LaTeX for Scientists and Engineers

David J. Buerger

Preface

LaTeX is a text formatting system that lets you typeset documents on personal or mainframe computers without the trouble of mastering complex typesetting machines. LaTeX[1] is well-suited for creating technical documents, books, reports, dissertations, manuals, and articles. It lets you create complex mathematical symbols and formulas, foreign language characters, floating tables and figures, table of contents, indexes, bibliographies, cross-references, and footnotes.

Documents created with LaTeX can be printed on a variety of laser and dot-matrix printers. Documents also can be printed on publishing quality phototypesetters. This capability gives you the opportunity for total control over the production of your document.

The LaTeX document preparation system was developed by Leslie Lamport [6]. LaTeX is a collection of special commands, or "macros," based on Donald E. Knuth's TeX program [4]. The system's typesetting therefore is produced by TeX. Unlike TeX, however, LaTeX lets you concentrate on a document's structure and content, rather than on its detailed formatting.

Some instruction books make LaTeX seem as hard to use as TeX. This book was written with the assumption that most users can productively use LaTeX without an exhaustive knowledge of every technical detail. Although LaTeX *for Scientists and Engineers* does not describe every LaTeX nook and cranny, the book's organization lets you pick and choose information germane to the task at hand, without burdening you with unnecessary details. LaTeX *for Scientists and Engineers* was written to provide a fast and easy way to learn how to produce technical documents with LaTeX.

This book assumes that you have no knowledge of TeX or LaTeX. It outlines the basic steps to create typeset documents with LaTeX. Each chapter presents concise discussions and examples. Exercises at the end

[1] Usually pronounced "*lah*-tekh," although some prefer "*lay*-tekh," or "*lay*-tex."

of selected chapters help you gain practical experience. Answers to all exercises are at the end of the book. An assortment of sample documents with illustrations of how they are created are in Appendix B. I assume that readers know how to use a word processor or text editor. All discussions and examples are based on the current version (2.09) of LaTeX.

I wish to thank my friends and associates who generously offered comments and encouragement towards the completion of this book. These include David H. Bailey, Nelson H. F. Beebe, James R. Celoni, S.J., Willis Dair, Ronald Danielson, Ruth Davis, Carl Fussell, Tom Golden, Leslie Lamport, Diane Levy, Maggie McClelland, Steven Nahmias, Oren Patashnik, David Raths, John Reuling, Dennis C. Smolarski, S.J., Paul Soukup, S.J., Randall Westgren, and Michael Zomlefer.

Appreciation also is due to Santa Clara University (Santa Clara, California) for providing computer facilities on which this book's numerous drafts were processed.

— D. J. B.

Cupertino, California
November 1, 1989

Contents

Preface xv

1 Technical Word Processing and Typography 1
- 1.1 Markup Systems and Technical Publishing 1
- 1.2 LaTeX: A Descriptive Markup System 3
- 1.3 The Language of Typography 4
- 1.4 What You Need to Use LaTeX 4
- 1.5 Where to Get More Help 6

2 Features Overview 7
- 2.1 Document Structure 7
- 2.2 Typing Text 7
- 2.3 Type Styles and Sizes 8
- 2.4 Special Characters 8
- 2.5 Formatting Environments 8
- 2.6 Math Symbols and Equations 9
- 2.7 Other Document Production Tools 9

3 Beginning Concepts 11
- 3.1 Input Files 11
- 3.2 Document Style 11
- 3.3 How to Run LaTeX 13
- 3.4 Special Characters 14
- 3.5 Spacing 17
- 3.6 In-Text Design Notes 20
- 3.7 Summary 20
- Problems 21

Formatting Environments **23**
4.1 The "Center" Environment 23
4.2 The "Flushleft" and "Flushright" Environments 24
4.3 List-Making Environments 25
4.3.1 Itemized Lists 25
4.3.2 Enumerated Lists 25
4.3.3 Description Lists 27
4.4 "Quote" and "Quotation" Environments 29
4.5 The "Verse" Environment 31
4.6 The "Verbatim" Environment 32
Problems 34

5 Changing the Appearance of Type **37**
5.1 Changing Typefaces 37
5.2 Changing Type Sizes 39
5.3 Mixing Typefaces and Sizes 40
5.4 Creating Special Symbols 40
Problems 42

6 Simple Math Typesetting **43**
6.1 Types of Math Environments 43
6.2 Simple Expressions and Equations 44
6.3 Math Symbols 46
Problems 47

7 Complex Math Typesetting **49**
7.1 Arrays 49
7.2 Multiline Equations: The "Eqnarray" Environment 51
7.3 Special Effects 52
Problems 53

8 Tables and Figures **55**
8.1 The "Tabbing" Environment 55
8.2 The "Tabular" Environment 58
8.3 The "Table" and "Figure" Environments 60
Problems 62

9 Footnotes and Cross-References **63**
9.1 Footnotes 63
9.2 Cross-References 64

10 Organizing a Document **67**
10.1 Section Commands 67
10.2 Title and Title Page 68
10.3 Table of Contents 69
10.3.1 Preface or Foreword 69
10.4 Preamble 70
10.4.1 Document Style Options 71
10.4.2 One-Sided vs. Two-Sided Output 71
10.4.3 Headers 72
10.4.4 Page Layout Options 72
10.5 Appendices 73
10.6 Bibliography, Glossary, and Index 73
10.7 Importing Files 73
Problems 78

11 Creating a Bibliography **79**
11.1 Creating a Bibliographic Database 79
11.2 Entry Types 81
11.3 Fields 82
11.4 Automatic Journal Citations 84
11.5 How to Cite References in the Text 85
11.6 Using BibTeX to Create a Bibliography 86
11.7 Making a Bibliography Without BibTeX 87
Problems 89

12 Creating a Glossary and Index **91**
12.1 Glossary 91
12.2 Index 91
12.2.1 Marking Words to Be Indexed 92
12.2.2 Using MakeIndex to Create an Index 94
Problems 96

13 Two-Column Documents **97**
13.1 The "Twocolumn" Option 97
13.2 The "Twocolumn" Proceedings Option 97
13.3 Switching to Two-Column Mode in Text Body 98
13.4 Parallel Text With the "Minipage" Environment 98
13.5 Creating Newsletters With LaTeX 99
Problems 101

14 Special Operations **103**
14.1 "Newcommand" and "Renewcommand" 103
14.2 "Newenvironment" and "Newtheorem" 105
14.3 Numbering Counters 107
14.4 Combining Graphics and Text 109
Problems 113

15 Deciphering Error Messages **115**
15.1 Common Errors 115
15.2 Interpreting Error Messages 116
15.3 LaTeX Error Messages 117
15.4 LaTeX and TeX Warnings 119
15.5 More Help on Pinpointing Errors 120

Appendices

A Answers to Chapter Exercises **123**

B Sample Input Files **139**

C Mathematical Symbols **151**

D Custom Font Sizes **157**
D.1 The "Newfont" Command 157
D.2 Font Samples 158

Glossary **179**

Bibliography **191**

Index **193**

List of Figures

10.1 LATEX Single-Column Page Layout Commands. 74
10.2 LATEX Two-Column Page Layout Commands. 75

List of Tables

2.1 U. S. Paper Currency 10

3.1 Accent Commands. 16
3.2 Punctuation and Foreign Language Symbols. 16

5.1 LaTeX Type Size Options 39
5.2 Font Sizes Available for LaTeX Typefaces (Font Classes: P = Preloaded, D = Loaded on Demand, X = Unavailable.) . 41

8.1 Tabbing Commands 57
8.2 Table/Figure Positioning Commands 60
8.3 Low-Speed Asynchronous Full-Duplex Modem Interface Leads 62

14.1 LaTeX Sectional Units and Environments With Corresponding Counters 108

B.1 Cray-2 Performance Figures 148

C.1 Greek Letters 151
C.2 Binary Operation Symbols 152
C.3 Relation Symbols 152
C.4 "Log-like" (Always in roman) Function Names 153
C.5 Arrow Symbols 153
C.6 Miscellaneous Symbols 154
C.7 Variable-Sized Symbols 154
C.8 Delimiters 155
C.9 Math Mode Accents 155

Chapter

1

Technical Word Processing and Typography

Professionals in mathematics, the sciences, engineering, and other related fields are just beginning to exploit computer software to produce technical publications. It seems ironic that despite the availability of a multitude of high-powered word processors and other text editing tools, the production of complex technical documents is often a frustrating experience.

The American Mathematical Society, for example, illustrates no less than 19 separate steps to turn a mathematics paper into a publication [12, pp. 5–6]. Clearly, steps such as dealing with publishers and referees will never be computerized. Phases like document creation and proofreading multiple galley drafts, however, can be substantially shortened or eliminated by using the right computer software.

Most technical writers no longer handwrite or typewrite drafts; the use of word processors or text editors on personal computers and large host computers is the practical norm. Yet popular editing software usually does little to help produce complex mathematical formulas, floating tables and figures, or other explanatory tools of the trade. If one needs to produce camera-ready documents, another time-draining issue often rears its head: that of mastering esoteric typesetting conventions. Some who buy and use "desktop publishing" software occasionally find that they now have two full-time jobs.

1.1 Markup Systems and Technical Publishing

A primary goal of technical writing is to communicate complex, logical ideas. Writers employ a host of techniques to accomplish this goal. On an elementary level, rules of punctuation are followed to clarify meaning: words are separated by spaces, periods mark the end of sentences, quotation marks or indentations offset quotations, extra carriage returns separate paragraphs, and numbers are typed to designate outline sections. These methods of marking up text to make it understandable are required whether one uses a word processor, a typewriter, or writes drafts by hand.

More advanced presentational markup techniques such as line centering, boldfacing or italicizing type, and interjecting large typefaces are typical capabilities of sophisticated word processors and desktop publishing software. Punctuation or presentational steps usually are applied manually. At other times, sections of text are tagged for special treatment. These visual systems are highly touted because they let writers see what they are going to get on paper.

Technical writers, however, may find this latter feature to be a liability. Visual markup systems focus attention on appearance, not content. The process of making the document look just right can take more time than producing the document's content. Furthermore, most scientists, mathematicians, and engineers are not trained document designers. Such lack of design expertise may lead to an ineffective presentation of their ideas.

One area in which bad document design can be especially harmful to the clarity of a technical publication is mathematics. Mathematical typesetting demands much more sophisticated design criteria than simple text. It has a larger alphabet of basic symbols, each of which has unique design characteristics different from standard typefaces. Complex spacing criteria apply to formulas, expressions, theorems, proofs, matrices, and arrays. While most visual markup software doesn't provide a way to create complex mathematical symbols and expressions, those few that do assume that users have complete expert knowledge about how they should be formatted.

Another drawback to visual markup software is lack of portability. Such software frequently runs on a limited number of hardware platforms. If a program works only on an Apple Macintosh, for example, it becomes difficult to co-publish a camera-ready paper or book with another author who uses different software on a Sun Microsystems workstation. Organisations that use diverse hardware and software platforms will find internal cooperative publishing difficult.

In light of these problems, many within the mathematical, scientific, and engineering professions have turned to procedural and descriptive markup systems to produce complex technical publications—systems usually not visual in nature. The display monitor shows only text, plus typewritten commands that control how the document will look after it is processed through a formatting program, and then printed on a laser printer or phototypesetter. Some procedural and descriptive systems provide post-processing software that allows one to preview the results as they will appear on paper. A few systems even let you preview the results while the system is processing the file.

Procedural markup programs such as TEX, and UNIX-based *nroff* and *troff*, are complicated to use in their native state. They require the user to type procedure command codes for virtually all aspects of document production. This includes steps as explicit as defining how much space will appear between lines and paragraphs, how far paragraphs are to be indented, and how individual characters are to be used in a document. Such systems' commands can be as complex as a typesetting system, and thus require considerable effort to master.

Descriptive markup systems also require users to type formatting commands into the body of a document's computer file. Unlike procedural markup systems, however, descriptive systems focus the user's attention on document content rather than appearance by having the user type commands that describe what is being typed, instead of how the text should appear in printed form. Technical writers thereby describe sections of text by what they are: Examples include chapter, section, sub-section, table, displayed math formula, and quotation. Printed formats are based on predefined document styles, each created by a typographic designer. Descriptive markup systems ideally free writers from document design concerns, and let them concentrate on content.

In contrast to visual markup software, descriptive markup systems automatically apply mathematical style conventions. Since descriptive markup documents usually are created from straight text files, they can be easily transferred between different computer hardware platforms, and through electronic mail systems.

For more discussion on the philosophy of markup systems, see [3,5], [6, pp. 5–8], and [8].

1.2 LaTEX: A Descriptive Markup System

LATEX is a document preparation system developed by Leslie Lamport [6]. LATEX also is a unique type of descriptive markup system, being a collection of special commands—"macros"—based on Donald E. Knuth's TEX program [4]. The system's typesetting therefore is produced by TEX. LATEX has been implemented on microcomputers, engineering workstations, minicomputers and mainframes using a variety of operating systems. It works virtually the same on all systems. LATEX enjoys worldwide usage among scientists, mathematicians, engineers, and other scholars and researchers who require its features. This allows great flexibility in producing and sharing typeset documents.

1.3 The Language of Typography

Specialists in every discipline have unique vocabularies that give precision to technical discussions. Not surprisingly, the field of typography has its own jargon. The language of typography becomes pervasive as one moves from typewriters and character-based print towards true typesetting.

Typography is the art of creating styles and arrangements of typeset material. Typographic terms such as *typeface*, *font*, *document style*, *justified text*, *leading*, *kerning*, and *serifs* deal with the nature and design of characters, how characters are spaced, how different types of margins affect a document's layout, and a host of other factors.

A typeface, for example, is a particular style of type such as boldface or italic. A font is a family or set of characters in the same typeface and size. A document style predetermines a complete document's layout.

The glossary at the end of this book defines many terms that you will become familiar with as you learn how to use LaTeX. The glossary also includes definitions for the most common LaTeX commands. The index will show you where to find step-by-step instructions on these commands and features.

1.4 What You Need to Use LaTeX

Detailed instructions on hardware and software required to run LaTeX on your computer are found in a book called the *Local Guide*. The *Local Guide* always comes with commercial versions of LaTeX. Institutions that support this software often write and distribute expanded versions tailored to their environment and user needs. The *Local Guide* covers topics such as: How to install LaTeX on your organization's hardware; how to process an input file (i.e. a text file that contains a document's text and formatting commands) through LaTeX; how to use local printer drivers, document previewers, and special style files. The *Local Guide* also contains a list of all fonts available on your system. Depending on how much TeX and LaTeX are used in your institution, the *Local Guide* can vary in size between a short pamphlet and a small book.

In general, input files can be composed on any computer, so long as the resulting text file can be transferred to a computer that has the required software to process and print it. Because LaTeX is a set of macros based on TEX, it cannot run unless you also have TEX installed on the same computer.

LATEX and TEX are available on many large host systems. If you are uncertain whether they are available at your institution, contact the computer center for more information. Mainframes and engineering workstations offer four distinct benefits that people who use LATEX on microcomputers may not enjoy:

- Host software often is free or available for a nominal charge. Once installed, every person with a host account can use it.
- Mainframes usually process LATEX documents much faster than microcomputers. If you are producing a long book or dissertation, the speed advantage could be critical in processing multiple drafts.
- Any microcomputer user with an asynchronous dialup link or network hookup to a mainframe can share LATEX files with users who otherwise have incompatible hardware.
- Most large hosts and workstation networks have expensive, high-speed laser printers available for general use.

LATEX and TEX implementations also are available on several different microcomputers. A hard disk is required because the full software distribution can require eight or more megabytes of disk storage space. Most of this space is required to store bit-mapped font files for previewing and printing documents. In addition, you should allow for a few extra megabytes temporarily required to process large files. LATEX runs on popular microcomputers, including:

IBM and Compatibles. IBM XTs, ATs, PS/2s, and compatible computers require 512 Kbytes of random-access memory (RAM), but work better with 640 Kbytes. Memory-resident programs should be disabled when using IBM-compatible TEX and LATEX implementations. An AT or 80386-class computer is preferable to an XT because XTs process LATEX pages very slowly. An AT processes LATEX files at a speed of roughly one page every 10 seconds compared to approximately 30 seconds per page on an XT. If you want to use software to preview the final form of your text on a monitor before printing it, you will need either a "Hercules-compatible," EGA, or VGA display card; CGA cards will not work with previewers. Commercial and public domain PC printer drivers are available for Epson and Toshiba-compatible dot-matrix printers, and PostScript, Hewlett-Packard, Cordata, QMS Lasergrafix, and Imagen laser printers.

Apple Macintosh. Apple Macintosh Plus computers, and newer models, require at least one megabyte of RAM plus a hard disk. You may prefer to work with a supplementary full-screen display for easier-to-read previewing. Output drivers are available for PostScript laser printers.

Commodore Amiga. Amiga software requires 512 Kbytes of RAM for TEX and one megabyte of RAM for LATEX, plus a hard disk. Output drivers are available for the NEC P series, Epson FX and LQ series, ImageWriter II, PostScript, QMS Kiss, and SmartWriter printers.

1.5 Where to Get More Help

In addition to LATEX *for Scientists and Engineers*, you should obtain and use a copy of your institution's *Local Guide*. It also is wise to cultivate relationships with people who have LATEX expertise. Such persons can provide valuable suggestions when you are stumped by a thorny formatting problem. Before consulting these people, however, be sure to first check these two books.

If you don't have access to a local expert, you should consider joining the TEX User's Group (TUG). You can contact TUG at 653 N. Main St., P.O. Box 9506, Providence, Rhode Island, 02940-9506; telephone (401)751-7760. Internet electronic mail can be sent to `tug@math.ams.com`. TUG conducts TEX and LATEX training seminars, and publishes a quarterly journal called *TUGboat*, edited by Barbara Beeton.

Internet electronic mail users should consider subscribing to TUG's informative news digest called TEXHaX, published in cooperation with the UNIX TEX distribution service at the University of Washington, and edited by Pierre MacKay and Tiina Modisett. TEXHaX is a question-and-answer forum whose participants include TEX and LATEX aficionados throughout the world. Subscriptions should be sent to this electronic mail address: `texhax-request@cs.washington.edu`; submissions should be sent to `texhax@cs.washington.edu`. This digest is also available on Usenet Netnews under the `/comp/text` newsgroup.

Chapter

2

Features Overview

LaTeX offers a rich assortment of tools to help writers clearly communicate ideas. This chapter gives a brief overview of some of LaTeX's features. The features introduced in this chapter are explained and illustrated throughout this book. Those of you who use text formatting systems such as *Scribe*, *nroff*, *troff*, *FinalWord*, *Sprint*, and *Perfect Writer* will find much familiar ground in LaTeX.

2.1 Document Structure

LaTeX is a document processing system. What you write is formatted according to a pre-defined document style. LaTeX has four standard document styles: `article`, `book`, `report`, and `letter`. The style you choose controls how TeX performs the actual typesetting. By changing the document style from one selection to another, you can automatically transform the layout appearance of your entire document into a new format.

Documents can be divided into traditional units like parts and chapters. You also can use more detailed text subdivisions, like sections and subsections. Based on your use of these sectional tools, a table of contents can be automatically generated. Other commands are used to create the title page, appendices, bibliography, and index.

2.2 Typing Text

Text is entered as you usually do with a word processor or text editor. By following a few simple conventions, LaTeX will transform your text input file into a typeset document.

TeX's automatic typesetting attributes include accurate hyphenation, proportional spacing between words, kerned letter combinations, and ligatures. Kerning is a typesetting technique that adjusts the amount of space between two characters. This distance is based on the width of each character. Some character pairs look better when they are moved closer together; some combinations look better when they are spaced farther apart. Ligatures are letter combinations joined together as one unit—for example, ff, fi, fl, ffi, and ffl.

2.3 Type Styles and Sizes

Typeset documents normally do not have underlined text; instead these words are *emphasized* in italic type. In this example, the italics were created by typing `{\em emphasized}`. The `\em` switched type to emphasized mode; the curly braces surrounding the word were used as a grouping technique to restrict emphasis to text only within their boundaries.

Other options are described in Section 5.1 that allow you to use **boldface**, SMALL CAPS, sans serif, *slanted*, and `typewriter` typefaces. Section 5.2 shows how you can change type sizes from this, to this, to this, to this, to this, to this, to this, to this, to this, to this. Appendix D describes how to change LaTeX's default Computer Modern Roman font to other fonts that come with LaTeX distributions.

2.4 Special Characters

A host of multilingual symbols make it easy to create text in some non-English languages. For example, "Nous sommes prêts à partir pour l'Université" is created by typing

```
''Nous sommes pr\^{e}ts \'{a} partir pour l'Universit\'{e}''
```

These symbol commands are listed in Tables 3.1 and 3.2.

2.5 Formatting Environments

Formatting environments allow you to control the appearance of text. An environment is a section of text upon which a specified formatting feature is applied. The typeface changes illustrated above are one example of environment changes. When curly braces are typed *around* a word with the "emphasis" command (e.g. `{\em emphasized}`), that command is restricted to text *within* the braces; this is analogous to scoping attributes used in computer programming. Another way to create an environment