# Technical Word Processing:

## PROS & CONS

*By Murray Sargent III*

Some people are interested in my work on the quantum theory of multiwave mixing. It's fascinating, really! In contrast, most people want to talk about technical word processing, my other love. No wonder, most scientists and engineers know they can benefit greatly from technical word processing; only a few relate to spontaneous emission and multiwave mixing.

This article spills the beans: how you can turn out twice as many published words with substantially greater polish than you could with the old Selectric typewriter cut-and-paste methods. Science is competitive. If you're still using the secretary/selectric methods, you have a severe handicap, and the IRS should give you two more exemptions. Here's why. First the cons, then the pros, and finally the bigger picture.

A favorite expression in today's word processing is, "What you see is what you get." Unfortunately for us scientists, what we see is generally not what we want. We want something resembling the beautiful proportionally spaced mathematical text published in scientific journals. What we see on a typewriter or usually on screen is text with equal character widths, pigeonholed into 80 columns by 25 lines.

In principle, computer screens can show facsimiles of the typeset text on screen by using bit-mapped graphics. This technique allows a program to specify the value of each dot on the screen. Different characters can have different widths, built-up formulas can be represented by appropriate combinations of dots, and facsimiles of the ultimate typeset text are possible, even if they lack in resolution.

Unfortunately, the computer has to work much harder to place all those dots where they belong. To achieve instantaneous screens (and correspondingly reduced eye fatigue), today's microcomputers need the help of hardware character generators, which return us to the 80 × 25 pigeonhole screens, unlike the typeset text we prefer. Larger computers have the raw power to handle bit-mapped text but either are very expensive or are stuck with terminals operating at inadequate speeds like 9600 baud. That's the major con: To have speed, you can't work with a facsimile of the typeset page—yet. You have to work with an encoding of the ultimate text. As this article shows, the encoding is actually remarkably readable.

Other cons include a need to touch type and the cost of a microcomputer like the IBM PC. For most of us the cost is irrelevant considering the time saved. But many active scientists have been duped by the society of old that said typing is for secretaries. The old society lacked the foresight to predict the computer age. For the next 10 years, anyhow, typing will remain the most efficient way to enter text into computers. Without question, touch typing is the most important course I took in high school. If you don't touch type, try one of the typing tutor programs. You'll learn fast.

### Text manipulation

By now everyone is familiar with the marvelous speed with which ordinary text can be manipulated on screen. You can move text around, insert old files to make new versions, send multiple letters as quickly as a single letter, reprint a document with changes at a moment's notice. By working directly on screen, instead of with pencil and paper or dictation, you can go

through multiple drafts in the same time you originally needed for the first draft. Subsequent revisions do not introduce new typos as retyping can. Word processing streamlines the clerical aspects of writing, allowing you to concentrate on the ideas. At present, at least, you still have to do the thinking. In short, you can cater to your perfectionist tendencies as never before and still spend less time producing a document.

The problem for scientists and engineers is in representing mathematical formulas on computers. There are three ways used today: 1) the precise but verbose typesetter methods used by programs like Knuth's TEX and UNIX's EQN/TROFF, 2) the Selectric lookalike method by which you move the cursor approximately to where a character belongs, and where it'll stay, and 3) mine. Having typed numerous manuscripts and more than half the book *Laser Physics* on a Selectric typewriter, I know how slow method two is. Having invented the first language capable of typesetting mathematical equations, SCROLL (published in 1969), I know how slow and unreadable method one is.

In response, or perhaps in self-defense, I developed a linear equation notation that is remarkably efficient in representing mathematical formulas. Coupled with an instantaneous graphics preview facility revealing built-up equation form, this method enables you to produce mathematical documents almost as fast as ordinary text.

The idea was inspired by arithmetic expressions in high-level programming languages, but unlike those languages it capitalizes on certain advances in computer hardware made over the last 20 years. Specifically, to obtain

$$\frac{\alpha_2}{\beta_2{}^3 + \gamma^3}$$

you type $\alpha_2/(\beta_2{}^3 + \gamma^3)$. The Greek letters are given by typing an appropriate alphabetic key while pressing the Alt and maybe the Shift keys. The program automatically measures the size of the numerator and denominator, centers one over the other, and prints. Like the typesetter methods, you don't have to measure anything—the computer automatically prints the desired proportionally spaced text. Like the Selectric method, formulas are usually readable in edit mode. Unlike either, you can type formulas in 10 times as fast.

Although very useful as is, the method isn't perfect and is still evolving. Hardware character generators are typically still limited to 512 characters if you're lucky, 256 if you're not. Mathematical text requires a minimum of 1024 characters, not including composites like $\dot{x}$. To allow for many possibilities with limited resources, we use various kludges like \$E for $\mathcal{E}$ and $\gamma_1 a$ for $\gamma_a$. As equations get more complicated, they become harder to read. For example, the equation [$P(\alpha)$ distribution for a photon number state]

$$P(\alpha) = [n!/2\pi r(2n)!]\exp(r^2)[-\partial/\partial r]^{!}2n\ \delta(r)$$

is converted to

$$P(\alpha) = \frac{n!}{2\pi r(2n)!}\ \exp(r^2)\ -\ \frac{\partial}{\partial r}^{2n}\qquad 6(r).$$

Certainly the built-up form is easier to read; hence the importance of the graphics preview facility. Unlike TEX and EQN/TROFF, not all mathematical constructs are represented. Almost all those needed by physicists are readily available, but more work is clearly needed for the general case.

The benefits of technical word processing described so far make it highly desirable for the scientist and engineer. Technical documents can be produced faster, better, and more accurately. The ultimate benefits transcend this arena. Three important possibilities seem apparent: accuracy of journal typesetting, technical documents on optical disks, and the science machine. Accuracy of journal typesetting is virtually here. The American Institute of Physics already accepts documents prepared with EQN/TROFF format. If you get your document right in-house, it'll be right in print. As described above, this is unfortunately no easy task, since EQN/TROFF has a cumbersome syntax. It should be straightforward to translate the linear equation format into either TEX or EQN/TROFF, thereby obtaining the advantages of all three.

Storing technical documents on optical disks easily readable by microcomputers will benefit science and engineering immeasurably. These little disks can hold up to 500 megabytes, which amounts to about 83,000 standard journal pages if the formulas and figures can be appropriately encoded. A shelf-full of such optical disks could contain copies of all the physics journals published to date. With appropriate indexing, we'd be able to have thorough access to the literature in our own offices.

To realize this miracle, formulas and figures have to be encoded. Using the precise but verbose syntaxes of EQN/TROFF and TEX doesn't make sense. With a bit more work, the concise linear equation format can. Similarly, figures would have to be reduced to standard line drawing commands wherever possible. In this fashion, authors could submit papers on diskettes, streamlining the editorial and typesetting processes and immediately allowing much wider access to the literature through microcomputer retrieval.

Perhaps the most exciting idea is the science machine. This hypothetical machine streamlines research not only in technical word processing, computations, and information retrieval, but in a synergetic way by integrating all three. The authors of FORTRAN named their computer language after FORmula TRANslation, but they went only halfway. Arithmetic expressions in FORTRAN or any other high-level language don't look anything like the original formulas written in scientific notation.

*Continued on page 37* ▶