No attempt will be made here to describe modern English usage in the matter of hyphens; its infinite variety defies description. ... There is, however, one principle that seems to command at least lip service from all authorities. That is that the hyphen is not an ornament but an aid to being understood, and should be employed only when it is needed for that purpose.

H. W. Fowler,
A Dictionary of
Modern English Usage,
Second edition, revised by
Sir Ernest Gowers,
Oxford University Press, 1965, p. 255

TUGBOAT

THE TEX USERS GROUP NEWSLETTER

EDITOR BARBARA BEETON

VOLUME 6, NUMBER 2 • JULY, 1985
PROVIDENCE • RHODE ISLAND • U.S.A.

Addresses of Officers, Authors and Others

AZZARELLO, Arlene
I P Sharp Associates
220 California Ave
Suite 201
Palo Alto, CA 94306
415-327-1700

BECK, Lawrence A.
Grumman Data Systems
R & D, MS B13-237
Woodbury, NY 11797
516-682-8478

BEETON, Barbara
American Mathematical Society
P.O. Box 6248
Providence, RI 02940
401-272-9500
bb@SU-AI

BLOCK, Neil
Hughes Aircraft Co.
Bldg. A1, M/S 3C923
P.O. Box 9339
Long Beach, CA 90810
213-513-4891

BONNETAIN, Jean-Luc
Arpanet: Bonnetain@SUMEX-AIM

BROWN, Heather
Computing Laboratory
Univ of Kent at Canterbury
Kent CT2 7NF, England
(0)227-66822 x7550

CARNES, Lance
163 Linden Lane
Mill Valley, CA 94941
415-388-8853

CHILDS, S. Bart
Dept of Computer Science
Texas A & M University
College Station, TX 77843
409-845-5470

CODE, Maria
Data Processing Services
1371 Sydney Dr
Sunnyvale, CA 94087

CRAWFORD, John M.
Computing Services Center
College of Administrative Science
Ohio State University
Columbus, OH 43210
614-422-1741
CSNet: Crawford-J@Ohio-State
BITNet: TS0135@OHSTVMA

DUPREE, Chuck
Datapoint, Inc.
9725 Datapoint Drive, MS N-22
San Antonio, TX 78284
512-699-7200

EPPSTEIN, Maureen
Administrative Publication Manager
Stanford University
Encina Hall, Room 200
Stanford, CA 94305
415-497-9254
Arpanet: MVEppstein@SU-Score

FERGUSON, Michael J.
INRS - Télécommunications
Univ du Québec
3 Place du Commerce
Verdun (H3E 1H6), Québec, Canada
514-768-6691

FUCHS, David
Department of Computer Science
Stanford University
Stanford, CA 94305
415-497-1646
DRF@SU-Score

FURUTA, Richard
Univ of Washington
Computer Science, FR-35
Seattle, WA 98195
206-543-7798
Furuta@Washington

GOUCHER, Raymond E.
TEX Users Group
P.O. Box 9506
Providence, RI 02940
401-272-9500 x232

GREENE, John
GA Technologies
P O Box 85608
San Diego, CA 92138

GROPP, William
Dept of Computer Science
Yale University
Box 2158 Yale Station
New Haven, CT 06520
203-436-3761
Arpanet: Gropp@Yale

HENDRICKSON, Amy
57 Longwood Ave #8
Brookline, MA 02146
617-738-8029

ION, Patrick D.
Mathematical Reviews
416 Fourth Street
P.O. Box 8604
Ann Arbor, MI 48107
313-763-6829

JÜRGENSEN, Helmut
Dept of Computer Science
Univ of Western Ontario
London N6A 5B7, Ontario, Canada
519-679-3039
Usenet: uwo!uwo-hobbit!jurgensen

KELLER, Arthur
University of Texas at Austin
Department of Computer Science
Austin, TX 78712

KELLY, Bill
Academic Computing Center
University of Wisconsin, Madison
1210 W. Dayton Street
Madison, WI 53706
608-262-9501

KNUTH, Donald E.
Department of Computer Science
Stanford University
Stanford, CA 94305
DEK@SU-AI

LØFSTEDT, Benedict
RECAU, Build. 540
Ny Munkegade
8000 Aarhus C, Denmark
06-128355

MacKAY, Pierre A.
University of Washington
Department of Computer Science, FR-35
Seattle, WA 98195
206-543-2386
MacKay@Washington

MALLETT, Rick
Computing Services
Room 1208 Arts Tower
Carleton University
Ottawa (K1S 5B6), Ontario Can
613-231-7145

MARRIOTT, Gregory
Thermodynamics Research Center
Texas A & M University
College Station, TX 77843
409-845-4940

MARSH, Ken
Thermodynamics Research Center
Texas A & M University
College Station, TX 77843
409-845-4940

MOHR, August
2670 Lode Street
Santa Cruz, CA 95061
408-475-9711

NAUGLE, Norman W.
P O Box 2736
Texas A & M University
College Station, TX 77841
409-845-3104

NICHOLS, Monte C.
Exploratory Chemistry Division
Sandia National Laboratories 8313
Livermore, CA 94550
415-422-2906

PALAIS, Richard S.
Department of Mathematics
Brandeis University
Waltham, MA 02154
617-647-2667

PIZER, Arnold
Department of Mathematics
University of Rochester
Rochester, NY 14627
716-275-4428

PLASS, Susan
Information Technology Services
Cypress Hall, Jordan Quadrangle
Stanford University
Stanford, CA 94305
415-497-3302

RIESEL, Hans
NADA
KTH (Royal Inst of Tech)
S-100 44 Stockholm, Sweden

ROKICKI, Tom
P O Box 2736
Texas A & M University
College Station, TX 77841
409-845-3104

SMITH, Barry
Kellerman & Smith
534 SW Third Ave
Portland, OR 97204
503-222-4234

SPIVAK, Michael
1660 West Alabama, #7
Houston, TX 77006

SPRAGENS, Alan
SLAC Computing Services
Stanford Linear Accelerator Ctr
Bin 97, P O Box 4349
Stanford, CA 94305
415-854-3300 x2849

STROMQUIST, Ralph
MACC
University of Wisconsin
1210 W. Dayton Street
Madison, WI 53706
608-262-8821

THEDFORD, Rilla
Intergraph Corporation
One Madison Industrial Park
Huntsville, AL 35807
205-772-2000

TUTTLE, Joey K.
I P Sharp Associates
220 California Avenue
Suite 201
Palo Alto, CA 94306
415-327-1700

WELLAND, Robert
Dept of Mathematics
Northwestern University
Lunt Hall
Evanston, IL 60201
312-492-3298

WHIDDEN, Samuel B.
American Mathematical Society
P.O. Box 6248
Providence, RI 02940
401-272-9500

ZABALA, Ignacio
Centro de Cálcolo
Facultades de Ciencias
Universidad de Valencia
Cametera de Ademuz
Valencia, Spain
011-34-6-357-4065

ZAPF, Hermann
Seitersweg 35
D-6100 Darmstadt, Fed Rep Germany

TUGboat, the newsletter of the TEX Users Group (TUG), is published irregularly by TUG at Providence, RI, and is distributed as a benefit of membership both to individual and institutional members. Manuscripts should be submitted to a member of the TUGboat Editorial Committee, whose names and addresses are listed inside the front cover. Articles of general interest, those not covered by any of the editorial departments listed, and all items submitted on magnetic tape or as camera-ready copy should be addressed to the Editor, Barbara Beeton, at the general correspondence address. Submissions to TUGboat are for the most part reproduced with minimal editing, and any questions regarding content or accuracy should be directed to the authors, with an information copy to the Editor.

All general correspondence to TUG should be addressed to TUG, P. O. Box 9506, Providence, RI 02940. All payments to TUG should be addressed to TUG, P. O. Box 594, Providence, RI 02901. Parcel post and items sent by delivery services should be addressed to TUG, c/o American Mathematical Society, 201 Charles St., Providence, RI 02904.

TUG Membership Dues and Privileges

Memberships and Subscriptions

1985 dues for individual members are as follows:

North America:
- New (first-time) members or subscribers: $20.
- Membership and subscription renewals: $30,
  reduced rate of $20 for renewals received before January 31, 1985.

Outside North America (includes air mail postage):
- New (first-time) members or subscribers: $25.
- Membership and subscription renewals: $35,
  reduced rate of $25 for renewals received before January 31, 1985.

Membership privileges include all issues of TUGboat published during the membership (calendar) year. Anyone inquiring about TUG will be sent a complimentary issue of TUGboat, along with a current copy of the membership list and forms for acquiring TeX82, joining TUG and ordering publications available from TUG.

Issues to addresses in the United States are mailed third class bulk, which may take up to six weeks to reach their destinations. If you have not received an issue to which you are entitled, write to TUG at the address given below.

Institutional Membership

1985 Institutional Membership dues for educational organizations are $200; for non-educational, $300. Membership privileges include:
- designating up to 5 persons as individual members,
- special reduced rates for participation at TUG meetings and
  TeX-related courses and for purchase or lease of videotapes.

In addition, institutional members are acknowledged in each issue of TUGboat. For further information, call Ray Goucher at (401) 272-9500, ext. 232.

Submitting Items for Publication in TUGboat

The deadline for submitting items for Vol. 6 (1985), No. 3, *will be* September 15, 1985; *the issue will be mailed early in* November. Contributions on magnetic tape or in camera copy form are encouraged; see "Submitting items to TUGboat", page 78, Vol. 5, No. 2. Editorial addresses are given on the inside front cover. For instructions on preparing magnetic tapes or for transferring items directly to the AMS computer, write or call Barbara Beeton at the address given, (401) 272-9500, ext. 299.

TUGboat Advertising and Mailing Lists

For information about advertising rates or the purchase of TUG mailing lists, write or call the TeX Users Group, Attention: Ray Goucher, P. O. Box 9506, Providence, RI 02940, (401) 272-9500, ext. 232.

TUG 1985 Annual Meeting
Stanford University, August 12-14, 1985

A preliminary list of speakers and topics has been released by Arlene Azzarello, who is handling program arrangements. The times of the presentations have not yet been set.

John S. Gourlay, Ohio State University: "Music printing using TeX". Many interesting problems are presented by music, including the construction of fonts; spacing and "casting off"; the construction of beams, slurs, and other very large musical symbols. This presentation will report progress on a TeX-based public domain music printing system.

Kevin J. Small, COS Information, Montreal: "TeX in a commercial production environment". This presentation will touch upon font creation for the Xerox 9700, Autologic μ-5 and Apollo workstation; device drivers; a menu approach to TeX macro creation.

Carl H. Smith, University of Maryland: "Interfacing *refer* with TeX". *Refer* is a program that accesses a data base of bibliographic information and produces *troff* macros; the interface was achieved with a Pascal program which transforms the *troff* macros into TeX macros.

Richard Southall, Stanford University: "Making a typeface with new METAFONT: problems and some solutions"

Question-and-answer and birds-of-a-feather sessions will also be held on a more *ad hoc* basis. Site Coordinators are expected to report on the status of the various implementations of TeX and METAFONT now in general distribution. Other presentations that are likely to be included are: introduction to TeX and TUG for new users; TeX82 news; TeX82 and WEB user experiences; Macro Wizards' roundtable; output devices and drivers; commercially available macro packages. And, as is customary, representatives of output device manufacturers will be on hand.

There is still room in the program for a few last-minute entries. Suggestions and volunteers should be made known, as soon as possible, to

Arlene Azzarello
I. P. Sharp Associates, Inc.
Suite 201, 220 California Avenue
Palo Alto, CA 94306-1683
(415) 327-1700

Software

Helmut Jürgensen
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A 5B7
Usenet: *uwo!uwo-hobbit!jurgensen*
519-679-3039

The software column has received no input recently. However, I have been promised — or let's be moderate: semi-promised — some contributions which look really interesting for the near future. But I should also like to receive papers for this column now. Please, if you have material which you think might go into this section of the TUGboat send it to me or contact me about it. Contributions I am looking for would include

- long or short papers on software related to computer type-setting in general, and to TeX, METAFONT, and WEB in particular,
- comments and questions about such software,
- announcements of the availability of such software, and
- suggestions concerning topics to be addressed in this column.

If in doubt about the suitability of a potential contribution why not contact me by mail, phone, or computer mail?

Right now, I am planning a survey of implementations which contain facilities for (pre-)viewing and interactive type-setting. This seems to be a topic of considerable interest to the TUGboat readers. If you have a contribution concerning this problem I should be happy to consider it for publication in this column.

But, of course there are also many other areas where specific software support is required to make TeX or METAFONT available or usable in a particular environment. I should like to receive contributions of this kind as well. And again, contributions concerning WEB are also welcome for this column.

You should send your contributions to the address given above. Phone calls or computer mail are welcome too.

A Multilingual T$\hat{\text{E}}$X

Michael J. Ferguson
INRS-Télécommunications
Montréal, Canada

This note details an extension to TEX that allows for multilingual hyphenation using **standard TEX fonts**, including words with accented letters. Switching between languages within a document is sufficiently simple and efficient that it could even be done on a word by word basis. Although it really has not been given a name, perhaps it could be called "T$\hat{\text{E}}$X".

Briefly the features of the extension are:

- A new **primitive** integer parameter **\language** has been introduced. The value of this parameter controls the set of language `\patterns` and hyphenation exceptions actually in force when hyphenation is attempted.
- Hyphenation exceptions are language dependent.
- Words with accents, such as "l'épicerie", will be hyphenated correctly. These modifications of TEX will work, with accented letters such as "é" built using TEX's accent primitive or resident in the font. In addition they will work whether the letter is accessed as a single character or as a ligature.
- `\lccode` 1 and 2 are now used to indicate which characters are accents. Note that `\lccode` 0 indicates a non letter.
- Theoretically any number of languages could be used. The only problem, at this point is a restriction to 256 of one of TEX's tables (the `trie_op` for those that know). The bilingual (French/English) running at INRS-Télécommunications uses, 238 of 256 of these table locations. French alone uses 108 and English alone 181.
- The changes are upward compatible — a standard `plain.tex` can be built into a format file by a modified `initex`. However, the format file has been modified so that a non-extended `plain.fmt` will cause a "fatal format error" if used with the extended TEX.

Some restrictions are as follows:

- Discretionary hyphenation spellings, as required in German, are not automatically included. However, it is felt that these could be added, in special format to the `\patterns` and handled during hyphenation much as ligatures.
- Accents must be in the same font as the characters in the word to be hyphenated. It is not clear whether this is an important restriction.
- A new value of **\language** determines both a set of hyphenation patterns and exceptions. There is no provision for using an additional set of hyphenation exceptions with an already existing set of patterns. For instance, if it was really important that "Random House" hyphenation be used rather than "Websters", a set of patterns for both would be required.

To change hyphenation rules it is only necessary to change the value of **\language**. However, since accents and certain characters may be legitimate in one language and not others, it may also be desirable to modify certain `\lccode`s. There are checks in the modifications to prevent disasters if **\language** is somehow not within the range allowed.

Modifications for Hyphenating Words with Accented Characters

The basic idea is as follows:

- Designate accents with `\lccode` of 1 or 2.
- Make accent kerns implicit so that they disappear before the word is sent to the hyphenation routine.
- Reconstitute the accents after the hyphens are returned from the hyphenation routine.

The net effect of this is that hyphenation patterns will be applied to words involving accents. This means, for instance, that the word "envôuterions" has the hyphens "en-vôute-ri-ons" if the English patterns in Plain are used, but is hyphenated as "en-vôu-te-rions" if the French patterns are used. Note that the English patterns inserts one incorrect hyphen and misses another. In addition there will never be a hyphen inserted between an accent and its following character since that case has never been given an odd number.

Two `\lccodes` were used to allow for different placement of the same accent symbol – for instance above or below the accented character. Initially it was thought that the cedilla " ¸ " would require such special treatment but that turned out not to be the case. The second value could be used for language dependent accent placements. However this is of limited utility at the moment as there is no way, other than recompiling TEX to modify the accent placement routine.

There are a few restrictions with respect to accents and hyphenations.

- The accent must come from the same font as the accented character.
- It is not possible to accent ligatures.
- It is not possible to put on more than one accent.
- Accents placed by raising or lowering boxes cannot be hyphenated. This means that the cedilla in some fonts may prevent hyphenation.*

Comments and Caveats

There are several extensions possible. The most obvious, and one that is probably necessary, is the introduction of discretionary spellings involving hyphenation ... such as those that occur in German. It appears to this author that the rules could be placed in the hyphenation patterns, and the invocation handled much like the multiple choices involved in ligatures. This was not done because of a lack of precise understanding of what was required.

Finally, there is one data structure in short supply. This is the `trie_op` that is one byte. The combined French and English patterns use 238 of 256 possible values. This suggests that for several languages the trie will have to be modified.

The changes to TEX have been rather simple, involving precise rather than massive surgery. The fact that it could be done at all amazes this author. The credit for this lies with squarely with Don Knuth. There are two major reasons. The first is obvious, to those who know, and is due to the incredible level of documentation possible in WEB. However, WEB by itself does not guarantee a well designed program. TEX, quite simply, is well designed. Although there is ample opportunity for obscure data dependencies, they do not seem to occur. Several times this author was worried about problems that did not exist. Under normal circumstances, they might have been there.

---

* Interestly the umlaut, "ö" appears to be incorrectly placed in the most fonts for (nice) use as an accent, at least in French. It is not clear why this is so.

Fonts

MATHEMATICAL SYMBOLS AND CYRILLIC FONTS READY FOR DISTRIBUTION

Barbara Beeton
American Mathematical Society

The first general release of fonts from the Euler series (named for Leonhard Euler, the great eighteenth-century Swiss mathematician) will be made in August, at about the time of the TUG meeting. This first release will consist of cyrillic and two 128-character fonts of mathematical symbols, all in various sizes and weights. It is our intention that these fonts be added to the standard distribution, and we will make an effort to provide the necessary files to all sites from which the TeX package is being distributed. It will not be possible to provide this material directly to users, since the Society's DEC 20 computer has proved singularly unsuitable for making tapes that can be read by any other kind of machine.

A master tape will be delivered to Stanford by one of the AMS attendees at the TUG meeting. This tape will contain the **METAFONT78** sources necessary to generate the fonts in distribution format, along with several files of macros, documentation and user instructions. Actual addition of this material to distribution tapes (and PC diskettes) has not yet been fully arranged, but it should be possible to announce these arrangements at the TUG meeting.

Cyrillic

The cyrillic font contains all letters found in the modern (post-revolutionary) cyrillic alphabet, as well as others found by Mathematical Reviews to be necessary for rendering bibliographic information in Russian, Ukrainian, Serbian, Georgian, and other Slavic and non-Slavic languages ordinarily published in cyrillic. Accents which normally occur in these languages, as well as in such words as names of mathematicians whose work is regularly translated

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	Њ	Љ	Џ	Э	I	Є	Ђ	Ћ	"0x
'01x	њ	љ	џ	э	i	є	ђ	ћ	
'02x	Ю	Ж					S	Я	"1x
'03x	ю	ж					s	я	
'04x	¨	!	”		˘	%	´	’	"2x
'05x	(	)	*		,	-	.	/	
'06x	0	1	2	3	4	5	6	7	"3x
'07x	8	9	:	;	«	ı	»	?	
'10x		А	Б	Ц	Д	Е	Ф	Г	"4x
'11x	Х	И	J	К	Л	М	Н	О	
'12x	П	Ч	Р	С	Т	У	В	Щ	"5x
'13x	Ш	Ы	З	[	“	]	Ь	Ъ	
'14x	‘	а	б	ц	д	е	ф	г	"6x
'15x	х	и	j	к	л	м	н	о	
'16x	п	ч	р	с	т	у	в	щ	"7x
'17x	ш	ы	з	–	—	№	ь	ъ	
	"8	"9	"A	"B	"C	"D	"E	"F	

The AMS Euler cyrillic font – ECYR10

Мещанский университет, находящийся на пути к Нескучному, праздновал на днях свой пятидесятилетний юблией. Кого возили в Титы или городскую больницу, тот, конечно, помнит здоровеннейший, трёхэтажный домище по правую руку с вывеской «Богадельная и Мещанские училища» и тому наверное встречались на пути вереницы ученических пар, солидно прогуливаемых надзирателями.

into Russian or one of the other languages covered by MR, are included in the font, as are all the digits and ordinary punctuation. Several cells are still empty: the number of such cells is not sufficient to hold all the additional pre-revolutionary Russian letters, and there is not yet enough experience to indicate what else might most usefully (for MR) be included.

The "basic" cyrillic font is ECYR10. Names have been assigned to a number of variations, not all of which exist yet. (In particular, there are no plans yet to create the METAFONT descriptions of the true "italic" letters.)

ECYR	lightface	ECSL	slanted
ECB	bold	ECBSL	bold slanted
ECBX	bold extended	ECSS	sans serif
ECI	italic	ECSSB	bold sans serif
ECBI	bold italic		

Font names have been assigned so that compression to 6 characters, using the first 3 and last 3 letters of longer names (a standard built into most implementations of TeX for operating systems having such a limit, and announced through TeXhax by David Fuchs), will always be unique.

Keying of cyrillic to be rendered with this font is in accord with the current MR transliteration scheme, e.g.

Khrushchëv (`Khrushch\"ev`) → Хрущёв
Zhurnal (`Zhurnal`) → Журнал
Kiïv (`Ki{\"\i}v`) → Киïв

`\font\tencyr=ecyr10 \def\cyr{\tencyr\cyracc}` `{\cyr ...}` sets the stage for proper transition to and from cyrillic. Most of the translation from keyed input to cyrillic is implemented by ligature instructions in the font itself. A few letters require a "chain" of ligatures: sh → ш, shc → 7 (an obvious absurdity, but the roman combination never occurs legitimately), shch → щ. Letters rendered with accents in transliteration are trapped by macros defined in the file CYRACC.DEF: ĭ (`\u\i`) → й, ī (`\=\i}`) → i, ï (`\"\i`) → ï. Both the macro accent traps and ligatures do the right thing in ordinary roman text and in `{\cyr ...}`; however, cyrillic items in tables cannot reliably be specified in the preamble, and in individual cells `\cyr` should be preceded by `\relax` to prevent premature expansion, and thus loss, of the macro instructions.

Hyphenation is not automatically suppressed, but the patterns used will be those for English in the absence of a local override. (We do not whether any Russian patterns exist.) For short passages, as the above sample, or isolated words, good luck may prevail.

Documentation accompanying this font will include full keying instructions, the ligature specifications, and, of course, CYRACC.DEF. Should the transliteration scheme in local use be different from the MR scheme (for example, an earlier MR scheme rendered щ as šč), it should be quite easy to modify CYRACC.DEF to accommodate it, and, if ligature changes are absolutely necessary, they may be implemented using the TeXware programs TFtoPL and PLtoTF.

Mathematical symbols

Mathematicians expanding the boundaries of their chosen areas often find that no suitably unambiguous notation exists with which to express new concepts. First attempts usually consist in seeking out ever more exotic alphabets, but this fount is rather rapidly exhausted. Non-alphabetic symbols modeled after, or constructed from combinations of, existing ones is probably the next most profitable approach. And failure in either of those attempts may yield something truly new. In any event, the net result is proliferation of symbols beyond what is available to most ordinary typesetting systems.

The original symbol fonts, CMSY and CMEX (currently AMSY and AMEX), contain the most frequently used mathematical symbols, plus whatever else was needed for *The Art of Computer Programming*, volume 2, and other projects that Don Knuth was working on at the time. Many other symbols are in common use in other subfields of mathematics, and the AMS and MR found it necessary to construct them. We have now filled one entire "extra symbols" font and most of a second.

The naming scheme devised for these "extra symbols" fonts also leaves room for a third. "Medium" and "bold" refer to the weight, medium being matched to the weight of the "basic" Computer Modern symbols in the CMSY font.

EUXM	symbols 1 medium	EUXB	symbols 1 bold
EUYM	symbols 2 medium	EUYB	symbols 2 bold
EUZM	symbols 3 medium	EUZB	symbols 3 bold

Following the charts of the two symbol fonts, the symbols are listed by type, corresponding to the conventions of *The TeXbook*, Appendix F. For each symbol is shown the font (1 or 2) and hex location, the symbol itself, and the symbol name. Symbol names have been assigned in accordance with Don Knuth's principles, and reviewed by him, but all responsibility for errors and misinterpretations of his comments resides with the AMS and MR staff members who worked on the font development.

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x	⊡	⊞	⊠	□	■	·	◊	⧫	″0x
´01x	↻	↺	↕	⇋	⊟	⊩	⊪	⊨	
´02x	↠	↞	⇇	⇉	⇈	⇊	↾	⇂	″1x
´03x	↿	⇃	↣	↢	⇆	⇄	↰	↱	
´04x	⇝	↭	↫	↬	≗	≿	≳	⪆	″2x
´05x	⊸	∴	∵	≑	≜	≾	≲	⪅	
´06x	⪕	⪖	⋞	⋟	≼	≦	⩽	≶	″3x
´07x	∖	/	≓	≒	≽	≧	⩾	≷	
´10x	⊏	⊐	⊳	⊲	⊵	⊴	★	≬	″4x
´11x	▼	▶	◀	△	▽	▵	▲	▿	
´12x	≖	⋚	⋛	⪋	⪌	¥	⇛	⇚	″5x
´13x	✓	⊻	⊼	⩞	⋔	∡	∢	∝	
´14x	⌣	⌢	⋐	⋑	⋓	⋒	⋏	⋎	″6x
´15x	⋋	⋌	⫅	⫆	≏	≎	⋘	⋙	
´16x	⌜	⌝	Ⓡ	Ⓢ	∐	∔	∽	⋍	″7x
´17x	⌞	⌟	✠	C	T	⊚	⊛	⊝	
	″8	″9	″A	″B	″C	″D	″E	″F	

Extra symbols, font 1 – EUXM10

	´0	´1	´2	´3	´4	´5	´6	´7	
´00x	≨︀	≩︀	⩽̸	⩾̸	≮	≯	⊀	⊁	″0x
´01x	≨	≩	⪇̸	⪈̸	⪇	⪈	⋠	⋡	
´02x	⋨	⋩	⋦	⋧	≦̸	≧̸	⪵	⪶	″1x
´03x	⪹	⪺	⪉	⪊	≁	≇			
´04x	⊊︀	⊋︀	⫅̸	⫆̸	⫋	⫌	⫋︀	⫌︀	″2x
´05x	⊊	⊋	⊈	⊉	∦	∤	∤	∦	
´06x	⊬	⊮	⊭	⊯	⋭	⋬	⋪	⋫	″3x
´07x	↚	↛	⇍	⇏	⇎	↮	⋇	∅	
´10x	∄	A	B	C	D	E	F	G	″4x
´11x	H	I	J	K	L	M	N	O	
´12x	P	Q	R	S	T	U	V	W	″5x
´13x	X	Y	Z						
´14x								ð	″6x
´15x		ℶ	ℷ	ℸ	⋖	⋗	⋉	⋊	
´16x	∣	∥	∖	∼	≈	≊	⪸	⪷	″7x
´17x	↶	↷	Ϝ	ϰ		ℏ	ℏ	∍	
	″8	″9	″A	″B	″C	″D	″E	″F	

Extra symbols, font 2 – EUYM10

Lowercase Greek letters.

2 "7A	$\digamma$	\digamma	2 "7B	$\varkappa$	\varkappa			

Uppercase blackboard bold letters.

2 "41	$\mathbb{A}$	\bbfA		. . .		2 "5A	$\mathbb{Z}$	\bbfZ

Hebrew letters.

2 "69	$\beth$	\beth	2 "6A	$\gimel$	\gimel	2 "6B	$\daleth$	\daleth

Miscellaneous symbols of type Ord.

2 "7E	$\hbar$	\hbar	1 "38	$\backprime$	\backprime	1 "39	$\prime$	\varprime
2 "7D	$\hslash$	\hslash	1 "03	$\square$	\square	1 "04	$\blacksquare$	\blacksquare
2 "3F	$\varnothing$	\varnothing	1 "46	$\bigstar$	\bigstar	1 "4E	$\blacktriangle$	\blacktriangle
1 "73	$\circledS$	\circledS	1 "4F	$\triangledown$	\triangledown	1 "48	$\blacktriangledown$	\blacktriangledown
2 "40	$\nexists$	\nexists	1 "06	$\lozenge$	\lozenge	1 "07	$\blacklozenge$	\blacklozenge
1 "7B	$\complement$	\complement	1 "5D	$\measuredangle$	\measuredangle	1 "5E	$\sphericalangle$	\sphericalangle

Binary operations.

1 "75	$\dotplus$	\dotplus	1 "65	$\Cap$	\Cap	1 "7D	$\circledcirc$	\circledcirc
2 "72	$\smallsetminus$	\smallsetminus	1 "64	$\Cup$	\Cup	1 "7E	$\circledast$	\circledast
2 "6E	$\ltimes$	\ltimes	1 "66	$\curlywedge$	\curlywedge	1 "7F	$\circleddash$	\circleddash
2 "6F	$\rtimes$	\rtimes	1 "67	$\curlyvee$	\curlyvee	1 "01	$\boxplus$	\boxplus
1 "68	$\leftthreetimes$	\leftthreetimes	1 "5A	$\barwedge$	\barwedge	1 "0C	$\boxminus$	\boxminus
1 "69	$\rightthreetimes$	\rightthreetimes	1 "59	$\veebar$	\veebar	1 "02	$\boxtimes$	\boxtimes
2 "3E	$\divideontimes$	\divideontimes	1 "5B	$\doublebarwedge$	\doublebarwedge	1 "00	$\boxdot$	\boxdot
1 "7C	$\intercal$	\intercal	1 "05	$\centerdot$	\centerdot			

Relations.

1 "35	$\leqq$	\leqq	1 "3D	$\geqq$	\geqq	1 "2B	$\doteqdot$	\doteqdot
1 "36	$\leqslant$	\leqslant	1 "3E	$\geqslant$	\geqslant	1 "24	$\circeq$	\circeq
1 "30	$\eqslantless$	\eqslantless	1 "31	$\eqslantgtr$	\eqslantgtr	1 "50	$\eqcirc$	\eqcirc
1 "2E	$\lesssim$	\lesssim	1 "26	$\gtrsim$	\gtrsim	1 "2C	$\triangleq$	\triangleq
1 "2F	$\lessapprox$	\lessapprox	1 "27	$\gtrapprox$	\gtrapprox	1 "3A	$\risingdotseq$	\risingdotseq
2 "6C	$\lessdot$	\lessdot	2 "6D	$\gtrdot$	\gtrdot	1 "3B	$\fallingdotseq$	\fallingdotseq
1 "6E	$\lll$	\lll	1 "6F	$\ggg$	\ggg	1 "76	$\backsim$	\backsim
1 "37	$\lessgtr$	\lessgtr	1 "3F	$\gtrless$	\gtrless	1 "77	$\backsimeq$	\backsimeq
1 "51	$\lesseqgtr$	\lesseqgtr	1 "52	$\gtreqless$	\gtreqless	2 "73	$\thicksim$	\thicksim
1 "53	$\lesseqqgtr$	\lesseqqgtr	1 "54	$\gtreqqless$	\gtreqqless	2 "74	$\thickapprox$	\thickapprox
1 "6A	$\subseteqq$	\subseteqq	1 "6B	$\supseteqq$	\supseteqq	2 "75	$\approxeq$	\approxeq
1 "62	$\Subset$	\Subset	1 "63	$\Supset$	\Supset	1 "6C	$\bumpeq$	\bumpeq
1 "40	$\sqsubset$	\sqsubset	1 "41	$\sqsupset$	\sqsupset	1 "6D	$\Bumpeq$	\Bumpeq
1 "34	$\preccurlyeq$	\preccurlyeq	1 "3C	$\succcurlyeq$	\succcurlyeq	1 "47	$\between$	\between
1 "32	$\curlyeqprec$	\curlyeqprec	1 "33	$\curlyeqsucc$	\curlyeqsucc	1 "5C	$\pitchfork$	\pitchfork
1 "2D	$\precsim$	\precsim	1 "25	$\succsim$	\succsim	1 "5F	$\varpropto$	\varpropto
2 "77	$\precapprox$	\precapprox	2 "76	$\succapprox$	\succapprox	1 "74	$\amalg$	\smallamalg
1 "43	$\vartriangleleft$	var. \triangleleft	1 "42	$\vartriangleright$	var. \triangleright	1 "4A	$\blacktriangleleft$	\blacktriangleleft
1 "45	$\trianglelefteq$	\trianglelefteq	1 "44	$\trianglerighteq$	\trianglerighteq	1 "49	$\blacktriangleright$	\blacktriangleright
1 "0F	$\vDash$	\vDash	1 "0D	$\Vdash$	\Vdash	1 "0E	$\Vvdash$	\Vvdash
1 "60	$\smallsmile$	\smallsmile	2 "70	$\shortmid$	\shortmid	1 "29	$\therefore$	\therefore
1 "61	$\smallfrown$	\smallfrown	2 "71	$\shortparallel$	\shortparallel	1 "2A	$\because$	\because
2 "7F	$\backepsilon$	\backepsilon						

Negated relations.

2 "04	$\nless$	\nless	2 "05	$\ngtr$	\ngtr	2 "1C	$\nsim$	\nsim
2 "02	$\nleq$	\nleq	2 "03	$\ngeq$	\ngeq	2 "1D	$\ncong$	\napprox
2 "0A	$\nleqslant$	\nleqslant	2 "0B	$\ngeqslant$	\ngeqslant	2 "2E	$\nshortmid$	\nshortmid
2 "14	$\nleqq$	\nleqq	2 "15	$\ngeqq$	\ngeqq	2 "2F	$\nshortparallel$	\nshortparallel
2 "0C	$\lneq$	\lneq	2 "0D	$\gneq$	\gneq	2 "2D	$\nmid$	\nmid
2 "08	$\lneqq$	\lneqq	2 "09	$\gneqq$	\gneqq	2 "2C	$\nparallel$	\nparallel
2 "00	$\lvertneqq$	\lvertneqq	2 "01	$\gvertneqq$	\gvertneqq	2 "30	$\nvdash$	\nvdash
2 "12	$\lnsim$	\lnsim	2 "13	$\gnsim$	\gnsim	2 "32	$\nvDash$	\nvDash
2 "1A	$\lnapprox$	\lnapprox	2 "1B	$\gnapprox$	\gnapprox	2 "31	$\nVdash$	\nVdash
2 "06	$\nprec$	\nprec	2 "07	$\nsucc$	\nsucc	2 "33	$\nVDash$	\nVDash
2 "0E	$\npreceq$	\npreceq	2 "0F	$\nsucceq$	\nsucceq	2 "36	$\ntriangleleft$	\ntriangleleft
2 "16	$\precneqq$	\precneqq	2 "17	$\succneqq$	\succneqq	2 "37	$\ntriangleright$	\ntriangleright
2 "10	$\precnsim$	\precnsim	2 "11	$\succnsim$	\succnsim	2 "35	$\ntrianglelefteq$	\ntrianglelefteq
2 "18	$\precnapprox$	\precnapprox	2 "19	$\succnapprox$	\succnapprox	2 "34	$\ntrianglerighteq$	\ntrianglerighteq
2 "2A	$\nsubseteq$	\nsubseteq	2 "2B	$\nsupseteq$	\nsupseteq			
2 "22	$\nsubseteqq$	\nsubseteqq	2 "23	$\nsupseteqq$	\nsupseteqq			
2 "28	$\subsetneq$	\subsetneq	2 "29	$\supsetneq$	\supsetneq			
2 "20	$\varsubsetneq$	var. \subsetneq	2 "21	$\varsupsetneq$	var. \supsetneq			
2 "24	$\subsetneqq$	\subsetneqq	2 "25	$\supsetneqq$	\supsetneqq			
2 "26	$\varsubsetneqq$	var. \subsetneqq	2 "27	$\varsupsetneqq$	var. \subsetneqq			

Arrows.

1 "12	$\leftleftarrows$	\leftleftarrows	1 "13	$\rightrightarrows$	\rightrightarrows	1 "14	$\upuparrows$	\upuparrows
1 "1C	$\leftrightarrows$	\leftrightarrows	1 "1D	$\rightleftarrows$	\rightleftarrows	1 "15	$\downdownarrows$	\downdownarrows
1 "57	$\Lleftarrow$	\Lleftarrow	1 "56	$\Rrightarrow$	\Rrightarrow	1 "0A	$\updownarrow$	\updownarrow
1 "11	$\twoheadleftarrow$	\twoheadleftarrow	1 "10	$\twoheadrightarrow$	\twoheadrightarrow	1 "18	$\upharpoonleft$	\upharpoonleft
1 "1B	$\leftarrowtail$	\leftarrowtail	1 "1A	$\rightarrowtail$	\rightarrowtail	1 "19	$\downharpoonleft$	\downharpoonleft
1 "22	$\looparrowleft$	\looparrowleft	1 "23	$\looparrowright$	\looparrowright	1 "16	$\upharpoonright$	\upharpoonright
2 "78	$\curvearrowleft$	\curvearrowleft	2 "79	$\curvearrowright$	\curvearrowright	1 "17	$\downharpoonright$	\downharpoonright
1 "09	$\circlearrowleft$	\circlearrowleft	1 "08	$\circlearrowright$	\circlearrowright	1 "0B	$\leftrightharpoons$	\leftrightharpoons
1 "1E	$\Lsh$	\Lsh	1 "1F	$\Rsh$	\Rsh	1 "28	$\multimap$	\multimap
1 "20	$\rightsquigarrow$	\rightsquigarrow	1 "21	$\leftrightsquigarrow$	\leftrightsquigarrow			

"Negated" arrows.

2 "38	$\nleftarrow$	\nleftarrow	2 "39	$\nrightarrow$	\nrightarrow	2 "3D	$\nleftrightarrow$	\nleftrightarrow
2 "3A	$\nLeftarrow$	\nLeftarrow	2 "3B	$\nRightarrow$	\nRightarrow	2 "3C	$\nLeftrightarrow$	\nLeftrightarrow

Delimiters.

1 "70	$\ulcorner$	\ulcorner	1 "71	$\urcorner$	\urcorner
1 "78	$\llcorner$	\llcorner	1 "79	$\lrcorner$	\lrcorner

Non-math symbols.

1 "58	✓	\checkmark	1 "72	®	\circledR
1 "7A	✠	\maltese	1 "55	¥	\yen

Alternate names.

1 "6E	$\lll$	\llless	1 "6F	$\ggg$	\gggtr	1 "2B	$\doteqdot$	\Doteq
1 "65	$\Cap$	\doublecap	1 "64	$\Cup$	\doublecup	1 "16	$\upharpoonright$	\restriction

Output Devices

Output Devices and Computers

Table I: Proof-Quality Devices

	Amdahl (MTS)	Apollo	CDC Cyber	DEC 10	DEC 20	DG MV 8000	Ether-net	HP 1000	HP 3000	HP 9000	IBM (MVS)	IBM (VM)	IBM PC	PERQ	Prime	Siemens (BS2000)	Sun	TI PC	VAX (Unix)	VAX (VMS)
C Itoh																				LSU
Canon											GMD			GMD		GMD			Canon	
DEC LN01																			UWash	LSU
DEC Ltr Ptr 100					OSU^d															
DEC VT125																				INFN
Diablo									TeXeT						OSU^p					
Epson								JDJW					A-W					TAMU		
Facit 4542																				INFN
Fla Data					MR												Textset			
GE 3000		COS																		
HP 2680							Stnfd		TeXeT											
HP 2688A										HP; CaTch										
IBM 3800; 4250; Sherpa												SLAC								
Imagen	UBC	OCLC		Stnfd; Vndblt	SRI; Clmbia		Imagen					SLAC	OCLC‡				Sun		UCIrv	K&S†
NDK 7700												IAM								
Printronix																		TAMU		
QMS Lasrgrfx	Textset	ScnLsr; Textset		Textset	Textset	TAMU					Textset	Textset		GMD	TAMU		Textset		Textset; UWash	TAMU
Qume									TeXeT											
screen prevue		Yale; Textset									GMD			GMD		GMD	Textset			Adld
Symbolics					UWash														UWash	Calma
Talaris				Talrs†	Talrs†						Talrs†								Talrs†	Talrs†
Tektronix 4014											UMilan									Adld; INFN
TI 855																		TAMU		
Varian					AMS															SciAp
Versatec			UKöln	GATch; Vndblt	UWash						UMilan	Wzmn			Lvmr				UWash	K&S†
Xerox Dover					CMU		Stnfd												Stnfd	
Xerox 2700			Bochum																	
Xerox 2700II					OSU^d															
Xerox 9700	UMich; Textset	COS		UDel								UDel					Textset			ACC

Notes:

* Still running TeX80
† Graphics supported
‡ Computer used only to support output device, not to run TeX at this installation.

Table II: Typesetters

	Amdahl (MTS)	Amdahl (MVS)	Apollo	CDC Cyber	DEC20	HP3000	HP9000	IBM (MVS)	IBM (VM)	Sun	Univac 1100	VAX (Unix)	VAX (VMS)
Agfa P400									IAM				
Alphatype CRS					AMS								
APS-5/Micro-5	Textset	WashStU	COS; Textset		Textset	Textset	HP	Textset	Textset	Textset		Textset	Intergraph†; Textset
Compugraphic 8400						USheffield							K&S†
Compugraphic 8600		WashStU		RECAU*							UWis*		
CRTronic													Eire
Harris 7500												SARA	
Linotron 202					Adapt								

Most of the interfaces listed in these charts are not on the standard distribution tapes. Some are considered proprietary. Information regarding these interfaces should be obtained directly from the sites listed.

Output device data is being maintained by Rilla Thedford. Anyone desiring more information or relaying new information can send it to her at the address given on the reverse of the title page or via the Arpanet:

`Rilla_Thedford%UMich-MTS@MIT`

The codes used in the charts are interpreted below, with a person's name given for a site when that information could be obtained and verified. If a contact's name appears in the current TUG membership list, no further information beyond a phone number is given. If the contact is not a current TUG member, the most recent address, and its source, are shown.

ACC (Advanced Computer Communications): Diane Cast, 720 Santa Barbara St., Santa Barbara, CA 93101, 805-963-9431 (DECUS, May '85)
Adapt (Adapt, Inc): Marc Berkowitz, 415-393-9500
Adld (Adelaide University, Australia): Andrew Trevorrow, (08) 228 5984
AMS (American Math Society): Ron Whitney, 401-272-9500
A-W (Addison-Wesley): 617-944-3700, ext. 2677
Bochum (Ruhr Universität Bochum): Norbert Schwarz, 49 234 700-4014
Calma:
CaTch (Cal Tech): Glen Gribble, 818-356-6988
Canon (Tokyo): Masaaki Nagashima, (03)758-2111
Clmbia (Columbia): Frank da Cruz, 212-280-5126
CMU (Carnegie-Mellon University): Howard Gayle, 412-578-3042
COS (COS Information, Montreal): Kevin Small, 514-738-2191
Eire (Bord Fáilte - Irish Tourist Board): James Cumiskey, Dublin 353-1-765871, ext. 1275
GATech (G A Technologies): Phil Andrews, 619-455-4583
GMD (Gesellschaft der Math und Datenfabrik, Bonn, Germany): Dr. Wolfgang Appelt
HP (Hewlett-Packard): Stuart Beatty, 303-226-3800, ext. 2067
IAM (Institut für Angewandte Math, Univ of Bonn, Germany): Bernd Schulze, 0228-733427
Imagen: Dan Curtis, 408-986-9400
INFN (INFN/CNAF, Bologna, Italy): Maria Luisa Luvisetto, 051-307572
Intgrph (Intergraph): Mike Cunningham, 205-772-2000
JDJW (JDJ Wordware): John D. Johnson, 415-965-3245
K&S (Kellerman & Smith): Barry Smith, 503-222-4234
LSU (Louisiana State University): Neal Stoltzfus, 504-388-1570
Lvmr (Lawrence Livermore Lab):
MR (Math Reviews): Dan Latterner, 313-996-5266
OCLC: Tom Hickey, 616-764-6075
OSU (Ohio State University): *DEC 20*: John Gourlay, 614-422-6653; *Prime*: John Crawford, 614-422-1741
RECAU (Aarhus University, Regional Computer Center): Benedict Løfstedt, 06-128355
SARA (Stichting Acad Rechenzentrum Amsterdam): Han Noot, Stichting Math Centrum, Tweede Boerhaavestraat 49, 1091 AL Amsterdam (TUGboat 5#1)
ScnLsr (Scan Laser, England): John Escott
SciAp (Science Applications): L. E. Fields, 619-458-2616
SLAC: Alan Spragens, 415-854-3300, ext. 2849
SRI:
Stnfd (Stanford):
Sun (Sun, Inc):
TAMU (Texas A&M): Bart Childs, 415-965-3245
TeXeT: Lance Carnes, 415-388-8853
Textset (Ann Arbor, Mich.): Bruce Baker, 313-996-3566
Talrs (Talaris): Sonny Burkett, 619-587-0787
UBC (Univ of British Columbia): Afton Cayford, 604-228-3045
UCIrv (Univ of California, Irvine):
UDel (Univ of Delaware): Daniel Grim, 302-451-1990
UKöln (Univ of Köln, Germany): Jochen Roderburg, 0221-/478-5372
UMich (Univ of Michigan): Hal Varian, 313-764-2364
UMilan (Università Degli Studi Milan, Italy): Tektronix: Dario Lucarella, 02/23.62.441 (329); Versatec: Giovanni Canzii, 02/23.52.93
USheffield (Univ of Sheffield, England): Ewart North, (0742)-78555, ext. 4307
UWash (Univ of Washington): Richard Furuta, 206-543-7798
UWis (Univ of Wisconsin): William Kelly, 608-262-9501

Vndblt (Vanderbilt University): H. Denson Burnum, 615-322-2357
WashStU (Washington State University): Dean Guenther, 509-335-0411
Wzmn (Weizmann Institute, Rehovot, Israel): Malka Cymbalista, 08-482443
Yale: Bill Gropp, 203-436-3761

Index to Sample Output from Various Devices

Camera copy for the following items in this issue of TUGboat was prepared on the devices indicated, and can be taken as representative of the output produced by those devices. Some items (noted below) were received as copy larger than 100%; these were reduced photographically using the PMT process. The bulk of this issue, as usual, has been prepared (all with TEX82) on the DEC 2060 and Alphatype CRS at the American Mathematical Society.

- Apple LaserWriter (300 dpi): Textset advertisement, p. 103.
- Canon CX (300 dpi): Metafoundry advertisement, p. 100.
- Epson LQ1500 (180 dpi): Norman Naugle, *An elementary sum*, p. 70; TI/PC running PC TEX.
- QMS Lasergrafix 800 (300 dpi): Norman Naugle and Tomas Rokicki, `\output= . . . \random`, p. 71; TI/PC with PC TEX.
  Gregory Marriott, *A TEX82 implementation on the HP9000 Series 500*, p. 80.
  MicroTEX advertisement (Addison-Wesley), p. 102; IBM PC using MicroTEX.
- QMS Lasergrafix 1200 (300 dpi): Michael J. Ferguson, *Multilingual TEX*, p. 57; VAX 11/780 (VMS).
- Toshiba P351 (180 dpi): PC TEX advertisement, p. 104; IBM PC/XT using PC TEX.
- Versatec (200 dpi): Hans Riesel, *Report on experience with TEX80*, p. 76; reduced from 130%; TEX80, DEC-20.
- Xerox Dover (384 dpi): Amy Hendrickson, *Some diagonal line hacks*, p. 83.

GRAPHICS COMMANDS FOR TEX DISCUSSION IN TEXHAX CONFERENCE

Alan Spragens
Stanford Linear Accelerator Center

The TEXhax network conference carried a number of comments concerning graphics and TEX during a period from about a year ago until about six months ago. Then the discussion petered out, presumably because no consensus was reached. My file of mail items comprising this discussion runs to 53 printed pages.

I wrote the following description of parts of that discussion as a memo to a committee at SLAC investigating how we might best create merged text and graphics on our computer systems. Although we have been creating such documents experimentally for some time in a variety of ways, it has required hacking. We're on the track of methods applicable to a variety of systems and devices, usable by our community of hundreds of physicists who do their own papers. I tried to give a flavor of the discussion and mention some ideas that seemed important to me rather than a summary, thinking that more interested parties should get hold of the actual material that came over the wire. Accordingly, I don't include here mention of important contributions from some of the main participants in the discussion, such as Todd Allen and William LeFebvre, and I hope they'll pardon the omission.

The TEX Project's "party line" on why the TEX language and DVI (TEX "device-independent" output) format lack graphics commands was stated by David Fuchs a year ago: (1) there is no way to provide the capabilities in a device-independent manner, and (2) the world lacks a standard, comprehensive, accepted language for describing computer graphics. Dave mentioned that TEX's designers recognized the need for graphics capabilities in a language specifying the appearance of a printed page, so they included the `\special` command for extending the language for just such a purpose. He exhorted people to consider the long range view, beyond present technologies, rather than dash off a "standard" that would be unsatisfactory in a couple of years, *e.g.*, consider shading, halftones, splines, color, *etc.* Since this "party line" message came over the net a year ago, I called David last week to ask if anything had changed. He said nothing had changed, that they had hoped that "Adobe would take over the world by now," but it hadn't. He also mentioned that a number of sites, including Stanford, had implemented various graphics languages via `\special` that were device-dependent and site-specific. He suggested not limiting ourselves to a particular language, but allow for inclusion of a standard (*e.g.*, put a "tag" such as `slac` into our commands). One suggestion that came up in the TeXhax discussion was to establish a "registry" of graphics commands under the ægis of the TeX Users Group to avoid incompatibilities, at least within the TeX community. Dave said such a registry had not been established to his knowledge.

The TeXhax discussion centered around commands for producing line graphics. There was little discussion of commands for inserting external files or of the more esoteric graphics functions such as color and shading. I think I can fairly say that the following line-graphic operators were generally agreed to be desirable: **line**, **circle**, **ellipse**, **arc**, and **spline**.

Two "systems" of defining these commands emerged, which I'll describe shortly. There was some discussion as to how the graphics commands should be implemented in TeX, that is, how macros to resolve them into `\special`'s would work, which I'm leaving out. (That information would be of interest to macro writers, however—contact me if you want it.) A suggestion that new TeX commands (instead of `\special`) be implemented for graphics was generally rejected.

Paul Grosso described a system wherein the different graphics capabilities of different devices are resolved by "tags" in the graphics language, *e.g.*, `\special{FOO:ABC}\special{BAR:XYZ}` where `FOO` and `BAR` are devices and `ABC` and `XYZ` are differing graphics commands for the two devices. This caused a number of complaints that TeX's device independent philosophy was thereby violated. It was suggested that DVI-to-device translator programs should *ignore* graphics commands unacceptable to target devices, rather than treat them as errors.

There was some discussion of *pen* specification; generally the people concerned with pens seemed to want them defined in terms of shape and size, and they wanted to be able to store pen definitions, perhaps by name or number, and recall them later. A suggestion that METAFONT-style pen definitions be adopted was favorably commented upon by several people. Such a definition might look like `\special{pen 5 ellipse 2pt 1pt}` for an elliptical pen shape whose width is 2 points, whose height is 1 point, and whose definition is stored as `pen 5`. Some people advocated that pen shapes be characters in a font; I didn't quite grasp the significance of this suggestion.

Regarding the set of graphic operators to be accepted, two distinct systems emerged in the discussion. One was called the *delta* system; the other was called the *join* system. I believe the salient characteristic of the *delta* system, as proposed by Pierre MacKay, is that graphics coördinates are given in units relative to the "current" position on the page. In the *join* system, coördinates are absolute. To illustrate, conceivable *delta* system commands are shown in Table 1, more or less as presented by Pierre MacKay working from `TiTROFF` specifications of Brian Kernighan.

`line dh dv`	draw line from current position by `dh dv`
`circle d`	draw circle of diameter `d` with left side here
`ellipse d1 d2`	draw ellipse of diameters `d1 d2`
`arc dh1 dv1 dh2 dv2`	draw arc from current to `dh1+dh2 dv1+dv2`, center at `dh1 dv1` from current position
`spline dh1 dv1 dh2 dv2 ...`	draw B-spline from current position to `dh1 dv1`, then to `dh2 dv2`, then to ...

Table 1. Graphics Commands of the *delta* System

In all of the above examples, `dh dv` is an increment on the current horizontal and vertical position, with down and right positive. The exact syntax of the commands could be somewhat different, depending on how the TeX macros implementing the `\special` were written. The point is to note that the *delta* system depends on relative coördinates. The *join* system, proposed by Howard Trickey and based on work of Ignacio Zabala, depends on absolute coordinates. This sounds less desirable until we see an important additional command that is proposed: `point` n, where n is an integer in the range, say, between 0 and 255. That is, the current cursor position is saved and associated with the numeric name. Thereafter, its absolute coördinates may be used in a set of line graphics commands similar to those of the *delta* system. An additional command of the *join* system would be a `join` command which would draw a line from one previously named point to another, perhaps using a "pen" of specified shape and size which could also have been defined and named. The usefulness of the *join* approach may be seen from the TeX code in Example 1, contributed to the discussion by Howard Trickey.

```
\def\p#1{\special{point #1}}
\def\j#1#2{\special{join 4 #1 #2}}
\tabskip=15pt
\halign{&&\hfil#\hfil\cr
John& Harry& Alexander\cr
\p1&  \p2&   \p3\cr
\noalign{\vskip 30pt}
\p4&  \p5&   \p6\cr
Helen& Janet& Amy\cr}
\j15\j24\j36
```

Example 1. The *join* Graphics System in TEX

This TEX code is a table specification which, were the `point` and `join` commands implemented, would create a table with three columns and draw lines from points centered under John, Harry, and Alexander to points centered over Janet, Helen, and Amy, respectively. The obvious advantage is that the user needn't know where the points `p1`–`p6` will be placed on the paper. It was noted that points in one system could be converted to points in the other with some TEXhackery. There was discussion of such questions as whether points and pens would be remembered across page boundaries. Pierre MacKay pointed out that the *join* system would be preferable if a graphic were closely joined with text, as in the example, but that the *delta* system would probably be better for graphic objects created independently of text.

Some discussion about how to treat the ends of splines escaped me, presumably because of my small experience drawing them. There seemed to be some agreement that options on spline-drawing commands should allow one or both of the end points ends to be "hidden" or "visible."

One entry in TEXhax mentioned the ANSI standard called GKS (Graphical Kernel Standard) which is being proposed. Copies of the committee's working papers can be obtained for $35.00 from

X3 Secretariat/CBEMA
311 First St. NW, Suite 500
Washington, D.C. 20001

Phil Andrews described the part of the standard called a "Virtual Device Metafile" (VDM) which establishes a set of primitives device drivers would be expected to handle. They include: **polyline, polygon, circle, arc, arc close** (pie or chord), and **cell array** (an array of colored points). More complicated figures such as splines are supposed to be drawn before the VDM is written.

Leslie Lamport's LATEX system was mentioned for its graphics capabilities. The interesting thing about LATEX's graphics is that they are generated by TEX itself, rather than at the DVI-device level; they require nothing more of the device and DVI translator program than is already present if TEX is working—the capability to place characters at coordinates as specified in the DVI. The LATEX graphics work by typesetting line segments, straight and curved, from special fonts supplied with the macros. Circles and arcs of various sizes are available as are lines at various slopes (TEX draws vertical and horizontal lines itself). Certainly, the LATEX graphics capabilities are quite limited, but they suffice for many purposes and require no enhancements to the TEX system and its friends. They do require a fat version of TEX, and I suspect some printer peculiarities may cause broken lines.

John Aspinall from MIT recommended two books on splines in response to a query about where they came from: *A Practical Guide to Splines* by Carl de Boor (Springer-Verlag, 1978) and "Local Control of Bias and Tension in Beta-Splines" by Brian A. Barsky and John C. Beatty in *ACM Transactions on Graphics*, **2**(2) April 1983. He noted that, traditionally, the spline was a draftsman's tool, a long, flexible piece of wood used to draw a smooth curve through a series of points. The spline was held in place by weights, called "ducks," a term which did not make the transition into mathematical jargon.

A number of rancorous exchanges were made debating the proper position of point `0,0` on the physical device page. This question was settled by David Fuchs who declared that point `0,0` is 1 inch down and 1 inch to the right of the top left corner of the actual output page. He then said, "the Great and Powerful Oz has spoken." That is the TEX standard; it is what DVI expects.

Finally, Charles Karney proposed three new `\special` commands: (1) to specify landscape/-portrait page orientation, (2) to print portions of text in a arbitrary rotation, and (3) to position text correctly with respect to a figure. I believe the meaning and implications of the first two proposed commands are fairly obvious. The third is more complicated and was described as follows (more or less). The idea is to allow TEXed labels (or callouts or nomenclature) on figures that are contained in separate graphics files. (Heard this idea before?) The graphics files don't know anything about TEX, and the DVI doesn't know anything about where the labels should go. Basically, Karney proposes that TEX typeset the labels and the graphic file specify where they should be placed. The two are coordinated by a tag given to each label. This proposal requires that the DVI-reading program read the graphics file and pull out the label specifications, storing their positions and rotations in a table. This information would then be inserted into the label specifications in the DVI so that the DVI-reading program could set the labels in their correct positions, possibly using the capability in Karney's proposal (2). Presumably the graphics-generating program would need to be able to generate the described label material; perhaps a pre- or post-processor could pick out the information and put it somewhere for the DVI-reading program, but the positioning coördinates, it seems, would need to come from the graphic generating program. As another respondent to the proposal asked, "Do most device-independent graphics packages offer a reasonable way of inserting a '`put label n here`' control sequence in their output stream?"

Miscellaneous activity at Texas A&M

Norman W. Naugle and Tomas G. Rokicki

The following three pages illustrate the output from several devices interfaced to TeX at Texas A&M, as well as announcing the availability of a C version of TeX82.

"An Elementary Sum" was output on an Epson LQ1500 (180dpi × 180dpi) using 200dpi fonts (180dpi fonts are not yet available). It was TeXed on a TI/PC running PCTeX (it also works on MicroTeX), and used an output driver and screen preview system written by Tomas Rokicki, which will soon be available on most MS/DOS machines.

We have drivers that work, or can quickly be made to work, on almost any reasonable dot matrix printer or screen display device (currently: TI-855 printer, TI/PC screen, LQ1500, and Printronix P-300). Almost any new driver can be written in a matter of two days. These drivers are written in `WEB` and translated into C, and then, in some cases, modified in machine language. They can be supplied for VAX/VMS, VAX/Unix, Unix in general, MS/DOS, Prime and DG (soon others), as well, of course, as for the QMS-800, 1200, 2400, and soon the Smartwriter. We are also considering the Postscript problem, but no actual work has begun. Some of the drivers suffer from the lack of fonts in the correct size (for example the TI screen), but most have a set of Almost (Computer) Modern fonts.

Our port of TeX to C is aimed at the Unix world, even the large machines, although our main interest is the small systems. The main advantage to C is its portability.

All of these will be available from the Texas A&M TeX Users Group. Write or call for information.

An Elementary Sum*

We show that $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$,
using only elementary trigonometry and algebra.

For the moment fix $n > 0$ and for $1 \le k \le n$ set $\theta_k = \frac{k\pi}{(2n+1)}$. The first step is to use De-Moivre's formula to construct a polynomial whose roots are $\cot^2(\theta_k)$, $k = 1, \ldots, n$. Recall that

$$\begin{aligned}
\sin[(2n+1)\theta] &= \Im(e^{(2n+1)i\theta}) \\
&= \Im\{[\cos(\theta) + i\sin(\theta)]^{2n+1}\} \\
&= \sum_{k=0}^{n} (-1)^k \binom{2n+1}{2k+1} \sin^{2k+1}(\theta)\cos^{2(n-k)}(\theta) \\
&= \left[\sum_{k=0}^{n} (-1)^k \binom{2n+1}{2k+1} \cot^{2(n-k)}(\theta)\right]\left[\sin^{2n+1}(\theta)\right]
\end{aligned}$$

Since $\sin(\frac{k\pi}{2n+1}) \ne 0$ for $k = 1, \ldots, n$, the roots of $p(x) = \sum_{k=0}^{n} \binom{2n+1}{2k+1}(-1)^k x^{n-k}$ are exactly $\cot^2(\theta_k)$.

For any polynomial $p(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_0$, the sum of the roots is equal to $-\frac{a_{n-1}}{a_n}$. Therefore,

$$\sum_{k=1}^{n} \cot^2(\theta_k) = \frac{\binom{2n+1}{3}}{\binom{2n+1}{1}} = \frac{(2n+1)2n(2n-1)}{3 \cdot 2 \cdot (2n+1)} = \frac{n(2n-1)}{3}.$$

$$\sum_{k=1}^{n} \csc^2(\theta_k) = \sum_{k=1}^{n} 1 + \cot^2(\theta_k) = \frac{2(n+1)n}{3}.$$

Also on $[0, \frac{\pi}{2})$, we know that $\tan(x) \ge x \ge \sin(x)$. Thus,

$$\tan(\theta_k) \ge \theta_k \ge \sin(\theta_k)$$

$$\cot^2(\theta_k) \le \frac{1}{\theta_k^2} \le \csc^2(\theta_k)$$

$$\frac{n(2n-1)}{3} \le \sum_{k=1}^{n} \frac{1}{\theta_k^2} \le \frac{2n(n+1)}{3}$$

$$\frac{\pi^2 n(2n-1)}{3(2n+1)^2} \le \sum_{k=1}^{n} \frac{1}{k^2} \le \frac{\pi^2 2n(n+1)}{3(2n+1)^2}$$

An application of the sandwich theorem completes the proof:

$$\lim_{n\to\infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}.$$

---

* A nice little proof of a beautiful, well known theorem. This theorem was proved in 1736 by Leonard Euler (1707–1783).

\output= . . . \random

TeX in C

TeX in C works! Using the same Pascal to C compiler we wrote to convert our drivers to C, we finally managed to convert and compile INITTeX on a VAX-11/750 running Berkeley 4.2 UNIX. Preliminary timing indicates that it runs several times as fast as the Pascal version on this machine! Since this is being written during May, we expect to be distributing TeX in C by the time this TUGboat is published.

C programs tend to be more compact and run faster than equivalent programs in Pascal on UNIX systems. Under Berkeley 4.2 UNIX on a VAX-11/750, our drivers in C tend to run about three times as fast as their Pascal versions. In addition, C shows more consistency from system to system than Pascal, making it easier to port. Finally, some systems simply lack Pascal compilers adequate to compile C without major hackery. We plan to port TeX in C to most common 68000 UNIX systems and any other systems where C works significantly better than Pascal.

Tony the Bit Map

This bit map was included with a \special command on our QMS driver. The positioning was done by TeX; the only hand measuring necessary was of the size of the picture. The picture was captured from an Apple Macintosh using a simple BASIC program and transformed into a hexadecimal bit map.

A Packed Pixel File Format

The wide variety of systems on which TeX is used here at A&M allows us to transfer pixel files and macro packages from system to system. Moving a binary file from one computer to another is one of the great unsolved problems in computer science, especially when the file is long and the telephone is the only method of communication. This difficulty was the impetus behind our development of the packed pixel file format.

The standard pixel file format is admittedly not very compact. On our VMS system, the pixel files occupy more than 12 megabytes. The main reason for the large size of these files is the storage of individual raster rows in 32-bit chunks. For instance, a 10-point font at 200 dots per inch, with an average character width of about 4 points or 11 pixels, wastes an average of 21 bits per 32-bit word.

Even packing the bits tightly together does not yield an efficient use of space. Most black pixels are followed by more black pixels for a reasonable stroke width. Similarly, seldom do single white pixels occur in a character. A great deal of space can often be saved by encoding the number of consecutive black or white pixels that occur, rather than the actual pixels themselves.

After analyzing pixel files at several resolutions, we created a packing scheme which reduces the total size of the 200 and 300 dots per inch pixel files by 72%. This packing scheme is nybble oriented and allows either a tightly packed bit map or an encoding of the character by pixel counts. It is simpler and easier to interpret than the generic font (gf) format. Input or output of the packed format can optionally be in 80-character hexadecimal records for easy transfer by magnetic tape or telephone.

We were surprised by the amount of compaction that was possible with our format. As we were developing drivers for microcomputers at the time and disk storage was at a premium, we wondered if the drivers could use the packed file format directly. We modified our drivers to look for and unpack a packed file if the standard pixel file was not on the disk. On most systems, the drivers ran faster reading the packed file format than they did with the original pixel format because of the reduced number of disk accesses.

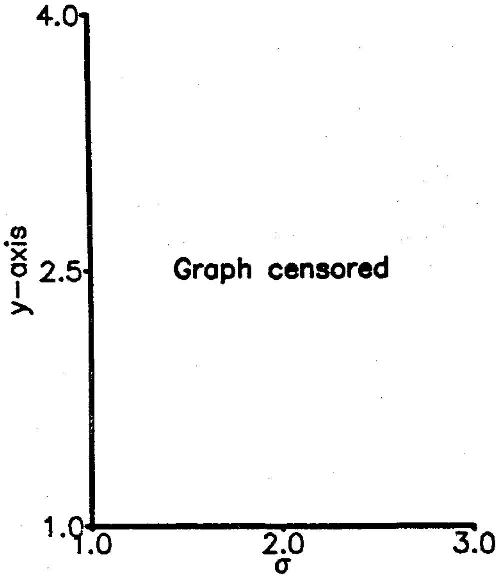
Sources for pxtopk and pktopx, the conversion programs, are available in WEB or C. The WEB files contain full documentation of the format, and have been implemented easily on several machines. The format and associated software are being distributed as freeware.

Compression Statistics (blocks)

	Pixel size	Packed size
200 dpi	13107	3912
300 dpi	20019	5086
Total	33126	8998

Note: These two pages were printed on a QMS-800 Lasergrafix connected to a TI/PC equipped with 768Kbyes of memory, a 10Mbyte hard disk, and PCTeX. Except for the graphics material, the output was previewed, before printing, on the TI screen.

Vector Graph



This 'graph' was also positioned by our QMS driver. The graph was created by a software package currently used in the Texas A&M Physics Department. The fonts are 'Hershey' fonts drawn with vector commands.

Typewriters to TEX: *was it inevitable?*

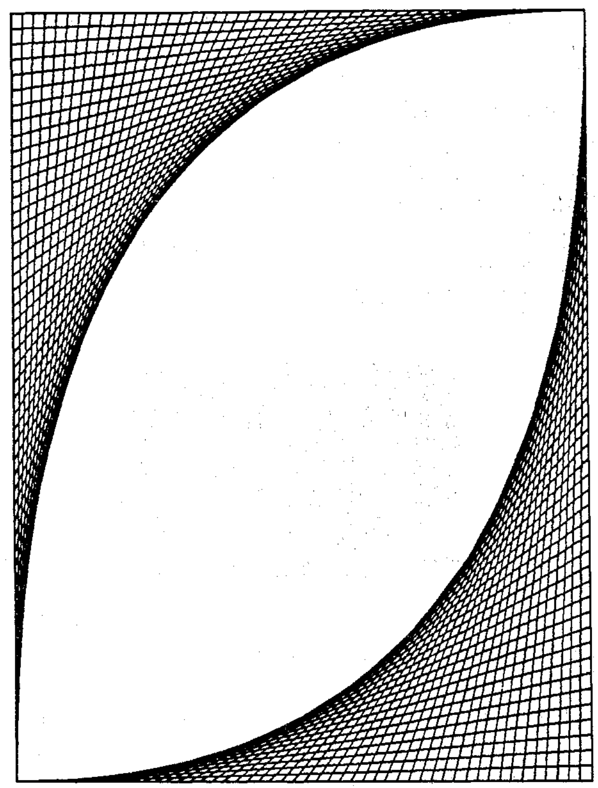
Our conversation with Dr. Matrix's stunning secretary was ended by his booming voice, "Come in, come in! I am just about to try something interesting. You must watch." Rather reluctantly, we followed him into his office.

The log table he used for a desk (the original one used by Noah's frustrated serpents) was strewn with an ant farm, rubber cement, some various papers and drawings, and a few sorry looking balloons. "I just cut a strip from that red balloon and looped it into a Möbius strip. Then, I patched the hole in this blue balloon with the single edge of the Möbius strip." We watched, fascinated, as he blew the balloon into a large sphere. "Now, we take the Möbius travelers ..." He took an ant from the farm on his table and placed it on the patch. The ant walked around the patch, and was suddenly inside the balloon. "So it does hold water!" he mused quietly to himself.

Placing the balloon on the desk, he turned back to us. "I am glad to see that TEX is getting so popular. But of course, it was in the numbers. Is it not obvious?" With a flourish, he typeset 72.27 on his SUN preview station. "Perfectly palindromic. But so is the number of lines on a typewritten page, 66. Certainly symmetry is a critical feature of beautiful documents?"

Noticing the rather skeptical looks on our faces, he asked us what the most popular document magnification was on our system. "But, does not 66 times 1.095 equal exactly 72.27?" Whipping out my pocket VAX, it was only a few seconds of work before I looked at him with stunned disbelief. "So TEX was, of course, the next logical step." Looking at my friend, he said, "Watch out. You have a prime social security number." Then, he turned back to his research. Glancing his SUN screen as I left the room, I caught a glimpse of the number 42. The ant crawled back out of the balloon, and seemed to be watching Dr. Matrix as he worked.

QUIC Graphics



This is an example of the graphics that can be created by direct QUIC commands. The calculations for the vectors were done by TEX using the `\loop` ... `\repeat` macros, and sent to the printer with QUIC literal `\special` commands. There was no external files or processing involved.

Norman W. Naugle and Tomas G. Rokicki
P. O. Box 2736
College Station, Texas 77841
(409) 845-3104

Site Reports

TEX ACTIVITY AT SPRING DECUS

Barbara Beeton

The theme of the Languages and Tools SIG at the Spring DECUS meeting (New Orleans, May 27–31) was "Tools for Software Documentation". TEX and friends were very well represented, and the various presentations were met with considerable enthusiasm. The progam included the following:

- Patti Anklam, DEC, *Tools for software document production*, describing the extensive front-end input and file management system and TEX output interface through which the VAX/VMS 4.0 software documentation was generated, all 10,000 pages of it!
- Barry Ferris, Talaris, *Literate programming — Programs as works of literature*, describing the WEB system.
- David Spencer, Oregon Software, *Automating document production*, describing the techniques used to generate several different manuals from a single set of TEX-encoded files.
- Sam Whidden, AMS, *TEX: Typesetting for almost everyone, an overview*, describing in general terms the WEB/TEX/METAFONT system; and a follow-on Q&A session, assisted by Barbara Beeton.

The presentations by Patti Anklam and David Spencer illustrated the extremes of interfacing documentation writers to a production system.

The DEC approach insulates them entirely from the typesetting language: a file/code management system provides templates to cue the writers for the (generically identified) data and text elements required by the document design, and typesetting instructions (e.g. TEX codes) are inserted only later by a preprocessor; in fact, one early implementation of the system applied DEC Runoff codes, before the TEX macros were working, so that draft output could be obtained for proofreading with no delay. Flexibility in a typographic sense is somewhat limited, but the document design stresses (and enforces) logical consistency, while the TEXperts in the documentation group are very cooperative about adding new structures when they can be justified.

At Oregon Software, the writers actually key in the TEX codes; a question as to whether this was a temptation got the answer, "Well, yes — you sometimes see a writer running down the hall, waving a piece of paper, shouting 'Look at this neat thing i just found!' ".

But both groups agree: TEX is a powerful tool that permits them to generate quality documentation, in both content and appearance, with a great deal of flexibility, in a reasonable, and often limited, amount of time.

Looking toward the future, the editor of the DECUS Proceedings has inquired about making available a TEX header to produce copy in the appropriate format. Such a header will be devised here at AMS, put onto the DECUS distribution tapes (for use by authors who already have Plain TEX available), and introduced formally at a session at the Fall DECUS in Anaheim.

CMS TEX SITE REPORT

Alan Spragens
Stanford Linear Accelerator Center

This site report is rather overdue, considering that there has been an "official" VM/CMS version of TEX available for a year and a half. The current release level is 1.1; work on the 1.3 TEX is in progress—at least I'm collecting useful files and polling CMS sites for ideas and information. In this report I'll describe some of the history and peculiarities of the CMS port and some of the things we'd like to do with it in the future. We can always use help—this is definitely a group effort. I apologize to any contributors to the current state of CMS TEX whose names I have omitted, and for any inaccuracies I may have introduced.

The VM/CMS port of TEX is a combination of efforts of many people. WEB change files for preliminary versions of TEX were brought to the TUG meeting in August 1983 and given to David Fuchs who combined them into a definitive version on the VM system at Stanford Linear Accelerator Center. The original change files were supplied by Roger Chaffee of SLAC (now at Metaphor), Craig Platt of the University of Manitoba, and Peter Sih of IBM Palo Alto Scientific Center. Bernd Schulze of the University of Bonn also contributed his experience from bringing up a preliminary TEX under CMS. All of this work was based on preliminary work adapting TEX to MVS by Eagle Berns and Susan Plass of Stanford University.

Peter Sih generated the system for the distribution tape and contributed device support for several IBM printers. I expect this same software will work for the IBM 3820 printer which doesn't yet have VM support. Many valuable revisions and additions were made by Robert Creasy, including notes on how to use the IBM printer-support utilities. Alan Spragens at SLAC coordinates communication among the CMS sites.

In addition to all the TeX stuff, the current tape has **METAFONT** version −40.0 and a few associated programs. Thomas Denier, of the University of Pennsylvania, made a good start on a CMS change file for **METAFONT** and allowed me to include it with the caveat that it is work in progress. Pete Sih reworked that change file so that it would compile, and that's what went out with TeX 1.1. Since then, several people have made working CMS implementations. Tom Denier continued his work for a while, and sent some driver software for VT-100 terminals running through the Yale ASCII terminal interface. Klaus Thull, at IBM in Heidelberg, brought up version −40.0 and has done work with gray-scale fonts using various output devices including the IBM Sherpa (which is odd because it "writes white," causing various oddities to appear in letter shapes.) His **METAFONT** also makes output for IBM graphics terminals through the GDDM driver. The same version is running at Weizmann Institute, thanks to Sig Handelman. Bernd Schulze brought up **METAFONT** version 0 at the University of Bonn where it is now creating letter-forms and logos. Bernd has written a program with which he converted the Hershey fonts into **METAFONT** input. His TeX is in production, running output to an AGFA P400 LED printer and 3277GA softcopy display, and has produced some large mathematical papers and a full-size book. All these folks have kindly offered to share their work, so I hope we'll have some good **METAFONT** material on the next CMS TeX distribution tape.

The CMS TeX tape includes "load-and-go" modules for TeX and its friends, as well as all the source files and change files. The change files for other systems, as distributed on the generic tape are also there. It has `fmt` files for LaTeX, AMS-TeX, and HPTeX and has a preliminary version of Art Ogawa's Pizzl macros, which were written at SLAC to format physics papers and theses. Peter Sih's device support for the IBM printers, including fonts, is distributed on separate tapes by DP Services, to keep the cost of all the tapes the same.

The CMS TeX has a few peculiarities worth noting here. First is the character translation business. The ASCII-to-EBCDIC conversion done at Stanford may not suit everyone. Of course, the ideal would be to have everything match *The TeXbook*, but EBCDIC terminals usually are missing some of the characters. So people with EBCDIC terminals will probably have to use characters that don't match in any case. Terminals lacking the backslash and brace characters like the 3277 are not fun; it helps to have the APL feature. Most IBM sites also use a lot of ASCII terminals and have some EBCDIC-to-ASCII translation facility. Those local translations will probably disagree with the Stanford ones. Translated back to ASCII, then, TeX still won't look right. Other than changing your ASCII terminal interface, there are at least four approaches to take: use the characters as they come on the tape; do a global edit on every source file on the tape; change some of the `catcodes` in a local file to input along with `PLAIN`; provide execs for setting CMS input and output translate tables with perhaps a profile for each terminal type. Pete Sih wrote an exec called `TEXTERM`, which is on the tape as an example of the fourth approach; at SLAC, we translate the offensive characters on all the source files. I doubt there is much disagreement among the translations aside from some of the non-alphanumerics, but a few of them are important TeX characters. I have received a number of well-considered opinions on this subject from CMS TeX sites, including comments that appeared in the TeXhax conference, which I hope will produce a better translation convention for the next CMS tape.

CMS TeX is invoked in pretty much the same way as any TeX, reading its arguments from the command line, as described in *The TeXbook*. Of course, this differs from normal CMS style in which spaces separate filenames from filetypes and so on. Also, Pascal/VS (at least the 2.1 version) has the unfortunate practice of picking up the tokenized, uppercase version of the CMS parameter list from the command line. That is, the command line is butchered before TeX sees it. For example, if you want to tell TeX to run a file in `\batchmode` without changing the file, so you invoke it with `tex \batchmode \input foo` you are liable to see TeX say `? Undefined control sequence \BATCHMO`. It's easy to avoid the problem with a front-end exec that puts the untokenized string in the console stack, and we've included one by Pete Sih (in EXEC2) on the tape. The only unfortunate consequence of this method is that TeX writes its "**" prompt before it reads the stacked string, and there is a pause during which a nervous user may type something, interrupting TeX.

Input files can be on any accessed disk and can be selected with a specification like `my.file.c` or with the default filetype of `tex` and filemode of * (first match in your search order) if only a filename is given. Files `quit.tex` (containing only an `\end`) and `null.tex` (containing a comment) are supplied to get TeX out of its "`can't find`..." loop. All output files are made on the A-disk. Log files have filetype `texlog`, which we use to avoid conflict with batch log files and such, although *The TeXbook* says `log`.

The `e` response to TeX's ? prompt invokes XEDIT with the current line number in the console stack. This usually puts the current line on the error line, although the user's XEDIT profile could destroy the stack. Pete Sih contributed a small assembler routine that passes a program attention interrupt to TeX, invoking its `errorstop` routine, rather than interrupting CMS. Thus, the CMS TeX is pretty standard in spite of its living in an environment somewhat different from a typical ASCII system.

Although several people have done "core-image" dumps of TeX with the `plain.fmt` information incorporated, saving a ready-made, format-included module, some experimentation at SLAC showed that the CMS I/O can read the `fmt` file at runtime efficiently enough that there is little or no time saved by reading the larger, format-included module.

The `\write15` stream has been co-opted to send its argument to CMS via the Pascal/VS CMS procedure. This feature enables TeX users to run execs, XEDIT or other shared-segment programs, or send messages via CP/CMS messaging facilities during TeX processing.

The tape includes four utilities to interface TeX to the IBM 4250 electro-erosion printer and 3800 model 3 page printer, all of which were written by Pete Sih. The first utility, `DVI4250`, converts a `DVI` file into a format edible by the 4250. You still need the CDPF device driver that comes with the 4250. The second utility, `DVI38PP` does the same for the 3800 model 3. If you have a Sherpa (or APA6670), its current-release software has an interface to accept the 3800 data stream. The third utility, `PXLCVT`, converts the old **METAFONT**'s `PXL` fonts into 4250 format and 3800 format. The fourth utility, `FONTPL`, makes TeX-type property-list (`PL`) files from IBM font files of filetype `FONT4250` and `FONT38PP`.

These utilities are also available on two auxiliary VM/CMS TeX tapes. Each tape comprises complete printer support for TeX on an IBM printer: one for the IBM 4250 electro-erosion printer, the other for the IBM APA6670 laser printer and 3800 model 3 page printer. On these tapes are the four utility programs previously described and the Computer Modern family of fonts, converted to the format accepted by the printer. The auxiliary tape for the 4250 is definitely recommended since 600 dot-per-inch `PXL` files are not part of the standard TeX Project font library. Of course, these considerations will be obviated by the new **METAFONT**.

As shown on the output device chart, a number of VM sites print TeX output on non-IBM printers. I would like to put more `DVI`-translators on the distribution tape. I'm looking forward to seeing Arthur Samuel's Imagen driver, which will use **METAFONT**'s `GF` files directly.

The next thing on the agenda, of course, is TeX 1.3 and the new **METAFONT**. Exactly when a new tape will come out is difficult to say, but I hope we'll be able to get something rolling in the next few months—maybe before or shortly after TUG meeting time. The memory scheme in TeX 1.3 should make for more efficient processing in VM's paged environment. Several people have inquired about making a shared-segment TeX, which would be nice. I believe that the current Pascal/VS is unable to produce re-entrant code; the current TeX module stores into itself. Nonetheless, I still inquire into the possibility at every opportunity; perhaps there is some way I don't know about to get Pascal/VS to allocate free storage at run time. I did get hold of an external procedure to link into a module that will enable Pascal/VS to use the untokenized CMS parameter list, so I hope we'll be able to avoid the `\BATCHMO` problem. Chris Thompson at Cambridge University has done some experiments manipulating the Pascal/VS stack in various ways which may produce a 20% or better improvement in TeX's execution time. Other improvements may be possible, and I hope we can implement a number of them this time around.

Report on Experience with TEX80

Hans Riesel
Royal Institute of Technology, Stockholm

Reason for this Report. In TUGboat 5 (1984) p. 11, Donald E. Knuth asks about some experience with TEX reached at the Royal Institute of Technology in Stockholm. This report has been written in order to give the author's personal answer to his question.

Project Description. TEX80 was introduced on a rather small scale at the Royal Institute of Technology in Stockholm in February 1982. The author of this report had at that time the privilige to attend a one week introductory course. My goal was to do the typesetting of a math book, *Prime Numbers and Computer Methods for Factorization*, Birkhäuser, Boston, 1985, pp. xvi+464. The work was finished in January 1985. The author has so far had practically no experience with later versions of TEX, so it may occasionally happen that the views expressed in this report are outdated by the development of more modern versions of TEX.

The computers used were various DEC-20 computers with a Versatec V80 as output unit. The Versatec has a resolution of 200 dot/inch and the driver magnifies everything by a factor of 1.301, so cmr10 is shown as cmr13. This output size is then photographically reduced before printing.

Useful Features of TEX. First I would like to mention some positive things with TEX. It is really great fun working with TEX—the author has no graphical or typographical education at all and has thus (mostly) experienced each page coming out of the Versatec printer as a very pleasant surprise.

I think that TEX's most valuable feature is the option for authors to write own makros. I have been taking advantage of this all the time, mainly in one of the following situations:

1. For details of graphical design, such as the indentation of the paragraph the reader is just looking at.
2. To circumvent bugs in TEX.
3. To postpone the solution of problems that could not be solved immediately, e.g. if "e.g." should be typeset in italics or in roman letters. In such a situation I just wrote a makro `\eg`, yielding e.g. and then I decided on the precise form later on. Having given the makro `\eg` its final form all e.g.'s appeared as decided in the next draft of the text. This turned out to be a very convenient way of working.
4. To aid people untrained in computer programming to use TEX. At the Institute for Numerical Analysis and Computing Science one of the secretaries took active interest in learning TEX and did a very fine job on typesetting reports. Some of the more difficult types of math formulas, however, did not come out too well, despite her efforts. In such situations the solution was to write a makro with parameters, which could easily be used by her to produce the wanted result.
5. To create complicated expressions such as

$$b_0+\cfrac{a_1}{b_1+\cfrac{a_2}{b_2+\cfrac{a_3}{b_3+\cdots+\cfrac{a_n}{b_n}}}}$$

The beautiful thing with a makro doing this is that TEX automatically expands the boxes involved, if needed. Thus by letting the parameters of the makro be themselves complicated expressions, things like

$$(1+x)^\mu=1+\cfrac{\mu x}{1+\cfrac{\frac{1(1-\mu)}{1\cdot 2}x}{1+\cfrac{\frac{1(1+\mu)}{2\cdot 3}x}{1+\cfrac{\frac{2(2-\mu)}{3\cdot 4}x}{1+\cdots}}}}$$

can easily be typeset.

Also very important is the use of a format file to define the typographical format of your work (where to put page numbers, running heads etc.). If all such information is collected in a separate file, it is handy to introduce changes, if needed. An example of this is the following. I made the mistake to discuss my book with a graphical designer at a much too late stage of the work, but was partly saved by having put all formatting information in one place. Thus I could fairly easily redesign the output format—as a matter of fact, quite substantial re-formatting took me only a couple of hours to achieve.

One very important consequence of using TEX is that you as an author are working much more intimately with your text than you do by the conventional method, i.e. by typing your manuscript or letting your secretary do the typing and then sending it to a publisher. I think that the quality of the text is enormously improved by this way of working, including its intellectual content. In particular the troublesome proof reading, which earlier was done a long time after the work had been submitted for publication, and which was often permitted to take only a short time (i.e. had to be done during night hours) is now done in parallel with the other processing of the text.—I furthermore think that it is much easier to "sell" a work to a publisher when it is nicely presented instead of looking dull and ugly as typed pages with inserted hand-written formulas frequently do.

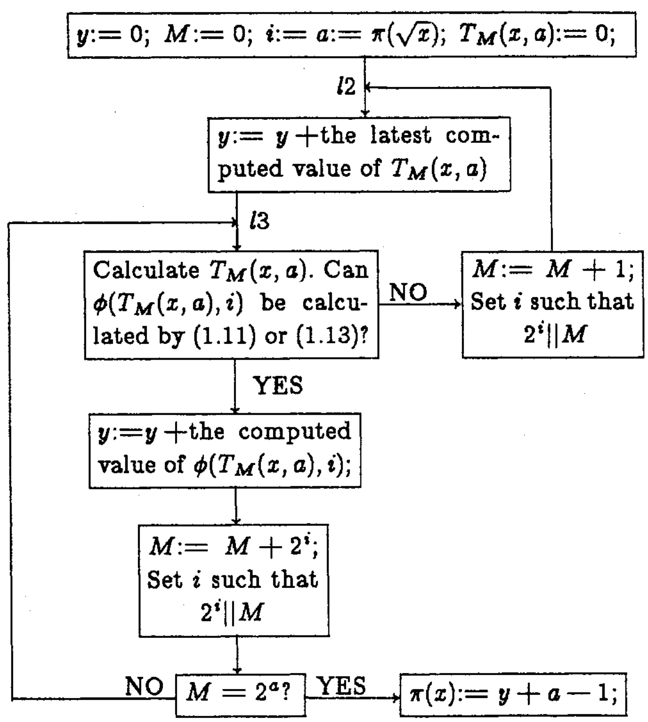
Some Limitations of TEX80. I made some attempts to draw flow-charts with TEX, but I have to advise against this. The best result I managed to achieve after about one week of work was the following:

I was quite discouraged by finding that I had insufficient patience to get rid of the small breaks which show up here and there in the flow-lines! (In the design of tables, however, where I encountered a similar problem, which simply *had* to be solved since my book contains some 100 pages of tables, I managed to find a remedy in the following macro

```
\def\norule{\noalign{\hrule height 0pt}}
```

which I wrote for the purpose.)

Four Professions in One. I here want to take the opportunity of pointing out one consequence of letting the author do the typesetting, a consequence of which I was not fully aware when I began using TEX three years ago. It is the fact that the author has to master, at least to some degree, three more professions, namely the profession of typographer, of graphic designer and of text editor. It does not



suffice to know something about typography.—The graphic designer helps you to create beautiful pages by balancing the types chosen for the running heads, the chapter titles and the subheadings and so forth, with the rest of the text. And, finally, the text editor checks that there are no inconsistencies in the use of grammar, of notation and so on throughout the work, and also that the usage chosen conforms with "normal" usage in the field in question.

Criticism of TEX80. The Fonts. Since I have no education at all in typography or graphic design, I felt quite happy with the result I got from TEX during the first two years I was writing my book. I used the computer as an electronic scratchpad, produced my typeset pages and found everything nice. When the job was almost finished I hit upon the idea to discuss a draft of the text with a graphic designer. He looked at the pages and asked me how I had managed to get them this way. I answered: "Well, I looked at some books which I thought looked nice and tried to imitate them!" A long silence followed. Then the book designer said: "Maybe those books you have been looking at don't represent the best in graphic design!"—But he was very kind and helpful and gave me some valuable advice which I followed and which fixed things up quite a bit.

This graphical designer had never had the opportunity to look closer into computer created fonts before I showed his my pages. His general reflexion was: "With so many bad fonts around, the last thing needed is another bad font!" His main objection to cmr was that the characters look a little bit too crowded, the serifs being too close to some other part of the letter. This creates an impression of a rather heavy type. The heavy looks could, fortunately, be mitigated by simply increasing the baselineskip by one pt, which I did. (So I finally used cmr10 on 13 pt.)

The other major objection from the graphical designer concerned the use of slanted instead of italics. He found this habit rather bad.—Curiously enough, cmi9 looks much better than cmi10, so I actually used italics in the lists of references, which were typeset in ninepoint, but I had to maintain cms10 in the main text.

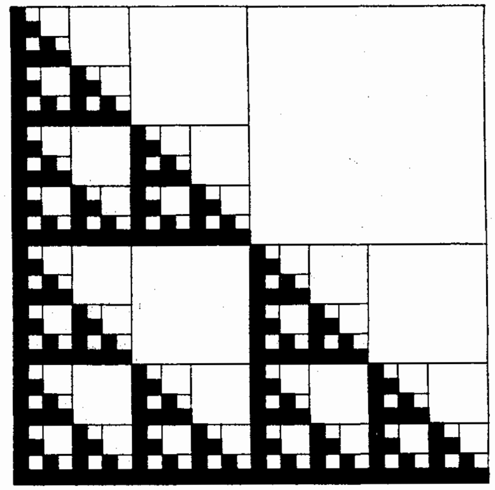
Furthermore, the spectrum of different fonts available turned out to be too limited for my purpose. I would have liked to have true small caps rather than having to use caps in a smaller font.—Touching the subject of fonts I have to inform the reader that there are three national characters in the Swedish alphabet, the å, ä and ö, which have to be incorporated before a font is ready for use in Swedish. Unfortunately the Swedish national characters replace some useful special characters on the terminal's keyboard, i.a. [and], and so *these* in turn have to be replaced by makros. The end result of this whole business is that we here at the Royal Institute have a slightly more complicated version of TEX than at Stanford and, worst of all, our TEX-files are incompatible with TEX-files in other countries. We thus have to send paper copies or films of camera ready texts rather than magnetic tapes when a text is to be reproduced abroad.

The Complexity of TEX. I think TEX, as being created by a computer scientist, is about as nice as it could be *for other computer scientists.* But for people ignorant of computer programming, TEX is much too difficult to learn to master. And, unfortunately, there does not seem to be a useful subset of reasonable complexity either—since TEX is built on the idea of boxes and glue, only the most trivial applications, such as writing letters, can be taught without explaining boxes and glue. But, perhaps this is how it has to be, since typesetting is evidently a much more complicated business than just using a typewriter. I have often found the reaction that TEX is too complicated, when trying to persuade colleagues at the Royal Institute to try it. The computer scientists learn it fast, but for other people the initial threshold often seems unsurmontable.

Future Prospects. Since the benefit is so large when an author is doing the typesetting himself or, at least, is supervising his secretary typesetting it, I think every effort should be made in order to achieve this mode of working. To try this idea I have made a couple of attempts to use TEX for papers in journals, one unsuccessful with a local journal (in Swedish) for math teachers, and one with *BIT,* which I found very rewarding. In this case I managed to persuade the editor to accept a camera ready copy of a paper of 18 pages. As a result, I was permitted to deliver the article about six months closer to its publication date than elsewise, a fact of which I took advantage by doing some late updating of the material in the article and in the list of references.—Unfortunately, however, the font used in *BIT* was not available in TEX80, so the typography of my paper contrasts somewhat with the rest of the material in the journal. (The same remark applies to this report, which had to be done in cmr instead of amr.) It also took me a much bigger effort to write the master file for the graphical format of *BIT* than one might expect an average author to be prepared to put in. But if master files with specs and with the appropriate fonts could be made easily available to authors of papers intended for the leading journals, I am quite sure that this way of working would be used more and thus would help to cut down production costs of math journals.

To link TEX with computer graphics is another important possibility. For my book I needed a picture of a certain fractal structure, which is shown here. This was done for me with the aid of TEX82 with a very short program written by a colleague of mine, Lennart Börjeson.

Finally, the reduction by TEX and similar systems of the very high costs involved in the typesetting of mathematical formulas, having so far prevented the publication of some highly specialized works with only a small market, will undoubtedly expand the domain of works, deemed economical to publish. This will clearly have a positive influence on the market of math books.—In this line of thought lies also the fact that works rejected by journals can now be published by their authors in a form that is at least typographically attractive.



Fractal structure, programmed by Lennart Börjeson using TEX82.

A TEX82 IMPLEMENTATION ON THE HP9000 SERIES 500.

Gregory Marriott
Thermodynamics Research Center
Texas A&M University

Here at TRC, one of our functions is to publish tables of thermodynamic properties of compounds. Until recently, the introductions to these publications were done using an IBM composer. Mathematical symbols were made using rub-on transfers. This made modification of the documents a long and drawn out process. When we first heard of TEX, it sounded like what we were looking for, but no version was available which could run on our computer.

Our computer is an HP9000 series 500. This is a 32-bit, 32-user system with HP-UX version 4.02. HP-UX is HP's version of UNIX, based on System III. We have 4 Mb of memory and 140 Mb of mass storage. Since we knew of nobody with an HP9000 implementation, I decided that I should try an implementation.

At the start, I had almost no experience with UNIX, none with Pascal, and I thought `WEB` was something spiders caught bugs in. (As it turned out, `WEB` caught several of **my** bugs!) I did, however, have the good fortune of having at my disposal a copy of *The WEB System of Structured Documentation*. This contained a description of `WEB`, but most important it included a step by step procedure for implementing TEX82. In a relatively short period of time, I learned enough `WEB`, Pascal and UNIX to bring up **tangle**. Shortly thereafter **weave** came to life. It all seemed easy. I hadn't run into a major problem. TEX looked like it was just a couple of days away.

Somewhere I found a copy of the TEX listing and started in on the index, in the section marked **system dependencies**. I spent the next couple of days making notes and a change file for **tangle**. When I ran **tangle** on **tex.web**, my hopes began to fade away. I discovered that I hadn't learned `WEB` as well as I thought. Many long editing (and hair pulling) sessions later, **tex.web** made it through **tangle** with no errors.

Eventually I worked up enough nerve to try to compile TEX. **TRY** is the key word in that sentence. I soon discovered that TEX was too large to compile with the Pascal compiler on our system. Both the program and its global variables overflowed their respective limits. All I could think of to do was call Hewlett Packard and ask them if there was some compiler option I was missing. I described my problem to their service representative, who said someone would call me back with an answer. Two days later, I got an answer. They said that the compiler was designed with the theory that if a program grew that large, it would have been developed in sections and compiled seperately. The only thing I could do was to split TEX into pieces and try it that way. The idea didn't exactly thrill me, but I had no other choice.

Splitting TEX into little pieces is not something one should take lightly. It never occured to me that one single program could have over 300 different procedures and functions! I didn't know how small I should make the pieces; should I cut it in half, thirds, quarters? I decided three pieces ought to be enough. It took me nearly two weeks to get all the external definitions straightened out and in the right place. (NOTE: now if I want to split TEX, I let scripts do all the work!) What a mess! Finally I was ready to try again. This time all the pieces compiled and linked together nicely. It even started to run.... started to. I soon found out that it wouldn't read font information, or correctly output **DVI** files. It turned out that this compiler doesn't support packed files of any kind. This meant that special I/O routines would be necessary in order for TEX to communicate properly. After a crash course in integer arithmetic and file storage techniques, I had some I/O methods devised which seemed to work. Now I just had to put them in the change file, re-**tangle tex.web**, and then split TEX up again. I can now say with confidence that TEX works on an HP9000.

It was only a day after I got TEX running when I discovered that a whole bunch of other programs, collectively called TEXware, were also on our tape. At first I just left them alone, but then I was told that most of them were necessary in order to test TEX well enough for it to be called TEX! I soon realized that if I had implemented these programs first, TEX would have been much, much easier. Now, all of these programs are running perfectly. My experience with TEX made them quite easy to change.

I would like to express my gratitude to Dr. Norman Naugle, of the Mathematics Department of Texas A&M, and to Tomas G. Rokicki for their assistance on this project. I would also like to mention that their DVI–QMS driver program is included on our tape.

Anyone who is interested in TEX82 for the HP9000 series 500, please contact:

Gregory Marriott or Ken Marsh
Thermodynamics Research Center
Texas A&M University
College Station, TX 77843
(409)845-4940

UNIX TeX SITE REPORT

Richard Furuta

The previous report in the March TUGboat covered through the beginning of February, 1985. This report is being written in early May, 1985.

The most significant developments this time are that the first Unix versions of METAFONT were placed on the Unix TeX distribution tape in late March (Version 0.77). The port is being carried out by Paul Richards of the University of Illinois. As noted in last issue's site report, the new METAFONT tickled previously undiscovered bugs in the Versatec and Imagen 8/300 device drivers. A fix from Chris Torek of the University of Maryland was installed in late April. At the same time, the device driver directory hierarchy was rearranged to reflect the commonality of the library files used by these two device drivers. Chris also contributed a new filter program, named **dviselect**, that extracts pages from a DVI file to allow them to be printed separately (many of the **dvitype**-based device drivers already perform this function).

Although versions of METAFONT are now available for Unix, corresponding font descriptions have not yet been released. Consequently, we continue to provide fonts in the PXL format.

It became increasingly clear to me during this period of time that 300 dot per inch laser printers are becoming rather common. In mid-March, we decided to remove the PXL fonts for the less common 480 dot per inch printers and to use the reclaimed space on the tape to provide a complete set of PXL fonts for the 300 dot per inch printers.

TeX software continues to develop. Version 1.3c of TeX went onto the Unix TeX tape in mid-March. LaTeX also was updated at that time.

In late March, Van Jacobson of Lawrence Berkeley Laboratory contributed **tgrind**, a program that prepares program sources for processing by TeX (a very similar relationship exists between **vgrind** and **troff** on Unix). Subsequently, Van Jacobson and Chris Torek sped the program up, and the faster version went onto the tape in late April. Scott Simpson of TRW contributed a QMS 1200 driver that was placed on the tape in late March. An update in late April added support for the QMS 800.

In closing, I'd like to note that we've received a number of interesting brochures recently from computer companies announcing new 4.2 bsd products. If you port TeX to one of these computers, we're interested in hearing from you. We are also looking for laser printer device drivers for the new printers on the market; particularly the DEC LN03 and the Apple LaserWriter.

VAX/VMS SITE REPORT

Barry Smith

Two items of interest here at Kellerman and Smith:

- The new 1.3 release of TeX is available through us, nicely packaged for VAX/VMS V4.0 or later. This includes changes to LaTeX through April, and also the new BIBTEX bibliographic reference manager.
- Macintosh TeX runs, and it runs TRIP.TEX in 6 minutes, 30 seconds real time on the Macintosh XL; one (1) glue setting differs from the reference in the eighth place. We're pleased with the time (a VAX 11/750 runs TRIP in 2 minutes, 40 seconds), especially because we've not put any effort into performance improvement, so there should be room for some reduction in that time. First, though, we'd like to see the output — perhaps by the time you read this we'll have a display driver ...

"small" TeX

Lance Carnes
163 Linden Lane
Mill Valley, CA 94941

"small" TeX implementations

Computer	Processor	Contact	Organization, Address
Hewlett-Packard 1000	16-bit	John Johnson	JDJ Wordware, Box 354, Cupertino, CA 95015; 415-965-3245
Hewlett-Packard 3000	16-bit	Lance Carnes	TeXeT, 163 Linden Lane, Mill Valley, CA 94941; 415-388-8853
DEC PDP-11/44 Plexus, Onyx IBM PC	16-bit[1] Z8000[1] 8086/88[1]	Dick Gauthier	TYX, 11250 Roger Bacon Dr., Suite 16, Reston, VA 22090; 703-471-0233
Apollo	MC68000	Thom Hickey	OCLC, Box 7777, Dublin, OH 43017; 614-764-6075
		Bill Gropp	Dept. of Computer Science, Yale University, Box 2158, Yale Station, New Haven, CT 06520; 203-436-3761
		Pierre Clouthier	COS Information, 5647, rue Ferrier, Montreal, H4C 1V4, P.Q.; 514-738-2191
Hewlett-Packard 9836	MC68000	Jim Crumly	Hewlett-Packard, Box 15, Boise, ID 83707; 208-376-6000 x2869
Sun Microsystems	MC68000	Jim Sterken	Textset, Box 7993, Ann Arbor, MI 48107; 313-996-3566
		Rick Furuta	University of Washington, Computer Science, FR-35, Seattle, WA 98195; 206-543-7798
Cyb	MC68000[2]	Norman Naugle	Mathematics Dept., Texas A&M University, College Station, TX 77843; 409-845-3104
Apple Macintosh, Lisa	MC68000	Barry Smith David Kellerman	Kellerman & Smith, 2342 SE 45th Ave., Portland, OR 97215; 503-232-4799
Masscomp	MC68000	Bart Childs	Dept. of Computer Science, Texas A&M University, College Station, TX 77843; 409-845-5470
Synapse	MC68000	Dick Wallenstein	Comcon, 5 Underwood Ct., Delran, NJ 08075; 609-764-1720
PERQ/IQL		Jaap van 't Ooster	Océ, St. Urbanusweg 43, 5900 MA Venlo, Holland
IBM PC, XT, AT	8088, 80286	Lance Carnes	Personal TeX, 20 Sunnyside, Suite H, Mill Valley, CA 94941; 415-388-8853
IBM XT, AT	8088, 80286	David Fuchs	Dept. of Computer Science, Stanford University, Stanford, CA 94305
IBM XT, AT	8088, 80286[3]	Ronny Bar-Gadda	446 College Ave., Palo Alto, CA 94306; 415-326-1275
HP 9000 Series 500	32-bit	Greg Marriott Ken Marsh	Thermodynamics Research Ctr., Texas A&M University, College Station, TX 77843; 409-845-4940

[1] not TeX82 [2] currently unimplementable [3] in progress or recently completed

Macros

Some Diagonal Line Hacks

Amy Hendrickson
TEXnology, Inc.

Occasionally the need arises for a slanted or diagonal line. How to do it in TEX? This macro makes it possible and at the same time demonstrates some of the new features of TEX82: loops, counters and conditionals.

The basic mechanism involved is to use a loop to move a dot over and down a specified amount a number of times until the condition that determines the width is met, at which time the loop is discontinued. The distance the dot has moved each time is sufficiently small that the single dot is no longer distinguishable and a slanted line is produced.

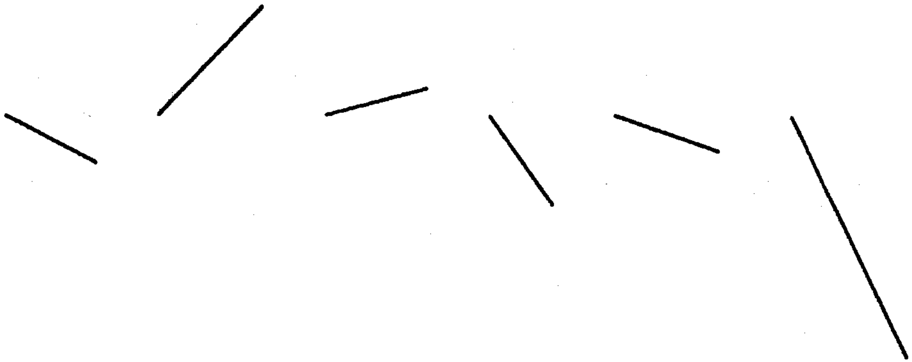
The user determines the slant of the line and the final width of the horizontal space the line will occupy in the `\newline` macro by specifying the width of the horizontal move each time, (#1), the length of the vertical move each time, (#2) and the final width desired, in points (#3).

Scaled points (sp) are used so that the dot can be moved a fraction of a point (remembering that counters will operate only with whole numbers). Here is the annotated code for `\newline`:

```
% 65536sp=1pt 72.27pt=1in
%Setting up counters
\newcount\originalvmove
\newcount\hwidth\newcount\numtimes
\newcount\hmove\newcount\vmove
%%%\vmove and \hmove are multiplied by 1000 to increase the
%%% distance traveled each time the loop is iterated
\def\newline#1#2#3{\numtimes=0 \vmove=1000\hmove=1000
\multiply\hmove #1 \multiply\vmove #2 \originalvmove=\the\vmove
%%%Following is the calculation to arrive at the number of
%%% times \hmove must be invoked to arrive at the desired
%%% width (\#3), given to the counter'\hwidth'
\hwidth=#3
\multiply\hwidth by65536%%%(number of sp in pt)
\divide\hwidth by\the\hmove
%%%Now the loop starts, and continues until the counter
%%%'\numtimes' is as large as the counter '\hwidth'
\loop\ifnum\numtimes<\hwidth
\hskip\the\hmove sp\lower\the\vmove sp\hbox
to 0pt{\hss.\hss}\advance\numtimes by 1\advance\vmove by\originalvmove
\repeat}
```

The values of the second argument can be negative as well as positive, yielding a line moving up from its origin instead of down. An example:

```
\newline{40}{20}{35}\qquad\newline{20}{-20}{40}\qquad
\newline{60}{-15}{40}\qquad\newline{30}{40}{25}\qquad\newline{30}{10}{40}
\qquad\newline{30}{60}{45}
```



You can tailor the line to fit a specified space, for instance in a table, by measuring the width you want and playing with the slant of the line until it fits the space correctly. A -\vskip is used after \newline to make it appear within the table.

First Symbol / Second Symbol		Nonfloating Insertion Symbols									Floating Insertion Symbols						Other Symbols					
		B	O	/	,	.	{+} { }	{+} { } {-}	{CR} { } {-}	CS {DB}	{Z} { }	{Z} { } {*}	{+} { } {*}	{+} { } {-}	CS {-}	A CS	9	X	S	V	P	P
Non-floating Insertion Symbols	B	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y		Y		Y	
	O	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y		Y		Y	
	/	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y		Y		Y	
	,	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y		Y		Y	
	.	Y	Y	Y	Y		Y			Y	Y		Y		Y							
	{+-}																					
	{+-}	Y	Y	Y	Y	Y			Y	Y	Y			Y	Y	Y			Y	Y	Y	
	cs					Y																

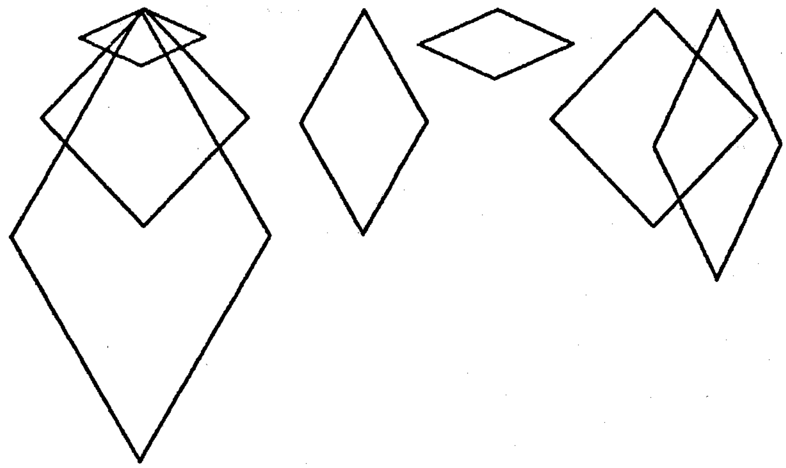
Making Diamond Shapes

A variation on the previous macro will make a diamond shape. This is useful for people making programming flow charts. Once again, the values given to it are for horizontal move (#1), vertical move, (#2), and the width of the whole diamond, in points (#3). If you copy and use this macro, one caution—at some point you can use up all the memory TEX has, so if you need a *really* big diamond shape, or need to use it many times in one file, you may need to increase your system memory.

Here is the code for \diamondline, made by combining four loops, one for each side of the diamond, and with one parameter of the specifications for the height or depth changed for each side, as noted in the comments.

```
\newcount\hwidth
% 65536sp=1pt
\newcount\hmove\newcount\vmove \newcount\originalvmove
\newcount\hwide
\newcount\numtimes
\def\diamondline#1#2#3{\numtimes=0 \vmove=1000\hmove=1000
\hwidth=#3
\multiply\hwidth by65536
\divide\hwidth by2
\multiply\hmove #1 \multiply\vmove #2
\divide\hwidth by\the\hmove
\originalvmove=\the\vmove
\loop\ifnum\numtimes<\hwidth
\hskip\the\hmove sp\lower\the\vmove sp\hbox to
0pt{\hss.\hss}\advance\numtimes by 1\advance\vmove
by\originalvmove \repeat
\numtimes=1
\loop\ifnum\numtimes<\hwidth
\hskip-\the\hmove sp\lower\the\vmove    %%%Notice \hskip-\the\hmove
sp\hbox to 0pt{\hss.\hss}\advance\numtimes %%%to move to the left
by 1\advance\vmove by\originalvmove
\repeat
\numtimes=0
\loop\ifnum\numtimes<\hwidth
\hskip-\the\hmove sp\lower\the\vmove sp\hbox to  %%%\hskip-\the\hmove
0pt{\hss.\hss}\advance\numtimes by           %%%to move to the left.
1\advance\vmove by -\originalvmove %%% This time we also need
\repeat               %%% -\originalmove to make the line move upwards
\numtimes=0
\loop\ifnum\numtimes<\hwidth       %%% This time \hskip is positive but
\hskip\the\hmove sp\lower\the\vmove sp\hbox to %%% we use -\originalmove
0pt{\hss.\hss}\advance\numtimes by  %%% to move up, which finishes
1\advance\vmove by-\originalvmove   %%% the diamond.
\repeat}
```

And here are some sample diamonds...



```
\diamondline{30}{50}{100}\diamondline{70}{30}{50}\diamondline{30}{30}{80}
\hskip80pt\diamondline{30}{50}{50}\hskip50pt\diamondline{70}{30}{60}
\hskip60pt\diamondline{30}{30}{80}\qquad\diamondline{30}{60}{50}
```

To which can be added horizontal and vertical arrows if they are to be used for a flowchart.

* * * * * * * * *

A (Possibly) Totally Useless Macro

Inspired by D. Knuth's prime number generator, I've put together a smaller, and humbler, number generator—this time for numbers in the Fibonacci series. Each new number in the Fibonacci series is made by combining the previous two numbers. It is known as a way of describing certain growth patterns, such as spirals in the cross-section of some seashells and as a measure of the 'golden section', a relationship thought to be especially harmonious by some artists and musicians. This macro will print out the Fibonacci series to as many instances as you call for in the argument (up to 47 that is, then the numbers get too large for TeX to manage).

Here is an example–`\fibonacci{26}`: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, and 121393.

```
\newcount\numbertimes \newcount\numone \newcount\numtwo
\newcount\savenumone \def\fibonacci#1{1, \numbertimes=2  \numone=0
\numtwo=1\loop \advance\numone by \numtwo \the\numone,
\savenumone=\the\numone \numone=\numtwo \numtwo=\savenumone
\advance\numbertimes by1\ifnum\numbertimes<#1\repeat
\ifnum\numbertimes=#1 \advance\numone by \numtwo\fi and \the\numone.}
```

MULTI-COLUMN LAYOUT IN TEX80

August Mohr

I want to describe how I get TEX to do some of the trickier kinds of 3-column layout as encountered in newsletters and magazines. It's not easy, but it works.

I should start out with the caveat that I program in TEX80 (TEX in C by Unidot/TyX) under UNIX, and not in TEX82. What I describe here would be handled differently in TEX82, I'm sure.

The Problem

The figure below is a sketch of a hypothetical page from the middle of a newsletter. Article "B" is to fit on the page in its entirety, while Article "A" continues onto the next page. In a page like this, pieces *a2* and *a3* must balance, while making allowance for *TITLE A* and the continuation line, and their sizes must also depend on the length of Article B. Pieces *b1* and *b2* must also balance.

```
|_____________header_____________|
 ______________________   _________
|                      | |         |
|       TITLE A        | |         |
|______________________| |         |
 _________   _________   |  (a3)   |
|         | |         |  |         |
|  text   | |  text   |  |         |
|         | |         |  |         |
|  (a1)   | |  (a2)   |  |_________|
|         | |_________|  continued
|         |
|         | =====================
|         |  ____________________
|         | |                    |
|         | |      TITLE B       |
|         | |____________________|
|         |  _________   _________
|         | |         | |         |
|         | |  text   | |  text   |
|         | |         | |         |
|         | |  (b1)   | |  (b2)   |
|         | |         | |         |
|         | |         | |         |
|_________| |_________| |_________|

|_____________footer_____________|
```

Example: "Page 11"

A Solution

In my solution, the `\output` routine keeps changing the `\vsize`, and saving the pages in successive `\box`-es. When the proper number of boxes have been filled, it puts out the completely assembled page. There are two primary pieces in this description: the "run" file, which identifies the text files, page number, and so on; and the "layout" file, which describes the structure into which the text will be fit.

The UNIX file system has been very helpful in keeping track of all the different files I use, and in allowing me to make my TEX programs "modular". I primarily use three directories within my working directory: **tex**, where I keep all the TEX macros and page structures; **text.x**, where I keep the text in its TEX form; and **run.pg**, where I keep the files which TEX initially reads. A subdirectory of **tex** is **tex/layout.pg** where the page layout descriptions are kept.

The Run File

To layout a page I create a "run" file, a numbered file in the directory **run.pg** such as **run.pg/11**, the run file for page 11. More than one page may be included in the run file, and I name the file accordingly (e.g. **run.pg/13.14**). I find it useful to keep the run files small, i.e. one or two pages, to speed processing time during the testing and debugging cycle. It is this run file that TEX initially sees (e.g. from a command line such as **tex run.pg/11**).

File: **run.pg/11**

```
\input tex/newsformat.x
\input tex/layout.pg/11
\beginsheet{11}
\dummynext

\tenpoint
\article{B.filename}
\recyclebox2
\article{A.filename}
\end
```

Within the run file, several files are `\input`. The main "driver" is **newsformat.x**, which is in the directory **tex**. This file `\input`-s all the other definition files, defines the `\output` routine, and defines all the macros used in the run files. The sub-directory **layout.pg** has a layout description for each page of the newsletter. If the run file covered more than one page, the appropriate layout files would also be `\input`.

The run file next identifies the page number and the file name of each article, and then `\end`-s. Notice that Article B is read before Article A. That is because the sizes of the pieces of Article A will depend upon the size of Article B. More on this later.

The following definitions are all excerpted from the file **newsformat.x**. The macro `\article`,

```
\def\article#1{\input text.x/#1 \eject}
```

presumes that its argument is a file in the directory **text.x**. The macro `\recyclebox2` has a simple definition:

```
\def\recyclebox#1{\unbox#1\vfil\eject}
```

but that will not mean much until you know how `\box2` gets filled. I'll get to that when describing the `\output` routine and the layout file.

Before going on, there are two more macros in the run file that should be defined. `\dummynext` sets a simple flag, indicating whether the layout of the next page (in our example, page 12) should be used for material that overflows the current page, or whether a simple "dummy" page should be used instead.

```
\def\dummynext{\gdef\nextisdummy{T}}
\def\nextisdummy{F}
```

If `\dummynext` is not used, the default is automatically defined. This flag makes it possible to debug page 11 before knowing that page 12 works. Otherwise, the run could die due to a fault on page 12, even if page 11 was perfect. I presume that this is peculiar to the C version of TEX80, that does not allow interactive debugging.

The last macro left to define in the run file is `\beginsheet`, which sets the page number and the initial `\vsize`. Like the macros above, it is defined in the file **newsformat.x**. Since the initial `\vsize` must be set according to the page number, some comparison test must be done. TEX's `\if` allows comparisons of single characters, and `\ifdimen` can compare dimensions. Rather than convert a page number into inches, an aesthetically unpleasing mixing of number and quantity, I chose to use `\ifx` to compare the numbers as strings. To do this, some initial definitions are required. I name these macros using Roman numerals to distinguish page numbers. (Getting around TEX's prohibition on numbers in macro names can be a bother. At first I tried letters A-P for pages 1 through 16 and used the simple `\if`, but Roman has proven easier to read, if a little more convoluted to program.) These are also used in the `\output` routine to identify page numbers.

```
\def\I    {1}   \def\II   {2}
\def\III  {3}   \def\IV   {4}
\def\V    {5}   \def\VI   {6}
\def\VII  {7}   \def\VIII {8}
\def\IX   {9}   \def\X    {10}
\def\XI  {11}   \def\XII  {12}
\def\XIII{13}   \def\XIV  {14}
\def\XV  {15}   \def\XVI  {16}
```

Only 16 numbers are defined because I only needed 16 pages in my publication. More may be added as needed, limited only by the space available for `\def`-s. (If more pages are to be allowed for, changes must be made in this list, in `\beginsheet`, and in the `\output` routine. More layout files must also be created.) With these definitions established, the definition of `\beginsheet` should be clearer.

```
\def\beginsheet #1{\xdef\sheet{#1}
          \setcount0 \sheet
          \ifx\sheet\I    {\setIvone}
    \else{\ifx\sheet\II   {\setIIvone}
    \else{\ifx\sheet\III  {\setIIIvone}
    \else{\ifx\sheet\IV   {\setIVvone}
    \else{\ifx\sheet\V    {\setVvone}
    \else{\ifx\sheet\VI   {\setVIvone}
    \else{\ifx\sheet\VII  {\setVIIvone}
    \else{\ifx\sheet\VIII{\setVIIIvone}
    \else{\ifx\sheet\IX   {\setIXvone}
    \else{\ifx\sheet\X    {\setXvone}
    \else{\ifx\sheet\XI   {\setXIvone}
    \else{\ifx\sheet\XII  {\setXIIvone}
    \else{\ifx\sheet\XIII{\setXIIIvone}
    \else{\ifx\sheet\XIV  {\setXIVvone}
    \else{\ifx\sheet\XV   {\setXVvone}
    \else{\ifx\sheet\XVI  {\setXVIvone}
    \else{\xdef\sheet{Z}\setZvone
        }}}}}}}}}}}}}}}}
    } %end beginsheet
```

In the example page, I use `\beginsheet{11}`, so the macro `\setXIvone` is called. This is defined in the file **layout.pg/11**. If the run file is set up properly, only the macro corresponding to the current page (`\setNNvone`: `NN` = `I`, `II`,...,`XVI`) will be accessed. The others will remain undefined because the layout files defining them will not have been read. This modularity saves processing time and definition space.

If `\beginsheet` is given an argument that it does not recognize, then `\sheet` is set to "Z" and `\setZvone` is called. `\setZvone` is defined in a layout file that describes a simple default or "dummy" layout. The default layout file (**tex/layout.pg/Z**) should be `\input` within **tex/newsformat.x** so that the default is always available, no matter what the run file contains. The construction of the dummy layout file should be a simple task, once I explain the required contents for layout files.

The Layout File

The layout file for the example page will illustrate all the important points. A layout file has two major parts: the `\vsize` definitions and the "build" macro, which defines the page structure. The `\vsize` definitions themselves have two parts, which I will describe separately.

The first part of the `\vsize` definitions defines a macro, `\setXIvdefs`, which is used in the `\output` routine. When this macro is interpreted, a series of definitions are established. Every macro that is defined within `\setXIvdefs` may used by the `\output` routine. Every macro that is used in a definition, which will have "XI" as part of its name in this case, will be defined later in the layout file.

tex/layout.pg/11 – part 1.

```
\def\setXIvdefs{
    \gdef\setvone   {\setXIvone}
    \gdef\setvtwo   {\setXIvtwo}
    \gdef\setvthree {\setXIvthree}
    \gdef\setvfour  {\setXIvfour}
    \gdef\setvfive  {\setXIvfive}
    \gdef\setvsix   {\setXIvsix}
    \gdef\setvseven {\setXIvseven}
    \gdef\setveight {\setXIveight}
    \gdef\setvnine  {\setXIvnine}
    \gdef\setvzero  {\setXIvzero}
    \gdef\setvnext  {\setXIvnext}
    \gdef\numberboxes{\XInumberboxes}
    }
\def\setXIvnext {\setXIIvone}
```

This first part also defines the `\setXIvnext` macro, which will be used to set the `\vsize` for the first piece of the following page (page 12). This first part of the `\vsize` definitions is repeated in every layout file, except that the "XI" in each macro name is changed to match the appropriate page number, and the `\set...next` is changed to match the following page. Note that `\setXIIvone` need not be defined in order for the page to be processed properly. The flag set by `\dummynext` will cause `\setZvnext` to be used instead. `\setXIIvone` would be used if I were setting pages 12 and 13 together in this run.

The next part of the `\vsize` definitions is not strictly required. I create my layout files by starting with a standard "template" file. I include this as a reminder of exactly what needs to be defined in the file, and in what format. The template includes all the macros that I must define, with "default" definitions for each.

As I redefine each macro to reflect the actual layout of the page, I comment out the initial definition. Thus, all macros in this section that are commented out will be defined further on, and any not commented should not be accessed by TeX. (Note that `\fullcol`, used below as a default, is defined as a dimension, the height of a full column of text, e.g. `\def\fullcol{56pc}`.)

tex/layout.pg/11 – part 2.

```
%\def\setXIvone    {\vsize\fullcol}
%\def\setXIvtwo    {\vsize\fullcol}
%\def\setXIvthree  {\vsize\fullcol}
%\def\setXIvfour   {\vsize\fullcol}
%\def\setXIvfive   {\vsize\fullcol}
%\def\setXIvsix    {\vsize\fullcol}
%\def\setXIvseven  {\vsize\fullcol}
%\def\setXIveight  {\vsize\fullcol}
\def\setXIvnine   {\vsize\fullcol}
\def\setXIvzero   {\vsize\fullcol}
%\def\XInumberboxes{3}
```

The last part of the `\vsize` definitions contains the actual definitions of the macros that were commented out above. These definitions come in two flavors: an explicit statement of a dimension, which may be a macro such as `\fullcol`, or else a description of how to calculate the `\vsize`.

tex/layout.pg/11 – part 3.

```
\def\setXIvone{\vsize 99pc
    \hsize\twowide}   %unboxed in set7
\def\setXIvtwo{\vsize 999pc
    \hsize\normalh}   %unboxed in set3
  %this box is recycled in the run file.
\def\setXIvthree{\save2
    \vbox{\unbox2}\vsize 0.5ht2}
\def\setXIvfour{\vsize 999pc}
                      %unboxed in set7
```

These four definitions are required to process Article B, the first one read in the run file. What I am doing here is determining the `\vsize` and `\hsize` that will be in effect when each piece of the page is read. Since the layout for the page treats the titles as separate pieces of text (not necessary if the title were only one column wide) there needs to be a separate `\vsize` for each title. (The `\hsize` is not necessary in "`three`" and "`four`" since it does not change.)

As the text is read, TeX builds onto the current `\page` until an `\eject` is encountered or the current `\vsize` is exceeded. To ensure that the pieces are broken properly, I give a somewhat large `\vsize` initially, and put an explicit `\eject` into the text between the title and the beginning of the text. By giving a large, arbitrary `\vsize`, I ensure that the `\eject` will control TEX.

In operation, `\beginsheet` (in the run file) accesses `\setXIvone` and sets the initial `\vsize`. When the `\eject` is encountered, the `\output` routine places that `\page` in `\box1`, and uses `\setXIvtwo` to set the new `\vsize`. Each time TEX enters the `\output` routine, the successive pieces of text are placed in successive boxes and a new `\vsize` is set.

Remember that in the run file Article B was read before Article A and that after Article B was read, the macro `\recyclebox2` was used. Look back at the run file and at the definitions of its macros to see what will happen at the end of Article B.

The macro `\article` has an explicit `\eject` at the end, so those 999pc will not be filled. The full text of Article B (less the title that came before the previous `\eject`) will be saved in `\box2`. `\setXIvthree` will be used to set the `\vsize` for the next piece. `\setXIvthree` unboxes `\box2` and saves it again, effectively relaxing any glue that may have been stretched to fill up that 999pc `\vsize`. Now `\box2` is the "actual" size of that text. The new `\vsize` is set to be just half of the height of `\box2`. All this happens at the end of Article B, at the `\eject` supplied by the `\article` macro, under the control of the `\output` routine.

When `\recyclebox` unboxes `\box2` again, it is read in as "text" by TEX. Be careful. It is not really text. It is a list of all the horizontal boxes that were collected in the vertical box that became `\box2`. Fonts, paragraphing, macros, and so on, may not be changed. The horizontal boxes will be read until an `\eject` is encountered, or `\vsize` is exceeded. In this case, it will be the `\vsize` set by `\setXIvthree` that controls the end of the piece. The new piece, saved in `\box3`, will become the left-hand piece of text for Article B.

The text that is being split may contain an odd or an even number of lines, and paragraph spacing may not be an even line-space, so it is unlikely that exactly half of `\box2` will fall exactly at the bottom of a line. For this reason, `\setXIvfour` allows for a larger `\vsize` than that used in `\setXIvthree`, and the `\eject` at the end of the `\recyclebox` definition will control the end of the next piece. That piece, saved in `\box4`, will be the right-hand piece of Article B.

Note the comments at the ends of the lines that have the arbitrary sizes (99pc or 999pc). I find it useful to keep track of where such oversize boxes will get taken down to their proper size.

So far I have dealt only with Article B, which is presumed to fit in its entirety within the planned page format. The only constraint upon the sizes of the pieces of Article B is that the two columns balance. Article A, on the other hand, has more complex dependencies for the sizes of its pieces.

Before describing the `\vsize` definitions for Article A, I need to define a few calculation tools.

Calculation tools

```
\def\add#1to#2into#3{\save0\vbox
      {\vskip#2\vskip#1}\xdef#3{1ht0}}
\def\subtract#1from#2into#3{\save0\vbox
      {\vskip#2\vskip-#1}\xdef#3{1ht0}}
\def\setgreaterof#1#2to#3{\ifdimen #1
           < #2{\xdef #3{#2}}
         \else{\xdef #3{#1}}}
```

In these calculation tools, parameters `#1` and `#2` are expected to be dimensions, or macros defined as dimensions, while `#3` is the name of a macro to be defined to be the result of the calculation.

Now I can go on with the `\vsize` definitions for this page, that cover Article B.

tex/layout.pg/11 – part 4.

```
\def\setXIvfive{\vsize 99pc
    \hsize\twowide}   %unboxed in set6
\def\setXIvsix{\save5\vbox{\unbox5}\!
    \subtract{1ht5}from{\fullcol
                 }into{\XIvsix}\!
    \vsize\XIvsix\hsize\normalh}
\def\setXIvseven{\save1\vbox{\unbox1}\!
    \save4\vbox{\unbox4}\!
    \subtract{1ht1}from{1ht6
             }into{\stilltoomuch}\!
    \setgreaterof{1ht3}{1ht4
             }to{\prevhigh}\!
    \subtract{\prevhigh
             }from{\stilltoomuch
             }into{\XIvseven}\!
    \vsize\XIvseven}
\def\setXIveight
   {\add{1ht5}to{1ht7}into{\prevcol}\!
    \subtract{\contheight}from{\prevcol
                 }into{\XIveight}\!
    \vsize\XIveight}
\def\XInumberboxes{8}
```

Piece five, the title of Article A, is handled just like piece one. It, too, needs an explicit `\eject` between the title and the text. To get the `\vsize` for piece six, I subtract the relaxed height of piece five (`\box5`) from the height of a full column (`\fullcol`). Note the temporary macro `\XIvsix`. Although it is named with Roman numerals, as the "set" macros, it could have been called anything, e.g. \finalv. The same is true for \XIvseven and \XIveight. All three could even have the same name without affecting the calculations. The macros \prevhigh and \prevcol are also temporary, and could be named anything.

To keep track of the calculation for piece seven, remember that \box2 was thrown away by \recyclebox, so Article B consists of \box1, \box3, and \box4. Piece seven then fits into the space left in a column after taking out \box1 (the title of Article B), \box5 (the title of Article A), and the taller of \box3 and \box4 (the text pieces of Article B which are supposed to balance).

Piece eight is then to be as tall as the combination of \box5 and \box7, after taking out the height of the "continuation line" (\contheight).

That is it! When all eight boxes are filled (OK, seven are filled; \box2 is now null) I can build the final page. The \XInumberboxes macro tells the \output routine how many boxes to fill for this page, before going on to the next page. When that number is reached, the \output routine calls the "build" macro.

The "build" macro describes how the finished "sheet" is assembled from its "pieces". Since it is used inside the output routine, between the header and the footer, it must be a vertical list.

tex/layout.pg/11 – part 5.

```
\def\buildXI{\!
  \hbox to \sheetwidth
    {\hbox to \normalh
      {\vbox to \fullcol
        {\box5\copy6}\!
      \hss}\!
    \hfil\vrule height 1ht6\hfil
    \vbox to \fullcol
      {\hbox to \twowide
        {\save8\vbox{\box8\contline}\!
        \box7\!
        \hfil\vrule height 1ht8 \hfil
        \box8}
      \vfil
      \hrule width \twowide height 4pt
      \vfil
      \vbox
        {\box1
        \hbox to \twowide
          {\ifdimen 1ht3 > 1ht4
            {\copy3\!
            \hfil\vrule height 1ht3\hfil
            \vbox to 1ht3{\unbox4}}
          \else
            {\vbox to 1ht4{\unbox3}\!
            \hfil\vrule height 1ht4\hfil
            \box4}\!
          }%end \hbox to \twowide
        }%end \vbox
      }%end  \vbox to \fullcol
    }\!  %end \hbox to \sheetwidth
  } %end \buildXI
```

Most sheets that I build end up having a horizontal outer level, a horizontal stack of columns, so any \buildNN is usually a single \hbox to \sheetwidth. Just about every \buildNN should have a height of \fullcol, the exceptions being pages such as Page 1 that do not get the normal header and footer. Such exceptions will require additional tests in the \output routine to change or leave out the header and footer, which I have not included.

The macro above, \buildXI, has three components within its outer \hbox: an \hbox, a \vrule, and a \vbox.

Notice that the \hbox has a width of \normalh, the normal \hsize of a text column. Inside this \hbox is a \vbox containing \box5 and \box6, and some \hss (horizontal shrink and stretch). The \hss makes up the difference between the width of the outer \hbox and the width of the enclosed \box5, which had an \hsize of \twowide, as wide as two columns of type.

The \vrule that is part of the outer \hbox, and all the other rules on the page, are optional design elements. I like to put them in because they make it much easier to read narrow tight columns of text, and because it is so easy to do compared to hand paste-up.

The last component of the all-encompassing \hbox to \sheetwidth is a \vbox consisting of three parts itself. The top half is an \hbox containing \box7 and \box8, next there is a separating \hrule (thicker than the hairline rule separating the columns), and the bottom half is a \vbox containing \box1, \box3, and \box4 (\box2 got recycled).

In the upper half, \box8 gets the pre-defined continuation line (\contline) added to the end of it. This makes it easy to get the \vrule between \box7 and \box8 to be the right height. The \vbox in the lower half contains \box1 followed by an \hbox containing \box3 and \box4. The \if ... \else ... construction is used to allow for not knowing whether \box3 is taller than \box4. I could have used \setgreaterof here, and then \unbox-ed both into \vbox-es of the same size, but I chose this method for variety.

This is essentially all that has to be done for each page: creating the run file and the layout file. (This includes, of course, debugging the layout and discovering where you run up against the system's limits).

As described above, the text files only need to allow for this structure in one way: if the the title and the text are to be treated as two separate pieces, such as when the `\hsize` changes from two columns wide to one column wide, the article-file must contain an explicit `\eject` between the parts. The new `\hsize` may then be set either within the text file, or else in the layout file, as part of the `\vsize` definitions (viz. `\def\XIvtwo{999pc\hsize\normalh}`).

So far I have only described the run files and the layout files, which are all that, in practice, need to be changed between pages or between editions. Now on to the `\output` routine, which does not change at all.

The Output Routine

The `\output` routine is responsible for changing the `\vsize`, and saving the completed `\page`-s in successive `\box`-es. When the proper number of boxes have been filled, it puts out the completely assembled page. Because the `\output` routine is a single, fairly long block, I will not interrupt my description of its workings to show segments of it, but instead will save the complete code until the end. There are many comments in the code, which should help make its workings clearer.

The `\vsize`-s for our sample page are defined in the layout file by the macros `\setXIvone`, `\setXIvtwo`, etc. As I said before, the `\output` routine uses macros `\setvone`, `\setvtwo`, etc. It is `\setXIvdefs` which connects those two series and links our definitions to the macros used in the output routine. Thus, the first thing that happens in the `\output` routine is to determine which sheet we are in, and read in the appropriate definitions. This only happens if this is the first time that the `\output` routine has been entered, i.e. when `\pieces` is "0", meaning that no pieces have previously been saved for this sheet. This process is similar to that in `\beginsheet`.

The second thing that happens is the saving of the `\page` in the appropriate `\box` and the incrementing of `\pieces`. The `\vsize` of the the next `\page` is also set.

The `\output` routine cannot set the initial `\vsize`, that of the first piece on the first sheet. In the run file, the macro `\beginsheet` does that by using the macro `\setXIvone`. The `\output` routine can, however, set the `\vsize` for the first of the next sheet, through the macro `\setvnext` which has been defined by `\setXIvdefs` to be `\setXIvnext` which calls `\setXIIvone`. If `\setXIIvnext` is not defined, i.e. because the file **tex/layout.pg/12** has not been `\input`, the macro `\dummynext` must be used to override `\setvnext` and use the dummy layout defined in **tex/layout.pg/Z**. This is what I did in the sample run file.

After saving the current `\page`, the `\output` routine tests to see if `\pieces`, the number of boxes filled so far, is equal to `\numberboxes`, the number of boxes used in the page layout. If not, TeX leaves the `\output` routine and continues reading text. Although the `\output` routine allows for all ten boxes to be filled, ending with `\box0`, in practice this is not a good idea, since the calculation tools, defined above, use `\box0`, and could inadvertently wipe out the stored text.

If there are now enough boxes to fill the sheet, the `\output` routine prepares for the next sheet under the control of the flag set by `\dummynext`, and defines the `\buildit` macro to be `\buildXI`.

So far I have not mentioned where the header and footer are defined. I recommend creating a file of all the "edition-dependent" macros and having that `\input` within **tex/newsformat.x**.

At this point, all that is left to do is to set the footer according to the page number in `\count0`, and output the finished sheet, with a `\header`, the `\buildit` macro describing the body, and the `\footer`. Then I clean up the `\box`-es so there is not a lot of stuff hanging around using up memory (`\def\null{\hbox{}}`), and it is all done.

Summary

With the `\output` routine taking care of most of the processing, creating a complete page is mostly a matter of creating the two files which describe it. The run file names the page number and the article-files, and the layout file defines the `\vsize`-s and how the finished page is built. The only other consideration is that the article-files (the text) must have an explicit `\eject` whenever you want the parts treated as separate pieces in the layout, such as when `\hsize` changes.

Well, it's not as simple as "`\input basic \input text \end`", but this is pretty complex page layout we're doing. Simplifications from here might include (barring a simpler design in the first place) having a "library" of a couple dozen parameterized "standard" pages, and calling them up as needed.

Appendix: The Output Routine Code

```
%-----------------------------------------------------------------------
% \pieces contains the number of pieces already stored, as a character,
% not a count.   Start assuming that nothing has been put into the sheet,
% and the first text goes in piece 1.

\def\pieces{0}

%-----------------------------------------------------------------------
% Since this is read in as part of the initialization in newsformat.x,
% both of these will be overridden if \beginsheet is used.
% Being able to use \setZvone presumes that layout.pg/Z has already
% been input.

\def\sheet{Z}   % sheet is the number of the sheet we're working on.
                % If nothing else is stated, use "Z" as dummy/default.

\setZvone       % Before setting any type, we need the vsize
                % for the first piece of the sheet.

%=======================================================================
%
%               ***** THE OUTPUT ROUTINE *****
%

\output{

%-----------------------------------------------------------------------
% When we have a full page-piece, if this is the first piece of the
% sheet, then set the vsizes for the sheet that we're in.
% The initial vsize must be set elsewhere.
% The default \setZvone or \beginsheet does that.

\if 0\pieces
     {\ifx \sheet\I     {\setIvdefs}
\else{\ifx \sheet\II    {\setIIvdefs}
\else{\ifx \sheet\III   {\setIIIvdefs}
\else{\ifx \sheet\IV    {\setIVvdefs}
\else{\ifx \sheet\V     {\setVvdefs}
\else{\ifx \sheet\VI    {\setVIvdefs}
\else{\ifx \sheet\VII   {\setVIIvdefs}
\else{\ifx \sheet\VIII  {\setVIIIvdefs}
\else{\ifx \sheet\IX    {\setIXvdefs}
\else{\ifx \sheet\X     {\setXvdefs}
\else{\ifx \sheet\XI    {\setXIvdefs}
\else{\ifx \sheet\XII   {\setXIIvdefs}
\else{\ifx \sheet\XIII  {\setXIIIvdefs}
\else{\ifx \sheet\XIV   {\setXIVvdefs}
\else{\ifx \sheet\XV    {\setXVvdefs}
\else{\ifx \sheet\XVI   {\setXVIvdefs}
\else{\gdef\sheet{Z}\setZvdefs
     }}}}}}}}}}}}}}}}}
\else{
}       %---------------------------------------------------------------
        % Don't do anything here if this isn't the first piece.
```

```
%------------------------------------------------------------------------
% Now that vdefs are set, save the piece and set \vsize for the next.
% \numberboxes is set inside \setNNvdefs; where NN is I, II,..., XVI, Z.
% If there are already N pieces, then set pieces=N+1 and save the
% page in a new box.   Call the setv...  macro defined in \setNNvdefs.

     \if 0\pieces{\gdef\pieces{1}\save1\page\setvtwo}
\else{\if 1\pieces{\gdef\pieces{2}\save2\page\setvthree}
\else{\if 2\pieces{\gdef\pieces{3}\save3\page\setvfour}
\else{\if 3\pieces{\gdef\pieces{4}\save4\page\setvfive}
\else{\if 4\pieces{\gdef\pieces{5}\save5\page\setvsix}
\else{\if 5\pieces{\gdef\pieces{6}\save6\page\setvseven}
\else{\if 6\pieces{\gdef\pieces{7}\save7\page\setveight}
\else{\if 7\pieces{\gdef\pieces{8}\save8\page\setvnine}
\else{\if 8\pieces{\gdef\pieces{9}\save9\page\setvzero}
\else{\if 9\pieces{\gdef\pieces{0}\save0\page\setvnext}
\else{         %-------------------------------------------------------
    }}}}}}}}}} % This last \else should never be encountered, since
               % \pieces will only ever be 0-9.

%------------------------------------------------------------------------
% If we have all the pieces for the sheet, reset the number of pieces.
% Use the \nextisdummy flag to override \setvnext by calling \setZvdefs.
% Reset the \sheet id to the next one, and define the \buildit routine.
%
% Then proceed to build the full sheet.
%
% The "\if \pieces\numberboxes{...}" block continues to the end of the
% \output routine.   The matching \else, accessed each time a piece is
% saved but the sheet is not complete, is at the very end.

\if \pieces\numberboxes
         {\gdef\pieces{0}
          \if T\nextisdummy{\setZvdefs}\else{}
          \setvnext
          \ifx \sheet\I   {\xdef\sheet{\II}  \gdef\buildit{\buildI}}
     \else{\ifx \sheet\II  {\xdef\sheet{\III} \gdef\buildit{\buildII}}
     \else{\ifx \sheet\III {\xdef\sheet{\IV}  \gdef\buildit{\buildIII}}
     \else{\ifx \sheet\IV  {\xdef\sheet{\V}   \gdef\buildit{\buildIV}}
     \else{\ifx \sheet\V   {\xdef\sheet{\VI}  \gdef\buildit{\buildV}}
     \else{\ifx \sheet\VI  {\xdef\sheet{\VII} \gdef\buildit{\buildVI}}
     \else{\ifx \sheet\VII {\xdef\sheet{\VIII}\gdef\buildit{\buildVII}}
     \else{\ifx \sheet\VIII{\xdef\sheet{\IX}  \gdef\buildit{\buildVIII}}
     \else{\ifx \sheet\IX  {\xdef\sheet{\X}   \gdef\buildit{\buildIX}}
     \else{\ifx \sheet\X   {\xdef\sheet{\XI}  \gdef\buildit{\buildX}}
     \else{\ifx \sheet\XI  {\xdef\sheet{\XII} \gdef\buildit{\buildXI}}
     \else{\ifx \sheet\XII {\xdef\sheet{\XIII}\gdef\buildit{\buildXII}}
     \else{\ifx \sheet\XIII{\xdef\sheet{\XIV} \gdef\buildit{\buildXIII}}
     \else{\ifx \sheet\XIV {\xdef\sheet{\XV}  \gdef\buildit{\buildXIV}}
     \else{\ifx \sheet\XV  {\xdef\sheet{\XVI} \gdef\buildit{\buildXV}}
     \else{\ifx \sheet\XVI {\xdef\sheet{Z}    \gdef\buildit{\buildXVI}}
     \else{\gdef\sheet{Z}\gdef\buildit{\buildZ}
          }}}}}}}}}}}}}}}}
          \if T\nextisdummy{\gdef\sheet{Z}}\else{}
              % Z is used for dummy pages and for sheets beyond \XVI.
```

```
    %---------------------------------------------------------------------
    % Having set up for the next sheet and defined \buildit,
    % proceed to build a complete output sheet:

    \ifeven0{\gdef\footer{\evenfoot}}
    \else   {\gdef\footer{\oddfoot}}

    \vbox to \sheeth
            {\header
            \buildit
            \footer
            \advcount0
            }

    %---------------------------------------------------------------------
    % Clean up by emptying all the boxes.

    \save1\null\save2\null\save3\null\save4\null\save5\null\save6\null
    \save7\null\save8\null\save9\null\save0\null

    %---------------------------------------------------------------------
    % This is the end of ``\if \pieces\numberboxes {..."
    % sheet-building block.

    }
\else{} % This else is accessed each time the sheet is not complete.

} % end of output block.

%=====================================================================

%=====================================================================
%
% (c) Copyright 1981, 1982, 1983, 1985 by August Mohr.    The described
% macros and TeX code may be used by TUG members for internal company
% and personal purposes only.    This TeX code may not be distributed, in
% any form, to non-members without written permission from the author.
% Any commercial use of this code, such as but not limited to producing
% a product, or part of a product, for sale, may only be with written
% permission of the author.    Any distribution of these macros and/or
% other code, must contain this copyright notice.
%
%=====================================================================
```

Letters

Transatlantic TeX

To the Editor:

On the morning of May 9 I found a message on my terminal from Uriel Frisch at the Observatoire de Nice overlooking the Riviera. The note said that if I received the message, we were connected through ARPANET, and if I replied he would send a TeX version of our joint paper. By noon I had a copy in my file space on the VAX at GA. They use a VAX at the observatory. After lunch, Phil Andrews did a little editing, and by 3pm we had a beautiful laser printed copy in our hands. It all worked the first time, and in one day.

The required editing was straightforward. The people in Nice changed the characters {, }, ^, and # to \1, \2, \3, and \4, respectively, to guard against errors in transmission. In addition, we found that all the tildes had been changed to carets.

The world is getting smaller all the time.

John Greene
GA Technologies
P. O. Box 85608
San Diego, CA 92138

Author vs. proofreader

To the Editor:

As told in the book “Les meilleurs amis du monde”, by Raymond Castans, published by J.-C. Lattès :

In a novel by Jacques Audiberti, a woman was saying to her lover:

“Je t'aime, donc je te dirai ‘tu’. ‘Tu’ est beau.” (I love thee, so I shall call thee ‘thou’. ‘Thou’ is beautiful.)

In the second sentence, “tu” is mentioned, not used; that is why the verb is “est” instead of “es”; to say “Thou art beautiful” in French, one says: “tu es beau” : the “t” makes all the difference.

Well, that was too subtle for the proof-readers. They changed the verb to “es”. The author was prepared for that, so he changed it back to “est” on the proofs. Second proof-reading: same result, same reaction from the author. Sixth proof-reading: same result, same reaction.

Finally, printing time came. The author went to the place where the book was being printed to make sure things would be done properly. The “t” was there, so he left satisfied.

When he got his first copy of his book, the first thing he checked was that. To his horror, the “t” was gone! As he learned later, during the printing a proof-reader passed by, picked up a torn proof page from the ground, and ordered the machines stopped so that the “t” could be removed...

Jean-Luc Bonnetain
Bonnetain@SUMEX-AIM.Arpa

News & Announcements

Calendar

Several Beginning TEX courses are listed below. All are intended for non-technical users, with the exception of the course to be held at Stanford before the TUG meeting, August 5–9.

1985

June 10–14	Beginning TEX, Harvard University, Cambridge, Massachusetts

Vanderbilt University, Nashville, Tennessee:

June 12–14	Intermediate TEX
June 18–19	TEX macro writing
June 20–21	TEX output routines

Illinois Institute of Technology, Chicago, Illinois:

July 17–19	Intermediate TEX
July 22–26	Beginning TEX

* * * * *

July 29– Aug. 2	Beginning TEX, Rutgers University, Busch Campus, Piscataway, New Jersey

Stanford University, Stanford, Calif.:

Aug. 5–9	Beginning TEX (for individuals with technical backgrounds)
* Aug. 8–9	Short Course: The elements of METAFONT design
Aug. 8–9	Short Course: Introduction to the internal workings of TEX82
* Aug. 12–14	TEX Users Group Annual Meeting
* Aug. 15–16	TEX Short Course: Traditional typography meets LATEX

Rutgers University, Busch Campus, Piscataway, New Jersey:

Aug. 19–21	Intermediate TEX
Aug. 22–23	TEX macro writing
Aug. 26–30	Beginning TEX

* * * * *

Sep. 15	TUGboat Volume 6, No. 3: Deadline for submission of manuscripts

PROTEXT II, Dublin, Ireland:
(see announcement, p. 98)

Oct. 21–22	Short Course: An introduction to text processing systems: Current problems and solutions
Oct. 23–25	Conference: The Second International Conference on Text Processing Systems

* * * * *

1986

Jan. 9–12	TEX exhibits, American Mathematical Society Annual Meeting, New Orleans, Louisiana
Jan. 31	TUGboat Volume 7, No. 1: Deadline for submission of manuscripts (tentative).
April 14–16	International Conference on Text Processing and Document Manipulation, University of Nottingham, England (see announcement, p. 98)

* Change from dates announced previously

Announcement: PROTEXT II The Second International Conference on Text Processing Systems

Two events on the subject of *Text Processing Systems*, jointly comprising PROTEXT II, will be held at the Berkeley Court Hotel, Dublin, Ireland, in October 1985:

- Short course: *An Introduction to Text Processing Systems: Current Problems and Solutions*, 21–22 October;
- The Second International Conference, 23–25 October.

The short course is a carefully coordinated introduction to computer-aided generalised text processing systems. It is intended both for the beginner, who has little previous knowledge of the subject, and also for the experienced person who wishes to broaden his knowledge into new areas. The scope of the short course will be the same as that of the Conference, but it will be at an introductory level. The Lecture Notes will be published in book form in advance of the short course.

The conference provides a forum for the discussion of the latest research on computer-aided generalised text processing systems. It is intended that participation will be evenly divided between academia and industry, and an even balance will be maintained between hardware and software topics. Several formal discussions on controversial issues of current interest will be organised. The Proceedings of the Conference will be published in book form within two months of the conference.

The Proceedings of PROTEXT I, held in Dublin, October 1984, are now in print. They include the full texts of 29 papers presented at the conference: the 6 keynote papers (including Pierre MacKay's "TEX's coming of age") and 23 shorter contributions. Lecture notes from a Workshop held in association with PROTEXT I are also available.

The deadline for contribution of papers to PROTEXT II, 1 June, has already passed. It is still possible to register to attend either event, or to reserve copies of the publications. Further information may be obtained from

PROTEXT II Organising Committee
c/o Boole Press Limited
P. O. Box 5, 51 Sandycove Road
Dún Laoghaire
Co. Dublin, Ireland
(+353-1) 808025

Announcement: International Conference on Text Processing and Document Manipulation

An international conference on *Text Processing and Document Manipulation* will be held at the University of Nottingham, England, on April 14–16, 1986. The conference is being organized by the Electronic Publishing Specialist Group of the British Computer Society in association with a number of sponsors. It is intended primarily for people actively working in the field, but will also provide a number of state-of-the-art surveys for those who wish to find out what is available. An associated exhibition will provide an opportunity for participants to see systems in action.

Although this conference is not associated with TUG in any way, it is expected that some of the papers will be based on TEX or METAFONT, and thus should be of interest to TUG members.

The conference will cover all aspects of computer document preparation, text processing, and printing. It will include considerations of document design, digital typography, authoring systems, videotex, and electronic publishing. Papers are invited on any topic related to document processing; they are due to be submitted not later than 1 September, 1985.

Further information can be obtained from the Program Committee Chairman:

Mrs. Heather Brown
Computing Laboratory
The University
Canterbury, Kent CT2 7NF, England

TUG 1984 Financial Report with Comparisons

May 31, 1985

	Actual 1982	Actual 1983	Actual 1984	Budget 1985
Income:				
Membership/Subscriptions				
Individual/Library[1,2]	$ 11,935	$ 13,629	$ 16,402	$ 20,000
Postage	588	1,358	-0-	-0-
Institutional Membership[1,2]				
Educational	200	5,780	8,000	8,000
Non-educational	200	3,612	10,200	11,100
Publications				
Back issue sales[3]	2,156	5,094	6,482	5,000
Other publications[4]	376	634	6,037	4,500
Meetings[2,5]				
Annual meeting	11,025	15,209	28,940	20,000
Other meetings/courses	31,518	14,187	60,655	140,000
Manufacturers reps' fees	300	450	425	600
Other				
Videotape sales/rental	2,200	4,998	2,090	2,000
Advertising/mailing list sales	-0-	325	962	1,000
Royalties/contributions[6]	-0-	-0-	4,649	10,000
Interest income	-0-	-0-	-0-	8,000
Total income	$ 60,498	$ 65,276	$ 144,842	$ 210,200
Expenses:				
TUGboat				
Printing	$ 2,220	$ 2,620	$ 3,402	$ 3,120
Postage	1,143	934	1,711	1,680
Editorial services	3,460	3,772	6,424	4,200
Clerical services	180	182	505	240
Computer expenses	2,100	2,220	1,979	2,640
Meetings*[,2,5]				
Annual meeting	2,130	2,200	2,746	3,000
Other meetings/courses	4,460	5,111	30,295	39,000
Other				
Other publications[4]	225	474	1,233	2,400
Royalties	-0-	-0-	540	-0-
ANSI meetings*[,7]	2,503	1,280	-0-	-0-
Legal and tax consulting*	-0-	-0-	1,836	2,000
Postage, general mailings	1,335	3,605	3,025	3,960
Printing back issues[3]	-0-	7,953	2,108	-0-
Printing, other	1,212	426	4,192	4,200
Administrative support[8]	-0-	12,923	25,376	57,600
Clerical services*	3,120	5,708	3,888	3,600
Subsidies*[,9]	-0-	-0-	-0-	-0-
Videotape duplication*	-0-	2,553	1,179	800
Computer expenses	1,455	4,637	5,813	4,800
Programming[10]	-0-	-0-	10,647	-0-
Miscellaneous[11]	1,688	1,622	3,102	2,640
Total expenses	$ 27,231	$ 58,220	$ 110,001	$ 135,880
Summary:				
Balance forward (Actual/Budget/Est.)	$ (8,660)	$ 24,607	$ 31,663	$ 66,504
Income (Actual/Budget/Est.)	60,498	65,276	144,842	210,200
Expenses (Actual/Budget/Est.)	$ (27,231)	$ (58,220)	$ (110,001)	$ (135,880)
Balance	$ 24,607	$ 31,663	$ 66,504	$ 140,824

TUG 1984 Financial Report (Continued)

Notes: All expense figures through 1984 include an AMS overhead charge of approximately 20%; also, 1985 expense figures, except those with an asterisk (*), include an overhead charge of 20%. Except as indicated, these remarks apply to the 1984 year.

1. As of December 31, there were 1,109 members/subscribers, including 76 Institutional Members: 39 educational; 37 non-educational.
2. Advertising of TUG and the TUG Meeting/Courses was accomplished through a news release to 19 trade publications, several of which are known to have published the notice, in addition to direct mailings to members and former members.
3. In 1984, 360 back issues were sold. Owing to this popularity, new-issue print orders are set high enough to accommodate expected back issue sales for 3 years.
4. More than 600 copies of Arthur Samuel's "First Grade TEX" and 10 copies of Hewlett-Packard's "The HP TEX Macros" were sold.
5. 153 members attended the 3-day TUG meeting at Stanford in August. In addition, 75 individuals attended the "Book Design Utilizing TEX" and 58 attended the "TEX for Beginners" course.
   The 1985 Stanford University Conference schedule is published in the Calendar Section of this issue; in addition, two-, three- and five-day TEX courses are being offered throughout the summer at Illinois Institute of Technology and Harvard, Rutgers and Vanderbilt Universities. Registration fees for the 1985 Stanford Meeting have been reduced substantially, which will result in a significant reduction in meeting receipts.
6. Don Knuth designated a portion of the royalties from sales of *The TEXbook* to be donated to TUG. $3,324 was received in 1984. In addition, Kellerman & Smith contributed $1,325, which represents a $25 royalty to TUG for each copy of VAX/VMS TEX sold. In January 1985 a contribution of $500 was received from Textset, Inc.
7. Larry Beck, Grumman Data Systems, replaced Lynne Price as TUG's liaison with ANSI X3V1.8,
8. The services of Ray Goucher, amounting to approximately half time in 1984 and full time in 1985, are purchased from the AMS. (1985 amounts include salary, AMS benefits and overhead.)
9. The Steering Committee made this amount available to the Finance Committee to subsidize travel and membership/participation fees for individuals when appropriate.
10. Reprogramming to improve the functioning of the TUG data base.
11. Postage/express charges, telephone tolls and supplies, plus programmer and clerical services not associated with production of TUGboat.

Respectfully submitted,
Samuel B. Whidden, Treasurer

TeX and METAFONT reports available from the Stanford Computer Science Department

The reports listed below are available from
Publications, Computer Science Department,
Stanford University, Stanford, CA 94305
(415) 497-4776

California residents add 6.5% to total charge.
M = microfiche only, $2.00 + tax

CS	824	Tung	"LCCD – a language for Chinese character design"	M
CS	828	Knuth, Plass	"Breaking paragraphs into lines"	M
CS	848	Tang	"On the problem of inputting Chinese characters"	M
CS	870	Plass	"Optimal pagination techniques for automatic typesetting systems"	M
CS	886	Knuth	"Concept of a Meta-font"	2.35
CS	901	Fuchs, Knuth	"Optimal font caching"	2.60
CS	914	Gu, Hobby	"Using string matching to compress Chinese characters"	2.50
CS	960	Zabala	"Interacting with graphic objects"	4.95
CS	965	Ghosh	"An approach to type design and text composition in Indian scripts"	6.45
CS	966	Bigelow, Ghosh	"A formal approach to lettershape description for type design"	3.60
CS	974	Gu, Hobby	"A Chinese meta-font"	2.70
CS	977	Liang	"Word Hy-phen-a-tion by com-put-er"	4.70
CS	978	Knuth	"Lessons learned from METAFONT"	3.05
CS	980	Knuth	"The WEB system of structured documentation"	8.10
CS	981	Knuth	"Literate Programming"	2.45
CS	985	Samuel	"First Grade TeX"	3.95
CS	1013	Désarménien	"How to Run TeX in French"	3.30
CS	1027	Knuth	"A Torture Test for TeX"	9.25

The following preliminary report is not yet available from Stanford but can be obtained from

Maria Code, 1379 Sydney Drive,
Sunnyvale CA 94087 (415) 735-8006

Knuth "TeXware"

Available in a technical bookstore or from Maria Code:

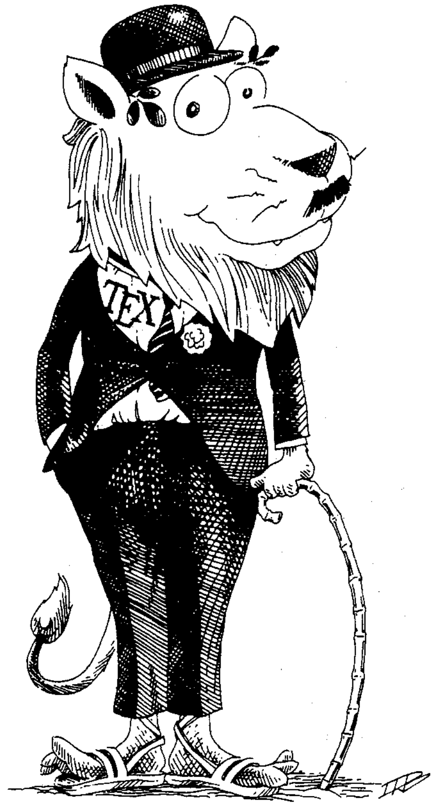
Knuth *The TeXbook*, published by Addison-Wesley (1984)
Knuth *TeX: The Program*, published by Addison-Wesley (1985)

MicroTeX™ is TeX

"I am pleased to verify that MicroTeX has passed the 'TRIP test suite,' giving results indistinguishable from those obtained on mainframe computers. It has indeed earned the right to be called TeX." — Donald E. Knuth (April 22, 1985)

Some things you should know about MicroTeX:

* TeX 1.4 implemented by David Fuchs for the IBM PC/XT, AT, and PC with a hard disk.
* Detailed system and driver documentation included.
* Fonts modified specifically for the Epson and IBM dot matrix printers.
* Optional font installation procedures to reduce disk storage space.
* Complete micro-, mini- and mainframe computer transportability.
* Ability to redirect DVI files to a disk, to be printed later.
* Driver "parallel processing" mode to increase document printing speed.
* Full utilization of extra memory.



Addison-Wesley is the publisher of several books related to TeX, including: *The TeXbook* and *TeX: The Program*, by Knuth, as well as the forthcoming *LaTeX: A Document Preparation System*, by Leslie Lamport, and the Proceedings of the First European Conference on *TeX for Scientific Documentation*, edited by Dario Lucarella.

MicroTeX System Requirements:

* IBM PC/XT, AT, or IBM PC with a hard disk, or an MS-DOS compatible with a hard disk.
* MS or PC DOS, version 2.0 or later.
* Minimum 512K memory and a partition on your hard disk with at least 1.5 megabytes storage for the MicroTeX system with minimum fonts, or 4 megabytes with all the fonts included in the package.

MicroTeX supports these printers:

* Epson RX-, FX-, MX-, 80 and 100 printers
* IBM Graphics printers and Dot Matrix printers with GRAFTRAX and GRAFTRAX PLUS.
* Okidata 92 or 93 (with Plug'n Play).

TeX is a trademark of the American Mathematical Society, MicroTeX is a trademark of Addison-Wesley Publishing Company, Inc.; IBM is a registered trademark of International Business Machines, Inc.; Epson is a registered trademark of Epson, Inc.; Okidata is a trademark of Oki Electric Industry Company.

MicroTeX Order Form 35184

☐ Yes, please send me ___ copy(ies) of MicroTeX at $495.00 each.
☐ My check is enclosed (Orders paid by check will be shipped free of charge.)
☐ Visa ☐ MasterCard (Interbank#__________) ☐ American Express

Card No.____________________ Exp. Date____________________

Signature____

☐ Please bill my company. Your company order form and P.O. number must accompany your order. Thank you.

Name____________________ Title____________________

Firm/Institution____________________ Department____________________

Street____________________ City/State/Zip____________________

Telephone(____) ____________ Ext.______

☐ I am interested in MicroTeX, but would like more information.
☐ Please keep me informed about all MicroTeX developments and related products.

Send your order to:

Addison-Wesley Publishing Company
Educational and Professional Technologies Division
Reading, MA 01867
Telephone: (617) 944-6795

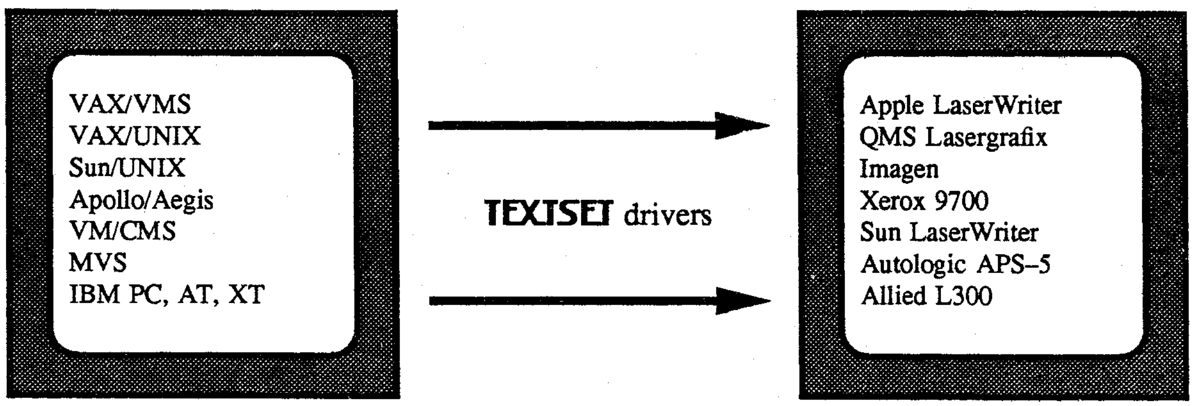
TEXTSET, Inc. 416 Fourth Street • P.O. Box 7993 • Ann Arbor • MI • 48107 • (313) 996-3566
technical composition software and services

June 19, 1985

Attention TeX Users:

Every day I hear the same sorts of questions from TeX users across the country. They usually go something like this: "I have an XX Computer System and have been using TeX for the past year. It would really be nice if I could get from my DVI file on the XX to a printed page off of my YY Printer. Can you help?" Some of them go like this: "I've been using TeX with my YY Printer for some time now, but I can only use a few of the available fonts this way and there are lots of other restrictions. Can you help?"

Well, I'm happy to report that most of the time the answer is that we *can* help. **TEXTSET** has developed versatile TeX device drivers for many popular laser printers and phototypesetters.



If you've been wondering about device support for your TeX system, just give me call. I'd be glad to send you product documentation and try to arrange a demonstration for you.

In fact, if you have any questions at all about TeX, just give me a call. If I can't give you an answer, I'll connect you with someone who can.

Sincerely,

Bruce Baker
Director
Market Development

ps. I produced this advertisement on an Apple LaserWriter provided to **TEXTSET** by Adobe Systems, Inc.

Request for Information

The TEX Users Group maintains a database and publishes a membership list containing information about the equipment on which members' organizations plan to or have installed TEX, and about the applications for which TEX would be used.

Please answer the questions below, in particular those regarding the status of TEX and the computer(s)/operating system(s) on which it runs or is being installed. (Especially for IBM and VAX, the operating system is more relevant than the model.)

If it has not yet been done for your site, please also answer the questions about output devices on the other side of this form, obtaining information from the most knowledgeable person at your installation if necessary. If this information has already been provided by another TUG member, please indicate that member's name, and the information will be repeated. If you need more space than is provided here, feel free to use additional paper.

If your current listing is correct, you need not answer these questions again. Your cooperation is appreciated.

- *Send completed form with remittance* (checks, money orders, UNESCO coupons) to:
  TEX Users Group
  P. O. Box 594
  Providence, Rhode Island 02901, U.S.A.
- *For foreign bank transfers* direct payment to the TEX Users Group, account #002-610871, at:
  Rhode Island Hospital Trust National Bank
  One Hospital Trust Plaza
  Providence, Rhode Island 02903-2449, U.S.A.
- *General correspondence* about TUG should be addressed to:
  TEX Users Group
  P. O. Box 9506
  Providence, Rhode Island 02940-9506, U.S.A.

Name: ____________________
Home [ ] Bus. [ ] Address: ____________________
____
____
____

QTY	ITEM	AMOUNT
	1985 TUGboat Subscription/TUG Membership (Jan.–Dec.) – **North America** New (first-time): [ ] $20.00 each Renewal: [ ] $30.00; [ ] $20.00 – reduced rate if renewed before January 31, 1985	
	1985 TUGboat Subscription/TUG Membership (Jan.–Dec.) – **Outside North America** * New (first-time): [ ] $25.00 each Renewal: [ ] $35.00; [ ] $25.00 – reduced rate if renewed before January 31, 1985	
	TUGboat back issues, $15.00 ** per issue, circle issue(s) desired: 1980 (v. 1) #1; 1981 (v. 2) #1, #2, #3; 1982 (v. 3) #1, #2; 1983 (v. 4) #1, #2; 1984 (v. 5) #1, #2	
	The TEXbook by Donald E. Knuth – $17.00 each	
	The TEXbook: Errata and Changes (included with TUGboat) – additional copies $3.00 each	
	First Grade TEX: A Beginner's TEX Manual by Arthur L. Samuel – $6.00 each	
	User's Guide to the HP TEX Macros by Susan Daniels – $6.00 each	
	TEX and Metafont: Errata and Changes (final edition, September 1983) – $4.00 each	
	TEX Lectures on Tape (see cover 3, Vol. 6, No. 2 and cover 3, Vol. 5, No. 2)	

* Air mail postage is included in the rates for all subscriptions and memberships outside North America.
** Discount: 5–7 copies, 10%; 8 or more, 15%

TOTAL ENCLOSED: ____________
(*Prepayment in U.S. dollars required*)

* * * *

Membership List Information

Institution (if not part of address):

Title:
Phone:
Specific applications or reason for interest in TEX:

My installation can offer the following software or technical support to TUG:

Please list high-level TEX users at your site who would not mind being contacted for information; give name, address, and telephone.

Date:
Status of TEX: [ ] Under consideration
[ ] Being installed
[ ] Up and running since
Approximate number of users:
Version of TEX: [ ] SAIL
Pascal: [ ] TEX82 [ ] TEX80
[ ] Other (describe)

From whom obtained:

Computer(s) and operating system(s):

Please answer the following questions regarding output devices used with TEX
unless this form has already been filled out by someone else at your site.
Use a separate form for each output device.

Name ______________________ Institution ______________________

A. Output device information
   Device name
   Model
   1. Knowledgeable contact at your site
      Name
      Telephone
   2. Device resolution (dots/inch)
   3. Print speed (average feet/minute in graphics mode)
   4. Physical size of device (height, width, depth)

   5. Purchase price
   6. Device type
      [ ] photographic [ ] electrostatic
      [ ] impact [ ] other (describe)

   7. Paper feed [ ] tractor feed
      [ ] friction, continuous form
      [ ] friction, sheet feed [ ] other (describe)

   8. Paper characteristics
      a. Paper type required by device
         [ ] plain [ ] electrostatic
         [ ] photographic [ ] other (describe)

      b. Special forms that can be used [ ] none
         [ ] preprinted one-part [ ] multi-part
         [ ] card stock [ ] other (describe)

      c. Paper dimensions (width, length)
         maximum
         usable
   9. Print mode
      [ ] Character: ( ) Ascii ( ) Other
      [ ] Graphics [ ] Both char/graphics
   10. Reliability of device
      [ ] Good [ ] Fair [ ] Poor
   11. Maintenance required
      [ ] Heavy [ ] Medium [ ] Light
   12. Recommended usage level
      [ ] Heavy [ ] Medium [ ] Light
   13. Manufacturer information
      a. Manufacturer name
         Contact person
         Address

         Telephone
      b. Delivery time
      c. Service [ ] Reliable [ ] Unreliable

B. Computer to which this device is interfaced
   1. Computer name
   2. Model
   3. Type of architecture *
   4. Operating system

C. Output device driver software
   [ ] Obtained from Stanford
   [ ] Written in-house
   [ ] Other (explain)

D. Separate interface hardware (if any) between host computer and output device (e.g. Z80)
   1. Separate interface hardware not needed because:
      [ ] Output device is run off-line
      [ ] O/D contains user-programmable micro
      [ ] Decided to drive O/D direct from host
   2. Name of interface device (if more than one, specify for each)

   3. Manufacturer information
      a. Manufacturer name
         Contact person
         Address

         Telephone
      b. Delivery time
      c. Purchase price
   4. Modifications
      [ ] Specified by Stanford
      [ ] Designed/built in-house
      [ ] Other (explain)

   5. Software for interface device
      [ ] Obtained from Stanford
      [ ] Written in-house
      [ ] Other (explain)

E. Fonts being used
   [ ] Computer Modern
   [ ] Fonts supplied by manufacturer
   [ ] Other (explain)

   1. From whom were fonts obtained?

   2. Are you using Metafont? [ ] Yes [ ] No

F. What are the strong points of your output device?

G. What are its drawbacks and how have you dealt with them?

H. Comments – overview of output device

* If your computer is "software compatible" with another type (e.g. Amdahl with IBM 370), indicate the type here.

TeX82 Order Form

The latest official versions of TeX software and documents are available from Maria Code by special arrangement with the Computer Science Department of Stanford University.

Eight different tapes are available. The generic distribution tape contains the source of TeX82, WEB, and the latest (prototype) version of WEB **METAFONT**, standard test programs for TeX and **METAFONT**, a few "change" files, the collection of fonts in TFM format, and other miscellaneous materials; a PASCAL compiler will be required to install programs from a generic tape. The TeX distribution tapes include the AMS-TeX, LaTeX and HP TeX macro packages; other macro packages will be added as they become available. The special distribution tapes are for the indicated systems only, and should be ordered for these systems instead of a generic tape. Two tapes are PXL font collections covering various magnifications at 200/240 dots/inch and 300 dots/inch respectively.

Each tape will be a separate 1200 foot reel which you may send in advance or purchase (for the tape media) at $10.00 each. Should you send a tape, you will receive back a different tape. Tapes may be ordered in ASCII or EBCDIC characters. You may request densities of 6250, 1600 or 800 (800 is discouraged since it is more trouble to make).

The tape price of $82.00 for the first tape and $62.00 for each additional tape (ordered at the same time) covers the cost of duplication, order processing, domestic postage and some of the costs at Stanford University. Extra postage is required for first class or export.

Manuals are available at the approximate cost of duplication and mailing. Prices for manuals are subject to change as revisions and additions are made. It is assumed that one set of manuals will suffice you. If you require more than two sets, please write for prices since we must ask for more money for postage and handling.

Please send a check or money order (payable on a US bank) along with your order if possible. Your purchase order will be accepted, as long as you are able to make payment within 30 days of shipment. Please check this out before sending a purchase order since many large firms seem to be unable to make prompt payment (or don't worry about it).

The order form contains a place to record the name and address of the person who will actually use the TeX tapes. This should *not* be someone in the purchasing department.

Your order will be filled with the most recent versions of software and manuals available from Stanford at the time your order is received. If you are waiting for some future release, please indicate this. Orders are normally filled within a few days. There may be periods (like short vacations) when it will take longer. You will be notified of any serious delays. If you want to inquire about your order you may call Maria Code at (408) 735-8006 between 9:30 a.m. and 2:30 p.m. West Coast time.

If you have questions regarding the implementation of TeX or the like, you must take these to Stanford University or some other friendly TeX user.

Now, please complete the order form on the reverse side.

TEX82 Order Form

**** TAPES **** density (6250, 1600 or 800) = ____________

TEX generic distribution tapes (PASCAL compiler required):

________ † ASCII format ________ † EBCDIC format

TEX distribution tapes in special formats:

________ † VAX/VMS Backup format ________ IBM VM/CMS format

________ † DEC 20/Tops-20 Dumper format ________ * IBM MVS format

† Includes WEB **METAFONT** * Not yet available; call before ordering

Font tapes:

________ Font library (200/240 dots/inch) ________ Font library (300 dots/inch)

________ Total number of tapes.

Tape costs: $82.00 for first tape; $62.00 for each additional.

Tape cost = $ ____________

Media costs: $10.00 for each tape required.

Media cost = $ ____________

** MANUALS **

________ TEX: The Program – $28.00 ________ A Torture Test for TEX – $8.00

________ WEB – $10.00 ________ TEXware – $8.00

________ TEXbook – $20.00 ________ LATEX (preliminary edition) – $8.00

Manuals cost = $ ____________

California orders only: add sales tax = $ ____________

Domestic book rate: no charge.
Domestic first class: $2.50 for each tape and each manual.
Export surface mail: $2.50 for each tape and each manual.
Export air mail to North America: $4.00 each.
Export air mail to Europe: $7.00 each.
Export air mail to other areas: $10.00 each.

Postage cost = $ ____________

(Make checks payable to Maria Code) Total order = $ ____________

Name and address for shipment: Person to contact (if different):

____________________ ____________________

____________________ ____________________

____________________ ____________________

____________________ ____________________

Telephone ____________

Send to: Maria Code, DP Services, 1371 Sydney Dr., Sunnyvale, CA 94087

3/85