

# Preface

More than a decade has passed since the publication of the first edition of *The L<sup>A</sup>T<sub>E</sub>X Graphics Companion*, and there have been many changes and new developments since 1996.

The second edition has seen a major change in the authorship: Frank, Michel and Sebastian have been joined by Denis and Herbert as authors, enriching the book with their knowledge and experience in individual subject areas.

As in the first edition, this book describes techniques and tricks of extended L<sup>A</sup>T<sub>E</sub>X typesetting in the area of graphics and fonts. We examine how to draw pictures with L<sup>A</sup>T<sub>E</sub>X and how to incorporate graphics files into a L<sup>A</sup>T<sub>E</sub>X document. We explain how to program pictures using METAFONT and METAPOST, as well as how to achieve special effects with small fragments of embedded PostScript. We look in detail at a whole range of tools for building graphics in T<sub>E</sub>X itself.

T<sub>E</sub>X is the world's première markup-based typesetting system, and PostScript (on which PDF is based) is the leading language for describing the printed page. We describe how they can produce even more beautiful results when they work together. T<sub>E</sub>X's mathematical capability, its paragraph building, its hyphenation, and its programmable extensibility can cooperate with the graphical flexibility and font-handling capabilities of PostScript and PDF to provide a rich partnership for both author and typesetter.

To be able to do justice to the graphics packages that have been further developed since the first edition, we decided to omit a description of PostScript and PDF tools, and of font technologies, from the printed version of this book. This material, which was covered in Chapters 10 and 11 of the first edition, has been substantially expanded and is now freely available (see <http://xml.cern.ch/lgc2>). It covers DVI-to-PostScript drivers, the free program `ghostscript` to view PostScript and PDF files, tools for manipulating PostScript and PDF files, and suggestions on how to combine the latest font technologies (PostScript Type 1 and OpenType) with L<sup>A</sup>T<sub>E</sub>X.

This volume is not a complete consumer guide to packages. In trying to teach by example, we present hundreds of self-contained code samples of the most useful types of solutions, based on proven and well-known implementations. But, given the space available, we cannot provide a full manual for every package. Our aim is simply to show how easy it is to use a given package and to indicate whether it seems to do what is required—not to dwell on the precise details of syntax or options. Nevertheless, we have described in more detail a few selected tools that we consider especially important.

We assume you know some  $\LaTeX$ ; you cannot read this book by itself if you have never used  $\TeX$  before. We recommend that you start with  *$\LaTeX$ : A Document Preparation System, Second Edition* [78], or the *Guide to  $\LaTeX$ , Fourth Edition* [76], and continue with *The  $\LaTeX$  Companion, Second Edition* [83], to explore some of the many (non-graphical) packages available.

## Why $\LaTeX$ , and why PostScript?

This book is about  $\LaTeX$ , graphics, PostScript, and its child PDF. We believe that the structured approach of a system like  $\LaTeX$  is the best way to use  $\TeX$ , and  $\LaTeX$  is by far the most widely used  $\TeX$  format. This means that it attracts contributors who develop new packages, and thus some of what we describe works only in  $\LaTeX$ . We apologize in advance for our  $\LaTeX$  bias to those who appreciate the elegance of the original plain  $\TeX$  format and its derivatives, and we promise them that most of the packages will work well with any  $\TeX$  dialect: the delights of systems such as `METAPOST`, `PSTricks`, `Xy-pic`, and `MusiX $\TeX$`  are open to all.

We also want to explain why we talk about PostScript so much. This language has been well established for almost two decades as an extremely flexible page-description language, and it remains the tool of choice for professional typesetters. Among the features that make it so attractive are these:

- The quantity, quality, and flexibility of Type 1 fonts
- The device-independence and portability of files
- The quality of graphics and the quantity of drawing packages that generate it
- The facilities for manipulating text
- The mature color-printing technology
- The encapsulation conventions that make it easy to embed PostScript graphics
- The availability of screen-based implementations (e.g., `ghostscript/ghostview`)

PostScript has spawned an enterprising child, the PDF (*Portable Document Format*) language, used by Adobe Acrobat and now well established as an exchange format for documents on the Web. Designed for screen display with hypertext features, PDF offers a new degree of portability and efficiency. Although not the main subject of this book, we nevertheless mention that  $\LaTeX$  can also produce “rich” PDF documents, and versions of  $\TeX$  (e.g., `pdf $\LaTeX$` ) that produce PDF directly are available.

Again, we apologize to those of you who are disappointed not to read about L<sup>A</sup>T<sub>E</sub>X's association with Mac's QuickDraw, or the Windows GDI, HPGL, PCL, etc., but with so many packages available, we had to make a choice.

Please note that the absence of a given package or tool in this book in no way implies that we consider it less useful or of inferior quality. We do think, though, that we have included a representative set of tools and packages, and we sincerely hope that you will find here one or more subjects to entertain you.

## How this book is arranged

This book is subdivided in two basic ways: by application area and by technique. We suggest that all readers look at Chapter 1 before going any further, because it introduces how we think about graphics and summarizes some techniques developed in later chapters. We also suggest that you read Chapter 2, which covers the L<sup>A</sup>T<sub>E</sub>X standard graphics package, since the tools for including graphics files will be needed often. Chapter 2 also covers `pict2e`, a package that reimplements L<sup>A</sup>T<sub>E</sub>X's `picture` environment using PostScript, and a further extension `curve2e`. Together these packages not only do away with most of the limitations inherent in the standard version of L<sup>A</sup>T<sub>E</sub>X's `picture`, but also offer new and powerful commands to draw arcs and curves with minimal effort.

*Basic information in Chapters 1 and 2*

We have tried to make it possible to read each of the other chapters separately; you may prefer to go straight to the chapters that cover your subject area or look at those that describe a particular tool. Two chapters each are dedicated to the generic systems METAPOST and PSTricks.

- 3 METAFONT and METAPOST: T<sub>E</sub>X's Mates** shows how to exploit the power of T<sub>E</sub>X's META languages (Knuth's METAFONT and its PostScript-based extension METAPOST). After introducing the basic functions, the basic METAPOST libraries are described, as well as available T<sub>E</sub>X interfaces and miscellaneous tools and utilities.
- 4 METAPOST Applications** introduces the METAPOST toolkit, and explains how to use METAPOST's unparalleled expressive power for describing many types of graphs, diagrams, and geometric constructs. Applications in the areas of science and engineering, 3-D representations, posters, etc. conclude the overview.
- 5 Harnessing PostScript Inside L<sup>A</sup>T<sub>E</sub>X: PSTricks** walks the reader through the various components of the PSTricks language, looking at such things as defining the coordinate system, lines and polygons, circles, ellipses and curves, arrows, labels, fill areas, and much more.
- 6 The Main PSTricks Packages** takes you even deeper into the world of PSTricks. Armed with the knowledge gained in Chapter 5, the reader will find here detailed descriptions of the most common PSTricks packages—in particular, `pst-plot` for plotting functions and data; `pst-node` for mastering nodes and their connections; `pst-tree` for creating tree diagrams; `pst-fill` for filling and tiling areas; `pst-3d` for creating 3-D effects, such as shadows and tilting; and `pst-3dplot` for handling 3-D functions and data sets. The chapter ends with a summary of PSTricks commands and keywords.

The next four chapters discuss problems in special application areas and survey more packages:

**7 The  $\Xy$ -pic Package** introduces a package that goes to great lengths to define a notation for many kinds of mathematics diagrams and implements it in a generic and portable way.

**8 Applications in Science, Technology, and Medicine** looks at chemical formulae and bonds, applications in bioinformatics, Feynman diagrams, timing diagrams, and electronic and optics circuits.

**9 Preparing Music Scores** first describes the principles of the powerful  $\text{Musix}\TeX$  package. Then several preprocessors providing a more convenient interface are introduced: `abc` for folk tunes, `PMX` for entering polyphonic music, and `M-Tx` (an offspring of `PMX`) for dealing with multi-voice lyrics in scores. We also take a short look at `LilyPond`, a modern music typesetter written in C++, and say a few words about  $\text{TeXmuse}$ .

**10 Playing Games** is for those who use  $\text{L}^{\text{A}}\text{TeX}$  for play as well as for work. It shows you how to describe chess games and typeset chess boards (the usual and oriental variants). This chapter also describes how to handle Go, backgammon, and card games. We conclude with crosswords in various forms and Sudokus, including how to typeset, solve, and generate them.

Our last chapter addresses an area of general interest: color, and some of its common uses in  $\text{L}^{\text{A}}\text{TeX}$ .

**11 The World of Color** starts with a short general introduction to color. Next comes an overview of the `xcolor` package and the `colortbl` package, that is based on `xcolor`. The final part discusses the `beamer` class for producing color slides with  $\text{L}^{\text{A}}\text{TeX}$ .

Appendix A describes ways to generate PDF from  $\text{L}^{\text{A}}\text{TeX}$ . Appendix B introduces CTAN and explains how to download the  $\text{L}^{\text{A}}\text{TeX}$  packages described in this book.

As mentioned earlier, material about PostScript and PDF tools, as well as information about how to use PostScript and OpenType fonts with  $\text{L}^{\text{A}}\text{TeX}$ , is available as supplementary material (see <http://xml.cern.ch/lgc2>), which covers the following subjects:

**PostScript Fonts and Beyond** describes the ins and outs of using PostScript fonts with  $\text{L}^{\text{A}}\text{TeX}$ . It also looks at the latest developments on how to integrate OpenType fonts by creating  $\text{TeX}$ -specific auxiliary files ( $\text{TeX}$  metrics, virtual fonts, etc.) or by reading the font's characteristics directly in the OpenType source.

**PostScript and PDF Tools** starts with a short introduction to the PostScript, PDF, and SVG languages. It then describes some freely available programs, in particular `dvips` and `pdflatex` to generate PostScript and PDF, `ghostscript` and `ghostview` to manipulate and view PostScript and PDF, plus a set of other tools that facilitate handling PostScript and PDF files and conversions.

## Typographic conventions

It is essential that the presentation of the material conveys immediately its function in the framework of the text. Therefore, we present below the typographic conventions used in this book.

Throughout the text, L<sup>A</sup>T<sub>E</sub>X command and environment names are set in mono-spaced type (e.g., `\includegraphics`, `sidewaystable`, `\begin{tabular}`), while names of package and class files are in sans serif type (e.g., `graphicx`). Commands to be typed by the user on a computer terminal are shown in monospaced type and are underlined (e.g., This is user input). *Commands, environments, packages, ...*

The syntax of the more complex L<sup>A</sup>T<sub>E</sub>X commands is presented inside a rectangular box. Command arguments are shown in italic type: *Syntax descriptions*

```
\includegraphics*[llx, lly] [urx, ury] {file}
```

In L<sup>A</sup>T<sub>E</sub>X, optional arguments are denoted with square brackets and the star indicates a variant form (i.e., is also optional), so the above box means that the `\includegraphics` command can come in six different incarnations:

```
\includegraphics{file}
\includegraphics [llx, lly] {file}
\includegraphics [llx, lly] [urx, ury] {file}
\includegraphics*{file}
\includegraphics* [llx, lly] {file}
\includegraphics* [llx, lly] [urx, ury] {file}
```

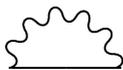
In case of PSTricks and, for example, the beamer class the syntax is not as straight forward and optional arguments may have other delimiters than brackets. For this reason they are shown with a gray background as in the following example:

```
\pstriangle* [settings] (xM, yM) (dx, dy)
```

Lines containing examples with L<sup>A</sup>T<sub>E</sub>X commands are indented and are typeset in a monospaced type at a size somewhat smaller than that of the main text: *Code examples ...*

```
\fmfdotn{v}{4}
\fmfv{decor.shape=circle,decor.filled=full,
      decor.size=2thick}{v1,v2,v3,v4}
```

However, in the majority of cases we provide complete examples together with the output they produce side by side: *... with output ...*



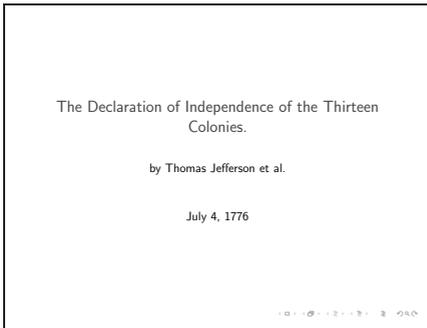
```
\usepackage{feyn}
$\feyn{fglf}$ \qqquad $\Feyn{fglf}$
```

Example  
0-0-1

Note that the preamble commands are always shown in blue in the example source.

... with several pages

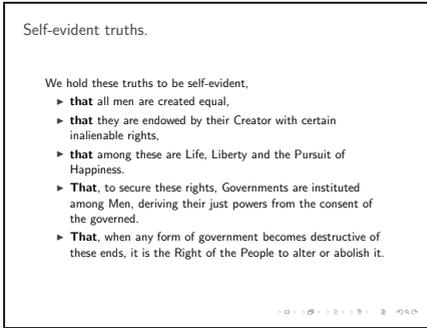
In case several pages need to be shown to prove a particular point, these are usually framed to indicate that we are showing material from several pages (this setup is repeatedly used in Section 11.4, where the `beamer` class for producing color slides with  $\LaTeX$ , is described), as shown here.



```
\documentclass{beamer}
\title{The Declaration of Independence of
the Thirteen Colonies.}
\author{by Thomas Jefferson et al.}
\date{July 4, 1776}
\frame{\maketitle}

\section{The unanimous Declaration}
\begin{frame}
\frametitle{Self-evident truths.}
We hold these truths to be self-evident,
\begin{itemize}
\item \textbf{that} all men are created equal,
\item \textbf{that} they are endowed by their
Creator with certain inalienable rights,
\item \textbf{that} among these are Life,
Liberty and the Pursuit of Happiness.
\item \textbf{That}, to secure these rights,
Governments are instituted among Men, deriving
their just powers from the consent of the governed.
\item \textbf{That}, when any form of government
becomes destructive of these ends, it is the Right
... further code omitted ...
```

Example 0-0-2



... with large output ...

For large examples, where the input and output cannot be shown conveniently alongside each other, the following layout is used:

```
\usepackage{feyn}
\begin{eqnarray}
\feyn{fcf} & \&\& \feyn{faf} + \feyn{fpf} + \cdots \\
& \&\& \sum_{n=0}^{\infty} \feyn{fsafs ( pfsafs)}^n
\end{eqnarray}
```

$$\text{---} \bigcirc \text{---} = \text{---} \rightarrow \text{---} + \text{---} \textcircled{\text{///}} \text{---} + \dots \tag{1}$$

$$= \sum_{n=0}^{\infty} \text{---} \rightarrow \text{---} \textcircled{\text{///}} \text{---} \rightarrow \text{---}^n \tag{2}$$

Example 0-0-3

Depending on the example content, some additional explanation might appear between input and output.

All of these examples are “complete” if, for the L<sup>A</sup>T<sub>E</sub>X examples, you mentally add a `\documentclass` line (with the `article` class<sup>1</sup> as an argument) and surround the body of the example with a `document` environment. In fact, this is how all the examples in this book were produced. When processing the book, special L<sup>A</sup>T<sub>E</sub>X commands take the source lines for an example and write them to an external file, thereby automatically adding the `\documentclass` or the relevant lines needed to run the example. This turns each example into a small but complete source document, which can then be externally processed (using a mechanism that runs each example as often as necessary; see also the next section on how to use the examples). The result is converted into small EPS graphics, which are then loaded in the appropriate place the next time L<sup>A</sup>T<sub>E</sub>X is run on the whole book. The implementation is based on the `fancyvrb` package, and is described in more details in *The L<sup>A</sup>T<sub>E</sub>X Companion* [83] (Section 3.4.3, in particular pages 162–163).

In some cases input for the examples may get very lengthy without providing additional insight to the reader. In that case some of it is replaced by the line “... further code omitted ...” to save space, as shown in Example 0-0-2. Technically this is achieved by placing the command `\empty` on a line by itself into the example code (where you will find it in the online version of the examples). When the example is processed to produce the output graphic this command is ignored, but when the code is read verbatim to show the input in the book, it serves as marker to end the code display.

*Omitting example code*

Throughout the book, blue notes are sprinkled in the margin to help you easily find certain information that would otherwise be hard to locate. In a few cases these notes exhibit a warning sign, indicating that you should probably read this information even if you are otherwise only skimming through the particular section.

 *Watch out for these*

## Using the examples

Our aim when producing this book was to make it as useful as possible for our readers. For this reason the book contains nearly 1200 complete, self-contained examples illustrating the main aspects of the packages and programs covered in the book.

We have put the source of the examples on CTAN (Comprehensive T<sub>E</sub>X Archive Network—see Appendix B) in the directory `info/examples/lgc2`. The examples are numbered per section, and each number is shown in a small box in the inner margin (e.g., 2-1-1 for the Example 2-1-1 on page 26). These numbers are also used for the external file names by appending a filetype that corresponds to the source. Most files are in L<sup>A</sup>T<sub>E</sub>X source format (with an extension of `.ltx` for a single page, or `.ltxb` for generating several pages when giving examples of the use of the `beamer` class). There are also plain T<sub>E</sub>X files (extension `.ptx`), METAPOST source files (extension `.mp`), MusiX<sub>T<sub>E</sub>X</sub> preprocessor source files (extensions `.abc`, `.abcplus`, `.pmx`, `.mtx`, and `.ly`), pic files (extension `.pic`), and m4 sources (extension `.m4`). For each of these types of sources there is a corresponding Unix script (`runabc`, `runabcpl`, `runltx`, `runltxb`, `runly`, `runm4`, `runmp`, `runmtx`, `runpic`, `runpmx`, `runptx`), which can be used as an example of how to run the given source file on a system where all the needed packages and software, as described in this book, are available.

*Online example sources*

---

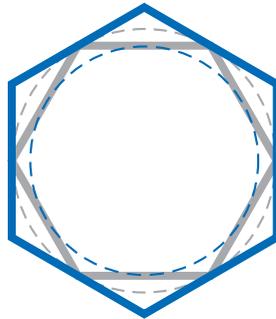
<sup>1</sup>Except for examples in Chapter 11 that require the `beamer` class.



```

\psset{unit=7mm}
\begin{pspicture}(-3,-3)(3,3)
  \psHexagon[linewidth=3pt,linecolor=red]{2.5}
  \pscircle[linestyle=dashed,linecolor=red]{2.5}
%
  \psHexagon[linewidth=3pt,linecolor=blue,HRInner=true]{2.5}
  \pscircle[linestyle=dashed,linecolor=blue]{2.17}
\end{pspicture}

```



Example  
0-0-4

You have two alternatives: You can copy the preamble code (i.e., code colored blue) into your own document preamble or you can place that code—but without the `\makeatletter` and `\makeatother` and with `\usepackage` replaced by `\RequirePackage`—in a package file (e.g., `myhexagon.sty`) and afterwards load this “package” in the preamble of your own documents with `\usepackage{myhexagon}`.

## Finding all those packages and programs

All of the packages and programs described in this book are freely available in public software archives; a few are in the public domain, but most are protected by copyright and available to you under an open-source license. Some programs are available only in source form or work only on certain computer platforms, and you should be prepared for a certain amount of “getting your hands dirty” in some cases. We also cannot guarantee that later versions of packages or programs will give results identical to those in our book. Many of these packages and programs remain under active development, and new or changed versions appear several times a year; we completed this book in spring 2007, and tested the examples with the versions current at that time.

In Appendix B we give full details on how to access CTAN sites and how to download files using the Internet. You can also purchase the *TeX Collection DVD* from one of the TeX Users Groups. This DVD contains implementations of TeX for various systems, many packages and fonts, in particular it provides you with all the L<sup>A</sup>TeX packages described in this book and *The L<sup>A</sup>TeX Companion, Second Edition*. Some programs (such as the ones described in the music chapter) are not available on CTAN (or the DVD) and must be downloaded from the location indicated in the text.

## Acknowledgments

We gratefully recognize all of our many colleagues in the T<sub>E</sub>X world who develop L<sup>A</sup>T<sub>E</sub>X packages—not only those described here, but also the hundreds of others that help users typeset their documents faster and better. Without the continuous effort of all these enthusiasts, T<sub>E</sub>X would not be the magnificent and flexible tool it is today.

We have many people to thank. Our primary debt, of course, is to the authors of the programs and packages we describe. Every author whom we contacted to discuss problems provided us with practical help in the spirit of the T<sub>E</sub>X community, and often gave us permission to reuse examples from their documentation.

We are greatly indebted to Eric Beitz, Ulrich Dirr, Ulrike Fischer, Federico Garcia, Uwe Kern, Claudia Krysztofiak, Aaron Lauda, Susan Leech O’Neale, Ross Moore, Janice Navarria, Han-Wen Nienhuys, Ralf Vogel, and Damien Wyart, for their careful reading of sections of the manuscript. Their numerous comments, suggestions, corrections, and hints have substantially improved the quality of the text. Special thanks go to Hubert Gäßlein, who greatly helped us at all stages of preparation, verification, and typesetting.

As he did with *The L<sup>A</sup>T<sub>E</sub>X Companion, Second Edition*, Richard Evans of Infodex Indexing Services in Raleigh, North Carolina, undertook the groundwork for the comprehensive indexes in the back of the book—thank you, Dick.

On the publishing side, we wish to thank Peter Gordon, our editor at Addison-Wesley, who gave us much-needed support and encouragement over the three years duration of this project. When it came to production, Elizabeth Ryan was unfailingly patient with our idiosyncrasies and steered us safely to completion. Jill Hobbs edited our dubious prose into real English; we greatly appreciate their work.

\* \* \*

Our families and friends have lived through the preparation of this book over several years, and we thank them for their patience and moral support.

## Feedback

We would like to ask you, dear reader, for your collaboration. We kindly invite you to send your comments, suggestions, or remarks to any of the authors. We shall be glad to correct any mistakes or oversights in a future edition, and are open to suggestions for improvements or the inclusion of important developments we may have overlooked. Any mistake or oversight found in this book and reported represents a gain for all readers. The latest version of the errata file (with contact details) can be found on the L<sup>A</sup>T<sub>E</sub>X project site at <http://www.latex-project.org/guides/lgc2.err> where you will also find an on-line version of the index and other extracts from the book.

*To Err is Human*

Michel Goossens  
Frank Mittelbach  
Sebastian Rahtz  
Denis Roegel  
Herbert Voß

June 2007