

## Scientific Book Publishing Using T<sub>E</sub>X

Tony Siegman  
Stanford University

A few months ago my publisher and I completed the final tasks in the production of a very large scientific textbook (31 chapters, 1300 pages, 600 figures, 2000 display equations) using T<sub>E</sub>X. This included preparation of both the source files and the final typeset pages using computer-based text-editing and typesetting facilities at Stanford University. In response to suggestions from Eric Berg of Stanford and Ray Goucher of TUG, this note is an attempt to set down a few of the lessons I learned about producing a book using T<sub>E</sub>X, including a few of the things I wish I'd thought about earlier in the project. These comments are focused primarily on organizational and procedural questions, rather than on technical problems involving T<sub>E</sub>X coding, macro packages, or computer hardware.

Book publishing is an old and highly refined art, with a highly developed and widely standardized set of procedures as to how an author's work goes from initial manuscript to bound books. Traditional book publishing is also an industry marked by a great deal of subcontracting. Book design, copy editing, preparation of art work, typesetting, printing and binding, and even publication coordination are very often subcontracted out by the primary publisher to firms who supply those services to many different publishers. Most major publishers, in fact, have probably never owned a printing press or even a typesetting machine.

This widespread use of subcontracting, along with the traditional character of the publishing industry, means that all the procedural steps from manuscript to final book—the way the work flow is organized and scheduled, the way records are kept and objects are labelled, the proofreaders' and printers' marks that are used, the understandings as to who does what and in what order—have become highly standardized and widely understood by all concerned. Technological developments which interrupt or short-circuit this understanding have a disturbing effect.

(Widespread subcontracting of the production steps has also had a second more subjective effect, in my opinion. It seems to me that many traditional publishers have not, at least in the past, developed a technologically oriented view of their industry. Over many years, when I wanted to converse with editors and publishers about word processors, text-editing programs, holograms, floppy disks, laser printers, and their impact on publishing technology,

the publishing people seemed to me to focus only on things like printers' union demands, and whether it was better to send material to England or Asia for low-cost typesetting. They seemed to me to have little interest in technological developments in their industry. This disinterest in the technological aspects of publishing is of course now rapidly changing.)

Publishing a book using T<sub>E</sub>X, especially when this involves direct author input of the T<sub>E</sub>X source code, as well as author (or at least noncommercial) generation of the final typeset pages, obviously throws many of the traditional procedures and organizational arrangements of book publishing into confusion. To gain maximum advantage from book preparation in T<sub>E</sub>X, while retaining traditional book quality, new and different organizational procedures will have to evolve. I would not want to say myself, from my limited experience to date, how these new procedures should develop. It seems clear that this evolution has yet to be completed, however, or even to have gotten very well started.

In my own case, the source file for my book was prepared, and then expanded, revised, reshuffled, and rearranged through innumerable versions over a 10 year period, largely by myself and my secretary, Judy Clark, working on home and office terminals connected to various Stanford mainframe computers. Early versions started out being formatted in Script on the campus IBM 360/370 mainframes. Things really began moving about 5 years ago when the source files up to that point were transferred first to a university and then a departmental DEC-20 machine, and T<sub>E</sub>X became available. Five or six years of steadily evolving class notes and draft output were generated, printed first on a Benson Varian 9211 printer and then on one of the earliest Canon/Imagen laser printers (which is still running magnificently today). If we began again today, of course, we would probably carry out all of this manuscript preparation using a PC-based T<sub>E</sub>X on our Macintoshes, obtaining draft output from a LaserWriter or something similar.

The final book, *Lasers*, is being published by University Science Books, headed by Bruce Armbruster, in Mill Valley, California, and will be distributed overseas by Oxford University Press. The book design was done by Robert Ishi of Oakland, California, and the copy editing by Aidan Kelly of Alameda, California. The art work was drawn by John Choi, draftsman in the Stanford Chemistry Department. T<sub>E</sub>X macros to implement Bob Ishi's design, along with the massaging of the source file into final shape, were largely done by

Stanford T<sub>E</sub>Xpert Laura Poplin, with assistance from Arthur Ogawa and Eric Berg. Final pages were typeset on the Stanford Computer Science Department's Autologic APS- $\mu$ 5 by Eric Berg, with counsel from Dave Fuchs. The art work was shot and pasted onto the typeset pages, and the book printed and bound, by the Maple-Vail Book Manufacturing Group. Publication coordination of all these people was accomplished, despite the confusions of a new and unfamiliar approach, by Bernie Scheier of Miller/Scheier Associates, Palo Alto.

Even when I began to use T<sub>E</sub>X, in its early days, I knew enough to delineate the main structural elements of the book using a simple limited set of macros (“\chapter”, “\section”, “\subsection”, “\sectionend”, “\figure”, “\problemsbegin”, “\problem”, “\problemsend”, and so forth), putting as little native T<sub>E</sub>X code in the source file as possible. I wrote simple definitions of these macros myself for producing draft output, assuming that a hired T<sub>E</sub>Xpert would modify and extend these macros when the final book design was developed. Any special symbols used in the laser field, such as the quantum-mechanical symbol “h-bar”, along with any mathematical symbols for which I thought I might later want to change notation, were also represented by control sequences.

I might of course have looked for some standard T<sub>E</sub>X macro package to employ in doing this. Few T<sub>E</sub>X macro packages existed at the time I began, however. In addition, my experience with macro packages, in several different situations, has always been that any macro package powerful enough to provide sophisticated capabilities seems inevitably to acquire a complicated syntax. Once the syntax of the macro package becomes sufficiently complicated, it becomes easier to learn the basic language and then name and define your own macros—stealing as many clever wrinkles from other packages as you can—rather than trying to learn the existing package syntax and make it do what you want.

One of the primary concepts in traditional publishing is of course the idea that the production of a book begins with an author's finished manuscript. Publishers assume there's going to be one. Copy editors assume they will have a stack of typewritten pages, on which they will write mysterious editing marks in red, which will eventually be read by the printer and must never be erased, along with editorial corrections in blue, and questions to the author in green, which must be answered in the margins in soft black pencil, and the circled question marks crossed out in purple, and so. This master copy

of the manuscript becomes a primary vehicle of communication between all those involved in the production process.

The idea that there is no paper manuscript, only a mysterious (and changeable) source file in a computer somewhere, is immediately upsetting to people in publishing, at least to the traditionalists. In my initial dealings with many of the production people I had the strong impression that what they really wanted most was to get the manuscript out of the g-d d-m computer and into their hands, so that they could start working on it in the way that God intended. (The copy editor did eventually end up sitting at one of our terminals, doing much of his editing of the source file on line.)

Traditional publishers—at least the ones I've encountered—also have a very strong reluctance to begin production work on a book until the author has turned in an absolutely complete, finished version of the manuscript and the associated art work. One can understand this viewpoint. It is probably based in part on well-founded experience that it always takes several times longer than the author promises for a manuscript to go from “virtually complete” to “really complete”. Authors are probably also notorious for telling publishers that the first half of their manuscript is absolutely finished and the publisher can start production work on it, while the author cleans up just a few unfinished bits in the second half. The author then comes back later on with expensive last-minute changes in the first half of the manuscript after it is halfway through the production process, thereby throwing everything into confusion.

Setting the production process in motion also has cash flow implications for the publisher—he must start paying the various subcontractors, with sales revenue still far in the future. Production people can also obviously work more efficiently, and follow schedules more effectively, if they have a complete and finished block of work handed to them all at once. Nonetheless, as I'll argue below, making plans as early as possible at least for how the production is to be accomplished is one step that can improve the efficiency and speed of the whole process.

To allow for maximum future flexibility, I had also adopted an approach in which every illustration in the book was given a symbolic name—e.g., “\pwrflow” for a diagram of the power flow in a laser system. Each figure was located in the source file by a \figure macro with syntax like

```
\figure{\pwrflow}{18}{20}{sidecap}
      {Power flow in a laser system.}
```

The arguments include the figure's symbolic name, its width and height, information as to whether it is a side-caption or bottom-caption figure, and the text for the figure caption. Encountering this macro in the source file bumps the current figure number up by one; assigns that number to the control sequence `\pwrflow`; reserves suitable space in the  $\TeX$  output (perhaps as a `\topinsert`) for the figure; and inserts a suitably formatted caption. I also wrote the symbolic name ("`\pwrflow`") in the upper right corner of the original art work for that figure. The figure was then referred to in the text by its symbolic name, e.g., "The flow of power between different elements in a laser is shown in Figure `\pwrflow`.". With this approach I could obviously insert a new figure anywhere in the book, or move a whole section of the book with its imbedded figures to some other place in the manuscript, at any time before the final  $\TeX$  run of the source file. So long as I kept all the symbolic names unique, the figure captions, references and renumbering would all be taken care of.

The production people were unable to cope, however, with the idea of illustration copy that was identified by names instead of numbers; and the copy editor and everyone else expressed great unhappiness at a manuscript which referred to "Figure `\pwrflow`" rather than "Figure 8.11". Each piece of art work for a book is supposed to have a figure number, written in red and circled, in the upper right-hand corner. The production coordinator wants to count these, and make sure they're all there; the copy editor wants to see that the figure number written on the art work matches what is in the text. It was easiest eventually to go back, after the final manuscript was frozen, and hand-convert each symbolic name into the corresponding figure number, both on the art work copy and in the source file.

I did not want to write the final macro definitions and  $\TeX$  coding myself to convert the book designer's specifications into typeset output pages, nor did I want to massage the final page markup and run the phototypesetter myself. Hence it was necessary to find local free-lance  $\TeX$ perts with publishing skills to carry out these tasks. We were fortunate eventually in finding skilled help in the Stanford community to do this work. People with advanced ability in  $\TeX$ , and especially people who combine high-level  $\TeX$  skills with a genuine eye for quality book layout, are not yet thick on the ground, however.

Minor complications also arose in getting the final typeset pages for the book prepared on the

typesetter belonging to the Computer Facilities group in the Stanford Computer Science Department. There were the usual purely technical problems of fonts and communication between machines, which were eventually overcome. Beyond this, however, there were administrative difficulties of the sort that easily arise when commercial and academic worlds meet. Stanford's Computer Science Department views its typesetter as a typical academic facility. It's there and available to faculty, with some staff help available in using it, and its operating costs must be paid for more or less on a cost-of-use basis; but it is not a commercial enterprise. The publication coordinator, concerned with schedules for getting the book printed, bound, and marketed, wanted to be able to make firm commitments to deliver the typeset pages to the printer on a scheduled date. Hence he wished to have a definite business commitment from the typesetting people that the work could be done by a certain date, for a certain fixed price. The Computer Science people, reasonably enough, were not in a position to give that kind of firm guarantee. Things did, of course, all work out in the end. Incidentally, after a small amount of looking we did not find any commercial typesetting firms, at least in the San Francisco Bay Area, that offered commercial-grade book-quality typesetting service starting from  $\TeX$  source files.

What are some of the recommendations I would make, or steps I would do differently, if I had to do the whole job over again?

1) First of all, I would bring the copy editor and the author together as early as possible in the manuscript preparation, after only a few chapters have been finished. If the author likes to write "A is soft, while B is hard", and the copy editor is to going to insist that this be written "A is soft, whereas B is hard", the author might as well know this as early as possible in the source file preparation. Questions of style, such as how figures and equations are to be numbered and referred to, should also be settled early on, thus avoiding tedious changes in the source file later on.

2) I would also put the book designer and the  $\TeX$  people in direct contact as early as possible in the production cycle, before either has begun their work. The book designer has primary responsibility for the aesthetic appearance of the book. Not all designers are yet familiar with the capabilities, and limitations, of  $\TeX$ ; some of them may still regard  $\TeX$  in the same light as more primitive word-processing or text-formatting systems — OK for in-house technical reports, but not really capable of serious bookmaking. Having the book designer

specify the book format in isolation, then giving this format to the  $\text{\TeX}$ pert and asking that it be implemented, does not make the most effective use of the talents on either side.

3) Obtaining aesthetically pleasing page breaks, which also produce the most effective relationship between subheadings, illustrations, tables, and text references to these elements, seems to be the most difficult and tedious part of the whole book production process in  $\text{\TeX}$ . A  $\text{\TeX}$ pert, a publishing person, and potentially the author must work together on many successive drafts, shifting the positions of figure macros, tables and other inserts, juggling headings, and perhaps cutting or expanding the text itself, until  $\text{\TeX}$  yields a layout for each pair of facing pages that is both aesthetically satisfactory and functionally effective.

Since any change at an earlier point in the source file can propagate forward and change subsequent page breaks, a tool for freezing the page breaks as one slowly works forward through the book is essential.  $\text{\TeX}$  can in fact do quite unexpected things in its attempt to minimize penalties on each page. Its behavior can be reminiscent of a *Scientific American* article from some years ago about work scheduling. The article described a situation in which a group of workers successively picked up tasks of different lengths from the top of a pile of work to be done, each worker coming back for a new task as soon as his or her previous task was completed. Under certain circumstances, increasing the number of workers in the team could actually increase the time required to finish the same pile of work. Similarly, shortening a paragraph by a few words on an early page in a chapter, in order to pull back an orphan line that had spilled onto the following page, would sometimes trigger  $\text{\TeX}$  to make readjustments in the page breaks on later pages that actually made the chapter a full page longer.

4) I did not attempt during the writing of this book to mark potential index entries or index terms in the source file, though I thought of doing this by using some kind of `"\indexentry{...}"` macro which could then write a preliminary index file for computer sorting and hand refinement. By the time the book went into production it seemed simpler and quicker to generate the index "by hand", rather than attempting to define such a macro and put the entries into the source file. If I did the whole thing again from scratch, I would probably include some sort of macro tool for marking the starting and stopping points for index terms in the source file. I would want this macro to note the starting

and stopping points for the index term by a small marginal note in draft printouts, and to write the index term and the relevant page numbers to an index file in the final  $\text{\TeX}$  run. This index file could then of course be computer-sorted to become the basis for quick preparation of the final index.

All in all, I would certainly produce any future book of this type using  $\text{\TeX}$ , as compared to any other way. Important benefits for author, publisher, and readers that came from preparing this book in  $\text{\TeX}$  included:

1) The ability to generate (easily) many draft chapters as typeset class notes which were available years before the complete book was finished, so that I could obtain student feedback and error checking on them.

2) The ability to make major revisions and rearrangements of the material all through the writing stages, and the ability to make smaller changes, including adding late references and results, revising paragraphs, and correcting errors, up to a very late stage in the final production process.

3) Freedom for the author from the terrible drudgery, after the manuscript has been finished, of having to again reread and correct in minute detail first the galley proofs and then the page proofs. Having been through that labor twice on earlier books, I'd never want to do it again.

4) Speed of publication from finished manuscript to printed book. With better advance organization we might have done things even faster, but the process still took considerably less than the traditional 9-month gestation period from completed manuscript to printed books. It is particularly helpful to be able to obtain instant draft output from a local laser printer at every stage throughout the production process, rather than having to send marked-up manuscript, or galley proofs, off to a distant typesetter and not receive galleys, or corrected page proofs, until weeks or months later. This gives an immediacy to the production process which keeps attention focused on the project.

5) Of great importance, economy. At a time when technical books are typically priced at 10¢ to 12¢ per page, this relatively specialized book with 1300 pages of technical mathematics will carry a retail list price under \$60.

Finally, last but far from least, preparing this book in  $\text{\TeX}$  gave me the opportunity to gain some appreciation of the remarkable intellectual achievement represented by the  $\text{\TeX}$  language, and seemingly by everything else that Donald Knuth does.