

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 coordinates, 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  $\text{T}_{\text{E}}\text{X}$  at Texas A&M, as well as announcing the availability of a C version of  $\text{T}_{\text{E}}\text{X}82$ .

"An Elementary Sum" was output on an Epson LQ1500 (180dpi  $\times$  180dpi) using 200dpi fonts (180dpi fonts are not yet available). It was  $\text{T}_{\text{E}}\text{X}$ ed on a TI/PC running  $\text{PCT}_{\text{E}}\text{X}$  (it also works on  $\text{MicroT}_{\text{E}}\text{X}$ ), 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  $\text{WEB}$  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  $\text{T}_{\text{E}}\text{X}$  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  $\text{T}_{\text{E}}\text{X}$  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  $1 \leq k \leq n$  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  $\cot^2(\theta_k)$ ,  $k = 1, \dots, n$ . 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  $\sin(\frac{k\pi}{2n+1}) \neq 0$  for  $k = 1, \dots, n$ , 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  $p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$ , 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  $\tan(x) \geq x \geq \sin(x)$ . Thus,

$$\tan(\theta_k) \geq \theta_k \geq \sin(\theta_k)$$

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

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

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

An application of the sandwich theorem completes the proof:

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

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

\output= . . .

\random

### T<sub>E</sub>X in C

T<sub>E</sub>X 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 INIT<sub>E</sub>X 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 T<sub>E</sub>X 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 T<sub>E</sub>X 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 T<sub>E</sub>X; 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 T<sub>E</sub>X 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.

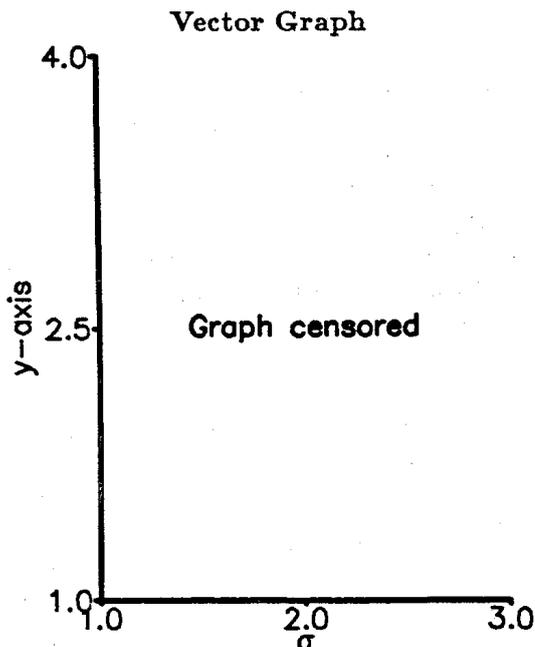
We were surprised by the amount of compaction that was possible with our format. As we were developing drivers for microcomputers at the time and disk storage was at a premium, we wondered if the drivers could use the packed file format directly. We modified our drivers to look for and unpack a packed 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 768Kbytes of memory, a 10Mbyte hard disk, and PCT<sub>E</sub>X. Except for the graphics material, the output was previewed, before printing, on the TI screen.



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 T<sub>E</sub>X:**  
*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.

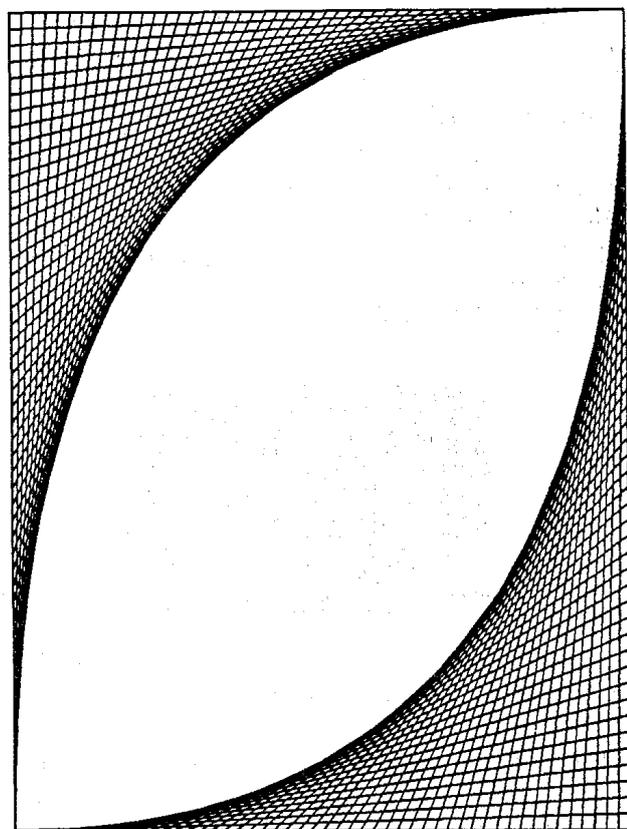
The log table he used for a desk (the original one used by Noah's frustrated serpents) was strewn with an ant farm, rubber cement, some various papers and drawings, and a few sorry looking balloons. "I just cut a strip from that red balloon and looped it into a Möbius strip. Then, I patched the hole 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 T<sub>E</sub>X 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 T<sub>E</sub>X 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 T<sub>E</sub>X 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