of patents that are owned now by Apple. It's kind of interesting to me that that is the case even though it's for the most part derivative work of what was in Metafont.

DEK: I've been very unhappy with the way patents are handled. But the more I look at it, the more I decide that it's a waste of time. I mean, my life is too short to fight with that, so I've just been staying away. But I know that the ideas for rendering... The main thing is that TrueType uses only quadratic splines, and that Type1 fonts use cubic splines, which allow you to get by with a lot fewer points where you have to specify things.

The quadratic has the great advantage that there's a real cheap way to render them. You can make hardware to draw a quadratic spline lickety-split. It's all Greek mathematics, the conic sections. You can describe a quadratic spline by a quadratic equation (x, y) so that the value of $f(x, y)$ is positive on one side of the curve and negative on the other side. And then you can just follow along pixel by pixel, and when x changes by one and y changes by one, you can see which way to move to draw the curve in the optimal way. And the mathematics is really simple for a quadratic. The corresponding thing for a cubic is six times as complicated, and it has extra very strange effects in it because cubic curves can have cusps in them that are hidden. They can have places where the function will be plus on both sides of the cubic, instead of plus on one side and minus on the other.

The algorithm that's like the quadratic one, but for cubics, turns out that you can be in something that looks like a very innocuous curve, but mathematically you're passing a singular point. That's sort of like a dividing by zero even though it doesn't look like there's any reason to do so. The bottom line is that the quadratic curves that TrueType uses allow extremely fast hardware implementations, in parallel.

Adv: The question is whether that matters of course, now that CPU's are a zillion times faster.

DEK: But for rendering, Metafont was very, very slow by comparison, although I'm amazed at how fast it goes now. Still, it has to be an order of magnitude better, and certainly that was a factor in getting TrueType adopted at the time that it was, because machines weren't that fast then. So TrueType was an intelligently chosen subset, but certainly all the ideas I've ever heard of about TrueType were, I believe, well known in the early '60s.

Adv: Back to this issue of preserving the past. I was reading some papers of Edsger Dijkstra. For a while, he used handwritten manuscripts and then a typewriter to actually distribute the work. And, his notation became much more typewriter-like, that he would use an underlined A or a boldfaced A instead of the traditional `\forall` symbol.

DEK: I've gotten some of his handwritten notes, but I don't remember the typewritten ones.

Adv: I was looking at the proceedings of the Marktoberdorf summer school in '90, where there were a couple of papers by him and his group. In any case, it occurred to me that TeX has made the traditional mathematical notations so accessible to practicing computer scientists, students, researchers, etc. It's very likely that if there hadn't been something like TeX, in other words if mathematical typesetting had remained strictly in the domain of book publishers, and people who did publishing as their profession, it's likely that the standard notations in computer science would have become much more typewriter-like, kind of ASCII-ized.

DEK: That's interesting.

Adv: As it is, if you look at *Principles of Programming Languages*, there's all this type theory in there with beautiful Greek letters, inference rules and so on. The visual aesthetic seems to have been inspired by the easy availability that TeX provides.

DEK: Certainly, TeX was partly influenced by the needs of computer science, that pushed beyond what mathematicians had needed. Computer scientists needed to see mathematical structure, but they also needed to see the kind of structure you have in programs. So we had to push the envelope in the ACM Journal in the '60s. To publish computer science papers, the typesetters using hand methods in the '60s were being forced to work harder by the computer scientists than by the mathematicians at that time. It's part of the need for a way to see the structure that's in the stuff you're working with. Since I'm writing books about computers, naturally I made TeX so that it could do that.

Adv: It's clear that TeX was inspired by the visual aesthetic of mathematics.

DEK: Oh yeah.

Adv: I'm just saying: to what extent mathematics and computer science today is being influenced by the visual aesthetic of TeX?

DEK: Well, I don't know. I think the fact that TeX was open-ended means that people can choose their notation themselves. They don't have to adopt somebody else's macros particularly. That was the bind that we were in before. We would have to first write our paper, then explain it to the printer, and the printer would maybe get it right. But now we're less hesitant to use a notation that's not traditional, but that we think is appropriate, because we know that we don't have to go through a noisy channel in the middle.

Adv: One of the other accomplishments of TeX that I continue to be impressed with is the consistent rendering, the idea that you have a source file and that on virtually any implementation of TeX, you'll get the same results.

DEK: That's what I insisted on the most. I didn't want to get paid, but I didn't want it to change.

Adv: If you look at, for example, a lot of the new systems that are being designed today, like Cascading Style Sheets for the Web, they allow a tremendous amount of latitude for the implementor, so you don't get anything approaching consistent rendering.

DEK: With the Web, it's a different tradeoff, between appearance and the ability for a search engine to find it.

Adv: Presentation vs. semantic markup, right, in general?

DEK: Right.

Adv: But it seems to me that TeX's model is actually a way to be better in both of those regards than the way that the Web has been evolving, because you do have at least the potential for semantic markup, and then you get the fact that you know what the presentation will be. So, one example from the *Digital Typography* book is that you described changing the wording of some of the exercises so that the symbols would line up better, that the parentheses wouldn't collide with descenders.

DEK: This wouldn't go too well to the Web, because people can certainly set the width of their browser window differently, and I'm not expecting that I would optimize for all of those things. But if I'm optimizing for a particular page size, then I would say, "why not." Mostly, the rewriting is so that I can get a figure to be on the same page where you want to see it, without turning the pages over, if you care about things like that. There were some parts of *The Art of Computer Programming* where I said, well, I gotta think of five lines of things to say on this page, and then I would think of ten lines. It was a motivation for research. I knew I needed five more lines of stuff to say on this page, because otherwise the illustrations aren't going to come out right.

Adv: If you are going to print, and of course a lot of people print from the Web, and usually on page sizes that are well defined in advance, with TEX you have at least the possibility of doing that. If you do reformat the width, then the semantic markup is still there so maybe it isn't as optimized, but you still get a reasonable rendering.

DEK: Well, what I was mostly concerned about was that the rendering would be the same in ten years. What you say is probably important, but I didn't have that so much in mind as the fact that on everybody's machine it would come out the same, not only now but in the future. Because there was so much software rot going on where stuff wouldn't work any more. And I didn't have time to keep updating... It's going to take many years to finish *TAoCP*, so I wanted to make sure that I could make Vol. 7 match Vol. 1.

Adv: But that seemed to be at odds with something else you said in *Digital Typography* about being surprised that people didn't make more custom versions of TEX for specialized applications. And if that had happened, and if some of those had become popular, wouldn't that have created similar problems?

DEK: I wasn't expecting them to become popular. I was expecting that they were going to be one-off things. If I were a publisher, that's what I would have done.

Adv: But if you create a one-off, that still creates the potential for software rot, that ten years from now...

DEK: Okay. I have to save my program. Yeah.

Adv: And that's hard. Usually.

DEK: The program is there in a bunch of change files, so as long as the C language isn't changing, then I'm ok.

Adv: Well, it isn't changing much.

DEK: See, the way I would make these custom versions is by using the change file facility that we built in when we started distributing TEX. It means that you have a master file, and then you have another file that says, these are the things I changed. And that's the way people ported TEX to lots of different operating systems, they would have it in these change files. I could change the master file, and their change file would probably still work.

Adv: I see. I didn't actually know about that aspect of the TEX distribution.

DEK: There's a change-file mechanism that's been quite successfully received in all the TEX distributions.

Adv: That sounds like it could be useful in other projects as well.

DEK: Isn't hasn't been picked up yet, but I don't know why [laugh].

Adv: Speaking of literate programming, the question that I have for you is: now that people are moving everything to the Web, including programming, do you think that's another chance for literate programming to become popular?

DEK: There's a lot of ferment in this direction. I still don't have the dvipdf program that I've got to get installed on my Linux. I guess I've got to try it. But a guy has worked out now that he can convert to Acrobat format, a literate program automatically comes out in Acrobat format, so that you can use all the features of the Acrobat reader to click on stuff to move around in the documentation.

Adv: Cross-referencing and so on?

DEK: Yeah, find a variable where it's declared, and all other uses of the variable. It's certainly a natural thing for hypertext, and this guy in Brazil has worked out a nice system. Still, the thing that's holding it back is probably that some programmers just don't like to document their stuff.

Adv: That's certainly a problem we've had in free software, that if there's a documentation file and then a code file, the two often get out of sync.

DEK: Things are slowly changing, but most of the horror stories that you hear about that stuff is because of the illiterate nature of the program.

Adv: So you think that literate programming is a way to make that better?

DEK: It's so much better than the alternative, but I think Jon Bentley explained it to me best, not much percentage of the world's population is really good at programming, and not much percentage of the world's population is really good at documenting, and here you need to be doing both. [laughs]

So, I think that in my experiments with Stanford students I found that more than half of the students that I worked with really hit it off well with literate programming. Maybe Stanford students aren't average. [laugh]

Adv: Right.

DEK: So you need somebody who's not afraid to write an essay, as well as not afraid to write a computer program.

They work together perfectly, but you have to be able to communicate to the computer, and you have to be able to communicate to the human being, and if you don't do both, then you can't expect your program to be as successful. Literate programming is just the best way I know to do both at the same time.

My new book on MMIX, where another example of what I hope people will consider is just a state-of-the-art, normal way to document programs. In this case, the program that I wrote for this MMIX simulator last year, I wouldn't have been able to finish the program without literate programming. It was just too mind boggling. Not only was literate programming a nice add-on, without that tool, I wouldn't have gotten the program done.

Adv: I saw the MMIX book just came out. I haven't gotten my copy yet, but I'm really looking forward to it. That is the new instruction set that you're using in *TAoCP*, right?

DEK: Yeah. Well, I'm going to use it eventually. It's there, but it's going to be a while before I replace it in volumes 1, 2 and 3. I gotta write volume 4 first.

Adv: MMIX is another digression for you, then.

DEK: It'll be used a small amount in Vol. 4. If I talk about machine code at all in Vol. 4, there are a few places, then it will be in there.

Adv: I was reading up on MMIX at the same time that the Transmeta processor came out, the Crusoe. I don't know if you're familiar with that, but it's a VLIW that does on-the-fly translation of a pretty much arbitrary instruction set into its own native format. Do you think that MMIX is really going to be very similar to the architectures that people will need to know, or is the increasing complexity of CISC chips and new things like VLIW going to make. . .

DEK: I don't know. These VLIW chips like the IA64 are getting very mixed reviews. I guess the Pentium translates into a RISC-like language internally.

Adv: Yeah, sort of.

DEK: I think in any case, this is a model that is going to be as close to efficient and clean at the same time as we're going to get in the foreseeable future. I wanted it to be realistic, but I also wanted it to be clean and easy to learn.

Adv: One of the things that intrigued me about this Crusoe chip that Transmeta is doing is that they have firmware that does dynamic translation of the source architecture. Obviously they're pushing x86 as the most commercially important source language, but I was just wondering if maybe MMIX would be a good candidate for. . .

DEK: If they want to play with it. But that will be just an academic exercise right now rather than a commercial one, because there isn't that much code written for it.

Adv: Well, the x86 has some serious problems, that the memory bandwidth is a lot higher than it needs to be because it doesn't have enough registers.

DEK: Oh, absolutely.

Adv: So, for this particular chip, it might make more sense for gcc to generate MMIX code.

DEK: Oh, I see what you mean. Well, that's interesting.

Adv: I don't know. I was just wondering if you had some thoughts on that.

DEK: Well, okay. So we need a library, all that kind of code. It's for the MMIXmasters to develop, because I haven't got time to mix with it.

Adv: The other thread I was interested about was this whole development of commercial TEX packages in the '80s, for PC's and so on. Those didn't catch on, and PostScript did, and the world is quite different because of that. Do you have have some stories about why that would have happened, or what you might have done differently?

DEK: Those programs were kind of heroic. At that time, PC's were limited. If you wanted to have an array with more than than 65kbytes in it, you had to go in to a special extended addressing mode. So it wasn't easy to port TEX to a PC in those days, with the 186 and the zero-86.

Except in the mathematics and physics community, there weren't that many people who were interested in the archival quality of stuff, or making something that looks best instead of good. I never expected TEX to be the universal thing that people would turn to for the quick-and-dirty stuff. I always thought of it as something that you turned to if you cared enough to send the very best.

It made me almost scared when I heard that college students in Japan were being forced to learn TEX, because I never wanted anybody to be compelled to dot their i's and cross their t's. I always thought of it for somebody who wanted something to pride in putting a little spit and polish on documenting. I didn't think of it as the world standard or something like that at all.

Adv: The things that, in my experience, are the most difficult for people who are learning TEX or using TEX, are for one thing, the difficulty of using PostScript fonts. And from what I understand, that's accidental.

DEK: Oh, there's no problem whatsoever, except since on every machine you needed to interface a different way, and everybody had a different subset of the fonts. There were six different PostScript Times Roman fonts, and they all had slightly different spacing, so there was no way to make that easy. Then Adobe was going to put out another Times Roman, because they could improve it. There was no idea of archiving these fonts or keeping them in a way that wasn't going to change from year to year the way other things in TEX were doing, so that was one of the hangups. Also, we didn't have that many drivers until about 1990 for PostScript. I had been using them for different projects, but since I wanted to make sure that I was technology independent for my own books, I did everything in bitmaps that I generated in a form that I knew would work 50 years from now. But when I would do a special project that was not for my books, but for something where I was supposed to make camera ready copy for some journal, then I would use commercial fonts. We finally made the virtual font mechanism popular, and people started understanding it and it took another ten years. Now, that's being used a lot.

Adv: It's better, but it's still difficult to just take a PostScript font and drop it into TEX and have TEX use it. That might be more of a packaging issue.

DEK: Yeah, You've got to change a couple of lines in some mysterious files that are not well documented.

Adv: And I've had problems with encodings too.

DEK: You have to be sure that you get the right font metrics.

Adv: It just seems like a hassle when I've tried to do it.

DEK: Well, it's a hassle to install new fonts right now, because the TEX world and the Microsoft world are separate.

Adv: Yeah.

DEK: So you've got to install the fonts in a few different places.

Adv: The other issue that is definitely intimidating for beginning users are the error messages and error reporting, especially when you're using a macro package like LATEX. It seems like the error messages that you get have virtually no relationship whatsoever to what you did wrong. I'm wondering, is that because of the macro nature of that process, an inherent limitation of systems that do what TEX does by evaluating huge piles of macros?

DEK: There is something inherent. If you have a language that has a certain global character to it so that if you make a mistake on line 100, there's no way for the computer to find out about it until line 200, because even though you made a mistake, it still made sense. Then you get into the situation that you're describing, where error messages are pretty inscrutable.

So if I had put more care into making a language that was more restrictive, that it would check a lot more stuff, then you'd catch an error much quicker.

Adv: Right.

DEK: Let me explain. TEX does have some things like this. It'll say that a macro can be defined to be `\outer`, which means that you're not supposed to have this macro in the middle of a parameter to another macro. So if somebody forgot a right brace, then it'll catch it before you've gone too far in the program. But suppose that I had made something that you have to end something on the same line it begins. For example, take a programming language. If you have a string starting with a double quote mark, you've got to finish the string on that same line. So if you forget a double quote mark, you know about it right away. But if you didn't have that rule, then you could leave out one double quote mark, and everything that was outside of a string would be treated as inside of a string, and vice versa.

Adv: And you generally get an error message the next time a quote mark appears in the program.

DEK: Yeah, you're bound to get an inscrutable error message. So, there's part of it inherent that TEX is a language that allows you to go on and on and still be syntactically correct.

Adv: And the other problem that I see with macros is that it makes it much more difficult to do an incremental update to a display. Of course, that's something that seems very interesting now, with big high-res displays commonly available, and probably less important when TEX was originally being created. If you make even a small change to a TEX document, you pretty much are forced to go through...

DEK: TEX doesn't know that your small change hasn't changed everything.

Adv: Right. So you can't really have a version of TEX that lets you edit in one window and have those...

DEK: But on this Pentium III, TEX just generates 50 pages by the time you hit carriage return.

Adv: Yeah, but there's still a difference between that and incremental editing where you're actually typing and seeing your changes updated in real time.

DEK: It's so close to real time now, even if I'm going through dvips and Ghostview, and going up to page 20. Right now, my Linux machine at school is ten times faster than my SparcStation at home, so it's worthwhile for me to go to school, just to get this effect, when I'm doing a lot of editing of spacing. So I'm sitting here with a 50 page document, and I say, "maybe I'll add a thin space here," and I'll re-TEX the entire document, and I'll re-convert to PostScript, and it's just a fraction of a second. And I can see right away that the thin space has gone in there. I can't believe it. And this is going to be so much more common in the future.

It's hard for me to believe because I know what TEX is doing, I know what dvips is doing, and yet it's fast, it's done.

Adv: Well, you've heard of Gates' law, which is that the speed of software halves every 18 months, right? [laugh] And I guess that TEX is an exception to that.

DEK: Well, it hasn't changed.

Adv: Exactly. Well, one of the things that I'm personally interested in is making the kind of stuff that's in TEX more accessible, to users in general, so I'm working on interactive editing in the Gnome project, which is a Linux thing. I'm very interested in doing things like the whole paragraph optimization and the nice hyphenation with all the different languages. Also, in the area of graphics, I've been doing work with splines and using some of the work that John Hobby did for the smooth fitting of splines to just the control points, so you don't have to fiddle with the Bezier control points. I've got some prototype software around that does that interactively. I'm very excited by it, because I feel like somebody who is not a mathematician but who is a font designer might be able to sit down with that, and really get the benefit of a lot of the ideas that were in Metafont, but not have to deal with it as mathematics. You just deal with it directly on the screen. Now, to you, you're a mathematician.

DEK: No, no. That's right. It never became natural. I tried to explain how, but even the idea of multiple masters was a pretty hard sell.

Adv: Yeah. But in any case, I'm very grateful to have the enormous body of work that's been done by you and your graduate students on TEX, Metafont, and related packages. My work is so much easier. All I have to do is adapt this stuff and steal all your best ideas.

DEK: Well, it's not... The hard stuff is still getting it so that it responds right to the people who need it. For that, you have to really be in that community. Also, to see how the users...

Adv: That's what I'm trying to do.

DEK: Like this error message business that you brought up. You have the same kind of thing in a C language program, where if you do something wrong, you might get a hundred error messages. But the amazing thing about the literate programming tool, CWEB, is that if I make five CWEB errors, I will tend to get five error messages.

Adv: Interesting.

DEK: Not only the first one will make sense, but I'll be able to look at the whole thing and fix them all and recompile. And it's always been that way. There's something about that design that localizes the errors. I'm not sure what the key is.

Adv: Well, that would be interesting to look into.

DEK: That's one of the tradeoffs that you have between how much redundancy is there so that the computer can check as you're going. If you're going to catch an error right away, that means that people have to write things a little bit redundant. That means that you put in a little bit of burden, but maybe it's still better to have that little bit of burden than to have your errors become inscrutable.

Adv: Aha. Unless you're one of those people that always types things right the first time.

DEK: Oh, no no. I have never met such a person.

Adv: All right, that's all the questions that I had.

DEK: Good luck to you.

Adv: It's really been a pleasure.

◇ Raph Levien
artofcode
940 Tyler St., Studio 6
Benicia, CA 94510
advogato@advogato.org

Turbulent transition

G. Grätzer

Abstract

It seems that more and more mathematical journals require submission in LaTeX. This welcome trend is causing a lot of problems for journals and mathematicians alike. This article describes what is happening and how could we get out of this unpleasant transition period.

1 Introduction

It was the promise of LaTeX that the submission of mathematical papers to journals will be revolutionized:

- You submit your mathematical article to a journal.
- On acceptance, the editor changes the name of the document class you used to that of the document class of the journal.
- Magically the article is transformed to conform to the format of the journal.
- Nobody touches the article proper, so no errors are introduced; since the change is trivial, there is no cost.

No need to mail and proofread galleys, no need for expensive print shops; everybody gains. There is no difficulty sending your articles to other mathematicians. Converting LaTeX to PDF is easy, so publishing your article (or journal) on the Web is almost cost free.

Talk to a publishing mathematician, and he will tell you many stories about the problems encountered submitting articles in LaTeX. Talk to an editor of a mathematical journal and the editor will relate many stories, albeit from a different viewpoint.

In this article, I will describe the way I see the situation and suggest remedies to ease this transition.

I write about the turbulent transition of mathematical journals from typesetting in print shops to working with LaTeX from the point of view of

1. a mathematician; I have written about 60 mathematical articles in LaTeX and one long mathematical book of almost 700 pages;
2. an Editor-in-chief of an international mathematical journal;
3. an author of many books and articles on LaTeX.

2 Anecdotal evidence

2.1 Submitting articles

In recent months, I have submitted five mathematical articles to international mathematical journals. Let us look at my experience.

2.1.1 First article

This article went to a journal published by the largest publisher of scientific journals in northern Europe. After some searching, I found that they had their own document class. The document class came with a user manual of about 25 pages. To conform to their rules, you have to start from scratch creating the front part of your paper: all the author and article macros have proprietary names and syntax.

Worse than that, the document class also contains some very minor improvements in equation numbering; as a result, the amsmath package doesn't work. They do not quite explain what should I do with my article, since they offer no substitute for the multiline math formulas I use.

The user manual is available as a TEX file or as a PS file. The TEX file I was unable to typeset. I got the error message shown in figure 1. This was not very helpful, since I was trying to typeset the package documentation! Luckily, I could print the PS file without any problem.

Despite the fact that I read the documentation fairly carefully, I have been unable to modify my amsart article. Three e-mail messages were exchanged with the author of the document class. Finally, I had the modified article.

The only amusing note from this experience was the acknowledgments section of the user manual, which reads: "I would like to thank Donald E. Knuth for the fact that he wrote this brilliant program, thereby indirectly supporting my wife, my cats and myself." I can see how he is gainfully occupied helping us convert articles to his document class.

2.1.2 Second article

This was submitted to a journal published by the largest publisher of scientific journals on the east coast of the United States. The document class was written for this publisher by a very well-known writer of TeX macros. When I started out, I was hit on the head with a 50 page manual. Again, almost all the author and article info macros had proprietary names (why is `\affiliation` better than `\address`?) and again there were some minor changes to math typesetting which caused the amsmath package to crash. Why did she have to rename

```
! Package  Error: No encoding for T1 PostScript fonts found.

See the package documentation for explanation.
Type  H <return>  for immediate help.
```

Figure 1

the \address macro? Why did she have to enhance the math, thereby breaking amsmath?

2.1.3 Third and fourth articles

These submissions went to two different journals on two different continents, but ended up being published by the largest publisher of scientific journals on the west coast of the U.S. This publisher does not work with a document class but assigns each article to a technical editor, who makes the format conform to the style of the journal.

I was lucky with the third article; the editor introduced only about a dozen errors, easily caught in the proofs. But with the fourth article I was really unlucky; the editor introduced so many errors that I had to write a letter pointing out that I do not know how to carry out all the changes necessary in the typeset version. They decided to scrap the printed version and the typeset the article from my source file again.

2.1.4 Fifth article

This I submitted to a North American mathematics journal. They have their own document class with a twist. They require that your theorems, lemmas, definitions be defined with thm, lem, and def in the \newtheorem commands, and the document class redefines these, specifying how they be numbered. I think this is really legitimate.

My adventures with article five continued after acceptance and the change in the document class. About a dozen e-mails were exchanged since the publishing office had difficulties with the included graphics files.

2.2 Accepting articles

I have attempted to convince my publisher to switch to LaTeX for the last six years. Last year they finally consented.

When I announced the new submission guidelines to the research community, there was quite a bit of concern. Although we emphasized: "We would strongly encourage electronic submissions of articles. If for some reason you cannot do that, then . . . ", a number of people were of the opinion that we penalize mathematicians who do not have access to LaTeX and the Internet. These concerns proved to be unfounded. Only about 1% of our new submission are not in the required format.

My own journal did not do much better with the printing of the LaTeX papers than the publishers I mention above. I published one paper under this new system with only minor problems, but I received a few weeks ago a letter of a page and a half from two authors of an article. They described in great detail the wholesale butchery of their article. The LaTeX typesetter decided to number by hand all the lemmas and theorems. Unfortunately, the cross references have not been edited. This was just one type of the many mistakes made, resulting in an article that was completely unusable. The last two issues again contain embarrassing mistakes.

2.3 Dispensing advice

As the author of some LaTeX books, I receive voluminous e-mail asking my advice on how to publish articles in LaTeX. To most of the questions I respond: I do not know.

It turns out that most of the questions are platform and application specific. "I use such and such application; how do I include the diagram it creates into a LaTeX article?"

Unfortunately, I am not familiar with the application and the platform; I know, however, that most applications are notorious for producing not very good PS and EPS files. My best advice is to try to switch to a reliable product — if possible.

3 What is a document class?

Many of the problems listed in Section 2.1 seem to come from document classes.

What is a legitimate document class for mathematical articles? It is not easy to axiomatize what should or should not be in a document class. The document class has the right to decide whether equation numbers should be on the left or on the right, but should not try to enhance the way equations are numbered (such tasks should be left to packages). Obviously the document class should format the front matter, should decide on the fonts used in the article and in the running heads, the shape of sections, theorems, and the like. What the document class should not do is interfere with things that we do with standard packages. It should not conflict with amsmath just as it should not conflict with the standard packages used to typeset large tables, and so on.

The AMS set a bad example with the AMS article document class, amsart, since the document class automatically loads the AMS math package, amsmath. This has been corrected only recently with the release of a version 2.0 of the AMS document classes; now the amsart document class can be invoked with the nomath option. amsart with this option is a real document class.

4 Suggestions

Most of the problems we experience in the transition come from inexperience: there are too many new LaTeX shops, technical editors not trained in LaTeX, on the publishing side; and authors unwilling to learn new ways of doing things, on the authoring side.

I think the following simple suggestions would help ease the pain of the transition:

4.1 Document classes

Keep the macro names of the AMS article document class, amsart, unless there is a very good reason to change or modify them.

Keep the user manual short, say, a page or two; make the manual available in standard CMR so any standard TeX installation can print it.

4.2 Submitting papers

Encourage article submissions in DVI or PDF form. The DVI form is the simplest if there are no included graphics. The PDF form is preferable with included graphics (of course, the authors should use PostScript CM and AMS fonts to produce the PDF file). The authors should write articles in standard LaTeX that can be easily adjusted to the journal's document class.

4.3 Accepted papers

The final submission should be in LaTeX. Make a "preprint" form of the document class available for the authors, so the editor does not have to worry about improper line breaks.

4.4 Avoid attachments

Make an ftp site available for submissions, thereby avoiding the problems that often arise with e-mail attachments.

◇ G. Grätzer
Department of Mathematics
University of Manitoba
Winnipeg MN, R3T 2N2
Canada
gratzer@cc.umanitoba.ca
http://server.maths.umanitoba.ca/
homepages/gratzer/

Font Forum

Thai fonts

Werner Lemberg

Abstract

This article describes how the Thai script works and how to implement the necessary ligatures for TEX using afm2tfm.

1 Some Historic Information

The Thai script has been derived, similar to almost all other southern asian scripts not directly influenced by China, from an ancient version of Indic Sanskrit, Brahmi. Over the years, the original letters have been adapted to the peculiarities of the Thai language, one example would be superscripted digits used for tone marks.

Regarding computers, Thailand followed other principles than those of India in the standardization of the script despite many similarities; all base letters are consonants with an inherent vowel (usually an *a*), and for the special case of Thai, an inherent tone. Vowel and tone can be modified by attaching other letters to the base consonant — before and after the base consonant, but also above and below. The Indic standard (ISCII) uses the logical order to store text, which means that the vowel always comes after the consonant even if it appears before the consonant graphically (such vowels are called independent vowels). Contrary to India's logical order, Thailand defines its industrial standard TIS-620 in that independent vowels must be stored in visual order. Unicode [8] follows TIS-620 in the processing of the Thai script.

In the following, linguistic aspects will be completely ignored, referring to graphical features only.

2 The Structure of Thai Letter Clusters

Since letter clusters are stored in visual order the graphical display of Thai is simplified to base letters, possibly with diacritical signs above and below — no need to reorder vowels. There are five possibilities how diacritical signs can be positioned in Thai.

1. base consonant + vowel above:

   ห + ◌ี = หี

2. base consonant + tone mark:

   ป + ◌่ = ป่

3. base consonant + vowel above + tone mark:

ป + ◌ี + ◌่ = ปี่

4. base consonant + vowel below:

ห + ◌ู = หู

5. base consonant + vowel below + tone mark:

ฎ + ◌ู + ◌่ = ฎู่

TIS-620 mandates that tone marks come last, but users sometimes ignore this. It is the duty of input methods for Thai to normalize incorrect input. Below, the standardized form is always expected.

It can already be seen in the above examples that diacritical signs change its positions horizontally and vertically dependent on the shape of the base glyph resp. whether another diacritical sign is used.

There are more peculiarities of the Thai script.

1. The vowel *sara am* ◌ำ will be split into the characters *nikhahit* ◌ํ and *sara aa* า if it is appended to a consonant. The character ◌ํ interacts with the preceding character.

   ก + ◌ำ = ก + ◌ํ + า = กำ

   ก + ◌่ + ◌ำ = ก + ◌ํ + ◌่ + า = ก่ำ

   If necessary, *nikhahit* and the tone mark must exchange its positions.

2. The two consonants *yo ying* ญ and *tho than* ฐ drop its lower part if combined with a lower vowel.

   ญ + ◌ู = ญู

   ฐ + ◌ู = ฐู

3. If *sara aa* า follows the independent vowel *ru* ฤ or *lu* ฦ (those two letters are used for Sanskrit), it will be replaced by the sign *lakkhangyao* ๅ.

   ฤ + า = ฤๅ

   ฦ + า = ฦๅ

3 Glyph Classes

To describe the necessary ligatures it is convenient to categorize Thai letters into various graphical glyph classes, ignoring all linguistical aspects. In the author's opinion, even incorrect or unrealistic combinations should be displayed in an optically pleasing way if possible.

$\boldsymbol{base}_{\boldsymbol{normal}}$ Normal base glyphs without special features.

$\boldsymbol{base}_{\boldsymbol{desc}}$ Base glyphs with descender.

$\boldsymbol{base}_{\boldsymbol{desclike}}$ As described above, glyphs of this class consist of two parts, omitting the lower one if combined with a lower vowel.

$\boldsymbol{base}_{\boldsymbol{asc}}$ Base glyphs with an ascender on the right side.

$\boldsymbol{base}_{\boldsymbol{indic}}$ The two independent vowels *ru* ฤ and *lu* ฦ.

$\boldsymbol{base}_{\boldsymbol{sign}}$ The sign *lakkhangyao* ๅ.

$\boldsymbol{base}_{\boldsymbol{sara\ am}}$ The vowel *sara am* ◌ำ.

$\boldsymbol{base}_{\boldsymbol{sara\ aa}}$ The vowel *sara aa* า.

lower Diacritical vowels below.

$\boldsymbol{upper}_{\boldsymbol{vowel}}$ Diacritical vowels above.

$\boldsymbol{upper}_{\boldsymbol{sign}}$ The sign *nikhahit* ◌ํ.

top Tone marks.

Now the glyph variant forms.

$\boldsymbol{base}_{\boldsymbol{descless}}$ The glyphs of class $\mathit{base}_{\mathit{desclike}}$ without the lower part.

$\boldsymbol{lower}_{\boldsymbol{low}}$ The glyphs of class *lower* shifted downwards.

$\boldsymbol{upper}_{\boldsymbol{vowel}_{\boldsymbol{left}}}$ The glyphs of class $\mathit{upper}_{\mathit{vowel}}$ shifted to the left.

$\boldsymbol{upper}_{\boldsymbol{sign}_{\boldsymbol{left}}}$ The glyphs of class $\mathit{upper}_{\mathit{sign}}$ shifted to the left.

$\boldsymbol{top}_{\boldsymbol{left}}$ The glyphs of class *top* shifted to the left.

$\boldsymbol{top}_{\boldsymbol{low}}$ The glyphs of class *top* shifted downwards.

$\boldsymbol{top}_{\boldsymbol{low\text{-}left}}$ The glyphs of class *top* shifted to the left and downwards.

4 Context Patterns

Using the glyph classes defined in the last section it is easy to describe the context patterns for base glyphs with diacritical signs. Surprisingly, these patterns are quite systematic. Patterns in table 1 which are marked with an asterisk do nothing and are listed for completeness only. As mentioned above, these patterns cover more combinations as existing in the Thai script.

Table 2 covers the ligatures of the character *sara am* ◌ำ. Finally, table 3 describes the letters specific to Sanskrit.

5 Intermezzo 1

A small introduction into the exotic variants of TeX's ligature mechanism which probably many users haven't seen before. Additionally, the documentation in the *METAFONTbook* is very sparse. In the following examples METAFONT's notation is used.

base	*lower*			→	*base*	*lower*			*
base		*upper*		→	*base*		*upper*		*
base			*top*	→	*base*			$\mathit{top}_{\mathit{low}}$	
base	*lower*		*top*	→	*base*	*lower*		$\mathit{top}_{\mathit{low}}$	
base		*upper*	*top*	→	*base*		*upper*	*top*	*
$\mathit{base}_{\mathit{desc}}$	*lower*			→	$\mathit{base}_{\mathit{desc}}$	$\mathit{lower}_{\mathit{low}}$			
$\mathit{base}_{\mathit{desc}}$		*upper*		→	$\mathit{base}_{\mathit{desc}}$		*upper*		*
$\mathit{base}_{\mathit{desc}}$			*top*	→	$\mathit{base}_{\mathit{desc}}$			$\mathit{top}_{\mathit{low}}$	
$\mathit{base}_{\mathit{desc}}$	*lower*		*top*	→	$\mathit{base}_{\mathit{desc}}$	$\mathit{lower}_{\mathit{low}}$		$\mathit{top}_{\mathit{low}}$	
$\mathit{base}_{\mathit{desc}}$		*upper*	*top*	→	$\mathit{base}_{\mathit{desc}}$		*upper*	*top*	*
$\mathit{base}_{\mathit{desclike}}$	*lower*			→	$\mathit{base}_{\mathit{descless}}$				
$\mathit{base}_{\mathit{desclike}}$		*upper*		→	$\mathit{base}_{\mathit{desclike}}$		*upper*		*
$\mathit{base}_{\mathit{desclike}}$			*top*	→	$\mathit{base}_{\mathit{desclike}}$			$\mathit{top}_{\mathit{low}}$	
$\mathit{base}_{\mathit{desclike}}$	*lower*		*top*	→	$\mathit{base}_{\mathit{descless}}$	*lower*		$\mathit{top}_{\mathit{low}}$	
$\mathit{base}_{\mathit{desclike}}$		*upper*	*top*	→	$\mathit{base}_{\mathit{desclike}}$		*upper*	*top*	*
$\mathit{base}_{\mathit{asc}}$	*lower*			→	$\mathit{base}_{\mathit{asc}}$	*lower*			*
$\mathit{base}_{\mathit{asc}}$		*upper*		→	$\mathit{base}_{\mathit{asc}}$		$\mathit{upper}_{\mathit{left}}$		
$\mathit{base}_{\mathit{asc}}$			*top*	→	$\mathit{base}_{\mathit{asc}}$			$\mathit{top}_{\mathit{low\text{-}left}}$	
$\mathit{base}_{\mathit{asc}}$	*lower*		*top*	→	$\mathit{base}_{\mathit{asc}}$	*lower*		$\mathit{top}_{\mathit{low\text{-}left}}$	
$\mathit{base}_{\mathit{asc}}$		*upper*	*top*	→	$\mathit{base}_{\mathit{asc}}$		$\mathit{upper}_{\mathit{left}}$	$\mathit{top}_{\mathit{left}}$	

Table 1: Context patterns for diacritical signs. Here, *base* refers to the union of the subclasses *normal*, *indic*, *sign*, *sara am*, and *sara aa* of *base*; *upper* is the union of the subclasses *vowel* and *sign* of *upper*.

base	$\mathit{base}_{\mathit{sara\ am}}$		→	*base*	$\mathit{upper}_{\mathit{sign}}$		$\mathit{base}_{\mathit{sara\ aa}}$
$\mathit{base}_{\mathit{asc}}$	$\mathit{base}_{\mathit{sara\ am}}$		→	$\mathit{base}_{\mathit{asc}}$	$\mathit{upper}_{\mathit{sign}_{\mathit{left}}}$		$\mathit{base}_{\mathit{sara\ aa}}$
base	*top*	$\mathit{base}_{\mathit{sara\ am}}$	→	*base*	$\mathit{upper}_{\mathit{sign}}$	*top*	$\mathit{base}_{\mathit{sara\ aa}}$
$\mathit{base}_{\mathit{asc}}$	*top*	$\mathit{base}_{\mathit{sara\ am}}$	→	$\mathit{base}_{\mathit{asc}}$	$\mathit{upper}_{\mathit{sign}_{\mathit{left}}}$	$\mathit{top}_{\mathit{left}}$	$\mathit{base}_{\mathit{sara\ aa}}$

Table 2: Context patterns for *sara am* ำ. Here, *base* denotes the union of subclasses *normal*, *desc*, and *desclike* of *base*.

$\mathit{base}_{\mathit{indic}}$	$\mathit{base}_{\mathit{sara\ aa}}$	→	$\mathit{base}_{\mathit{indic}}$	$\mathit{base}_{\mathit{sign}}$

Table 3: Context patterns for *ru* ฤ and *lu* ฦ.

The usual ligature action of two glyphs *a* and *b* is the replacement of both glyphs with another glyph *c*.

```
a b =: c
```

Another possibility is to retain the left or the right original glyph (before resp. after the ligature) or both.

```
a b |=: c   a b =:| c   a b |=:| c
```

The first rule creates *ac*, the second *cb*, and the last *acb*. In all three cases, the current point after appying the ligature rule is still at the first glyph of the replaced glyphs, and TEX simply restarts there to check ligatures (and kernings). A classical example is

f f i → ff i → ffi

To advance the current point to the right, append either `>` or `>>` (the latter is only possible if you retain both input glyphs). Here are the remaining four ligature rules.

```
  a b |=:> c   a b =:|> c
a b |=:|> c   a b |=:|>> c
```

For Thai ligatures, the most often needed rule is `|=:` (i.e., retain the left glyph and stay at the same position before applying the next ligature rule). Note that using `|=:>` instead is not a good idea since this would prohibit kerning between the left glyph and the ligature.

6 Ligature Rules

As just explained, TEX can only handle context patterns of length 2, whereas Thai needs patterns of length 3. It was an interesting challenge to find out whether the problem can be solved with TEX's somewhat restricted ligature rules — the gentle reader is invited to find a solution by herself! There won't be any difficulties in understanding ligatures afterwards.

The tables 4, 5, and 6 use the same conventions as tables 1, 2, and 3, respectively. The current point isn't increased in any of the rules.

Most of the ligature rules can be derived easily by handling the patterns sequentially (quite similar to logic puzzles found in various magazines), but at the end there remain two patterns which apparently contradict.

$$\begin{array}{llllll} \textit{base} & \textit{lower} & \textit{top} \rightarrow & \textit{base} & \textit{lower} & \textit{top}_{\textit{low}} \\ \textit{base}_{\textit{asc}} & \textit{lower} & \textit{top} \rightarrow & \textit{base}_{\textit{asc}} & \textit{lower} & \textit{top}_{\textit{low-left}} \end{array}$$

After applying ligature rules for the first two glyph classes it is necessary to handle the context '*lower top*', but depending on the previous glyph class *top* must be replaced with $\textit{top}_{\textit{left}}$ and $\textit{top}_{\textit{low-left}}$, respectively. With a context pattern length of 3 this would be easy to solve, but TEX doesn't have this feature. What to do?

The context '*base lower top*' must be distinguished from '$\textit{base}_{\textit{asc}}$ *lower top*', i.e., two different *lower* classes are needed depending on the previous character since TEX is not able to forward information from one ligature cycle to the next. The idea is now to create an 'alias class', a class which behaves identically to the original one. The glyphs in this alias class are the same, but different glyph indices resp. glyph names are assigned to it. A closer look to table 4 shows that $\textit{lower}_{\textit{left}}$ isn't a typo but the alias class of *lower*.

7 Intermezzo 2

`afm2tfm` [6] uses two encoding vectors to create metrics files for TEX. The first maps from the Type 1 font to the *raw* font (converting glyph names to glyph indices):

```
/ps_to_raw [ ... /bar ... /bar ... ] def
```

The second encoding vector used for the *virtual* font (which will contain the ligatures) maps from glyph indices back to glyph names, so we can finally assign different glyph names to identical glyphs:

```
/raw_to_vf [ ... /bar1 ... /bar ... ] def
```

`bar1` and `bar` now both access the same glyph.

Unfortunately, `afm2tfm` can only use glyph names for the virtual font which are already in the base font, so some manual work is needed to overcome this restriction. It is planned to reimplement Thai ligatures using the fontinst package [2] which doesn't have this problem.

8 The Implementation

After solving the problem theoretically now the practical implementation. Only some interesting details are shown. Since it is not possible in `afm2tfm` to collect glyphs in classes the number of all ligature rules is quite big (464 in total). The file `thai.enc`, part of the CJK package [3], contains the complete solution. It also contains detailed installation instructions how to use `afm2tfm`.

All glyph names follow the *Adobe Glyph List (AGL)* [1]. There are no predefined Adobe glyph names for Thai, so the prefix 'uni' with attached Unicode value will be used for all glyphs which are encoded in Unicode. Example: The letter *ko kai* ก gets the name `uni0E01`. Glyph variants are identified by a postfix. Example: The left-shifted glyph variant of the vowel *sara uee* ◌ื is called `uni0E37.left`.

The following listing describes some of the ligatures, explaining its function.

$base$	top	→	$base$	top_{low}
$base_{desc}$	$lower$	→	$base_{desc}$	$lower_{low}$
$base_{desc}$	top	→	$base_{desc}$	top_{low}
$base_{desclike}$	$lower$	→	$base_{descless}$	$lower$
$base_{desclike}$	top	→	$base_{desclike}$	top_{low}
$base_{asc}$	$lower$	→	$base_{asc}$	$lower_{left}$
$base_{asc}$	$upper$	→	$base_{asc}$	$upper_{left}$
$base_{asc}$	top	→	$base_{asc}$	$top_{low\text{-}left}$
$lower$	top	→	$lower$	top_{low}
$lower_{low}$	top	→	$lower_{low}$	top_{low}
$upper_{left}$	top	→	$upper_{left}$	top_{left}
$lower_{left}$	top	→	$lower_{left}$	$top_{low\text{-}left}$

Table 4: Ligature rules for diacritical marks.

$base$	$base_{sara\ am}$	→	$base$	$upper_{sign}$	$base_{sara\ am}$
$base_{asc}$	$base_{sara\ am}$	→	$base_{asc}$	$upper_{sign_{left}}$	$base_{sara\ am}$
$upper_{sign}$	$base_{sara\ am}$	→	$upper_{sign}$	$base_{sara\ aa}$	
$upper_{sign_{left}}$	$base_{sara\ am}$	→	$upper_{sign_{left}}$	$base_{sara\ aa}$	
top_{low}	$base_{sara\ am}$	→	top_{low}	top	$base_{sara\ am}$
top_{low}	top	→	$upper_{sign}$	top	
top	$base_{sara\ am}$	→	top	$base_{sara\ aa}$	
$top_{low\text{-}left}$	$base_{sara\ am}$	→	$top_{low\text{-}left}$	top_{left}	$base_{sara\ am}$
$top_{low\text{-}left}$	top_{left}	→	$upper_{sign_{left}}$	top_{left}	
top_{left}	$base_{sara\ am}$	→	top_{left}	$base_{sara\ aa}$	

Table 5: Ligature rules for *sara am* ำ.

$base_{indic}$	$base_{sara\ aa}$	→	$base_{indic}$	$base_{sign}$

Table 6: Ligature rule for *ru* ฤ and *lu* ฦ.

- Rule: $\mathit{base}\ \mathit{top} \rightarrow \mathit{base}\ \mathit{top}_{\mathit{low}}$

  This rule needs 225 ligatures (45 base glyphs × 5 tone marks); this is almost 50 % of all rules.

  ```
  % LIGKERN uni0E01 uni0E48 |=: uni0E48.low ;
  % LIGKERN uni0E02 uni0E48 |=: uni0E48.low ;
  ...
  % LIGKERN uni0E01 uni0E49 |=: uni0E49.low ;
  % LIGKERN uni0E02 uni0E49 |=: uni0E49.low ;
  ...
  % LIGKERN uni0E41 uni0E4C |=: uni0E4C.low ;
  % LIGKERN uni0E46 uni0E4C |=: uni0E4C.low ;
  ```

- Rule: $\mathit{base}_{\mathit{desclike}}\ \mathit{lower} \rightarrow \mathit{base}_{\mathit{descless}}\ \mathit{lower}$

  Here the left glyph will be replaced.

  ```
  % LIGKERN uni0E0D uni0E38 =:| uni0E0D.descless ;
  % LIGKERN uni0E10 uni0E38 =:| uni0E10.descless ;
  ...
  ```

- Rule: $\mathit{base}\ \mathit{base}_{\mathit{sara\ am}} \rightarrow$
  $\mathit{base}\ \mathit{upper}_{\mathit{sign}}\ \mathit{base}_{\mathit{sara\ am}}$

  The context pattern

  $$a\ b \rightarrow a\ c\ d$$

  has to be transformed to the following for TeX (as shown in table 5):

  $$\begin{gathered} a\ b \rightarrow a\ c\ b \\ c\ b \rightarrow c\ d \end{gathered}$$

  The first ligature is of interest:

  ```
  % LIGKERN uni0E01 uni0E33 |=:| uni0E4D ;
  % LIGKERN uni0E02 uni0E33 |=:| uni0E4D ;
  ...
  ```

9 The Font Encoding

The real encoding of the virtual font is irrelevant for ligature rules because glyph names have been used exclusively. Nevertheless, it has practical advantages to use TIS-620 as a font encoding also, filling unused positions with glyph variants (this is similar to the Unicode++ font encoding of Ω [5]). Especially users of Plain TeX will benefit if input and output encoding are identical. Care must be taken in LaTeX documents to avoid the use of `\uppercase` and `\lowercase` commands so that Thai letters aren't modified due to incorrect `\lccode` and `\uccode` values.

Table 7 shows the used encoding of the Thai glyphs.

10 Problems

From a typographical point of view, all problems are solved. To believe that it is now possible to simply enter Thai for getting correct output is an error, though. Two serious obstacles must be mastered first: finding word breaks and insertion of intercharacter glue.

Words in Thai are *not* separated with spaces, and they aren't hyphenated either.[1] A space, usually much bigger than a space for the Latin script, has a similar function to an em-dash or to a semicolon; its primary use is to structure a sentence.

Correct recognition of words in Thai is a very complex problem which can be solved without errors by sentence analysis only. The CJK package uses a relatively simple algorithm developed by Vuthichai Ampornaramveth วุฒิชัย อัมพรอร่ามเวทย์ which basically searches for the longest words in a dictionary (this is implemented as a Lisp package for Emacs — it is assumed that the next major version of Emacs will contain this module directly). Due to missing context analysis it can't guarantee error-free results in all cases.

Another complication is that Thai tends to very long words, making the search for good break points in justified paragraphs difficult. A legitimate solution is to moderately apply some intercharacter glue (cf. figure 1). An even better solution would be the use of *Multiple Master* fonts or similar fonts to enable small variations of the typeface. Newer versions of pdfTeX have experimental support for stretching and squeezing of fonts [7].

11 Thai Support In The CJK Package

The results presented in this paper will be part of the next version of the CJK package (daily snapshots of the development archive are available from `ftp://ftp.ffii.org/pub/cjk/devel/cjk-current.tar.gz`). Included are (in addition to `thai.enc` and other auxiliary files) encoding, metrics, and font definition files for the freely available Thai font families DBThai and Norasi [4]. The latter uses glyphs created by Yannis Haralamous and Tereza Tranaka and are still under development; all Thai examples in this article have been typeset with it.

Intercharacter glue and word break points will be inserted automatically by the Emacs interface cjk-enc. This package could be considered as a generalized inputenc package which is able to handle multiple character sets in Emacs simultaneously, and which does correct translation to LaTeX transparently to the user.

12 Acknowledgements

Most of the development work on Thai ligatures has been done during a two-week research visit in Tsukuba (Japan) on invitation of the Electrotechnical Laboratory (ETL).

[1] For narrow-column printing, hyphenation is used, but it isn't considered as good typography.

	0	1	2	3	4	5	6	7
20	ุ	ู	ฺ	่	้	๊	๋	์
21	่	้	๊	๋	์		๎	ํ
22	ญ		ั	็	ิ	ี	ึ	ื
23	่	้	๊	๋	์			ฐ
24		ก	ข	ฃ	ค	ฅ	ฆ	ง
25	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ
26	ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ท
27	ธ	น	บ	ป	ผ	ฝ	พ	ฟ
30	ภ	ม	ย	ร	ฤ	ล	ฦ	ว
31	ศ	ษ	ส	ห	ฬ	อ	ฮ	ฯ
32	ะ	ั	า	ำ	ิ	ี	ึ	ื
33	ุ	ู	ฺ	‘	’	“	”	฿
34	เ	แ	โ	ใ	ไ	ๅ	ๆ	็
35	่	้	๊	๋	์	ํ	๎	๏
36	๐	๑	๒	๓	๔	๕	๖	๗
37	๘	๙	๚	๛	ุ	ู	ฺ	

Table 7: The Thai encoding of the CJK package. Range 241–373 (0xA1–0xFB) without 333–336 (0xDB–0xDE) is TIS-620, the rest are glyph variants. 241–316 are consonants. 320–332, 340–344, and 347 are vowels. 337 is the Thai currency symbol, Baht. 350–353 are tone marks. 360–371 are the digits 0 to 9. The alias class to *lower* (at position 330-332), *lower*$_{left}$, is at position 330–332.

รายการ FAQ นี้สร้างขึ้นเพื่อสรุปคำถามที่ถามกันบ่อยครั้งและคำ
ตอบคำถามในรูปแบบทีสะดวก. โครงสร้างของรายการ FAQ นี้เปลี่ยนไ
ปมากตั้งแต่รุ่นที่แล้ว. **ดูรายละเอียดสำหรับโครงสร้างใหม่ได้จากช่วง
“โครงสร้างและวิธีการอ่าน FAQ.”**

รายการ FAQ นี้สร้างขึ้นเพื่อสรุปคำถามที่ถามกันบ่อยครั้งและคำ
ตอบคำถามในรูปแบบทีสะดวก. โครงสร้างของรายการ FAQ นี้เปลี่ยนไ
ปมากตั้งแต่รุ่นที่แล้ว. **ดูรายละเอียดสำหรับโครงสร้างใหม่ได้จากช่วง
“โครงสร้างและวิธีการอ่าน FAQ.”**

Figure 1: The same text, with and without intercharacter glue. To suppress warnings and error messages for the above variant, `\tolerance` had to be set to 8000 and `\badness` to 10000. `\baselinestretch` has the value 1.2.

References

[1] The Adobe Glyph List. `http://partners.adobe.com/asn/developer/typeforum/unicodegn.html`.

[2] Allan Jeffrey et al. The fontinst package. Available from CTAN and its mirrors, e.g. `ftp://ftp.dante.de/pub/tex/fonts/utilities/fontinst`.

[3] Werner Lemberg. The CJK package. `http://cjk.ffii.org`.

[4] Surapant Meknavin and Theppitak Karoonboonyanan. The thailatex package. `ftp://opensource.thai.net/pub/linux-tle/updates/SOURCES/thailatex-0.2.%1.tar.gz`. The implementation for Thai in this package is incompatible to the one described in this article. For this reason, the Babel module of the CJK package is called ‘thaicjk’ and not ‘thai’.

[5] John Plaice and Yannis Haralambous. The Ω system. `http://www.gutenberg.eu.org/omega`. Almost all modern TEX distributions contain support for Ω.

[6] Tomas Rokicki. The `afm2tfm` program. Part of the dvips package which is available from virtually all TEX distributions.

[7] Han The Thanh. pdfTEX. `ftp://ftp.cstug.cz/pub/tex/local/cstug/thanh/pdftex/latest`. pdfTEX is, similar to Ω, already part of most modern TEX distributions. The given URL specifies the primary address of pdfTEX since it still in development, sometimes with incompatible changes.

[8] The Unicode Standard. `http://www.unicode.org`.

◇ Werner Lemberg
Kl. Beurhausstr. 1
44137 Dortmund
wl@gnu.org

Exploiting Rich Fonts

Sivan Toledo

1 Introduction

Rich fonts with hundreds or thousands of glyphs are becoming widely available. Such fonts are now bundled with operating systems, printers, and consumer software. Rich fonts enable us to achieve better typographic results than possible with conventional fonts. This paper explores three kinds of rich fonts, their typographic features, and ways to exploit these features in TEX systems.

The three kinds of fonts that the paper explores are:

1. PostScript 3 fonts. Adobe defined a core set of fonts to be included in all PostScript 3 devices [1]. Two families in the core set, Hoefler Text and Apple Chancery, are particularly interesting, since their fonts contain hundreds of glyphs. This article focuses on one family, Hoefler Text, a general-purpose text family. The glyph contents of Apple Chancery is fairly similar to that of Adobe Poetica, for which TEX support has already been developed [7, 5].
2. WGL4 Fonts. WGL4 (Windows Glyph List 4) is a glyph list published by Microsoft and intended to support all European languages, including Greek and Russian. Many fonts bundled with Microsoft's Office products are WGL4 fonts, and the inclusion of Greek letters in them offers new creative possibilities for mathematical typesetting.
3. Palatino Linotype. Palatino Linotype is a font family bundled with Microsoft's Windows 2000 Professional operating system. These fonts are WGL4 fonts, but they also contain many more Latin glyphs than most WGL4 fonts, and thus allow more typographic options.

Both Hoefler-Text and Palatino-Linotype enable one to achieve higher typographic quality than is possible with other font families, including Computer Modern and families for which there are expert fonts. The fonts in these families include several sets of numerals for different uses and more standard ligatures than other text typefaces. They also contain quaint ligatures, alternate glyphs and swashes, but these are not particularly rare.

We can expect additional rich fonts to become available. Adobe is preparing new OpenType "Pro" fonts, which will also include multiple sets of numerals and other features. One "Pro" font family is already available, Tekton Pro, which is bundled with Adobe's InDesign page layout program. Adobe

Hoefler Text is a rich font family. IT INCLUDES SMALL CAPITALS (*IN ALL THE SHAPES* ***AND WEIGHTS***). It includes both old-style numerals, such as 69978, and lining numerals, such as 69978. It includes many standard ligatures, such as fj for fjord, quaint ligatures such as ſt for adding interſt. It also includes alternate and swash characters, as in Quick, *wow, Hoefler & Hoefler, fore & aft*. The fonts also include components for fractions, such as 6^{23}⁄47. The fonts contain both fixed width and variable width numerals, for tables and running text, respectively. Compare the two columns:

Variable Width	Fixed Width
1111.11	1111.11
4567.23	4567.23
1111.11	1111.11
4567.23	4567.23

Figure 1: The typographic features of the Hoefler Text family (most but not all).

Palatino Linotype is a rich font family. IT INCLUDES SMALL CAPITALS (*IN ALL THE SHAPES* ***AND WEIGHTS***). It includes both old-style numerals, such as 69978, and lining numerals, such as 69978. It includes many standard ligatures, such as fj for fjord, quaint ligatures such as ſp for ſpecial occasions. The fonts also include components for fractions, such as 6^{23}⁄47. Since the fonts contain Greek letters, we can use them for mathematical typesetting, as in $\max 2x + 5\alpha = \Sigma_{i=1}^{\beta}(8\xi + \Omega)$. There are even two variants of theta (θ and ϑ), phi (ϕ and φ), sigma (σ and ς), and pi (π and ϖ). There is only one version of epsilon, however (ε). The Upsilon (Υ) is different from the Latin Y. The fonts contain both fixed width and variable width numerals, for tables and running text, respectively. There are only variable-width lining numerals; there are no old-style variable-width numerals. Compare the two columns:

Variable Width	Fixed Width
None	1111.11
None	4567.23
1111.11	1111.11
4567.23	4567.23

Figure 2: The typographic features of the Palatino-Linotype family.

Franklin Gothic is a family of WGL4 fonts. Such fonts don't have a rich set of Latin glyphs, but they do have glyphs for Greek and Cyrillic letters. We can use the Greek letters and several symbols for mathematical typesetting, as in $T_1(\alpha, \xi) = \Theta(n^2) = \Sigma_i x_i$. In many of these fonts, however, the shape of some of the letters, especially α, is not suitable for mathematics.

Figure 3: The typographic features of the Franklin-Gothic family.

has announced that it will issue additional "Pro" families, including Adobe Garamond, Minion, and Myriad. The TEX community should, therefore, prepare to exploit these fonts.

Adobe's OpenType fonts are beyond the scope of this paper because their glyphs are stored in a so-called CFF format, a form of compressed Type1 outlines. To the best of my knowledge, there are no tools available today which enable the use of such fonts with TEX systems. The other fonts that this paper discusses have outlines in TrueType format which is relatively easy to use with TEX.

2 TrueType and OpenType Fonts

All the fonts discussed in this article are TrueType or OpenType fonts. The TrueType font format was originally developped by Apple and is now widely used on Machintosh, Windows, and Unix computers. The glyph in a TrueType font are described by outlines (quadratic splines). The hints that ensure high-quality rendering at low resolution are essentially little program that move the control points of the outlines to fit the discrete pixel grid.

TrueType fonts offer two advantages over PostScript Type1 fonts, another widely used outline font format. First, since the hints are programs it is possible to achieve higher quality rendering at low resolution than with Type1 fonts, which have declarative hints. (Hinting TrueType fonts well generally requires a significant effort by an expert font engineer, however.) Second, TrueType fonts combine the description of glyphs, metric information (glyph metrics and kerning pairs), and auxiliary information into a single file, so TrueType fonts are often easier to install and manage than PostScript fonts.

There are now several variants of TrueType fonts. First, Windows and Macintosh computers use slightly different formats. This article deals only with Windows fonts, which are also usable on Unix and Linux. TrueType GX, now known as AAT (Apple Advanced Typography), is an extension of the original TrueType font format that was designed by Apple to offer advanced typographic refinement and control [2]. Apple commissioned and distributed a few GX fonts as part of the Macintosh's operating system, including Hoefler Text, Apple Chancery, and Skia. The only major font manufacturer that offers GX fonts is Linotype. GX fonts offer the typesetter pair kerning, tracking, ligatures, baseline adjustments, vertical substitution, optical alignment of line edges, accent positioning, several sets of numerals, fractions, and glyph variations. Most of these features work automatically; several are under the control of the typesetter (actually the typesetting program), such as selection of glyph variants and type of numerals.

OpenType is another enhancement to TrueType. The OpenType specification was developped by Adobe and Microsoft. OpenType offers two major advantages over conventional TrueType. First, OpenType fonts support advanced typographic features such as glyph substitution (ligatures, small capitals, old-style numerals, contextual shaping of arabic, and so on). Second, The glyphs in OpenType fonts can be described using the normal TrueType machinery or using Type1 machinery. This allows lossless conversion of Type1 fonts to OpenType.

Both GX and OpenType enhance TrueType buy adding so-called *tables*, or data structures, to the font file. The basic TrueType structures are maintained, however, so GX fonts and OpenType fonts with TrueType outlines are also valid TrueType fonts (OpenType fonts with Type1 outlines are not valid TrueType fonts).

3 PostScript 3 Core Fonts

PostScript 3 is new enhancement of the PostScript language (the previous standard is called PostScript Level 2). Several high-end laser printers already support PostScript 3.

Among the enhancements in PostScript 3 over PostScript levels 1 and 2 is a new set of core fonts that must be supported by all PostScript 3 devices. The core sets includes 136 fonts:

- The 35 standard PostScript fonts (Avant Garde, Bookman, Courier, Helvetica, Helvetica Condensed, New Century Schoolbook, Palatino, Times, Symbol, Zapf Chancery, and Zapf Dingbats).
- Some of the core fonts in Microsoft Windows operating systems (Arial and Times New).
- Some of the core fonts in the Apple Macintosh operating system (Chicago, Geneva, Monaco, and New York), as well as two of Apple's GX font families (Apple Chancery and Hoefler Text).

- Several fonts that are bundled with Microsoft Office products and with HP printers.
- Additional Adobe font families, such as Stempel Garamond, Carta, Joana, and Univers.

The 136 core fonts are resident in PostScript 3 printers, such as HP's 2500CM, Lexmark's T series, and Xerox's DocuColor CP40. Furthermore, to enable printer users to utilize all the fonts resident in their printers, printer manufacturers bundle screen fonts with the printers. Thus, users of PostScript 3 printers have access to all 136 fonts in either PostScript Type1 format or in TrueType format. Both HP and Xerox distribute the complete set of screen fonts for both Macintosh and Windows (the Windows-format fonts can be used on Unix and Linux systems as well). Thus, users of Microsoft Windows, Unix and Linux systems who purchase such printers now have access to fonts that were once available only on Macintosh systems. These fonts can be easily used by TEX and Omega[1].

4 The Hoefler-Text Font Family

Hoefler Text, designed by Jonathan Hoefler, was originally commissioned by Apple. The fonts were originally available in Apple's TrueType GX format only. It is now available from the Hoefler Type Foundry (www.typography.com) in PostScript Type1 as well. It is also available in Windows TrueType format as a screen font bundled with several printers, as explained above.

The Windows TrueType versions of the Apple fonts in the PostScript 3 core set retain all the glyphs that are present in the GX fonts, but not all the metric information. The glyph sizes are, of course, present, and so are the kerning pairs (the number of pairs ranges from 1784 in the regular font to 6907 in the black italic). The other metric and substitution information cannot be represented in non-GX TrueType fonts and is lost. (Most of the extra GX font information can be represented in OpenType fonts, a newer enhancement of the TrueType format, but my fonts were not OpenType fonts.)

PostScript 3's core font set includes five fonts from the Hoefler Text family: regular, italic, black, black italic, and ornaments (bold-weight fonts are available from the Hoefler Type Foundry, but are not part of the PostScript 3 core fonts). The regular font actually hides another font consisting of engraved capitals. The text fonts include between 711–881 glyphs each. These glyphs can be classified into the following categories:

- Latin letters, accents, and precomposed characters.
- Swash capitals in the italic fonts; most letters have one swash form, A and M have two, some have none. There is also a long-tail Q in the upright fonts.
- Initial and final swash variants for a few lowercase letters in the italic fonts.
- Small capitals in all the fonts. The italic fonts also contain a few variant small capitals.
- Standard, quaint, and archaic ligatures. The so-called standard ligatures include not only fi, fl, ff, ffi, and ffl, but also fb, fj, ffb, and many more (16 altogether in the italic fonts, less in the uprights). The quaint ligatures include ligatures for ct and st. Again, the italic fonts have many more than the uprights. The archaic ligatures are ligatures involving long-s, which is also present in all the fonts. The standard ligatures are supposed to always be used, the quaint ones under manual control of the typesetter. There are also swash forms for some of the ligatures.
- Four sets of numerals: variable-width old-style numerals (the default), variable-width lining numerals, fixed-width old-style and lining numerals. The fixed-width numerals are designed for tables, the variable-width for all other applications.
- Three sets of punctuation marks and monetary symbols. The default set is designed for text composed with capitals and lowercase, another set for all capitals, and a third set for text composed with small capitals only. The monetary symbols of the second set are also suitable for using with lining numerals in normal text, since their hight is aligned with that of the lining numerals.
- Fraction components and a few precomposed fractions. The components include a fraction slash and numerals for the numerator and denominator. When composed, the height of the fractions is aligned with the height of the capitals and lining figures.
- Inferior and superior lowercase letters and numerals designed for superscripts and subscripts. These are not particularly useful for TEX because the character sets are incomplete, but they should be suitable for marking footnotes.

[1] Omega is a TEX-like program supporting 16-bit characters. It is now a standard part of some TEX distributions, such as teTEX, so most users can use it directly without any need to install new software. Omega preloaded with LATEX is invoked using the command `lambda`.

- Symbols: symbols used in text (e.g., section and paragraph symbols) and a few mathematical symbols.

5 TEX Usage

From the user's point of view, the Hoefler Text family for TEX includes the seven font families shown in Figure 4:

- The default family uses variable-width old-style numerals and contains eight fonts: upright medium and bold, italic medium and bold, and four caps-and-small-caps fonts.
- The engraved family with one font, containing only capitals (lowercase characters are mapped to uppercase glyphs).
- Three families with four fonts each (regular, italic, bold, bold-italic) containing either variable-width lining numerals, fixed-width old-style numerals, or fixed-width lining numerals. These are upper-and-lowercase fonts.
- A family of four (regular, italic, bold, bold-italic) fonts that map lowercase to upper case and use punctuation and monetary signs that align with the capitals (e.i., they are slightly raised compared to the default signs).
- A family of four fonts that map both upper and lower case to small-caps glyphs and use punctuation and monetary signs that align with the small capitals.

The fonts use the T1 encoding. A LATEX package that I have written defines the families using the New Font Selection Scheme.

These fonts activate the so-called standard ligatures (including fb, fj, etc.) automatically. The fonts also contain quaint ligatures and alternate and swash glyphs, but these are not used automatically. To select an alternate character, user must insert the ^ characters (ASCII circumflex) after the base letter (this assumes that ^ is a letter; more on that later). Two consecutive ^'s after a base letter generates the second alternate glyph, if it exists. To activate a quaint ligature, the user inserts a ^ between the constituent letters of the ligature. Figure 5 shows how to use this technique.

A very similar mechanism is used to typeset fractions. To typeset a fraction whose numerator and denominator are separated by a fraction slash, the user types `/^` to start the fraction, then the numerator, a slash, and the denominator. This mechanism works in horizontal mode, not math mode. Figure 5 demonstrates this mechanism as well.

TEX must see the code for the ^ letter itself, which normally needs to be specified in LATEX input file using the macro `\textasciicircum`. I mostly use LYX[2] to prepare input files for LATEX, so I simply type the ^ and LYX inserts the macro in the LATEX input file. If you use a text editor to prepare input files, a character that TEX considers to be a letter, perhaps the asterisk, would be preferable.

6 Technicalities

Let us first explain the basics of using TrueType and OpenType fonts, rich or conventional, with TEX. While some TEX implementations, such as TrueTEX, support TrueType fonts directly, most TEX systems require that TrueType fonts be prepared before they can be used. The purpose of the conversion is to produce metric files (.tfm, .ofm for Omega) and sometimes virtual fonts (.vf, .ovf for Omega) for TEX or Omega, and to produce a "glyph container" for the output processor (e.g., `dvips`). The metric and virtual-font files are produced in two steps. First, a utility extracts the metric information from the TrueType fonts and stores it in an AFM file in the format that Type1 metrics is stored in. Several free utilities, such as `ttf2afm` and `ttfps` can perform this conversion. Second, we need to convert the TrueType font file to a format that the output processor can process. We can produce bitmapped fonts at various sizes using the utility `ttf2pk`. The resulting pk font files can be used by most DVI processors. The disadvantage of bitmapped fonts is that PDF files containing them don't render very well on the screen. We can also convert the TrueType font files into scalable font files that can be embeded in PostScript files. One method is to convert the outlines to Type1 outlines. This is a lossy process in that the hints in the original font are lost, but it produces fonts that can be used on any PostScript device. Another method is to convert the TrueType font file to a Type42 PostScript font file. A Type42 font is simply a TrueType font file wrapped in a PostScript program. Ghostscript can process such fonts, as well as most PostScript Level 2 printers, but some older PostScript printers cannot. Adobe distiller can embed such fonts in PDF documents. The free utilities `ttfps` and `gfontview` can convert TrueType fonts to Type42 fonts. PdfTEX can embed TrueType fonts directly in its output, but it does

[2] LYX (`www.lyx.org`) is a visual editor for LATEX. LYX is not a what-you-see-is-what-you-get system, but its screen display is a reasonable rendition of the LATEXfile: section titles appear in a large bold font, mathematics are shown using greek symbols and large delimiters, and so on. Mathematics can be entred using LATEXcommands: type `\alpha` and hit Escape and a greek α appears.

Variable width old-style	(Notice the Punctuation, 2691)
Fixed width old-style	(Notice the Punctuation, 2691)
Variable width lining	(Notice the Punctuation, 2691)
Fixed width lining	(Notice the Punctuation, 2691)
Caps and small caps	(NOTICE THE PUNCTUATION, 2691)
All capitals	(NOTICE THE PUNCTUATION, 2691)
All small capitals	(NOTICE THE PUNCTUATION, 2691)
Engraved capitals	(PUNCTUATION, 2691)

Figure 4: Hoefler font families

Input	**Typeset output**
`Quick Q^uick More M^ore M^^ore`	Quick Quick *More More More*
`w^ow at^ ffk^ & &^ &^^`	*wow at ffk & & &*
`c^t s^t s^p i^s g^g gg T^h`	ct st *sp is gg gg Th*
`/^1/8 /^11/64`	1⁄8 11⁄64

Figure 5: How to use alternates, swashes, quaint ligatures, and fractions

need TEX metric files (.tfm). To the best of my knowledge, there are no utilities that can prepare OpenType fonts with Type1 outlines for use with TEX output processors or with pdfTEX.

To use the Hoefler fonts with TEX and Omega, we must generate metric files. To use the fonts with `dvips` and `odvips`, we also have to generate PostScript fonts. I generated `afm` metric files from the TrueType fonts using `ttfps`. I then used the afm files to build the TEX/Omega virtual fonts and the TEX/Omega font metric files. `Ttfps` can also generate PostScript Type42 fonts files. It turned out, however, that `ttfps` produced defective Type42 fonts because the Hoefler fonts are large. I therefore produced another set of Type42 fonts using a program called `gfontview`; these were fine. (The problem with `ttfps` is that it encodes the ttf file in one PostScript array, and these array grow too large on large fonts. The ttf file should be broken into several arrays on table boundaries, which is what `gfontview` does.) The PostScript output files that dvips and odvips produce using these fonts can be processed by ghostscript (e.g., they can be screen previewed), can be printed on most PostScript Level 2 printers, and can be converted to PDF using Adobe Distiller. Of course, if you have a PostScript 3 printer there is no need to embed the Type42 fonts in the output file at all. The TrueType fonts can also be embedded in PDF files directly by pdfTEX—there is no need for Type42 fonts here. A final word on using these fonts in PostScript and PDF documents: you should check the license that come with the printer software to determine what uses of the fonts are legal and which are not.

Let's turn our attention to the virtual fonts that represent the fonts for TEX and Omega. The TEX virtual fonts manage to contain both all the glyphs defined by the T1 encoding as well as all the alternates, swashes, extra ligatures, and fraction components because they are 16-bit Omega fonts. That is, the lower 256 glyphs in the virtual fonts are the glyphs defined by the T1 encoding. Beyond glyph 255, the fonts contain all the remaining glyphs in the Hoefler fonts in a random order. The extra glyphs are accessed using ligatures of characters in the range 0–255. Thus, an 8-bit input file that uses only one font can generate an output file with more than 256 glyphs. For example, the characters f and j are automatically combined into an fj glyph using ligature specification in the file. The index of the fj glyph in the font is larger than 255. Likewise, the characters M and ˆ are combined into a glyph of a swash M.

Fractions are supported in a similar way. The `/^` combination is transformed into a little used glyph, a superior comma. A superior comma followed by a numeral is changed into a superior numeral. Any superior numeral followed by another numeral changes the second numeral into a superior numeral as well. A superior numeral followed by a slash changes the slash into a fraction slash, and so on. This requires a lot of ligatures (111 to be precise) but it works remarkably well.

Not surprisingly, activation of quaint ligatures also uses the same mechanism: a lowercase letter followed by a ˆ changes the letter into a superior letter, and a superior letter followed by a letter generates the quaint ligature.

I generated the fonts using a program that I wrote called `afm2ofm`. The program is a modification of the C program `afm2tfm` from the `dvips` distribution. The new program, called `afm2ofm`, differs from the original in several ways. First, it generates several 8-bit container fonts that together contain all the glyphs in the TrueType font. For each such font, the program generates an encoding file (.enc), to be used by odvips, and a tfm file, to be used both by Omega and by odvips. Because the ordering of glyphs in the 8-bit fonts is unimportant, they can be generated automatically. Second, it maps all of these 8-bit fonts into the virtual font that it generates, and the virtual font can thus contains more than 256 glyphs. The lower 256 conform to an input encoding (T1 in my case, or T1 modified to use small caps, etc.). The other glyphs in the TrueType font are mapped into free slots beginning at index 128 (this is actually the original behavior of `afm2tfm`; all I had to do here was to let continue beyond slot 255). The rest is standard: the ligatures are specified using the `LIGKERN` command in the input encoding file.

The same task can probably also be accomplished by `fontinst`.

One remarkable feature of this method is that it degrades only slightly if you insist on using 8-bit TEX rather than Omega. If you run `afm2tfm` and `vptovf` rather than `afm2ofm` and `ovp2ovf` the same input encoding files, you get perfectly usable `vf` and `tfm` files. They will include all the glyphs in the T1 encoding, including the ligatures ff, ffl, and ffi, but you will not have access to the alternates, swashes, extra ligatures, and fraction.

7 WGL4 Fonts

WGL4 is a list of glyphs defined by Microsoft, which contains all the letters that are used in European languages, including Greek and Russian. So-called WGL4 fonts are TrueType or OpenType fonts that contain all the glyphs in the list. Beyond the fact that one can use WGL4 fonts to set text in many languages, the Greek letters can sometimes be used for mathematical typesetting.

Many WGL4 fonts are available. Quite a few are freely available from Microsoft at `www.microsoft.com/typography` (Andale Mono, Arial, Arial Black, Georgia, Verdana, Comic Sans, Impact, Times New Roman, and Courier New). Others are bundled with Microsoft's various Office Products, such as Word and Publisher and/or with Windows operating systems: Franklin Gothic, Century Gothic, Monotype Garamond, Century Schoolbook, Bookman Old Style, Corsiva, Mistral Tahoma, Lucida Console, and Lucida Unicode. (Microsoft Publisher 2000 comes with a few additional latin-only rich fonts: Californian FB, Berlin Sans FB, Agency FB, and High Tower Text, Niagara, Magneto.) Figure 6 shows samples of many of these fonts.

The WGL4 list contains 652 glyphs, but many fonts contain a few less or more glyphs. Comic Sans, for example, contains only 574 glyphs, even though it is classified as a WGL4 font by Microsoft. Several fonts contain a lot more glyphs. Arial, Times New Roman, Courier New and Tahoma contain over 1200 glyphs, supporting also Arabic, Hebrew, and Thai.

Not all the Greek glyphs in these fonts are suitable for mathematical typesetting. In particular, the small alpha is often indistinguishable from a Latin small 'a', and the capital Upsilon is sometimes indistinguishable from a Latin capital 'Y'. The small beta is always different from a small Latin 'b', but it's shape is sometimes quite different from the betas that we are used to see in mathematical contexts.

In one of the WGL4 families, Georgia, the numerals are old-style, and hence not particularly suitable for modern mathematical typesetting. Version 1.0 of Verdana, however, had lining (actually semi-lining) numerals, so if you still have it, you might want to use its numerals in math mode together with glyphs from recent versions of Georgia.

Finally, the selection of Greek letters in WGL4 fonts is smaller than the selection in TEX's fonts. In particular, these fonts contain only one glyph for epsilon, theta, sigma, and phi, whereas TEX has two for each.

These deficiencies does not mean that you can't use the Greek glyphs in WGL4 fonts for mathematics, but that you might have to use a few glyphs from other fonts when you build macros to use these fonts.

Two fonts which seem to me particularly useful for mathematics are Georgia (good alpha and beta, Upsilon is Y) and Franklin Gothic, a highly readable sans serif (even better alpha and beta, Upsilon is Y). The Franklin Gothic family also includes condensed medium and demi-bold fonts, which are suitable for titles.

From a technical point of view these fonts can be treated just like the Hoefler Text fonts. Since the set of useful glyphs is smaller, however, 8-bit fonts are probably fine. To simplify the use of these fonts in math, I also produced virtual fonts with OMS and OML encoding and declared the fonts in LATEX as a new math version. This is not a complete solution, since it does not provide substitutes for glyphs missing in WGL4, but it enabled me to taste the possibilities. Figure 3 shows a bit of math set in Franklin Gothic Book.

Ideally, one should design glyphs to complement these fonts, in much the same way that Haralambous did for Baskerville [6]. Some glyphs can be borrowed from other fonts. For example, one might use the Hebrew aleph from Arial, a sans serif font, when typesetting math with Franklin Gothic say.

Many of these fonts contain extensive pair kerning data. The fonts in the Franklin Gothic family have 1000 pairs, and fonts in the Arial, Times New Roman, Garamond, families also have hundreds of pairs. The fonts in Georgia, Verdana, Century Gothic, and Monotype Corsiva, on the other hand, have no kerning pairs at all.

8 Palatino Linotype

Palatino Linotype is a rich font family bundled with Windows 2000 Professional. As such, we can expect it to become widely available in the future. The fonts are WGL4 fonts, but they also have advanced typographic features not found in most WGL4 fonts. (They also contain many precomposed glyphs for Vietnamese and Greek.) The typographic features in the fonts include three sets of numerals (fixed- and variable-width lining and fixed-width old style), numerous standard and quaint ligatures, precomposed fractions and fraction components, small capitals (and some small punctuation marks to go with them), superior letters and numerals, currency symbols, and a few mathematical symbols. Figure 2 demonstrates many of these features.

The glyphs of the Greek letters are suitable for math typesetting[3]. The alpha and Upsilon are well differentiated from the Latin a and Y, and both them and the beta are suitable as math symbols. There are also variant forms of theta, phi, sigma, and pi. There is only one variant of epsilon, but this is less important, because they two variants in Computer Modern are too similar to be used together in the same document.

According to design information contained in the font files, Herman Zapf (the designer of Palatino) "has drawn numerous additional characters to include an extensive range of ligatures, numerals, fractions, and support for both Cyrillic and both monotonic and polytonic Greek." I do not know whether Zapf designed the Greek letters with math in mind, but this is possible considering his experience in designing math fonts gained when he designed AMS Euler, based on specifications and critique from Knuth [8, 9].

The fonts have pair kerning data that is extracted by `ttfps`, but the number of pairs is not extensive, between 71 and 148 pairs per font. The kerning pairs that are present are for unaccented Latin glyphs and for punctuation. There is no kerning data for accented glyphs, non-Latin glyphs, or small caps. It is possible that there is more kerning data in the "GPOS" table of the fonts (named tables hold data structures in a TrueType font file), not in the "kern" table, which `ttfps` uses to build the `afm` file, but I have not checked that.

Technically speaking, the fonts are converted to use with TEX just like the Hoefler Text fonts.

9 Conclusion

Macro packages, especially LATEX, should support the features of rich fonts. In particular, I believe that macro packages should automatically selects the correct numerals for most situations. The enhancement to LATEX that I envision would automatically select fixed-width numerals in tables, and would automatically select lining numerals when there numerals are isolated (as in page numbers and the table of contents) or when the numeral is likely to be followed by a capital (as in section numbers). There are also cases where the LATEX style can also automatically select the best punctuation, especially parentheses, brackets, and braces. Essentially, when lining figures are enclosed alone in parentheses, the parentheses should also be lining, as in reference number in many bibTEX styles and in equation numbers.

These enhancements would be easier to implement if the font selection scheme supported these font features, which are orthogonal to series and shape, but an easier hack might be to change the encoding (say from T1 to T1Lining or T1Fixed).

Either way, the wider availability of rich fonts, such as Hoefler Text, WGL4 Fonts, Palatino Linotype, and more importantly, the immanent introduction of Adobe's rich "Pro" fonts, imply that the community should discuss this issue if TEX-based systems are to remain in the top tier of typesetting.

The Greek letters in WGL4 fonts, in particular, enable a significant improvement in the typography technical documents. For the first time it is now possible to use several font families in a document with a matching mathematical style for each family. For example, we can use Palatino for the text of the document, with a matching math style with most of

[3] Diego Puga designed an unrelated set of greek letters for mathematical typesetting with Palatino. His package, `mathpazo`, contains Type1 fonts with his greek letters and a few other symbols, as well as virtual fonts that combine his greek letters, latin letters from the Palatino fonts, and mathematical symbols from Computer Modern. The package is available on CTAN.

Arial	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	אבג
	The quick brown fox	*169*	*ΔΛΨΩαβγζξ*	*ЉЖЗИЮбвдж*	אבג
Courier	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	אבג
New	**The quick brown fox**	**169**	**ΔΛΨΩαβγζξ**	**ЉЖЗИЮбвдж**	**אבג**
	The quick brown fox	*169*	*ΔΛΨΩαβγζξ*	*ЉЖЗИЮбвдж*	אבג
Times New	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	אבג
Roman	**The quick brown fox**	**169**	**ΔΛΨΩαβγζξ**	**ЉЖЗИЮбвдж**	**אבג**
	The quick brown fox	*169*	*ΔΛΨΩαβγζξ*	*ЉЖЗИЮбвдж*	אבג
Comic Sans	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	
Andale Mono	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	
Georgia	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	
	The quick brown fox	**169**	**ΔΛΨΩαβγζξ**	**ЉЖЗИЮбвдж**	
	The quick brown fox	*169*	*ΔΛΨΩαβγζξ*	*ЉЖЗИЮбвдж*	
Verdana	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	
	The quick brown fox	*169*	*ΔΛΨΩαβγζξ*	*ЉЖЗИЮбвдж*	
Bookman	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	
Old Style	*The quick brown fox*	*169*	*ΔΛΨΩαβγζξ*	*ЉЖЗИЮбвдж*	
Century	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	
Gothic	*The quick brown fox*	*169*	*ΔΛΨΩαβγζξ*	*ЉЖЗИЮбвдж*	
Garamond	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	
	The quick brown fox	**169**	**ΔΛΨΩαβγζξ**	**ЉЖЗИЮбвдж**	
	The quick brown fox	*169*	*ΔΛΨΩαβγζξ*	*ЉЖЗИЮбвдж*	
Century	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	
Schoolbook	*The quick brown fox*	*169*	*ΔΛΨΩαβγζξ*	*ЉЖЗИЮбвдж*	
Corsiva	*The quick brown fox*	*169*	*ΔΛΨΩαβγζξ*	*ЉЖЗИЮбвдж*	
Mistral	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	
Tahoma	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	אבג
Tekton Pro	The quick brown fox	169	Small caps	Old-style numerals 169	
	The quick brown fox	**169**	**Small caps**	**Old-style numerals 169**	
	The quick brown fox	*169*	*Small caps*	*Old-style numerals 169*	
Californian FB	The quick brown fox	169	Small caps	Quaint ligatures ct	
	The quick brown fox	*169*	*Small caps*	*Quaint ligatures ct*	
High Tower	The quick brown fox	169	Small caps	Old-Style numerals 169	
Text	*The quick brown fox*	*169*			
Berlin Sans FB	The quick brown fox	169	Alts WWSS&&ggOO	Old-style numerals 169	
	The quick brown fox	**169**	**Alts WWSS&&ggOO**	**Old-style figs 169**	
Agency FB	The quick brown fox	169	Alternates AAKKMMRRVVWWXXYY		
Niagara	The quick brown fox	169	Small caps	Small numerals 169	
Magneto	*The quick brown fox*	*169*	*Alternates A A H H f f i i w w x x*		
Apple	*The quick brown fox*	*169*	*Small caps*	*Old-style numerals 169*	
Chancery			*Alts AAAAAA*	*Quaint ligatures ctThctsb*	
Lucida Sans	The quick brown fox	169	ΔΛΨΩαβγζξ	ЉЖЗИЮбвдж	אבג

Figure 6: Samples of some of the rich fonts that are mentioned in the article. The samples show numerals and latin text in all the fonts, and rich text features such as greek, cyrillic, and hebrew letters, small capitals and old-style numerals, quaint ligatures and alternate glyphs. Not all fonts in each family are shown.

the symbols taken from Palatino fonts, and Franklin Gothic for captions and titles. There are bold version of the mathematical symbols, so the math can match the text in weight, not only style. This would represent a significant improvement over today's typography, with regular-weight serifed math in bold titles and sans serif captions, for example.

Rich fonts such as Hoefler Text also raise the issue of visual editing. Glyph variations are meant to be selected visually. TEX and LATEX packages that support rich fonts, such as Alan Hoenig's package for the Adobe Poetica family, rely on a combination of plain-text markup to select glyphs and/or on automatic variant substitution, say a swash at the end of every word. Although Hoenig has achieved magnificent results with this method, it is not the most direct way of using such fonts.

A better way would be for LATEX editors, particularly LYX, to allow the user to select glyphs visually and to automatically produce the plain-text markup in the LATEX input file. LYX would use the font, say Hoefler Text, as a screen font and allow the user to select glyphs for which there are variants or to activate individual quaint ligatures. This, however, requires significant changes in LYX.

References

[1] Adobe Systems Inc., *PostScript 3 Core Font Set Overview*, Technical Note 5609, 1997. Available online at `http://www.adobe.com`. A PDF document listing samples of the fonts in the Core Set is also available online.

[2] Apple Computer Inc., *Inside Macintosh: QuickDraw GX Typography*, Addison-Wesley, 1994. Available online at `http://developer.apple.com/techpubs/mac/GXTypography/GXTypography-2.html`.

[3] Apple Computer Inc., *TrueType Reference Manual*, 1997. Available online at `http://fonts.apple.com`.

[4] The Hoefler Type Foundry, *Catalog Number 2*. No date. Available online at `http://www.typography.com`.

[5] Yannis Haralambous and John Plaice, "First applications of Omega: Adobe Poetica, Arabic, Greek, Khmer," *TUGboat* 15(3), September 1994, pages 344–352.

[6] Yannis Haralambous, *Une police pour la Societe mathematique de France: le SMF Baskerville*, Les Cahiers GUTenberg 32, 1999, pages 5–20.

[7] Alan Hoenig, "The Poetica family: fancy fonts with TEX and LATEX," *TUGboat* 16(3), September 1995, pages 244–252.

[8] Donald E. Knuth and Hermann Zapf, "AMS Euler—A New Typeface for Mathematics," *Scholarly Publishing* 20(3), April 1989, pages 131–157.

[9] David R Siegel, "The Euler Project at Stanford," Technical Report, Stanford University, 1985.

◇ Sivan Toledo
School of Computer Science,
Tel-Aviv University
Tel-Aviv 69987, Israel
`sivan@math.tau.ac.il`

Software & Tools

Even more MetaFun with METAPOST: A request for permission

Alexander Berdnikov, Hans Hagen,
Taco Hoekwater and Bogusław Jackowski

Introduction

In May 2000, at the Bachotek conference in Poland, an evening session was organized on extending METAPOST. It's out of love for this program, not out of frustration, that we discussed this topic, and Bachotek provided the right ambiance. For the record: some more people were present at the meeting than the four of us.

1 Accuracy

There are two limitations that sometimes conflict with our requirements. First, there is an upper bound of 4K bp, which can be raised to 32K by disabling a warning, but then some internal intermediate results can fail. The upper bound is not a real limitation in METAFONT, since normally (type 1) fonts are designed on a 1000 times 1000 bp grid. However, in the area where METAPOST is used, this limitation can hurt.

The second limitation concerns the accuracy. There are situations where one expects (for instance) two points to be the same, but when they are calculated in a different way, they slightly differ.

2 Unfill

Instead of implementing an unfill as “fill with white”, a real unfill operator should be available. Such a mechanism can probably be combined with the weighted paths mentioned below.

3 Picture operations

In METAPOST a picture is a collection of paths, drawn or filled, using some color and a certain pen. The full range of operations available in METAFONT should also apply to METAPOST: shift, scale, rotate, add, subtract, etc.

4 Weighted paths

Where in METAFONT a path can have a weight, in METAPOST only a color can be applied. In addition to color, a weight should be available, and METAPOST should be capable of dealing with them in a similar way. If needed, a bitmap model, like the one available in METAFONT, should be available in parallel (for calculations, etc.).

5 Intersections

Calculating the intersection points of two paths should be implemented in the core system and not in macros. In a similar way it should be possible to reduce overlapping and self-overlapping contours into elementary ones. Although difficult border cases are very hard to solve, METAPOST should be capable of handling cases of average complexity.

6 Pens

Circular and elliptical pens should be re-implemented in a different way. Currently these pens are reduced to drawing a path with a certain thickness, if needed combined with a PostScript transformation. Instead of drawing, the effective path should be a filled combination of Bezier curves. The user should be able to specify the accuracy as well as methods, such as: try to approximate the resulting contour as well as possible, keep the approximation inside the contour, or keep it outside.

7 Fonts

Although METAPOST provides a way to embed text typeset by TEX, and therefore gives access to the full TEX machinery, it would be nice if the more raw text processing would also honor ligatures and kerning. The code that is needed can be derived from other programs in the TEX suite.

8 Specials

Specials should be re-implemented and become path or point specific in order to achieve special effects.

Conclusion

We are aware of the fact that the schedule of the author of METAPOST, John Hobby, does not permit him to implement these extensions, so we are pleased to hear that he is willing to grant us permission not only to explore these extensions, but also to realize them, and that he is willing to participate in the process of extending METAPOST.

Since we know that stability as well as compatibility are big issues in the TEX community, we are very well aware how delicate the process is of merging extensions as proposed by us and John himself into the existing program. We will do our best to make sure that the burden of merging code will be minimized. And, most of all, we will try not to spoil the beauty of the existing code.

We love METAPOST too much to compete, but we long to complete.

◇ Alexander Berdnikov
Institute of Analytical
Instrumentation
St. Petersburg, Russia
berd@ianin.spb.su

◇ Hans Hagen
PRAGMA ADE
Ridderstraat 27
8061 GH Hasselt, The Netherlands
pragma@wxs.nl

◇ Taco Hoekwater
Kluwer Academic Publishers
Achterom 119
3311KB Dordrecht, The
Netherlands
taco.hoekwater@wkap.nl

◇ Bogusław Jackowski
BOP sc.
ul. Piastowska 70
80-331 Gdańsk Oliwa, Poland
b.jackowski@gust.org.pl

Extending METAPOST: Response to “Even more MetaFun”

John D. Hobby

Introduction

I agree that METAPOST could use some improvements and bug fixes, but I have been too busy to do any of this work during the last two years or so and I do not expect that situation to improve soon. It is OK with me if others want to make upward-compatible improvements to `mp.web`, but such implementors should know enough about the program to be able to go through the “The command codes” section of `mp.web` and be familiar with everything it refers to. To avoid competing versions, extensions should probably be sent to me (`hobby@research.bell-labs.com`) for inclusion in the master version.

9 Accuracy

It would improve the METAPOST language to switch from 32-bit scaled integers to 64-bit floating point but this would be a lot of work and it would have to be done very carefully. The scale factors of 2^{16} and 2^{28} could be retained, but it would still be necessary to go through the entire program changing numerical constants and looking for places where the new arithmetic violates hidden assumptions. The code for solving linear equations is especially likely to need attention.

Another concern is how to ensure that METAPOST continues to give the same numerical results in all implementations. It is may be safe to assume 64-bit IEEE standard floating point is now available on all systems, but some compilers use a higher precision for intermediate results and it would be necessary to defeat that somehow. There is also the concern that `mp.web` is technically a Pascal program and not all installations are based on Pascal-to-C translation.

I considered all these issues when I implemented the graph package, and I decided that it was a lot easier to design the graph macros around METAPOST’s `mlog` and `mexp` operators.

10 Unfill and weighted paths

The current implementation keeps track of the painting order for the basic components of a picture but does not try determine exactly which parts overlap. Unfill may be easier than allowing weighted paths with add and subtract, but implementing all these features together would require an ability to keep track of a picture as a spline-bounded planar subdivision. This would require sophisticated algorithms and data structures, and geometric algorithms of this type are famous for being extremely difficult to implement robustly. Also, it is not clear how to make fonts fit into this model.

It would be very tricky to make the new features upward compatible. In “Drawing Graphs with METAPOST” (CSTR 164), I describe a `for` ... `within` iteration that allows one to pick apart a picture and see all its basic components in the painting order. This is fundamental to the `graph.mp` macros and would be very hard to reconcile with new features such as add, subtract and cull.

11 Bitmaps

The request for a bitmap model like the one available in METAFONT is quite natural, provided that it is somehow generalized to handle color. A basic design would not be too hard to implement unless there is to be an operator that takes a picture and converts it into a bitmap as the PostScript interpreter would.

12 Intersections

There could certainly be a primitive that attempts to find all the intersections between two paths instead of just a single intersection point. If the operator is to take two paths, intersect their interiors, and return a path that describes this region, paths would have to be generalized to allow multiple noncontiguous outlines. One way to do this would involve a new type of path join operator that behaves syntactically like `--` or `...`, takes one unit of “time” along the path, and means that there is a gap in the path. It would also be necessary to generalize `fill` to accept non-closed paths.

13 Pens

The main advantages of the current treatment of circular and elliptical pens are that it is robust and generates simple PostScript. The main disadvantage I am aware of is that elliptical pens interact poorly with dashed lines. The present scheme also makes it hard to convert METAPOST output into a Type 1 font, but that is not METAPOST’s intended application.

I think it would be a mistake to make all pen operations outline-based. Outline-based pens could be provided as an alternative, but this would not be easy to implement. Polygonal pens are easier to express in terms of outlines, but the implementation was a lot of work and it still has known bugs.

14 Specials

The METAPOST language already has a very simple `special` mechanism and any improvements will have to be upward compatible. Perhaps a different key word is needed.

15 Fonts

I agree that it would be nice to have METAPOST's primitives handle ligatures and kerning.

16 Other features

There has been some demand for other features that would not be too hard to implement.

- Make dash patterns work with polygonal pens.
- Add operators to determine whether a point is "inside" a path or what its winding number is with respect to the path. Is some point "painted" by a given picture, and if so what color?
- Add an operator for the area "inside" a path and the total area affected by a picture.
- Allow `shipout` to an arbitrary file, not just the job name with a numeric extension.
- Have a `settex` command that takes a string expression and uses it to decide what version of TEXto use for `btex` ... `etex`. It would have to occur before the first `btex` and it should be restrictive enough not to be a security risk.

Conclusion

Some of the new features I have discussed would be easy to implement, and many of the others would be nice to have if someone can find the time to implement them properly. However, I am inclined to think that it would be a mistake to do outline-based versions of elliptical pens or to add weighted paths with add and subtract. Care should be taken to make any additional features work reliably and fit in with the rest of the language.

◇ John D. Hobby
Bell Laboratories
Room 2C-458
700 Mountain Ave.
Murray Hill, NJ 07974-0636
`hobby@research.bell-labs.com`

Hyphenation Exception Log

Barbara Beeton

This is the periodic update of the list of words that TEX fails to hyphenate properly. The list last appeared in full in *TUGboat* 16, no. 1, starting on page 12, with an update in *TUGboat* 20, no. 1, pages 50–51. The present list contains only new items reported since then.

A copy of this article with the complete list can be found on the TUG Web pages, via a link in the on-line Table of Contents.

Owing to the length of the complete list, it has been subdivided into two parts: English words, and names and non-English words that occur in English texts. This update follows that division.

This list is specific to the hyphenation patterns that appear in the original `hyphen.tex`, that is, the patterns for U.S. English. If such information for other patterns becomes available, consideration will be given to listing that too. (See below, “Hyphenation for languages other than English”.)

In the list below, the first column gives results from TEX's `\showhyphens{...}`; entries in the second column are suitable for inclusion in a `\hyphenation{...}` list.

In most instances, inflected forms are not shown for nouns and verbs; note that all forms must be specified in a `\hyphenation{...}` list if they occur in your document.

Thanks to all who have submitted entries to the list. Since some suggestions demonstrated a lack of familiarity with the rules of the hyphenation algorithm, here is a short reminder of the relevant idiosyncrasies. Hyphens will not be inserted before the number of letters specified by `\lefthyphenmin`, nor after the number of letters specified by `\righthyphenmin`. For U.S. English, `\lefthyphenmin=2` and `\righthyphenmin=3`; thus no word shorter than five letters will be hyphenated. (For the details, see *The TEXbook*, page 454. For a digression on other views of hyphenation rules, see below under “English hyphenation”.) This particular rule is violated in some of the words listed; however, if a word is hyphenated correctly by TEX except for “missing” hyphens at the beginning or end, it has not been included here.

Some other permissible hyphens have been omitted for reasons of style or clarity. While this is at least partly a matter of personal taste, an author should think of the reader when deciding whether or not to permit just one more break-point in some obscure or confusing word. There really are times when a bit of rewriting is preferable.

One other warning: Some words can be more than one part of speech, depending on context, and have different hyphenations; for example, ‘analyses’ can be either a verb or a plural noun. If such a word appears in this list, hyphens are shown only for the portions of the word that would be hyphenated the same regardless of usage. These words are marked with a ‘*’; additional hyphenation points, if needed in your document, should be inserted with discretionary hyphens.

The reference used to check these hyphenations is *Webster’s Third New International Dictionary*, Unabridged.

English hyphenation

It has been pointed out to me that the hyphenation rules of British English are based on the etymology of the words being hyphenated as opposed to the “syllabic” principles used in the U.S. Furthermore, in the U.K., it is considered bad style to hyphenate a word after only two letters. In order to make TEX defer hyphenation until after three initial letters, set `\lefthyphenmin=3`.

Of course, British hyphenation patterns should be used as well. A set of patterns for UK English has been created by Dominik Wujastyk and Graham Toal, using Frank Liang’s `PATGEN` and based on a file of 114925 British-hyphenated words generously made available to Dominik by Oxford University Press. (This list of words and the hyphenation break points in the words are copyright to the OUP and may not be redistributed.) The file of hyphenation patterns may be freely distributed; it is posted on CTAN in the file `tex-archive/language/hyphenation/ukhyph.tex` and can be retrieved by anonymous FTP or using a Web browser.

Hyphenation for languages other than English

Patterns now exist for many languages other than English, including languages using accented alphabets. CTAN holds an extensive collection of patterns in `tex-archive/language/hyphenation` and its subdirectories.

The List — English words

All entries in the list below have been reported since a supplement to the list was last published in 1999.

`ac-cusative`	ac-cu-sa-tive
`cryp-togram(s)`	cryp-to-gram(s)
`cuneiform`	cu-nei-form
`demisemi-qua-ver`	demi-semi-qua-ver
`dyslexia`	dys-lexia
`hemidemisemi-qua-ver`	hemi-demi-semi-qua-ver
`ir-re-ducible`	ir-re-duc-ible
`province(s)`	prov-ince(s)
`provin-cial`	pro-vin-cial
`re-ducible`	re-duc-ible
`reusable`	re-us-able
`reuse`	re-use
`salta-tion`	sal-ta-tion
`semidi-rect`	semi-di-rect
`semisim-ple`	semi-sim-ple
`sesquipedalian`	ses-qui-pe-da-lian
`straphanger`	strap-hanger
`sub-d-if-fer-en-tial`	sub-dif-fer-en-tial

Names and non-English words used in English text

`Be-mbo`	Bembo
`Wirtschaft`	Wirt-schaft

◇ Barbara Beeton

`bnb@Math.AMS.com`

Hints & Tricks

"Hey — it works!"

Jeremy Gibbons

Welcome again to *"Hey — it works!"*, a column devoted to useful or surprising (LA)TEX and META techniques. I am writing this time from TUG2000, ably organized by Sebastian Rahtz and Kim Bruce in my home town of Oxford (from which you may conclude that my timekeeping has not made our noble editor's life any easier — sorry, Barbara!).

In this issue we have three items: one describing a macro of unknown provenance for yielding a non-punctuating comma for a decimal point, a slightly related one inspired by André Van Ryckeghem from Belgium on currency conversion, and one by Pedro

Palao Gostanza from Madrid on counting the number of parameter uses in the expansion text of a macro.

◇ Jeremy Gibbons
Oxford University Computing Laboratory
Wolfson Building, Parks Road
Oxford OX1 3QD, UK
jeremy.gibbons@comlab.ox.ac.uk
http://www.comlab.ox.ac.uk/oucl/people/jeremy.gibbons.html

1 Decimal comma

In some languages, a comma is used instead of a full stop to separate the whole and fractional parts of a decimal number: 3,1415. Unless one does something special about it, TEX prints instead $3, 1415$, treating the comma as punctuation rather than an ordinary symbol and so putting some space after it.

Someone recently posted the following style file to comp.text.tex[1]:

```
\mathchardef\ocomma="013B
\mathchardef\pcomma="613B
\mathcode`\,="8000
{\catcode`\,=\active
 \gdef,{\obeyspaces
   \futurelet\next\smartcomma}}
\def\smartcomma{\if\space\next
  \pcomma\else\ocomma\fi}
```

This style file defines two different macros to generate a comma, one as an ordinary symbol and one as a punctuation. The comma is then made active in maths mode, and defined to look ahead to the next token; if this next token is a space, a punctuation comma is used, and otherwise the ordinary comma is used. Thus, `3,1415` yields the decimal number 3,1415, whereas `(3, 4)` yields the pair $(3, 4)$.

It strikes me that it would be better to determine whether the next token is a digit, and only to use the ordinary comma when it is; then it does not matter whether a space is used in, say, (x, y). But with a little experimentation I could not make this work; can anyone help?

2 Two decimal digits

On comp.text.tex recently, André Van Ryckeghem asked how to perform currency conversion. He already had the translation working on integer values, with a macro like the following:

```
\def\convert#1#2{{% evaluate #1/#2
  \count0=#1\relax % numerator
  \count1=#2\relax % denominator
  \count2=\count1 \divide\count2 by 2\relax
  \advance\count0 by \count2\relax
  \divide\count0 by \count1\relax
  %
  \the\count0\relax
}}
```

The main point here is how to divide one value m by another n and round the result to the *nearest* integer, instead of the usual rounding down to the *next smaller* integer; this is done by computing instead $(m+n/2)/n$ and rounding down as usual. For example, at the time of writing, the exchange rate is ¥162 to £1, so with

```
\convert{34567}{162}
```

we learn that ¥34567 is approximately £213.

André's problem was to incorporate decimal output. For example, as it stands we learn only that ¥345 is approximately £2, which is not very accurate. On the other hand, real number arithmetic is inappropriate, because we do not want umpteen digits after the decimal point (£2.12962). Instead, we define

```
\def\convert#1#2{{% evaluate #1/#2, to 2dp
  \count0=#1\relax
  \multiply\count0 by 100\relax % 100*numerator
  \count1=#2\relax % denominator
  \count2=\count1\relax
  \divide\count2 by 2\relax % half denominator
  \advance\count0 by \count2\relax
  \divide\count0 by \count1\relax % rounded divn
  %
  \count3=\count0\relax
  \divide\count3 by 100\relax % whole part
  \count4=\count3\relax
  \multiply\count4 by -100\relax
  \advance\count4 by \count0\relax % frac part
  %
  \the\count3.%
  \ifnum \count4<10\relax 0\fi
  \the\count4\relax
}}
```

The difference is that we work throughout with two more significant figures (that is, we multiply the numerator by 100). To print the result, we finally divide by 100 again to print the whole part, then print the remainder (possibly with a leading zero, if it is just a single digit) to get the fractional part. With this definition, we find that ¥345 is more accurately £2.13.

Note that all arithmetic is performed with TEX's 32-bit integers, so is limited to about nine digits. There are strong arguments for using arbitrary precision integers for financial computations. André observes that a much more elaborate and robust

[1] Unfortunately, the identity of the original author has been lost. If it was you, let me know and I'll give an update in my next column. The original was called komma.sty, dated 1998/06/28, and had some @ symbols in the macro names.

solution than this is provided by Melchior Franz' `euro` package, which performs and typesets conversions between arbitrary currencies, using Michael Mehlich's `fp` package for exact arithmetic with 36 significant digits (both packages being available on CTAN).

3 Number of parameter tokens

While doing some meta-macros I was in need to count how many parameter tokens appear in a definition parameter text. A complete solution would need the capability of TeX to match against a `{`.

It is easy to work with parameter texts if they are stored in *saturated* macros: macros with nine undelimited parameter tokens. The three following saturated macros containing parameter text will be used as a running example.

```
\def\pp#1#2#3#4#5#6#7#8#9{%
  #1trivial#2parameter#3}
\def\qq#1#2#3#4#5#6#7#8#9{%
  #1\undefined#2parameter#3}
\def\kk#1#2#3#4#5#6#7#8#9{%
  #1problem#2\gobbledisttag#3}
```

In the rest of this note, *parameter text* usually means parameter text stored in a saturated macro. The goal is to define a macro `\nop` returning in a counter the number of parameter tokens in a parameter text; the counter and the parameter text are, in this order, the only arguments of `\nop`.

The main idea is simple: substitute each parameter token for a counting code like

```
\advance\counta by 1
```

The following macro allows us to put the same thing in each parameter token

```
\def\applyall#1#2{#1%
  {#2}{#2}{#2}{#2}{#2}{#2}{#2}{#2}{#2}}
```

The difficult part is to throw away all the material between the parameter tokens; we will call it *disturbing* material. My first solution was really simple and worked almost always.

```
\def\nop#1#2{#1=0
  \expandafter\expandafter\expandafter
    \gobbledist\applyall#2%
    {\gobbledisttag\advance#1by1
     \gobbledist}%
    \gobbledisttag}
\def\gobbledist#1\gobbledisttag{}
```

But this fails if `\gobbledisttag` appears in the parameter text; that is, `\pp` and `\qq` parameter texts are counted without any problem, but `\kk` raises an error because of a dangling `\gobbledisttag`. Although this is not a practical restriction, it is an aesthetic one. To cope with it we need to mark the beginning of a parameter token with something that cannot be used in parameter text: outer macros, `{`, `}` or `#`. But if the mark cannot be used in a parameter text, neither can it appear in the definition of `\gobbledist` (1). So, a trick different from a *gobble* macro is needed to throw away the material between parameter tokens (2).

Expanded definitions (`\edef`) are a good place to look next. In cooperation with brace hacks from the TeXbook, allow us to put disturbing material inside braces.

```
\def\nopB#1#2{#1=0
  \expandafter\expandafter\expandafter
    \voidefdist\applyall#2%
    {\voidefdistend\advance#1by1
     \voidefdist}%
    \voidefdistend}
\def\voidefdist{\edef\aux{\iffalse}\fi}
\def\voidefdistend{\iffalse{\else}\fi}
```

Disturbing material ends in the definition of the `\aux` macro. But this solution is even worse because TeX does not allow undefined token in an expanded definition; both `\qq` and `\kk` give errors because neither `\undefined` nor `\gobbledisttag` are defined. If the parameter text has conditionals, errors can happen far beyond `\nop`.

Neither normal definitions, nor token lists help in this problem, because they need balanced braces; they surely solve it completely if combined with the `\scantoken` extension of ϵ-TeX. Boxes definitions are worse than expanded definitions.

It took me some time to realize that, in TeXory, (2) cannot be derived from (1) because TeX allows to match against tokens that cannot appear in a parameter text: if a parameter text ends with `#`, TeX will match against an open brace (*The TeXbook*, p. 204). This observation turned me back to search for a *gobble*-like solution.

```
\def\nop#1#2{#1=0
  \expandafter\expandafter\expandafter
    \gobbledist\applyall#2%
    {{}\advance#1by1 \gobbledist}%
    {}}
\def\gobbledist#1#{\gobble}
\def\gobble#1{}
```

This solution is a bit strange because usual TeX practice dictates that, in order to catch something, it should be surrounded with `{`···`}` but, in some sense, we are surrounding the disturbing material with `}`···`{`.

Although the last definition of `\nop` is aesthetically more pleasant than the first one, it has its same drawback: `\nop` cannot count how many parameter tokens has `\gobbledist`! I am sure that the reader will find a nice fix.

◇ Pedro Palao Gostanza
Universidad Complutense de Madrid, Spain
`ecceso@sip.ucm.es`

The Treasure Chest

Packages posted to CTAN

The first five months of 2000 have been amazingly productive in terms of new and updated packages going onto CTAN: 182 are listed here, compared with 59 for the last quarter of 1999.[1] If this keeps up, we'll have to move the inventories into some other place and reserve this column just for package tours!

And if you're really *really* interested in stats . . .

Jan	Feb	Mar	Apr	May	June
23	25	46	19	38	31

. . . the above table shows how many entries for each month appear in this inventory, keeping in mind that each inventory list (should) only count a package once, regardless of how many upgrades are posted to CTAN within the period in question.

Notes on inventory format

1. Packages are listed in the month of the latest upgrade; that is, if a version is posted in Oct. and a new version in Dec., only the newer one will be listed.
2. Monthly entries are in alpha-order, for easier reference.
3. Unless otherwise stated, packages are located in:

   `macros/latex/contrib/supported/`

   However, there are two main branches under `contrib/`: `supported/` and `other/`. The default used here is `supported`, and where `other` is the location, its path will simply be noted as `.../other/`, to cut down on path length.[2]

January 2000

`ae`: in `fonts/ae`
This package is a set of virtual fonts for creating PDF files with T1-encoded CMR fonts.

`aifont`: in `fonts/cm/ai`
Virtual fonts and other related files for remapping the BSR/Y&Y/AMS Type 1 Computer Modern fonts. This helps to produce more robust PDF output from pdfTeX, dvips/gs, and dvipdf (upgrade to v. 1.2).

`bibtopic`: A package for multiple bibliographies. The most significant change is added compatibility for `natbib` v. 7.0.

`circuit_macros`: in `graphics/`
A set of macros for drawing high-quality electric circuit diagrams containing fundamental elements, amplifiers, transistors, and basic logic gates to include in TeX, LaTeX, or similar documents. Some tools and examples for other types of diagrams also included.

`CWEBbin`: in `web/c_cpp/`
A set of change files (to be applied with the TIE processor) that make the original sources usable with ANSI-C/C++ compilers on UNIX/Linux, MS Windows, and Amiga.

`devnag2`: in `language/devanagari/`
V. 2.0 upgrade for preprocessor, package, and fonts for Devanagari.

`dtk`: in `dante/dtk/`
The directory contains the complete list of publications of *Die TeXnische Komoedie* (nos. 0/89 to 4/99), sometimes including abstracts and keywords. It is also the base for the Web search service under `www.dante.de/dante/DTK/inhalt.html` (in German).

`dviincl`: in `graphics/metapost/`
A tiny package for including a `.dvi` page into the `.eps` files generated by METAPOST.

`epmtfe`: in `systems/os2/`
Upgrade.

`epsincl`: in `graphics/metapost/`
The package facilitates including `.eps` files in METAPOST documents; it makes use of (G)AWK.

`eulervm`: in `fonts/`
This is a set of *virtual* math fonts, based on Euler and CM. Included is a LaTeX package, which makes them easy to use, particularly in conjunction with Type1 PostScript text fonts.

`extsizes`: in `.../other/`
Upgrades to class files.

`HLaTeX`: in `fonts/korean/`
Upgrade to packages and Korean fonts.

`koma-script`: Reimplementation of the LaTeX classes (`article`, `report`, `book`, `letter`), "implementing European rules of typography and paper formats as documented in Tschichold (*Selected Papers on Book Design and Typography*).

`ltt`: in `info/`
This is the code for the examples in the book *LaTeX Tips und Tricks* (ISBN 3-932588-37-1). The code includes application examples as well as packages and class files developed in the book.

`multibib`: The package allows the user to create references to multiple bibliographies within one document, thus providing a complementary functionality to packages such as `bibunits` or `chapterbib`, which make it possible to create one bibliography for multiple yet different parts of the document.

`ochem`: in `support/`
Typesetting chemical reactions and formulae.

[1] See *TUGboat* 20,4 (1999), pages 370–374.

[2] If anyone has a suggestion on a better shorthand, please send it in. –Ch.

references: in support/
Bibliographic software supporting preparation of scientific manuscripts, storage of bibliographic data of articles in periodicals, books and articles in books. Supports BibTeX, commercial word processors; also allows import of bibliographic data from material downloaded in MEDLINE format.

ttf-tetex: in info/TrueType/
A description on using TrueType fonts with teTeX, available in .html, .pdf and .ps file formats.

wordcount: The package provides a relatively easy way of estimating the number of words in a LaTeX document.

xmlcatalogue: in macros/xmltex/contrib/
Documentation for the xmlcatalogue packages.

xmltex: in macros/xmltex/
xmltex is a system for typesetting XML files with TeX. It may be used on its own or in conjunction with another TeX format (LaTeX is assumed for most examples).

xtrcode: in support/
A small Perl script for extraction of environment contents.

February 2000

bakoma: in systems/win32/bakoma/dst/
CM Bright Fonts for BaKoMa TeX.

CMacTeX: in nonfree/systems/mac/
V. 3.4 upgrade of CMacTeX.

cmbright: in fonts/
Updates the CM Bright typefaces to v. 1.0f; all CM Bright and CM Typewriter Light fonts will provide a correct Euro symbol now.

concrete: in systems/win32/bakoma/dst/
Concrete Fonts for BaKoMa TeX.

crop: The package provides different forms of cropmarks for trimming paper stacks, for camera alignment, and for visualizing the page dimensions, as well as options for reflecting and inverting the whole document.

fancyhdrBoxed: in support/
Provides easy and fast creation of page headers for the documentation of software engineering projects, using a Mini language implemented in Python that generates PSTricks code embedded in fancyhdr headers.

hc: Replacement for the LaTeX default classes, based on the KoMa-Script bundle and the seminar class.

hypht1.tex: in language/hyphenation/
Additional hyphenation patterns to be loaded in conjunction with each "normal" set of patterns.

impose: in support/
A set of PostScript utilities for two-up printing of DSC-compliant PostScript (including that from Netscape, dvips, and FrameMaker). Requires Ghostscript and psutils.

iso: The iso class is for typesetting ISO documents according to "ISO/IEC Directives, Part 3" (3rd ed., 1997). Configuration files for TeX4ht also provided.

isorot: A package for rotating of document elements (a combination of the lscape package and an extension of the rotating package. Designed for use with both the iso class and any normal class.

iso10303: Step packages for typesetting ISO 10303 International Standard documents. Requires the iso class.

makeglos: A package to include a glossary in a document. The glossary must be prepared by an external program such as xindy or makeindex.

play: A package and class for typesetting plays and quotations from plays.

pssplit: in support/
C program and DOS executables to print selected pages from a PostScript file.

pstoedit: in support/
Converts PostScript and PDF files to other vector graphic formats so that they can be edited graphically.

ps2eps: in support/
A tool to produce EPS/EPSF files from usual one-paged PostScript documents. Requirements: Perl, ghostscript and an ANSI-C compiler if your platform is not Linux, Solaris, Digital UNIX or Windows 2000/9x/NT (binaries included).

refcheck: This package is intended to check references. It looks for numbered but unlabelled equations; for labels not used in the text; for unused bibliography references. As well, it displays \label and \bibitem keys in text margins.

syntax: in .../other/
Package to create syntax diagrams, using special environments and commands. Requires fancybox package; documentation in German.

tocloft: The package provides control over the typography of the Table of Contents, List of Figures, and List of Tables.

webguide: in info/
"A Brief Guide to LaTeX Tools for Web Publishing."

WinShell: in systems/win32/
Upgrade to WinShell, a graphical user interface for easy working with TeX. It is *not* a TeX system.

wsuipa2tipa: in support/
C program files (incl. Flex source, DOS executable) to translate a LaTeX file with wsuipa font commands into a file with TIPA font commands.

xmlplay: in macros/xmltex/contrib/
An xmltex package for typsetting the plays of Shakespeare, as marked up by Jon Bosak.

xtab: Automatically breaks long tables so that they can span multiple pages. An extension of the supertabular package.

March 2000

appendix: Provides additional appendexing capabilities. New in this release is support for per-chapter (or section for non-chaptered documents) appendices.

captdef: in `.../other/misc/`
This small package permits you to have things with 'proper' captions in the body of your text, without using explicit floats.

ccaption: Provides: continuation captions, unnumbered captions and legends, captions outside float environments, bilingual captions, etc. It also enables the definition of new float environments and their captions. New in this release is more flexible positioning of subfigure (continuation) captions.

contour: Generates a colored contour around a given text in order to enable printing text over a background without the need of a color box around the text.

custom-bib: A package to generate customized BibTeX bibliography styles from a generic file by means of the `docstrip` program.

cwbl-spanish: in `.../supported/cweb/contrib/`
A package for translating the cweb class strings into Spanish.

cweb-hy: in `.../supported/cweb/contrib/`
To allow a file using this class to be included in other documents (e.g., as an appendix to some report (the file `nodoc.tex` may be of help). Also makes it possible to insert hyperlinks from the places where a chunk of code is referenced to the chunk itself, including the indices produced at the end of a `cweb` document. An evolution of the `cweb` class, originally by Joachim Schrod.

dinbrief: This document class implements a document layout for writing letters according to the rules of the German standardization institute, DIN.

dvipsconfig: in `dviware/`
A set of PostScript header files for dvips to control various printer functions such as paper size, duplex, and paper source (e.g., manual feeder, envelope feeder, and trays 1, 2, and 3).

emtextds: in `systems/os2/emtex-contrib/`
Upgrade of the emtex/TDS distribution for OS/2.

english: in `systems/win32/winedt/dict/`
Upgrade to English dictionary.

fancynum: A package for typesetting numbers, in particular those numbers written by computers.

faq: in `usergrps/uktug/`
New but interim version of the FAQ by the UK TeX Users Group.

flashcard: in `.../other/misc/`
A class file for typesetting flash cards: a question on one side and the answer on the other.

flashcards: A class for typesetting two-sided cards, particularly for use as flash cards. [**Note:** *Not* the same as `flashcard`.]

float: This package improves the interface for defining floating objects, such as figures and tables. It adds the notion of a 'float style', which governs the appearance of floats.

fnts: in `macros/plain/contrib/`
Provides font selection for plain TeX.

games: in `systems/win32/bakoma/contrib/`
This module includes popular macro packages described in Chaps. 7 and 8 of the *LaTeX Graphics Companion*.

hyperref: A maintenance release.

ifmtarg: in `.../misc/`
Provides provides "if-then-else" and "if-else" commands for an empty macro argument.

limap: in `.../other/gene/`
Provides a style and class for typesetting maps and blocks according to the Information Mapping method.

listbib: Generates listings of bibliographic databases in BibTeX format (indented for archival purposes). Included is a `listbib.bst` that is better suited for this purpose than the standard styles.

MHequ: in `.../other/`
A package to create easily multicolumn equation environments, and to tag the equations therein.

midpage: in `.../misc/`
Provides an environment for vertical centering (corresponding to the `center` environment for horizontal centering).

mparhack: Implements a workaround for the LaTeX bug that `\marginpar`s will sometimes come out at the wrong margin.

needspace: in `.../misc/`
Provides a command to disable page breaking within a given vertical space. If there is not enough space between the command and the bottom of the page, a new page will be started.

newlfm: This class integrates the `letter` class with `fancyhdr` and `geometry`.

nextpage: in `.../misc/`
Provides additional `\clearto...page` and `\moveto ...page` commands, which are generalizations of the standard `\clear...page` and `\newpage` commands.

nomencl: A package to generate and format a nomenclature using MakeIndex.

oztex-german: in `nonfree/systems/mac/`
System upgrade.

psnfss-beta: in `macros/latex/required/`
Upgrade of the new `psnfss`, v. 8.1, and it is hopefully the final thing, except for a few parts of the documentation (unless any new bugs are detected). See the file `changes.txt` for the latest news.

rtf2latex2e: in `support/`
An RTF-to-LaTeX translator, it can handle the latest RTF versions from MS Word 97/98/2000, StarOffice, and other word processors. It has support

for tables, figures, and to some extent equations. It runs on Macintosh, Linux, UNIX, and Windows.

`semaphore:` in `fonts/`
Semaphore alphabet font in METAFONT (3 variants, each with 4 shapes).

`setspace:` This style option provides commands and environments for doing double and one-and-a-half spacing based on point size.

`sf298:` A package for generating a completed standard form 298 as prescribed by ANSI standard Z39.18 for report documentation (e.g, on a U.S. government contract).

`sidecap:` The environments to make it easy to typeset table and figure captions sideways.

`smflatex:` A set of classes, packages, and BibTeX styles for the Société mathématique de France (SMF) publications.

`svjour:` in `nonfree/.../springer/`
Springer-Verlag journal macros.

`TeXmacs:` in `systems/unix/`
A text editor (beta release) inspired by the popular TeX typesetting system and the `emacs` editor. Runs on PCs under Linux and on SUN computers: "It is reasonable to expect that it will run on most UNIX/X-Windows systems in the near future."

`tocbibind:` A package to add document elements like a bibliography or an index to the Table of Contents. The "List of ..." headings can also be put into the ToC.

`twoup:` Provides the necessary LaTeX preliminaries needed to use PostScript tools (as found in MiKTeX and many other TeX implementations) to print booklet and two-up output (i.e., two logical pages printed side-by-side on one side of one sheet of paper).

`utf2any:` in `support/`
Translates a file encoded in UTF-7 or UTF-8 (Unicode) into any 7- or 8-bit text format. Currently, mapping tables are supplied for LaTeX, HTML, ISO-8859-1, and iso-8859-15.

`vmargin:` Allows margin settings to take into account the paper size. All new is that it is now in `docstrip` format so the documentation should be much better. No real code changes.

`WinLatex:` in `systems/win32/`
A front end for TeX (Win98/NT; needs Microsoft VisualBasic DLLs.

`wp2latex:` in `support/`
Conversion from Wordperfect 6–8 into LaTeX.

`yplan:` in `.../other/`
A Perl version of Dick Nickalls' `yplan`, which generates annual calendars for any year after 1999.

April 2000

`anonchap:` in `.../misc/`
Provides commands to make `\chapter`s be typeset like `\section`s (i.e., like numbered `\chapter*`s).

`br:` in `systems/win32/winedt/dict/`
This contribution consists of a much demanded Brazilian/Portuguese word list for WinEdt. A link to this (and all other available dictionaries for WinEdt) is available on `www.winedt.com`.

`breve.ps:` in `info/lshort/portuguese/`
"Breve Introducao ao LaTeX 2_ε" is a tutorial in Portuguese about LaTeX 2_ε. It has about 80 pages organized into 5 chapters and 4 appendices.

`dinat:` in `biblio/bibtex/contrib/german/dinat/`
Bibliography style files intended for texts in German. They draw up bibliographies in accordance with the German DIN 1505, parts 2 and 3. For more information see the included documentation.

`draftcopy:` This package is used to print on some pages the word 'DRAFT' (or the language-dependent analogo) 'behind' the intended stuff (upgrade to v. 2.13).

`fncychap.patch:` in `.../fncychap/`
A small patch to `fncychap.sty` that corrects a bug in the treatment of `\chapter*` in the package's Glenn style of chapter headings.

`hyphenation-greek:` in `language/greek/package-babel/hyphenation/filippou/`
These files contain the hyphenation patterns for ancient and modern Greek in polytonic (multi-accent) and monotonic (uni-accent) systems. The patterns will work perfectly with the `greek` option of `babel` or Dryllerakis' GreeKTeX. For any other Greek package, the patterns will have to be re-coded. A brief description of the files is given in the file `readme.txt`. [CTAN note: the files are located in a new subdirectory to distinguish them from the existing file `grhyph.tex`]

`isi2bib:` in `biblio/bibtex/utils/isi2bib/`
`isi2bib` converts an ISI (Institute for Scientific Information) database file to a BibTeX file. This script replaces `bids.to.bibtex`.

`latexdraw:` in `nonfree/support/`
An X-Window-based and CAD-orientated drawing program which generates LaTeX based on PSTricks.

`minutes:` A package for setting minutes and building collections of minutes.

`pmcstex.:` in `systems/os2/pmcstex/`
pmCSTeX for EPM is a macro package for the OS/2 editor EPM (Enhanced Programmers Editor, version 6.03b or higher) (package upgrade: v. 16.4.2000 (April 16, 2000).

`simplified-latex:` in `info/`
Revision of Harvey Greenberg book, *A Simplified Introduction to LaTeX*.

`TeXshade:` A LaTeX package for setting nucleotide and peptide alignments (package upgrade).

`TeXtopo:` A LaTeX package for setting shaded and annotated membrane protein topology plots and helical wheels.

`thumbpdf:` in `support/`
Upgrade to v. 2.4. The package provides support

for thumbnails with pdfTeX, and plain/LaTeX formats. Requirements: Perl5, Ghostscript, pdfTeX.

`ttf-tetex:` in `info/TrueType/`
Upgrade.

`version.sty:` in `.../other/misc/`
Formerly listed in `latex209`, this package still works in LaTeX 2_ε and has been moved as noted.

`winedt:` in `systems/win32/`
Official release of the new version, now called WinEdt 5. See also: `www.winedt.com`.

`xtab:` Upgrade to v. 2.3.

May 2000

`acmconf:` Upgrade to v. 1.3. This class may be used to typeset articles to be published in the proceedings of ACM (Association for Computing Machinery) conferences and workshops.

`acronym:` Upgrade to v. 1.6. Added the smaller option and the macros `\acsfont` and `\acfsfont` used to control the appearance of `\acs` and `\acf`.

`aeguill:` Upgrade to v. 0.99. This file gives French guillemets built with the Polish CMR fonts (default), WNCYR fonts, the LASY fonts or with the EC fonts. This is useful in conjunction with the `ae` package (this package loads the `ae` package in case it has not been loaded) and with or without the `french` package.

`akletter:` Upgrade to v. 1.5h. This class implements a fixed layout for the first page of a letter, with with more or less variable contents, to be specified by the user.

`amscls:` Upgrade patches: the most serious bug fixed in these patches is incorrect theorem spacing (too small) in the `amsbook`/`amsart`/`amsproc` document classes.

`antp:` in `fonts/psfonts/polish/`
A replica of the Antykwa Półtawskiego font in PostScript Type 1 format — preliminary version. This font was designed in the 1920s and '30 by the Polish graphic artist and typographer Adam Półtawski and was widely used by Polish printing houses as long as metal types were in use (until ca. the '60s).

`bakoma:` in `systems/win32/`
Upgrade to v. 2.30 of the BaKoMa TeX system. For information about changes see: `dst/changes.html` in the same directory.

`collection:` The `collection` class lets you bundle individual articles into a single document. Many options have been provided and the documentation is now provided.

`dvi2bitmap:` in `nonfree/dviware/`
Upgrade to v. 0.7. `dvi2bitmap` is a utility to convert `.dvi` files directly to bitmaps, without going through the complicated route of conversion via PostScript and PNM. It can generate XBM and GIF bitmaps, plus PNG, if you have the *libpng* library installed. The program is written in C++. [CTAN note: Users should be aware that, apart from the 'nocommercial' terms imposed by the developer's employers, there are issues surrounding the use of patented technology in producing GIF output.]

`epstopdf:` in `support/`
Upgrade to v. 2.26 (bugfix). Converts `.eps` files to encapsulated `.pdf` files. It is written in C and does not require Perl; however, it does require Ghostscript. A compiled version for Windows95/98/NT is included. Since the sources are included, it can be ported to any system having a C-complier.

`eso-pic:` in `.../ms/contrib/`
This package makes it easy to add some picture commands to every page, including compatibility with `html.sty` and a new example (background picture).

`fixbbl.py:` in `biblio/bibtex/utils/fixbbl/`
A python script to deal with the well-known BibTeX bug whereby it inserts a '`%`' in a url to cause the bibliography line to fit into its minuscule vision of what is an acceptable input line for TeX.

`isi2bibtex:` in `biblio/bibtex/utils/isi2bibtex/`
Package upgrade to v. 0.32.

`jurabib:` Upgrade to v. 0.4j. A package for the use of BibTeX databases for German legal texts. It can be used both together with or independently from the `jura` document class.

`labelmag:` in `systems/win32/texhelpers/labelmag/`
A Win32 program for managing a collection of labels to be typeset by TeX and printed on demand. The collection will be held together within a single file; labels may be kept within categories. As an advanced feature, mailing labels, for instance, may query an ODBC data source. The formatting macros may also be of interest for people not willing or able to use `labelmag`.

`lgrind:` in `support/lgrind/bin/`
A Win32 executable — it should run under almost any Windows and it supports long filenames.

`ltxmc***:` in `systems/win32/util/`
Set of LaTeX macros to integrate with your text editor. Also helps you write document templates (with options and packages) and creates the skeleton of an array, a table, or complex environments, by selecting the appropriate button on a toolbar.

`marvosym:` in `fonts/psfonts/`
The Martin Vogels Symbole (marvosym) font is a Type 1 font. The `marvosym` style file lets you use this font in LaTeX2e documents (package upgrade).

`mathsPIC:` in `graphics/pictex/`
`mathsPIC` is a DOS filter for use with PiCTeX, and facilitates drawing maths and geometry figures. This upgrade (v. 1.9b) fixes a bug associated with the `DrawAngleArrow`, `drawArrow`, and `drawThickArrow` commands, which arose when the y-coordinate of the terminal point was negative, causing a failure when the output file was LaTeX'd. All the v. 1.9a

documentation files apply as before. The manual is available in .tex, .dvi, .ps.

mil3: in info/
Final versions of the sample text files for forthcoming George Gratzer book, *Math into LaTeX* (3rd ed.).

ochem: in support/
Upgrade to v. 3.0 with some errors fixed, new bond types added, and a package for typesetting Kekule-like structures in the subdirectory ochem/kv.

pdfscreen: Upgrade to v. 1.3. This package helps to redesign the PDF output of your normal documents fit to be read in a computer monitor while retaining the freedom to format it for conventional printing.

plpsfont: in language/polish/
Upgrade to v. 1.14. Encoding files for the PL collection of fonts in PostScript Type 1 format for use with Windows. Fonts use the same .tfm files as generated from the pl-mf package.

pl-mf: in language/polish/
Upgrade: small bug fixes and code clean-up.

pl-tfm: in language/polish/
New tfm files generated from the new pl-mf sources.

psnfss-beta: in macros/latex/required/psnfss-beta/
An improved release of the PSNFSS v. 8.1 documentation.

psnfss-src: in fonts/psfonts/
Contains the sources (i.e., makefiles and fontinst scripts) of the new PSNFSS v. 8.1.

qxcm: in fonts/psfonts/polish/qfonts/
An experimental collection of extended CM fonts in PostScript Type 1 format (public domain). The original Computer Modern layout is extended to QX layout as an alternative to EC (Cork) encoding, usable also in the Windows environment. Fonts can be adapted to any needed TeX encoding; as they contain the most common European characters, the fonts can be used in preparing good PDF files, for example.

savefnmark: Sometimes the same footnote applies to more than one location in a table. With this package the mark of a footnote can be saved into a name, and re-used subsequently without creating another footnote at the bottom.

SIunits: Upgrade to v. 1.9. Package typesets physical units following the rules of the International System of Units (SI).

smartref: in .../other/
Upgrade to v. 1.6. The purpose of this package is to extend the capabilities of the \ref command.

TeXmacs: in systems/unix/TeXmacs/
Upgrade to v. 0.2.4g.

TeXmacs home page: www.math.u-psud.fr/~anh/TeXmacs/TeXmacs.html

texmalli: in info/
A quick introduction to using LaTeX, written in Finnish as a sample document (with the source being as relevant to the reader as the typeset result).

titlesec: Upgrade to v. 2.3.4. Essentially a replacement — partial or total — for the LaTeX macros related to sections; namely, titles, headers, and contents. This release fixes some bugs, adds a minor new feature, and provides a new contact e-mail address for the developer.

tmview: in dviware/
Upgrade to v. 00.05. The new version supports linux' framebuffer device and double-page viewing.

ttf2pt1: in fonts/utilities/
Converts TrueType fonts into PS Type 1 fonts.

ttf-tetex: in info/TrueType/
Upgrade to documentation on using TrueType fonts. In the new version the document also describes how to use the automatically generated slanted and small capitals versions of the font.

TVS: in support/
TeX Versioning System (TVS) is a Perl script which collects all files needed for re-typesetting TeX documents. It does this by parsing TeX logs. TVS is able to handle filenames intelligently.

June 2000

abstract: The abstract package gives you control over the typesetting of the abstract environment, and in particular provides for a one-column abstract in a two-column paper.

amscls: Some patches for the amscls distribution, including a most serious bug fix likely to affect page breaks in existing documents. See 00readme.txt.

amsmath: Some patches for the amsmath distribution. In addition, the files amsdtx.dtx and amsdtx.cls have been moved to the amscls distribution. See the file 00readme.txt.

arrayjob: in macros/generic/
The package provides array data structures in (LA)TeX, in the sense of classic procedural programming languages such as Fortran, Ada, or C, and macros to manipulate them. Arrays can be mono- or bi-dimensional.

catdvi: in dviware/
Upgrade to v. 0.10. A DVI-to-plain text translator aiming to be a superior free replacement for dvi2tty. The program is under development; home page: catdvi.sourceforge.net.

chngpage: in .../misc/
The package provides commands to change the page layout in the middle of a document (e.g., make the textblock wider or narrower, and/or longer or shorter, and/or shift it vertically or horizontally).

combine: Upgrade to v. 0.41. The combine class lets you bundle individual documents into a single document, such as when preparing a conference proceedings.

The auxiliary combinet package puts the titles and authors from \maketitle commands into the main document's table of contents.

`dvichk`: in `dviware/`
This package checks for the page numbers of a formatted text in a `.dvi`/`.log` file created by the TEX formatter and displays the page numbers found on standard output channel.

`epsfview.sea.hqx`: in `systems/mac/`
An AppleScript tool (for Mac) mainly intended for viewing figures generated with METAPOST, even if they have negative coordinates, and based on the Perl script shown on p. 459 of *The Graphics Companion*.

`expressg`: Upgrade to v. 1.3. The `expressg` MetaPost package provides facilities for drawing diagrams that consist of boxes, lines and annotations. Particular support is provided for creating EXPRESS-G diagrams.

`faq`: in `usergrps/uktug/`
A new version of the FAQ, both on CTAN and on the web.

`Fontmap.cmr`: in `fonts/cm/ps-type1/contrib/`
This file make the Computer Modern font files available to Ghostscript. Using a recent Ghostscript, this configuration file helps viewing/processing PostScript/PDF files without embedded fonts.

`geometry`: Upgrade to v. 2.3. This package provides an easy and flexible user interface to customize page layout. It implements auto-centering and auto-balancing mechanisms so that the users have only to give the least description for the page layout.

`ifmslide`: A package to produce printed slides with LATEX2e and on-line presentations with pdfTEX.

`ifsym`: in `fonts/`
METAFONTs for miscellaneous symbols (alpinistic, electronic, geometric, etc.)

`isodate`: Upgrade to v. 1.04. This package provides commands to switch between different date formats (standard, ISO, numeric, TEX package). They are used by the `\today` command and the `\printdate` and `\printdateTeX` commands, which print any date.

`jas99_m.bst`: in `biblio/bibtex/contrib/`
BIBTEX style based on `jas99.bst`, modified for better conformity to the American Meteorological Society (AMS) style.

`latex2man`: in `support/`
A tool to translate UNIX manual pages written with LATEX into a format understood by the UNIX man(1)-command. Alternatively, HTML or TexInfo code can also be produced.

`LaTeX_WIDE`: in `nonfree/systems/win32/`
Demonstration version of the program LATEX WIDE (a multifunctional editor for LATEX in Windows 95/98/NT/2000).

`lhelp`: This package defines macros that are useful for many documents. It is a large collection of simple "little helpers" that do not really warrant a separate package on their own.

`makecmds`: The package provides several additional commands along the lines of the traditional `\(new|renew|provide)command` and friends. [CTAN note: `makecmd.sty` is thus superseded and has been deleted from `.../misc/`.]

`mathpazo`: in `fonts/`
The Pazo Math fonts are a family of PostScript fonts suitable for typesetting math in combination with the Palatino family of text fonts.

`poster.txt`: in `support/`
A plain text version of the package `man` page. The `poster` package resizes a PostScript image to print on larger media and/or multiple sheets.

`secdot`: in `.../other/misc/`
Makes the numbers of `\section` commands come out with a trailing dot. Includes a command whereby the same can be made to happen with other sectioning commands.

`skak`: in `fonts/skak/`
This is a package for typesetting chess. The moves of the game can be given using PGN syntax and diagrams of the current board position can be shown in the document.

`substr`: This package provides commands to deal with substrings in strings: to determine if a string contains a substring, and to count appearances of a substring in a string.

`texdoctk`: in `systems/unix/teTeX/1.0/contrib/`
Upgrade to v. 0.4.0. A Perl/Tk-based GUI for easy access to package documentation; the databases it uses are based on the `texmf/doc` subtrees of teTEX v. 1.0.x, but database files for local configurations with modified/extended directories can be derived from them.

`TeXnicCenter`: in `systems/win32/`
An integrated development environment (IDE) for developing LATEX documents on windows (Windows 95, 98, 2000, NT 4.0).

`titling`: The titling package gives you control over the typesetting of the `\maketitle` command, and makes the `\title`, `\author` and `\date` information permanently available.

`varindex`: in `.../misc/`
A package that provides a convenient front-end for the `\index` command (e.g., generate multiple index entries in almost any form via a single command). Extremely customizable. Works with all versions of LATEX and probably most other TEX formats, too.

`winshell`: in `systems/win32/`
Bug fix. A graphical user interface for easy working with TEX. Website: `www.winshell.de`

◇ Christina Thiele
15 Wiltshire Circle
Nepean, Ontario
K2J 4K9 Canada
`cthiele@ccs.carleton.ca`

LaTeX News

Issue 13, June 2000

Yearly release cycle

We announced in *LaTeX News 11* that we intended to switch to a 12-monthly release schedule. With the present (June 2000) release, this switch is being made: thus the next release of LaTeX will be dated June 2001. We shall of course continue, as in the past, to release patches as needed to fix significant bugs.

PSNFSS: Quote of the Month

> You should say in the LaTeX News that Walter Schmidt has taken over PSNFSS from me. It gives me a certain pleasure to be able to draw a line under that part of my life…
>
> Sebastian Rahtz

The PSNFSS material, which supports the use of common PostScript fonts with LaTeX, has been thoroughly updated. Most noticeably, the mathpple package, which used to be distributed separately, is now part of the basic PSNFSS bundle; this package provides mathematical typesetting with the Palatino typeface family. In addition, numerous bugs and flaws have been fixed and the distribution has been 'cleaned up'. The file `changes.txt` contains a detailed list of these changes.

The documentation (in `psnfss2e.pdf`) has been completely rewritten to provide a comprehensive introduction to the use of PostScript fonts.

Notice that the new PSNFSS needs updated files for font metrics, virtual fonts and font definitions. If you received the new version (8.1) as part of a complete TeX system then these new font files should also have been installed. However, if you intend to install or update PSNFSS yourself, please read the instructions in the file `00readme.txt` of the new PSNFSS distribution.

Support for commercial PostScript fonts, such as Lucida Bright, has been removed from the basic distribution; it is now available from CTAN: `http://www.tex.ac.uk/tex-archive/macros/latex/contrib/supported/psnfssx`.

New AMS-LaTeX

Version 2.0 of AMS-LaTeX was released on December 1, 1999. It can be obtained via `ftp://ftp.ams.org/pub/tex/` or `http://www.ams.org/tex/amslatex.html`, as well from CTAN: `http://www.tex.ac.uk/tex-archive/macros/latex/required/amslatex`.

This release consists chiefly of bug fixes and consolidation of the existing features. The division of AMS-LaTeX into two main parts (the math packages; the AMS document classes) has been made more pronounced. The files `diffs-m.txt`, `diffs-c.txt`, `amsmath.faq`, and `amsclass.faq` describe the changes and address some common questions.

The primary documentation files remain `amsldoc.tex`, for the amsmath package, and `instr-l.tex`, for the AMS document classes. The documentation for the amsthm package, however, has been moved from `amsldoc.tex` to a separate document `amsthdoc.tex`.

New input encoding latin4

The package inputenc has, thanks to Hana Skoumalová, been extended to cover the latin4 input encoding; this covers Baltic and Scandinavian languages as well as Greenland Inuit and Lappish.

New experimental code

In *LaTeX News 12* we announced some ongoing work towards a 'Designer Interface for LaTeX' and we presented some early results thereof. Since then, at Gutenberg 2000 in Toulouse and TUG 2000 in Oxford, we described a new output routine and an improved method of handling vertical mode material between paragraphs. In combination these support higher quality *automated*[1] page-breaking and page make-up for complex pages—the best yet achieved with TeX!

A paper describing the new output routine is at `http://www.latex-project.org/papers/xo-pfloat.pdf` All code examples and documentation are available at `http://www.latex-project.org/code/experimental/` This directory has been extended to contain

galley Prototype implementation of the interface for manipulating vertical material in galleys.

xinitials Prototype implementation of the interface for paragraph initials (needs the `galley` package.

xtheorem Contributed example using the `template` package to provide a designer interface for theorem environments.

xoutput A prototype implementation of the new output routine as described in the `xo-pfloat.pdf` paper. Expected availability: at or shortly after the TUG 2000 conference.

[1] The stress here is on automated!

LaTeX News

Issue 13, June 2000

Yearly release cycle

We announced in *LaTeX News 11* that we intended to switch to a 12-monthly release schedule. With the present (June 2000) release, this switch is being made: thus the next release of LaTeX will be dated June 2001. We shall of course continue, as in the past, to release patches as needed to fix significant bugs.

PSNFSS: Quote of the Month

> You should say in the LaTeX News that Walter Schmidt has taken over PSNFSS from me. It gives me a certain pleasure to be able to draw a line under that part of my life...
>
> Sebastian Rahtz

The PSNFSS material, which supports the use of common PostScript fonts with LaTeX, has been thoroughly updated. Most noticeably, the mathpple package, which used to be distributed separately, is now part of the basic PSNFSS bundle; this package provides mathematical typesetting with the Palatino typeface family. In addition, numerous bugs and flaws have been fixed and the distribution has been 'cleaned up'. The file `changes.txt` contains a detailed list of these changes.

The documentation (in `psnfss2e.pdf`) has been completely rewritten to provide a comprehensive introduction to the use of PostScript fonts.

Notice that the new PSNFSS needs updated files for font metrics, virtual fonts and font definitions. If you received the new version (8.1) as part of a complete TeX system then these new font files should also have been installed. However, if you intend to install or update PSNFSS yourself, please read the instructions in the file `00readme.txt` of the new PSNFSS distribution.

Support for commercial PostScript fonts, such as Lucida Bright, has been removed from the basic distribution; it is now available from CTAN: `http://www.tex.ac.uk/tex-archive/macros/latex/contrib/supported/psnfssx`.

New AMS-LaTeX

Version 2.0 of AMS-LaTeX was released on December 1, 1999. It can be obtained via `ftp://ftp.ams.org/pub/tex/` or `http://www.ams.org/tex/amslatex.html`, as well from CTAN: `http://www.tex.ac.uk/tex-archive/macros/latex/required/amslatex`.

This release consists chiefly of bug fixes and consolidation of the existing features. The division of AMS-LaTeX into two main parts (the math packages; the AMS document classes) has been made more pronounced. The files `diffs-m.txt`, `diffs-c.txt`, `amsmath.faq`, and `amsclass.faq` describe the changes and address some common questions.

The primary documentation files remain `amsldoc.tex`, for the amsmath package, and `instr-l.tex`, for the AMS document classes. The documentation for the amsthm package, however, has been moved from `amsldoc.tex` to a separate document `amsthdoc.tex`.

New input encoding latin4

The package inputenc has, thanks to Hana Skoumalová, been extended to cover the latin4 input encoding; this covers Baltic and Scandinavian languages as well as Greenland Inuit and Lappish.

New experimental code

In *LaTeX News 12* we announced some ongoing work towards a 'Designer Interface for LaTeX' and we presented some early results thereof. Since then, at Gutenberg 2000 in Toulouse and TUG 2000 in Oxford, we described a new output routine and an improved method of handling vertical mode material between paragraphs. In combination these support higher quality *automated*[1] page-breaking and page make-up for complex pages—the best yet achieved with TeX!

A paper describing the new output routine is at `http://www.latex-project.org/papers/xo-pfloat.pdf` All code examples and documentation are available at `http://www.latex-project.org/code/experimental/` This directory has been extended to contain

galley Prototype implementation of the interface for manipulating vertical material in galleys.

xinitials Prototype implementation of the interface for paragraph initials (needs the `galley` package.

xtheorem Contributed example using the `template` package to provide a designer interface for theorem environments.

xoutput A prototype implementation of the new output routine as described in the `xo-pfloat.pdf` paper. Expected availability: at or shortly after the TUG 2000 conference.

[1] The stress here is on automated!

Calendar

2000

Aug 12 – 18 **TUG 2000** — The 21st annual meeting of the TeX Users Group, “TeX enters a new millennium”, Wadham College, Oxford, UK. For information, visit `http://tug2000.tug.org/`.

Aug 28 – Sep 1 Seybold San Francisco, San Francisco, California. For information, visit `http://www.seyboldseminars.com/Events`.

Sep 11 – 12 PODDP ’00: 5th International Workshop on Principles of Digital Document Processing, Munich, Germany. For information, visit `http://www.cs.uwm.edu/~poddp00`.

Sep 13 – 15 DDEP00: 8th International Conference on Digital Documents and Electronic Publishing, Munich, Germany. For information, visit `http://www11.in.tum.de/DDEP00`.

Sep 21 DK-TUG, 2nd Annual General Meeting, Aarhus University. For information, visit `http://sunsite.dk/dk-tug/`.

Oct 6 – 7 DANTE, 23rd meeting, Fern-universität Hagen, Germany. For information, visit `http://www.dante.de/dante/mv/mv23/`.

Oct 20 – 21 MathML and Technologies for Math on the Web, Urbana-Champaign, Illinois. For information, visit `http://www.mathmlconference.org`.

Nov 13 – Jan 6 Gutenberg exhibit, including working replica of his original printing press, Louisville Free Public Library, Louisville, Kentucky.

Nov 16 NTG 26th Meeting, Hotel Haarhuis, Arnhem, The Netherlands. For information, contact `ntg@ntg.nl` or visit `http://www.ntg.nl/bijeen/bijeen26.html`.

Nov 17 – 19 Conference: Eric Gill & St. Dominic’s Press, University of Notre Dame, Notre Dame, Indiana; three concurrent exhibitions of Gill’s and related work will be held in the University museums and library. For information, visit `http://www.nd.edu/~jsherman/gill/`.

Nov 20 *TUGboat* **21** (4), deadline for reports and news items.

Nov 24 UK TUG Annual General Meeting & Technical session, City University, London, UK. For information, visit `http://uk.tug.org/UK-TuG/Meetings.html`.

Dec 3 – 7 XML 2000/Markup Technologies 2000, Washington, DC. For information, visit `http://www.gca.org/attend/2000_conferences/XML_2000/`.

2001

Feb 28 – Mar 3 DANTE 2001, 24th meeting, Fachhochschule Rosenheim, Germany. For information, visit `http://www.dante.de/dante2001/`.

Mar 1 **TUG 2001 election**, nominations due to TUG office

Apr 29 – May 1 BachoTeX 2001, 9th annual meeting of the Polish TeX Users’ Group (GUST), “Contemporary publishing TeXnology”, Bachotek, Brodnica Lake District, Poland. For information, visit `http://www.gust.org.pl/BachoTeX/`.

May 14 – 17 Congrès GUTenberg 2001, “Le document au XXIe Siècle”, Metz, France. For information, visit `http://www.gutenberg.eu.org/manif/gut2001/`.

Jun 6 – 8 Society for Scholarly Publishing, 23rd annual meeting, San Francisco, California. For information, visit `http://www.sspnet.org`.

Jun 13 – 17 ACH/ALLC 2001: Joint International Conference of the Association for Computers and the Humanities, and Association for Literary and Linguistic Computing, New York University, New York. For information, visit `http://www.nyu.edu/its/humanities/ach_allc2001/`.

Aug 12 – 16 **TUG 2001** — The 22nd annual meeting of the TeX Users Group, “2001L A TeX Live Odyssey, University of Delaware, Newark, Delaware. For information, visit `http://www.tug.org/tug2001/`.

Status as of 30 September 2000

For additional information on TUG-sponsored events listed above, contact the TUG office (+1 503 223-9994, fax: +1 503 223-3960, e-mail: `office@tug.org`). For events sponsored by other organizations, please use the contact address provided.

Additional type-related events and news items are listed in the Sans Serif Web pages, at `http://www.quixote.com/serif/sans`.

Aug 12 – 17 SIGGRAPH 2001, Los Angeles, California. For information, visit `http://www.siggraph.org/s2001/`.

Sep 23 – 27 EuroTeX 2001, “TeX and Meta: the Good, the Bad and the Ugly Bits”, Kerkrade, Netherlands. For information, visit `http://www.ntg.nl/eurotex/`.

Oct 24 – 26 4th International Conference on The Electronic Document, Toulouse, France. For information, visit `http://www.irit.fr/CIDE2001/`.

Late-Breaking News

Production Notes

Mimi Burbank

We're beginning to catch up. Barbara Beeton, the anchor holding *TUGboat* safe in thrashing seas, has had to reduce her accrued vacation time during the past several months. Me, I've been ill since August and generally getting an excellent introduction to the inadequacies of HMO's. Christina has just recently returned from China, where she and her husband Mike traveled in October to pick up their adopted daughter, Anna. She is now in the process of "being trained by the baby."

Enclosed in this issue are the three CTAN CD-ROMs of course. This should go quite nicely with the current "Treasure Chest" list, and the others throughout the year. These columns, together with the TL5 documentation, should give you a good idea of what and where to look for things which might be of use to you.

Output The final camera copy was prepared at CSIT on a Linux running Red Hat 7, using the *TeX Live* 4 setup, i386-linux, which is based on the *Web2c* TeX implementation version 7.3 by Karl Berry and Olaf Weber. PostScript output, using outline fonts, was produced using Radical Eye Software's dvips(k) 5.85, and printed on an HP LaserJet 4000 TN printer at 1200dpi.

We had a PostScript problem with one graphic image which was a TIFF image converted by tiff2ps — this image would not print on any of our HP printers, but would print on our old QMS printer – the interpreter being more forgiving on this machine. Can you identify the article in which this occured?

I hope to have the TeX Live 5 installation completed soon and the next issue should be produced using this year's version of TeX Live!

Coming In Future Issues The next issue will contain the proceedings of the 21st annual meeting of the TeX Users Group, held at Wadham College, Oxford, UK, with Robin Fairbairns serving as the Proceedings Editor.

◇ Mimi Burbank
CSIT
Florida State University
Tallahassee, FL 32306 – 4120
mimi@csit.fsu.edu

TUG 2000

Wadham College, Oxford, UK
August 13th–16th, 2000

The 21st Annual Conference of the TEX Users Group will take place at Wadham College, Oxford, between Sunday 13th August and Wednesday 16th August 2000. Tutorials will be given on the 17th and 18th August.

The Location

Oxford is a small, pleasant city with an internationally famous university. The city is full of ancient buildings, beautiful gardens, libraries and bookshops. The conference will be held in Wadham College, a traditional college (founded 1613) in the centre of the city. Oxford is easily reached from London, and is a good starting point for visiting much of southern England.

The Conference

The conference will feature talks on all aspects of TEX and its relationship to both traditional and electronic document preparation and processing.
The Annual General Meeting of the TEX Users' Group will be held during the period of the conference.
We expect the cost to a typical delegate to be about £300, including accommodation and meals; cheaper accommodation and bursaries will also be available.
The conference chairman is Sebastian Rahtz (Oxford University Computing Services) and local organisation is led by Kim Roberts (Oxford University Press).

Dates and Contacts

15th January 2000	Proposals for papers
31st January 2000	Acceptance of papers
15th February 2000	Publication of booking form and prices
31st March 2000	Delivery of papers for refereeing
31st May 2000	Delivery of final papers
General enquiries:	`tug2000-enquiries@tug.org`
Paper submissions:	`tug2000-papers@tug.org`

Sebastian Rahtz
OUCS
13 Banbury Road
Oxford OX2 6NN, UK
Tel: +44 1865 283431
http://tug2000.tug.org/

TUG Business

Report from the TUG Treasurer

Donald DeLand

A comparative summary of TUG finances for 1996 through 1999 is presented below, as well as a balance sheet for the close of 1999. Although these accounts have been reviewed by TUG's accountants, the accountants have not audited or verified the information presented here.

Because of changes in accounting practices over the years, not all items are itemized in the comparative financial statement. Minor discrepancies in the P&L totals may exist because of rounding errors.

Editor's note: Financial reports were last published in *TUGboat* **17**:4 (1996), pages 406–408.)

PROFIT & LOSS, 1996–1999

	1996	1997	1998	1999
Number of members	1531	1436	2035	2090
Income				
Member dues	60,457	68,781	82,829	124,042
Advertising				100
Conferences		23,885	24,516	36,543
Sales	16,991	4,100		6,064
Contributions/Interest	22,579	9,780	21,127	7,514
MTOT *				(2,523)
Total	100,027	106,546	128,472	171,740
Cost of Goods				
TUGboat	22,488	34,495	23,724	36,834
CD-ROMS				12,672
Conference		19,754	34,425	21,355
Bursary Fund				1,400
LaTeX 3 (transfer of assets in trust)				12,852
Delivery				
Misc.	4,192	3,811		831
Total	26,680	58,060	58,149	85,944
Expenses				
Wages	67,649	25,796	31,594	56,787
Equipment			409	7,432
Operations	44,502	44,000	44,327	14,664
Other				
Total	112,151	69,796	76,330	78,883
Net Income	(38,804)	(21,310)	(6,007)	6,913

* MTOT is the account to which BankCard merchant fees are posted.

BALANCE SHEET, December 31, 1999

	1999
Assets	
Cash	
Net balance, Bank of America checking	344.07
Net balance, Santa Barbara checking	678.27
Money market, Bank of America	50,407.77
CD, Bank of America	34,693.19
Bursary fund trust money market, Bank of America	4,201.32
Total Cash	90,324.62
Accounts Receivable	10,451.72
Other Current Assets	500.00
Fixed Assets	
Equipment	28,887.42
Accumulated Depreciation	(22,345.82)
Total Fixed Assets	6,541.60
Total Assets	107,817.94
Liabilities & Equity	
Liabilities	
Accounts Payable	21,328.95
Suspense	1,818.42
Total Liabilities	23,147.37
Equity	
Opening Balance Equity	(7,596.54)
Permanent Restricted	6,542.00
Prior Earnings	(41,016.99)
Retained Earnings	12,003.14
Temporary Restricted	46,767.00
Unrestricted	61,060.88
Net Income	6,911.08
Total Equity	84,670.57
Total Liabilities & Equity	107,817.94

◇ Donald DeLand
Integre Technical Publishing Co.
4015-B Carlisle N.E.
Albuquerque, NM 87107 U.S.A.
deland@cs.unm.edu

2001 TeX Users Group Election

Barbara Beeton
for the Elections Committee

The positions of TUG President and of 4 members of the Board of Directors will be open as of the 2001 Annual Board Meeting, which will take place in conjunction with the 22nd Annual TUG Meeting to be held in August 2001 at the University of Delaware.

The current President, Mimi Jett, has stated her intention to stand for re-election. The terms of the three directors whose terms will expire in 2001 are Arthur Ogawa, Patricia Monohon, and Petr Sojka. One additional director position is currently unoccupied. Continuing directors, with terms ending in 2003, are Barbara Beeton, Karl Berry, Kaja Christiansen, Don DeLand, Susan DeMeritt, Stephanie Hogue, Judy Johnson, Ross Moore, Cheryl Ponchin, Kristoffer Rose and Philip Taylor.

The election to choose the new President and Board members will be held in Spring of 2001. Nominations for these openings are now being invited.

The Bylaws provide that "Any member may be nominated for election to the office of TUG President/to the Board by submitting a nomination petition in accordance with the TUG Election Procedures. Election ... shall be by written mail ballot of the entire membership, carried out in accordance with those same Procedures."

The name of any member may be placed in nomination for election to one of the open offices by submission of a petition, signed by two other members in good standing, to the TUG office at least two weeks (14 days) prior to the mailing of ballots. (A candidate's membership dues for 2001 will be expected to be paid by the nomination deadline.) A nomination form follows this announcement; forms may also be obtained from the TUG office, and electronically via the TUG Web pages at `http://www.tug.org`.

Along with a nomination form, each candidate is asked to supply a passport-size photograph, a short biography, and a statement of intent to be included with the ballot; the biography and statement of intent together may not exceed 400 words. The deadline for receipt at the TUG office of nomination forms and ballot information is **1 March 2001**.

Ballots will be mailed to all members within 30 days after the close of nominations. Marked ballots must be returned no more than six (6) weeks following the mailing; the exact dates will be noted on the ballots.

Ballots will be counted by a disinterested party not part of the TUG organization. The results of the election should be available early in June, and will be announced in a future issue of *TUGboat* as well as through various TeX-related electronic lists.

2001 TUG Election — Nomination Form

Only TUG members whose dues have been paid for 2001 will be eligible to participate in the election. The signatures of two (2) members in good standing at the time they sign the nomination form are required in addition to that of the nominee. **Type or print** names clearly, using the name by which you are known to TUG. Names that cannot be identified from the TUG membership records will not be accepted as valid.

The undersigned TUG members propose the nomination of:

Name of Nominee:

Signature:

Date:

for the position of (check one):

□ **TUG President**

□ **Member of the TUG Board of Directors**

for a term beginning with the 2001 Annual Meeting, **August 2001**.

Members supporting this nomination:

1.
(please print)

(signature) (date)

2.
(please print)

(signature) (date)

Return this nomination form to the TUG office (FAXed forms will be accepted). Nomination forms and all required supplementary material (photograph, biography and personal statement for inclusion on the ballot) must be received in the TUG office no later than **1 March 2001**.[1] It is the responsibility of the candidate to ensure that this deadline is met. Under no circumstances will incomplete applications be accepted.

□ nomination form
□ photograph
□ biography/personal statement

TeX Users Group **FAX:** +1 503 223-3960
Nominations for 2001 Election
1466 NW Naito Parkway, Suite 3141
Portland, OR 97209-2820
U.S.A.

[1] Supplementary material may be sent separately from the form, and supporting signatures need not all appear on one form.

TEX
USERS
GROUP

Promoting the use of TeX throughout the world

mailing address:
P.O. Box 2311
Portland, OR 97208–2311 USA

shipping address:
1466 NW Naito Parkway,
Suite 3141
Portland, OR 97209–2820 USA

Phone:	+1 503 223–9994
Fax:	+1 503 223–3960
Email:	`office@tug.org`
WWW:	`www.tug.org`

President:	Mimi Jett
Vice-President:	Kristoffer Høgsbro Rose
Treasurer:	Donald W. DeLand
Secretary:	Arthur Ogawa

2000 TUG Membership Form

Rates for TUG membership and TUGboat subscription are listed below. Please check the appropriate boxes and mail payment (in US dollars, drawn on a United States bank) along with a copy of this form. If paying by credit card, you may fax the completed form to the number at left.

- 2000 TUGboat includes Volume 21, nos. 1–4.
- 2000 CD-ROMs include TeX Live 5 (1 disk) and Dante's CTAN (3 disk set).
- *Multi-year orders:* You may use this year's rate to pay for more than one year of membership.
- Orders received after March 1, 2000: please add $10 to cover the additional expense of shipping back numbers of TUGboat and CD-ROMs.

		Rate	Amount
Annual membership for 2000 (TUGboat and CD-ROMs)	□	$65	____
Student/Senior membership for 2000 (TUGboat and CD-ROMs) (please attach photocopy of 2000 student/senior ID)	□	$35	____
Subscription for 2000 (TUGboat and CD-ROMs) (Non-voting)	□	$75	____
Shipping charge if after March 1, 2000.	□	$10	____
Materials for 1999† (TUGboat Volume 20, TeX Live 4, 1999 CTAN CD-ROMs)	□	$75	____
Voluntary donations			
General TUG contribution	□		____
Contribution to Bursary Fund*	□		____
		Total $	____

Payment (check one) □ Payment enclosed □ Charge Visa/Mastercard/AmEx

Account Number: ______________________________

Exp. date: ________ Signature: ______________________________

*The Bursary Fund provides financial assistance to members who otherwise would be unable to attend the TUG Annual Meeting.

† If you are a new TUG member wishing to receive TeX Live and CTAN right away, please order this item along with your 2000 membership.

We use the information you provide to mail you products, publications, notices, and (for voting members) official ballots, or in a printed or electronic membership list, available to TUG members only.

Note: TUG neither sells its membership list nor provides it to anyone outside of its own membership.

If you do *not* wish to have your name or other information in our membership list, please check here: □ .

Name: ______________________________

Department: ______________________________

Institution: ______________________________

Address: ______________________________

____

____

Phone: ______________________________ Fax: ______________________________

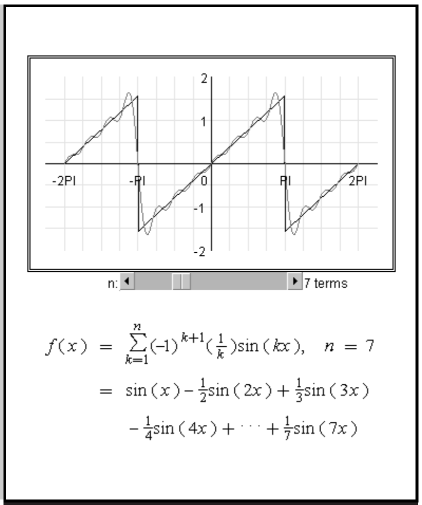
Email address: ______________________________

Position: ______________________________ Affiliation: ______________________________

www.software.ibm.com/techexplorer

techexplorer Hypermedia Browser v3

Live Web Math with Interactivity



$$\begin{aligned} f(x) &= \sum_{k=1}^{n}(-1)^{k+1}\left(\frac{1}{k}\right)\sin(kx), \quad n = 7 \\ &= \sin(x) - \tfrac{1}{2}\sin(2x) + \tfrac{1}{3}\sin(3x) \\ &\quad - \tfrac{1}{4}\sin(4x) + \cdots + \tfrac{1}{7}\sin(7x) \end{aligned}$$

techexplorer is a plug-in for Web browsers that allows users to have live, interactive mathematics on an HTML or XML page. The Intro Edition is a free LaTeX or MathML reader. The Pro Edition includes advanced features for truly interactive mathematical collaboration and publishing.

IBM Research announces techexplorer 3, released in conjunction with the MathML International Conference 2000. Special new features include: Interoperability with *Mathematica* 4.1 from Wolfram Research, Inc.; A Macintosh version for both the Professional and Introductory Editions; ActiveX allows techexplorer to interactively work within many Windows applications, including Microsoft IE, Word, and Powerpoint; Support for W3C DOM and MathML standards allows interactivity with applets, multi-media resources, animations, and many other applications.

Write. . . create. . . collaborate. . . techexplorer makes your browser a scientific publishing environment. View LaTeX and MathML files; create interactive documents with Java applets, video or audio clips, animations, and graphics.

NEW FEATURES:
- Macintosh platform
- *Mathematica*™ connectivity
- ActiveX control
- DOM API
- LaTeX-2-techexplorer filter

KEY FEATURES:
- MathML rendering and conversion
- LaTeX viewing and editing
- ActiveX control
- DOM API functionality and support
- Create pop-up and pull-down menus
- In-line video and auto-play audio
- AMS symbol fonts
- Expression editor
- Interact with many tools and applications
- Java and JavaScript

BROWSERS:
- Netscape, MS Internet Explorer, Opera

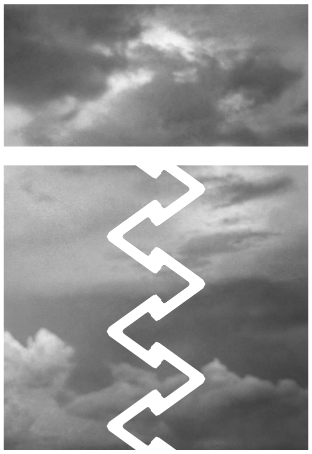
PLATFORMS:
- Pro: Windows 95/98/NT, Macintosh OS 8.6/9
- Intro: Windows, Mac, Linux, AIX, Solaris

$29.95 from ibm.com

introducing

TEXTURES® 2.0

WITH SYNCHRONICITY



Again the Macintosh delivers a new TeX with a revolution in human interface.

As computer power has advanced, the Macintosh has consistently been the leader in the human and humane connection to technology, and Textures has consistently led in bringing ease of use to TeX users. First with Textures 1.0, the first truly integrated TeX system. Then with Lightning Textures, the first truly interactive TeX system. Now, with Textures 2.0 and Synchronicity, Blue Sky Research again delivers a striking advance in TeX interactivity and productivity.

With Synchronicity, your TeX input documents are reliably and automatically cross-linked, correlated, or "synchronized" with the finished TeX typeset pages. Every piece of the finished product is tied directly to the source code from which it was generated, and vice-versa. To go from TeX input directly and exactly to the corresponding typeset characters, just click. It's that simple: just click, and Textures will take you instantly and precisely to the corresponding location. And it goes both ways: just click on any typeset character, and Textures will take you directly to the TeX code that produced it. No matter how many input files you have, no matter what macros you use, Synchronicity will take you there, instantly and dependably.

Improve YOUR *performance:*

GET SYNCHRONICITY

BLUE SKY RESEARCH
317 SW ALDER STREET
PORTLAND, OR 97204 USA

800 622 8398
503 222 9571
WWW.BLUESKY.COM

TeX Users Group

Memberships and Subscriptions

TUGboat (ISSN 0896-3207) is published quarterly by the TeX Users Group, 1466 NW Naito Parkway, Suite 3141, Portland, OR 97209-2820, U.S.A.

2000 dues for individual members are as follows:

- Ordinary members: $75.
- Students: $45.

Membership in the TeX Users Group is for the calendar year, and includes all issues of *TUGboat* for the year in which membership begins or is renewed. Individual membership is open only to named individuals, and carries with it such rights and responsibilities as voting in TUG elections. A membership form is provided on page 150.

TUGboat subscriptions are available to organizations and others wishing to receive *TUGboat* in a name other than that of an individual. Subscription rates: $85 a year, including air mail delivery.

Periodical-class postage paid at Portland, OR, and additional mailing offices. Postmaster: Send address changes to *TUGboat*, TeX Users Group, 1466 NW Naito Parkway, Suite 3141, Portland, OR 97209-2820, U.S.A.

Institutional Membership

Institutional Membership is a means of showing continuing interest in and support for both TeX and the TeX Users Group. For further information, contact the TUG office (`office@tug.org`).

Printed in U.S.A.

Addresses

General correspondence,
payments, etc.
TeX Users Group
P. O. Box 2311
Portland, OR 97208-2311
U.S.A.

Delivery services,
parcels, visitors
TeX Users Group
1466 NW Naito Parkway
Suite 3141
Portland, OR 97209-2820
U.S.A.

Telephone

+1 503 223-9994

Fax

+1 503 223-3960

Electronic Mail

(Internet)

General correspondence,
membership, subscriptions:
`office@tug.org`

Submissions to *TUGboat*,
letters to the Editor:
`TUGboat@tug.org`

Technical support for
TeX users:
`support@tug.org`

To contact the
Board of Directors:
`board@tug.org`

World Wide Web

`http://www.tug.org/`
`http://www.tug.org/TUGboat/`

Problems not resolved?

The TUG Board wants to hear from you:
Please email to `board@tug.org`

TUGboat

During 2000, the communications of the TEX Users Group will be published in four issues. The September issue (Vol. 21, No. 3) will contain the Proceedings of the 2000 TUG Annual Meeting.

TUGboat is distributed as a benefit of membership to all members.

Submissions to *TUGboat* are reviewed by volunteers and checked by the Editor before publication. However, the authors are still assumed to be the experts. Questions regarding content or accuracy should therefore be directed to the authors, with an information copy to the Editor.

Submitting Items for Publication

The next regular issue will be Vol. 21, No. 4. As production and mailing have been delayed, deadlines for contributions have already passed. Mailing is scheduled for late December. The third issue for 2000 is expected to be sent to the printer very late in November. Deadlines for other future issues are listed in the Calendar, page 144.

Manuscripts should be submitted to a member of the *TUGboat* Editorial Board. Articles of general interest, those not covered by any of the editorial departments listed, and all items submitted on magnetic media or as camera-ready copy should be addressed to the Editor, Barbara Beeton, or to the Production Manager, Mimi Burbank (see addresses on p. 99).

Contributions in electronic form are encouraged, via electronic mail, on diskette, or made available for the Editor to retrieve by anonymous FTP; contributions in the form of camera copy are also accepted. The *TUGboat* "style files", for use with either `plain` TEX or LATEX, are available from CTAN. For authors who have no network FTP access, they will be sent on request; please specify which is preferred. Send e-mail to `TUGboat@tug.org`, or write or call the TUG office.

This is also the preferred address for submitting contributions via electronic mail.

Reviewers

Additional reviewers are needed, to assist in checking new articles for completeness, accuracy, and presentation. Volunteers are invited to submit their names and interests for consideration; write to `TUGboat@tug.org` or to the Editor, Barbara Beeton (see address on p. 99).

TUGboat Advertising and Mailing Lists

For information about advertising rates, publication schedules or the purchase of TUG mailing lists, write or call the TUG office.

TUGboat Editorial Board

Barbara Beeton, *Editor*
Mimi Burbank, *Production Manager*
Victor Eijkhout, *Associate Editor, Macros*
Jeremy Gibbons, *Associate Editor,*
"Hey — it works!"
Alan Hoenig, *Associate Editor, Fonts*
Christina Thiele, *Associate Editor,*
Topics in the Humanities

Production Team:
Barbara Beeton, Mimi Burbank (Manager), Robin Fairbairns, Michael Sofka, Christina Thiele

See page 99 for addresses.

Other TUG Publications

TUG publishes the series *TEXniques*, in which have appeared reference materials and user manuals for macro packages and TEX-related software, as well as the Proceedings of the 1987 and 1988 Annual Meetings. Other publications on TEXnical subjects also appear from time to time.

TUG is interested in considering additional manuscripts for publication. These might include manuals, instructional materials, documentation, or works on any other topic that might be useful to the TEX community in general. Provision can be made for including macro packages or software in computer-readable form. If you have any such items or know of any that you would like considered for publication, send the information to the attention of the Publications Committee at `tug-pub@tug.org` or in care of the TUG office.

Trademarks

Many trademarked names appear in the pages of *TUGboat*. If there is any question about whether a name is or is not a trademark, prudence dictates that it should be treated as if it is. The following list of trademarks which appear in this issue may not be complete.

MS/DOS is a trademark of Microsoft Corporation
METAFONT is a trademark of Addison-Wesley Inc.
PC TEX is a registered trademark of Personal TEX, Inc.
PostScript is a trademark of Adobe Systems, Inc.
techexplorer is a trademark of IBM Research.
TEX and AMS-TEX are trademarks of the American Mathematical Society.
Textures is a trademark of Blue Sky Research.
UNIX is a registered trademark of X/Open Co. Ltd.

Addresses

TeX Users Group Office
Robin Laakso
1466 NW Naito Parkway, Suite 3141
Portland, OR 97209-2820 U.S.A.
+1 503 223-9994
Fax: +1 503-223-3960
office@tug.org

Barbara Beeton
American Mathematical Society
P. O. Box 6248
Providence, RI 02940 U.S.A.
+1 401 455-4014
bnb@ams.org, tugboat@tug.org

Alexander Berdnikov
Institute of Analytical
Instrumentation
Rizskii pr. 26
198103 St. Petersburg, Russia
berd@ianin.spb.su

Karl Berry
685 Larry Ave. N
Keizer, OR 97303 U.S.A.
karl@tug.org

Mimi R. Burbank
CSIT, 408 Dirac Science Library
Florida State University
Tallahassee, FL 32306-4130 U.S.A.
+1 850 644-2440
mimi@csit.fsu.edu

Kaja Christiansen
Dept. of Computer Science
Arhus Univ., Ny Munkegade
Bldg 540
DK-8000 Aarhus C, Denmark
kaja@daimi.aau.dk

Donald DeLand
Integre Technical Publishing Co.
4015-B Carlisle N.E.
Albuquerque, NM 87107 U.S.A.
deland@cs.unm.edu

Susan DeMeritt
IDA/CCR La Jolla
4320 Westerra Court
San Diego, CA 92121 U.S.A.
+1 619 622-5455
sue@ccrwest.org

Victor Eijkhout
Computer Science Department
111 Ayres Hall
University of Tennessee
Knoxville, TN 37996-1301 U.S.A.
victor@eijkhout.net

Robin Fairbairns
32 Lilac Court
Cherryhinton Rd.
Cambridge, CB1 4AY, U.K.
Robin.Fairbairns@cl.cam.ac.uk

Peter Flynn
Computer Centre
University College
Cork, Ireland
+353 21 902609
pflynn@imbolc.ucc.ie

Jeremy Gibbons
Oxford University Computing
Laboratory
Wolfson Building, Parks Road
Oxford OX1 3QD, U.K.
+44 1865 283508; Fax: +44 1865
273839
Jeremy.Gibbons@comlab.ox.ac.uk
http://www.comlab.ox.ac.uk/oucl/
people/jeremy.gibbons.html

Pedro Palao Gostanza
Universidad Complutense
Madrid, Spain
ecceso@sip.ucm.es

George Grätzer
Department of Mathematics
University of Manitoba
Winnipeg MN, R3T 2N2
Canada
gratzer@cc.umanitoba.ca

Hans Hagen
Pragma ADE
Ridderstraat 27
8061 GH Hasselt, The Netherlands
pragma@wxs.nl

John D. Hobby
Bell Laboratories
Room 2C-458
700 Mountain Ave.
Murray Hill, NJ 07974-0636
hobby@research.bell-labs.com

Taco Hoekwater
Kluwer Academic Publishers
Achterom 119
3311KB Dordrecht, The Netherlands
taco.hoekwater@wkap.nl

Alan Hoenig
17 Bay Avenue
Huntington, NY 11743 U.S.A.
+1 516 385-0736
ajhjj@cunyvm.cuny.edu
ahoenig@suffolk.lib.ny.us

Stephanie Hogue
The TypeWright
801 Highland Road
Lansdale, PA 19446 U.S.A.
shogue@typewright.com

Bogusław Jackowski
BOP sc., ul. Piastowska 70
80-331 Gdańsk Oliwa, Poland
B.Jackowski@gust.org.pl

Mimi Jett
IBM Research
1211 SW 5th Avenue
Suite 1000
Portland, OR 97204 U.S.A.
+1 503 294-6332
jett@us.ibm.com

Judy Johnson
jannejohnson@yahoo.com

Donald E. Knuth
Department of Computer Science
Stanford University
Stanford, CA 94305 U.S.A.

Werner Lemberg
Kl. Beurhausstr. 1
44137 Dortmund, Germany
wl@gnu.org

Raph Levien
artofcode
940 Tyler St., Studio 6
Benicia, CA 94510 U.S.A.
raph@acm.org

Frank Mittelbach
LaTeX3 Project Team
latex-l@relay.urz.uni-heidelberg.de

Patricia Monohon
University of California San Francisco
Dill Research Lab
3333 California Street, #415
San Francisco, CA 94118 U.S.A.
+1 415 502-2839
pmonohon@zimm.ucsf.edu

Ross Moore
Macquarie University
NSW 2109, Australia
ross@maths.mq.edu.au

Arthur Ogawa
40453 Cherokee Oaks Drive
Three Rivers CA 93271 U.S.A.
+1 209 561-4585; Fax: +1 209 561-4584
ogawa@teleport.com

Cheryl Ponchin
Center for Communications Research
Institute for Defense Analyses
29 Thanet Road
Princeton NJ 08540-3699
cheryl@ccr-p.ida.org

Roy Preston
4 Avon Wharf
Bridge Street
Christchurch
Dorset BH23 1DY, England UK
preston@lds.co.uk
http://www.lds.co.uk/preston/

Kristoffer Høgsbro Rose
IBM
T. J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532 U.S.A.
`krisrose@tug.org`
`http://tug.org/~krisrose`

Petr Sojka
`petr.sojka@tug.org`

Philip Taylor
The Computer Centre,
Royal Holloway and Bedford
New College,
University of London,
Egham Hill
Egham, Surrey TW20 0EX, U.K.
`P.Taylor@vax.rhbnc.ac.uk`

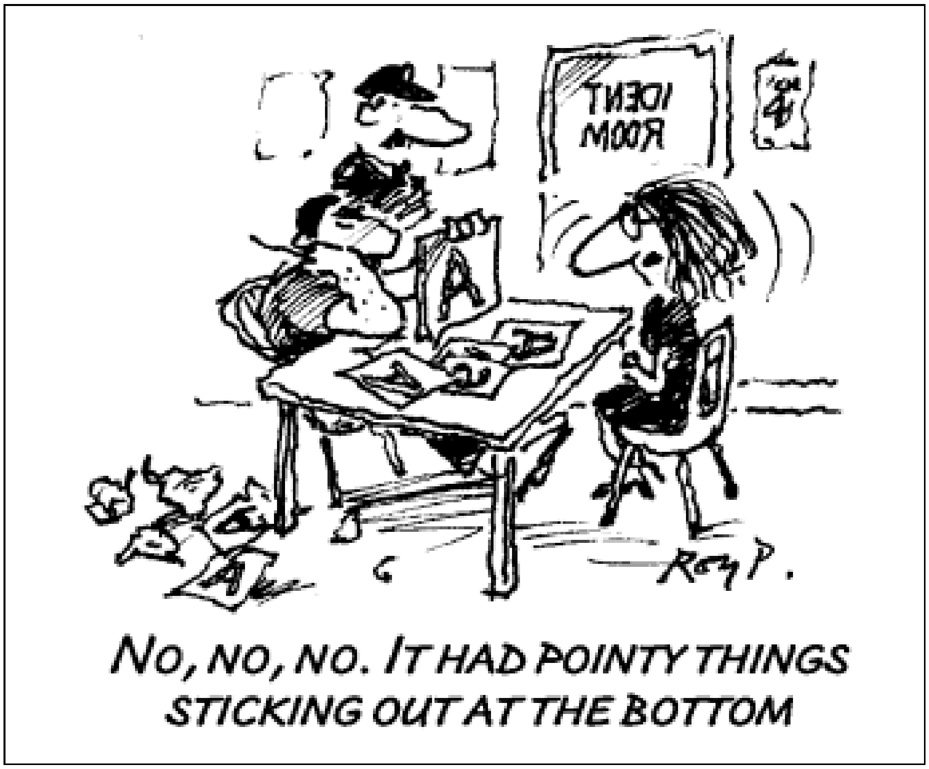
Christina Thiele
15 Wiltshire Circle
Nepean K2J 4K9, Ontario Canada
`cthiele@ccs.carleton.ca`

Sivan Toledo
School of Mathematical Sciences
Tel-Aviv University
Tel-Aviv 69978, Israel
`sivan@math.tau.ac.il`

Hermann Zapf
Seitersweg 35
D-64287 Darmstadt, Germany

Cartoon

by Roy Preston



Calendar

2000

Aug 12 – 18 **TUG 2000** — The 21st annual meeting of the TeX Users Group, "TeX enters a new millennium", Wadham College, Oxford, UK. For information, visit `http://tug2000.tug.org/`.

Aug 28 – Sep 1 Seybold San Francisco, San Francisco, California. For information, visit `http://www.seyboldseminars.com/Events`.

Sep 11 – 12 PODDP '00: 5th International Workshop on Principles of Digital Document Processing, Munich, Germany. For information, visit `http://www.cs.uwm.edu/~poddp00`.

Sep 13 – 15 DDEP00: 8th International Conference on Digital Documents and Electronic Publishing, Munich, Germany. For information, visit `http://www11.in.tum.de/DDEP00`.

Sep 21 DK-TUG, 2nd Annual General Meeting, Aarhus University. For information, visit `http://sunsite.dk/dk-tug/`.

Oct 6 – 7 DANTE, 23rd meeting, Fern-universität Hagen, Germany. For information, visit `http://www.dante.de/dante/mv/mv23/`.

Oct 20 – 21 MathML and Technologies for Math on the Web, Urbana-Champaign, Illinois. For information, visit `http://www.mathmlconference.org`.

Nov 13 – Jan 6 Gutenberg exhibit, including working replica of his original printing press, Louisville Free Public Library, Louisville, Kentucky.

Nov 16 NTG 26th Meeting, Hotel Haarhuis, Arnhem, The Netherlands. For information, contact `ntg@ntg.nl` or visit `http://www.ntg.nl/bijeen/bijeen26.html`.

Nov 17 – 19 Conference: Eric Gill & St. Dominic's Press, University of Notre Dame, Notre Dame, Indiana; three concurrent exhibitions of Gill's and related work will be held in the University museums and library. For information, visit `http://www.nd.edu/~jsherman/gill/`.

Nov 20 *TUGboat* **21** (4), deadline for reports and news items.

Nov 24 UK TUG Annual General Meeting & Technical session, City University, London, UK. For information, visit `http://uk.tug.org/UK-TuG/Meetings.html`.

Dec 3 – 7 XML 2000/Markup Technologies 2000, Washington, DC. For information, visit `http://www.gca.org/attend/2000_conferences/XML_2000/`.

2001

Feb 28 – Mar 3 DANTE 2001, 24th meeting, Fachhochschule Rosenheim, Germany. For information, visit `http://www.dante.de/dante2001/`.

Mar 1 **TUG 2001 election**, nominations due to TUG office

Apr 29 – May 1 BachoTeX 2001, 9th annual meeting of the Polish TeX Users' Group (GUST), "Contemporary publishing TeXnology", Bachotek, Brodnica Lake District, Poland. For information, visit `http://www.gust.org.pl/BachoTeX/`.

May 14 – 17 Congrès GUTenberg 2001, "Le document au XXIe Siècle", Metz, France. For information, visit `http://www.gutenberg.eu.org/manif/gut2001/`.

Jun 6 – 8 Society for Scholarly Publishing, 23rd annual meeting, San Francisco, California. For information, visit `http://www.sspnet.org`.

Jun 13 – 17 ACH/ALLC 2001: Joint International Conference of the Association for Computers and the Humanities, and Association for Literary and Linguistic Computing, New York University, New York. For information, visit `http://www.nyu.edu/its/humanities/ach_allc2001/`.

Aug 12 – 16 **TUG 2001** — The 22nd annual meeting of the TeX Users Group, "2001L A TeX Live Odyssey, University of Delaware, Newark, Delaware. For information, visit `http://www.tug.org/tug2001/`.

Status as of 30 September 2000

For additional information on TUG-sponsored events listed above, contact the TUG office (+1 503 223-9994, fax: +1 503 223-3960, e-mail: `office@tug.org`). For events sponsored by other organizations, please use the contact address provided.

Additional type-related events and news items are listed in the Sans Serif Web pages, at `http://www.quixote.com/serif/sans`.

Aug 12 – 17 SIGGRAPH 2001, Los Angeles, California. For information, visit `http://www.siggraph.org/s2001/`.

Sep 23 – 27 EuroTeX 2001, "TeX and Meta: the Good, the Bad and the Ugly Bits", Kerkrade, Netherlands. For information, visit `http://www.ntg.nl/eurotex/`.

Oct 24 – 26 4th International Conference on The Electronic Document, Toulouse, France. For information, visit `http://www.irit.fr/CIDE2001/`.

Institutional Members

American Mathematical Society,
Providence, *Rhode Island*

Center for Computing Services,
Bowie, *Maryland*

CNRS - IDRIS,
Orsay, *France*

College of William & Mary,
Department of Computer Science,
Williamsburg, *Virginia*

CSTUG, *Praha*, *Czech Republic*

Florida State University,
Supercomputer Computations
Research, *Tallahassee*, *Florida*

Hong Kong University of
Science and Technology,
Department of Computer Science,
Hong Kong, *China*

IBM Corporation,
T J Watson Research Center,
Yorktown, *New York*

ICC Corporation,
Portland, *Oregon*

Institute for Advanced Study,
Princeton, *New Jersey*

Institute for Defense Analyses,
Center for Communications
Research, *Princeton*, *New Jersey*

Iowa State University,
Computation Center,
Ames, *Iowa*

Kluwer Academic Publishers,
Dordrecht, *The Netherlands*

KTH Royal Institute of
Technology, *Stockholm*, *Sweden*

Marquette University,
Department of Mathematics,
Statistics and Computer Science,
Milwaukee, *Wisconsin*

Masaryk University,
Faculty of Informatics,
Brno, *Czechoslovakia*

Max Planck Institut
für Mathematik,
Bonn, *Germany*

National Institute for Child
& Human Development,
Bethesda, *Maryland*

New York University,
Academic Computing Facility,
New York, *New York*

Princeton University,
Department of Mathematics,
Princeton, *New Jersey*

Space Telescope Science Institute,
Baltimore, *Maryland*

Springer-Verlag Heidelberg,
Heidelberg, *Germany*

Springer-Verlag New York, Inc.,
New York, *New York*

Stanford Linear Accelerator
Center (SLAC),
Stanford, *California*

Stanford University,
Computer Science Department,
Stanford, *California*

Stockholm University,
Department of Mathematics,
Stockholm, *Sweden*

University of Canterbury,
Computer Services Centre,
Christchurch, *New Zealand*

University College, Cork,
Computer Centre,
Cork, *Ireland*

University of Delaware,
Computing and Network Services,
Newark, *Delaware*

Universität Koblenz–Landau,
Fachbereich Informatik,
Koblenz, *Germany*

University of Oslo,
Institute of Informatics,
Blindern, *Oslo*, *Norway*

Università degli Studi di Trieste,
Trieste, *Italy*

Vanderbilt University,
Nashville, *Tennessee*

Vrije Universiteit,
Amsterdam, *The Netherlands*

TeX Consulting & Production Services

Information about these services can be obtained from:

TeX Users Group
1466 NW Naito Parkway, Suite 3141
Portland, OR 97209-2820, U.S.A.
Phone: +1 503 223-9994

Fax: +1 503 223-3960
Email: office@tug.org
URL: http://www.tug.org/consultants.html

North America

Hargreaves, Kathryn
135 Center Hill Road,
Plymouth, MA 02360-1364;
(508) 224-2367; letters@cs.umb.edu

I write in TeX, LaTeX, METAFONT, MetaPost, PostScript, HTML, Perl, Awk, C, C++, Visual C++, Java, JavaScript, and do CGI scripting. I take special care with mathematics. I also copyedit, proofread, write documentation, do spiral binding, scan images, program, hack fonts, and design letterforms, ads, newsletters, journals, proceedings and books. I'm a journeyman typographer and began typesetting and designing in 1979. I coauthored *TeX for the Impatient* (Addison-Wesley, 1990) and some psychophysics research papers. I have an MFA in Painting/Sculpture/Graphic Arts and an MSc in Computer Science. Among numerous other things, I'm currently doing some digital type and human vision research, and am a webmaster at the Department of Engineering and Applied Sciences, Harvard University. For more information, see: http://www.cs.umb.edu/ kathryn.

Loew, Elizabeth
President, TeXniques, Inc.,
675 Massachusetts Avenue, 6th Floor,
Cambridge, MA 02139;
(617) 876-2333; Fax: (781) 344-8158
Email: loew@texniques.com

Complete book and journal production in the areas of mathematics, physics, engineering, and biology. Services include copyediting, layout, art sizing, preparation of electronic figures; we keyboard from raw manuscript or tweak TeX files.

Ogawa, Arthur
40453 Cherokee Oaks Drive,
Three Rivers, CA 93271-9743;
(209) 561-4585
Email: Ogawa@teleport.com

Bookbuilding services, including design, copyedit, art, and composition; color is my speciality. Custom TeX macros and LaTeX2ε document classes and packages. Instruction, support, and consultation for workgroups and authors. Application development in LaTeX, TeX, SGML, PostScript, Java, and ßC++. Database and corporate publishing. Extensive references.

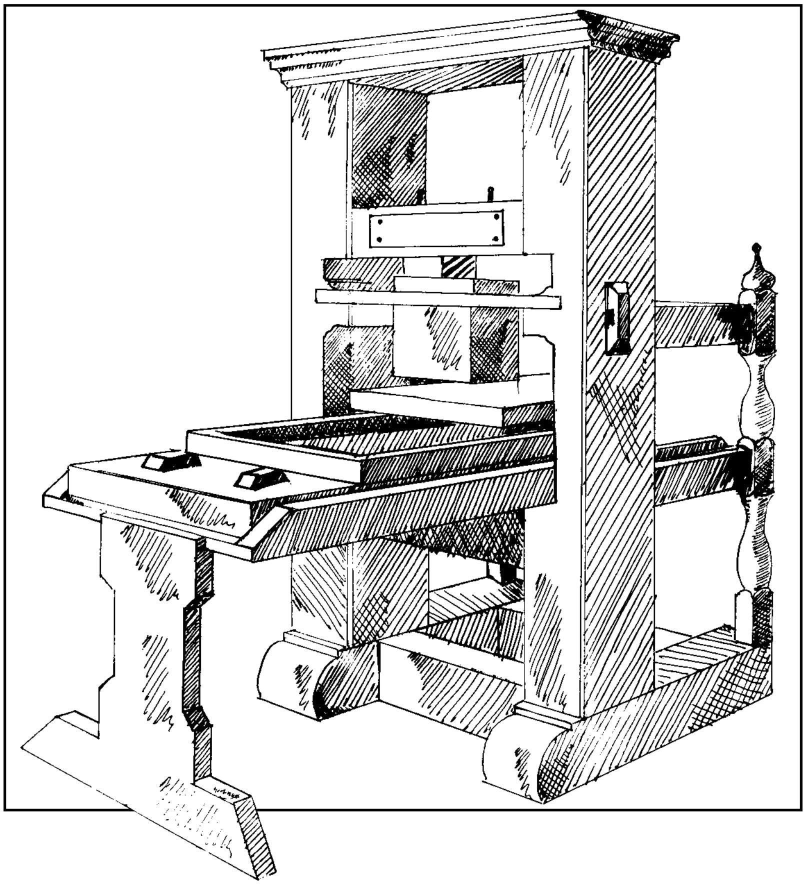
Outside North America

DocuTeXing: TeX Typesetting Facility
43 Ibn Kotaiba Street,
Nasr City, Cairo 11471, Egypt
+20 2 4034178; Fax: +20 2 4034178
Email: main-office@DocuTeXing.com

DocuTeXing provides high-quality TeX and LaTeX typesetting services to authors, editors, and publishers. Our services extend from simple typesetting and technical illustrations to full production of electronic journals. For more information, samples, and references, please visit our web site: http://www.DocuTeXing.com or contact us by e-mail.

TUGboat

The Communications of the TEX Users Group



Volume 21, Number 2, June 2000

Sir Benjamin Backbite: … I think you will like them, when you shall see them on a beautiful quarto page, where a neat rivulet of text shall meander through a meadow of margin.

Richard Brinsley Sheridan
The School for Scandal (1777)

TUGBOAT

COMMUNICATIONS OF THE TEX USERS GROUP

EDITOR BARBARA BEETON

VOLUME 21, NUMBER 2 • JUNE 2000
PORTLAND • OREGON • U.S.A.

TUGboat

Volume 21, Number 2 / June 2000

TeX Users Group

Memberships and Subscriptions

TUGboat (ISSN 0896-3207) is published quarterly by the TeX Users Group, 1466 NW Naito Parkway, Suite 3141, Portland, OR 97209-2820, U.S.A.

2000 dues for individual members are as follows:

- Ordinary members: $75.
- Students: $45.

Membership in the TeX Users Group is for the calendar year, and includes all issues of *TUGboat* for the year in which membership begins or is renewed. Individual membership is open only to named individuals, and carries with it such rights and responsibilities as voting in TUG elections. A membership form is provided on page 150.

TUGboat subscriptions are available to organizations and others wishing to receive *TUGboat* in a name other than that of an individual. Subscription rates: $85 a year, including air mail delivery.

Periodical-class postage paid at Portland, OR, and additional mailing offices. Postmaster: Send address changes to *TUGboat*, TeX Users Group, 1466 NW Naito Parkway, Suite 3141, Portland, OR 97209-2820, U.S.A.

Institutional Membership

Institutional Membership is a means of showing continuing interest in and support for both TeX and the TeX Users Group. For further information, contact the TUG office (`office@tug.org`).

Printed in U.S.A.

Addresses

General correspondence,
payments, etc.
TeX Users Group
P. O. Box 2311
Portland, OR 97208-2311
U.S.A.

Delivery services,
parcels, visitors
TeX Users Group
1466 NW Naito Parkway
Suite 3141
Portland, OR 97209-2820
U.S.A.

Telephone

+1 503 223-9994

Fax

+1 503 223-3960

Electronic Mail

(Internet)

General correspondence,
membership, subscriptions:
`office@tug.org`

Submissions to *TUGboat*,
letters to the Editor:
`TUGboat@tug.org`

Technical support for
TeX users:
`support@tug.org`

To contact the
Board of Directors:
`board@tug.org`

World Wide Web

`http://www.tug.org/`
`http://www.tug.org/TUGboat/`

Problems not resolved?

The TUG Board wants to hear from you:
Please email to `board@tug.org`

TUGboat

During 2000, the communications of the TeX Users Group will be published in four issues. The September issue (Vol. 21, No. 3) will contain the Proceedings of the 2000 TUG Annual Meeting.

TUGboat is distributed as a benefit of membership to all members.

Submissions to *TUGboat* are reviewed by volunteers and checked by the Editor before publication. However, the authors are still assumed to be the experts. Questions regarding content or accuracy should therefore be directed to the authors, with an information copy to the Editor.

Submitting Items for Publication

The next regular issue will be Vol. 21, No. 4. As production and mailing have been delayed, deadlines for contributions have already passed. Mailing is scheduled for late December. The third issue for 2000 is expected to be sent to the printer very late in November. Deadlines for other future issues are listed in the Calendar, page 144.

Manuscripts should be submitted to a member of the *TUGboat* Editorial Board. Articles of general interest, those not covered by any of the editorial departments listed, and all items submitted on magnetic media or as camera-ready copy should be addressed to the Editor, Barbara Beeton, or to the Production Manager, Mimi Burbank (see addresses on p. 99).

Contributions in electronic form are encouraged, via electronic mail, on diskette, or made available for the Editor to retrieve by anonymous FTP; contributions in the form of camera copy are also accepted. The *TUGboat* "style files", for use with either `plain` TeX or LaTeX, are available from CTAN. For authors who have no network FTP access, they will be sent on request; please specify which is preferred. Send e-mail to `TUGboat@tug.org`, or write or call the TUG office.

This is also the preferred address for submitting contributions via electronic mail.

Reviewers

Additional reviewers are needed, to assist in checking new articles for completeness, accuracy, and presentation. Volunteers are invited to submit their names and interests for consideration; write to `TUGboat@tug.org` or to the Editor, Barbara Beeton (see address on p. 99).

TUGboat Advertising and Mailing Lists

For information about advertising rates, publication schedules or the purchase of TUG mailing lists, write or call the TUG office.

TUGboat Editorial Board

Production Team:

See page 99 for addresses.

Other TUG Publications

TUG publishes the series *TeXniques*, in which have appeared reference materials and user manuals for macro packages and TeX-related software, as well as the Proceedings of the 1987 and 1988 Annual Meetings. Other publications on TeXnical subjects also appear from time to time.

TUG is interested in considering additional manuscripts for publication. These might include manuals, instructional materials, documentation, or works on any other topic that might be useful to the TeX community in general. Provision can be made for including macro packages or software in computer-readable form. If you have any such items or know of any that you would like considered for publication, send the information to the attention of the Publications Committee at `tug-pub@tug.org` or in care of the TUG office.

Trademarks

Many trademarked names appear in the pages of *TUGboat*. If there is any question about whether a name is or is not a trademark, prudence dictates that it should be treated as if it is. The following list of trademarks which appear in this issue may not be complete.

MS/DOS is a trademark of Microsoft Corporation
METAFONT is a trademark of Addison-Wesley Inc.
PC TeX is a registered trademark of Personal TeX, Inc.
PostScript is a trademark of Adobe Systems, Inc.
techexplorer is a trademark of IBM Research.
TeX and AMS-TeX are trademarks of the American Mathematical Society.
Textures is a trademark of Blue Sky Research.
UNIX is a registered trademark of X/Open Co. Ltd.