## Historical review of TEX3

Peter Flynn

**Abstract**

While clearing out cupboards during lockdown, I came across several oddities, including this review of TEX3 which `.exe` magazine asked me for in April 1991, during negotiations into being taken over. They subsequently collapsed as a print publication, and the article was therefore never published, meaning it's still my copyright. It was interesting to see what I had picked up on for them, as they had asked me to write it from the point of view of a new TEX user with experience in other systems.

It referenced five figures which were long since lost, but which I have reconstructed from memory and the descriptions. The original was formatted in unmarked monospace, according to the publisher's requirements. This version has been used without substantive textual change for *TUGboat*, but the plaintext symbolic notations like `\TeX\` have been replaced with the genuine logos, acronyms and URIs have been marked, some previously unnoticed typos have been corrected, and oddities of plaintext markup have been converted to LATEX.

Most companies, addresses, electronic access, etc., mentioned no longer exist. The references are kept as a matter of nostalgia and history. The article begins below the rule.

---

*The grand-daddy of DTP systems offers a radically different approach to typography on the desktop. Peter Flynn investigated and found it still has a lot to offer.*

### Introduction

Out of the current soup of DTP systems, a few leaders have emerged among the visual packages. *Ventura*, *Pagemaker*, and Quark *XPress* each has its own benefits and peculiarities, as any user will have discovered, but there is another form of DTP altogether, known as 'logical', which is where systems like TEX fit in.

Contrary to popular opinion, DTP was not invented in the mid-80s on the Apple Mac, but in 1978 on a DEC-20 minicomputer. The program was TEX, and it allowed users for the first time to produce printers' quality typesetting from their terminals, using the new 'laser-beam' printers.

Heady stuff indeed at the time, given that word-processing had barely yet been given a name.

The difference between visual and logical systems is straightforward enough once you understand what it is you're trying to do (the terms 'visual' and 'logical' were coined ca. 1988 [4]).

Visual systems rely on the operator's skill in manipulating a bitmapped screen image, usually by hand, eye and mouse. The relationship between text structure and appearance is undefined, or specified by simple tags in style sheets. The method allows freehand adjustment to appearance, such as positioning, scaling, distorting or overlaying of graphics and text, but places a high degree of reliance on manual dexterity and the visual judgment of the user. Each page is normally made up individually, so that any knock-on effect of changes must also be attended to.

Logical systems rely on the author's or editor's skill to bind structure tightly to appearance, using commands specified in a style file or embedded in the text to define positioning and typography. The text file is edited externally, and processed in its entirety to ensure the effect of changes is properly accounted for from page to page. Adjustments to appearance are made by changing the commands in the text, and by the use of fully programmable macros. This largely removes the need for manual or visual intervention in long or repetitive texts, or in applications where automation, regularity or dimensional accuracy is required.

### Origins

TEX is, in effect, a document compiler, taking a piece of source 'code' (your text) and acting on the instructions you embedded in it. The original program was written by Don Knuth to typeset a new edition of his famous *Art of Computer Programming* series, because ordinary commercial typesetters either couldn't handle the technical matter, or charged too much. (The lowered 'E' in the TEX logotype distinguishes it in print from TEX, an old Honeywell editor, for copyright reasons, and emphasizes the relationship with typography.)

He generously made TEX publicly available, so it can be had in commercial and non-commercial versions for almost every machine in existence from Apple IIGS, Amiga, and Archimedes, up through PC and Mac, to Sun and other UNIX systems, VAX/VMS, IBM and other mainframes, and even the Cray (that should shift some lead!). All implementations are compatible with each other, differing only in memory capacity and speed, depending on the operating environment. This level of availability must make it the most widely-spread DTP system in use: certainly in today's increasingly networked environment it is a significant factor to be able to establish completely compatible DTP across almost every platform with no need for investment in special hardware.

TEX has had a mixed reception in the past. It has variously been criticised for being difficult to use, lacking graphics, only having one typeface, for not being WYSIWYG and for not being a visual-based system (about as sensible as criticising a fish for not being a chicken). Part of the reason is that, coming from the public domain, there was no one to trumpet its abilities or explain its working. Some reviewers who should know better have not distinguished between visual and logical systems, and some were making their comments based on old versions, hearsay, and incomplete information. However, as we shall see, it is in fact quite easy to use, can handle plenty of graphics, lots of typefaces, and has some of the best screen displays around.

Quality is one area that has always been commented on favourably, and it is carefully guarded: for an implementation to call itself TEX it must pass a stringent test specified by Knuth.

**Availability**

Perhaps one of the reasons for the 'well-kept secret' ethos surrounding TEX is the fact that the original source code was placed in the public domain. Anyone can have this code for the asking: it was written in WEB, a 'literate programming language' also devised by Knuth, which produces Pascal source. There are many commercial versions of TEX as well, written by companies or individuals who have optimised the code for specific operating systems, and ship the system as a package. Because of its popularity and heavy use in research and technical typesetting, and in publishing trade and the academic field, there has been little need for glossy full-colour whole-page adverts for it in magazines aimed at the domestic or office market. This is perhaps a pity, as many visual DTP users who need reliable, accurate and automated formatting are still doing their work manually, unaware that TEX exists!

Those who have anxiety attacks at the mention of the phrase 'public domain' can take comfort in the fact that it is the source code itself which is available, so if you have misgivings about viruses, you can check it and compile it yourself. There are no known instances of an infected copy of TEX being distributed, and all the public versions available from the regular sources are checked out before being made available. If in doubt, of course, buy a commercial copy.

My review commercial copy (PC-TEX) came from UniTEX Systems (for details see end) and my public-domain ones (emTEX and SBTEX) came from the TEX server archive (`TEX.AC.UK`) at Aston University, Birmingham, via the email network. I also used an older commercial VAX/VMS version (Kellerman & Smith) and a PD one for the Macintosh called OzTEX. I didn't have a UNIX machine available, so I was unable to investigate UNIX-flavoured TEX personally, but given the rigorous quality control, it is reasonable to assume it performs as claimed.

There are other public-domain and shareware versions too, from a variety of bulletin boards (e.g. CIX), user groups and file server hosts on the wide-area networks all over Europe and the United States. UK, Irish and continental European users of normal internetwork email can order from Aston or Heidelberg (for details see end) but if you're trapped on BT Gold or EirMail, forget it and contact your national User Group. In addition there are other regular commercial versions such as *Textures* and TurboTEX (for the Mac and PC respectively) and a low-cost commercial version for the PC (DOSTEX).

**Installation**

TEX for desktop machines needs about 5MB of disk space to live in, and runs in 512KB of memory. On larger machines, where there has traditionally been more disk space, implementations tend to spread themselves around a bit more. If you want to add more fonts, design packages, CAD, specialist macros, foreign-language hyphenation and the endless other goodies, you will need more room. On the other hand, if all you want to do is publish, say, a typeset database listing, you can trim right down to just over 1MB (plus your own data/text storage, of course).

PC-TEX came in the usual style of PC software binder: the `INSTALL` routine worked perfectly and took around 15 minutes to unpack everything from the nine disks. emTEX arrived as several `.BOO` archive files (encodings into printable characters which allow 8-bit binary to traverse the 7-bit email networks) which DEBOOed and unZIPped (with the `-d` option) to recreate all the files and the whole directory structure, in about 20 minutes. I wish some other software I could mention unpacked with even half the care and intelligence that has been spent on organising both these versions. The Mac version I unpacked in a similar manner from BINHEXed STUFFIT archives without problems. The VAX software originally came on standard half-inch magtape and took rather longer, because of the need to establish logical names and pathways to cater for multiuser operation.

Drivers are available for most printers (i.e. those emulating IBM Graphics and ProPrinters, Epson FX and LQ, NEC Pinwriter, PostScript and HP LaserJet) and you can get or make drivers for almost anything which puts marks on paper, even a fax: if you have

something esoteric, there is generic driver code available for you to roll your own. There's even a driver for a CalComp pen plotter if you want letters several feet high! Because TEX produces an output file which is entirely device-independent, you aren't tied to using any particular supplier's driver, or even any particular machine.

TEX itself knows nothing about fonts except the height and width of each character, plus a few other parameters, which it reads from font metric files. The print drivers, however, normally use bitmap font files, not outlines, so you need to order the right resolution with your printer driver (usually 180, 240, 300 or 360 dpi depending on your printer). Using bitmaps is a two-edged sword: you tend to get far better quality, but you are restricted to those sizes you have on disk. However, there is a companion type-design program for TEX called METAFONT, which can create additional optimised font files at any resolution for any dot-matrix or laser printer (and some typesetters). METAFONT is well worth having if you want extra odd font files at specific sizes, and if you're into type design, this is probably one of the most powerful tools around. If you use PostScript, of course, you don't need any bitmaps, just the font metrics. I shall have more to say about fonts later.

```
\noindent This is a very short test, to make
sure the program is working correctly. This
paragraph starts flush left, and shows the
appearance of {\bf bold face} type.

This paragraph is indented, and shows the
appearance of {\it italics}. It contains the
math formula $z*n=x*n+y*n$. Such a formula
might also be displayed $$z*n=x*n+y*nS$$ to
make it more prominent.

\bye
```

**Figure 1**: Source text of the file `TEST.TEX`

## Documentation

The manual for all implementations of TEX is *The TEXbook* by Donald Knuth, published by Addison-Wesley [2]. This explains everything, in progressively finer detail. The early chapters are excellent, and can get even a total novice producing the goods very quickly. Learning TEX is not difficult, but like any DTP system, it is nevertheless not trivial, and the later chapters need some careful reading, as there is a lot in them. There are plenty of worked examples, though, far more than in any other manual I have ever seen (you can never have enough, in my view).

The documentation accompanying PC-TEX is a more digestible spiral-bound manual which summarises all the introductory matter of *The TEXbook* and looks less forbidding for a beginner [5].

The PD versions came with documentation on how to set up and run the programs (in most cases better explained than most 'professional' manuals) but they do assume you have *The TEXbook* for details of how to format text.

For the user who just wants to produce something with a relatively simple standardised layout, the LATEX macro package which comes with TEX is the easiest route, as it has several carefully-designed sample document styles. The LATEX book which describes these is also an Addison-Wesley publication but comes free with PC-TEX [3].

There are some very good introductory booklets available free as TEX input files, so you just process and print them. The best two I found were *A Gentle Introduction to TEX* by Michael Doob [1] and *Introduction to TEX on VAX/VMS* by Joe St Sauver [6]. This last one sounds a bit operating-system specific, but if you just ignore the VMS bits it is a very good guide. There are others covering the many add-ons to TEX, again usually supplied on disk as `.TEX` files for you to print yourself.

This is a very short test, to make sure the program is working correctly. This paragraph starts flush left, and shows the appearance of **bold face** type.

This paragraph is indented, and shows the appearance of *italics*. It contains the math formula $z * n = x * n + y * n$. Such a formula might also be displayed

$$z * n = x * n + y * nS$$

to make it more prominent.

**Figure 2**: Output of `TEST.TEX` (HP LaserJet III)

## Operation

TEX operates in a radically different manner to visual DTP. You create your document in a plain ASCII file, embedding formatting commands in the text. When you run the program, it processes your file and makes a compact, portable device-independent typeset file, which is used by the preview and print drivers to produce the output. Because it doesn't use a graphic display during processing, repetitive production jobs such as database publishing can run unattended once they are set up.

A test file (`test.tex`) which demonstrates the method of operation is supplied as standard (see

Figure 1 and Figure 2 for the input and output). The DVI (DeVice-Independent) file can be printed on any supported printer without the need to reprocess your source text: just run the relevant print driver. The concept is similar to PostScript, in that once you have the final version off your laser or dot-matrix printer, you don't have to reformat or reprocess for a higher-resolution device. Unlike PS, however, you are not restricted to specific output devices or fonts.

All the formatting commands are mnemonic, so they are fairly straightforward to learn. They all begin with a backslash (e.g. `\parindent=2em` or `\baselineskip=15pt`, see Figure 1). The use of an ASCII file means you can continue to use your favourite editor (or wordprocessor in 'non-document' or 'ASCII export' mode), so there are no new menus or keystrokes to learn. No editor is supplied, as it is assumed that every machine already has one in some form or another. I'm uncertain about this: although it is easy to get many excellent editor/wordprocessor programs, it surely could have been possible to throw in a good public-domain editor, or license a shareware one, even if only for beginners. I suppose you could even use EDLIN if you were a masochist.

If you've got existing wordprocessor files, there are converter programs from WordPerfect and MS-Word into TEX and LATEX.

As with any DTP system, once you go beyond elementary formatting you do need to be aware of what you want to do. Typographical training is a much-neglected part of office life so far, although it is noticeably easier to get good typographic quality in your output using TEX than with some visual systems. However, as I mentioned earlier, there are so many predefined layouts available that many users don't bother to delve into the deeper recesses of TEX's abilities.

The macro facilities mean there is a lot of underlying power available. TEX is in fact a typographical programming language, which means you can program decision-making `\if` statements to modify the appearance depending on the text being processed. The effect of this is to dramatically increase the reusability of your text: using LATEX's predefined styles, for example, you can turn a text file into a business report by marking the component parts (title, sections, subsections etc.) and adding three commands at the top and one at the bottom. If your analysis was so good you wanted to turn it into a chapter of your new book, change the document style `report` to `book` and that's it. To publish it as an article, change `book` to `article`, and change the chapters into sections. This is the whole essence of logic-based DTP: once the structure of the text is marked, formatting and reformatting becomes relatively trivial — just change the macro definitions.

There are some tricks and traps of course. One small confusion for beginners arises over the use of curly braces: TEX uses them to group together text which you want treated in a particular way, so for example, to italicise text you type `{\it your text}` in curly braces, with the '`\it`' command immediately inside the opening brace: this restricts the effect of italics to the text in the braces. So far so good. But curly braces are also used to delimit arguments, so for example, typing `\centerline{some text}` centres the text between the margins. There are very cogent reasons for this, but it means a second or so's thought until you get used to it.

Once you've processed your file, you will want to see the results. The WYSIWYG screen in emTEX is one of the best I have seen. It is configurable for all the conventional PC screens from mono CGA to the 64/256 colour VGA display, and on the higher resolution devices it uses grey-scaling to give the image better definition. You can shrink the image small enough to fit two A4 pages on a single screen, or enlarge it big enough to see only a few words at a time. An on-screen ruler lets you measure dimensions, and the image can be rotated, mirrored and inverted. Despite the fact that TEX drivers use bitmap fonts, the previewer does not require its own set of fonts at screen resolution: it can use whatever you have around for your printer, reducing your disk storage needs. Other suppliers also have very good previewers, especially the Preview program from Arbortext.

The print drivers have similar facilities for positioning the output on the page, so it is possible to print portrait or landscape, mirrored or inverted, even on dot-matrix printers. You can easily print selected pages in any order, with multiple pages per sheet in various orientations, so it is possible to make booklets correctly paginated for folding straight off the printer.

The only thing that causes an initial stumbling block is the sequence of edit–process–view–print. This is common to all logical systems, in that the entire file has to be processed before it can be seen. The reason is partly historical, in that TEX grew up before bitmapped screens were commonplace; and partly inherent in a logic-based system, in that type placement on the page naturally depends on what fit onto previous pages. PC-TEX does in fact sell a version which displays as you go, and similar versions are available for the Amiga and some other fast machines, but these are very much the exception.

### Timings

TEX is fast: I clocked just over one-and-a-half seconds per page on a 16MHz 80386 clone with 640KB and a 28ms unfragmented hard disk for processing plain continuous text in emTEX. By comparison, PC-TEX's 80386-specific executable on the same machine but using 2MB of memory gave me under half a second per page on the same file.

Printing speed is limited by the throughput of the printer itself. The HP LaserJet driver printed at the full eight pages a minute, with only a brief pause at the start to download the font glyphs.

Printing full-page graphics on a dot-matrix printer is always tedious, but I counted 34 seconds per A4 page of solid text on an Epson LQ800, which is acceptable for short drafts.

emTEX runs very happily under DesqView/386, my own preference for a working environment, with PC-Write in one window, the TEX engine in another and the WYSIWYG screen in another (plus my usual assortment of comms, spreadsheet and database). I haven't tested it under Windows, but there's a Windows (Microsoft Paintbrush-style) screen driver in emTEX, and the TEX program and print drivers should cause no problems as DOS tasks.

### When something goes wrong

TEX's error messages are explicit but technical. Pressing `H` when it pauses at an error gives some further explanation, and often points you to the relevant section of *The TEXbook*. It is assumed that you have read some of this, or similar explanatory documentation, because otherwise a message such as '`Overfull \hbox at line 432`' is not going to mean much (in fact it refers to a justification or hyphenation problem, the h(orizontal) box being the page element it is trying to justify).

'Errors' in this sense means one of two things: either you have mistyped a command word, which is easy to correct, as it tells you what and where; or there is a fault in the logic of your instructions, which is harder to spot. Unfortunately, computers cannot guess your intentions, although TEX makes a damn good try. Missing a closing curly-brace is a common typing error, and so is forgetting to turn off a special mode, such as mathematical setting, but this is not logically determinable until your text tries to do something that TEX knows cannot be done in whatever mode you were in, like finding a paragraph-end in the middle of text supposed to be centred. All the program can do is report what went wrong and where: it's up to you to find out why, and this can be tricky in complex work, as there may be a considerable amount of innocuous text between the cause of the problem and the actual place where TEX spotted things going astray.

Justification and hyphenation problems are rare in normal text, because TEX justifies an entire paragraph at one go, rather than line by line. This results in a wonderfully smooth and professional finish compared with the woefully ragged and uneven spacing found in some systems, but there are occasions when you have to fix an obtuse word with a discretionary hyphen.

The most common mistake I made was forgetting to turn off a typestyle such as boldface, resulting in the remainder of the document being in bold type, but that is easy to spot and easy to fix, as it is obvious from the display or printout where the error starts. Omitting or wrongly delimiting an argument causes unexpected results for the unwary: typing `\centerline` and then omitting any argument and carrying on with the text makes TEX centre the first character of the next word and then complain that it can't fit the surrounding text on the line. Perfectly reasonable, but a strong case for RTFM (Read The Flaming Manual).

Fortunately the results of RTFM are well worth it: after the learning curve had flattened a little (an afternoon), doing some automated formatting, even including some mathematics, produced such a professionally-formatted page that the idea of going back to placing everything with the mouse by hand is one I can't contemplate.

TEX has extremely few bugs, as Don Knuth has been offering hard cash to anyone who can find one, starting at 1 cent and doubling on each occasion. As the current rate is only $40.96 after 12 years, you are not likely to find that many more.

When you need more serious help, therefore, it will be with typography, not bugs, and you will need access to a design or typographic expert fluent in TEX. In the case of commercial software, this is often provided by the supplier, and either included in the price or charged as support. In the case of public-domain TEX, you can either call your user group (see details at end), or if you have email access, send a message to one of the many support conferences, where there are hundreds of experts who will gladly help. For specialist or large-scale development, there are also many TEX consultants who charge normal commercial rates.

### Fonts

The default font is Computer Modern (see Figure 2), a redrawing (by Knuth) of Monotype Modern 8A, and resembling Century Schoolbook, but less bulky. It suits excellently for continuous text, which is what

TEX was originally designed for. Because of this, and because Computer Modern has all the scientific and mathematical symbols built-in (many more than in your average DTP system), a lot of TEX users never bother to get away from it. This is a pity, as it has led to the myth that TEX only works with the CM typeface. A browse through *TUGboat*, the TEX Users Group journal, soon explodes the myth: TEX works fine with Bitstream fonts, PostScript, and anything from the Hewlett-Packard downloadable stable, as well as with other METAFONT designs.

Mind you, you have to pay money for some of these, but I used Bitstream Swiss and Humanist (Helvetica and Optima), the Type 1 Adobe PostScript fonts, and some HP softfonts generated by Glyphix, all without problems (although Glyphix doesn't provide a pound [sterling] sign, being American). These fonts have to be in TEX format, which means a small conversion program from PC-TEX Inc. for Bitstream outlines. If you don't have a PostScript printer, you can buy the Adobe fonts as bitmaps from the Kinch Computer Company, and there is a public-domain HPtoTeX program available to convert HP downloadables. The Austin Code Works also has a whole stash of fonts already in TEX format, and as I said earlier, if you restrict yourself to a PostScript printer, you can do away with bitmaps entirely, although you lose some of the mathematical and scientific symbols, and the interletter spacing is a little poxy in places.

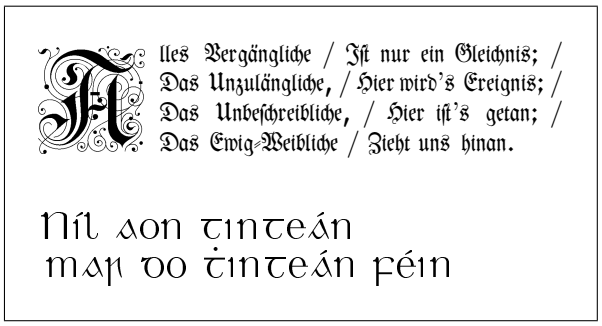


**Figure 3**: Fraktur with decorated initial, and an Irish text typeface

METAFONT is a programming language for making your own fonts. Font design itself is a decidedly non-trivial activity, but the program can be used with several public-domain font definitions to create font files. There is an increasing number of new fonts being done in METAFONT, and I tested a mixture: some well-established ones like Pandora by Neenie Billawala and Hermann Zapf's Euler (from the American Mathematical Society); Helvetica and Times (more akin to Morison's original, not Times New Roman) from the MetaFoundry (Dublin, Ohio); and two new ones, a modern Irish typeface by Micheál Ó Searcóid of University College, Dublin; and a new Fraktur and Schwabacher by Yannis Haralambous of CITI Lille, which has some stunning decorated initials (see Figure 3 and Figure 4).

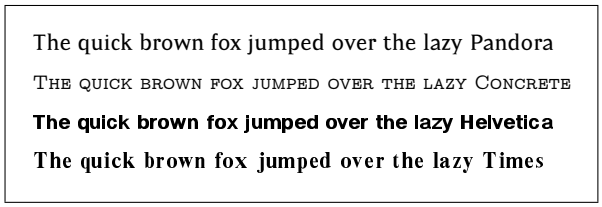


**Figure 4**: Comparison of some text fonts used in TEX

Interestingly, *TUGboat* says that the resident 'Wizard of Fonts' is Hermann Zapf himself, so they clearly have some good high-level advice on tap. There are many non-Latin fonts available: Cyrillic, Greek, Devanagari, Arabic, Turkish, and Japanese, even a Tengwar script for Tolkien fans! The ScholarTEX package provides Persian, Ottoman, Pashto, Urdu, Hebrew, Yiddish, Syriac, Armenian, Greek, Ancient Greek and Latin, Fraktur and Schwabacher, Anglo-Saxon, Irish, Glagolitic, Coptic, Calligraphic Arabic and Sanskrit. There's a CM-compatible International Phonetic Alphabet (IPA) from the University of Washington.


**Figure 5**: Difference between design sizes

TEX provides the ability to scale any individual font or the whole document to any size (but you still need the bitmaps at the right dot-density in order to print), but *The TEXbook* makes it clear that 5pt type is best done with a real 5pt design, and 50pt type with a 50pt design, rather than by scaling another design size up or down (see Figure 5). This is of course exactly what typographers have been doing since Gutenberg, but some authors of modern systems failed initially to understand the need, and gave only outline fonts. This was partly acknowledged by Adobe's use of 'hints', but it has led to some disastrous results in terms of legibility and

appearance, as anyone who has tried scaling 110pt type down to 5pt can see. Using bitmaps of fonts at different design sizes does place some restrictions on users' disk space, but the selection of Computer Modern shipped with TEX provides a good everyday working subset as a mixture of design sizes and scaled fonts. It is clear that the reason for METAFONT is that you can generate fonts at the size required in a few minutes and then junk the bitmaps when you are finished, if you are short of space (some systems, such as the version for Amiga, do the font generation automatically when you reference a font which is not on disk).

### Graphics

The TEX typesetting engine itself, of course, is not a graphics processor: no logical system is, and even most visual DTP systems make poor drawing packages. Most DTP users are accustomed to scanning artwork and then embedding it into the document, and the same applies to TEX, although you can also use the more elementary line-drawing abilities of the LATEX macro package, and the more advanced facilities of PICTEX or TEXCAD.

Scanning artwork usually means touching it up in a paintbox program such as PC Paintbrush, and you can then make it into a character in a font file, using a neat little routine from Micro Programs Inc, or if your printer can't handle large downloadable glyphs, you can use the same routine to make a printstream output file which the print driver can handle itself.

This means you can also embed printstream output of some other application in the TEX output (e.g. Encapsulated PostScript, or HP's PCL generated by spreadsheets, business graphics packages, CAD, drawing programs etc.). There are also several routines developed for using half-tones (photographs), in monochrome or colour, although the 300 dpi resolution of a normal laser printer scarcely does them justice.

### Extras

Most implementations of TEX include the macro package LATEX, which provides the facilities for structured documents, with variations on autonumbering sections, subsections, paragraphs etc.; automated table-of-contents and indexes; simple diagram and graph-drawing; forward and backward references and a whole stack of other stuff, including BIBTEX, a bibliographic database program.

There's PICTEX, for drawing more complex diagrams such as flowcharts or organisation charts; TEXCAD, a little CAD package which produces LATEX source code as output; and SliTEX for making multicolour overhead transparency separations. This has recently been extended by David Love of the Daresbury Labs to produce output on colour PostScript printers — what he terms TEXnicolor.

METAFONT, the font-making program, is supplied with emTEX plus the complete source code for all the Computer Modern fonts, and some font manipulation tools to go with it. With most commercial implementations, though, METAFONT is a chargeable extra.

The public-domain bolt-ons are legion. In addition to the wide range of styles under LATEX, assorted generous people have written and donated macro packages to do circuit diagrams, molecular (hydrocarbon) diagrams, critical text editions, style files for journals, music typesetting, dictionaries, database output, newsletters, calendars; TEX has been used for most things at one time or another over the last dozen years, so there is a wealth of experience to draw on. There are also plenty of agencies and consultants who will design styles in TEX to your specifications.

### Conclusion

The gripes I referred to earlier (lack of fonts, graphics and styles, visual presentation and perceived difficulty of use) seem to have been a problem of users' perception, rather than any deficiency in the TEX system itself. TEX is, however, something of an acquired taste, but has the capability to outperform most other systems in its field in quality, flexibility and automation.

Directly comparing TEX with a visual system is not strictly valid, as the two kinds of DTP are usually aimed at different requirements, although there is a large area of overlap. Those users who feel uneasy with an asynchronous visual feedback are probably more productive with a visual system, but they may have to spend more effort to achieve the same level of quality.

Equally, there are tasks more suited to visual systems than to logical ones: attention-grabbing advertising and newsletters which need to rely on a complex blend of overlapping colour graphics and type for optical appeal are far faster to produce using a visual system than a logical one.

In the end it comes down to suiting the facilities offered to the typographic skill and knowledge of the user, as well as to the demands of the task. Despite the initial appeal of visual systems, TEX is still worth a careful look.

## A Original biography

Peter Flynn is in charge of research and academic computing at University College Cork, Ireland. He has been Technical Consultant to a large City computer bureau, deputy DP manager for one of the UK Training Boards, and a teacher of programming and systems analysis. His hobbies are early music, reading and surfing. He can be reached by email as `pflynn` on BIX and CIX, and through the wide-area networks as `cbts8001@iruccvax.ucc.ie`.

## B Network sources of public-domain software

Email users (e.g. CIX, JANET, HEANET, BITNET, UUCP etc.) can retrieve files from the Aston archive of TEXware: send a one-line electronic mail message to `texserver@tex.ac.uk` saying
`send[tex-archive]00directory.list`
(this is a large file containing a directory of everything in the archive). Individual files can then be retrieved by the same method.

A one-line mail to `listserv@dhdurzl.bitnet` saying just `index` will retrieve the file list from the server in Heidelberg (files are got by sending `get` followed by the filename). Some additional TEX material is kept at `mailserv@ymir.claremont.edu`.

All network servers respond to the single-word command `help` by sending you a help file about what is in them and how to access them. BT Gold and EirMail are unfortunately not connected to the international email networks, but Gold 400 is.

In case of difficulty, contact the UKTEX Users Group, c/o Aston University Computer Centre, Birmingham (email `uktex@aston.ac.uk`).

## C Commercial software

- PC-TEX. UniTEX Systems, 12 Dale View Road, Sheffield.
- µTEX. Arbortext Inc, 535 West William Street, Ann Arbor, Michigan 48103, USA (fax +1 313 996 3573).
- *Textures* (Mac), £350; CM fonts as PostScript outlines, £300. TEXpert Systems, PO Box 1897, London NW6 1DQ (fax +44 71 433-3576).
- For other operating systems, contact the UK TEX Users Group (address above) for details of suppliers.
- WordPerfect-to-TEX converter, $249. K-Talk Communications, 30 West First Avenue, Suite 100, Columbus, Ohio 43201, USA (fax +1 614 294 3704).
- TurboTEX, $150; Adobe bitmaps, $200. Kinch Computer Company, 501 South Meadow Street, Ithaca, NY 14850, USA (fax +1 607 273 0484). Kinch also provides a fax driver for TEX.
- Bitmap fonts. Austin Code Works, 11100 Leafwood Lane, Austin, Texas 78750-3464, USA (fax +1 512 258 1342, email `info@acw.com`);
- Capture, $100; TEXPIC (graphics), $79. Micro Programs, Inc, 251 Jackson Avenue, Syosset NY 11791-4117, USA (tel +1 516 921 1351).
- ScholarTEX (out shortly): Yannis Haralambous, rue Breughel 101/11, FR-59650 Villeneuve d'Ascq, France (tel +33 20.05.28.80, email `yannis@FRCITL81.bitnet`).

## D Pull quotes

1. All implementations are compatible with each other, differing only in memory capacity and speed, depending on the operating environment.
2. Omitting or wrongly delimiting an argument causes unexpected results for the unwary: a strong case for RTFM (Read The Flaming Manual).
3. Despite the initial appeal of visual systems, TEX is still worth a careful look.

## References

[1] M. Doob. A Gentle Introduction to TEX. Aston University, UK: UK TEX Archive, Jan. 1990. `ctan.org/pkg/gentle`.

[2] D. Knuth. *The TEXbook*. Addison-Wesley, Boston, MA, 18th edition, May 1990.

[3] L. Lamport. *LATEX: A Document Preparation System*. Addison-Wesley, Boston, MA, Jan. 1986.

[4] L. Lamport. Document Production: Visual or Logical? *TUGboat* 9(1):8, Jan. 1988. `tug.org/TUGboat/tb09-1/tb20lamport.pdf`

[5] M. Spivak. *The PC-TEX Manual*. Personal TEX, Inc., San Francisco, CA, 01 1985.

[6] J. St Sauver. Introduction to TEX on VAX/VMS. Claremont University, CA: YMIR TEX Archive, Jan. 1991.

◇ Peter Flynn
Textual Therapy Division,
Silmaril Consultants
Cork, Ireland
Phone: +353 86 824 5333
`peter (at) silmaril dot ie`
`blogs.silmaril.ie/peter`