The LaTeX Graphics Companion

Second Edition

Michel Goossens
Frank Mittelbach
Sebastian Rahtz
Denis Roegel
Herbert Voß

Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers discounts on this book when ordered in quantity for bulk purchases and special sales. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside of the United States, please contact:

International Sales
international@pearsoned.com

Visit Addison-Wesley on the Web: www.awprofessional.com

Library of Congress Cataloging-in-Publication Data

```
The LaTeX Graphics companion / Michel Goossens ... [et al.]. -- 2nd ed.
       p. cm.
  Includes bibliographical references and index.
  ISBN 978-0-321-50892-8 (pbk. : alk. paper)
 1.  LaTeX (Computer file) 2.  Computerized typesetting. 3.  PostScript
(Computer program language) 4.  Scientific illustration--Computer programs.
5.  Mathematics printing--Computer programs. 6.  Technical
publishing--Computer programs.  I. Goossens, Michel.
  Z253.4.L38G663 2008
  686.2'2544536-dc22                               2007010278
```

Copyright © 2008 by Pearson Education, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The foregoing notwithstanding, the examples contained in this book and obtainable online on CTAN are made available under the LaTeX Project Public License (for information on the LPPL, see www.latex-project.org/lppl).

For information on obtaining permission for use of material from this work, please submit a written request to:

Pearson Education, Inc.
Rights and Contracts Department
75 Arlington Street, Suite 300
Boston, MA 02116
Fax: (617) 848-7047

ISBN 10: 0-321-50892-0
ISBN 13: 978-0-321-50892-8

Text printed in the United States on recycled paper at Courier in Westford, Massachusetts.

First printing, July 2007

We dedicate this book to the hundreds of LaTeX developers
whose contributions are showcased in it,
and we salute their enthusiasm and hard work.

We would also like to remember with affection and thanks
Daniel Taupin, whose MusiXTeX system is described in
Chapter 9, and who passed away in 2003, a great loss to our community.

Preface

More than a decade has passed since the publication of the first edition of *The LaTeX 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 LaTeX typesetting in the area of graphics and fonts. We examine how to draw pictures with LaTeX and how to incorporate graphics files into a LaTeX 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 TeX itself.

TeX 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. TeX'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 LaTeX.

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 MusiXTeX 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., pdflatex) that produce PDF directly are available.

Again, we apologize to those of you who are disappointed not to read about LaTeX'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

Basic information in Chapters 1 and 2

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 LaTeX standard graphics package, since the tools for including graphics files will be needed often. Chapter 2 also covers pict2e, a package that reimplements LaTeX'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 LaTeX's `picture`, but also offer new and powerful commands to draw arcs and curves with mininal effort.

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: TeX's Mates shows how to exploit the power of TeX'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 TeX 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 LaTeX: 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 MusiXTEX 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 TEX*muse*.

10 Playing Games is for those who use LATEX 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 LATEX.

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 LATEX.

Appendix A describes ways to generate PDF from LATEX. Appendix B introduces CTAN and explains how to download the LATEX 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 LATEX, 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 LATEX. It also looks at the latest developments on how to integrate OpenType fonts by creating TEX-specific auxiliary files (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.

Commands, environments, packages, . . .

Throughout the text, LaTeX 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`).

Syntax descriptions

The syntax of the more complex LaTeX commands is presented inside a rectangular box. Command arguments are shown in italic type:

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

In LaTeX, 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 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] (x_M,y_M) (dx,dy)
```

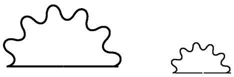
Code examples . . .

Lines containing examples with LaTeX commands are indented and are typeset in a monospaced type at a size somewhat smaller than that of the main text:

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

. . . with output . . .

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



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

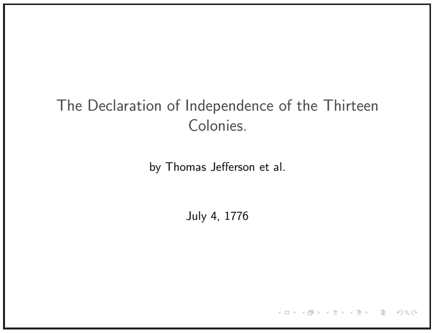
Example
0-0-1

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.

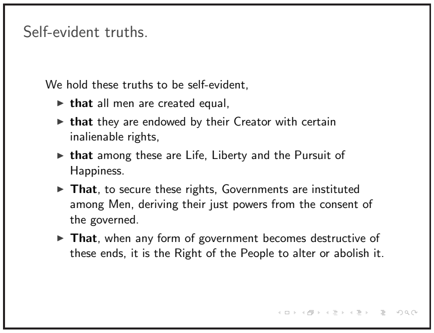
Example 0-0-2



The Declaration of Independence of the Thirteen Colonies.

by Thomas Jefferson et al.

July 4, 1776



Self-evident truths.

We hold these truths to be self-evident,

- **that** all men are created equal,
- **that** they are endowed by their Creator with certain inalienable rights,
- **that** among these are Life, Liberty and the Pursuit of Happiness.
- **That**, to secure these rights, Governments are instituted among Men, deriving their just powers from the consent of the governed.
- **That**, when any form of government becomes destructive of these ends, it is the Right of the People to alter or abolish it.

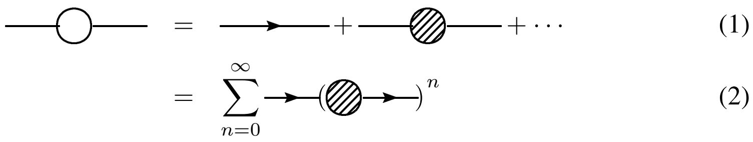
```
\documentclass{beamer}
\title{The Declaration of Independence of
       the Thirteen Colonies.}
\author{by Thomas Jefferson et al.}
\date{July 4, 1776}
\frame{\maketitle}

\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 ...
```

… 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}
```

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 LaTeX examples, you mentally add a `\documentclass` line (with the article class[1] 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 LaTeX 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 LaTeX is run on the whole book. The implementation is based on the fancyvrb package, and is described in more details in *The LaTeX Companion* [83] (Section 3.4.3, in particular pages 162–163).

Omitting example code

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.

Watch out for these

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.

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.

Online example sources

We have put the source of the examples on CTAN (Comprehensive TeX 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 LaTeX 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 TeX files (extension `.ptx`), METAPOST source files (extension `.mp`), MusiXTeX 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.

[1] Except for examples in Chapter 11 that require the beamer class.

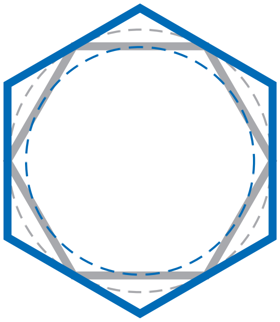
To reuse any of the examples it is usually sufficient to copy the preamble code (typeset in blue) into the preamble of your document and, if necessary, adjust the document text as shown. In some cases it might be more convenient to place the preamble code into your own package (or class file), thus allowing you to load this package in multiple documents using `\usepackage`. If you want to do the latter, there are two points to observe:

- Any use of the `\usepackage` command in the preamble code should be replaced by `\RequirePackage`, which is the equivalent command for use in package and class files (see e.g., Section A.4.5 of *The LaTeX Companion* [83]).
- Any occurrence of `\makeatletter` and `\makeatother` *must* be removed from the preamble code. This is very important because the `\makeatother` would stop correct reading of such a file.

So let us assume you wish to reuse the code from the following (rather complex) example:

```
\usepackage{pstricks,pst-xkey}

\makeatletter                        % ‘@’ now normal "letter"

\newif\ifHRInner
\def\psset@HRInner#1{\@nameuse{HRInner#1}}
\psset@HRInner{false}
\def\psHexagon{\pst@object{psHexagon}}
\def\psHexagon@i{\@ifnextchar({\psHexagon@ii}%
                              {\psHexagon@ii(0,0)}}
\def\psHexagon@ii(#1)#2{%
  \begin@ClosedObj%                      closed object
    \pst@@getcoor{#1}%                   get center
    \pssetlength\pst@dimc{#2}%           set radius to pt
    \addto@pscode{%                      PostScript
      \pst@coor T %                      xM yM new origin
      \psk@dimen CLW mul %               set line width
      /Radius \pst@number\pst@dimc\space % save radius
        \ifHRInner\space 3 sqrt 2 div div \fi def % inner?
      /angle \ifHRInner 30 \else 0 \fi def % starting angle
      Radius angle PtoC moveto %         go to first point
      6 { %                              6 iterations
        /angle angle 60 add def %        alpha = alpha+60
        Radius angle PtoC L %            line to next point
      } repeat
      closepath %                        closed object
    }%
    \def\pst@linetype{3}%                set linetype
    \showpointsfalse%                    do not show base points
  \end@ClosedObj%                        end
  \ignorespaces}%                        swallow spaces

\makeatother                  % ‘@’ is restored as "non-letter"
```

```
\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 LaTeX packages described in this book and *The LaTeX 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 TeX world who develop LaTeX 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, TeX 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 TeX 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 LaTeX 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

To Err is Human

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 LaTeX project site at `http://www.latex-project.org/guides/lgc2.err` where you will also find an online version of the index and other extracts from the book.

Michel Goossens
Frank Mittelbach
Sebastian Rahtz
Denis Roegel
Herbert Voß

June 2007

CHAPTER 1

Graphics with LaTeX

1.1 Graphics systems and typesetting 2
1.2 Drawing types 3
1.3 TeX's interfaces 6
1.4 Graphics languages 10
1.5 Choosing a package 21

The phrase "A picture paints a thousand words" seems to have entered the English language thanks to Frederick R. Barnard in *Printer's Ink*, 8 December 1921, retelling a Chinese proverb.[1] However, while LaTeX is quite good at typesetting words in a beautiful manner, LaTeX manuals usually tell you little or nothing about how to handle graphics. This book attempts to fill that gap by describing tools and TeXniques that let you generate, manipulate, and integrate graphics with your text.

In these days of the multimedia PC, graphics appear in various places. With many products we get ready-to-use collections of clipart graphics; in shops we can buy CD-ROMs with "the best photos" of important places; and so forth. As we shall see, all such graphics can be included in a LaTeX document as long as they are available in a suitable format. Fortunately, many popular graphic formats either are directly supported or can be converted via a program that allows transformation into a supported representation.

If you want to become your own graphic artist, you can use stand-alone dedicated drawing tools, such as the freely available dia (`www.gnome.org/projects/dia`) and xfig (`www.xfig.org/userman`) on Linux, or the commercial products Adobe Illustrator (`www.adobe.com/illustrator`) or Corel Draw (`www.corel.com/coreldraw`) on a Mac or PC. Spreadsheet programs, or one of the modern calculation tools like Mathematica

[1] Paul Martin Lester (`commfaculty.fullerton.edu/lester/writings/letters.html`) states that the literal translation of the "phony" Chinese proverb should rather be "A picture's meaning can express ten thousand words". He, rightly, emphasizes that pictures cannot and should not replace words, but both are complementary and contribute equally to the understanding of the meaning of a work.

(www.wolfram.com/mathematica), Maple (www.maplesoft.com/maple), and MATLAB (www.mathworks.com/matlab), or their freely available GNU variant Octave (www.octave.org) and its plotting complements Octaviz (octaviz.sourceforge.net) and Octplot (octplot.sourceforge.net), can also produce graphics by using one of their many graphical output representations. With the help of a scanner or a digital camera you can produce digital photos, images of hand-drawn pictures, or other graphics that can be manipulated with their accompanying software. In all these cases it is easy to generate files that can be directly referenced in the LaTeX source through the commands of the graphics package described in Chapter 2.

If needed, LaTeX can also offer a closer integration with the typesetting system than that possible by such programs. Such integration is necessary if you want to use the same fonts in text and graphics, or more generally if the "style" of the graphics should depend on the overall style of the document. Close integration of graphics with the surrounding text clearly requires generation of the graphic by the typesetting system itself, because otherwise any change in the document layout style requires extensive manual labor and the whole process becomes very error-prone.

* * *

This chapter considers graphic objects from different angles. First. we look at the requirements that various applications impose on graphic objects. Next, we analyze the types of drawings that appear in documents and the strategies typically employed to generate, integrate, and manipulate such graphics. Then, we discuss the interfaces offered by TeX for dealing with graphic objects. Armed with this knowledge, we end the chapter with a short survey of graphics languages built within and around TeX. This overview will help you select the right tool for the job at hand. In fact, the current chapter also gives some examples of languages and approaches not covered in detail elsewhere in the book. Thus this survey should provide you with enough information to decide whether or not to follow the pointers and obtain such a package for a particular application.

1.1 Graphics systems and typesetting

When speaking about "graphic objects", we should first define the term. One extreme position is to view everything put on paper as a graphic object, including the characters of the fonts used. This quite revolutionary view was, in fact, adopted in the design of the page description language PostScript, in which characters can be composed and manipulated by exactly the same functions as other graphic objects (we will see some examples of this in Chapters 5 and 6, which describe PSTricks and its support packages).

Most typesetting systems, including TeX, do not try to deploy such a general model but instead restrict their functional domain to a subset of general graphic objects—for example, by providing very sophisticated functions to place characters, resolve ligatures, etc., but omitting operators to produce arbitrary lines, construct and fill regions, and so forth. As a result the term "graphics" for most LaTeX users is a synonym for "artwork", thereby ignoring the fact that LaTeX already has a graphics language—the picture mode.

When discussing the graphical capabilities of an ideal typesetting system, we must remember that different applications have different, sometimes conflicting requirements:

- One extreme is the need for complete portability between platforms; another is to take into account even differences in the way printers put ink onto paper.
- A graphic might need to be correctly scaled to a certain size depending on factors of the visual environment created by the typesetting system, e.g., the measure of the text.
- It is also possible that parts of the graphic should not scale linearly. For example, it might be important for readability to ensure that textual parts of a graphic do not become smaller or larger than some limit. It might also be required that, when a graphic is scaled by, say, 10% to fit the line, any included text must stay the same, so as to avoid making it larger than the characters in the main document body.
- It might be required that the graphical object be closely integrated with the surrounding text, such as by using the same fonts as in other parts of the document or more generally by containing objects that should change their appearance if the overall style of the document is changed. (The latter is especially important if the document is described by its logical content rather than by its visual appearance, with the intention of reusing it in various contexts and forms.)

As LaTeX is a general-purpose typesetting system used for all types of applications, the preceding requirements and more might arise in various situations. As we will see throughout this book, a large number of them can be handled with grace, if not to perfection. In some cases an appropriate solution was anything but obvious and developing the mature macro packages and programs we now have took a decade or more of work.

1.2 Drawing types

The typology of graphics at the beginning of this chapter focused on the question of the integration with the LaTeX system, and divided the graphics into externally and internally generated ones. A different perspective would be to start from the types of graphics we might encounter in documents and discuss possible ways to generate and incorporate them.

A first class of graphics to be included are treated by LaTeX as a single object, a "black box", without an accessible inner structure. LaTeX, via its graphics package (described in Chapter 2), is interested only in the rectangular dimensions of the graphic image, its "bounding box". The graphics will be included in the output "as is", possibly after some simple manipulation, such as scaling or rotation. On top of that LaTeX can also produce a caption and legend to allow proper referencing from within the document. The main categories are as follows:

1. *Free-hand pictures* drawn without a computer, such as the drawing of a glass bead in Figure 1.1. For use in LaTeX, such a graphic must to be transformed into a digital image, using, for example, a scanner.

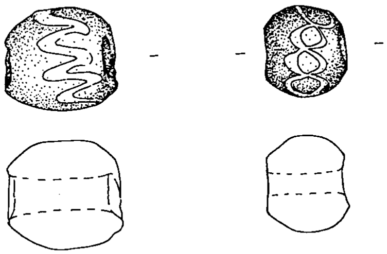


Figure 1.1: Pen and ink drawing of a bead

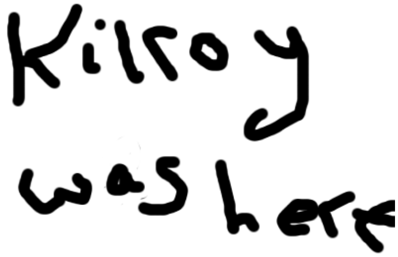


Figure 1.2: Bitmap drawing output created with GIMP

2. *"Art" graphics* drawn with bitmap tools on a computer, such as the example in Figure 1.2, which are to some extent the computer equivalents of pen and ink drawings. This drawing was created with GIMP, the GNU Image Manipulation Program (`www.gimp.org`), using a deliberately crude technique. The distinctive characteristic of this type of drawing is that the resolution chosen in the generation process cannot easily be changed without loss of quality (or alternatively without a lot of manual labor). In other respects such a picture is like a free-hand drawing: there is generally no desire to integrate the drawing with the text or to worry about conformity of typefaces.

3. *Photographs* either created directly using a digital camera or scanned like hand-drawn pictures. In the latter case the continuous tones of the photograph are converted into a distinct range of colors or gray levels (black-and-white photographs treated in this way are known as half-tones). Full-color reproduction requires sophisticated printing techniques, but this issue arises at the printing stage and does not normally affect the typesetting. Figure 1.3 shows how LATEX can distort the image.

A second class of graphics is the "object-oriented" type, where the information is stored in the form of abstract objects that incorporate no device-dependent information (unlike bitmap graphics, where the storage format just contains information about whether a certain spot is black or white, making them resolution-dependent). This device independence makes it easy to reuse the graphic with different output devices and allows us to manipulate individual aspects of the graphic during the design process.

There are essentially three types of such graphics systems: one in which LATEX mainly remains passive (it just takes into account the bounding box of the picture), and two others that relate to graphics that contain more complex text, in particular formulae. For the latter types it is important to use LATEX to typeset text within the graphic because the symbols in formulae and their typeset form carry a precise semantic meaning. Therefore one must take great care to ensure that their visual representation is identical in both text and associated graphics.

1. *Self-contained object-oriented graphics.* The ducks of Figure 1.4, which was produced with Adobe Illustrator, were created by drawing one object in terms of curves and then

Figure 1.3: Digitally transformed image (vertically stretched)

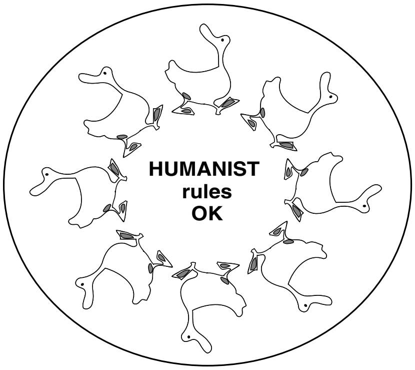


Figure 1.4: Object-oriented drawing

copying and rotating it many times. This type of drawing often also contains textual annotations comparable to typeset text. Although it is usually possible to add text to the graphic with external tools such Illustrator, it is not in general possible to use LaTeX to typeset this text (although psfrag provides a solution in some circumstances).

2. *Algorithmic display graphics* (e.g., histograms, graphs). These drawings are created without human interaction but often contain text that should match the document text. The scale and distance between elements is an essential characteristic of the drawing.

   Extensive plotting and diagram facilities are provided by many LaTeX packages building on the `picture` mode, by generic TeX packages such as PiCTeX [139], DraTex [39], and tikz [115]; and by PSTricks (see Chapters 5 and 6). All these solutions let us deploy the full power of LaTeX's typesetting functions within textual parts of the graphic and thus integrate it perfectly with surrounding document elements.

3. *Algorithmic structural graphics*, which can be derived from a textual representation. Unlike with the previous category, often merely the spatial relationship between elements is important with these graphics, not the elements' exact position or size. Examples are category diagrams, chemical formulae, trees, and flowcharts. Such graphics are natural candidates for generation by graphics languages internal to LaTeX that provide high-level interfaces which focus on objects and relationships and decide final placement and layout automatically.

   Of the general-purpose languages, the METAPOST system (Chapters 3 and 4) is perhaps the most flexible one for this type of graphics, although PiCTeX, Xy-pic (Chapter 7), PSTricks (Chapters 5 and 6), and DraTex are also suitable. They are based on different paradigms, and differ greatly in approach, focus, and user interface, but they all have found their place in the LaTeX world. We describe small specialized languages tailored for specific application domains such as physics, chemistry or electronics diagrams (Chapter 8), music (Chapter 9), and games (Chapter 10). For special applications such as tree drawing, many other LaTeX languages are available as well (see [13], for instance).

As we see, many types of graphics exist, each with its own requirements. The first three types essentially present themselves as black boxes to LaTeX and thus their use within a LaTeX document involves no more than their inclusion and in some cases their manipulation as a whole. The necessary functionality is discussed in detail in Chapter 2.

In scientific texts, the other types of graphics are by far the more common. Examples include maps [119], chemical structures, or commutative diagrams. They are for the most part based on an object-oriented approach, specifying objects and their relations in an abstract way using a suitable language. Close integration with the surrounding text can be achieved, if needed, by choosing one of the graphics languages described in this book.

In some cases interactive drawing programs can be instructed to output their results in one of the graphics languages built directly on top of LaTeX's `picture` mode. Widely used examples under Linux are dia and xfig, whose pictures, although externally produced, can be influenced by layout decisions within the document. Note, however, that such mechanically produced LaTeX code is normally not suitable for further manual editing and manipulation is practically limited to layout facilities implemented by the chosen graphics language. Nevertheless, in certain situations this approach can offer the best of two worlds.

1.3 TeX's interfaces

To understand the merits of the different approaches to graphics as implemented by various packages, it is helpful to consider yet another point of view: the interfaces provided by TeX for dealing with them. Describing the methods by which graphics can be generated, included, or manipulated will give you some feeling for such important issues as portability, quality, and resource requirements of individual solutions. We assume that the reader has a reasonable understanding of how TeX works—that is, the progression from source file to a DVI file that is processed by a driver to produce printed pages. Of course, the DVI stage can be skipped when using pdflatex, but the various ways of including the graphics material are still identical.

In the following we first look at ways of including externally generated graphics (i.e., those that appear as black boxes to TeX) and methods to manipulate them. Then we consider interfaces provided to build graphics languages within TeX.

1.3.1 Methods of integration

TeX offers two major facilities for integrating graphics as a whole: one involving the `\special` command, and the other using the font interface.

Using `\special` commands

The TeXbook [70] does not describe ways to directly include externally generated graphics. The only command available is the `\special` command, which by itself does nothing, but does enable us to access capabilities that might be present in the post-processor (DVI driver or pdflatex). To quote Knuth [70, page 229]:

> The `\special` command enables you to make use of special equipment that might be available to you, e.g., for printing books in glorious TeXnicolor.

CHAPTER 2

Standard LATEX Interfaces

2.1 Inclusion of graphics files 23
2.2 Manipulating graphical objects 36
2.3 Line graphics 42

Since the introduction of LATEX 2ε in 1994, LATEX has offered a uniform syntax for including every kind of graphics file that can be handled by the different drivers. In addition, all kinds of graphic operations (such as resizing and rotating) as well as color support are available.

These features are not part of the LATEX 2ε kernel, but rather are loaded by the standard, fully supported color, graphics, and graphicx extension packages. Because the TEX program does not have any direct methods for graphic manipulation, the packages must rely on features supplied by the "driver" used to print the `dvi` file. Unfortunately, not all drivers support the same features, and even the internal method of accessing these extensions varies among drivers. Consequently, all of these packages take options, such as `dvips`, to specify which external driver is being used. Through this method, unavoidable device-dependent information is localized in a single place, the preamble of the document.

In this chapter we start by looking at graphics file inclusion. LATEX offers both a simple interface (graphics), which can be combined with the separate rotation and scaling commands, and a more complex interface (graphicx), which features a powerful set of manipulation options. The chapter concludes with a discussion of the pict2e package, which implements the driver encapsulation concept for line graphics and with a brief description of the curve2e package, which is not part of the "standard LATEX interface" but nevertheless represents an interesting extension to pict2e. Color support is covered in Chapter 11.

2.1 Inclusion of graphics files

The packages graphics and graphicx can both be used to scale, rotate, and reflect LATEX material or to include graphics files prepared with other programs. The difference between

Table 2.1: Overview of color and graphics capabilities of device drivers

Option	*Author of Driver*	*Features*
`dvips`	T. Rokicki	All functions (reference driver; option also used by xdvi)
`dvipdf`	S. Lesenko	All functions
`dvipdfm`	S. Lesenko	All functions
`dvipsone`	Y&Y	All functions
`dviwin`	H. Sendoukas	File inclusion
`emtex`	E. Mattes	File inclusion only, but no scaling
`pdftex`	Hàn Thế Thành	All functions for pdftex program
`pctexps`	PCTeX	File inclusion, color, rotation
`pctexwin`	PCTeX	File inclusion, color, rotation
`pctex32`	PCTeX	All functions
`pctexhp`	PCTeX	File inclusion only
`truetex`	Kinch	Graphics inclusion and some color
`tcidvi`	Kinch	TrueTeX with extra support for Scientific Word
`textures`	Blue Sky	All functions for Textures program
`vtex`	Micropress	All functions for VTeX program

the two is that graphics uses a combination of macros with a "standard" or TEX-like syntax, while the "extended" or "enhanced" graphicx package presents a key/value interface for specifying optional parameters to the `\includegraphics` and `\rotatebox` commands.

2.1.1 Options for graphics and graphicx

When using LATEX's graphics packages, the necessary space for the typeset material after performing a file inclusion or applying some geometric transformation is reserved on the output page. It is, however, the task of the *device driver* (e.g., dvips, xdvi, dvipsone) to perform the actual inclusion or transformation in question and to show the correct result. Given that different drivers may require different code to carry out an action, such as rotation, one has to specify the target driver as an option to the graphics packages—for example, option `dvips` if you use one of the graphics packages with Tom Rokicki's dvips program, or option `textures` if you use one of the graphics packages and work on a Macintosh using Blue Sky's Textures program.

Some drivers, such as previewers, are incapable of performing certain functions. Hence they may display the typeset material so that it overlaps with the surrounding text. Table 2.1 gives an overview of the more important drivers currently supported and their possible limitations. Support for older driver programs exists usually as well—you can search for it on CTAN.

The driver-specific code is stored in files with the extension `.def`—for example, `dvips.def` for the PostScript driver dvips. As most of these files are maintained by third parties, the standard LATEX distribution contains only a subset of the available files and not necessarily the latest versions. While there is usually no problem if LATEX is installed as part of a full TEX installation, you should watch out for incompatibilities if you update the LATEX graphics packages manually.

Setting a default driver

It is also possible to specify a default driver using the \ExecuteOptions declaration in the *configuration* file graphics.cfg. For example, \ExecuteOptions{dvips} makes the dvips drivers become the default. In this case the graphics packages pick up the driver code for the dvips TeX system on a PC if the package is called without a driver option. Most current TeX installations are distributed with a ready-to-use graphics.cfg file.

In addition to the driver options, the packages support some options controlling which features are enabled (or disabled):

draft Suppress all "special" features, such as including external graphics files in the final output. The layout of the page will not be affected, because LaTeX still reads the size information concerning the bounding box of the external material. This option is of particular interest when a document is under development and you do not want to download the (often huge) graphics files each time you print the typeset result. When draft mode is activated, the picture is replaced by a box of the correct size containing the name of the external file.

final The opposite of draft. This option can be useful when, for instance, "draft" mode was specified as a global option with the \documentclass command (e.g., for showing overfull boxes), but you do not want to suppress the graphics as well.

hiresbb In PostScript files, look for bounding box comments that are of the form %%HiResBoundingBox (which typically have real values) instead of the standard %%BoundingBox (which should have integer values).

hiderotate Do not show the rotated material (for instance, when the previewer cannot rotate material and produces error messages).

hidescale Do not show the scaled material (for instance, when the previewer does not support scaling).

With the graphicx package, the options draft, final, and hiresbb are also available locally for individual \includegraphics commands, that is, they can be selected for individual graphics.

2.1.2 The \includegraphics syntax in the graphics package

With the graphics package, you can include an image file by using the following command:

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

If the [*urx*,*ury*] argument is present, it specifies the coordinates of the upper-right corner of the image as a pair of TeX dimensions. The default units are big (PostScript) points; thus [1in,1in] and [72,72] are equivalent. If only one optional argument is given, the lower-left corner of the image is assumed to be located at [0,0]. Otherwise, [*llx*,*lly*] specifies the coordinates of that point. Without optional arguments, the size of the graphic is determined by reading the external *file* (containing the graphics itself or a description thereof, as discussed later).

```
%!PS-Adobe-2.0
%%BoundingBox:100 100 150 150
100 100     translate % put origin at 100 100
  0   0     moveto    % define current point
 50  50     rlineto   % trace diagonal line
 50 neg 0   rlineto   % trace horizontal line
 50  50 neg rlineto   % trace other diagonal line
stroke                % draw (stroke) the lines
  0   0     moveto    % redefine current point
/Times-Roman findfont % get Times-Roman font
 50         scalefont % scale it to 50 big points
            setfont   % make it the current font
(W) show              % draw an uppercase W
```

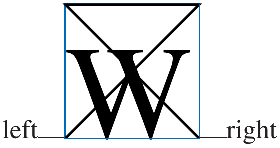
Figure 2.1: The contents of the file `w.eps`

The starred form of the `\includegraphics` command “clips” the graphics image to the size of the specified bounding box. In the normal form (without the `*`), any part of the graphics image that falls outside the specified bounding box overprints the surrounding text.

The examples in the current and next sections use a small PostScript program (in a file `w.eps`) that paints a large uppercase letter “W” and a few lines. Its source is shown in Figure 2.1. Note the `BoundingBox` declaration, which stipulates that the image starts at the point `100, 100` (in big points), and goes up to `150, 150`; that is, its natural size is 50 big points by 50 big points.

In the examples we always embed the `\includegraphics` command in an `\fbox` (with a blue frame and zero `\fboxsep`) to show the space that LaTeX reserves for the included image. In addition, the baseline is indicated by the horizontal rules produced by the `\HR` command, defined as an abbreviation for `\rule{1em}{0.4pt}`.

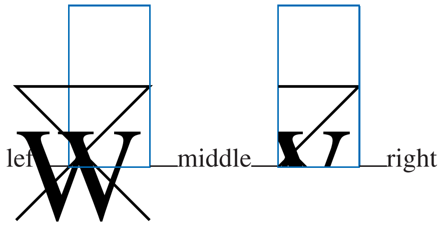
The first example shows the inclusion of the `w.eps` graphic at its natural size. Here the picture and its bounding box coincide nicely.



```
\usepackage{graphics,color}
\newcommand\HR{\rule{1em}{0.4pt}}
\newcommand\bluefbox[1]{\textcolor{blue}{%
   \setlength\fboxsep{0pt}\fbox{\textcolor{black}{#1}}}}
left\HR \bluefbox{\includegraphics{w.eps}}\HR right
```

Example 2-1-1

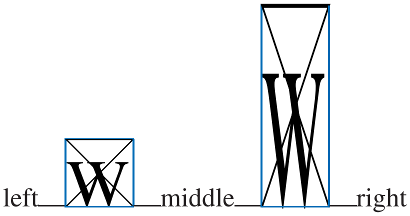
Next, we specify a box that corresponds to a part of the picture (and an area outside it) so that some parts fall outside its boundaries, overlaying the material surrounding the picture. If the starred form of the command is used, then the picture is clipped to the box (specified as optional arguments), as shown on the right.



```
\usepackage{graphics,color}
% \bluefbox and \HR as before
left\HR
 \bluefbox{\includegraphics
          [120,120][150,180]{w.eps}}%
\HR middle\HR
 \bluefbox{\includegraphics*
          [120,120][150,180]{w.eps}}%
\HR right
```

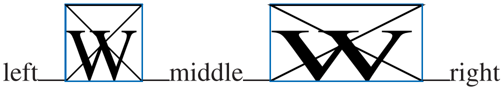
Example 2-1-2

In the remaining examples we combine the \includegraphics command with other commands of the graphics package to show various methods of manipulating an included image. (Their exact syntax is discussed in detail in Section 2.2.) We start with the \scalebox and \resizebox commands. In both cases we can either specify a change in one dimension and have the other scale proportionally, or specify both dimensions to distort the image.



```
\usepackage{graphics,color}
% \bluefbox and \HR as before
left\HR
  \bluefbox{\scalebox{.5}{%
    \includegraphics{w.eps}}}%
\HR middle\HR
  \bluefbox{\scalebox{.5}[1.5]{%
    \includegraphics{w.eps}}}%
\HR right
```

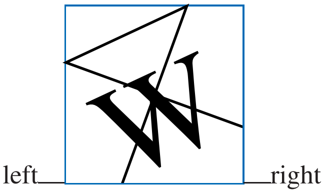
Example 2-1-3



```
\usepackage{graphics,color}
% \bluefbox and \HR as before
left\HR
  \bluefbox{\resizebox{10mm}{!}{%
    \includegraphics{w.eps}}}%
\HR middle\HR
  \bluefbox{\resizebox{20mm}{10mm}{%
    \includegraphics{w.eps}}}%
\HR right
```

Example 2-1-4

Adding rotations makes things even more interesting. Note that in comparison to Example 2-1-1 on the facing page the space reserved by LaTeX is far bigger. LaTeX “thinks” in rectangular boxes, so it selects the smallest size that can hold the rotated image.



```
\usepackage{graphics,color}
% \bluefbox and \HR as before
left\HR
  \bluefbox{\rotatebox{25}{%
    \includegraphics{w.eps}}}%
\HR right
```

Example 2-1-5

2.1.3 The `\includegraphics` syntax in the graphicx package

The extended graphics package **graphicx** also implements `\includegraphics` but offers a syntax for including external graphics files that is somewhat more transparent and user-friendly. With today's TEX implementations, the resultant processing overhead is negligible, so we suggest using this interface.

```
\includegraphics*[key/val-list]{file}
```

The starred form of this command exists only for compatibility with the standard version of `\includegraphics`, as described in Section 2.1.2. It is equivalent to specifying the `clip` key.

The *key/val-list* is a comma-separated list of *key*=*value* pairs for keys that take a value. For Boolean keys, specifying just the key is equivalent to *key*=`true`; not specifying the key is equivalent to *key*=`false`. Possible keys are listed below:

`bb` The bounding box of the graphics image. Its value field must contain four dimensions, separated by spaces. This specification will overwrite the bounding box information that might be present in the external file.[1]

`hiresbb` Makes LATEX search for `%%HiResBoundingBox` comments, which specify the bounding box information with decimal precision, as used by some applications. In contrast, the normal `%%BoundingBox` comment can take only integer values. It is a Boolean value, either "`true`" or "`false`".

`viewport` Defines the area of the graphic for which LATEX reserves space. Material outside this will still be print unless `trim` is used. The key takes four dimension arguments (like `bb`), but the origin is with respect to the bounding box specified in the file or with the `bb` keyword. For example, to describe a 20 bp square 10 bp to the right and 15 bp above the lower-left corner of the picture you would specify `viewport=10 15 30 35`.

`trim` Same functionality as the `viewport` key, but this time the four dimensions correspond to the amount of space to be trimmed (cut off) at the left-hand side, bottom, right-hand side, and top of the included graphics.

`natheight,natwidth` The natural height and width of the figure, respectively.[2]

`angle` The rotation angle (in degrees, counterclockwise).

`origin` The origin for the rotation, similar to the `origin` parameter of the `\rotatebox` command described on page 40.

`width` The required width (the width of the image is scaled to that value).

[1] There also exists an obsolete form kept for backward compatibility only: `[bbllx=a, bblly=b, bburx=c, bbury=d]` is equivalent to `[bb = a b c d]`, so the latter form should be used.

[2] These arguments can be used for setting the lower-left coordinate to `(0 0)` and the upper-right coordinate to `(natwidth natheight)` and are thus equivalent to `bb=0 0 w h`, where `w` and `h` are the values specified for these two parameters.

CHAPTER 3

METAFONT and METAPOST: TEX's Mates

3.1 The META language 52
3.2 Differences between METAPOST and METAFONT 60
3.3 Running the META programs 68
3.4 Some basic METAPOST libraries 74
3.5 The METAOBJ package 80
3.6 TEX interfaces: getting the best of both worlds 120
3.7 From METAPOST and to METAPOST 137
3.8 The future of METAPOST 138

In designing the TEX typesetting system, Donald Knuth soon realized that he would also have to write his own font design program. He devised METAFONT, a language for describing shapes, and a program to interpret that language and turn the shapes into a pattern of dots for a printing or viewing device. The result of Knuth's work was TEX, METAFONT, and the extensive Computer Modern font family written in METAFONT. METAFONT has also been used to create special-purpose symbol fonts and some other font families.

The development of METAFONT as a font description language paralleled to some extent that of the PostScript language, which also describes character shapes very elegantly. PostScript's strategy, however, is to leave the rendering of the shape until the final printing stage, whereas METAFONT seeks to precompute the bitmap output and print it on a fairly dumb printing device.

Font design is a decidedly specialist art, and one that most of us are ill equipped to tackle. METAFONT, however, defines a very powerful language that can cope with most graphical tasks. A sibling program, METAPOST, was developed that uses essentially the same language but generates PostScript instead of bitmaps. Together, the two provide an excellent companion facility with which (LA)TEX users can illustrate their documents, particularly when they want pictures that graphically express some mathematical construct; this is not surprising, given that Knuth's aim was to describe font shapes mathematically. Applications vary from drawing Hilbert or Sierpiński curves (described in Section 4.4.3) to plotting data in graphs and expressing relationships in graphical form.

In this chapter we consider how to use both METAFONT and METAPOST (henceforth we use META to mean "both METAFONT and METAPOST") to draw pictures and shapes other than characters in fonts.

Our coverage of META is divided into six parts. We start with a brief look at the META language basics; our aim is to give readers new to META some ideas of its facilities and the level at which pictures can be designed. We try to explain commands as they are used, but some examples may contain META code that is not explicitly described.

We next consider in some detail the extra facilities of the METAPOST language, in particular the inclusion of text and color in figures.

The third section examines how the META programs are run and how resulting figures can be included in a LaTeX document. The following section describes the general-purpose METAPOST libraries, covering in particular boxing macros and the METAOBJ package.

We then look at programs that write META commands for you, concentrating on the mfpic (LA)TEX package. We conclude with an overview of miscellaneous tools and utilities related to METAPOST.

For some applications, such as drawing of graphs, diagrams, geometrical figures, and 3-D objects, higher-level macro packages have been developed, which define their own languages for the user. These packages are described in Chapter 4.

3.1 The META language

The full intricacies of METAFONT are described in loving detail in [72]; the manual for METAPOST [47] not only describes the differences between the two systems, but is itself a good introduction to META. Alan Hoenig's book *TEX Unbound* [49] provides a wealth of material on METAFONT techniques. Articles over many years in the journal *TUGboat* are also vital reading for those who want to delve deeply into METAFONT and METAPOST.

The job of the META language is to describe shapes; these shapes can then be filled, scaled, rotated, reflected, skewed, and shifted, among other complex transformations. Indeed, META programs can be regarded as specialized equation-solving systems that have the side effect of producing pictures.

META offers all the facilities of a conventional programming language. Program flow control, for example, is provided by a `for` ...`endfor` construct, with the usual conditionals. You can write parameterized macros or subroutines, and there are facilities for local variables and grouping to limit the scope of value changes. Some of these features are described with more detail in the METAPOST section, although they are also available in METAFONT.

Because a lot of the work in writing META programs deals with describing geometrical shapes, the numeric support is extensive. For instance, Pythagorean addition (++) and subtraction (+-+) are directly supported. Useful numeric functions include `length` x

(absolute value of x), `sqrt` x (square root of x), `sind` x (sine of x degrees), `cosd` x (cosine of x degrees), `angle` (x, y) (arctangent of y/x), `floor` x (largest integer $\leq x$), `uniformdeviate` x (uniformly distributed random number between 0 and x), and `normaldeviate` (normally distributed random number with mean 0 and standard deviation 1).

A variety of complex data types are defined, including `boolean`, `numeric`, `pair`, `path`, `pen`, `picture`, `string`, and `transform`. Here we can look at some of these in more detail:

pair "Points" in two-dimensional space are represented in META with the type `pair`. Constants of type `pair` have the form (x, y), where x and y are both `numeric` constants. A variable p of type `pair` is equal to the `pair` expression (`xpart` p, `ypart` p).

path A `path` is a continuous curve, which is composed of a chain of *segments*. Each segment has a shape determined by four *control points*. Two of the control points, the *key* points, are the segment's end points; very often we let META determine the other two control points.

pen Pens, a distinctive feature of META, are filled convex shapes that are moved along `path`s and affect the way lines are drawn in the result. Two pens are initially present in META: `nullpen` and `pencircle`. `nullpen` is the single point $(0, 0)$; it contains no pixels and can be used to fill a region without changing its boundary. By contrast, `pencircle` is circular, with the points $(\pm 0.5, 0)$ and $(0, \pm 0.5)$ on its circumference. Other pens are constructed as convex polygons via `makepen` c, where c is a closed `path`; the key points of c become the vertices of the pen. Pens themselves can be transformed.

picture A `picture` is a data type that can be used to store a sequence of META drawing commands; the result of a complete META program is often built up from the interaction of a set of pictures. The meaning of $v + w$ in METAFONT, for example, is a picture in which each pixel is the sum of the two pixels occupying the same position in pictures v and w, respectively.

transform Affine transforms are the natural transformations of Euclidean geometry—that is, the linear transformations augmented by translation. META can construct any affine transform and provides seven primitive ones [72, p. 141]: *shifted*, *scaled*, *xscaled*, *yscaled*, *slanted*, *rotated*, and *zscaled*. The effect of most of the operations is self-evident; the last one, *zscaled*, uses a pair of numbers, interpreted as a complex number in Cartesian coordinates (i.e., complex multiplication).

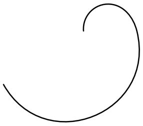
Finally, META is famous for its ability to solve linear equations, including equations that involve points. In particular, you can define a point in terms of other points. For example, `z3=1/2[z1,z2]` defines `z3` as the point in the middle of the line from `z1` to `z2`.

3.1.1 First examples of META programs

Let us first look at some examples of META code, all drawn using METAPOST. You should have little difficulty making these examples run under METAFONT as well, except that you may encounter problems with high-resolution output devices, as METAFONT can run out of memory when composing large pictures—remember that METAFONT generates a bitmap output. This book was typeset at 2400 dpi, and some METAFONT examples were impossible to run at this resolution. Your only recourse is to work at a lower resolution (e.g., 300 dpi) or to break your picture into separate "characters" in a font and join them together in LaTeX. It is almost certainly easier to use METAPOST, as it generates PostScript that can be rendered directly by many printers or turned into PDF.

We do not show the "wrapper" code that is always necessary to turn these examples into a self-contained document. See the notes in Section 3.3.1 on page 68 for information on how METAFONT creates a character and Section 3.3.2 on page 71 for more on how METAPOST creates a figure.

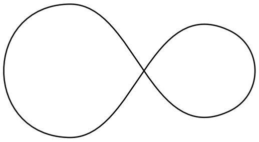
The simplest statement in META is `draw`, which takes a sequence of points separated by `..` and connects them with curves:



```
draw (0,0)..(50,20)..(40,30)..(30,20);
```

Example 3-1-1

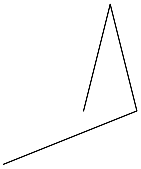
The default unit here is a PostScript point (1/72 inch, TeX's "big point"). To close a object smoothly between its last and first points, the sequence can be terminated by `cycle`:



```
draw (0, 50)..(0,0)..
  (60,40)..(60,10)..cycle;
```

Example 3-1-2

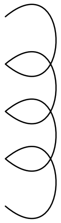
Straight lines are drawn by putting `--` instead of `..` between the points (the lines are actually implemented as specially constrained curves):



```
draw (0,0)--(50,20)--(40,60)--(30,20);
```

Example 3-1-3

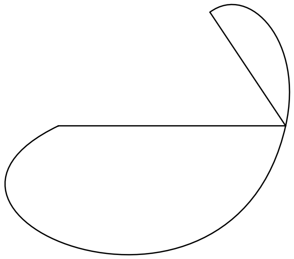
There are several ways of controlling curves: one can vary the angles at the start and end of the curve with `dir`, the points that are to be the extremes (the upmost, the leftmost, and so forth), and the inflection of the curve (with `tension` and `curl`). Thus the following code draws a crude coil by judicious use of dir. Instead of the default units, we express all dimensions in terms of a unit of 2.5 cm, defined at the start:



```
u=2.5cm;
path p;
p= (0,0) {dir 130}..
   {dir -130}(0.25u,0){dir 130}..
   {dir -130}(0.5u,0){dir 130}..
   {dir -130}(0.75u,0){dir 130}..
   {dir -130}(u,0);
draw p rotated -90;
```

Example 3-1-4

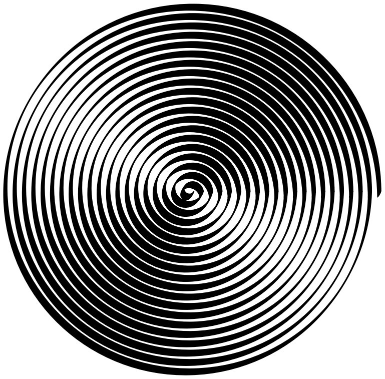
The next example shows the effect of curl. Here a straight line is drawn between three points and then a curve is drawn between the same points, with curl values:



```
path p,q;
u=.5cm;
q=(0u,0u)--(6u,0u)--(4u,3u);
draw q;
p=(0u,0u){curl 4000}..(6u,0u)
  ..{curl 4000}(4u,3u);
draw p;
```

Example 3-1-5

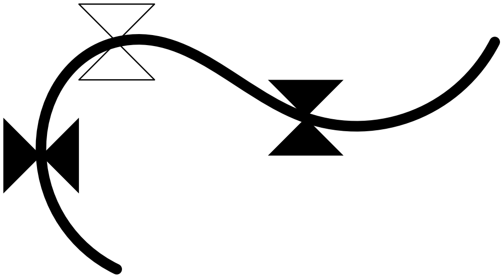
To demonstrate META's unusual "pens", we approximate a spiral drawn with a strange "nib". A colored version of this drawing appears in Color Plate I(a).



```
pickup pencircle scaled 3pt
   yscaled .2pt rotated 60;
n:=5;
for i := (n*20) step -(n) until (n):
 draw ((i,0)..(0,i)..(-i,0)
      ..(0,-(i-n))..(i-n,0)) scaled 0.7;
endfor
```

Example 3-1-6

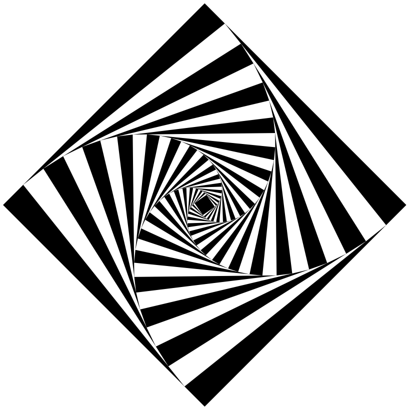
A very characteristic technique with META is creating a path and then using it several times with different transformations. The following code is an extract from a drawing of a kite's tail. Note that shapes can be made solid by using `fill` instead of `draw`:



```
u=1cm;
path p[];
p1:=(.5u,.5u)--(1.5u,.5u)--(.5u,1.5u)
 --(1.5u,1.5u)--(.5u,.5u)--cycle;
fill (p1 shifted (0,2.5u))
 rotatedaround ((u,3.5u),90);
draw p1 shifted (u,4u);
fill p1 shifted (3.5u,3u);
p2 =(2u,2u)..(u,3.5u)..(2u,5u)
 ..(4.5u,4u)..(7u,5u);
pickup pencircle scaled 4pt;
draw p2;
```

Example 3-1-7

A more complicated picture, courtesy of Alan Hoenig from his book *TEX Unbound* [49], demonstrates looping commands. Boxes of gradually decreasing size are drawn alternately white and black, with each one being rotated slightly with respect to the previous box.



```
boolean timetofillbox; timetofillbox := true;
partway := 0.9; l := .45in; u := 1.05in;
n := 4; theta := 360/n; z1 = (0,u);
for i := 2 upto n:
  z[i] = z1 rotated ((i-1)*theta);
endfor
forever:
  path p;    p := z1
  for j := 2 upto n: --z[j] endfor  --cycle;
  if timetofillbox:
    fill p;  timetofillbox := false;
  else:
    unfill p;  timetofillbox := true;
  fi
  pair Z[];
  for j := 1 upto n:
    Z[j] := partway[z[j-1],z[j]];
  endfor
  Z1 := partway[z[n],z1];
  for j := 1 upto n:
    x[j] := xpart Z[j]; y[j] := ypart Z[j];
  endfor
  if not timetofillbox: l := abs(z1); fi
  exitif l < .05u;
endfor
```

Example 3-1-8

CHAPTER 4

METAPOST Applications

4.1 A drawing toolkit 141
4.2 Representing data with graphs 157
4.3 Diagrams 176
4.4 Geometry 189
4.5 Science and engineering applications 196
4.6 3-D extensions 207

Chapter 3 gave a general overview of METAFONT and METAPOST, as well as an extensive description of two multipurpose structuring packages, **boxes** and METAOBJ. However, as is the case for LaTeX, solutions to many problems can often be found by using existing high-level packages. Sometimes several different METAPOST packages are aimed at the same tasks, and these packages come with both advantages and drawbacks.

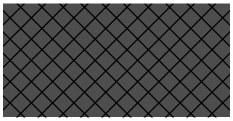
Unfortunately, the perfect package is seldom at hand. It is therefore useful to have a general idea of what can be achieved in METAPOST, and to have some kind of toolbox for problem solving. Understanding a number of basic tricks will enable the beginner to supplement existing packages and achieve the desired results.

In this chapter, we start with a review of a number of basic problems and show how these problems can be solved. Then we describe some standard applications of METAPOST, ranging from geometry to physics.

4.1 A drawing toolkit

This section is devoted to a number of advanced features, which are located somewhere between low-level METAPOST code and full application packages. We like to consider all these features as a kind of toolkit, which can be used with benefit in wider applications.

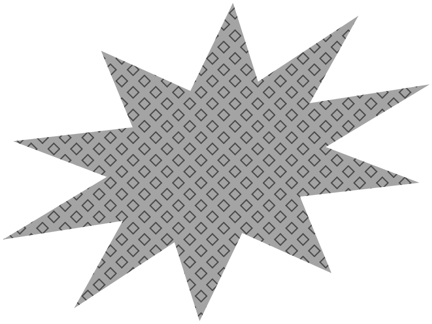
Bogusław Jackowski's hatching package provides a more elaborate way to achieve hatching patterns, by redefining the withcolor primitive in such a way that it represents hatching parameters when the blue component of the color is negative. The following examples illustrate this principle.



Example 4-1-10

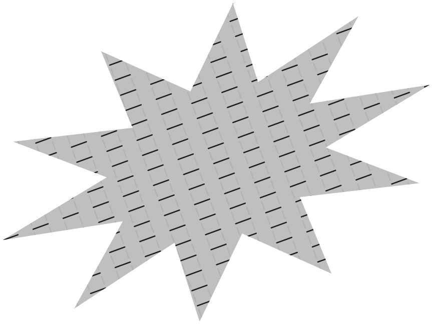
```
input hatching;
path p;
p:=unitsquare xscaled 30mm yscaled 15mm;
hatchfill p withcolor red
            withcolor (45,2mm,-.5bp)
            withcolor (-45,2mm,-.5bp);
```

The next three examples use a special closed path shaped as a star, defined by the star macro:



Example 4-1-11

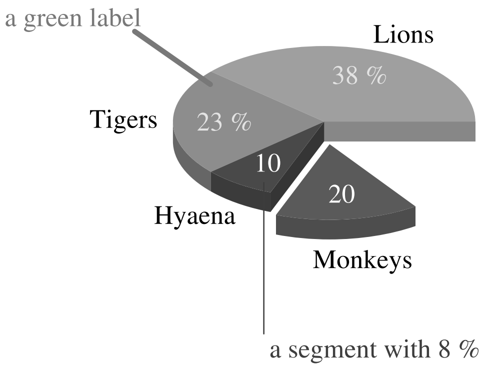
```
input hatching;
vardef star(expr n) =
 for i_:=0 upto 2n-1:
  if odd i_: 1/2 fi (right rotated (180*(i_/n))) --
 endfor cycle
enddef;
interim hatch_match:=0;
path p;
p:=star(10) xscaled 30mm
            yscaled 20mm
            rotated 20;
hatchfill p withcolor (0,1,.5);
draw image(hatchfill p
           withcolor (45,3bp,-.5bp)
           withcolor (-45,3bp,-.5bp);
      ) withcolor red dashed evenly;
```



Example 4-1-12

```
input hatching;
% star macro defined as above
path p;
p:=star(10) xscaled 30mm
            yscaled 20mm
            rotated 20;
interim hatch_match:=0;
hatchoptions(withcolor blue
             dashed evenly scaled 2);
hatchfill p withcolor .75white
            withcolor (20,6bp,-.5bp);
hatchoptions(withcolor (blue+green)
             dashed evenly
             shifted (3/2bp,0));
hatchfill p withcolor (110,6bp,-.5bp);
```

A more elaborate example appears below. The 8% corresponds to 10 being 8% of 50 + 30 + 10 + 20 + 20.



```
input piechartmp
SetupColors((.7,.7),this,this);
SetupPercent(this, " %");
Segment(50,"Lions");  Segment(30,"Tigers");
Segment(10,"Hyaena"); Segment(20,"Monkeys");
Segment(20,"Warthogs");
SegmentState(4,this,0.3);
SegmentState(5,invisible,this);
PieChart(2cm,0.15,60,0,0);
Label.auto(0)(name)(outwards,0);
Label(3,4,5)(value)(inwards,0) withcolor white;
Label(1,2)(percent)(inwards,0) withcolor (1,1,0);
Label.lrt(3)("a segment with ",percent)
            ((0.9,0.8),(0,-2cm)) withcolor .8red;
pickup pencircle scaled 2pt;
Label.auto(2)("a green label")
        ((0.9,0.1),(-1cm,7mm)) withcolor .8green;
```

Example 4-2-26

This example has labels with spaces and needs a font with spaces—hence the `defaultfont` declaration. This is not a problem when we are using TEX labels.

`SetupNumbers`(*precision*,*delimiter*)

Setup commands

In addition to the `SetupPercent` commands, several other setup commands are available. The first, `SetupNumbers`, sets the accuracy and delimiter used. `SetupNumbers(2,",")` will, for instance, round at two places and use a comma delimiter.

`SetupColors`(*auto-SV*,*shading-SV*,*grayscale*)

This command specifies the colors used for segments. The three arguments are as follows:

auto-SV is a pair (S, V), where S is the saturation and V is the value in the HSV model. The hue H is taken from the position of the segment.

shading-SV is a pair giving the maximum values of (S, V) for shaded areas in segments. The default is (0.4,0.3).

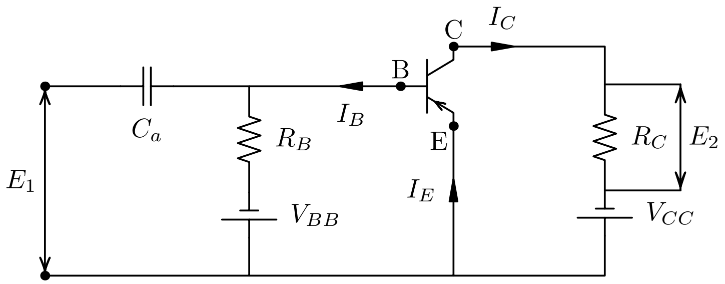
grayscale is a Boolean that, when set to `true`, switches the colors to grayscale.

`SetupText`(*Mode*,*TeXFormat*,*TeXSettings*)

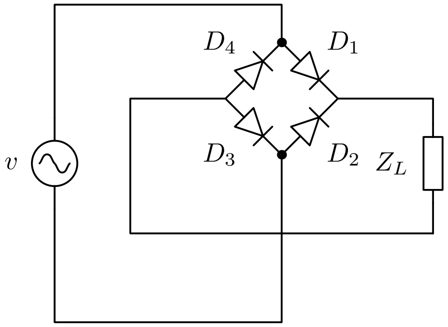
This command sets up how text is handled, using three arguments:

Mode is an integer specifying the way labels are typeset: 0 is for string-based typesetting (default); 1 is for external TEX-based typesetting using *TeXFormat* and *TeXSettings*; 2 is

```
ctext.rt(R.C.l+(1cm,0),R.C.r+(1cm,0),"$E_2$",witharrow);
```



Example 4-5-7

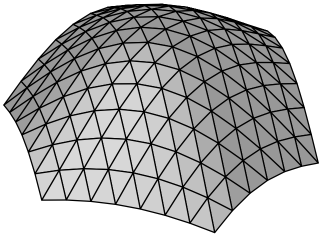


```
input makecirc;
initlatex("\usepackage{amsmath,amssymb}");
source.a(origin,AC,90,"v","");
junction.a(S.a.p+(3cm,1cm),"")(top);
diode.a(J.a,normal,-45,pinA,"D_1","");
diode.b(D.a.K,normal,-135,pinK,"D_2","");
diode.c(D.b.A,normal,135,pinK,"D_3","");
diode.d(D.c.A,normal,45,pinA,"D_4","");
junction.b(D.b.A,"")(bot);
centerto.A(S.a.n,S.a.p)(5cm,imp);
impedance.a(A,90,"Z_L","");
wireU(S.a.p,D.a.A,1.5cm,udsq);
wireU(S.a.n,D.b.A,-1.5cm,udsq);
wire(D.a.K,Z.a.r,rlsq);
wire(Z.a.l,Z.a.l+(0,-4mm),nsq);
wireU(Z.a.l+(0,-4mm),D.d.A,-4cm,rlsq);
```

Example 4-5-8

```
input makecirc;
initlatex("\usepackage{amsmath,amssymb}");
transformer.a(origin,mid,0);
diode.a(tf.a.ss+(5mm,1cm),normal,0,pinA,"D_1","");
diode.b(tf.a.si+(5mm,-1cm),normal,0,pinA,"D_2","");
impedance.a(D.a.K+(2cm,-4mm),-90,"Z_L","300\ohm");
wire(tf.a.ss,D.a.A,udsq);wire(tf.a.si,D.b.A,udsq);
wire(D.a.K,Z.a.l,rlsq);wire(Z.a.r,tf.a.m,udsq);
wire(D.b.K,D.a.K+(5mm,0),rlsq);
junction.a(D.a.K+(5mm,0),"")(top);
centerto.A(tf.a.pi,tf.a.ps)(-15mm,sac);
source.a(A,AC,90,"220 V","v");
wire(S.a.p,tf.a.ps,udsq);wire(S.a.n,tf.a.pi,udsq);
centreof.A((xpart S.a.p,ypart tf.a.ps),tf.a.ps,cur);
current.a(c.A,phi.A,"i(t)","5 A");
imesh(tf.a.ss+(1cm,0),15mm,1cm,cw,0,"I_{cc}");
```

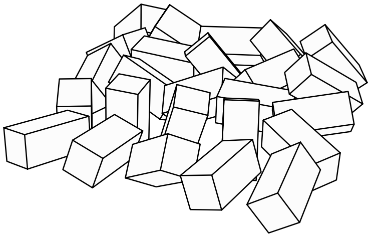
hexagonal meshes Given a function $z = f(x, y)$, a hexagonal mesh can be obtained with the `hexagonaltrimesh` macro.



```
input featpost3Dplus2D
def zsurface( expr xc, yc ) =
        cosd(xc*57)*cosd(yc*57)
        +4*mexp(-(xc**2+yc**2)*6.4)
enddef;

f := 7*(4,1,5);
Spread := 35;
LightSource := 10*(4,-3,4);
SubColor := 0.4background;

numeric np, ssize;
path chair;
np = 20;
ssize = 5;

hexagonaltrimesh( true,np,ssize,zsurface);
```

Example 4-6-2

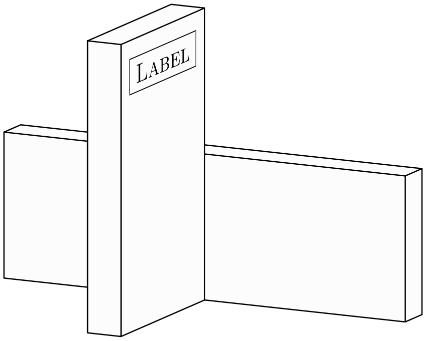
cubes The `kindofcube` macro produces a cube in an orientation depending on its parameters. In this example, each cube erases what has been drawn under it, so that it gives the illusion of the removal of hidden parts.



```
input featpost3Dplus2D
Spread := 30;
f := 5.4*(1.5,0.5,1);
numeric gridstep, sidenumber,
        i, j, coord, aa, ab, ac;
color pa;
gridstep = 0.7;
sidenumber = 4;
coord = 0.5*sidenumber*gridstep;
for i=0 upto sidenumber:
  for j=0 upto sidenumber:
    pa := (-coord+j*gridstep,-coord+i*gridstep,0);
    aa := uniformdeviate(360);
    ab := uniformdeviate(180);
    ac := uniformdeviate(90);
    kindofcube(false, false,
              pa, aa, ab, ac, 0.4, 0.4, 0.9 );
  endfor;
endfor;
```

Example 4-6-3

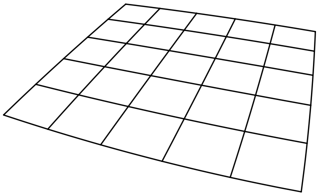
labels in space The next example shows how labels can be drawn in space using the `labelinspace` macro.



Example 4-6-4

```
input featpost3Dplus2D
verbatimtex
%&latex
\documentclass{article}
\begin{document}
etex

f := 1.1*(2,1,0.5);
ParallelProj := true;
kindofcube(false,true,(0,-0.5,0),
           90,0,0,1.2,0.1,0.4);
kindofcube(false,true,(0,0,0),
           0,0,0,0.5,0.1,0.8);
labelinspace(false,(0.45,0.1,0.65),
             (-0.4,0,0),(0,0,0.1),
    btex \framebox{\textsc{Label}} etex);
```

projected segments The last example shows how points can be defined in space, and `pathofstraightline` used to draw a segment joining the projections of these points.



Example 4-6-5

```
input featpost3Dplus2D
SphericalDistortion := true;
Spread := 50;
f := 0.4*(1.5,0.5,1);
numeric gridstep, sidenumber, i, coord;
color pa, pb, pc, pd;
gridstep = 0.1;
sidenumber = 5;
coord = 0.5*sidenumber*gridstep;
for i=0 upto sidenumber:
  pa := (-coord,-coord+i*gridstep,0);
  pb := (coord,-coord+i*gridstep,0);
  pc := (-coord+i*gridstep,-coord,0);
  pd := (-coord+i*gridstep,coord,0);
  draw pathofstraightline( pa, pb );
  draw pathofstraightline( pc, pd );
endfor;
```

3DLDF

Laurence D. Finston's ambitious extension to METAPOST, 3DLDF (`http://www.gnu.org/software/3dldf/LDF.html`) is written in C++ using CWEB. 3DLDF (the author's initials) takes an input similar to METAPOST and outputs pure METAPOST code. The package currently computes the intersections of various projected curves, and the author plans to implement the removal of hidden parts.

CHAPTER 5

Harnessing PostScript Inside LaTeX: PSTricks

5.1 The components of PSTricks 214
5.2 Setting keywords, lengths, and coordinates 217
5.3 The pspicture environment 220
5.4 The coordinate system 223
5.5 Grids 224
5.6 Lines and polygons 231
5.7 Circles, ellipses, and curves 240
5.8 Dots and symbols 249
5.9 Filling areas 253
5.10 Arrows 259
5.11 Labels 265
5.12 Boxes 269
5.13 User styles and objects 279
5.14 Coordinates 296
5.15 The PSTricks core 302

As we saw in Chapter 1, one way of drawing graphics with LaTeX is to embed low-level picture drawing primitives for the target device into LaTeX macros, so that full typesetting information is available and we can work in a familiar macro programming environment. When the target device is something as rich as the full PostScript language, this can result in a very powerful system. While many macro packages have implemented access to some parts of PostScript for this purpose, the most complete is undoubtedly PSTricks. In the next two chapters, we survey its capabilities and demonstrate some of the power that results from combining LaTeX and PostScript.

We do not attempt to describe absolutely every PSTricks-related macro, nor do we give examples of all the possible combinations and tricks, as this would require a large book of its own, e.g., [135]. We have, however, tried to describe and give examples of all the important features of the basic packages. You'll find a lot of useful information on the official PSTricks Web site at `http://PSTricks.tug.org/`.

Because there are a great many commands and especially keywords in PSTricks, we provide a summary description at the end of the next chapter (Section 6.8 on page 459). PSTricks and its related packages are extremely powerful, and their facilities may take some time to understand. It is also documented in the individual packages and [127, 135], and its implementation is described in [126].

5.1 The components of PSTricks

The PSTricks project was started by Timothy Van Zandt a long time ago and is one of the oldest TEX packages still in use.

> I started in 1991. Initially I was just trying to develop tools for my own use. Then I thought it would be nice to package them so that others could use them. It soon became tempting to add lots of features, not just the ones I needed. When this became so interesting that it interfered with my "day job", I gave up the project "cold turkey", in 1994.
>
> [Timothy Van Zandt]

After Timothy Van Zandt stopped working on the project, Denis Girou took over the task to care for PSTricks, mainly fixing bugs and writing some more new packages; nowadays this job is done by Herbert Voß. Several developers are working on existing and new packages, which is the reason why the number of these additional packages, which depend on the basic PSTricks, is still increasing. A selection of them is discussed in Chapter 6, and the full list is available at the official Web site at `http://PSTricks.tug.org`.

5.1.1 The kernel

The basic PSTricks package file is `pstricks.tex`, which provides the basic unit handling, and basic graphic macros like dots, lines, frames, and so on. For some historical reason the packages pstricks, pst-plot, pst-node, and pst-tree build the core of PSTricks and are all available on CTAN in the directory `CTAN:/graphics/pstricks/base/generic/`. Each PSTricks package has a corresponding LATEX style file, and the basic ones are stored in `CTAN:/graphics/pstricks/base/latex/`. In general, the style files do nothing other than load the TEX file via the `\input` macro.

The basic PSTricks packages consist of a core of picture-drawing primitives implemented by `\special` commands that pass PostScript code to a driver, mainly dvips. The packages also contain a set of higher-level macros for particular applications, like pst-plot or pst-node. With it you can

- Draw lines, polygons, circles, and curves.
- Place and manipulate TEX text.

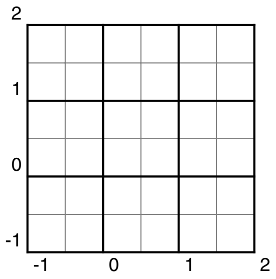
\psgrid[*settings*] (x_0, y_0) (x_1, y_1) (x_2, y_2)

The \psgrid macro is a very powerful tool for drawing coordinate grids. The syntax is easy to use, but is valid only for Cartesian coordinate systems.

When no coordinates have been specified, \psgrid takes the ones defined by the enclosing pspicture environment or, if not inside such an environment, a 10×10 rectangle in the current units is assumed. If only one coordinate pair is given, it is taken to denote one corner and (0,0) is established as the opposite corner. When using two coordinate pairs, any two opposite corners of the grid should be specified. With three coordinate pairs given, the first pair determines the intersection point of the lines to be labeled and the other two pairs are interpreted as in the previous case.

In short: (x_0, y_0) defaults to (x_1, y_1); the default for the latter is (0,0), and (outside of a pspicture environment) the default for (x_2, y_2) is (10,10).

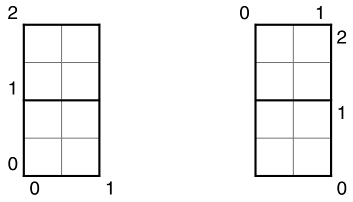
The labels are positioned along the two lines that intersect at (x_0, y_0), on the side of the line pointing away from (x_2, y_2), and shifted slightly horizontally or vertically towards the latter coordinate so they won't interfere with other lines. In the next example, \psgrid has no arguments, so it takes all coordinates from the surrounding pspicture environment. The keywords used in this and the following examples are discussed in detail in Section 5.5.1 on the following page.



Example 5-5-1

```
\usepackage{pstricks}
\psset{griddots=0,gridlabels=7pt,subgriddiv=2}
\begin{pspicture}(-1,-1)(2,2)
  \psgrid
\end{pspicture}
```

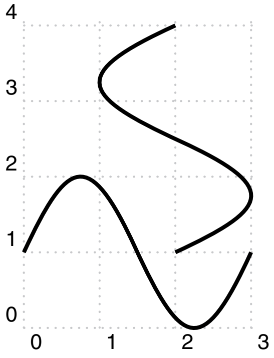
With only one pair of coordinates, \psgrid assumes that (0,0) is the opposite corner. Exchanging the order of the coordinate pairs, as in the second figure, changes the position of the labels from the left and bottom sides to the right and top sides of the rectangle, respectively. (See also the last example below with three pairs of coordinates.)



Example 5-5-2

```
\usepackage{pstricks}
\begin{pspicture}(-1,-1)(2,2)
  \psgrid[griddots=0,gridlabels=7pt,subgriddiv=2](1,2)
\end{pspicture}
\begin{pspicture}(-1,-1)(2,2)
  \psgrid[griddots=0,gridlabels=7pt,
          subgriddiv=2](1,2)(0,0)
\end{pspicture}
```

This is also demonstrated in the next example.



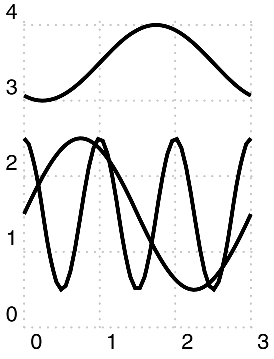
```
\usepackage{pstricks,pst-plot}
\begin{pspicture}[showgrid=true](3,4)
  \pscustom[linewidth=1.5pt]{%
    \translate(0,1)
    \psplot{0}{3}{x 180.0 mul 1.5 div sin}
    \translate(2,0)
    \swapaxes
    \psplot[liftpen=2]{0}{3}{x 180.0 mul 1.5 div sin}}
\end{pspicture}
```

Example 5-13-18

`\msave` `\mrestore`

With this pair of macros, the currently valid coordinate system may be saved and restored, respectively. In contrast to what happens with `\gsave` and `\grestore` pairs, all other values such as line type, thickness, etc., will remain unaffected. The `\msave` and `\mrestore` commands must be used in pairs! They can be nested arbitrarily both with themselves and with `\gsave` and `\grestore`. Care must be taken to ensure that this nesting is pairwise balanced.

The next example plots the first sine function with the origin of ordinates set by `\translate(0,1.5)`. Thereafter, the state of the coordinate system is saved, a new origin is set with `\translate(1,2)`[1], and another sine function is plotted. Following that, the old state is restored with `\mrestore` and the origin of ordinates is back at `(0,1.5)` again. The later cosine function is plotted with this origin.



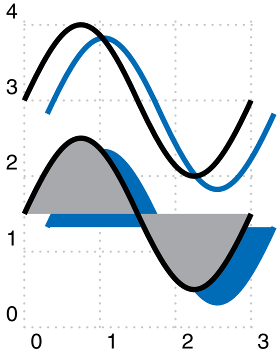
```
\usepackage{pstricks,pst-plot}
\begin{pspicture}[showgrid=true](3,4)
  \pscustom[linewidth=1.5pt]{%
    \translate(0,1.5)
    \psplot{0}{3}{x 180.0 mul 1.5 div sin}
    \msave
      \translate(1,2)
      \scale{1 0.5}
      \psplot[liftpen=2]{-1}{2}{x 180.0 mul 1.5 div sin}
    \mrestore
    \psplot[liftpen=2]{0}{3}{x 180.0 mul 0.5 div cos}}
\end{pspicture}
```

Example 5-13-19

[1] Referring to the current origin `(0,1.5)` a `\translate(1,2)` corresponds to the absolute coordinates `(1,3.5)`.

\openshadow [*settings*]

The \openshadow command creates a copy of the current path, using the specified shadow key values (see page 239). Whether the shadow path thus obtained is stroked or filled depends on the parameter settings supplied with \openshadow itself and/or \pscustom, as can be seen in the example.

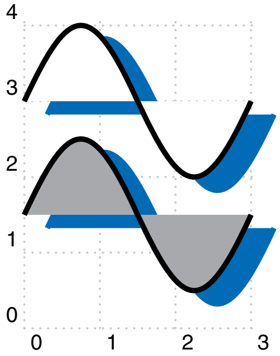


```
\usepackage{pstricks,pst-plot}
\begin{pspicture}[showgrid=true](3,4)
  \pscustom[linewidth=2pt]{%
    \translate(0,3)
    \psplot{0}{3}{x 180.0 mul 1.5 div sin}
    \openshadow[shadowsize=10pt,shadowangle=-30,
                shadowcolor=blue]}
  \pscustom[linewidth=2pt,fillcolor=red,
            fillstyle=solid]{%
    \translate(0,1.5)
    \psplot{0}{3}{x 180.0 mul 1.5 div sin}
    \openshadow[shadowsize=10pt,shadowangle=-30,
                shadowcolor=blue]}
\end{pspicture}
```

Example 5-13-20

\closedshadow [*settings*]

The \closedshadow command *always* creates a filled shadow of the region enclosed by the current path, as if it were a non-transparent environment.

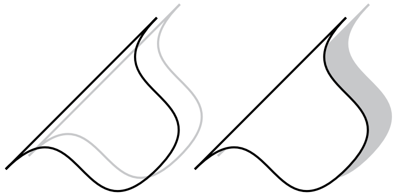


```
\usepackage{pstricks,pst-plot}
\begin{pspicture}[showgrid=true](3,4)
  \pscustom[linewidth=2pt]{%
    \translate(0,3)
    \psplot{0}{3}{x 180.0 mul 1.5 div sin}
    \closedshadow[shadowsize=10pt,shadowangle=-30,
                  shadowcolor=blue]}
  \pscustom[linewidth=2pt,fillcolor=red,
            fillstyle=none]{% <-- no effect!
    \translate(0,1.5)
    \psplot{0}{3}{x 180.0 mul 1.5 div sin}
    \closedshadow[shadowsize=10pt,shadowangle=-30,
                  shadowcolor=blue]}
\end{pspicture}
```

Example 5-13-21

The method used for producing the shadow should be noted. PSTricks simply creates a copy of the closed path, translates it according to the demands of shadowsize and shadowangle, fills it with shadowcolor, and then refills the original path with fillcolor, which is white by default. The \openshadow macro doesn't fill the original path with the current fillcolor, so that the underlying shadow copy is visible (and in this example, not filled). The \closedshadow fills the original path, so that the underlying copy looks like a real shadow.

```
\usepackage{pstricks}
\begin{pspicture}(0,-0.25)(5,2)
\pscustom[fillstyle=none,shadowcolor=lightgray,fillcolor=blue]{%
  \psbezier(0,0)(1,1)(1,-1)(2,0) \psbezier(2,0)(3,1)(1,1)(2,2)
  \closepath
  \openshadow[shadowsize=10pt,fillcolor=white,shadowangle=30]}
\rput(2.5,0){%
\pscustom[fillstyle=none,shadowcolor=lightgray,fillcolor=blue]{%
  \psbezier(0,0)(1,1)(1,-1)(2,0) \psbezier(2,0)(3,1)(1,1)(2,2)
  \closepath
  \closedshadow[shadowsize=10pt,fillcolor=white,shadowangle=30]}}
\end{pspicture}
```

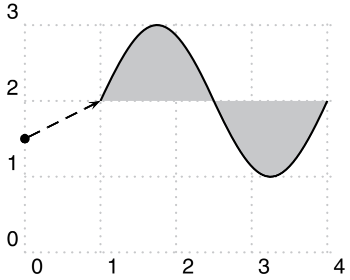


Example 5-13-22

This strategy is to be kept in mind when specifying, with the keyword \pscustom, a fillcolor that differs from white: in such cases the macro \closedshadow has to be given the correct fill color.

\movepath(dx, dy)

The \movepath command shifts the current path by (dx, dy). If the original path is needed later on, the \movepath operation has to be encapsulated within a \gsave/\grestore pair.



```
\usepackage{pstricks,pst-plot}
\begin{pspicture}[showgrid=true](4,3)
  \pscustom[fillcolor=lightgray,fillstyle=solid]{%
    \translate(0,1.5)
    \psplot{0}{3}{x 180.0 mul 1.5 div sin}
    \movepath(1,0.5)}
  \psline[linestyle=dashed]{*->}(0,1.5)(1,2)
\end{pspicture}
```

Example 5-13-23

CHAPTER 6

The Main PSTricks Packages

6.1 pst-plot—Plotting functions and data 313
6.2 pst-node—Nodes and connections 334
6.3 pst-tree—Typesetting trees 366
6.4 pst-fill—Filling and tiling 383
6.5 pst-3d—Shadows, tilting, and three-dimensional representations 388
6.6 pst-3dplot—3-D parallel projections of functions and data 400
6.7 Short overview of other PSTricks packages 417
6.8 Summary of PSTricks commands and keywords 459

The “main” packages of PSTricks nowadays have this name only for historical reasons. PSTricks is used for those packages listed in the pst-all package. We do not follow this list here. Instead, we describe the most common ones (e.g., pst-plot, pst-node) in some detail. Section 6.7 then gives an overview of other packages, showing at least one characteristic example to help you understand the purpose of each package and approach that it takes.

6.1 pst-plot—Plotting functions and data

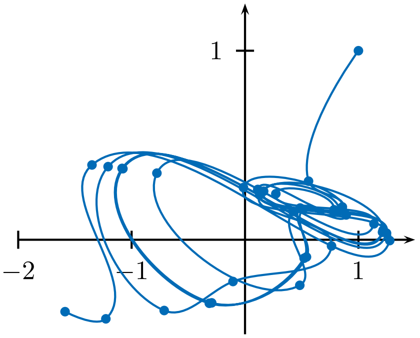
The base package pstricks provides some macros to plot function values and coordinates, as listed in Table 6.1. All of these macros accept an arbitrary number of coordinate pairs as arguments.

The pst-plot package provides improved commands for plotting external data and functions as well as coordinate axes [59, 60, 131]. It supports only two-dimensional data pairs. For plotting (x, y, z) data triplets or three-dimensional functions, you can use the pst-3dplot package discussed in Section 6.6, which supports a parallel projection of 3-D objects [132, 134].

\listplot

In contrast to the preceding plot commands, the argument of `\listplot` is first expanded if it contains TEX macros; otherwise, it is passed to PostScript without change. In the process, TEX macros are replaced with their corresponding replacement text. It is possible to include entire PostScript programs in the argument to `\listplot`, as shown in Example 6-1-33.

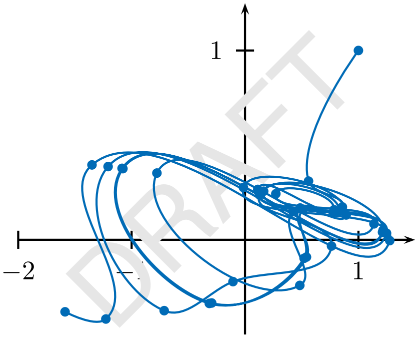
The first example illustrates the Hénon attractor.



```
\usepackage{pstricks,pst-plot}
% definition of \henon with data points like this:
% \newcommand\henon{ 1.00000000  1.00000000
%                    0.56000000  0.31000000
%                    ... many more ...}

\psset{xunit=1.5cm, yunit=2.5cm}
\begin{pspicture}(-2,-0.5)(1.5,1.25)
  \psaxes{->}(0,0)(-2,-0.5)(1.5,1.25)
  \listplot[showpoints=true,plotstyle=curve,
    linecolor=blue]{\henon}
\end{pspicture}
```

Example 6-1-32

The second example includes the watermark “DRAFT”, which was added to the original data with additional PostScript code.



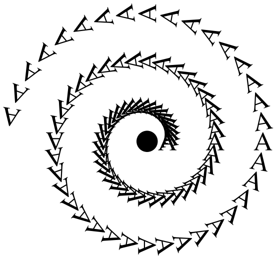
```
\usepackage{pstricks,pst-plot}
% \henon as in previous example
\newcommand{\dataA}{\henon
  gsave
  /Helvetica findfont 40 scalefont setfont
  45 rotate
  0.9 setgray
  -60 10 moveto (DRAFT) show
  grestore }
\psset{xunit=1.5cm, yunit=2.5cm}
\begin{pspicture}(-2,-0.5)(1.5,1.25)
  \psaxes{->}(0,0)(-2,-0.5)(1.5,1.25)
  \listplot[showpoints=true,linecolor=blue,
        plotstyle=curve]{\dataA}
\end{pspicture}
```

Example 6-1-33

Instead of modifying the data set passed to `\listplot`, you can redefine the `\ScalePoints` macro in pst-plot. For example, if you wanted to exchange the x and y val-

It works only in conjunction with the \nput command (see page 359).

Example 6-2-54



```
\usepackage{pstricks,pst-node,multido}
\begin{pspicture}(4.5,4.5)
  \cnode*(2,2){4pt}{A}
  \multido{\nA=0+10,\rB=0+0.5}{90}{%
     \nput[rot=\nA,%
        labelsep=\rB pt]{\nA}{A}{A}}
\end{pspicture}
```

6.2.5 Putting labels on node connections

In Section 5.11 on page 265, we already discussed several commands that allow arbitrary placement of marks with respect to labels. In the context of connections, there are some special commands to consider. After a connection has been drawn, the coordinates of two points are stored temporarily until a new connection is drawn. This data may prove very useful for positioning the labels to be attached to such a connection. Of course, it also implies that label commands should come immediately after connection commands.

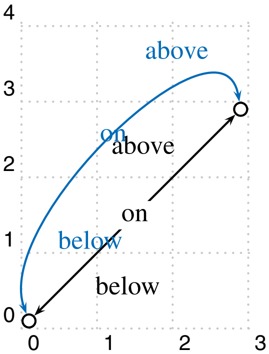
In Section 6.2.4 on page 348, which discussed the allowed keywords, you will find many examples of the placement of labels. In this section we will review the various commands once again.

```
\ncput* [settings] {object}  \naput* [settings] {object}  \nbput* [settings] {object}
```

n labels

The n label commands are always based on the visible length of a connection, without attention to the actual node centers. By default, the label is placed in the middle of this visible connection, which can be changed with the appropriate keyword. The letter c indicates *connected* (on the line), and a and b indicate *above* and *below* the line, respectively. The starred versions produce opaque material, which means you can overwrite lines with a label to gain increased visibility.

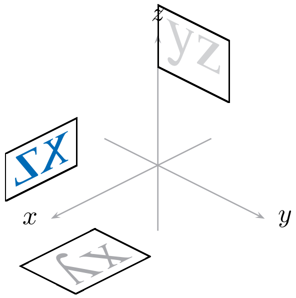
Example 6-2-55



```
\usepackage{pstricks,pst-node}
\begin{pspicture}[showgrid=true](3,4)
  \cnode(0.1,0.1){0.1cm}{A}  \cnode(2.9,2.9){0.1cm}{B}
  \ncline{<->}{A}{B} \ncput*{on}
  \naput[npos=0.75]{above} \nbput[npos=0.25]{below}
  \nccurve[angleA=110,angleB=100,
    linecolor=blue]{<->}{A}{B}
  \ncput{\textcolor{blue}{on}}
  \naput[npos=0.75]{\textcolor{blue}{above}}
  \nbput[npos=0.25]{\textcolor{blue}{below}}
\end{pspicture}
```

The pOrigin *key*

The keyword pOrigin is the positioning key, which is passed to the command \rput. Its effects concern only \pstThreeDPut, and the default value is based on the defaults for \rput (see Section 5.11.1 on page 266).



```
\usepackage{pstricks,pst-3dplot}
\begin{pspicture}(-2,-1)(1,2.5)
  \pstThreeDCoor[xMin=-1,xMax=2,yMin=-1,
                 yMax=2,zMin=-1,zMax=2]
  \pstPlanePut[pOrigin=c](0,0,-1){\fbox{\Huge\red xy}}
  \pstPlanePut[plane=xz,pOrigin=rb](0,0,0)
              {\fbox{\Huge\blue xz}}
  \pstPlanePut[plane=yz,pOrigin=lb](0,0,1.5)
              {\fbox{\Huge\green yz}}
\end{pspicture}
```

Example 6-6-28

The hiddenLine *key*

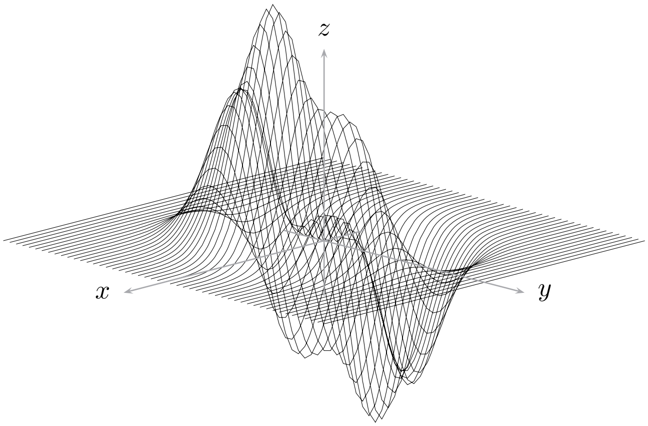
The keyword hiddenLine enables a very simple “hidden-line algorithm”: the lines are plotted with the command \pscustom and then filled with the predefined fill style hiddenStyle.

```
\newpsstyle{hiddenStyle}{fillstyle=solid,fillcolor=white}
```

You can overwrite this style as required. Just keep in mind that the curves must be built from the end to the beginning; otherwise, the hidden lines will be visible. For examples, see Section 6.6.2 on page 406.

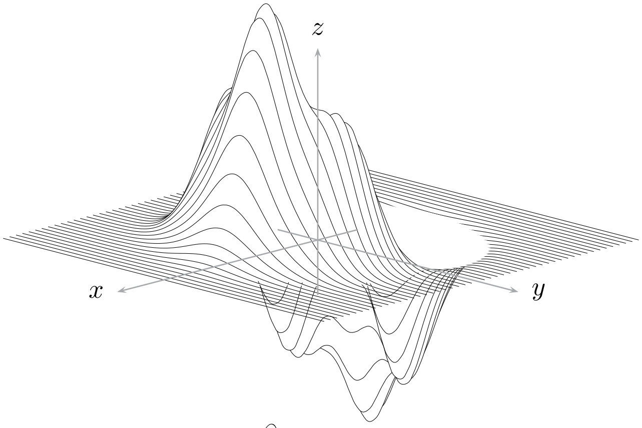
The drawStyle *key*

The keyword drawStyle defines the manner in which the function is plotted. Possible key values are xLines, yLines, xyLines, and yxLines. The values refer to the plotting sequence; that is, xLines has the lines drawn in the x direction, whereas yxLines means that they are first drawn in the y direction and then in the x direction.



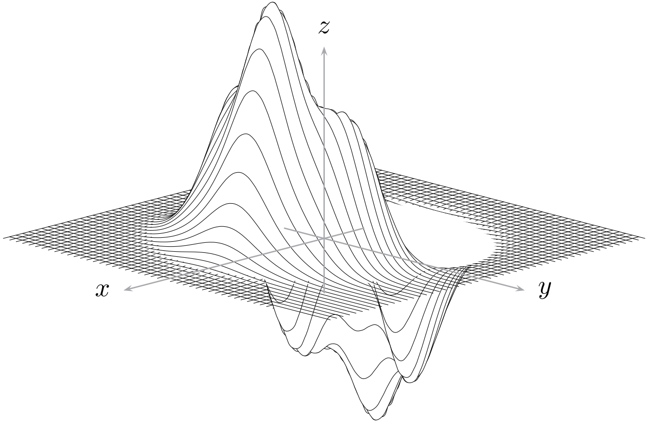
```
\usepackage{pstricks,pst-3dplot}
% \func as defined in Example 6-6-13
\begin{pspicture}(-6,-3)(6,4)
  \psset{Beta=15,unit=0.75}
  \psplotThreeD[plotstyle=line,
     drawStyle=xLines,
     yPlotpoints=50,xPlotpoints=50,
     linewidth=0.2pt](-4,4)(-4,4)
    {\func}
  \pstThreeDCoor[xMax=5,yMax=5,
                 zMax=3.5]
\end{pspicture}
```

Example 6-6-29



```
\usepackage{pstricks,pst-3dplot}
% \func as defined in Example 6-6-13
\begin{pspicture}(-6,-3)(6,4)
  \psset{Beta=15,unit=0.75}
  \psplotThreeD[plotstyle=curve,%
     drawStyle=yLines,%
     hiddenLine=true,%
     yPlotpoints=50,xPlotpoints=50,%
     linewidth=0.2pt](-4,4)(-4,4){\func}
  \pstThreeDCoor[xMax=5,yMax=5,zMax=3.5]
\end{pspicture}
```

Example 6-6-30

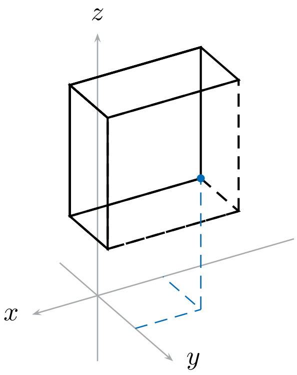


```
\usepackage{pstricks,pst-3dplot}
% \func as defined in Example 6-6-13
\begin{pspicture}(-6,-3)(6,4)
  \psset{Beta=15,unit=0.75}
  \psplotThreeD[%
    plotstyle=curve,drawStyle=xyLines,%
    hiddenLine=true,%
    yPlotpoints=50,xPlotpoints=50,%
    linewidth=0.2pt](-4,4)(-4,4){\func}
  \pstThreeDCoor[xMax=5,yMax=5,zMax=3.5]
\end{pspicture}
```

Example 6-6-31

The visibleLineStyle *and* invisibleLineStyle *keys*

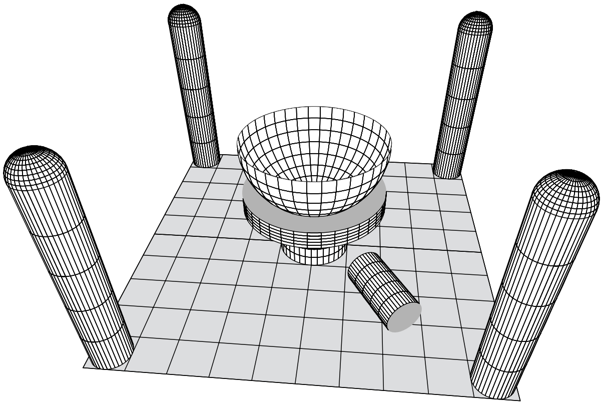
The keywords visibleLineStyle and invisibleLineStyle refer to the drawing of bodies: the macro tries to identify hidden lines and draws them with the line style invisibleLineStyle, while drawing the visible ones with the style visibleLineStyle.



```
\usepackage{pstricks,pst-3dplot}
\begin{pspicture}(-1,-1)(3,3.25)
  \psset{Alpha=30}
  \pstThreeDCoor[xMin=-3,xMax=1,yMax=2,zMax=4]
  \pstThreeDBox(-1,1,2)(0,0,2)(2,0,0)(0,1,0)
  \pstThreeDDot[drawCoor=true,linecolor=blue](-1,1,2)
\end{pspicture}
```

Example 6-6-32

```
\CylindreThreeD(0,0,0){10}{15}          \CylindreThreeD(0,0,15){20}{5}
\DemiSphereThreeD[RotX=180](0,0,35){20}
\SphereCreuseThreeD[RotX=180](0,0,35){20}
{ \psset{RotY=90,RotX=0,RotZ=30}  \CylindreThreeD(15,15,5){5}{20}  }
\multido{\iCY=-45+90}{2}{\CylindreThreeD(45,\iCY,0){5}{50}
                         \DemiSphereThreeD(45,\iCY,50){5}}
\end{pspicture}
```

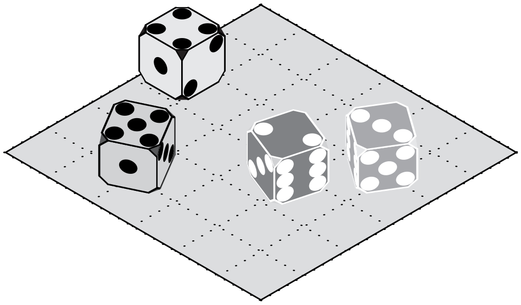


Example 6-7-39

The pst-ob3d package

This package allows you to draw basic three-dimensional objects such as cubes (which can be deformed to rectangular parallelepipeds) and dies. The package author is Denis Girou.

```
\usepackage{pst-ob3d}
\ThreeDput{\psframe[fillstyle=solid,fillcolor=black!15](6,6)
           \psgrid[subgriddiv=0,gridlabels=0,griddots=5](6,6)}
\psset{fillstyle=solid,dotscale=2,RandomFaces=true,Corners=true}
\randomi=123456 \PstDie[fillcolor=black!10](1,3,0)
\randomi=271354 \PstDie[fillcolor=black!20,viewpoint=1  0.3 1,
                        CornersColor=black!80](0.3,1.5,0)
\psset{linecolor=white}
\randomi=93850516 \PstDie[fillcolor=black!60,viewpoint=1 -0.5 1,
                          CornersColor=black!20](3,3,0)
\randomi=8873165  \PstDie[fillcolor=black!40,viewpoint=1 -0.2 1,
                          CornersColor=black!10](2,5,0)
```



Example 6-7-40

CHAPTER 7

The Xy-pic Package

7.1 Introducing Xy-pic . . . 467
7.2 Basic constructs . . . 469
7.3 Extensions . . . 474
7.4 Features . . . 478
7.5 Further examples . . . 509

XY-pic is a general-purpose drawing package based on TEX. It works smoothly with most formats, including LATEX, AMS-LATEX, AMS-TEX, and plain TEX. It has been used to typeset complicated diagrams from numerous application areas, including category theory, automata, algebra, geometry, neural networks, and knot theory. XY-pic's generic syntax lets you use a consistent mnemonic notation system that is based on the *logical* construction of diagrams by the combination of various elementary *visual* components. You can also write macros by combining these basic elements consistently to form higher-level structures specific to the intended application.

XY-pic was originally written by Kristoffer Høgsbro Rose [105]. Later Ross Moore joined the development effort and the ensuing collaboration resulted in extensive revisions and extensions [104, 106].

7.1 Introducing Xy-pic

The XY-pic system is built around an object-oriented drawing language called the *kernel*: this is a notation for composing "objects" with "methods" that correspond to the meaningful drawing operations on the object.

The kernel supports the following basic graphic notions (see Section 7.2):

- *Positions* can be specified in various formats. In particular, user-defined coordinates can be absolute or relative to previous positions, objects, object edges, or points on connections.

- *Objects* can have several forms—e.g., circular, elliptic, and rectangular—and can be adjusted in several ways, even depending on the *direction* of other objects. In particular, an object can be used to *connect* two other objects.

Enhancements to the kernel, called "options", have two main varieties: *extensions* (see Section 7.3) add more objects and methods to the repertoire (such as "curving" and "framing"), while *features* (see Section 7.4) provide notations for particular application areas (e.g., "arrows", "matrices", "polygons", "lattices", "knots"). In general, extensions provide visual components, whereas features add domain-specific notations for their logical composition.

This chapter gives examples of XY-pic's use in various application areas. Through this "teach by example" approach, it serves as a complement to the *XY-pic User's Guide* [106], which introduces the most used features, and the *XY-pic Reference Manual* [104], which describes the syntax of all XY-pic commands and their arguments. A study of our examples should put you in an excellent position to start drawing your own diagrams; we hope it will also convince you of the beauty, power, and flexibility of the XY-pic package.

A first example of XY-pic code

XY-pic consists of various modules. If you are not sure which ones to load, it is probably best to load "a large set", as follows:[1]

```
\usepackage[all]{xy}
```

Once you know enough about XY-pic to identify which functions you want to use, then you can specify only the extensions or features that are actually needed. For instance,

```
\usepackage[curve,arrow,cmactex]{xy}
```

loads the `curve` extension and `arrow` feature, which are tuned to produce `\special` commands understood by Thomas Kiffe's CMacTeX Macintosh port of TEX programs.

To get an idea of the philosophy on which XY-pic is based, let us first look at how we "construct" an XY-picture. To make things relatively easy, we consider a matrix-like diagram. As explained in more detail in Section 7.4.2, the principal way to create a diagram is with the command `\xymatrix{`*spec*`}`, where *spec* is the specification of the *matrix entries*, which, in general, are aligned in *rows* and *columns*. Just as in a `tabular` environment, entries inside a row are separated by ampersands and successive rows are separated by `\\`.

$$\begin{tikzcd}
A & \boxed{\sum_{i=n}^{m} i^2} & \\
 & \bullet & D \arrow[ul]
\end{tikzcd}$$

```
\usepackage[all]{xy}
\[
\xymatrix{
 A & *+[F]{\sum_{i=n}^m {i^2}} \\
   & {\bullet} & D \ar[ul]
}\]
```

Example 7-1-1

[1] For formats other than LATEX, use the command `\input xy` followed by `\xyoption{all}`. The `all` option loads the `curve`, `frame`, `tips`, `line`, `rotate`, and `color` extensions as well as the `matrix`, `arrow`, and `graph` features. Any other features or extensions needed must be loaded separately.

This example has two rows of three columns and shows a good deal about how XY-pic interprets commands.

- By default, entries inside XY-pic environments are typeset in mathematics mode, using "text style", and are centered.
- In many cases you may not start entries with a bare macro name—such names must be enclosed in braces or be otherwise "protected".
- As in a `tabular` environment, empty entries at the end of rows can be omitted if not referred to.
- Elements can be addressed by their *relative* ("logical") position in the diagram; thus `\ar[ul]` draws an arrow from the "current" position to the matrix cell "one up and one to the left".
- The *format* and *shape* of an element can be customized by specifying an "entry modifier" (e.g., "`[F]`" tells XY-pic to frame the entry).

If you have questions or need some help, you can address the XY-pic mailing list `xy-pic@tug.org`, to which you can subscribe by visiting the Web site `http://tug.org/mailman/listinfo/xy-pic`.

7.2 Basic constructs

A thorough knowledge of how XY-pic interprets the various commands will let you exploit its many functions fully. It will also help you understand the subtleties of the various extensions and features introduced in later sections.

A kernel XY-picture is enclosed in an `xy` environment:[1]

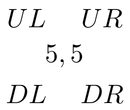
```
\begin{xy}...\end{xy}
```

The location at which an XY-pic object is being "dropped" is called its "position". In fact, in most cases only the coordinates or shape of the "current position" is set.

7.2.1 Initial positions

The simplest form of XY-pic position is called *absolute*, written `<`X`,`Y`>`. The coordinates X and Y are the offsets *right* and *above* the origin of the picture, which thus lies at `<0cm,0cm>`. Simple arithmetic operators can be used to position the current point. A comma is used to separate one position from another:

Example 7-2-1



```
\usepackage{xy}
\[\begin{xy}
  0*{DL} ,+/r1cm/*{DR}
 ,<0cm,1cm>*{UL} ,<1cm,1cm>*{UR}
 ,(5,5)*{5,5}
\end{xy}\]
```

[1] When using XY-pic with formats other than LaTeX, use `\xy...\endxy`.

Squares and triangles can be easily combined to create more complex diagrams. A special kind of diagram is the “pullback”, which is created as follows.

$$\begin{tikzcd}
T \arrow[drr, "x"] \arrow[dr, dotted, "(x,y)"] \arrow[ddr, "y"'] & & \\
& X \times_Z Y \arrow[r, "p"'] \arrow[d, "g"] & X \arrow[d, "f"] \\
& Y \arrow[r, "g"] & Z
\end{tikzcd}$$

```
\usepackage{diagxy}
\[\bfig
  \pullback|brra|
            [X\times_ZY`X`Y`Z;p`g`f`g]%
             />`{.>}`>/[T;x`(x,y)`y]
\efig\]
```

Example 7-4-9

In homology one often encounters 3×3 and 3×2 diagrams. They are typeset with the `\iiixiii` and `\iiixii` commands, respectively, whose default behavior is displayed in the following examples. The usual order for the arrow parameters is first all horizontal arrows and then all vertical ones, left to right, and then top to bottom.

```
\usepackage{diagxy}
$\bfig \iiixiii[A`B`C`D`E`F`G`H`I; 1`2`3`4`5`6`7`8`9`10`11`12] \efig$
\quad
$\bfig \iiixii[A`B`C`D`E`F; 1`2`3`4`5`6`7] \efig$
```

$$\begin{tikzcd}
A \arrow[r, "1"] \arrow[d, "7"'] & B \arrow[r, "2"] \arrow[d, "8" description] & C \arrow[d, "9"] \\
D \arrow[r, "3" description] \arrow[d, "10"'] & E \arrow[r, "4" description] \arrow[d, "11" description] & F \arrow[d, "12"] \\
G \arrow[r, "5"'] & H \arrow[r, "6"'] & I
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, "1"] \arrow[d, "5"'] & B \arrow[r, "2"] \arrow[d, "6" description] & C \arrow[d, "7"] \\
D \arrow[r, "3"'] & E \arrow[r, "4"'] & F
\end{tikzcd}$$

Example 7-4-10

A more interesting example of a 3×2 diagram is the following, where we add annotations (text and matrices) to the arrows. The placement of the arrow labels is specified with the first argument. Recall the order in which the arrow characteristics should be specified (see Example 7-4-10). We also load the amsmath package since we use the `pmatrix` environment.

```
\usepackage{diagxy,amsmath}
\[\bfig
 \iiixii|aaaalmr|<1000,800>
        [X`Y`Z`X\oplus X_0`Y\oplus X_0\oplus Z_0`Z\oplus Z_0;
         f_1`f_2`\begin{pmatrix}f_1&0\\0&1\\0&0\end{pmatrix}`
                 \begin{pmatrix}f_2&0&0\\0&0&1\end{pmatrix}`
```

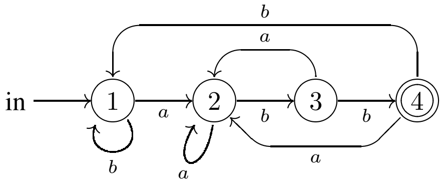
```
                    \begin{pmatrix}1\\0\end{pmatrix}`
                    \begin{pmatrix}1\\0\\0\end{pmatrix}`
                    \begin{pmatrix}1\\0\end{pmatrix}]
\efig\]
```

$$\begin{tikzcd}
X \arrow[r, "f_1"] \arrow[d, swap, "\begin{pmatrix}1\\0\end{pmatrix}"] & Y \arrow[r, "f_2"] \arrow[d, description, "\begin{pmatrix}1\\0\\0\end{pmatrix}"] & Z \arrow[d, "\begin{pmatrix}1\\0\end{pmatrix}"] \\
X \oplus X_0 \arrow[r, "\begin{pmatrix}f_1 & 0\\0 & 1\\0 & 0\end{pmatrix}"] & Y \oplus X_0 \oplus Z_0 \arrow[r, "\begin{pmatrix}f_2 & 0 & 0\\0 & 0 & 1\end{pmatrix}"] & Z \oplus Z_0
\end{tikzcd}$$

Example 7-4-11

Finite-state and stack diagrams

Finite-state diagrams can also be typeset in a straightforward way:



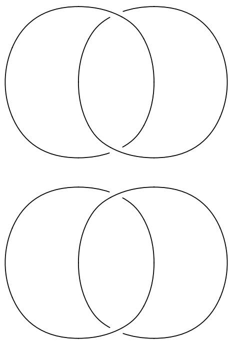
```
\usepackage[matrix,curve,arrow,tips,frame]{xy}
\[\UseTips
\entrymodifiers={++[o][F]}
\xymatrix @-1mm {
 *+\txt{in} \ar[r]
& 1 \ar@(dr,dl)[]^b \ar[r]_a
& 2 \ar@(d,dl)[]^a  \ar[r]_b
& 3 \ar `u[l] `^d[l]_a [l] \ar[r]_b
& *++[o][F=]{4}
  \ar `dl_l[ll]+/d6mm/`l_ul[ll]^a [ll]
  \ar `u^l[lll]+/u1cm/`l^d[lll]_b [lll]
}\]
```

Example 7-4-12

In this kind of diagram,[1] all states (elements) are enclosed in circles; here we use the `\entrymodifiers` command to specify the default modifier to realize this goal. To get nice arrowheads on the end of curves, we use Computer Modern tips. To keep the diagram a little more compact, we reduce the interelement spacing by 1 mm (`@-1mm` before the opening brace of the `\xymatrix` command). Starting an entry with an asterisk (i.e., using the form `*`⟨*object*⟩) overrides the default settings from `\entrymodifiers`; this feature is used in the leftmost cell to eliminate the frame and in the rightmost cell to typeset a double circle. Note that in the latter case the complete modifier specification had to be given. The only other tricky bit is the use of displacements towards the exterior, which add 6 mm (for a) and 1 cm (for b) in establishing the locations of the turns.

[1] We based our example on the deterministic finite automaton diagram in [7, p. 136]; another representation of the same diagram can be found in [106, Section 3.4], and we also used it for Example 3-4-10 on p. 79.

Note the use of the ^ character in the first position of the label “5”, which places the label “above” the arrow while the (default) _ character places it “below”.

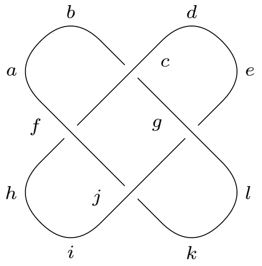


```
\usepackage[curve,knot,graph,dvips]{xy}
\[\xygraph{!{0;/r10mm/:}
          !{\vover}
      [u] !{\hcap[-2]}
      [d] !{\vover-}
    [ruu] !{\hcap[2]}
}\]
\[\begin{xy} 0;/r10mm/:
  ,\hcap[-2]\vunder\vunder-
  ,+(1,2),\hcap[2]
\end{xy}\]
```

Example 7-4-39

Since all knot crossings are, by default, bounded by a rectangle of one coordinate unit, and since loop and cap commands do not change the current point, it is convenient to use the graph feature to put together the various pieces of knot crossings and joins. This is shown in the top part of Example 7-4-39, where the \vover and \hcap commands position the elements by using “turtle” movements (up, down, left, right). The bottom part presents a variant diagram in which an explicit coordinate move was used to place the final \hcap. Note the use of the scaling factors, [2] or [-2].

Commands are also available to combine pieces in which the strings are basically at angles of 45 degrees, as in this next example.



```
\usepackage[curve,knot,arrow,dvips]{xy}
\[\renewcommand{\labelstyle}{\scriptstyle}
\begin{xy} 0;/r8mm/:
 ,{\xcapv-|{a}}
 , +(0,1) ,{\xcaph|{b}\xunderh|{c}%
           \xcaph|{d}\xcapv|{e}}
 ,-(3,0),{\xoverh|{f}}
 ,+(1,0),{\xoverh|{g}}
 ,-(3,1),{\xcapv-|{h}\xcaph-|{i}}
 ,+(0,1),{\xunderh-|{j}}
 ,+(0,-1),{\xcaph-|{k}}
 ,+(0,1),{\xcapv|{l}}
\end{xy}\]
```

Example 7-4-40

The placement of the various pieces in this construction is easy to follow by looking at the labels.

CHAPTER 8

Applications in Science, Technology, and Medicine

8.1 Typographical rules for scientific texts 512
8.2 Typesetting chemical formulae 518
8.3 Alignment and topology plots in bioinformatics 547
8.4 Drawing Feynman diagrams 555
8.5 Typesetting timing diagrams 572
8.6 Electronics and optics circuits 576

Because of its unsurpassed mathematical typesetting, TEX is widely used in the area of science, technology, and medicine (STM). It is not surprising, therefore, that the STM community has developed a number of packages to typeset the diagrams and schematics needed in their various disciplines. Chapter 8 of *The LATEX Companion, Second Edition* [83], describes in detail the AMS-LATEX package, which makes marking up (higher) mathematics rather more convenient than with TEX's basic commands. Chapter 10 of that book mentions a few simple packages, such as epic, eepic, and pspicture (or the recently released pict2e), which complement LATEX's `picture` environment for drawing "simple" generic graphics. Of course, the general packages, such as METAPOST (Chapters 3 and 4) and PSTricks (Chapters 5 and 6), or even the slightly more directed XY-pic package (Chapter 7) may provide all the functionality you need to typeset even the most complex graphics. Nevertheless, the specific needs of a given user community are often better served by a more targeted approach; the packages covered in this chapter address such problem areas.

In scientific texts, precision and consistency are of the utmost importance. Therefore we start with a brief discussion of typographic conventions in scientific texts. The next two sections describe packages for typesetting chemical structures and complex biological protein topologies. Section 8.4 explores various ways of constructing Feynman diagrams, an important tool used by physicists. The last two sections turn to electronics and describe dedicated packages for drawing timing and circuit diagrams.

8.1 Typographical rules for scientific texts

In scientific texts the typographic representation of a symbol carries a semantic meaning. Authors working in these areas should, therefore, be aware of and adhere to these typographical conventions. A brief summary of the most important rules for composing scientific texts follows (see also [52, 53, 56, 69]).

The most important rule in all circumstances is *consistency*: a given symbol should always be presented in the same way, whether it appears in the text body, a title, a figure, a table, or a formula; on the main line or as a superscript or subscript. An important corollary for LaTeX users is this: always typeset a symbol in either math or text mode—never mix the two, even if the results appear to be the same. Indeed, with LaTeX, the final visual appearance may change substantially when using a different class file or after adding a new package. For example, when using PostScript fonts, digits in text are taken from the PostScript text face and can look quite different from those in formulae. Therefore, it is good practice to always typeset numbers that refer to a result or part of a formula in math mode—i.e., surrounded by `$`.

In scientific texts, many symbols are traditionally typeset as *Roman* (upright) characters[1] and may not be understood properly otherwise. The most important such symbols are described here:[2]

- *Units*—for example, g, cm, s, keV. Note that physical *constants* are usually set in italics, so that units involving constants are mixed Roman–italics, e.g., keV/c (where c is the speed of light, a constant). Unit symbols are never followed by a period (see Section 8.1.1).
- *Chemical elements*—for example Ne, O, Cu—and *elementary particle names*—for example, p, K, q, H. To help the typist produce typographically correct texts, packages that contain commands representing the various names have been developed. In particular, chemists can use chemsym (see Section 8.1.2), while the PEN (Particle Entity Notation) scheme has been proposed for high-energy physics [34].[3]
- Standard mathematical functions ($\sin$, $\det$, $\cos$, $\tan$, $\Re$, $\Im$, etc.), for which the built-in LaTeX functions should be used.
- Numbers.

[1] With LaTeX, Roman type in mathematics mode can be achieved by the `\mathrm` command.

[2] See `http://physics.nist.gov/Document/typefaces.pdf` for a convenient two-page overview.

[3] Andy Buckley's heppennames package is an implementation of the PEN notation. He also wrote hepnicenames, which complements heppennames by providing more "user-friendly" names for often-occurring particles. These packages do, however, allow you too much freedom by offering the possibility to define the output style for the particle names. For instance, you can typeset their symbols in italic, a style still often (wrongly) used in American physics journals, rather than in Roman, as mandated by the IUPAP rules [56] described here. See Section 8.4.2 for an example of how these packages are used in practice.

Table 8.1: The importance of typographic rules in scientific texts

Roman Type		*Italic Type*	
A	ampere (electric unit)	A	atomic number (variable)
e	electron (particle name)	e	electron charge (constant)
g	gluon (particle name)	g	gravitational constant
l	liter (volume unit)	l	length (variable)
m	meter (length unit)	m	mass (variable)
p	proton (particle name)	p	momentum (variable)
q	quark (particle name)	q	electric charge (variable)
s	second (time unit)	s	c.m. energy squared (variable)
t	tonne (weight unit)	t	time (variable)
V	volt (electric unit)	V	volume (variable)
Z	Z boson (particle name)	Z	atomic charge (variable)

- Names of waves or states (p-wave) and covariant couplings (A for axial, V for vector); names of monopoles (E for electric, M for magnetic).
- Abbreviations that are pieces of words (exp for experimental; min for minimum).
- The “d” in integrands (e.g., $\mathrm{d}p$).

Obeying these typesetting conventions helps the reader understand at first glance the meaning of a symbol. Table 8.1 shows a few examples in which the meaning of a symbol depends on its typographic representation.

8.1.1 Getting the units right

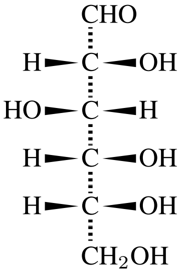
The importance of correctly typesetting units was recognized early, and several authors have developed packages to help users in this respect. Axel Reichert made a first step with his units and nicefrac packages. More recent and complete approaches are Patrick Happel’s unitsdef package and Danie Els’s SIstyle package. Both contain useful rules for expressing values of quantities.[1] SIstyle can be used together with Marcel Heldoorn’s SIunits package. This package, which we shall describe next, is by far the more complete and provides full support for all units defined by the International System of Units (abbreviated SI[2]), the modern form of the metric system. It is the world’s most widely used system of units, both in everyday com-

[1] The requirements for formatting and typesetting of SI units and numbers are described in the NIST (National Institute of Standards and Technology) document `http://physics.nist.gov/Document/sp811.pdf`. A very handy checklist for reviewing compuscripts is available from `http://physics.nist.gov/cuu/Units/rules.html`.

[2] From the French name *Système International d’Unités*. The SI was adopted by the “General Conference on Weights and Measures”, which is also known under its French acronym CGPM (*Conférence Générale des Poids et Mesures*; see `http://www.bipm.fr/en/convention/cgpm/`). The CGPM meets in Paris once every four years, and the last CGPM was held in October 2003. The SI is a coherent system based on seven base units as defined in the CGPM 1960 and subsequent conferences. An overview of the SI system is available in the brochure `http://www1.bipm.org/utils/common/pdf/si_brochure_8_en.pdf` (eighth edition, 2006).

Configurations, conformations, and reaction schemes

Numerous configurations of tetrahedral molecules with wedged bonds can be drawn using variants of the command `\tetrahedral`. For instance, the following Fischer diagram, which shows the absolute configuration of the sugar D-glucose, uses four nested `\tetrahedral` commands.

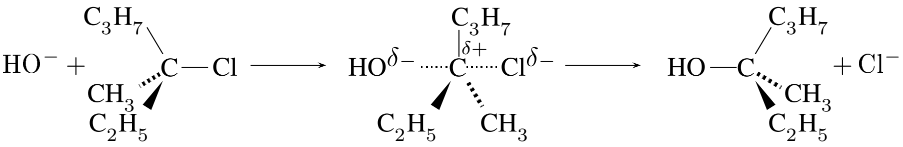


```
\usepackage{xymtexps}
\changeunitlength{0.09pt}
\tetrahedral{0==C;1A==CHO;%
             2B==H;4B==OH;3A==%
 \tetrahedral{0==C;1==(yl);%
              2B==HO;4B==H;3A==%
  \tetrahedral{0==C;1==(yl);%
               2B==H;4B==OH;3A==%
   \tetrahedral{0==C;1==(yl);%
                2B==H;4B==OH;3A==CH$\sb{2}$OH}}}}
```

Example 8-2-39

Finally, reaction schemes containing tetrahedral molecules with wedged bonds can also be handled. For instance, consider the Walden inversion reaction, which is drawn with the help of the `chemeqn` environment and the `\reactrarrow` command, both of which are defined in the **chemist** package (part of the XΥMTEX distribution).

```
\usepackage{xymtexps,chmst-ps}
\begin{chemeqn}
HO\sp{-}~+~
\raisebox{-28pt}{\ltetrahedralS{0==C;1==Cl;%
                                2==C$\sb{3}$H$\sb{7}$;%
                                3A==CH$\sb{3}$;4B==C$\sb{2}$H$\sb{5}$}}
\reactrarrow{0pt}{1cm}{}{}\qquad
\raisebox{-28pt}{\dtrigpyramid[{0{~~$\delta+$}}]%
                              {0==C;4A==HO$\sp{\delta-}$;%
                               5A==Cl$\sp{\delta-}$;%
                               1==C$\sb{3}$H$\sb{7}$;%
                               2A==CH$\sb{3}$;%
                               3B==C$\sb{2}$H$\sb{5}$}}
\quad\reactrarrow{0pt}{1cm}{}{}\quad
\raisebox{-28pt}{\rtetrahedralS{0==C;1==HO;%
                                2==C$\sb{3}$H$\sb{7}$;%
                                3A==CH$\sb{3}$;4B==C$\sb{2}$H$\sb{5}$}}
~+~Cl\sp{-} \label{myeqn}
\end{chemeqn}
```



(1)

Example 8-2-40

8.3.2 Membrane protein topology plots

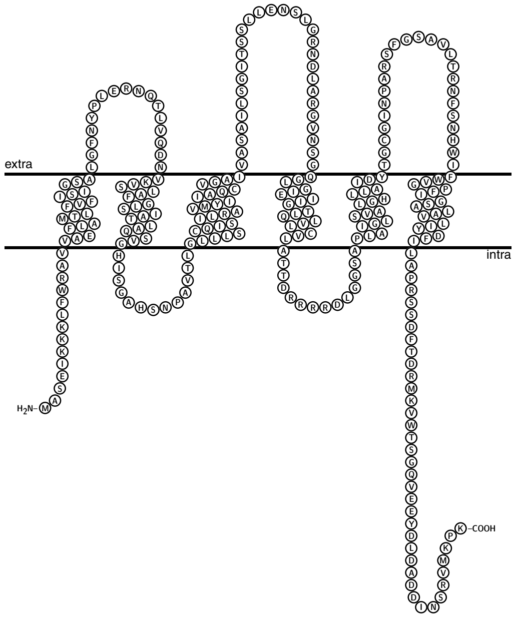
Eric Beitz also wrote the textopo package, which provides a LaTeX interface to generate shaded membrane protein topology plots. This package provides two new environments, `textopo` and `helicalwheel`.

The `textopo` environment displays schematic topology plots of membrane proteins. It allows you to import sequence and topology data or alignment files in various formats. You can also manually enter the sequence and the positions of the membrane spanning domains within the environment. The package implementation will generate a basic layout from these data, which can be further adjusted by adding labels, special styles for the presentation of residues, automatic or manual shading, and annotations.

```
\begin{textopo}[parameterfile]
    textopo commands
\end{textopo}
```

The parameter file *parameterfile*, which is optional, can contain any command defined by the textopo package to specify user parameter settings. The `textopo` environment itself must contain at least one command to load the sequence and topology data for the protein that must be plotted (i.e., `\getsequence` or `\sequence` and `\MRs`, which specify the positions of the membrane regions).

The following example, which uses the file `AQP1.PHD`, comes with the distribution.



```
\usepackage[]{textopo}
\begin{textopo}
 \getsequence{PHD}{AQP1.phd}
% no transmembrane labels
 \hideTMlabels
% small font size (range 1-10)
 \scaletopo{2}
\end{textopo}
```

Example 8-3-6

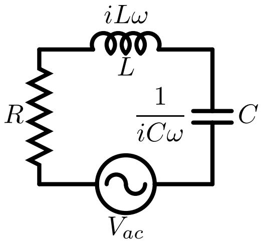
The second environment, `helicalwheel`, is in its functionality quite similar to `textopo`, but produces output that shows helical transmembrane spans as helical wheels command sequence for this precedure on a Unix machine would be similar to the following (depending on where the m4 files are stored):

```
m4 /usr/local/lib/m4/libcct.m4 cirexa.m4 > cirexa.pic
gpic -t cirexa.pic > cirexa.tex
```

This leaves us with a TEX file cirexa.tex, which contains only the tpic code for the example. To process it further, we could include it into a LATEX source using \input. This stores the picture in a box register named \graph, so we have to add a \usebox{\graph} statement into the document at the spot where we want it to appear.

Customizing the diagram

To show the flexibility of the circuit_macros approach, let us modify our example slightly to see how it behaves with an alternating current.

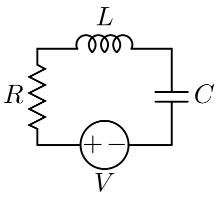


```
.PS
cct_init
linethick=1.6
define(‘dimen_’,0.6)
loopwid = 0.9; loopht = 0.7
  source(left_ loopwid,AC); llabel(,V_{ac},)
  resistor(up_ loopht,5); llabel(,R,)
  inductor(right_ loopwid,W); rlabel(,L,); llabel(,iL\omega,)
  capacitor(down_ loopht,); llabel(,C,)
                    rlabel(,\displaystyle\frac{1}{iC\omega},)
.PE
\usebox{\graph}
```

Example 8-6-11

After specifying thick lines, we draw an alternating current (AC) source. The resistor is made a little bigger, and we specify a complex value for the impedance of the self and the capacitor. Note how we place text at either side of the element with the llabel and rlabel commands. As the label text is set in mathematics mode, you can freely use math symbols and other specific commands for math mode (e.g., \displaystyle to choose a larger type size for the capacitor's numerator and denominator).

Some authors prefer to draw their circuit elements using a grid. We can write an m4 macro grid, which has two arguments \$1 and \$2 that define the x and y coordinates at which the element is to be drawn.



```
.PS
cct_init
gridsize = 0.1
define(‘grid’,‘(gridsize*‘$1’,gridsize*‘$2’)’)
  source(left_ from grid(7,0) to grid(0,0),V); llabel(,V,)
  resistor(up_ from grid(0,0) to grid(0,5),4); llabel(,R,)
  inductor(right_ from grid(0,5) to grid(7,5),W); llabel(,L,)
  capacitor(down_ from grid(7,5) to grid(7,0)); llabel(,C,)
.PE
\usebox{\graph}
```

Example 8-6-12

CHAPTER 9

Preparing Music Scores

9.1 Using TEX for scores—An overview 589
9.2 Using MusiXTEX 590
9.3 abc2mtex—Easy writing of tunes 600
9.4 Preprocessors for MusiXTEX 615
9.5 The PMX preprocessor 618
9.6 M-Tx—Music from TeXt 651
9.7 The music engraver LilyPond 661
9.8 TEX*muse*—TEX and METAFONT working together 666

Preparing music scores of high quality is a complex task, since music notation can represent a huge amount of information about the structure and performance of a musical piece.[1] While reading a score for performing a music piece, musicians must gather all the information they need, including the pitch and the length of the notes, the rhythm, and the articulation. Depending on the instrument, the musical notation may span more than a single stave (e.g., three or more for the organ), so the amount of data to be processed concurrently can be quite large. This makes great demands on the musician's ability, especially when sight-reading a piece. The quality of the typeset score plays an important role in this process since it must clearly show the structure of the piece.

High-quality music typesetting requires a good eye and much experience. Until recently, this type of work has been done by highly trained music engravers who manage, according to Helene Wanske [136], no more than one or two pages per day. As in typesetting of text, a criterion of high quality is the overall look of the page, especially the distribution of black and white. Several texts about music notation practice have been published, but they cannot replace a practitioner when it comes to ensuring the aesthetic form of the score as a whole. The Production Committee of the Music Publisher's Association has pub-

[1] The Web site `http://www.music-notation.info/` provides a set of pointers to music notation languages, programs, fonts, etc.

lished a text that outlines a series of standards for music notation (`http://www.mpa.org/notation/notation.pdf`). *The Big Site of Music Notation and Engraving* (`http://www.coloradocollege.edu/dept/MU/Musicpress/`) intends to provide a helpful source for musicians, typesetters, students, publishers, and anyone else who is interested in music notation and engraving. See also Jean-Pierre Coulon's *Essay on the true art of music engraving* (`http://icking-music-archive.org/lists/sottisier/sottieng.pdf`).

In recent years several computer systems for writing scores have been developed. Encore (`www.encoremusic.com`), Finale (`www.finalemusic.com`), and Sibelius (`www.sibelius.com`) are examples of commercial products, while Rosegarden (`http://www.rosegardenmusic.com/`) and noteedit (`http://developer.berlios.de/projects/noteedit`) are freely available developments. All of these programs are of the WYSIWYG (What You See Is What You Get) type, and most of them have reached a genuine state of perfection. However, they cannot yet replace an experienced music engraver. All they can do to ensure high-quality typesetting is to create a "nice" draft: they contribute to a high-quality score only if they leave the aesthetic decisions to the *experienced* user.

This role is even more evident when one considers nonstandard situations, which are encountered in modern music, for which notational requirements are hard to standardize at all. Indeed, music, as a live art form, evolves continuously, and its current practice is often quite distinct from that of the 18th and 19th centuries, when the "standard" music notation was consolidated. Whereas standard notational practices are quite sufficient for popular and commercial music (and thus the favored target for commercial software), "modern" music goes well beyond this traditional form, in particular in its graphic representation. Moreover, musicology has notational needs (e.g., symbols for highlighting certain notes, unusual ties, superposition of staves) for the analysis of all kinds of music—classical and contemporary, western and oriental, ethnic from various peoples of the world—that go well beyond the possibilities of current professional typesetting applications. What is needed is a programmable system, and here TEX can be an important player.

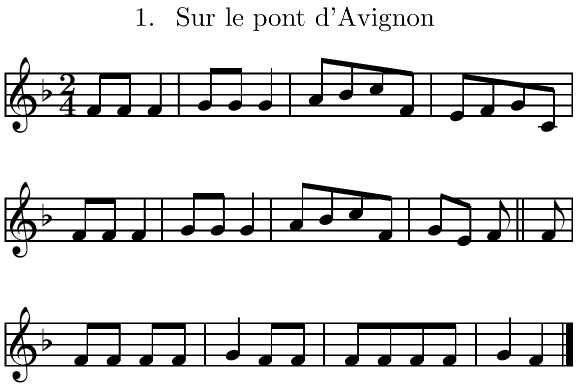
In this chapter, after a short historical introduction (Section 9.1), we first consider MusiXTEX, a set of TEX macros that build a very powerful and flexible tool for typesetting scores. As MusiXTEX makes no aesthetic decisions—these choices must all be made by the typesetter—it is quite complex to use. Therefore several preprocessors have been developed to provide an easier interface. In Section 9.3, we introduce the abc language, which is in widespread use for folk tunes. In Section 9.5, we describe the PMX language, which makes entering polyphonic music more convenient. In Section 9.6, we have a look at the M-Tx language, an offspring of PMX, which adds, among other features, support for dealing with multi-voice lyrics in scores. In Section 9.7, we introduce LilyPond, a music typesetter written in C++, while Section 9.8 says a few words about TEX*muse*.

The *Werner Icking Music Archive* (`http://icking-music-archive.org`) contains a lot of material related to music software. In particular, it is the definitive archive of software related to MusiXTEX, including pointers to the latest developments of abc, PMX, M-Tx, and their brethren. It also contains hundreds of freely available music scores typeset with MusiXTEX, often with accompanying input files, so that it is an ideal source of examples.

This chapter is somewhat unusual as it contains little LATEX: MusiXTEX is essentially low-level TEX, albeit with a LATEX interface; some of the programs discussed to translate musical languages, such as abc, even bypass TEX altogether. We nevertheless believe that it is approa little practice, most users can play a tune directly from the abc notation (without generating sheet music output). Moreover, the simplicity and clarity of the notation make it a straightforward matter to notate tunes that are stored in a computer file. In addition, these files can be easily exchanged by e-mail, thus enabling dissemination and discussion of the music. In fact, the abc language has become the de facto standard among folk musicians, and thousands of tunes in abc notation are now available on the Internet (see, e.g., `http://abcnotation.org.uk/tunes.html`).

9.3.1 Writing an abc source

To see how an abc source is built up, consider the following example:



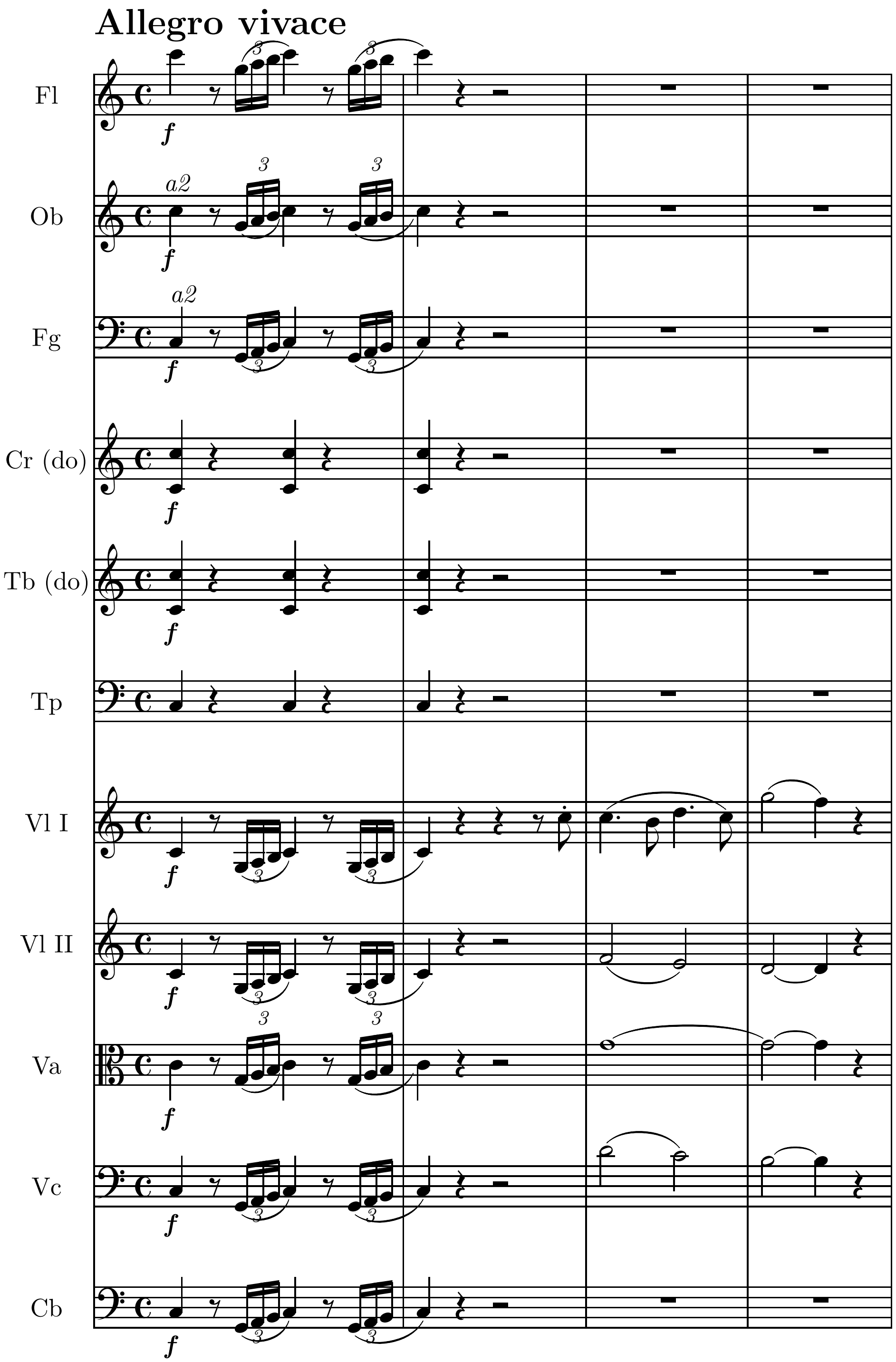
```
X:1
T: Sur le pont d'Avignon
M:2/4
L:1/8
K:F
FF F2 | GG G2 | ABcF | EFGC |
FF F2 | GG G2 | ABcF | GE F || F |
FF FF | G2 FF | FFFF | G2 F2 |]
```

Example 9-3-1

An abc source consists of two parts: a header and a body. The header (shown in blue in the examples) contains information fields, each starting with an uppercase letter to denote the kind of information, followed by a colon. The body consists of the music piece itself. Within the body, additional information fields can be inserted that are used for changes to the header information (e.g., the key, meter, or tempo).

Table 9.3 shows all possible information fields, most of which are optional. A few words about the more important ones follow.

- Musical information:
  - `K:` the key, consisting of a capital letter possibly followed by a `#` or `b` for sharp or flat, respectively. You can use major keys (e.g., `K:Emaj`) or minor keys (`K:gmin`), or specify other modes, such as Mixolydian (`K:AMix`) and Dorian modes (`K:EDor`).
  - `L:` the default note length (i.e., `L:1/4` for a quarter note, `L:1/8` for an eighth note, etc.). The default note length is also set automatically by the meter field `M:`.
  - `M:` the meter, such as `M:3/4`, `M:C` (common time), or `M:C|` (cut time).



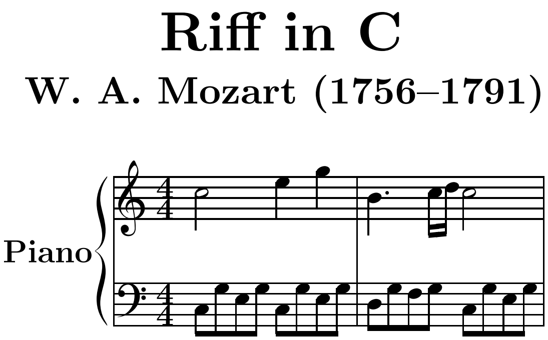
Example 9-5-36

9.6 M-Tx—Music from TeXt

After describing the PMX language we now turn to Dirk Laurie's M-Tx language,[1] which adds a layer of convenience to PMX, making entering information—in particular, in the preamble—more intuitive. By its very conception, it offers also a straightforward way for adding words (lyrics) to the music.

Let us first have another look at Section 9.4 on page 615, especially the example comparing the coding of the first bars of the Mozart piece. One large difference between PMX and M-Tx coding is that, with M-Tx voice (instrument) lines are input *as they are printed* (i.e., from top to bottom), whereas with PMX they are entered last line first (i.e., from bottom to top).

Example 9-6-1



```
Title: Riff in C
Composer: W. A. Mozart (1756--1791)
Style: piano
Name: Piano
Meter: 4/4
Size: 16
Indent: 0.18

%% w70m
c2+       e4    g   | b4d-  c1 d c2         |
c8 g+ e g c- g+ e g | d g f g    c- g+ e g  |
```

Example 9-6-1 was compiled by the M-Tx processor prepmx, which transforms the M-Tx input file into a PMX file to be run through the pmxab processor.

```
> prepmx 9-6-1
==> This is M-Tx 0.60 (Music from TeXt) <16 March 2005>
==>> Input from file 9-6-1.mtx
Writing to 9-6-1.pmx
instrumentNames = TRUE
PrePMX done.  Now run PMX.

> pmxab 9-6-1
 This is PMX, Version 2.506, 14 Nov 04
 Opening 9-6-1.pmx
 Starting first PMX pass
  Bar 1  Bar 2
 Done with first pass
 Starting second PMX pass
  Bar 1  Bar 2
 Writing ./9-6-1.tex
 Done with second PMX pass.
```

The prepmx processor has several options, all of which are described in the M-Tx manual.

[1] The M-Tx entry on the home page `http://icking-music-archive.org/software/indexmt6.html` of the Icking Music Archive provides pointers to the latest version of the distribution, manual, examples, and related utilities.

9.7 The music engraver LilyPond

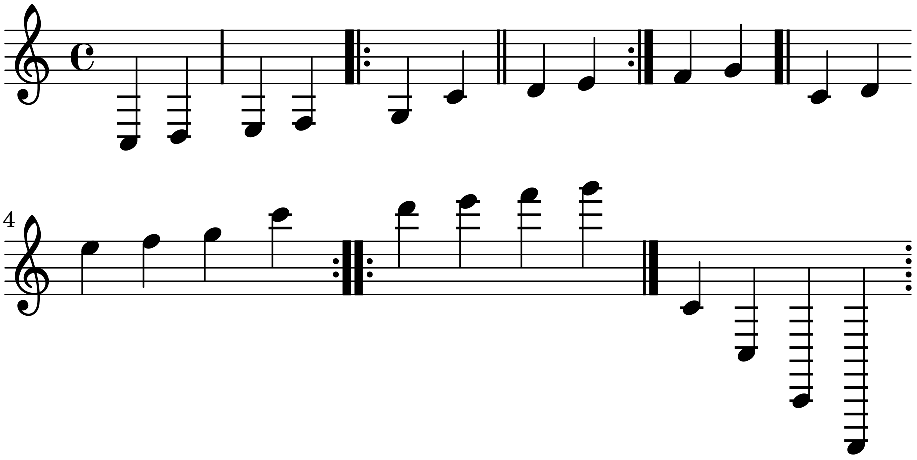
In 1996, in the previous edition of this book, we described Jan Nieuwenhuizen’s MPP MusiXTeX preprocessor [89]. Since then, Jan and his colleague Han-Wen Nienhuys have abandoned that system and developed LilyPond,[1] an “automated engraving system that formats music beautifully and automatically and has a friendly syntax for its input files”. They no longer use TeX as the basic typesetting engine but have developed a large C++ program (more than 6000 lines of code); they also use Python and Scheme code, as well as a specially designed font family (*feta*), which is available in various formats (PostScript Type 1, OpenType, and SVG).

9.7.1 The LilyPond source language

To typeset one note, four kinds of information can be specified: *notename*, *octave*, *duration*, and *features*. Only the *notename* is mandatory. All this information is coded in the given order with no intervening spaces; a blank separates two notes.

Notes are denoted by lowercase letters. A comma (`,`) following the letter transposes the note one octave deeper, while a right quote (`’`) makes it an octave higher. To generate different clefs, use the command `\clef` followed by either `treble`, `alto`, `tenor`, or `bass`. The following example shows some pitches and ways to generate different kinds of *bar lines*.

```
{c d   \bar "|" e f    \bar "|:"  g c’        \bar "||"
 d’ e’ \bar ":|" f’ g’ \bar ".|"  c’ d’       \bar ".||." \break
 e’’ f’’ g’’ c’’’      \bar ":|:"
 d’’’ e’’’ f’’’ g’’’   \bar "|." c’ c c, c,,  \bar ":"    }
```



Example 9-7-1

[1] The LilyPond home page is at `www.lilypond.org`, where you can download the latest version of the system. There is also a tutorial, the reference guide, and much more. Of particular interest is the essay “What is behind LilyPond?”, which explains the authors’ views on problems in music notation (software) and their approach to solving them.

CHAPTER 10

Playing Games

10.1 Chess 668
10.2 Xiangqi—Chinese chess 687
10.3 Go 690
10.4 Backgammon 696
10.5 Card games 698
10.6 Crosswords in various forms 702
10.7 Sudokus 709

Board and card games have a long history, and thousands of books in many languages have been dedicated to chess, Go, cards, and the like. These books almost always use diagrams to explain the rules or show the evolution of a game. In the present chapter we look at a number of examples showing how to prepare such graphical presentations with LaTeX.

Most game packages are concerned with making available either a special font for typesetting the right symbols or macros for producing nice examples of the state of play. The highly developed field of chess notation, however, lends itself well to an algorithmic typesetting system like LaTeX. The chess packages, with which we begin, keep track of the state of moves and allow various forms of output.

We move next to the rather similar games of Chinese chess and Go, followed by backgammon. We then look at cards, where the classic game of bridge has a special package, before concluding the chapter with the esoteric subject of crossword and Sudoku puzzles. Although crossword design is not a game, it has some similar typesetting problems, and LaTeX-using crossword makers will enjoy using the sophisticated package to help them. In the case of Sudoku, there is even a package that generates new puzzles or solves existing ones.

```
\ahead    \dummy    \ddummy
```

It is, of course, also possible to talk about the next move in a commentary started with `\[` or `[`: simply prefix the first move inside with `\ahead`.

If certain moves are irrelevant for the analysis you can use `\dummy` or `\ddummy` to advance the game state by one or two half-moves, respectively. This means that skak can't follow the position on the board any longer, so texmate immediately disables this functionality with `\SkakOff` upon encountering these commands for the remainder of the variation.

French Defense analysis:
1. e4 e6 2. d4 d5 3. ♘c3 ♗d4 4. e×d5 e×d5 5. ♗d3 ♘c6 6. a3 ♗e7 7. ♗f4! [7...♘×d4?! 8. ♗b5+! ♘c6 9. ♘×d5 ♗d6 10. ♕e2+ ♘ge7 11. ♖d1 ♗d7 12. ♗×c6 ♗×c6 13. ♘×c7+!+−] **7...a6!** [7...♗e6 8. ♘f3 ♘f6 (8...♗g4 9. h3! ♗h5 10. ♘b5!♖c8 ♗f5!+−) 9. ♘b5! ♖c8 10. ♘e5! ♘×e5 11. d×e5... 12. ♘×a7] **8. ♘f3!**

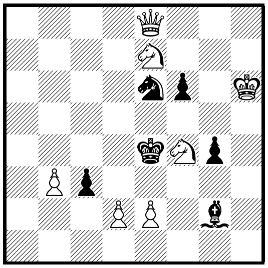
```
\usepackage{texmate}
\setchessfontfamily{leipzig}
French Defense analysis:\\
| e4 e6 ; d4 d5 ; Nc3 Bd4 ; exd5 exd5 ;
Bd3 Nc6 ; a3 Be7 ; Bf4! [ \ahead Nxd4?! ;
Bb5+! Nc6 ; Nxd5 Bd6 ; Qe2+ Nge7 ;
Rd1 Bd7 ; Bxc6 Bxc6 ; Nxc7+!\wdecisive ]
a6! [ Be6 ; Nf3 Nf6 [ Bg4 ; h3! Bh5 ; Nb5!
Rc8 ; Bf5!\wdecisive] ; Nb5! Rc8 ;
Ne5! Nxe5 ; dxe5 \dummy\,\dots Nxa7 ] Nf3! |
```

Example 10-1-19

If there are multiple variations to discuss as alternatives at a certain point in the game, you can use the `variations` environment or its starred form.

`\begin{variations}\var` *variation*$_1$ `\var` *variation*$_2$ ... `\end{variations}`

Inside the `variations` environment, each variation is introduced with a `\var` command. This will typeset the first move of a variation in boldface and separate variations by a semicolon. Alternatively, you can use `\var*`, in which case no special formatting is applied. The starred form `variations*` of the environment is equivalent to using `\var*` for all variations.

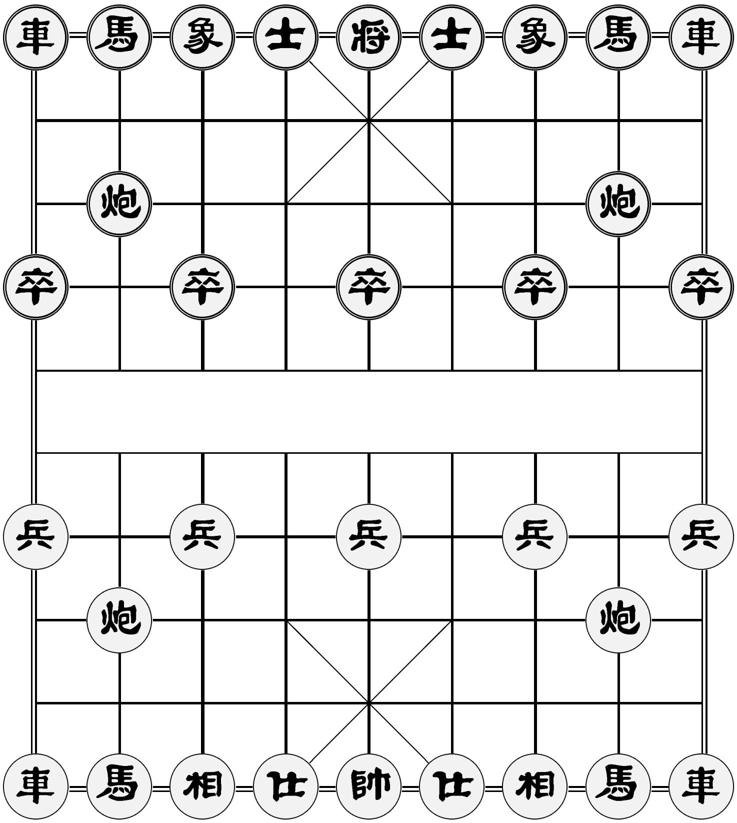


Mate in 3 moves by Bayersdorfer, 1888

1. ♘d3!△2. ♕a8+ ♔d4 3. ♕a4# [**1...♘d4** 2. ♘c5+ ♔e5 (2...♔f4 3. ♕b8#) 3. ♕b8# ; **1...c×d2** 2. ♘f5! △♕×e6# ♔d5 (2...♔×f5 3. ♕g6#) 3. ♕a8#]

```
\usepackage{texmate}
\setchessfontfamily{leipzig}

\position{4Q3/4N3/4np1K/8/4kNp1/1Pp5/3PP1b1/8}
\shortstack{\showboard\\
   Mate in 3 moves by Bayersdorfer, 1888}

| Nd3! \Threat<\withidea Qa8+ Kd4 Qa4 \#>
  [\ahead\begin{variations}
     \var  Nd4 Nc5+ Ke5 [Kf4 Qb8 \#] Qb8 \#
     \var cxd2 Nf5! \threat<Qxe6 \#>
          Kd5 [Kxf5 Qg6 \#] Qa8 \#
   \end{variations}] |
```

Example 10-1-20



Example 10-2-2

Figure 10.1: Initial setup of Chinese chess game (xiangqi)

The following listing, a mate situation after four moves, gives an example of the use of this command. The board situation after these four moves is shown in Example 10-2-4 on the following page.

1. 炮 h3–e3 馬 b0–a8

2. 炮 e3×e7 車 a0–a9

3. 炮 b3–b5 馬 h0–g8

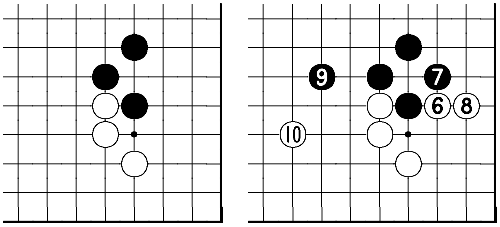
4. 炮 b5–e5 mates!

Example 10-2-3

```
\usepackage{cchess}
\newcommand\x{$\times$} % a shortcut to denote capture
\begin{tabbing}
1. \= \textpiece{c}h3--e3 \qquad
                            \=\textpiece{N}b0--a8 \\
2. \> \textpiece{c}e3\x e7 \>\textpiece{R}a0--a9 \\
3. \> \textpiece{c}b3--b5  \>\textpiece{N}h0--g8 \\
4. \> \textpiece{c}b5--e5 mates!
\end{tabbing}
```

The position environment draws a complete board. Within its body, the \piece command is used to place the individual pieces.

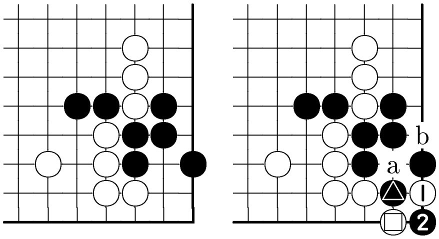
indicates the color of the first stone being placed. This method is most suitable to record games or longer sequences where the order of play needs to be indicated.



```
\usepackage{igo}
\white[\igonone]{q3,q5,p5,p6,p4,q7}
\showgoban[m1,t8]
\white[6]{r5,r6,s5,n6,m4}
\showgoban
```

Example 10-3-2

If `\white` or `\black` is used without an optional argument or if the optional argument is `\igotriangle`, `\igosquare`, `\igocircle`, or `\igocross`, then all stones typeset are of the same color and decorated with the respective glyph as specified by the optional argument. This input method is most suitable for documenting Go problems, where the order of stones placed previously is unimportant.

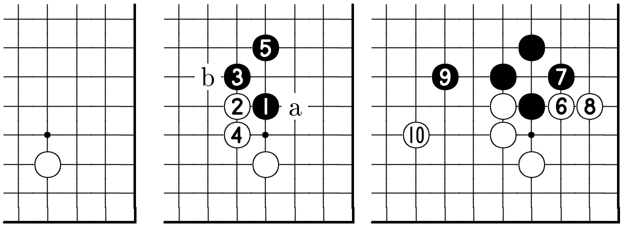


```
\usepackage{igo}
\white{o3,q2,q3,q4,r2,r5,r6,r7}
\black{p5,q5,r3,r4,s4,s5,t3}
\showgoban
\black[\igotriangle]{s2}
\white[\igosquare]{s1}
\gobansymbol{s3}{a}\gobansymbol{t4}{b}
\white[1]{t2,t1}
\showgoban
```

Example 10-3-3

```
\cleargobansymbols
```

Once the progress in a game has been shown in a diagram, it is customary to show the already placed stones in later diagrams without numbers, achieved by issuing a `\cleargobansymbols` command. This helps in identifying newly placed stones and makes the diagrams more readable. Whether numbering is continued is a matter of taste. Although igo supports sequentially numbered stones for a full game, for readability it is usually better to restart numbering when three-digit numbers are reached and you can afford to typeset more than a single diagram.



```
\usepackage{igo}
\white{q3}
\showgoban[p1,t8]
\black[1]{q5,p5,p6,p4,q7}
\gobansymbol{r5}{a}\gobansymbol{o6}{b}
\showgoban[n1,t8]\cleargobansymbols
\white[6]{r5,r6,s5,n6,m4}
\showgoban
```

Example 10-3-4

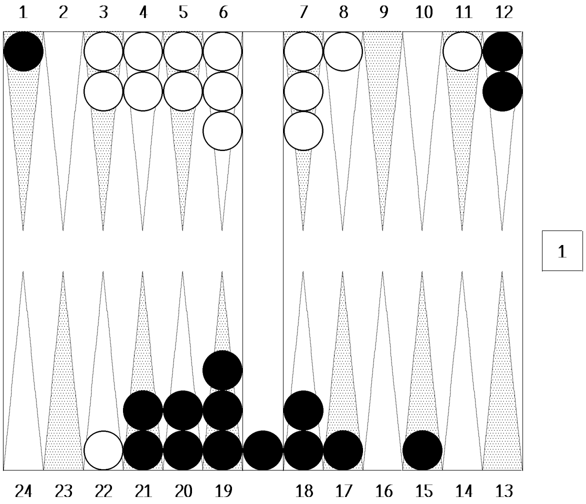
10.4 Backgammon

Jörg Richter's package bg defines two LaTeX environments, `position` and `game`, to display backgammon games. The `position` environment draws a single board and is thus convenient for discussing a problem, while with the `game` environment you can enter each move individually. In the latter case the board positions are stored internally, allowing the "current" status to be drawn at any time.

By convention, the homes of both players are on the left-hand side, with white's home at the top and black's home at the bottom. Unlike in the other packages discussed so far, positions on the board are not denoted with absolute coordinates but rather are numbered as viewed by the party whose move is being placed (e.g., white's 24 corresponds to black's 1, and so on). Moves are always performed from high to low numbers, and the cube is always on the right-hand side of the board.

```
\begin{position}...\end{position}
```

The `position` environment initializes an empty board into which stones are placed by the commands described below. Some of these commands also allow you to customize some aspects of the board's layout. The board is printed when the `\end{position}` command is encountered. Example 10-4-1 shows the use of various commands of the `position` environment.



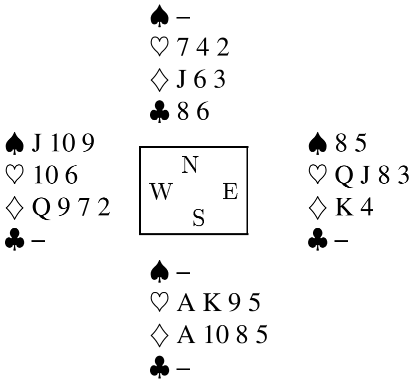
White to play 3–2

```
\usepackage{bg}
\begin{position}
\normalboard
\whitepoint{3}{2}    \whitepoint{4}{2}
\whitepoint{5}{2}    \whitepoint{6}{3}
\whitepoint{7}{3}    \whitepoint{8}{1}
\whitepoint{11}{1}   \whitepoint{22}{1}
\blackpoint{24}{1}   \blackpoint{13}{2}
\blackpoint{10}{1}   \blackpoint{8}{1}
\blackpoint{7}{2}    \blackpoint{6}{3}
\blackpoint{5}{2}    \blackpoint{4}{2}
\blackbar{1}
\shownumbers \middlecube{1} \showcube
\whiteonmove
\boardcaption{White to play 3--2}
\end{position}
```

Example 10-4-1

```
\blackpoint{p}{n}    \whitepoint{p}{n}
```

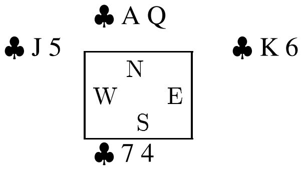
These two commands are used to place stones on the board; *n* denotes the number of stones to place and *p* denotes the point where they are positioned. It is important to remember that these points are numbered downwards from 24 relative to the home position of each player.



```
\usepackage{bridge}
\crdima{}{}
 {\hand{--}{7 4 2}{J 6 3}{8 6}}
 {\hand{J 10 9}{10 6}{Q 9 7 2}{--}}
 {\hand{8 5}{Q J 8 3}{K 4}{--}}
 {\hand{--}{A K 9 5}{A 10 8 5}{--}}
```

Example 10-5-4

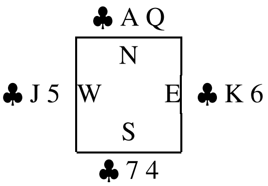
In discussing certain techniques of play, often only the card distribution in a single suit is shown. In that case it would be nice not to use the `\hand` command in the arguments of `\crdima`, but unfortunately the result is not quite what we would expect.



```
\usepackage{bridge}
\crdima{}{}
  {\club{} A Q}
  {\club{} J 5}{\club{} K 6}
  {\club{} 7 4}
```

Example 10-5-5

In this case a solution using the `tabular` environment gives better results. The first argument specifies the suit of interest, and the other arguments correspond to the four players (with the same order as in the `\crdima` command). Note the use of the `\multicolumn` command to suppress the vertical lines in the first and last rows.

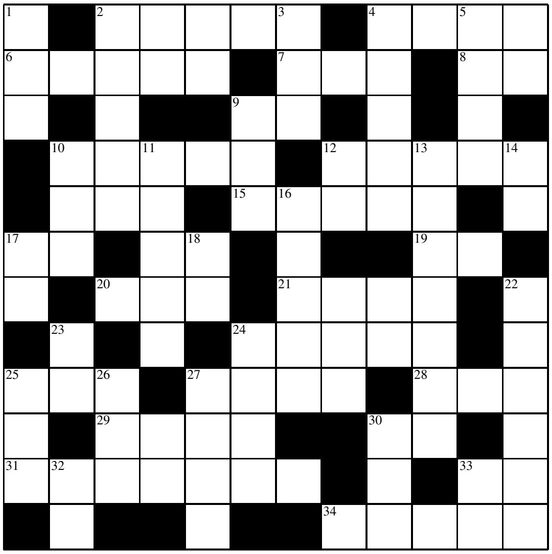


```
\usepackage{bridge}
\newcommand{\Crdexa}[5]{{\renewcommand\arraystretch{1.2}%
   \begin{tabular}{l|@{}c@{}|l}
     \multicolumn{1}{c}{} & \multicolumn{1}{c}{#1 #2}  \\
     \cline{2-2}
                          &    N      &                \\
                    #1 #3 &W\hfill\hfill E& #1 #4      \\
                          &    S      &                \\
     \cline{2-2}
     \multicolumn{1}{c}{} & \multicolumn{1}{c}{#1 #5}
   \end{tabular}}}
\Crdexa{\club}{A Q}{J 5}{K 6}{7 4}
```

Example 10-5-6

Bidding

An important part of the bridge game is the initial bidding phase, in which the players decide who plays the contract. To document such a bidding sequence, Kees van der Laan introduced a `bidding` environment as an application of LaTeX's standard `tabbing` environment.



ACROSS

2 Gap between tree node labels and the node in PSTricks (5)
4 Modern replacement for scissors and glue (4)
6 A Unicode TEX variant (5)
7 … you always wanted to know but never dared to ask (3)
8 A graphics key that needs four numbers (2)
10 Called bb in Karl Berry's font-naming schemes (5)
12 A way to make your pages into thumbnails (5)
15 You can do it to a box but it isn't proper LATEX (5)
19 In LATEX denotes $\wp$; in other circumstances might mean a word processor (2)
20 Result of a TEX run (3)
21 A language whose name should probably have five letters, but then it was developed for Unix (4)
24 It's not Intel (5)
25 A pointer misspelled (3)
27 Testing your LATEX knowledge: $\prec$ (4)
28 Label for a signal line (3)
29 Another name for the LATEX3 project team on c.t.t. (4)
30 One way to get a sharp in MusiXTEX (2)
31 A figure or plan intended to explain rather than represent actual appearance (7)
33 72.27 to an inch (2)
34 see **1d** (5)

Example 10-6-1

DOWN

1 & a34 Grand wizard of TEX (3,5)
2 A ready-to-run TEX for Unix (5)
3 A novice golfer's dream (3)
4 LATEX 2_ε name for document style (5)
5 Double beam above notes in MusiXTEX (4)
9 Either/or—mathematically speaking (3)
10 German beer (3)
11 Save your coordinates (PSTricks) (5)
12 Approximation of TEX's version number (2)
13 A PostScript operator (7)
14 Probability function (2)
16 A divine messenger misspelled (5)
17 How do you get an Å? (2)
18 ξ (2)
22 LATEX has rigid and rubber ones (6)
23 Amor uses them and XY-pic calls them (2)
24 Length of the line segment where the connector joins the first node (4)
25 Files containing LATEX font-definition documentation (3)
26 η—don't say this is all Greek to you (3)
27 $\perp$, also the first letters of everlasting (4)
30 We plot it in Chapter 4 (3)
32 TEX's name for inch (2)
33 Lula is chief of (2)

Figure 10.2: A sample crossword for you to fill in (done with crosswrd)

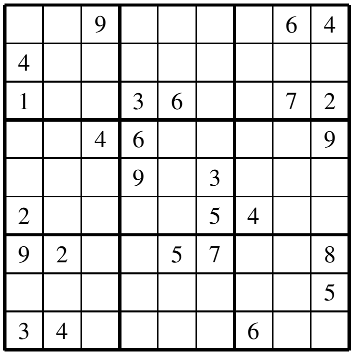
The size of the grid can be adjusted by setting \sudokusize (the default value is 10cm), and the size and font for the numbers can be manipulated by redefining \sudokuformat as shown in Example 10-7-1. The default definition uses \Huge to fit the larger grid size. The package also offers the environment sudoku, which is simply an abbreviation for sudoku-block inside a center environment.

10.7.2 sudokubundle—Solving and generating Sudokus

In 2006, Peter Wilson published a bundle of three packages that not only typeset but also attempt to solve existing Sudokus or generate new ones. In contrast to the **sudoku** package, with Wilson's bundle the puzzles have to be stored in external files and require a somewhat different input syntax.

In these external files, only the first nine lines are relevant. Each must consist of nine characters, either a dot (representing an empty cell) or one of the numbers 1 to 9 (indicating prefilled cells). Any further lines can be used for comments and will not be read by LaTeX.

The **printsudoku** package provides the command \sudoku for typesetting such files. It also offers a \writepuzzle command to write external Sudokus into separate files, but for this purpose a filecontents* environment, as used in the next example, or a simple text editor is equally or even more suitable.



		9					6	4
4								
1			3	6			7	2
		4	6					9
			9		3			
2					5	4		
9	2			5	7			8
								5
3	4					6		

```
\usepackage{printsudoku}
\begin{filecontents*}{sample.sud}
..9....64
4........
1..36..72
..46....9
...9.3...
2....54..
92..57..8
........5
34....6..
A moderate challenge
\end{filecontents*}
\cluefont{\small}
\cellsize{1.2\baselineskip}
\sudoku{sample.sud}
```

Example 10-7-2

As seen in the previous example, the size of the puzzle and the numbers inside are controlled through \cluefont (default \Huge) and \cellsize (default 2.5\baselineskip), respectively. Note that compared to the **sudoku** package these are declarations, rather than length registers or macros, and thus are changed in a different way. For example, to get sans serif numbers, we would need to use \sffamily instead of using \textsf.

The **solvesudoku** package attempts to solve a given puzzle and prints the solution as far as it was able to produce it. Given that TeX isn't the best language in which to implement complicated algorithms, it does a surprisingly good job and is able to fully resolve most

CHAPTER 11

The World of Color

11.1 An introduction to color 714
11.2 Colors with LaTeX — The color and xcolor packages 719
11.3 Coloring tables 737
11.4 Color slides with LaTeX — The beamer class 752

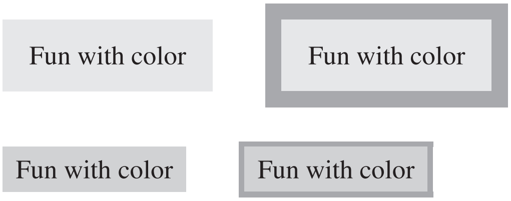
For many people, color is indispensable for effective graphics. All of the modern interactive drawing packages support coloring of lines, filling objects with color, etc., and all of the standard bitmap file formats such as GIF (Graphics Interchange Format), PNG (Portable Network Graphic), JPEG (Joint Photographic Experts Group), PBM (Portable Bitmap), TIFF (Tagged Image File Format), BMP (Windows Bitmap), SVG (Scalable Vector Graphic), and Encapsulated PostScript support color. Thus, if you generate a picture with a drawing package, and then import it into your LaTeX document using the packages described in Chapter 2, you should have no problems if your printing or viewing device supports color. However, you do have to know something about how color is represented and which color model you are using. We discuss these issues in the first part of this chapter.

If you prepare your graphics using LaTeX itself or simply want colored text, you need some special support from both LaTeX and your driver. The main body of this chapter describes the extended LaTeX xcolor package, which we believe is powerful enough to meet almost all needs and is capable of working with most other packages. xcolor extends the old color package with features such as color mixing, color sequences, and tabular shading.

LaTeX users often request color for use in presentations. The xcolor package can, of course, be used with old LaTeX slides classes, but we devote some space to explaining a more sophisticated class, beamer, and give lots of examples of its facilities.

As the book is printed in two colors, it is possible to show some color effects in examples. All other colors will appear in grayscale throughout the text. However, we repeat selected examples in the color plates. We indicate when the reader should refer to the full-color version. You can also take the example source code, run it through LaTeX or pdfLaTeX, and view the PostScript or PDF output.

Some further examples (also in Color Plate XIII b) show how to control the exact form of the box with the \fbox parameters \fboxrule and \fboxsep, which specify the thickness of the rule and the size of the shaded area respectively.



```
\usepackage{color}
\setlength{\fboxrule}{6pt}%
\setlength{\fboxsep}{10pt}%
\colorbox{yellow}{Fun with color}\qquad
\fcolorbox{red}{yellow}{Fun with color}
\par\bigskip\par
\setlength{\fboxrule}{2pt}%
\setlength{\fboxsep}{5pt}%
\colorbox{green}{Fun with color}\qquad
\fcolorbox{red}{green}{Fun with color}
```

Example 11-2-6

Combining the use of PostScript fonts and color, you can construct lists with colorful elements; the \ding command is part of the pifont package described in [83, p. 378].

❑ On the first day of Christmas my true love sent to me

  ☛ a partridge in a pear tree

❑ On the second day of Christmas my true love sent to me

  ☛ two turtle doves

  ☛ and a partridge in a pear tree

❑ On the third day of Christmas my true love sent to me

  ☛ three French hens

  ☛ two turtle doves

  ☛ and a partridge in a pear tree

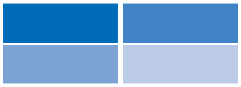
```
\usepackage{pifont,color}
\newenvironment{coldinglist}[1]
        {\begin{list}{\textcolor{blue}{\ding{#1}}}{}}
        {\end{list}}
\newcommand\OnThe[1]{On the \textcolor{blue}{#1} day of
                     Christmas my true love sent to me}
\begin{coldinglist}{113}
 \item \OnThe{first}
 \begin{coldinglist}{42}
    \item a partridge in a pear tree
 \end{coldinglist}
 \item \OnThe{second}
 \begin{coldinglist}{42}
    \item two turtle doves
    \item and a partridge in a pear tree
 \end{coldinglist}
 \item \OnThe{third}
 \begin{coldinglist}{42}
    \item three French hens
    \item two turtle doves
    \item and a partridge in a pear tree
 \end{coldinglist}
\end{coldinglist}
```

Example 11-2-7

More complicated color support can be obtained in the framework of the colortbl package, which allows you to produce colored tables (see Section 11.3) or the beamer class, which makes color slides (see Section 11.4).

followed by a number. This number describes the percentage of this color to use in the mix, with the remainder being white.

Example 11-2-13

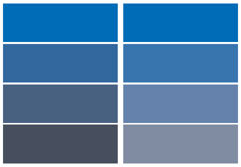


```
\usepackage{xcolor}
\newcommand\blob[1]{{\color{#1}\rule{1.5cm}{5mm}}}
\blob{blue} \blob{blue!75} \\ \blob{blue!50} \blob{blue!25}
```

Tone and shade

What we see in this example is actually an abbreviation of the more general syntax for mixing colors: if the second color in the mix is not white, you have to specify it as well by adding it to the right, again separated by an exclamation mark. The next example shows the mixing of blue with black (called adding tone) and gray (called shading).

Example 11-2-14



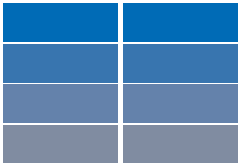
```
\usepackage{xcolor}
\newcommand\blob[1]{{\color{#1}\rule{1.5cm}{5mm}}}
\blob{blue}          \blob{blue}\\
\blob{blue!75!black} \blob{blue!75!gray}\\
\blob{blue!50!black} \blob{blue!50!gray}\\
\blob{blue!25!black} \blob{blue!25!gray}
```

Colorful mix

It is also possible to mix more than two colors in this way, but you have to understand how the algorithm works to do it successfully. Assume you have the three colors in individual buckets and some empty buckets for mixing. You mix the first two colors according to the specified percentage into a free bucket. That gives you a new color in that bucket. Then you use this color and mix it with the third color again into a free bucket, etc.

If you want to mix several colors with a specific percentage in the final mix, that can still be quite tricky. The next example reimplements the mix of blue and gray (which is a 50% mix of black and white) from the previous example. Here it is clearly simpler to first mix black and white and then blue to obtain the same results as before.

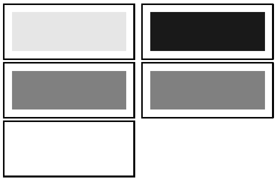
Example 11-2-15



```
\usepackage{xcolor}
\newcommand\blob[1]{{\color{#1}\rule{1.5cm}{5mm}}}
\blob{blue}                    \blob{blue}        \\
\blob{white!50!black!25!blue}  \blob{blue!75!gray}\\
\blob{white!50!black!50!blue}  \blob{blue!50!gray}\\
\blob{white!50!black!75!blue}  \blob{blue!25!gray}
```

It is also possible to specify the complement of a color or color mix with this syntax, by putting a minus sign before the specification. The complement is the color that, if combined with the original color, yields white. However, in the example below, mixing the colors `test` and `anti` yields gray due to the fact that each of the colors in the mix consists of 50% white. Only the extended specification in the third row (explained afterwards) allows us to use 100% of each color, i.e., combine them.

Example 11-2-16



```
\usepackage{xcolor}
\colorlet{test}{yellow!90} \colorlet{anti}{-test}
\newcommand\blob[1]{\fbox{\color{#1}\rule{1.5cm}{5mm}}}
\blob{test}                \blob{anti}        \\
\blob{test!50!anti}        \blob{gray}        \\
\blob{rgb,1:test,1;anti,1}
```

To draw attention to individual rows of a table, we can put a band of color behind them (Color Plate XVI e):

Table title

Description	**Column 1**	**Column 2**
Row one	mmmmm	mmmm
Row two	mmmm	mmm
Row three	mmmmm	mmmmm
Row four	mmmmm	mmmm
Totals	mmmmm	mmmmm

```
\usepackage{colortbl}
\newcommand\panel[1]{\multicolumn{1}%
             {>{\columncolor{magenta}}#1}}
\begin{tabular}{lrr}
  \large\textbf{Table title}\\[2mm]
  \textbf{Description}
          & \textbf{Column 1}
             & \textbf{Column 2}\\[1mm]
  Row one & mmmmm & mmmm \\
  Row two &  mmmm & mmm  \\
  \panel{l}{Row three}
          & \panel{r}{mmmmm}
             & \panel{r}{mmmmm} \\
  Row four& mmmmm & mmmm  \\ \cline{2-3}
  Totals  & mmmmm & mmmmm
\end{tabular}
```

Example 11-3-13

But we can do even better: color the whole table, and leave the row to be emphasized with a white background (Color Plate XVI f):

Table title

Description	**Column 1**	**Column 2**
Row one	mmmmm	mmmm
Row two	mmmm	mmm
Row three	mmmmm	mmmmm
Row four	mmmmm	mmmm
Totals	mmmmm	mmmmm

```
\usepackage{colortbl}
\newcommand\panel[1]{\multicolumn{1}%
             {>{\columncolor{white}}#1}}
\colorbox{magenta}{%
\arrayrulecolor{black}
\begin{tabular}{lrr}
  \large\textbf{Table title}\\[2mm]
  \textbf{Description}
         & \textbf{Column 1}
             & \textbf{Column 2}\\[1mm]
  Row one & mmmmm & mmmm \\
  Row two &  mmmm & mmm  \\
  \panel{l}{Row three}
          & \panel{r}{mmmmm}
             & \panel{r}{mmmmm} \\
  Row four& mmmmm & mmmm \\ \cline{2-3}
  Totals  & mmmmm & mmmmm
\end{tabular}}
```

Example 11-3-14

This is completely analogous to the previous example except that the `\columncolor` command now uses the color white, while the `\colorbox` at the beginning makes the whole table magenta.

Now we look at ways to highlight columns rather than rows. We use the \columncolor command to specify the color of the columns (Color Plate XVI g):

Table title

Description	**Column 1**	**Column 2**
Row one	mmmmm	mmmm
Row two	mmmm	mmm
Row three	mmmmm	mmmmm
Row four	mmmmm	mmmm
Totals	mmmmm	mmmmm

```
\usepackage{colortbl}
\definecolor{Bluec}{cmyk}{.60,0,0,0}
\begin{tabular}{l>{\columncolor{Bluec}}rr}
\large\textbf{Table title}\\[2mm]
\textbf{Description} & \textbf{Column 1}
             & \textbf{Column 2} \\[1mm]
Row one  & mmmmm & mmmm \\
Row two  &  mmmm &  mmm \\
Row three& mmmmm & mmmmm \\
Row four & mmmmm & mmmm \\
Totals   & mmmmm & mmmmm
\end{tabular}
```

Example 11-3-15

Colored panels of this type are often used to highlight connected regions in a table. The blue shade (Bluec) is defined at the beginning with the standard \definecolor command, although we could also have combined it with \columncolor as

```
\columncolor[cmyk]{.60,0,0,0}
```

Another feature often encountered in color work is the color gradient (Color Plate XVI h). Here we use various levels of cyan defined at the start for successive rows. We use the extended mixing possibilities of xcolor to achieve this effect:

Table title

Description	**Column 1**	**Column 2**
Row one	mmmmm	mmmm
Row two	mmmm	mmm
Row three	mmmmm	mmmmm
Row four	mmmmm	mmmm
Totals	mmmmm	mmmmm

```
\usepackage[table]{xcolor}
\definecolor{Cyan}{cmyk}{1,0,0,0.3}
\begin{tabular}{l rr}
\rowcolor{Cyan}\multicolumn{3}{l}
    {\large\textbf{\strut Table title}}\\[2mm]
\rowcolor{Cyan}
\textbf{Description} & \textbf{Column 1}
             & \textbf{Column 2} \\[1mm]
\rowcolor{Cyan!20}Row one   & mmmmm & mmmm  \\
\rowcolor{Cyan!40}Row two   &  mmmm &  mmm  \\
\rowcolor{Cyan!60}Row three& mmmmm & mmmmm\\
\rowcolor{Cyan!80}Row four  & mmmmm & mmmm  \\
\rowcolor{Cyan}   Totals    & mmmmm & mmmmm
\end{tabular}
```

Example 11-3-16

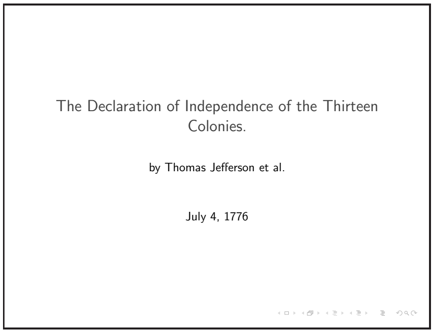
Although this task requires specifying colors for each row, the result can be quite pleasing. This technique is certainly one of those most often used to produce attractive and easily readable tabular material.

One might expect to be able to achieve the same effect by defining a color series and stepping it through each row. However, as it turns out, this approach results in the color changing for every cell: due to the implementation, the color expression is evaluated each

11.4.2 Your first slides

The beamer class comes with lengthy documentation, example files, and a lot of ready-made templates for different colors and layouts. The following example shows the default output. It is difficult to choose the right layout for the presentation—when people are more impressed by the fancy layout than by the contents, then there is something wrong! For a first-time user, it is sensible to use some of the predefined themes of beamer, and to attempt to write your own only after gaining some experience with this class.

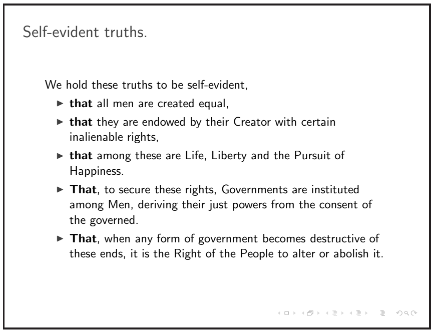
Let us start with a simple pair of slides:



The Declaration of Independence of the Thirteen Colonies.

by Thomas Jefferson et al.

July 4, 1776



Self-evident truths.

We hold these truths to be self-evident,

- **that** all men are created equal,
- **that** they are endowed by their Creator with certain inalienable rights,
- **that** among these are Life, Liberty and the Pursuit of Happiness.
- **That**, to secure these rights, Governments are instituted among Men, deriving their just powers from the consent of the governed.
- **That**, when any form of government becomes destructive of these ends, it is the Right of the People to alter or abolish it.

```
\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
 of the People to alter or abolish it.
 \end{itemize}
\end{frame}
```

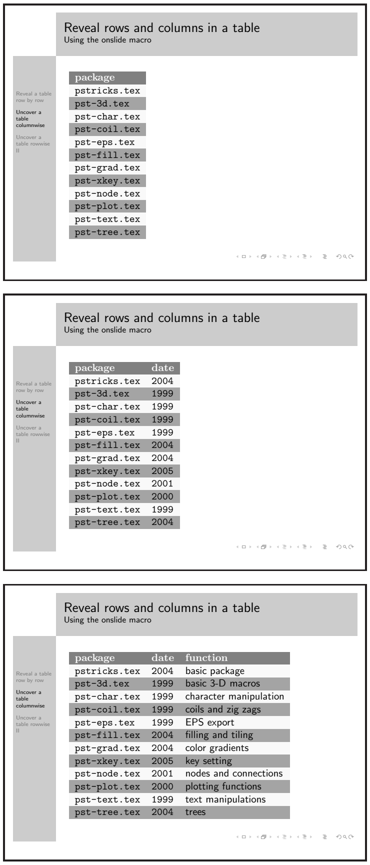
Example 11-4-1

We can change appearance of the slides by choosing variants in five style levels for beamer: the theme, the outer layout, the inner layout, the color theme, and the font theme. In each case you can use the standard LaTeX `\usepackage` mechanism by preceding the style name with the word `beamertheme`, `beameroutertheme`, `beamerinnertheme`, `beamercolortheme`, or `beamerfonttheme` respectively.

Table 11.4 lists the predefined styles that come with beamer. These themes are not official, and their contents and layout depend on what users have contributed to the community.

In the next step we choose the Malmoe main theme; this is just a name for the theme and not the official layout of the Swedish university!

the end of the last column, the use of \onslide without a specification ensures that the first column on the next row is once more shown normally, so that the whole first column is seen (the last slide is also shown in Color Plate XVI x).



Reveal rows and columns in a table
Using the onslide macro

Reveal a table row by row
Uncover a table columnwise
Uncover a table rowwise II

package
pstricks.tex
pst-3d.tex
pst-char.tex
pst-coil.tex
pst-eps.tex
pst-fill.tex
pst-grad.tex
pst-xkey.tex
pst-node.tex
pst-plot.tex
pst-text.tex
pst-tree.tex

Reveal rows and columns in a table
Using the onslide macro

Reveal a table row by row
Uncover a table columnwise
Uncover a table rowwise II

package	date
pstricks.tex	2004
pst-3d.tex	1999
pst-char.tex	1999
pst-coil.tex	1999
pst-eps.tex	1999
pst-fill.tex	2004
pst-grad.tex	2004
pst-xkey.tex	2005
pst-node.tex	2001
pst-plot.tex	2000
pst-text.tex	1999
pst-tree.tex	2004

Reveal rows and columns in a table
Using the onslide macro

Reveal a table row by row
Uncover a table columnwise
Uncover a table rowwise II

package	date	function
pstricks.tex	2004	basic package
pst-3d.tex	1999	basic 3-D macros
pst-char.tex	1999	character manipulation
pst-coil.tex	1999	coils and zig zags
pst-eps.tex	1999	EPS export
pst-fill.tex	2004	filling and tiling
pst-grad.tex	2004	color gradients
pst-xkey.tex	2005	key setting
pst-node.tex	2001	nodes and connections
pst-plot.tex	2000	plotting functions
pst-text.tex	1999	text manipulations
pst-tree.tex	2004	trees

Example 11-4-11

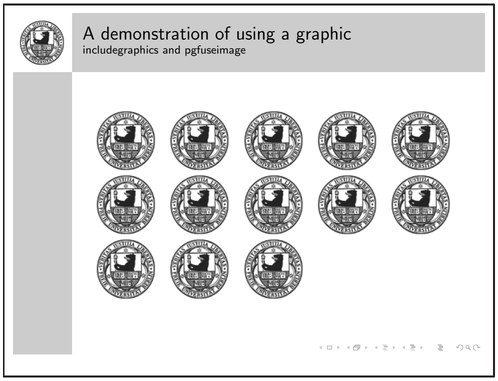
```
\documentclass[xcolor=table]{beamer}
\usetheme{Malmoe}
\useoutertheme{sidebar}
\usecolortheme{dove}
\newcommand\bfrm[1]
           {\textbf{\textrm{\textcolor{white}{#1}}}}
\section{Reveal a table row by row}
\begin{frame}
  \frametitle{Reveal rows and columns in a table}
  \framesubtitle{Using the pause macro}
  ...
\end{frame}
\section{Uncover a table columnwise}
\begin{frame}
 \frametitle{Reveal rows and columns in a table}
 \framesubtitle{Using the onslide macro}
 \rowcolors[]{1}{blue!40}{yellow!20}
 \begin{tabular}{>{\ttfamily}l<{\onslide<2->}|%
   >{\ttfamily}l<{\onslide<3->}l<{\onslide}@{}}
 \rowcolor{gray}
   \bfrm{package}&\bfrm{date}&\bfrm{function} \\
  pstricks.tex & 2004 & basic package         \\
  pst-3d.tex   & 1999 & basic 3-D macros      \\
  pst-char.tex & 1999 & character manipulation\\
  pst-coil.tex & 1999 & coils and zig zags    \\
  pst-eps.tex  & 1999 & EPS export            \\
  pst-fill.tex & 2004 & filling and tiling    \\
... further code omitted ...
```

\onslide can also be used to show specific rows of a table, as we saw earlier with \pause. The following example shows the third and fifth slides of the frame. Note that in the example the \onslide commands are added at the end of the rows (affecting the next) and not at the beginning, as that would trigger the coloring of the row.

```
\documentclass[xcolor=table]{beamer}
\usetheme{Malmoe}  \useoutertheme{sidebar}  \usecolortheme{dove}
\newcommand\bfrm[1]{\textbf{\textrm{\textcolor{white}{#1}}}}
\section{Reveal a table row by row}     \begin{frame}  ... \end{frame}
\section{Uncover a table columnwise}    \begin{frame}  ... \end{frame}
\section{Uncover a table rowwise II}
\begin{frame}
  \frametitle{Reveal rows and columns in a table}
```

```
\includegraphics<overlay spec.> [key/vals] {file name}
\pgfdeclareimage{key/vals}{beamer name }{file name}
\pgfuseimage{key/vals}{beamer name}
```

The following example shows both ways of using a graphic. The screenshot is the thirteenth slide, which is easy to control because each line has five pictures. The automatic slide control is done by the option `<+->` together with the `\only` and `\includegraphics` macros.

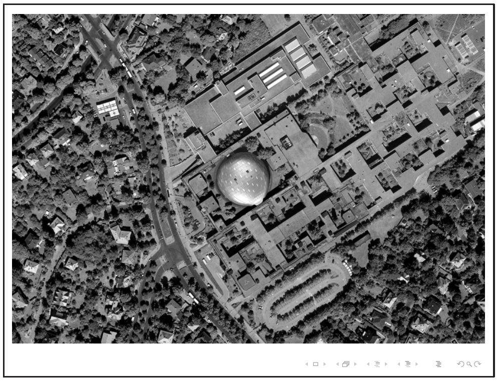


A demonstration of using a graphic
includegraphics and pgfuseimage

```
\documentclass{beamer}  \usetheme{Malmoe}
\useoutertheme{sidebar} \usecolortheme{dove}
\pgfdeclareimage[width=2cm]{fu}{fu-berlin}
\newcommand\FU{\only<+->{\pgfuseimage{fu}}}
\newcommand\fu
   {\includegraphics<+->[width=2cm]{fu-berlin}}
\logo{\includegraphics[width=1.5cm]{fu-berlin}}
\begin{frame}
 \frametitle{A demonstration of using a graphic}
 \framesubtitle{includegraphics and pgfuseimage}
 \FU \fu \FU \fu \FU\par \fu \FU \fu \FU \fu\par
 \FU \fu \FU \fu \FU
\end{frame}
```

Example
11-4-34

Often a full-screen graphic is needed, which is possible with an empty frame (keyword `plain`) and filling the background canvas with the graphic.



```
\documentclass{beamer}  \usetheme{Malmoe}
\useoutertheme{sidebar} \usecolortheme{dove}
\setbeamertemplate{background canvas}{%
     \includegraphics[width=\paperwidth]%
                     {fu-berlin-air}}
\begin{frame}[plain]
\end{frame}
```

Example
11-4-35

This image shows the main campus of the Free University of Berlin and is courtesy of Foster & Partners.

11.4.8 Managing your templates

The beamer class is totally driven by templates, and nearly everything can be overwritten or simply defined by the user. In general there are three kinds of templates:

General Index

Symbols

- ! syntax (xcolor), *731, 732*
- \! (LilyPond), 665
- !! syntax (xcolor), 735
- !!+ syntax (xcolor), 735
- !![*num*] syntax (xcolor), 732, *735*
- "..." syntax
  - (abc), *608*
  - (pic), *19*
- ' syntax
  - (LilyPond), *661*, 662
  - (MusiXTeX), 592
  - (abc), *603*
- '' syntax
  - (LilyPond), *661*, *665*
  - (MusiXTeX), 592
- ''' syntax
  - (LilyPond), *661*, *663*, *665*
  - (MusiXTeX), 592
- \((pst-pdf), 800
- (" syntax (M-Tx), *655*
- (...) syntax
  - (LilyPond), ***663***, *664, 665*
  - (M-Tx), *654, 655*
  - (PMX), ***634***, *635–638*, 648
  - (abc), *607, 608*
- (2 syntax (abc), *605*
- (3 syntax (abc), *605*
- (4 syntax (abc), *605*
- (5 syntax (abc), *605*
- (6 syntax (abc), *605*
- (7 syntax (abc), *605*
- (8 syntax (abc), *605*
- (~...)~ syntax (M-Tx), 655
- \) (pst-pdf), 800
-)(syntax (M-Tx), 655
- * syntax (cwpuzzle), ***704***, *705*
- + syntax
  - (PMX), ***623***, 624, **625**
  - (m-ch-en), 544
  - (texmate), *683*
- , syntax
  - (LilyPond), 661, 662
  - (PMX), *624*, **625**
  - (abc), *603*
- - syntax
  - (LilyPond), *663*
  - (M-Tx), *655*
  - (PMX), ***623***, 624, **625**, *628*
  - (abc), *607, 608*, *611*
  - (m-ch-en), 544
  - (xcolor), *731*, 732
- \- (circ), 579
- -- syntax
  - (LilyPond), *665*
  - (M-Tx), *655*
- . syntax
  - (MusiXTeX), 594
  - (PMX), *624*, **625**
  - (abc), *607*
  - (cwpuzzle), *704, 705*

- . syntax *(cont.)*
  - (printsudoku), *710*
  - (sudoku), *709*
  - (xcolor), *733*
- \. (circ), 579
- .PE syntax (pic), 17, 583
- .PS syntax (pic), 17, 583
- .c syntax (pic), *19*
- .n syntax (pic), *19*
- .ne syntax (pic), *19*
- .nw syntax (pic), *19*
- .se syntax (pic), *19*
- .sw syntax (pic), *19*
- .| syntax (LilyPond), *661, 662*
- .||. syntax (LilyPond), *661, 662*
- / syntax (abc), *608*
- : syntax
  - (LilyPond), *661, 662*
  - (PMX), 631
  - (xcolor), *732*
- :: syntax
  - (M-Tx), 654
  - (abc), *603*
- :| syntax
  - (LilyPond), *661, 662*
  - (M-Tx), 654
  - (abc), *603, 604*
- :|: syntax (LilyPond), *661, 662*
- ; syntax (xcolor), *732*
- < syntax
  - (M-Tx), *658*
  - (MusiXTeX), 592
  - (PMX), *624*, **625**
  - (abc), *604, 605*
- <. syntax (M-Tx), *658*
- <...> syntax (LilyPond), ***663***, *665*
- << syntax (abc), *604, 605*
- <<...>> syntax (LilyPond), *664, 665*
- <<< syntax (abc), *604*
- = syntax
  - (MusiXTeX), 592
  - (abc), *605*
- > syntax
  - (M-Tx), *658*
  - (MusiXTeX), 592
  - (PMX), *624*, **625**
  - (abc), *604, 605*
  - (colortbl), 751
- \> (LilyPond), 665
- >. syntax (M-Tx), *658*
- >> syntax (abc), *604, 605*
- >>> syntax (abc), *604*
- ? syntax (PMX), *629*
- \[(texmate), **680**, ***681***, 682, 683, 686
- [...] syntax
  - (LilyPond), ***663***, *664, 665*
  - (M-Tx), 654
  - (PMX), **631**, *632*, *634*
  - (abc), *608*
  - (cwpuzzle), *704, 705*
  - (texmate), ***680***, 681–683, *686, 687*
- [...]/ syntax (LilyPond), *664*
- [1 syntax (abc), *603, 604*
- [2 syntax (abc), *603, 604*
- [j syntax (PMX), *633*
- \# (texmate), *681–683*
- & syntax (MusiXTeX), 591, 596
- ^ syntax
  - (LilyPond), *663*
  - (MusiXTeX), 592, **593**
  - (abc), *605*, *607*
  - (chemsym), 517
- ^^ syntax (abc), *605*
- ~ syntax
  - (M-Tx), 657
  - (abc), *607*
  - (colortbl), 751
- \ syntax (abc), *604*, *608*
- {" syntax (M-Tx), 655
- {...} syntax
  - (M-Tx), **655**, *657*
  - (abc), *607*
- {~...}~ syntax (M-Tx), *655*
- {} syntax (cwpuzzle), **704**, *705*
- }{ syntax (M-Tx), **655**, *657*
- _ syntax
  - (LilyPond), *663, 664*
  - (M-Tx), *655*
  - (MusiXTeX), 592, **593**
  - (abc), *605*, *611*
  - (chemsym), 517
- __ syntax (abc), *605*
- \] (texmate), 680, *681*
-]-[syntax (PMX), *632*
-][syntax (PMX), *632*
-]j syntax (PMX), *633*
- ‘ syntax
  - (MusiXTeX), 592
  - (dvips), *35*
- | syntax, 668
  - (LilyPond), *661, 662*
  - (M-Tx), 654, 657
  - (MusiXTeX), 591, 596
  - (abc), ***601***, ***603, 604***, *605*, *607*, ***608***
  - (cwpuzzle), *704, 705*
  - (sudoku), *709*
  - (texmate), ***680***, *681*, *683*, *686*
- |. syntax (LilyPond), *661, 662*

|: syntax
  (LilyPond), *661, 662*
  (M-Tx), 654
  (abc), *603*
|] syntax
  (M-Tx), 654
  (abc), ***601***, ***603, 604***, *605*, *607*, ***608***
|| syntax
  (LilyPond), *661, 662*
  (M-Tx), 654
  (abc), *603*
0-0 syntax (texmate), *686*
0-0-0 syntax (texmate), *683*
1,4-dibromobenzene, *521*, *523*
10pt option (beamer), 753
12pt option (beamer), 753
14pt option (beamer), 753
17pt option (beamer), 753
20pt option (beamer), 753
3-D, *see META and* PSTricks *index*
8pt option (beamer), 753
9pt option (beamer), 753

@

@+ syntax (M-Tx), ***658***, *659*
@- syntax (M-Tx), 658
@< syntax (M-Tx), 658
@= syntax (M-Tx), 658
@> syntax (M-Tx), 658
@^ syntax (M-Tx), 658
@v syntax (M-Tx), 658

A

A syntax (PMX), *630*
\A (circ), ***577***, *581*
a syntax (PMX), 625, 631
Aa syntax (PMX), 643
Ab syntax (PMX), *632*, **643**
abbreviations, scientific texts, 513
.abc file extension, xxxi
abc env. (abc), *612*, *614, 615*
abc language, xxviii, **600–615**, 654
abc package, 612–615
abc notation system, *see* music scores (abc2mtex)
abc2midi program, 610, 648
abc2mtex program, 590, **600–612**, 662
\abcinput (abc), *612*, 615
abcm2ps program, 602, 610, 611, 614, 615, 617
abcPlus language, 600, 609, 610, 617, 648
.abcplus file extension, xxxi
Abp syntax (PMX), *633*
absorption, color, 717
accents (musical), *592*, *607*
  LilyPond, *663*
\acciaccatura (LilyPond), ***663***, *664*
accidentals (musical)
  attaching to note names, 622
  examples, *592*
  positioning, *624*, 628
  symbols, *605*
Acrobat Distiller program, 797, 798
actions, slides, *770*
active option (pst-pdf), 800
\ACtoDC (circ), *578*
Ad syntax (PMX), 643
\adamantane (ccycle), *531*
additive color space, 715
addpgf key (chessboard), *669*
addpieces key (chessboard), *669*
ADJ syntax (m-ch-en), **544**, 545
Adobe Acrobat program, 21
Adobe Reader program, 12, 804, 817
Adobe Illustrator program, 1, 4, 21
Adobe Photoshop program, 17
Ae syntax (PMX), 643
\afterb (texmate), *686*
\afterno (texmate), *686*
\afterw (texmate), 686
againcovered key (beamer), 768
\againframe (beamer), **759**, *761*
\ahead (texmate), *681*, ***682***, *683*
AI syntax (PMX), 643
Ai syntax (PMX), 643
AlDraTex package, 15
\alert (beamer), 761, *771*, **790**, *791*
alertblock env. (beamer), **778**, *779*
algorithmic display drawings, 5
algorithmic structural drawings, 5
alignment
  nucleotide sequences, 548–550
  peptide sequences, 548–550
aliphat package, 520, **532**
aliphatic compounds, 532, 533
all option (beamer), 753
\allabreve (MusiXTEX), 592
allegro (musical), *646*
allegro vivace (musical), *644*
\allmatchspecial
  (texshade), *548*
  (textopo), *552*
\allowdisplaybreaks (beamer), 759
allowdisplaybreaks key (beamer), 759
allowframebreaks key (beamer), **759**, *782*
allowsframebreaks key (beamer), 759
alltt package, 790
\alt (beamer), 768
altenv env. (beamer), 770
alto syntax (LilyPond), **661**, *664*
\altoclef (MusiXTEX), 592

\Amp (circ), *578*
\ampere (Slunits), 514, *515, 516*
\amperemetresecond (Slunits), *516*
amsmath package, 752, 753, 759
amssymb package, 515
amstex package, 517
amsthm package, 753
\analysistop (texmate), 686
\AND (circ), *578*
angle key (graphicx), **28**, *31, 32*
\animate (beamer), 774
\animatevalue (beamer), 774
animation, *see META index*
animation, slides, 774
annotations, *see also* commentaries
    chemical formulas, *547*
    chess, 675
    music scores, 657, 658
    timing diagrams, 573
anthracene derivatives, *525*
\anthracenev (carom), 524, ***525***
Ap syntax (PMX), *636*, **643**
\appendix (beamer), 779
\applyshading (textopo), *552*
Ar syntax (PMX), 643
\Arc (curve2e), 47, *50*
arc (pic), 17
arcs (Feynman diagrams)
    edges, *572*
    segments with arrows, *560*
aromatic carbocycles, *525*
\arpeggio (MusiXTEX), 592
arpeggio (musical), *629*
array env., 8, 737
array package, 737, **764**
\arrayrulecolor (colortbl), **741**, *742*, ***745***, *746*, *749–751*
\arrayrulewidth rigid length, *742*
arrow (pic), 17
\ArrowArc (axodraw), 558, *560*
\ArrowArcn (axodraw), 558
\ArrowLine (axodraw), 558, *559–561*
arrows
    Feynman diagrams, *559–561*
    styles, *44*
    timing diagrams, 575
art graphics, **4**, 22
article option (beamer), 753
article document class, xxxi
AS syntax (PMX), 643
As syntax (PMX), 643
aspect ratio, keeping, 29, *31*, *38*
\at (circ), 580
atan (pic), 19
\AtBeginPart (beamer), 780
atom derivation, *539*
atoms, aligning with bonds, *546*
\atpin (circ), **580**, *581*
\atto (Slunits), 515
\author (beamer), *754*, *757*, *761*
\autoBeamOff (LilyPond), *663*
AutoCAD program, 17, 21
automata, *see META and* PSTricks *index*
automata drawings, 15
Av syntax (PMX), 643
axodraw package, 555, **558–561**

B

B syntax (m-ch-en), 542, **544**
b key (beamer), **759**, 781
b syntax (PMX), ***635***, *637*
\B2Text (axodraw), 558
babel package, 515
Bach musical example, *590*, *610*
backgammon, *696*, 697, 698
background syntax (beamer), *794, 795*
background color, documents, *723, 724*, 725
background canvas syntax (beamer), *792*, *795*
\backturn (MusiXTEX), 592
\bar
    (LilyPond), *661, 662*
    (MusiXTEX), **591**, *594–596*, *599*
bar package, 15
bar charts, *see META and* PSTricks *index*
bar codes, *see* PSTricks *index*
bars (musical)
    changes, 654
    double, *603*
    repeats, *603*, 639
    symbols, *603*, 639
    thick, *603*
    thin, *603*
Bars/line: syntax (M-Tx), 652
Bartok musical example, *596*
base units, 514
basic option (circ), **577**, 578
basic duration (musical), 622
\bass (MusiXTEX), *596*
bass syntax (LilyPond), **661**, *665*
\bassclef (MusiXTEX), 592
bb key (graphicx), **28**, 29, *30*
\bbetter (texmate), *680, 681*
\BBox (axodraw), 558
\BBoxc (axodraw), 558
\BCirc (axodraw), 558
beamer option (beamer), 753
beamer document class, xxxi, **752, 753**, 754–758, **759**, 760–796
beamerboxesrounded env. (beamer), 775, *776*, 778
beamercolorbox env. (beamer), 775, ***776***, 777, *794*
\beamergotobutton (beamer), *784, 785*
beamerouterthemesidebar package, 774

beamerpauses counter (beamer), *788*
\beamertemplatearticlebibitems (beamer), *782*
\beamertemplatebookbibitems (beamer), *782*
beams (musical)
  grouping notes, *606*
  jumping staves, *633*
  LilyPond, 663
  M-Tx, 654, 655
  MusiXTeX, 597
  PMX, 631, *632, 633*
  xtuplets, *627, 628*
\becquerel (Slunits), 514
\beforeb (texmate), 686
\beforeno (texmate), 686
\belo (texmate), 683
\benzofuranev (hetarom), 530
\benzofuranevi (hetarom), 530
\benzoxazolev (hetarom), 530
\benzoxazolevi (hetarom), 530
bes syntax (LilyPond), **662**, *663*
\betteris (skak), *678*
Bézier curves
  cubic, *47*
  quadratic, 46, *47*
\bfseries (chessfss), 671
bg key (beamer), *776*, *778*, *794*
bg package, 696–698
\Bi (chemsym), *518*
\bibitem (beamer), *782*
bibliographies, slides, *782*
bibtex program, 801, 806
\bicycheph (ccycle), *531*
\bicychepv (ccycle), *531*
\bid (tlgc), *702*
bidding env. (bridge), 699, 701, ***702***
\bigboard (bg), 697
bigger option (beamer), 753
bioinformatics, *see also* scientific texts
  membrane protein topology plots, *551–553*
  nucleotide sequences
    aligning, 548–550
    highlighting, *548–550*
    sequence fingerprints, *550*
    shading, 548–550
  peptide sequences
    aligning, 548–550
    highlighting, *548–550*
    sequence fingerprints, *550*
    shading, 548–550
\bishop (chessfss), *672*
\black (igo), **691**, *692–695*
black syntax (xcolor), 722, 726
"black box" drawings, 3, 4
black-and-white, 721
\blackbar (bg), *696*, **697**
\blackcube (bg), 697
\blackname (texmate), *683*
\blackonmove (bg), 697
\blackpoint (bg), *696*
\blackstone (igo), *695*
blending color, 737
\BLens (circ), ***580***, *581*
blobs (Feynman diagrams), 566
block env. (beamer), *777*, ***778***, *779*
block environments, slides, *778, 779*
block body syntax (beamer), *778*
block title syntax (beamer), *778*
blocks (musical), 622
blue syntax (xcolor), ***722***, *723*, **726**, 727
\bluefbox (tlgc), *26*
bm2font program, 7
\bmove (skak), 679
\bname (texmate), **685**, 686
board games, *see* backgammon, *see* chess, *see* Go
\boardcaption (bg), *696*, **697**, *698*
\boardfont (chessfss), *673*
boardfontencoding key (chessboard), *669*
\boardsymbol (chessfss), 673
bodyCol syntax (beamer), *776*
bonds (chemical)
  aligning atoms or molecules, *546*
  between C atoms, *542*
  derivation, *539*
  description, 543
  directions, *535, 536*
  identifiers, 544
  modifiers, *522*
border key (chessboard), *669*
\bornane (ccycle), *531*
\bottomdiagramnames (texmate), 686
bounding box
  aspect ratio, keeping, 29
  clipping graphics to, 29, *30*
  comments, 25, 28
  draft mode, 25, 29, *30*
  final mode, 25
  fitting to graphics, *26, 27*
  height, 28, 29, *32*
  \includegraphics syntax, 28–32
  resizing, *27*
  rotated material, hiding, 25
  rotating, *27*, *31, 32*
  scaled material, hiding, 25
  scaling, *27*, 29
  specifying, 28, *30*
  trimming space, 28, *30*
  viewports, 28, *30*
  width, 28, 29
BoundingBox (PostScript), **25**, 26, **28**, 34, 35
box (pic), 17, *19*

- `box` option (circ), 577
- `\Boxc` (axodraw), 558
- boxes, *see also* frames
  - colored, in documents, *723, 724*
  - slides, text in, *775, 776*
- `\boxit` (MusiXTeX), 592
- `\bracket` (MusiXTeX), 592
- `\break` (LilyPond), *661*
- `\breve` (LilyPond), *663*
- bridge package, 699–702
- bridge (card game)
  - bidding, *702*
  - dealing, 699, *700, 701*
- `bridge.tex` file (bridge), **699**, 700
- broken musical rhythms, *604*
- `brown` syntax (xcolor), 726
- `\BSplit` (circ), ***580***, *581*
- `\BText` (axodraw), 558
- `\BTri` (axodraw), 558
- `\BUF` (circ), *578*
- `\bundle` (circ), 579
- `\bupperhand` (texmate), *680*
- `\bzdrh` (carom), ***521***, 523, 524, ***525***, ***534***, *535, 536*
- `\bzdrv` (carom), ***521***, *522*, 524, ***525***, *536*

C

- `C` syntax
  - (PMX), *639*
  - (m-ch-en), 544
- `\C` (circ), *577*
- `c` key (beamer), **759**, 781
- `c` option (beamer), 753
- `\C2Text` (axodraw), 558
- `C:` syntax
  - (M-Tx), *656*
  - (abc), *608*, *610*
- `\ca` (MusiXTeX), **593**, *594, 595*
- `\caesura` (MusiXTeX), 592
- calc program, 21
- calculations, drawing tools for, 1
- calendars, *see* PSTricks *index*
- `\Cam` (circ), ***580***, *581*
- `\candela` (SIunits), 514
- captions
  - chess, 684–686
  - Go board, 694
- carbocycles, *524*
- carbocyclic compounds, *527*
- `CARBON` syntax (m-ch-en), *541*, ***542***
- `\CArc` (axodraw), 558, *560*
- card games
  - bridge
    - bidding, *702*
    - dealing, 699, *700, 701*
  - suits, representing, 698, *699*
- caret (`^`), sharp symbol, *605*
- carets (`^^`), double flat symbol, *605*
- carom package, 520, **524**
- `CB` syntax (m-ch-en), *541*
- `\cbezier` (pict2e), 46, *47*
- `\CBox`
  - (axodraw), 558
  - (tlgc), *733*
- `\CBoxc` (axodraw), 558
- `\cbreath` (MusiXTeX), 592
- `\cc` (circ), **579**, *581*
- `\cca` (MusiXTeX), **593**, *594, 595*
- `\ccccl` (MusiXTeX), 592, **594**
- `\ccccu` (MusiXTeX), 592, **594**
- `\cccl` (MusiXTeX), 592, **594**
- `\cccu` (MusiXTeX), 592, **593**
- cchess package, 687–690
- `cchessboard.tex` file (cchess), 688
- `\CCirc` (axodraw), 558
- `\ccl` (MusiXTeX), 592, **593**, *594, 595*
- `\ccu` (MusiXTeX), 592, **593**, *594, 595*
- ccycle package, 520, **530**
- `\cdfl` (MusiXTeX), 592
- `cdot` option (SIunits), 515
- `\cdsh` (MusiXTeX), 592
- `\Cel` (circ), *577*
- `\cellcolor` (colortbl), **741**, 748, *749*
- cells (table), color, *741*
- `\cellsize`
  - (createsudoku), *711*
  - (printsudoku), *710*
  - (solvesudoku), *711*
- `\celsius` (SIunits), 514
- `center` key (beamer), 777
- `\centerto` (circ), *581*
- `\centi` (SIunits), 515, *516*
- `.cfg` file extension (graphics), 25
- `\cfl` (MusiXTeX), 592
- CGM language, 13
- CGM (Computer Graphics Metafile), 13
- CGM-Open Consortium, 13
- `\CH` (chemsym), 517
- `\chair` (ccycle), *531*
- `\changeunitlength`
  - (xymtexps), ***538***, *539, 540*
  - (xymtex), *538*
- character-based diagrams and pictures, 13
- charges on atoms, *524*, *526*
- charts, *see also* META *and* PSTricks *index*, *see* graphs
- ChemDraw program, 21
- `chemeqn` env. (chemist), *540*
- `\chemical` (m-ch-en), *541*, ***542***, *543–545*, ***546***, *547*
- chemical bonds, *see* bonds (chemical)

- chemical formulas, *see also* scientific texts
  - 1,4-dibromobenzene, *521*, *523*
  - aliphatic compounds, 532, 533
  - annotation, *547*
  - anthracene derivatives, *525*
  - aromatic carbocycles, *525*
  - atom derivation, *539*
  - bonds
    - aligning atoms or molecules, *546*
    - derivation, *539*
    - description, 543
    - directions, *535, 536*
    - identifiers, 544
    - modifiers, *522*
  - carbocycles, *524*
  - carbocyclic compounds, *527*
  - charges on atoms, *524*, *526*
  - combinations, 543
  - command syntax, 520–522
  - configuration, *540*
  - conformations, *540*
  - conventions, 520
  - cyclohexane chair forms, *531*
  - decaline derivatives, *525*
  - definitions, *543*
  - derivation, 539
  - elements, symbols for, 512
  - endocyclic bonds, *523*
  - ethylene derivatives, *533*
  - four-member carbon cycles, *528*
  - furanoses, *532*
  - fused five- and six-member rings, *530*
  - fused rings, *524*
  - fusing ring units, 536
  - handidness of substituents, *522*, *531*, 535
  - heterocyclic compounds, 528–530
  - indane derivatives, *528*
  - inside paragraphs, *547*
  - lower-order cycles, *527, 528*
  - Periodic Table of the Elements, 519
  - phenanthrene derivatives, *525*
  - polymethylene commands, 538
  - PostScript output, 537, 538
  - pyranoses derivatives, *532*
  - reaction schemes, *540*
  - stereochemical compounds, 530–532
  - stereochemistry effects, 538
  - steroid derivatives, *525, 526*
  - structures
    - atoms, aligning with bonds, *546*
    - basic commands for, *541, 542*
    - bond identifiers, 544
    - bonds, 543
    - bonds, aligning atoms or molecules, *546*
    - chemical bonds, *542*
- chemical formulas *(cont.)*
    - combinations, *544, 545*
    - combining, *534*
    - complex, *534, 535*
    - libraries of, 543
    - molecules, aligning with bonds, *546*
    - moving, *544, 545*
    - positioning, *544, 545*
    - reaction equations, *545*
    - rotating, *544, 545*
    - substructures, *543*
  - substitution derivation, *539*
  - tetrahedral compounds, *532*, 533
  - tetrahedron carbon configurations, *533*
  - tetraline derivatives, *525*
  - three-member carbon cycles, *528*
  - tricyclic carbocycles, *525*
  - trigonal units, 532, *533*
- chemical symbols, 517, *518*
- chemist package, 537, 540
- chemstr package, 520
- chemsym package, 512, **517**, **518**, 519
- chess
  - $ (dollar sign), comment indicator, *678*
  - board
    - annotations, 675
    - displaying, 674, *675*, 676, 677
    - hiding pieces, 676
    - highlighting, *676*
    - next move indicator, 676
    - printing, 675
    - size, 675
    - specifying, 674–677
  - captions, 684–686
  - Chinese, 687, *688–690*
    - pieces, *688*
  - coloring the board, 668, *669*
  - diagrams
    - adjusting layout, *686, 687*
    - typesetting, 684, *685, 686*
  - documenting a game, 679
  - ending games, *683*
  - FEN (Forsyth-Edwards-Notation), 674
  - fonts
    - Figurine symbols, *671*
    - generic mechanism, 669–673
    - list of, *670*
    - normal characters, 671
    - selecting, *672, 673*
    - switching, *672*
  - informational symbols, *674*
  - moves
    - error detection, *678*
    - printing, 677
    - recording, 675

- chess *(cont.)*
    - specifying, 677, *678*
    - style, changing, *679*
  - nested variations, 679
  - notation
    - commentaries, *681*, 682
    - overview, 680–683
    - threats, *681*
    - variations, *680*, *682, 683*
  - online resources, 687
  - overview, 668
  - setting up position, *684*
  - starting games, *683*
  - titles, 683
- chess package, 668, 677, 680, 687, 690, 691
- `\chessboard` (chessboard), *669*
- chessboard package, **668, 669**, 673
- `\chessevent` (texmate), *683*
- chessfss package, 668, **669–673**, 674, 678, 680
- `\chessopening` (texmate), 683
- `\chl` (MusiXTeX), 592
- chmst-ps package, 537
- chords (musical)
  - abc2mtex, *608*
  - LilyPond, *663*
  - M-Tx, *656, 657*
  - MusiXTeX, *594*
  - PMX, *628, 629*
- `\chu` (MusiXTeX), 592
- CIE (Commission Internationale de l'Eclairage), color spaces, 715
- `\cinnolinev` (hetarom), 530
- `\cinnolinevi` (hetarom), 530
- `\circ`, *39*
- circ package, 576–582
- `\circle`, *43*
  - (curve2e), *49*
  - (pict2e), *43*, *45*, *47*
- `circle` (pic), 17
- `\circle*`, *43*
  - (pict2e), *43*, 45
- `\circleit` (MusiXTeX), 592
- circles
  - drawing, *45*
- `circuit` env. (circ), **578**, *581*
- `\cl` (MusiXTeX), 592, **593**, *599*
- `\clear` (igo), **694**, *695*
- `\cleargoban` (igo), 694
- `\cleargobansymbols` (igo), ***692***, *695*
- clearing, Go board, 694
- `\clef` (LilyPond), **661**, *664, 665*
- clef changes (musical), *639*
- clefs (musical), *592*, 653
- `\cline` (colortbl), 741
- `clip` key (graphicx), 28, **29**, *30*
- clipping graphics to bounding box, 29, *30*
- `clockwise` option (rotating), 42
- `\club`
  - (bridge), *701, 702*
  - (tlgc), *699*
- `\clubsuit`, 698, *699*
- `\Clue` (cwpuzzle), ***705***, 706
- `\clue` (crosswrd), *703, 704*
- `\cluefont`
  - (createsudoku), *711*
  - (printsudoku), *710*
  - (solvesudoku), *711*
- `cmy` option (xcolor), 721
- `cmy` syntax (xcolor), 728, 729
- `cmyk` option (xcolor), 721
- `cmyk` syntax
  - (color), 720
  - (xcolor), 720, *723*, *725*, 727–730
- CMYK (Cyan, Magenta, Yellow, Black) color, **715**, 719
- `\cna` (MusiXTeX), 592
- `\Co` (chemsym), *518*
- `collision` option (chemsym), 517
- color
  - absorption, 717
  - adding tone, *731*
  - additive color space, 715
  - and light, 714
  - and readability, 718
  - black-and-white, 721
  - blending, 737
  - categories of (PostScript), 715
  - color package
    - defining colors, 726–728
    - options, 720–722
    - overview, 719, 720
  - Commission Internationale de l'Eclairage, 715
  - complement, specifying, *731*
  - contrast, 718
  - core model, *732*
  - Crayola colors, 719
  - cultural connotations, 716
  - defining
    - assigning to names, 734, *735, 736*
    - sets of colors, *727*
    - single colors, 726, *727*
  - device color spaces, 715
  - error warnings, 721
  - expressions
    - current color, *733*
    - extended, 732
    - PSTricks, *733*
    - standard, 732
  - Feynman diagrams, 567
  - four-color harmonics, 718
  - Grassman's Law, 714

- color *(cont.)*
  - harmonic color circle, 717
  - harmonies, 717, 718
  - intensity, 718
  - masking, 737
  - mixing, *731*
  - models supported, 719
  - monochrome, 721
  - overview, 719, 720
  - primary colors, 717
  - purity, 718
  - saturation, 717
  - secondary colors, 717
  - series, 734, *735, 736*
  - shading, *731*
  - slides, *see* slides (color)
  - special color spaces, 715
  - spectrum, displaying, *729*
  - subtractive color space, 715
  - symbolic values, 716
  - tables, *see* tables, color
  - text
    - documents, 725
    - slides, *775, 776*
    - tables, *745*, *748*
  - theories, 714, 715
  - three-color harmonics, 718
  - three-color theory, 714
  - tinting, *731*
  - two-color harmonics, 718
  - undefined colors, 721
  - within documents
    - background, *723, 724*, 725
    - colored boxes, *723, 724*
    - lists, *724*
    - mixing colors, *723*, 725
    - named colors, 725
    - portability, 723
    - special concerns, 725
    - specifying by color model, *722*
    - specifying by name, *722*
    - stored boxes, 725
    - tables, 724
    - text inside a box, 725
  - xcolor package
    - color models, 728–730
    - extended specification, 734
    - options, 720–722
    - overview, 719, 720
  - Young-Helmholtz Law, 714
- `\color`
  - (beamer), 788, *789*
  - (colortbl), 741
  - (color), 741, *744, 745*
  - (curve2e), *48–50*
- `\color` *(cont.)*
  - (xcolor), 720, ***722***, *723*, *725*
- `color` key
  - (beamer), *795*
  - (chessboard), *669*
- color package, 719–722, 726, 728, 730, 737
- color models
  - CIE color spaces, 715
  - CMYK (Cyan, Magenta, Yellow, Black), **715**, 719
  - gray, 719
  - HSB (Hue, Saturation, Brightness) color, 715, **719**
  - HSV (Hue, Saturation, Value) color, 715
  - named
    - behavior options, 721
    - in LaTeX documents, 725
    - support for, 719
  - overview, 715
  - RGB (Red, Green, Blue) color, **715**, 719
  - target, specifying, 730
  - xcolor package, 728–730
- `color.cfg` file (xcolor), 720
- `color.pro` file (dvips), 725
- `\colorbox`
  - (color), 743, *744*, *746*, *749*
  - (xcolor), 720, ***723***, *724*, *729*, *733*
- colordvi package, 719
- `coloremph` key (chessboard), *669*
- `\colorlet` (xcolor), **726**, *727*, 730
- `\colorseriescycle` (xcolor), 734
- colortbl package, 720, 721, **737–751**
- `colsep` key (beamer), 777
- `colsep*` key (beamer), 777
- `\column` (beamer), 781
- `column` env. (beamer), *780*, 781
- `\columncolor` (colortbl), **737**, *738*, 739, 741, *746–748*, *750, 751*
- `columns` env. (beamer), *780*, **781**
- columns (table), color, *738*, *747*
- `\columnwidth` rigid length, 33
- comma (,), octave indicator, *603*
- `command` key (graphicx), 29
- commentaries, chess, *681*, 682, *see also* annotations
- Commission Internationale de l'Eclairage (CIE), color spaces, 715
- complementary color, specifying, *731*
- complex numbers, representing, *49*, 50
- complex vertices (Feynman diagrams), 567
- `Composer:` syntax (M-Tx), *651*, **652**
- compound time signatures (musical), *605*
- Comprehensive TeX Archive Network, *see* CTAN
- `compress` option (beamer), 753
- computer generated drawings, 5
- Computer Graphics Metafile (CGM), 13
- `\connection` (circ), *581*
- contrast, 718
- `\conttimingcounter` (timing), 573

- convert program, 806
- \COOH (chemsym), 517
- \copyfromgoban (igo), **694**, *695*
- copying, Go board, 694, *695*
- \copytogoban (igo), **694**, *695*
- \CopyVect (curve2e), *49, 50*
- Corel Draw program, 1
- corollory env. (beamer), 769
- cos (pic), 19
- \coulomb (SIunits), 514
- \coulombpercubicmetrenp (SIunits), *516*
- counterclockwise option (rotating), 42
- \COval (axodraw), 558
- \cql (MusiXTeX), 592
- \cqu (MusiXTeX), 592
- Crayola colors, 719
- \Crdexa (tlgc), *701*
- \crdima (bridge), **699**, *700, 701*
- createsudoku package, 710–712
- crossword env. (crosswrd), *703*
- crosswords
  - {} (curly braces), empty cell indicator, 704, *705*
  - classical puzzles, *705, 706*
  - creating, 702, *703, 704*, 705
  - external puzzle generation, 709
  - fill-in puzzles, *707*
  - layout adjustment, 708
  - number puzzles, 707, *708*
- crosswrd package, 702–704
- CRZ syntax (m-ch-en), 546
- \csh (MusiXTeX), 592
- CTAN (Comprehensive TeX Archive Network)
  - archived files, finding and transferring, 813
  - description, 810
  - files, from the command line, *814*
  - TeX file catalogue, 811
  - web access, 810, 811, *812, 813*, 814
- \CText (axodraw), 558
- \CTri (axodraw), 558
- \cu (MusiXTeX), 592, **593**, *594, 595*
- cubic Bézier curves, *47*
- cultural connotations of color, 716
- curly braces ({})
  - around arguments (musical), 596
  - empty crossword cell indicator, 704, *705*
  - grace notes (musical), *607*
- currentsection key (beamer), 783
- currentsubsection key (beamer), 783
- \Curve
  - (axodraw), 558
  - (curve2e), 47, *48, 49*
- curve2e package, 47–50
- curves
  - Bézier
    - cubic, *47*

- curves *(cont.)*
    - quadratic, 46, *47*
  - drawing, 47, *48–50*
- curves package, 15, 47
- \Cvar (circ), *577*
- cwpuzzle package, **704–708**, 709
- cyan syntax (xcolor), 722, 726
- Cyan, Magenta, Yellow, Black (CMYK) color, **715**, 719
- \cyclobutane (lowcycle), *527, 528*
- cyclohexane chair forms, *531*
- \cyclohexaneh (carom), *523*, ***524***, ***527***, *535*
- \cyclohexanev (carom), *522, 523*, ***524***, ***527***, *535*, ***538***
- \cyclopentaneh (lowcycle), *526, 527*
- \cyclopentanehi (lowcycle), *527*
- \cyclopentanev (lowcycle), *526, 527*
- \cyclopentanevi (lowcycle), *526, 527*
- \cyclopropane (lowcycle), *528*
- \cyclopropaneh (lowcycle), *527*
- \cyclopropanehi (lowcycle), *527*
- \cyclopropanev (lowcycle), ***527***, *539*
- \cyclopropanevi (lowcycle), *527*

D

- D syntax (PMX), *638*
- \D (circ), *577*
- d syntax
  - (M-Tx), *654*
  - (PMX), *624*, **625**
- “d” in integrands, 513
- D"..." syntax (PMX), *638*
- D<...D> syntax (PMX), *638*
- \DANTE (tlgc), *729*
- darkgray syntax (xcolor), 726
- \DashArrowArc (axodraw), 558
- \DashArrowArcn (axodraw), 558
- \DashArrowLine (axodraw), 558
- \DashCArc (axodraw), 558
- \DashCurve (axodraw), 558
- \dashed (circ), 579
- dashed (pic), *19*
- \dashhasheddash (xymtexps), *538*
- \DashLine (axodraw), 559
- \date (beamer), *754*, *757*, *761*
- date in head/foot syntax (beamer), *777*
- DB syntax (m-ch-en), 544
- \Dcap (circ), *577*
- dcolumn package, 737
- \dcqu (MusiXTeX), 592
- dd syntax (PMX), *624*, **625**
- \ddummy (texmate), 682
- \deca (SIunits), 515
- \decaheteroh (hetarom), *529*
- \decaheterohi (hetarom), *529*
- \decaheterov (hetarom), ***529***, *530*
- \decaheterovb (hetarom), *529*

\decaheterovi (hetarom), *529*
\decaheterovt (hetarom), *529*
decaline derivatives, *525*
\decalineh (carom), 524, ***527***
\decalinev (carom), 524, ***527***
\decalinevb (carom), *527*
\decalinevt (carom), *527*
\decamethylene (methylen), *538*
\decamethylenei (methylen), *538*
\deci (SIunits), 515
\DeclareGraphicsExtensions (graphics/graphicx), **33**, *34*
\DeclareGraphicsRule (graphics/graphicx), 29, ***34***, *35*
dedicated drawing tools, *see* drawing tools (dedicated)
.def file extension (graphics/graphicx), 24
\defconsensus (texshade), *548*
define (pic), *19*
\definechemical (m-ch-en), *543*
\definecolor
(color), 743, *747, 748*, *751*
(xcolor), 720, 721, **726**, *727*, 734
\definecolorseries (xcolor), **734**, *735, 736*
\definecolorset (xcolor), ***727***, *728*
definition env. (beamer), 769
definitions env. (beamer), 769
\DEP (MusiXTeX), 592
\depth (graphics/graphicx), 38
depth key (graphicx), 29
derivation, 539
derived units, 514
description env. (beamer), 786
device color spaces, 715
\DFF (circ), *579*
\dhqu (MusiXTeX), 592
dia program, 1, 6
\Diagram (feyn), **556**, *557*
\diagram (texmate), 684
\diagramcache (texmate), 685
\diagrammove (texmate), 686
\diagramnumber (texmate), 686
diagrams, *see also* graphs
character-based, 13
typesetting, 16
\diagramsign (texmate), 685
\diam
(bridge), *702*
(tlgc), *699*
\diamondsuit, 698, *699*
\dimethylene (methylen), *538*
\dimethylenei (methylen), *538*
\ding (pifont), *724*
direction key (beamer), 775
\DirFromAngle (curve2e), *49, 50*
Disable: syntax (M-Tx), 652
displaymath env. (pst-pdf), 800
displaymath option (pst-pdf), 800
dissolves, slides, *774, 775*
diversity package, 549
\DividE (curve2e), 49
\DividECurve (curve2e), *49, 50*
\dmass (circ), *580*
document env., xxxi
documentation, *see also* online resources
command-line interface, *815*
panel interface, *816*
search by name, *815*
search by product, *816*
texdoc, *815*
texdock, *816*
\documentclass, xxxi
dollar sign ($), comment indicator (chess), *678*
\dontindentwhite (bg), 698
\dontshowcube (bg), **697**, 698
\dontshowmoves (bg), 698
\dontshownumbers (bg), 697
\doqu (MusiXTeX), 592
dotted notes (musical), 622
dotted rhythms (musical), *604*
\dottedline (epic), 521
double bars (musical), *603*
double flat symbol (musical), *605*
double quotes ("..."), guitar chords, *608*
\doublerulesepcolor (colortbl), **742**, 751
doublets (musical), *605*
doubly dotted notes (musical), 622
down (pic), *19*
down fermata ornaments (musical), 630
\downbow (MusiXTeX), 592
\downtrio (MusiXTeX), 592
dp key (beamer), 777
dpic program, 583
\dqu (MusiXTeX), 592
DR syntax (m-ch-en), 544
draft key (graphicx), **29**, *30*
draft option
(beamer), 753
(graphics/graphicx), 25
(pst-pdf), 800
DraTex package, 5, 15
drawing graphic objects, *see* graphics languages, *see* manipulating graphic objects
drawing tools (dedicated), *see also* graphics languages
calculations, 1
Corel Draw, 1
dia, 1
for plotting, 2, 17
gnuplot, 17
Maple, 2
Mathematica, 2
MATLAB, 2
Octave, 2

- drawing tools (dedicated) *(cont.)*
    - Octaviz, 2
    - Octplot, 2
    - overview, 1, 2
    - xfig, 1
- drawing types
    - algorithmic display, 5
    - algorithmic structural, 5
    - art graphics
        - choosing a language for, 22
        - description, 4
    - "black box", 3, 4
    - computer generated, 5
    - derived from textual representation, 5
    - free-hand pictures, 3, *4*
    - object-oriented, 4, *5*
    - overview, 3–6
    - photographs, 4
    - self-contained object-oriented, 4
    - single object, 3, 4
- \drumclef (MusiXTeX), 592
- \ds (MusiXTeX), 592, **594**
- \dtetrastereo (aliphat), *533*
- \Dtext (circ), *581*
- \Dtrigonal (aliphat), *533*
- \dtrigonal (aliphat), *533*
- \dtrigpyramid (xymtexps), *540*
- \duevolte (MusiXTeX), 592
- \dummy (texmate), *681*, ***682***
- duration key (beamer), *775*
- duration of musical notes, 622, 662
- Dusty Miller musical example, *608*
- dvi2svg program, 13
- dvipdf option
    - (graphics/graphicx), 24
    - (xcolor), 721
- dvipdf program, 24
- dvipdfm option
    - (graphics/graphicx), 24
    - (pict2e), 43
    - (xcolor), 721
- dvipdfm program, 24, 797, 798, 803
- dvipdfmx option (xcolor), 721
- dvipdfmx program, 797–799, 803, 804, 806
- dvips option
    - (graphics/graphicx), 24
    - (pict2e), 43
    - (xcolor), 721
- dvips program, xxviii, 11, 16, 17, 24, 25, 558, 614, 618, 637, 719, 721, 722, 725, 797–801, 803–806
- dvips.def file (graphics/graphicx), 24
- dvipsnames option (xcolor), **721**, 727
- dvipsone option
    - (graphics/graphicx), 24
    - (xcolor), 721
- dvipsone program, 17, 24
- dvisvg program, 13
- dvisvgm program, 13
- dviwin option
    - (graphics/graphicx), 24
    - (xcolor), 721
- dviwin program, 24
- dynamic key (beamer), 767
- dynamical marks (musical), *638*

E

- e syntax (PMX), 625, *628*
- E: syntax (abc), 602, *608*
- EB syntax (m-ch-en), 544
- \EBox (axodraw), 558
- \ECO (texmate), 683
- edges (Feynman diagrams), *572*
- eepic package, 17, 20, 511, 521, 522
- electrical circuits, *see META and* PSTricks *index*
- electronic box symbols, *578*
- electronics diagrams
    - drawing position, moving, 580
    - electronic box symbols, *578*
    - font for, 576–582
    - gate symbols, *578*
    - integrated circuit symbols, *579*
    - interactive generation, 586
    - junctions, 579
    - m4 macro processor, *583–585*
    - npn transistor, *581*
    - optics, *581*
    - pin connections, 579
    - symbol connections, 579
    - symbols, *577*
    - trigger symbols, *578*
- \elemskip rigid length (MusiXTeX), **595**, 597, 602
- ellipse (pic), 17, *19*
- emphfields key (chessboard), *669*
- \empty, xxxi
- emTeX program, 24
- emtex option
    - (graphics/graphicx), 24
    - (xcolor), 721
- Enable: syntax (M-Tx), 652
- encapsulation, 35, 36
- Encore program, 588
- \endextract (MusiXTeX), **594**, *596*
- endocyclic bonds, *523*
- \endpiece (MusiXTeX), **594**, *599*
- engineering drawings, *see* bioinformatics, *see* chemical formulas, *see* Feynman diagrams, *see* scientific texts
- \enotes (MusiXTeX), **591**, *594–596*, *599*
- enpassant package, 670
- \ensuremath, *699*
- enumerate env. (beamer), 770, 786

- `envcountsec` option (beamer), 753
- `environment` key (beamer), 759
- Environment Variables
    - `TEX` (METAPOST), **63**, 64
- epic package, 15, 511, 520–522, 537
- ePiX language, 20
- ePiX program, 20
- `.eps` file extension, 35
    - (graphics/graphicx), 35
- EPS (Encapsulated PostScript), 35, 36
- epsfig package, 42
- epstopdf program, 804, 806
- `eqnarray` env. (pst-pdf), 800
- equal sign (=), natural symbol (musical), *605*
- `equation` env. (pst-pdf), 800
- `EQUILIBRIUM` syntax (m-ch-en), ***542***, 546
- `ER` syntax (m-ch-en), 544
- etex program, 14
- `\ethanestereo` (aliphat), *533*
- `\ethylene` (aliphat), *533*
- ethylene derivatives, *533*
- `\Ethylenev` (aliphat), 533
- `\ethylenev` (aliphat), *533*
- `\ETri` (axodraw), 558
- evince program, 12
- `\exa` (SIunits), 515
- `example` env. (beamer), *769*
- `exampleblock` env. (beamer), **778**, *779*
- examples, this book, xxxi, xxxiii
- Excel program, 21
- exclamation points (!!), color expression, 732
- `\ExecuteOptions`, 25
- expression marks (musical), 657, 658
- `ext` key (graphicx), 29
- external vertices (Feynman diagrams), *564*
- `\extrarowheight` rigid length (array), *738–741*
- extsizes package, 753

F

- `\f` (MusiXTeX), *599*
- `f` syntax (PMX), *624*, **625**, 631, *636*
- `fact` env. (beamer), 769
- `family` key (beamer), 793
- `family*` key (beamer), 793
- FAQs (Frequently Asked Questions), 809, *see also* online resources
- `\farad` (SIunits), 514
- `\fboxrule` rigid length (xcolor), *723, 724*
- `\fboxsep` rigid length (xcolor), *724*, 748
- `fc` syntax (PMX), 625
- `\fcolorbox` (xcolor), 720, ***723***, *724*
- `\fdmass` (circ), *580*
- `\feature` (texshade), *549*
- `\featureslarge` (texshade), *549*
- `\femto` (SIunits), 515
- `\femtobarn` (hepunits), *516*
- FEN (Forsyth-Edwards-Notation), 674
- `\fenboard`
    - (skak), **674**, *675–677*
    - (texmate), 684
- `\fermatadown` (MusiXTeX), 592
- `\Fermataup` (MusiXTeX), 592
- `\fermataup` (MusiXTeX), 592
- `\Feyn` (feyn), *557*
- `\feyn` (feyn), *555–557*
- feyn package, 555–558
- FeynArts package, 555
- feynman package, 555
- Feynman diagrams, *see also* scientific texts
    - arc segments with arrows, *560*
    - arrows, *559–561*
    - direct use of META commands, 572
    - font for, *555–557*
    - history of, 555
    - immediate mode
        - arcs, *572*
        - definition, 563
        - diagrams in equations, *570*
        - edges, *572*
        - freezing diagrams, 570
        - labels, *571*
        - loop diagrams, *569*
        - overview, 569–572
    - overview, 561–563
    - photons, *561*
    - PostScript, 558–561
    - transformers, 572
    - vertex dots, *560*
    - vertex mode
        - algorithmic layout, 563–569
        - blobs, 566
        - coloring diagrams, 567
        - complex vertices, 567
        - definition, 563
        - external vertices, placing, *564*
        - fill styles, 564
        - freezing a diagram, 567
        - internal vertices, 566
        - labels, 567, *568*, 569
        - line styles, 565
        - line thickness, 566
        - line-drawing keywords, 566
        - polygon keywords, 567, 568
        - vertex styles, 564
        - vertex-drawing keywords, 567
        - vertices, as dots, 566
        - vertices, connecting, 565
    - zigzag lines, *559, 560*
- feynmf package, 561–572
- feynmp package, **562**, 572

- ff syntax (PMX), 625
- ffc syntax (PMX), 625
- \fff (circ), *579*
- fg key (beamer), *776*, *794, 795*
- \figfont (chessfss), *670*, **671**
- \figsymbol (chessfss), *671*
- figure env. (beamer), 780
- figures, slides, 780
- Figurine chess symbols, *671*
- file extensions
  - search order, 33, 34
  - specifying, 29, 34, 35
- file name parsing, suppressing, 29
- file type, specifying, 34
- filecontents* env., *710*
- fill styles (Feynman diagrams), 564
- fill-in puzzles, *707*, *see also* crosswords
- final option
  - (graphics/graphicx), 25
  - (pst-pdf), 800
- Finale program, 588
- \fingerprint (texshade), *550*
- finite state diagrams, *see META and* PSTricks *index*
- firstsection key (beamer), 783
- FIVE syntax (m-ch-en), *542*
- \fivefuseh (fusering), 537
- \fivefusehi (fusering), 537
- \fivefusev (fusering), 537
- \fivefusevi (fusering), 537
- \fiveheteroh (hetarom), *529*
- \fiveheterohi (hetarom), *529*
- \fiveheterov (hetarom), *528*, ***529***, *539*
- \fiveheterovi (hetarom), *529*
- \fiveunitv (hetarom), *534*
- \fiveunitvi (hetarom), *534*
- \fla (MusiXTEX), 593
- \flageolett (MusiXTEX), 592
- flat symbol (musical), *605*
- Flats: syntax (M-Tx), **652**, *656*
- flow program, 16
- flow charts, *16*, *see also META index*
- flow language, *16*
- \fmf (feynmf), *561*, ***565***, *567–572*
- \fmfblob (feynmf), *566*
- \fmfblobn (feynmf), 566
- \fmfbottom (feynmf), 565
- \fmfbottomn (feynmf), 565
- \fmfcmd (feynmf), *572*
- \fmfcurved (feynmf), 565
- \fmfcyclen (feynmf), **565**, *572*
- \fmfdot (feynmf), *561*, **566**, *568, 569*
- \fmfdotn (feynmf), ***566***, *570*
- fmffile env. (feynmf), *562*
- \fmffixed (feynmf), *569*, **570**
- \fmffreeze (feynmf), **567**, *569*, **570**
- fmfgraph env. (feynmf), **568**, *569*
- fmfgraph* env. (feynmf), *561*, ***568***, *570–572*
- \fmfi (feynmf), *569*, **570**
- \fmfiequ (feynmf), *569*
- \fmfipair (feynmf), 570
- \fmfipath (feynmf), *569*, **570**
- \fmfiv (feynmf), *569*, **570**
- \fmflabel (feynmf), ***568***, *570*
- \fmfleft (feynmf), *561*, **565**, *569–572*
- \fmfleftn (feynmf), **565**, *568*, 569
- \fmfn (feynmf), 565
- \fmfpen (feynmf), 566
- \fmfpoly (feynmf), 567
- \fmfrcyclen (feynmf), 565
- \fmfright (feynmf), *561*, **565**, *569–572*
- \fmfrightn (feynmf), **565**, *568*, 569
- \fmfstraight (feynmf), 565
- \fmfsurround (feynmf), 565
- \fmftop (feynmf), 565
- \fmftopn (feynmf), 565
- \fmfv (feynmf), *566*
- \fmfvn (feynmf), 566
- \fmpolyn (feynmf), 567
- .fmt file extension (abc), 612
- foiltex package, 719
- fontenc package, 752
- fonts
  - cchess46 (cchess), 688
  - chess
    - Figurine symbols, *671*
    - generic mechanism, 669–673
    - list of, *670*
    - normal characters, 671
    - selecting, *672, 673*
    - switching, *672*
  - electronics diagrams, 576–582
  - feyn (feyn), 555–557
  - Feynman diagrams, *555–557*
  - gosign50 (go), *691*
  - optics diagrams, 576–582
  - skaknew (skak), 673
  - slides, *758*
  - Symbol (pstricks), 250
  - timing diagrams, 573
  - ZapfDingbats (pstricks), 249, 250
- footline syntax (beamer), *773*, *777*
- \footnote (beamer), *789*
- footnotes, slides, *789*
- Forsyth-Edwards-Notation (FEN), 674
- FOUR syntax (m-ch-en), *542*
- four-color harmonics, 718
- four-member carbon cycles, *528*
- \fourhetero (hetarom), *528*, ***529***
- fractals, *see META and* PSTricks *index*
- fragile key (beamer), **759**, *790, 791*

\Frame (cwpuzzle), **704**, *705*
\frame (beamer), *754*, **758**, *761*
  frame env. (beamer), *754*, **758, 759**, *761*, **776, 784, 790**
  frame key (beamer), *789*, 790
\frameblock (texshade), *549*
  frames, *see also* boxes
    slides, creating, 758
    text in slides, *775, 776*
\framesubtitle (beamer), 759
  framesubtitle syntax (beamer), *794*
\frametitle (beamer), *754, 755*, **759**
  frametitle syntax (beamer), *794*
  free-hand pictures, 3, *4*
  freezing a Feynman diagram, 567, 570
  Frequently Asked Questions (FAQs), *see* online resources
\from (circ), 580
  from (pic), *19*
\frompin (circ), **580**, *581*
\fullboard (bg), 697
\fullincr (bg), 698
\furanose (hcycle), ***532***, *539*
  furanoses, *532*
  fused five- and six-member rings, *530*
  fused rings, *524*
  fusering package, 537
  fusing ring units, 536

G

\G (circ), *578*
\G2Text (axodraw), 558
  game env. (bg), **696, 697**, *698*
  games, *see* backgammon, *see* bridge, *see* chess, *see* crosswords, *see* Go, *see* Sudoku
\gapchar (texshade), *550*
  gastex package, 15
  gate option (circ), 577
  gate symbols, *578*
\gauss (hepunits), *516*
\GBox (axodraw), 558
\GBoxc (axodraw), 558
\GCirc (axodraw), 558
\generalmeter (MusiXTeX), *596*, *599*
\generalsignature (MusiXTeX), **593**, *596*
\generategrid (createsudoku), *711*
\genfile (createsudoku), 711
  gensud.sud file (createsudoku), *711*
\geometricskipscale (MusiXTeX), 595
  geometry, *see META and* PSTricks *index*
\getproblem (solvesudoku), *711*
\getsequence (textopo), *551, 552*
\GeV (hepunits), *516*
.gf file extension (feynmf), 563
  gftopk program, 563
  ghostscript program, xxv, xxvi, xxviii, 11, 12, 798
  ghostview program, xxvi, xxviii, 10, 36, 804
\GHz (hepunits), *516*
\giga (SIunits), 515
  GIMP program, 4, 17
  gis syntax (LilyPond), 662
  GIVES syntax (m-ch-en), 546
  global A options (musical), 643
\GlueArc (axodraw), 558
\Gluon (axodraw), 558
\GND (circ), *581*
  gnuplot program, **17**, 18
  Go
    goban (board)
      captions, 694
      clearing, 694
      copying, 694, *695*
      displaying, 693, *694*
      rotating, *695*
      size, 694
    history of, 690, 691
    stones
      identifying, *692*
      placing, 691, *692*, 693
      typesetting, *695*
  go package, 690, 691
\gobansize (igo), 693
\gobansymbol (igo), *692*, ***693***, *694*
\gosign (tlgc), *691*
\GOval (axodraw), 558
  gpic program, 16, **17**, 19, 583, 584
  grace notes (musical)
    {} (curly braces), *607*
    ~ (tilde), *607*
    in xtuplets, 627
    LilyPond, 663
    PMX, 627, 629, *630*
  gracings (musical), *607*
  grad syntax (xcolor), 734–736
  gradients (table), color, *747, 748*
\gram (SIunits), *516*
  graphic objects
    conflicting requirements, 3
    definition, 2
    drawing, *see* graphics languages, *see* manipulating graphic objects
    manipulating, *see* manipulating graphic objects
    typesetting, 2, 3
  graphics
    elements, SVG, 12
    files, including, *see* including graphics files
    rotating
      bounding box, *27*, *31, 32*
      graphic objects, *39–42*
      \includegraphics keys, 29
      reference points, 40–42

- graphics *(cont.)*
  - scaling
    - bounding box, *27*, 29
    - graphic objects, *37*
    - \includegraphics keys, 29, *30*
    - text, *37*
  - slides, *792*
  - systems, typesetting, 2, 3
- graphics package, 2, 3, 7, 8, 10, **23–27**, 30, **33–40**, 791
- graphics languages, *see also* drawing tools
  - AlDraTex package, *15*
  - DraTex package, *15*
  - CGM (Computer Graphics Metafile), 13
  - character-based diagrams and pictures, 13
  - choosing, 21, 22
  - diagrams, typesetting, 16
  - ePix, *20*
  - flow language, *16*
  - for basic objects, 17, 18, *19*, 20
  - for plotting, 17, *18*
  - gnuplot, 17
  - pic, 17, 19
  - graphs
    - drawing, 17, *18*
    - typesetting, 16
  - kernel drawing language, 16
  - LaTeX picture mode extensions, 15, 16
  - METAPOST, *see META index*
  - PDF (Portable Document Format), 11, 12
  - pic, 17–20
  - PiCTeX, 13, *14*
  - pictures, 17–20
  - pictures from fonts, 13
  - PostScript, 10, 11
  - PSTricks, *see* PSTricks *index*
  - structured drawing, 20
  - SVG (Scalable Vector Graphics), 12, 13
  - TeX-based, 13–17
  - WebCGM, 13
  - Xy-pic, 16
- graphics.cfg file (graphics/graphicx), 25
- \graphicspath (graphics/graphicx), *33*
- graphicx package, **23–25**, **28–42**, 800
- graphs, *see also* META, PSTricks, and Xy-pic index, *see also* diagrams, *see also* plotting
  - drawing, 17, *18*
  - graphics languages
    - drawing, 17, *18*
    - typesetting, 16
  - histogram, *14*
  - pie chart, *15*
  - typesetting, 16
- GRASS program, 21
- Grassman's Law, 714
- Gray option (xcolor), 721
- Gray syntax (xcolor), 728, 729
- \gray (Slunits), 514
- gray option (xcolor), 721
- gray syntax
  - (color), 720
  - (xcolor), 720, *723*, 728–730
- gray color model, 719
- \grcl (MusiXTeX), 592
- \grcu (MusiXTeX), 592
- green syntax (xcolor), ***722***, **726**, 727
- \gregorianCclef (MusiXTeX), 592
- \gregorianFclef (MusiXTeX), 592
- grid key (beamer), *794*
- grids, *see META and* PSTricks *index*
- \GText (axodraw), 558
- \GTri (axodraw), 558
- guitar chords, *608*, *611, 612*
- guitar diagrams, drawing, *612*
- gunzip program, 35
- \Gvar (circ), *578*

H

- H syntax (PMX), *636*
- \H (chemsym), *517*
- \h (chemsym), 517
- h syntax (PMX), **631**, *632*, *636*
- \ha (MusiXTeX), 593
- \halfboard (bg), 697
- \halfincr (bg), 698
- \hand
  - (bridge), *700–702*
  - (tlgc), *699*
- handidness of substituents, *522*, *531*, 535
- handout option (beamer), 753
- \hanthracenev (lowcycle), *527*
- \hanthracenv (carom), 524
- harmonic color circle, 717
- harmonies, color, 717, 718
- \HBLens (circ), *580*
- \hbox, 725
- hcycle package, 520, **532**
- headerCol syntax (beamer), *776*
- headings (table), color, *748*
- \heart
  - (bridge), *702*
  - (tlgc), *699*
- \heartsuit, 698, *699*
- \hecto (Slunits), 515
- \height (graphics/graphicx), *38*
- height (pic), *19*
- height key (graphicx), **29**, *31, 32*
- helicalwheel env. (textopo), 551, **552**
- helixwheel env. (textopo), *552*
- help, *see* online resources
- \henry (Slunits), 514

hepnicenames package, 512, 560
heppennames package, 512, 560
\heptamethylene (methylen), *538*
\heptamethylenei (methylen), *538*
hepunits package, 516, 517
\hertz (Slunits), 514
hetarom package, 520, **528**, 530, **534**
hetaromh package, 520, **528**, **534**
heterocyclic compounds, 528–530
\hexamethylene (methylen), *538*
\hexamethylenei (methylen), *538*
\hflipgoban (igo), *695*
HH syntax (PMX), *636*
\hhline
(colortbl), 751
(hhline), *750*
hhline package, 737, 742, **750**
hide key value (beamer), 753
hideallsubsections key (beamer), 783
\hideconsensus (texshade), *548*
hideerrors option (xcolor), 721
\hidelegend (textopo), *553*
\hidemoves (skak), **677**, *678*, 679
\hidenumbering (texshade), *549*
hideothersubsections key (beamer), 783
hiderotate option (graphics/graphicx), 25
\hiderowcolors (xcolor), 740
hidescale option (graphics/graphicx), 25
\hideTMlabels (textopo), *551*
hiding/showing
chess pieces, 676
slides
alternative text, *769*
opaqueness, *768*
slide elements, *767*
specific rows, *765*
successive columns, *763*
successive rows, *763*
transparency, *768*
high-energy physics, units, *516*
\highlight (skak), *676*
highlighting
chess, *676*
nucleotide sequences, *548–550*
peptide sequences, *548–550*
slides, parts of elements, *771*
table elements, with color, *745*, *749, 750*
text in tables, *744*
highlydynamic key (beamer), 767
hiresbb key (graphicx), 28
hiresbb option (graphics/graphicx), 25
HiResBoundingBox (PostScript), 25, 28
\hl (MusiXTeX), 592, **593**
\hline (colortbl), 741
How To Ask Questions The Smart Way, 810
\Hpause (MusiXTeX), 592
\hpause (MusiXTeX), 592, **594**, *599*
\hpausep (MusiXTeX), 592
\hphenanthrenev
(carom), 524
(lowcycle), *527*
\HR (tlgc), *26*
\hs (MusiXTeX), 592
HSB option (xcolor), 721
HSB syntax (xcolor), 728, 729
Hsb syntax (xcolor), 728, 729
hsb option (xcolor), 721
hsb syntax
(color), 720
(xcolor), 720, 728, 729
HSB (Hue, Saturation, Brightness) color, 715, **719**
\HSLens (circ), *580*
HSV (Hue, Saturation, Value) color, 715
ht key (beamer), 777, *794*
HTML option (xcolor), 721
HTML syntax (xcolor), 728, *729*
\htopin (circ), ***579***, *581*
\hu (MusiXTeX), 592, **593**, *594*
\HVLens (circ), *580*
\hyperlink (beamer), *784, 785*
\hyperlinkappendixend (beamer), 786
\hyperlinkappendixstart (beamer), 786
\hyperlinkdocumentend (beamer), 786
\hyperlinkdocumentstart (beamer), 786
\hyperlinkframeend (beamer), 786
\hyperlinkframeendprev (beamer), 786
\hyperlinkframestart (beamer), 786
\hyperlinkframestartnext (beamer), 786
\hyperlinkmovie (beamer), 774
\hyperlinkmute (beamer), 774
\hyperlinkpresentationend (beamer), 786
\hyperlinkpresentationstart (beamer), 786
hyperlinks, slides, *784–818*
\hyperlinkslidenext (beamer), 786
\hyperlinkslideprev (beamer), 786
\hyperlinksound (beamer), 774
hyperref option
(beamer), 753
(xcolor), 721
hyperref package, 721, 753, **783**, 798, 803–805
\hypertarget (beamer), **783**, *784, 785*
hyphen (-), tie symbol, *607, 608*

I

I syntax (PMX), 648
\I (circ), *577*
i syntax (pic), *19*
I: syntax (abc), *608*
\ib (MusiXTeX), *599*
\ibbu (MusiXTeX), 597

- \ibl (MusiXTEX), *596*, **597**
- \ibu (MusiXTEX), *596*, **597**
- ic option (circ), 577
- \ifont (texmate), 687
- ignorebg key (beamer), 777
- ignoreonframetext option (beamer), 753
- igo package, 691–695
- \igobreakafterdiagram (igo), 694
- \igocircle (igo), 692
- \igocross (igo), **692**, *695*
- \igofontsize (igo), *693*, **694**
- \igonone (igo), **691**, *692*
- \igosquare (igo), ***692***, *695*
- \igotriangle (igo), ***692***, *695*
- \iiclose (texmate), *687*
- \iiiclose (texmate), *687*
- \iiifont (texmate), *687*
- \iiiopen (texmate), *687*
- \iiopen (texmate), *687*
- illustrations, *see* drawing
- Illustrator program, 586
- image file location, specifying, 33
- ImageMagick program, 7, 17
- images, *see* drawing
- \imidazolev (hetarom), 530
- \imidazolevi (hetarom), 530
- immediate mode (Feynman diagrams)
  - arcs, *572*
  - definition, 563
  - diagrams in equations, *570*
  - edges, *572*
  - freezing diagrams, 570
  - labels, *571*
  - loop diagrams, *569*
  - overview, 569–572
- \Impulse (circ), *578*
- inactive option (pst-pdf), 800
- \includegraphics
  - (beamer), **791**, ***792***, *794*
  - (graphics), ***26***, *27*, 33–35
  - (graphicx), 24, 25, **28**, *30–32*, 33–35
- \includegraphics*
  - (graphics), **25**, *27*
  - (graphicx), 28
- including graphics files
  - aspect ratio, keeping, 29, *31*
  - bounding box
    - aspect ratio, keeping, 29
    - clipping graphics to, 29, *30*
    - comments, 25, 28
    - draft mode, 25, 29, *30*
    - final mode, 25
    - fitting to graphics, *26, 27*
    - height, 28, 29, *32*
    - \includegraphics syntax, 28–32

including graphics files *(cont.)*

    - resizing, *27*
    - rotated material, hiding, 25
    - rotating, *27*, *31, 32*
    - scaled material, hiding, 25
    - scaling, *27*, 29
    - specifying, 28, *30*
    - trimming space, 28, *30*
    - viewports, 28, *30*
    - width, 28, 29
  - commands, inserting, 35
  - declarations, 33–35
  - default key values, setting, 32, 33
  - draft mode, 25, *30*
  - encapsulation, 35, 36
  - file extensions
    - search order, 33, 34
    - specifying, 29, 34, 35
  - file name parsing, suppressing, 29
  - file type, specifying, 34
  - final mode, 25
  - height, 28, 29, *31, 32*
  - image size, 29
  - \includegraphics syntax, 25–32
  - location of image files, 33
  - options, 24, 25
  - rotated material, hiding, 25
  - rotation, 29, *31, 32*
  - scaled material, hiding, 25
  - scaling, 29, *30*
  - scaling factor, 29, *30*
  - trimming space, 28, *30*
  - viewports, 28, *30*
  - width, 28, 29, *31*
- indane derivatives, *528*
- \indaneh (lowcycle), *527*
- \indanehi (lowcycle), *527, 528*
- \indanev (lowcycle), 526, ***527, 528***
- \indanevi (lowcycle), *527*
- Indent: syntax (M-Tx), *651*, **652**
- \indentwhite (bg), 698
- \indolev (hetarom), 530
- \indolevi (hetarom), 530
- \indolizinev (hetarom), 530
- \indolizinevi (hetarom), 530
- \inffont (chessfss), 673
- \infsymbol (chessfss), *673*
- inputenc package, 752, 753
- \insertbackfindforwardnavigationsymbol (beamer), 773
- \insertdocnavigationsymbol (beamer), 773
- \insertframenavigationsymbol (beamer), *773*
- \insertframenumber (beamer), *777*
- \insertframesubtitle (beamer), *794*
- \insertlogo (beamer), **776**, 777

\insertsectionnavigationsymbol (beamer), 773
\insertshortdate (beamer), *777*
\insertshortframetitle (beamer), 759
\insertslidenavigationsymbol (beamer), *773*
\insertsubsectionnavigationsymbol (beamer), 773
\inserttotalframenumber (beamer), *777*
\inst (beamer), *761*
\institute (beamer), *761*
\instrumentnumber (MusiXTeX), *596*
instruments (musical)
  clefs, *621*
  definition, 617
  names, 621
  number of, 596, 619
integrated circuit symbols, *579*
intensity, color, 718
internal vertices (Feynman diagrams), 566
International System of Units (SI), 512–516
internote spacing (musical), 602
\invfemtobarn (hepunits), *516*
\invisible (beamer), **768**, *784*
invisible key (beamer), 767
invisibleenv env. (beamer), 770
\invpicobarn (hepunits), *516*
\islurd (MusiXTeX), 597
\isluru (MusiXTeX), *596*, **597**, *599*
\isobenzofuranev (hetarom), 530
\isobenzofuranevi (hetarom), 530
\isoindolev (hetarom), 520, 530
\isoindolevi (hetarom), 530
\isoquinolinev (hetarom), 530
\isoquinolinevi (hetarom), 530
\isotope (isotope), 518
isotope package, 518
\isotopestyle (isotope), 518
\isoxazolev (hetarom), 530
\isoxazolevi (hetarom), 530
\item (beamer), *770*, ***786***, *787, 788*
itemize env. (beamer), *771*, ***772***, **786**, *787*
\itenu (MusiXTeX), *599*
\IvaR (circ), *577*
\ivfont (texmate), 687

J

j syntax (PMX), 631
\JKMSFF (circ), *579*
\joule (SIunits), 514, *516*
\jouleperkilogramkelvinnp (SIunits), *516*
.jpeg file extension (pst-pdf), 806
\junction (circ), 579
junctions, 579

K

K syntax (PMX), **640**, *641*
K type slurs (musical), *636*
K: syntax (abc), ***601***, **603**, *604–606*
\kat (SIunits), 514
keepaspectratio key (graphicx), **29**, *31, 32*
\keepreducing (solvesudoku), *711*
\kelvin (SIunits), 514, *516*
\kemtkn (chemsym), 517
kernel drawing language, 16
\key (LilyPond), ***662***, *663–665*
key (musical)
  changes, *641*
  LilyPond, *662*
  notation, 601
  signature, 620
keyval package, 33
\kilo (SIunits), 515
\kilogram (SIunits), 514
\kilogrampersecondcubicmetrenp (SIunits), *516*
\king (chessfss), 672
\kinveV (hepunits), *516*
\knight (chessfss), *672*
\kqu (MusiXTeX), 592

L

L syntax (PMX), 642
\L (circ), *577*
\l (MusiXTeX), 592
l syntax (PMX), **625**, 631, *633*, *637*, *641*
\l... (MusiXTeX), 594
L: syntax
  (M-Tx), *655*, *659, 660*
  (abc), *601*, 603, *604*
\La (circ), *577*
lab apparatus, *see* PSTricks *index*
\label (beamer), **783**, *785*
label key (beamer), **759**, *761*
\labelregion (textopo), *553*
labels
  Feynman diagrams, 567, *568*, 569, *571*
  slides, *785*
  timing diagrams, 573
\labelstyle (textopo), *553*
large option (skak), 675
\largeboard
  (cchess), 690
  (skak), 675
\largegoban (igo), 694
\larw (timing), 575
\Laser (circ), ***580***, *581*
last syntax (xcolor), 734
\lastmove (skak), 679
latex program, 797, 800, 801, 803, 804, 806
LaTeX files, obtaining
  web access, 810, 811, *812, 813*, 814
\LED (circ), *577*

left (pic), *19*
left key (beamer), 777
\leftdiagramturn (texmate), 686
\leftrepeat (MusiXTeX), 592
\leftrightrepeat (MusiXTeX), 592
leftskip key (beamer), 777, *794*
libcct.m4 file (pic), 583
\lifthpause (MusiXTeX), 592
\liftpause (MusiXTeX), 592
light, and color, 714
lightgray syntax (xcolor), 726
LilyPond language, 661–665
LilyPond program, xxviii, **661–665**
LilyPond notation system, *see* music scores (LilyPond)
\LinAxis (axodraw), 559
\LINE (curve2e), 47, *48–50*
\Line
    (axodraw), 559
    (curve2e), 47, *48–50*
\line, *43*
    (curve2e), 47, *48–50*
    (pict2e), *43*, 44
line (pic), 17
line graphics
    arrow styles, *44*
    Bézier curves
        cubic, *47*
        quadratic, 46, *47*
    circles, *45*
    curves, 47, *48–50*
    limitations, 42, *43*
    ovals, 45, *46*
    overview, 42, 43
    radii, specifying, 45, *46*
    representing complex numbers, *49*, 50
    slope arguments, 44
line styles (Feynman diagrams), 565
line-drawing keywords (Feynman diagrams), 566
lines (musical)
    breaks, 642
    definition, 617
lines (rules), *see also* connections
    styles
        Feynman diagrams, **564, 565**, 566
        thickness, 566
    tables, color
        adding, *748*
        inside the table, *749*
        partial, 751
        selected, *750*
        whole table, *741*
        width, *751*
\linethickness, 47
    (pict2e), *44, 45*, 46
    (timing), 576
\linewidth rigid length, 33
linewidth key (chessboard), *669*
linguistics, *see* PSTricks *and XY-pic index*
list env., *724*
list items, slides, *786–788*
listings package, 790
lists, colored, *724*
\lmoiety (chemstr), ***522***, 526
\ln (circ), 579
\loadgame (skak), 679
locant package, 520
.log file extension (feynmf), 562, 567
\LogAxis (axodraw), 559
logical circuit diagrams, *see XY-pic index*
logical meter (musical), 620
\logo (beamer), *776, 777*, *792*, *794*
logos, slides, *776, 777*
\longa (LilyPond), *663*
\LongArrow (axodraw), 559
\LongArrowArc (axodraw), 559
\LongArrowArcn (axodraw), 559
longtable package, **517**, 737, 742
loop diagrams (Feynman diagrams), *569*
\loopextent (textopo), *552, 553*
\loopfoot (textopo), *553*
lowcycle package, 520, **526**
lower key (beamer), 778
lower-order cycles, *527, 528*
\lppz (MusiXTeX), 592
\lpz (MusiXTeX), 592
\lpzst (MusiXTeX), 592
\lsf (MusiXTeX), 592
\lsfz (MusiXTeX), 592
\lsqu (MusiXTeX), 592
\lst (MusiXTeX), 592
\ltetrahedralS (aliphat), *540*
\ltrigona (aliphat), *533*
.ltx file extension, xxxi
ltxarrows option (pict2e), *44*
.ltxb file extension, xxxi
\lumiunits (hepunits), *516*
.ly file extension, xxxi
    (LilyPond), 665
\lyl (chemstr), *535, 536*
lyrics (musical)
    global adjustment, 653
    M-Tx, *659, 660*
    PMX, 647

M

m syntax (PMX), **629**, *630*, **631**, **640**
M type slurs (musical), *637, 638*
m-ch-en package, 541–547
M-Tx notation system, *see* music scores (M-Tx)
M-Tx language, xxviii, 616, 617, **651–660**

- M-Tx program, 647
- .m4 file extension, xxxi
- m4 program, **576**, **583**, 584
- M: syntax (abc), ***601***, ***604***, *605, 606*
- magenta syntax (xcolor), 722, 726
- magnifying glass effect, *see* PSTricks *index*
- \mainline (skak), **677**, *678, 679*
- \major (LilyPond), *663–665*
- \makeatletter, **xxxii**, xxxiii
- \makeatother, **xxxii**, xxxiii
- \makebarchess (texmate), 680
- \makebarother (texmate), 680
- \makebox
  - zero-width, *37*
  - (cwpuzzle), *705*
- makecirc package, 576
- \makediagrams (texmate), **685**, *686*
- \makediagramsfont (texmate), 686
- \makegametitle (texmate), *683*
- makeindex program, 806
- \maketitle (beamer), *754*, *757*, **761**
- manipulating graphic objects
  - aspect ratio, keeping, *38*
  - height, changing, *38, 39*, *see also* bounding box
  - line graphics
    - arrow styles, *44*
    - circles, *45*, *see also* circles, *see also* ovals
    - cubic Bézier curves, *47*
    - curves, 47, *48–50*
    - limitations, 42, *43*
    - ovals, 45, *46*
    - overview, 42, 43
    - quadratic Bézier curves, 46, *47*
    - radii, specifying, 45, *46*
    - representing complex numbers, *49*, 50
    - slope arguments, 44
  - resizing, *38, 39*
  - rotating
    - LaTeX box, *39–42*
    - reference point, 40–42
  - scaling, *37*
  - width, changing, *38, 39*
- Maple program, 2
- markfields key (chessboard), *669*
- markfile key (chessboard), *669*
- markstyle key (chessboard), *669*
- masking color, 737
- Mathematica program, 1, 21
- mathematical functions, symbols for, 512
- mathematical plots, *see* PSTricks *index*
- \mathrm, 512
- MATLAB program, 2
- matrices, *see* PSTricks and Xy-pic index
- \maxovalrad (pict2e), **45**, 46
- mechanical drawings, *see* META *index*
- mediumqspace option (Slunits), 515
- mediumspace option (Slunits), 515
- \mega (Slunits), 515
- membrane protein topology plots, *551–553*
- META language, 21, *see also* META *index*
- METAFONT, *see* META *index*
- METAOBJ package, *see* META *index*
- METAPOST, *see* META *index*
- meter (musical)
  - abc notation system, 601
  - changes, 640, 654
  - logical, 620
  - M-Tx, 654
  - PMX, 640
  - representation, *620*
- Meter: syntax (M-Tx), *651*, **652**
- \meterC (MusiXTeX), 592
- \meterfrac (MusiXTeX), *596*, *599*
- \meterplus (MusiXTeX), 592
- methylen package, 537
- \metre (Slunits), 514, *516*
- \metron (MusiXTeX), 592
- \Mev (hepunits), *516*
- \MeVoverc (hepunits), *516*
- \meVoverc (hepunits), *516*
- \MeVovercsq (hepunits), *516*
- mfpic package, 21, 583
- \MHz (hepunits), *516*
- \micro (Slunits), 515, *516*
- .mid file extension (PMX), 648
- \middlecube (bg), *696*, **697**
- MIDI language, 610, 647–649, 660
- MIDI mnemonics, 649
- \milli (Slunits), 515
- minus sign (-), color expression, 732
- \Mirror (circ), ***580***, *581*
- \mirrorgoban (igo), 695
- mixing color, *731*
- \mode (beamer), *760*, ***796***
- \mode* (beamer), 753, 796
- \mole (Slunits), 514, *516*
- molecules, aligning with bonds, *546*
- \momentum (feyn), **556**, *557*
- monochrome, 721
- monochrome option (xcolor), 721
- \Mordent (MusiXTeX), 592
- \mordent (MusiXTeX), 592
- MOV syntax (m-ch-en), 544
- \move (bg), **697**, *698*
- move (pic), *19*
- mover option (skak), 676
- \moverel (circ), 580
- moveroff option (skak), 676
- \movie (beamer), 774
- movies, slides, 774

- Mozart example, *651*
- .mp file extension, xxxi
- mpost program, 637
- \mrad (hepunits), *516*
- \MRs (textopo), 551, *553*
- .mtx file extension, xxxi
- \multicolumn, 701
  - (colortbl), 737, 739
- \multido (multido), *45*
- multimedia package, 774
- \MultVect (curve2e), *49, 50*
- music env. (MusiXTeX), ***594***, *595, 596*, *599*
- music scores, overview, 587–589
- music scores (abc2mtex)
  - abc notation system, 600
    - ' (right quote), octave indicator, *603*
    - (...), slur symbol, *607, 608*
    - , (comma), octave indicator, *603*
    - - (hyphen), tie symbol, *607, 608*
    - = (equal sign), natural symbol, *605*
    - [] (square brackets), chord symbols, *608*
    - "..." (double quotes), guitar chords, *608*
    - {} (curly braces), grace notes, *607*
    - ~ (tilde), grace notes, *607*
    - ^(caret), sharp symbol, *605*
    - ^^(carets), double flat symbol, *605*
    - _ (underscore), flat symbol, *605*
    - __ (underscores), double flat symbol, *605*
    - accents, *607*
    - accidentals, *605*
    - bar symbols, *603*
    - bars, *603*
    - beams, *606*
    - broken rhythms, *604*
    - changing key, *606*
    - chords, *608*
    - compound time signatures, *605*
    - dotted rhythms, *604*
    - double bars, *603*
    - doublets, *605*
    - Dusty Miller example, *608*
    - fiddler instructions, *607*
    - gracings, *607*
    - guitar chords, *608*
    - information fields, description of, 601, 602
    - information fields, table of, 602
    - internote spacing, 602
    - key, 601
    - lowercase letters, *603*
    - meter, 601
    - musical information, 601
    - note length, 601, *603, 604*
    - note pitch, *603*
    - order of symbols, *608*
    - pitch, 603, *604*

- music scores (abc2mtex) *(cont.)*
    - quadruplets, *605*
    - repeat symbols, *603*
    - sequence number, 602
    - slurs, *607*
    - song title, 602
    - staccato marks, *607*
    - tempo, 602
    - ties, *607*
    - triplets, *605*
    - uppercase letters, *603*
    - writing source, *601*
  - abcPlus extensions, 609–612
  - Bach example, *610*
  - external programs, calling, 615
  - guitar chords, *611, 612*
  - guitar diagrams, drawing, *612*
  - including in LaTeX documents, *612–614*, 615
  - overview, 600
  - PostScript definitions, *612*
  - writing to PDF, 614
- music scores (LilyPond)
  - accents, *663*
  - chords, *663*
  - notes
    - accents, *663*
    - beams, 663
    - chords, *663*
    - duration, 662, *663*
    - grace notes, 663
    - key, *662*
    - notation, *661*
    - ornaments, *664*
    - pitch, *662*
    - slurs, *663, 664*
    - triplets, *664*
  - ornaments, *664*
  - rests, *663*
  - running LilyPond, *665*
  - slurs, *663, 664*
  - source language, 661–665
  - triplets, *664*
- music scores (M-Tx)
  - annotations, 657, 658
  - bar changes, 654
  - beams, 654, 655
  - body of file, 654–658
  - chords, *656, 657*
  - clefs, 653
  - expression marks, 657, 658
  - horizontal adjustment, 658
  - instruments, definition, 617
  - lines, definition, 617
  - lyrics, *659, 660*
    - global adjustment, 653

- music scores (M-Tx) *(cont.)*
  - meter changes, 654
  - Mozart example, *651*
  - overview, 651, 652
  - pickups, 654
  - preamble of file, 652, 653
  - slurs
    - blind, *655*
    - broken, *655*
    - description, 654, 655
    - dotted, *655*
    - notation, *654*
  - staves, 617, 652
  - symbols, definition, 617
  - systems, definition, 617
  - vertical adjustment, 658
  - voice
    - definition, 617
    - labels, 653
    - spacing after, 653
  - words, definition, 617
- music scores (MusiXTeX)
  - {} (curly braces), around arguments, 596
  - Bach example, *590*
  - Bartok example, *596*
  - beams, 597
  - chords, *594*
  - commands, *592*
  - instruments, number of, 596
  - notes
    - commands, 595
    - pitch, 590, 593
    - spacing, *595*
    - symbols, *592*, 593, *594*
    - timing, 590
  - preprocessors, 615, *616, 617*
  - running MusiXTeX, 597, *598, 599*
  - slurs, 597
  - source structure, 591
  - type sizes, 596
- music scores (PMX)
  - % (percent sign), comment indicator, 619
  - allegro, *646*
  - allegro vivace, *644*
  - blocks, 622
  - body of file, 621
  - horizontal spacing, manual adjustment, 643
  - inline TeX commands, 646
  - instruments
    - clefs, *621*
    - definition, 617
    - names, 621
    - number of, 619
  - key signature, 620
  - lines, definition, 617
- music scores (PMX) *(cont.)*
  - logical meter, 620
  - lyrics, 647
  - meter representation, *620*
  - MIDI, 647
  - MIDI mnemonics, 649
  - notation, all voices
    - bar symbols, 639
    - bars, 639
    - global A options, 643
    - key changes, *641*
    - line breaks, 642
    - meter changes, 640
    - page breaks, 642
    - page layout, 642
    - page numbering, 642
    - repeats, 639
    - text blocks, *641*
    - title blocks, *641*
    - voltas, *640*
  - notation, staves
    - accidentals, 622, *624*, 628
    - arpeggio, *629*
    - basic duration, 622
    - beams, 631, *632, 633*
    - beams for xtuplets, *627, 628*
    - chords, *628, 629*
    - clef changes, *639*
    - definition, 617
    - dotted notes, 622
    - doubly dotted notes, 622
    - down fermata ornaments, 630
    - duration of notes, 622
    - dynamical marks, *638*
    - grace notes, 629, *630*
    - grace notes, in xtuplets, 627
    - height, 620
    - horizontal displacement, *624*
    - note parameters, *624, 625*
    - notes, 622, *623, 624*
    - number of, 619
    - octaves, *623*
    - on staves, 622–624
    - ornaments, *630*, 631
    - parameters, 623, *624, 625*
    - pitch, 622
    - pointed rhythms, *624*
    - rests, 625, *626*
    - slurs, *634–638*
    - staccato ornaments, 630
    - stems, 623, 624
    - tenuto ornaments, 630
    - ties, 634, 635, *637*
    - xtuplets, 626, *627, 628*

- music scores (PMX) *(cont.)*
  - notes
    - accidentals, 622, *624*, 628
    - basic duration, 622
    - dotted, 622
    - doubly dotted, 622
    - duration, 622
    - grace notes, 629, *630*
    - horizontal displacement, *624*
    - octaves, *623*
    - on staves, 622–624
    - parameters, 623, *624, 625*
    - pitch, 622
    - pointed rhythms, *624*
    - stems, 623, 624
  - numerical parameters, 619, 620
  - output path, 621
  - overview, 618
  - page height and width, 642
  - pages, number of, 620
  - parts of, 619
  - pickup bar length, *620*
  - pickups, 620
  - PMX commands, *650*
  - preamble of file, 619, 620, *621*
  - signature, *620*
  - splitting apart, 647, *648*
  - structure of a score, 619
  - symbols, definition, 617
  - systems
    - definition, 617
    - indentation, 620
    - number of, 620
  - voice, definition, 617
  - words, definition, 617
- music scores (TeX)
  - inline commands, 646
  - overview, 589, 590
  - with METAFONT, 666
- music scores (TeX*muse*), 666
- MusicTeX package, 589
- musixflx program, **595**, **597**, 599, 618
- musixlyr.tex package, 647, 659, 660
- musixpss program, 637
- MusiXTeX package, xxvi, xxviii, xxxi, 588, **589–599**, 602, 615–617, 623, 628, 634, 635, 646–648, 658, 660, 661
- MusiXTeX notation system, *see* music scores (MusiXTeX)
- `.mx1` file extension (MusiXTeX), **597**, 598, 599
- `.mx2` file extension (MusiXTeX), **597**, 598
- `myhexagon.sty` file (tlgc), xxxiii
- `\MyRot` (tlgc), *39*

N

- `n` syntax (PMX), *624*, **625**
- `Name:` syntax (M-Tx), *651*, **652**
- `named` syntax
  - (color), 720
  - (xcolor), 720, 722, *727*
- named colors
  - behavior options, 721
  - support for, 719
  - within documents, 725
- `\nameseq` (texshade), *549*
- `\namesit` (texshade), *549*
- `\namesrm` (texshade), *549*
- `\NAND` (circ), *578*
- `\nano` (Slunits), 515
- `\naphdrh` (carom), ***524, 525***, ***535***, *536*
- `\naphdrv` (carom), *524, 525*
- `\naphdrvb` (carom), *525*
- `\naphdrvt` (carom), *525*
- nassflow package, 15
- `natheight` key (graphicx), 28
- `natural` option (xcolor), 721
- natural symbol (musical), *605*
- `natwidth` key (graphicx), 28
- navigation bar, slides, 772, *773*, 774
- `navigation symbols` syntax (beamer), *773*, *777*
- `\nbb` (MusiXTeX), *599*
- `nc` syntax (PMX), 625
- nesting chess variations, 679
- netpbm program, 7
- nets, drawing, 15
- `\newcolumntype` (array), *738*
- `\newgame`
  - (skak), **674**, 675, *678, 679*
  - (texmate), 683
- `NEWMAN` syntax (m-ch-en), *542*
- news groups, 810, *see also* online resources
- `\newton` (Slunits), 514
- Newtonian mechanics symbols, *580*
- `\nextdiagrambottom` (texmate), **685**, *686*
- `\nextdiagramtop` (texmate), **685**, *686*
- `\nfet` (circ), *577*
- nicefrac package, 513
- `\nl` (circ), *581*
- `noamsthm` option (beamer), 753
- `\nobarnumbers` (MusiXTeX), *599*
- `\nodiagrammove` (texmate), 686
- `\nodiagramnames` (texmate), 686
- `\nodiagramnumber` (texmate), 686
- `\nodiagramturn` (texmate), 686
- `\nonaheteroh` (hetarom), *529*
- `\nonaheterohi` (hetarom), *529*
- `\nonaheterov` (hetarom), 520, ***529***, *530*
- `\nonaheterovi` (hetarom), ***529***, *539*
- `\nonamethylene` (methylen), *538*
- `\nonamethylenei` (methylen), *538*
- `nopstricks` option (pst-pdf), 800
- `\NOR` (circ), *578*

`normal` option (skak), 675
`normal text` syntax (beamer), *795*
`\normalboard`
  (bg), **697**, *698*
  (cchess), 690
  (skak), 675
`\normalgoban` (igo), 694
`\normalsize` (LilyPond), *663*
notation (chess)
  commentaries, *681*, 682
  overview, 680–683
  threats, *681*
  variations, *680*, *682, 683*
notation (musical), *see also* music scores (abc2mtex)
  all voices
    bar symbols, 639
    bars, 639
    global A options, 643
    key changes, *641*
    line breaks, 642
    meter changes, 640
    page breaks, 642
    page layout, 642
    page numbering, 642
    repeats, 639
    text blocks, *641*
    title blocks, *641*
    voltas, *640*
  staves
    accidentals, 622, *624*, 628
    arpeggio, *629*
    basic duration, 622
    beams, 631, *632, 633*
    beams for xtuplets, *627, 628*
    chords, *628, 629*
    clef changes, *639*
    definition, 617
    dotted notes, 622
    doubly dotted notes, 622
    down fermata ornaments, 630
    duration of notes, 622
    dynamical marks, *638*
    grace notes, 629, *630*
    grace notes, in xtuplets, 627
    height, 620
    horizontal displacement, *624*
    note parameters, *624, 625*
    notes, 622, *623, 624*
    number of, 619
    octaves, *623*
    on staves, 622–624
    ornaments, *630*, 631
    parameters, 623, *624, 625*
    pitch, 622
    pointed rhythms, *624*
notation (musical) *(cont.)*
    rests, 625, *626*
    slurs, *634–638*
    staccato ornaments, 630
    stems, 623, 624
    tenuto ornaments, 630
    ties, 634, 635, *637*
    xtuplets, 626, *627, 628*
`\notationOff` (skak), 675
`notationoff` option (skak), 675
`\notationOn` (skak), 675
`notationon` option (skak), 675
noteedit program, 588
`\NOTEs` (MusiXTEX), 595
`\NOTes` (MusiXTEX), **595**, *599*
`\NOtes` (MusiXTEX), 591, ***595***, *596*, *599*
`\Notes` (MusiXTEX), 591, *594*, ***595***, *596*, *599*
`\notes` (MusiXTEX), **591**, ***595***, 596, *599*
`notes` option (beamer), 753
notes (annotations), *see* annotations, *see* commentaries
notes (musical)
  accents (LilyPond), *663*
  accidentals, 622, *624*, 628
  basic duration, 622
  beams, 663
  chords (LilyPond), *663*
  commands, 595
  describing staves, 622, *623*, 624
  dotted, 622, *624*
  doubly dotted, 622
  duration, 622
    LilyPond, 662, *663*
  examples, *592*
  grace notes
    `{}` (curly braces), *607*
    `~` (tilde), *607*
    in xtuplets, 627
    LilyPond, 663
    PMX, 627, 629, *630*
  horizontal displacement, *624*
  internote spacing, 602
  key (LilyPond), *662*
  length, 601, *603, 604*
  notation, *661*
  octaves, *623*
  on staves, 622–624
  ornaments (LilyPond), *664*
  parameters, 623, *624, 625*
    accidentals, *624, 625*
    beam inhibit, *624, 625*
    dotted notes, *624, 625*
    shift of position, *624, 625*
    stems, *624, 625*
    xtuplets, *625*

- notes (musical) *(cont.)*
  - pitch
    - abc2mtex, *603, 604*
    - LilyPond, *662*
    - MusiXTEX, 590
    - specifying, 593, 622
  - pointed rhythms, *624*
  - slurs (LilyPond), *663, 664*
  - spacing, *595*
  - stems, 623, 624
  - symbols, *592*, 593, *594*
  - timing, 590
  - triplets (LilyPond), *664*
- \noteskip rigid length (MusiXTEX), 595
- \NOTesp (MusiXTEX), 595
- \NOtesp (MusiXTEX), **595**, *599*
- \Notesp (MusiXTEX), 595
- \notesp (MusiXTEX), 595
- notheorems option (beamer), 753
- notightpage option (pst-pdf), 800
- \npn (circ), ***577***, *581*
- \NRSFF (circ), *579*
- \Nterm (textopo), *553*
- nucleotide sequences
  - aligning, 548–550
  - highlighting, *548–550*
  - sequence fingerprints, *550*
  - shading, 548–550
- \NULL (circ), *579*
- number puzzles, 707, *708*, *see also* crosswords
- numbers, symbols for, 512
- \nv (circ), 579
- \nvmos (circ), *577*

O

- \O (chemsym), *517*
- o(syntax (PMX), *630*, **631**
- o) syntax (PMX), *630*, **631**
- o+ syntax (PMX), *630*, **631**
- o. syntax (PMX), *630*, **631**
- o.: syntax (PMX), *630*
- O: syntax (abc), *608*
- o: syntax (PMX), *630*
- o> syntax (PMX), *630*, **631**
- o^ syntax (PMX), 631
- o_ syntax (PMX), *630*, **631**
- o~ syntax (PMX), *630*
- \oa (circ), *581*
- ob syntax (PMX), *630*, **631**
- object-oriented drawings, 4, *5*
- oc syntax (PMX), *630*, **631**
- \octamethylene (methylen), *538*
- \octamethylenei (methylen), *538*
- Octave program, 2
- Octave: syntax (M-Tx), 652
- octaves (musical), *623*
- Octaviz program, 2
- \octfindown (MusiXTEX), 592
- \octfinup (MusiXTEX), 592
- Octplot program, 2
- oe? syntax (PMX), *630*, **631**
- oef syntax (PMX), *630*, **631**
- oef? syntax (PMX), 631
- oen syntax (PMX), *630*, **631**
- oen? syntax (PMX), 631
- oes syntax (PMX), *630*, **631**
- oes? syntax (PMX), *630*, **631**
- of syntax (PMX), *630*, **631**
- ofd syntax (PMX), *630*, **631**
- OFF syntax (m-ch-en), 546
- og syntax (PMX), *630*, **631**
- \OH (chemsym), 517
- \ohm (SIunits), 514
- oldgate option (circ), 577
- \oldGclef (MusiXTEX), 592
- \OM (circ), *581*
- om syntax (PMX), *630*, **631**
- ONE syntax (m-ch-en), ***542***, 546
- online access to CTAN, 810, 811, *812, 813*, 814
- online resources
  - Adobe Illustrator, 1
  - Adobe Photoshop, 17
  - archived files, finding and transferring, 813
  - automata diagrams, 15
  - CGM-Open Consortium, 13
  - CTAN (Comprehensive TEX Archive Network), 810
    - web access, 810, 811, *812*, 813, 814
  - dedicated drawing tools, 1, 2
  - documentation
    - command-line interface, *815*
    - panel interface, *816*
    - search by name, *815*
    - search by product, *816*
    - texdoc, *815*
    - texdock, *816*
  - DVI to SVG conversion, 13
  - FAQs (Frequently Asked Questions), 809
  - files, getting from the command line, *814*
  - How To Ask Questions The Smart Way, 810
  - nets, drawing, 15
  - news groups, 810
  - PDF viewers, 12
  - plotting programs, 17
  - program files, obtaining
    - web access, 810, 811, *812*, 813, 814
  - TEX file catalogue, 811
  - TEX files, 810
  - TEX user groups, 817, 818
  - TUG home page, 810, *811*
- \only (beamer), 766, *767*, *775*, ***780***, *785*, 786, **792**

only key value (beamer), 753
\onlyenv (beamer), 769
onlyenv env. (beamer), *769*, **770**
onlyslideswithnotes key value (beamer), 753
onlytextwidth key (beamer), 781
\onslide (beamer), ***763***, **764**, ***765***, 767
\OO (chemsym), 517
op syntax (PMX), *630*, **631**
\opaqueness (beamer), 767, 768
opaqueness, slides, *768*
openoffice program, 21
optics option (circ), 577
optics diagrams, *see also META and* PSTricks *index*
  example, *581*
  font for, 576–582
  symbols, *580*
Options: syntax (M-Tx), 652
\OR (circ), *578*
orange syntax (xcolor), 726
origin key (graphicx), **28**, *33*, **40**, 41
original option (pict2e), 43
ornaments (musical)
  description, 630, 631
  example, *630*
  LilyPond, *664*
  table of, 631
\oscillograph (circ), *578*
oscilloscope channels, *see* PSTricks *index*
oT syntax (PMX), *630*, **631**
ot syntax (PMX), *630*, **631**
oT0 syntax (PMX), *630*
oT1 syntax (PMX), *630*
oTO syntax (PMX), 631
oTt syntax (PMX), *630*, **631**
ou syntax (PMX), *630*, **631**
\Oval (axodraw), 559
\oval, *43*
  (pict2e), *43*, **45**, *46*
ovals, drawing, 45, *46*
overlayarea env. (beamer), 770
overlays, slide, *see* slides (color), overlay specification
overprint env. (beamer), 770
ox syntax (PMX), *630*, **631**
\oxazolev (hetarom), 530
\oxazolevi (hetarom), 530
\oxqu (MusiXTEX), 592
oztex option (pict2e), 43

P

P syntax (PMX), 642
\P (chemsym), 517
packages
  PICTEX, 5, **13**, **14**, 541
  XY-pic, xxvi, xxviii, 5, 9, 16, *see also XY-pic index*
  XΥMTEX, 520–540

packages *(cont.)*
  abc, 612–615
  AlDraTex, 15
  aliphat, 520, **532**
  alltt, 790
  amsmath, 361, 483, 484, 752, 753, 759
  amssymb, 515
  amstex, 517
  amsthm, 753
  array, 737, **764**
  arrayjob, 322
  axodraw, 555, **558–561**
  babel, 124, 515
  bar, 15, 162
  beamerouterthemesidebar, 774
  bg, 696–698
  bridge, 699–702
  calc, 323
  carom, 520, **524**
  cchess, 687–690
  ccycle, 520, **530**
  chemist, 537, 540
  chemstr, 520
  chemsym, 512, **517, 518**, 519
  chess, 668, 677, 680, 687, 690, 691
  chessboard, **668, 669**, 673
  chessfss, 668, **669–673**, 674, 678, 680
  chmst-ps, 537
  circ, 576–582
  color, 215, 216, 235, 304, 719–722, 726, 728, 730, 737
  colordvi, 719
  colortbl, 720, 721, **737–751**
  createsudoku, 710–712
  crosswrd, 702–704
  curve2e, 47–50
  curves, 15, 47
  cwpuzzle, **704–708**, 709
  dcolumn, 737
  diagram, 482
  diagxy, 482
  diversity, 549
  DraTex, 5, 15
  eepic, 17, 20, 511, 521, 522
  emp, 120, 121, 167
  enpassant, 670
  epic, 15, 511, 520–522, 537
  epsfig, 42
  extsizes, 753
  feyn, 555–558
  FeynArts, 555
  feynman, 555
  feynmf, 120, **561–572**
  feynmp, 120, **562**, 572
  foiltex, 719
  fontenc, 752

packages *(cont.)*
- fp, 458
- fusering, 537
- gastex, 15, **438, 439**
- go, 690, 691
- graphics, 2, 3, 7, 8, 10, **23–27**, 30, **33–40**, 72, 277, 791
- graphicx, **23–25**, **28–42**, 800
- hcycle, 520, **532**
- hepnicenames, 512, 560
- heppennames, 512, 560
- hepunits, 516, 517
- hetarom, 520, **528**, 530, **534**
- hetaromh, 520, **528**, **534**
- hhline, 737, 742, **750**
- hyperref, 721, 753, **783**, 798, 803–805
- ifthen, 136, 323, 503
- igo, 691–695
- infix-RPN, 430
- inputenc, 752, 753
- isotope, 518
- keyval, 33, 217
- listings, 790
- locant, 520
- longtable, **517**, 737, 742
- lowcycle, 520, **526**
- m-ch-en, 541–547
- makecirc, 576
- makeplot, 430
- mathptm, 65
- methylen, 537
- mfpic, 21, 52, 120, **122–136**, 139, 583
- mproof, 73, 74
- mpsproof, 73, 74
- multido, 216, **458, 459**
- multimedia, 774
- MusicTEX, 589
- MusiXTEX, xxvi, xxviii, xxxi, 588, **589–599**, 602, 615–617, 623, 628, 634, 635, 646–648, 658, 660, 661
- musixlyr.tex, 647, 659, 660
- nassflow, 15
- nicefrac, 513
- paralist, 683
- pict2e, 7, 15, **42–47**, 511
- pictexwd, 14
- pifont, 724
- polymers, 537
- ppchtex, 541–547
- preview, 458, 800–802
- printsudoku, 710–712
- psfrag, 5
- psgo, 691
- pspicture, 47, 511
- pst-3d, 216, **388–400**
- pst-3dplot, 217, 234, 313, 388, **400–416**
- pst-all, **216**, 313

packages *(cont.)*
- pst-asr, 217, **424**
- pst-bar, 450
- pst-barcode, 453
- pst-blur, 449, 450
- pst-calendar, 452
- pst-circ, 309, **435**
- pst-coil, 216, **455, 456**
- pst-dbicons, 445
- pst-eps, 216, **457**
- pst-eucl, VIII, **426**
- pst-fill, 216, 255, 257, **383–387**
- pst-fr3d, 388, **447**
- pst-fractal, 456, 457
- pst-func, 427
- pst-geo, 437, 438
- pst-gr3d, 388, **447**
- pst-grad, 216, **448**
- pst-infixplot, 429, 430
- pst-jtree, 425
- pst-labo, 433
- pst-lens, 452
- pst-light3d, 447
- pst-map2d, 438
- pst-map2dll, 438
- pst-map3d, 438
- pst-map3dll, 388, 438
- pst-math, 224, **428**, 429
- pst-node, 214, 216, 313, **334–366**, 379, 424
- pst-ob3d, 388, **446**
- pst-optic, 434
- pst-osci, 434
- pst-pdf, **457, 458**, 797, **800–803**, 805, 806
- pst-pdgr, 431
- pst-plot, 214, 216, 266, **313–334**, 400, 406, 424, 426
- pst-poly, 431
- pst-slpe, 449
- pst-spectra, 432
- pst-stru, 436
- pst-text, 216, **451**
- pst-tree, 214, 216, **366–382**, 424
- pst-uml, **442**, 443
- pst-view3d, 400
- pst-vue3d, 388, 393, **445**
- pst-xkey, 217, **310–312**
- pstcol, 215
- pstricks, **213–466**, 515, 797, 800
- pstricks-add, 224, 257, 318, 323, **418–424**
- rotating, **42**, 392
- rrgtrees, 424, 425
- sfg, 442
- SIstyle, 513
- SIunits, 513–516
- sizeredc, 537
- skak, 668, 669, **673–679**, 680, 682

- packages *(cont.)*
  - slashed, 557
  - SliTeX, 752
  - solvesudoku, 710–712
  - sudoku, 709, 710
  - texmate, 668, 669, 673, 679, **680–687**
  - texshade, **547–550**, 552
  - textopo, 547, **551–555**
  - tikz, 5
  - timing, 572–576
  - tlgc, 835
  - ucs, 753
  - uml, 443
  - units, 513
  - unitsdef, 513
  - vaucanson-g, 439, 440
  - xcolor, 7, 215, 216, 235, 258, 304, 406, 713, **719–737**, 740, 747, 753
  - xkeyval, **217**, 310
  - xq, 688
  - xyling, 491
  - xymtex, 520, 537
  - xymtexps, 537
  - xymtx-ps, 537
  - xytree, 491
- padding key (chessboard), *669*
- \pagecolor (xcolor), 720, **725**
- Pages syntax (M-Tx), *655*
- pages (musical)
  - breaks, 642
  - layout, 642
  - numbering, 642
- Pages: syntax (M-Tx), 652
- paralist package, 683
- \parbox, *37*, *40*
- parens ((...)), slur symbol, *607, 608*
- parent key (beamer), *778*, 793
- \part (beamer), 779
- part key (beamer), *782*, **783**
- Part: syntax (M-Tx), 652
- \pascal (SIunits), 514
- \PAUSe (MusiXTeX), 592
- \PAuse (MusiXTeX), 592
- \pause
  - (MusiXTeX), 592, **594**
  - (beamer), *763*, 764, 765, 783
- \pausep (MusiXTeX), 592
- pausesections key (beamer), *782*, **783**
- pausesubsections key (beamer), 783
- \pawn (chessfss), 672
- PBM (portable bitmap) format, 7
- pbmtopk program, 7
- PCTeX program, 11
- pctex32 option
  - (graphics/graphicx), 24
  - (xcolor), 721
- pctex32 program, 24
- pctexhp option
  - (graphics/graphicx), 24
  - (xcolor), 721
- pctexhp program, 24
- pctexps option
  - (graphics/graphicx), 24
  - (xcolor), 721
- pctexps program, 24
- pctexwin option
  - (graphics/graphicx), 24
  - (xcolor), 721
- pctexwin program, 24
- PDF language, 11, 12
- .pdf file extension (pst-pdf), 806
- pdfcrop program, 804
- pdfinfo program, 804
- pdflatex program, xxvi, xxviii, 6, 7, 797, 800, 801, 803, 805, 806
- PDFs
  - creating
    - dvipdfm program, 798–800
    - dvipdfmx program, 798–800
    - from LaTeX, 803–807
    - from PostScript, 800, *801*, 802, 803
    - music scores, 614
    - overview, 797
    - pst-pdf package, 800, *801*, 802, 803
  - description, 11, 12
  - viewers, 12
  - vs. PostScript, 11, 12
- pdftex option
  - (graphics/graphicx), 24
  - (pict2e), 43
  - (xcolor), 721
- pdftex program, 14, 24, 618, 721, 797, 798
- pdftops program, 806
- \PED (MusiXTeX), 592
- \pentamethylene (methylen), *538*
- \pentamethylenei (methylen), *538*
- peptide sequences
  - aligning, 548–550
  - highlighting, *548–550*
  - sequence fingerprints, *550*
  - shading, 548–550
- \per (SIunits), *516*
- percent sign (%), comment indicator, 619
- Periodic Table of the Elements, 519
- pertab.tex file (chemsym), 517
- \peta (SIunits), 515
- \pfet (circ), *577*
- pgfborder key (chessboard), *669*
- \pgfdeclareimage (beamer), *776*, *777*, *792*

- \pgfuseimage (beamer), *777*, *792*
- pgn2ltx program, 687
- phenanthrene derivatives, *525*
- \phenanthrenev (carom), 524, ***525***
- photographs, 4
- \Photon (axodraw), 559, *561*
- \PhotonArc (axodraw), 559
- photons (Feynman diagrams), *561*
- physics option (circ), 577
- physics diagrams, *see META index*
- \PianoStaff (LilyPond), *665*
- .pic file extension, xxxi
- pic language, 17–20
- pic program, 17, 583, 585
- pickups (musical), 620, 654
  - bar length, *620*
- \pico (SIunits), 515
- \picobarn (hepunits), *516*
- pict2e package, 7, 15, **42–47**, 511
- PICTeX package, 5, **13, 14**, 541
- pictexwd package, 14
- picture env., xxvii, 5–7, 9, 15, 16, 19, 20, 44, 520, 534, 541, 555, 568, 573, 797
  - (axodraw), 559
  - (cwpuzzle), 705, 708
  - (pict2e), 42
- pictures, *see also* drawing
  - character-based, 13
  - from fonts, 13
  - photographs, 4
  - pic language, 17–20
- pie charts, *see META index*
- \piece (cchess), 688, 689, ***690***
- piececolor key (chessboard), *669*
- pifont package, 724
- pin connections, 579
- \Pinhole (circ), ***580***, *581*
- pitch (musical)
  - abc notation system, *603*
  - abc2mtex, *603*
  - LilyPond, *662*
  - MusiXTeX, 590, 593
  - PMX, 622
- .pk file extension (feynmf), 563
- placement, *see* positioning
- plain key (beamer), **759**, *792*
- plotting, *see also* graphs
  - drawing tools for, 2, 17
  - gnuplot, 17, *18*
  - programs for, 17
- PLUS syntax (m-ch-en), 546
- plus sign (+), color expression, 732
- \PM (circ), *580*
- PMX language, xxviii, 616, 617, **618–649**, 651–654, 656, 657, 659, 660
- .pmx file extension, xxxi
  - (PMX), 618, 647
- PMX notation system, *see* music scores (PMX)
- PMX: syntax (M-Tx), 652
- pmxab program, 590, **618–649**, 651
- pmxaerr.dat file (PMX), 618
- .png file extension (pst-pdf), 806
- \pnp (circ), *577*
- Poet: syntax (M-Tx), 652
- pointed rhythms (musical), *624*
- \Polar (circ), ***580***, *581*
- polygon keywords (Feynman diagrams), 567, 568
- \polyline (curve2e), 47, *49*
- polymers package, 537
- polymethylene commands, 538
- portable bitmap (PBM) format, 7
- \position (texmate), *682*, ***684***
- position env.
  - (bg), ***696***, 697, 698
  - (cchess), **688**, 689, *690*
- postit syntax (beamer), *776*
- PostScript
  - description, 10, 11
  - drivers, 11
  - Feynman diagrams, 558–561
  - from TeX DVI, 11
  - PDFs from, 800, *801*, 802, 803
  - viewing, 10, 11
  - vs. PDF, 11, 12
- PostScript language, 10, 11
- postscript env. (pst-pdf), 802
- \power (SIunits), *516*
- \PP (chemsym), 517
- \pp (LilyPond), *664*
- ppchtex package, 541–547
- \Pr (chemsym), 517
- \pr (chemsym), 517
- \preparediagram (texmate), 685
- prepmx program, 651–660
- presentation option (beamer), 753
- presentations, *see* slides
- preview package, 800–802
- \PreviewEnvironment (pst-pdf), 801
- primary colors, 717
- \printarrow (skak), *676*
- \printboard (bg), **697**, *698*
- printing
  - chess board, 675
  - chess moves, 675, 677
- \printknightmove (skak), *676*
- printsudoku package, 710–712
- program files, obtaining
  - web access, 810, 811, *812, 813*, 814
- prologue option (xcolor), 721
- proof env. (beamer), 753, *769*

\protect (igo), 695
\providecolor (xcolor), **726**, *727*, 728
\providecolorset (xcolor), **727**, *728*
.ps file extension (graphics/graphicx), 35
ps option (skak), *676*
.ps.bb file extension (graphics/graphicx), 35
.ps.gz file extension (graphics/graphicx), 35
ps2eps program, 615
ps2epsi program, 615
ps2pdf program, 797, 801–806
ps2pdf13 program, 804, 805
psfrag package, 5
\psframebox (xcolor), *733*
psgo package, 691
psmatrix env. (pst-pdf), 800
pspicture env. (pst-pdf), 800
pspicture package, 47, 511
pst-eucl package, VIII
pst-pdf package, 797, **800–803**, 805, 806
\pst@object (pst-pdf), 800
pstarrows option (pict2e), *44*
PSTricks, *see* PSTricks *index*
pstricks option (pst-pdf), 800
pstricks package, 515, 797, 800
\pt (MusiXTeX), 594
\pteridinev (hetarom), 530
\pteridinevi (hetarom), 530
\PText (axodraw), 559
.ptx file extension, xxxi
\purinev (hetarom), 520, 530
\purinevi (hetarom), 530
purity of color, 718
purple syntax (xcolor), 726
\put
(curve2e), *48, 49*
(cwpuzzle), *705*
Puzzle env. (cwpuzzle), ***704***, *705*, *707, 708*
\PuzzleBlackBox (cwpuzzle), 708
\PuzzleClueFont (cwpuzzle), *708*
PuzzleClues env. (cwpuzzle), *705*
\PuzzleFont (cwpuzzle), 708
\PuzzleHook (cwpuzzle), 705, **708**
\PuzzleLetters (cwpuzzle), *708*
\PuzzleLettersText (cwpuzzle), *708*
\PuzzleNumberFont (cwpuzzle), *708*
\PuzzleNumbers (cwpuzzle), *708*
puzzles, *see* crosswords, *see* Sudoku
\PuzzleSolution (cwpuzzle), **705**, *706*, *708*
\PuzzleUnitlength rigid length (cwpuzzle), *708*
\PuzzleUnsolved (cwpuzzle), 705
PuzzleWords env. (cwpuzzle), *707*
\PuzzleWordsText (cwpuzzle), *707*
\pvmos (circ), *577*
\pyranose (hcycle), *532*
pyranoses derivatives, *532*
\pyrazinev (hetarom), ***524***, 530
\pyrazolev (hetarom), 530
\pyrazolevi (hetarom), 530
\pyridazinev (hetarom), 530
\pyridazinevi (hetarom), 530
\pyridinev (hetarom), 530
\pyridinevi (hetarom), 530
\pyrimidinev (hetarom), 530
\pyrimidinevi (hetarom), 530
\pyrrolev (hetarom), 530
\pyrrolevi (hetarom), 530
Python program, 661

Q

\Q (circ), *577*
Q: syntax (abc), 602, *610*
\qa (MusiXTeX), **593**, *594, 595*
\qb (MusiXTeX), *596*, **597**, *599*
\qbezier, 46, *47*
(pict2e), 46, *47*
\qbeziermax, 46
\ql (MusiXTeX), 592, **593**, *596*, 597, *599*
\qlp (MusiXTeX), *599*
\qp (MusiXTeX), 592, **594**, *599*
\qqs (MusiXTeX), 592
\qs (MusiXTeX), 592
\qu (MusiXTeX), 592, **593**, *594–596*, 597
quadratic Bézier curves, 46, *47*
quadruplets (musical), *605*
\queen (chessfss), 672
\quinazolinev (hetarom), 530
\quinazolinevi (hetarom), 530
\quinolinev (hetarom), 530
\quinolinevi (hetarom), 530
\quinoxalinev (hetarom), 530
\qupp (MusiXTeX), 592

R

R syntax
(PMX), 639
(m-ch-en), 542, 544
\R (circ), ***577***, *581*
r syntax (PMX), **625**, *626*, **628**
\r... (MusiXTeX), 594
R: syntax (abc), *608*
radii, specifying, 45, *46*
rand (pic), 19
\rarw (timing), 575
\rawboard (bg), 697
Rb syntax (PMX), **639**, *640*
rb syntax (PMX), 625, *626*
RD syntax (PMX), **639**, *640*
Rd syntax (PMX), **639**, *640*
Rdl syntax (PMX), *640*

\Re (chemsym), 517
\re (chemsym), 517
reaction equations, *545*
reaction schemes, *540*
\reactrarrow (chemist), *540*
read key (graphicx), **29**, 34
readability, and color, 718
\reciprocal (Slunits), *516*
rect (pic), *19*
red syntax (xcolor), ***722***, **726**, 727
\reduceallcells (solvesudoku), *711*
\reducedsizepicture (xymtex), *538*
\reflectbox (graphics/graphicx), *37*
\relative (LilyPond), *662–665*
repeat symbols (musical), *603*
repeats (musical), 639
\RequirePackage, xxxii
\resetcolorseries (xcolor), **734**, *735, 736*
\resigns (texmate), *683*
\resizebox
    (graphics/graphicx), ***38***, *39*
    (graphics), *27*
\resizebox* (graphics/graphicx), **38**, *39*
resizing
    bounding box, *27*
    graphic objects, *38, 39*
    text, *38, 39*
\restoregame (skak), 679
rests (musical), *592*, 625, *626*
    LilyPond, *663*
\reverseallabreve (MusiXTEX), 592
\reverseC (MusiXTEX), 592
RGB option (xcolor), 721
RGB syntax (xcolor), 728, 729
rgb option (xcolor), 721
rgb syntax
    (color), 720
    (xcolor), 720, *722*, 727–729, 732
RGB (Red, Green, Blue) color, **715**, 719
\rh (MusiXTEX), *594*
right (pic), *19*
right key (beamer), 777
\rightdiagramturn (texmate), 686
\rightrepeat (MusiXTEX), 592
rightskip key (beamer), 777, *794*
Rl syntax (PMX), *640*
Rlr syntax (PMX), *640*
rm syntax (PMX), 625, *626*
\rmoiety (chemstr), *522*
\rook (chessfss), 672
\roqu (MusiXTEX), 592
Rosegarden program, 588
ROT syntax (m-ch-en), **544**, 545
rotate env. (rotating), *42*
\rotatebox
    (graphics/graphicx), 36, ***39***, *40*
    (graphics), *27*
    (graphicx), 24, 33, *39*, ***40***, 42
rotated material, hiding, 25
\rotategoban (igo), 695
\rotategobanleft (igo), 695
\rotategobanright (igo), 695
rotating
    bounding box, *27*, *31, 32*
    chemical structures, *544, 545*
    Go board, *695*
    graphic objects, *39–42*
    \includegraphics keys, 29
    reference points, 40–42
rotating package, 42
rounded key (beamer), 777, *778*
\rowcolor
    (colortbl), ***739***, 740, 741, *747, 748*, *750, 751*
    (xcolor), *763*, *765*
\rowcolors (xcolor), **740**, 741, *751*, *763*, *765*
rows (table), color
    alternate, *739, 740*
    selected, *746*
rp syntax (PMX), 625, *626*
\rpcubed (Slunits), *516*
rpo syntax (PMX), 625, *626*
\rq (MusiXTEX), *596*
Rr syntax (PMX), 640
\rsqu (MusiXTEX), 592
\rtetrahedralS (aliphat), *540*
\rText (axodraw), 559
\rtrigonal (aliphat), *533*
\Rvar (circ), *577*
\ryl (chemstr), *535, 536*
RZ syntax (m-ch-en), **542**, 543, **544**
Rz syntax (PMX), *640*

S

S syntax (m-ch-en), 544
\S
    (chemsym), 517
    (circ), *577*
s syntax
    (LilyPond), 662
    (PMX), *624*, **625**, ***630***, **634**, 648
sample.sud file (tglc), *710, 711*
saturation, 717
\savegame (skak), 679
SB env. (chemsym), 517
SB syntax (m-ch-en), 544
Sb env. (amstex), 517
\sbox, 725
sc syntax (PMX), 625
Scalable Vector Graphics (SVG), 12, 13

- `scale` key (graphicx), **29**, *30*
- `\scalebox`
  - (beamer), 774
  - (graphics/graphicx), *37*
  - (graphics), *27*
- scaled material, hiding, 25
- `\scaletopo` (textopo), *551*, *553*
- scaling
  - bounding box, *27*, 29
  - graphic objects, *37*
  - `\includegraphics` keys, 29, *30*
  - text, *37*
- scaling factor, 29, *30*
- Scheme program, 661
- scid program, 687
- science diagrams, *see* PSTricks *index*
- scientific texts, *see also* bioinformatics, *see also* chemical formulas, *see also* Feynman diagrams
  - abbreviations, 513
  - chemical elements, symbols for, 512
  - chemical symbols, 517, *518*
  - consistency, 512
  - “d” in integrands, 513
  - electronics diagrams
    - drawing position, moving, 580
    - electronic box symbols, *578*
    - examples, *581, 582*
    - font for, 576–582
    - gate symbols, *578*
    - integrated circuit symbols, *579*
    - interactive generation, 586
    - junctions, 579
    - m4 macro processor, *583–585*
    - pin connections, 579
    - symbol connections, 579
    - symbols, *577*
    - trigger symbols, *578*
  - mathematical functions, symbols for, 512
  - Newtonian mechanics symbols, *580*
  - numbers, symbols for, 512
  - optics diagrams
    - experimental setup, *581*
    - font for, 576–582
    - symbols, *580*
  - state names, symbols for, 513
  - symbols, 512
  - table of, 512
  - timing diagrams
    - annotation, 573
    - arrows, 575
    - customizing, 576
    - fonts, specifying, 573
    - labels, 573
    - overview, 572–576
    - separation between lines, 576
- scientific texts *(cont.)*
    - signal lines, 573
    - symbols argument, 573, *575*
    - timing values, 573
    - vertical line adjustment, 576
    - vertical lines, 576
  - units
    - base, 514
    - combining, *516*
    - derived, 514
    - high-energy physics, *516*
    - prefixes, 514
    - SI (International System of Units), 512–516
    - spacing between, *515*
    - symbols for, 512
    - typeset style, 515
  - wave names, symbols for, 513
- Scientific Word program, 24
- scor2prt program, 647
- `\ScrL` (circ), ***580***, *581*
- `\ScrTL` (circ), *580*
- `\sDEP` (MusiXTeX), 592
- `\second` (SIunits), 514, *516*
- secondary colors, 717
- `\section` (beamer), 779
- sectioning commands, slides, 779
- `sections` key (beamer), 783
- `sectionstyle` key (beamer), 783
- `\segno` (MusiXTeX), 592
- `\selectcolormodel` (xcolor), 730
- self-contained object-oriented drawings, 4
- `semiverbatim` env. (beamer), *790, 791*
- `sep` key (beamer), *776*, 777
- `\seqtype` (texshade), *549*
- `\sequence` (textopo), 551, *553*
- sequence fingerprints, *550*
- `series` key (beamer), 793, *794*
- `series*` key (beamer), 793
- `\setbeamercolor` (beamer), *760*, *776*, *778*, **793**, *794*
- `\setbeamercovered` (beamer), *760*, 767
- `\setbeamerfont` (beamer), *778*, **788**, *789*, **793**, *794*
- `\setbeamertemplate` (beamer), *773*, 774, *777, 778*, **793**, *794, 795*
- `\setboardfontfamily`
  - (chessfss), *673*
  - (skak), *675*
- `\setboardfontsize` (chessfss), *673*
- `\setchessboard` (chessboard), *669*
- `\setchessfontfamily`
  - (chessfss), 673
  - (skak), *678, 679*
  - (texmate), *683*, *686*
- `\setclef` (MusiXTeX), *596*
- `\SetColor` (axodraw), 559
- `\setends` (texshade), *548–550*

- \setfigfontfamily
  - (chessfss), *670*, ***671***
  - (skak), *678*
- \setfigstyle (chessfss), *672*
- \setinffontfamily (chessfss), 673
- \setkeys
  - (graphicx), *33*
  - (keyval), 33
- \SetOffset (axodraw), 559
- \SetPFont (axodraw), 559
- setpieces key (chessboard), *669*
- \SetScale (axodraw), 559
- \SetScaledOffset (axodraw), 559
- \setstaffs (MusiXTEX), *596*
- \setsudrandom (createsudoku), *711*
- \setTextDecresc (LilyPond), *664*, 665
- \settextfigchars (chessfss), *672*
- \settextfigfontfamily (chessfss), *672*
- \settextfiglanguage (chessfss), *672*
- \setupboard (skak), 675
- \setupchemical (m-ch-en), ***541***, *545*
- \setvolta (MusiXTEX), 592
- \setvoltabox (MusiXTEX), 592
- \SetWidth (axodraw), 559
- \sh (MusiXTEX), 593
- \shadincolors (texshade), *550*
- shading
  - color, *731*
  - nucleotide sequences, 548–550
  - peptide sequences, 548–550
- \shadingmode (texshade), *549, 550*
- shadow key (beamer), *776*, 777, ***778***
- \Shake (MusiXTEX), 592
- \shake (MusiXTEX), 592
- \Shakel (MusiXTEX), 592
- \Shakene (MusiXTEX), 592
- \Shakenw (MusiXTEX), 592
- \Shakesw (MusiXTEX), 592
- shape key (beamer), *789*, 793
- shape* key (beamer), 793
- sharp symbol (musical), *605*
- Sharps: syntax (M-Tx), **652**, *658*, *660*
- \shift (circ), **580**, *581*
- shortenstart key (chessboard), *669*
- \shortstack (igo), *693–695*
- show key value (beamer), 753
- \showall (skak), **676**, *677*
- \showallbut (skak), **676**, *677*
- \showboard
  - (skak), ***675***, *676–678*
  - (texmate), 680, 684
- \showconsensus (texshade), *548*
- \showcube (bg), *696*, **697**
- showerrors option (xcolor), 721
- \showfullgoban (igo), *693*
- \showgoban (igo), *692*, **693**, *694, 695*
- showing, *see* hiding/showing
- \showinverseboard (skak), 675
- \showlegend (texshade), *550*
- showmover key (chessboard), *669*
- \showmoverOff (skak), 676
- \showmoverOn (skak), *676*
- \showmoves (bg), 698
- \shownames (texshade), *549*
- \shownumbers (bg), *696*, **697**
- \showonly (skak), **676**, *677*
- \showonlyblack (skak), 676
- \showonlywhite (skak), 676
- \showrowcolors (xcolor), 740
- \showruler (texshade), *549*
- shrink key (beamer), 759
- SI (International System of Units), 512–516
- Sibelius program, 588
- sidebar left syntax (beamer), *773*
- sidebar right syntax (beamer), *777*
- sidewaysfigure env. (rotating), 42
- sidewaystable env. (rotating), 42
- \sievert (SIunits), 514
- signal lines, 573
- sin (pic), 19
- single-object drawings, 3, 4
- SIstyle package, 513
- \SIunits (SIunits), *515*
- SIunits package, 513–516
- SIunits.cfg file (SIunits), 516
- SIX syntax (m-ch-en), *542*
- \sixfuseh (fusering), 537
- \sixfusehi (fusering), 537
- \sixfusev (fusering), 537
- \sixfusevi (fusering), 537
- \sixheteroh (hetarom), *529*
- \sixheterohi (hetarom), *529*
- \sixheterov (hetarom), ***523***, *528*, ***529***
- \sixheterovi (hetarom), *529*
- \sixunitv (hetarom), *534*
- Size syntax (M-Tx), *655*
- size key (beamer), *778*, 793, *794*
- size* key (beamer), 793
- Size: syntax (M-Tx), 652
- sizeredc package, 537
- \sk (MusiXTEX), 595
- skak package, 668, 669, **673–679**, 680, 682
- \SkakOff (texmate), *680*, 682
- \slashed (slashed), *557*
- slashed package, 557
- \SLens (circ), ***580***, *581*
- \slide (MusiXTEX), 592
- slides document class, 713

- slides (color)
  - choosing colors, *756*
  - creating, 754–758
  - fonts, *758*
  - frames, creating, 758
  - hiding/showing, *see* slides (color), overlay specification
  - macros, *758*
  - main features, 752
  - modes, 752
  - options
    - beamer class, 752
    - conditional, 760
    - `frame` environment, 759
  - presentation structure, 758, 759, *760, 761*
  - styles, 754
  - tables, 780
  - templates, 754
  - themes, *754–757*
  - title pages, *761*
  - titles, 759
- slides (color), overlay specification
  - actions, *770*
  - animation, 774
  - bibliographies, *782*
  - block environments, *778, 779*
  - boxed text, *775, 776*
  - colored text, *775, 776*
  - creating, *763*
  - definition, 760, 762
  - dissolves, *774, 775*
  - dynamic text, holding static, *770*
  - figures, 780
  - footnotes, *789*
  - for existing LaTeX environments, *769*
  - framing text, *775, 776*
  - graphics, *792*
  - hiding/showing
    - alternative text, *769*
    - opaqueness, *768*
    - slide elements, *767*
    - specific rows, *765*
    - successive columns, *763*
    - successive rows, *763*
    - transparency, *768*
  - highlighting parts of elements, *771*
  - hyperlinks, *784–818*
  - labels, *785*
  - list items, *786–788*
  - logos, *776, 777*
  - movies, 774
  - multiple columns, *780*
  - navigation bar, 772, *773*, 774
  - overlay areas, *770*
  - preformatted text, *790, 791*
  - sectioning commands, 779

- slides (color), overlay specification *(cont.)*
  - sound, 774
  - source code representation, *791*
  - specifying, *765*
  - table of contents, 782
  - tables, 780
  - text styles, *789*
  - transitions, *774, 775*
  - verbatim text, *790, 791*
  - video, 774
- `\sline` (timing), *574*, **576**
- SliTeX package, 752
- slope arguments, 44
- slurs (musical)
  - abc2mtex, *607*
  - blind, *655*
  - broken, *655*
  - description, 654, 655
  - dotted, *655*
  - K type, *636*
  - LilyPond, *663, 664*
  - M type, *637, 638*
  - MusiXTeX commands, 597
  - notation, *654*
  - PMX, *634, 635*, 636–638
- `\small`
  - (LilyPond), *663*
  - (chessfss), 671
- `small` option (skak), 675
- `\smallaltoclef` (MusiXTeX), 592
- `\smallbassclef` (MusiXTeX), 592
- `\smallboard`
  - (bg), *696*, **697**
  - (cchess), *690*
  - (skak), 675, *678*
- `smaller` option (beamer), 753
- `\smallgoban` (igo), 694
- `\smallmusicsize` (MusiXTeX), 596
- `\smalltrebleclef` (MusiXTeX), 592
- solvesudoku package, 710–712
- song title, 602
- `\sound` (beamer), 774
- sound, slides, 774
- source code representation, slides, *791*
- `SPACE` syntax (m-ch-en), 546
- Space syntax (M-Tx), *655*
- space, trimming, 28, *30*
- `Space:` syntax (M-Tx), **652**, *659*, 660
- `\spade`
  - (bridge), *700*, *702*
  - (tlgc), *699*
- `\spadesuit`, 698, *699*
- `\special`, 6–8, **9**, 15–17, 20, 22, 35, 583, 690, 797
  - (tpic), 583
  - (xcolor), 719

- special color spaces, 715
- spectrum, displaying, *729*
- \sPED (MusiXTEX), 592
- \spind (circ), *580*
- \spinu (circ), *580*
- spline (pic), 17, *19*
- \spring (circ), *580*
- SPSS program, 21
- sqrt (pic), 19
- \squ (MusiXTEX), 592
- \square
  - (Slunits), *516*
  - (aliphat), *532*
- square brackets ([ ])
  - chord symbols (musical), *608*
- \squared (Slunits), *516*
- \squaremetrepersquaresecondnp (Slunits), *516*
- squeeze key (beamer), 759
- SR syntax (m-ch-en), 544
- \SS (chemsym), 517
- ss syntax (PMX), *624*, **625**
- ssc syntax (PMX), 625
- \ST (circ), *578*
- staccato marks (musical), *607*
- staccato ornaments (musical), 630
- \Staff (LilyPond), *665*
- Start: syntax (M-Tx), 652
- \startchemical (m-ch-en), ***541, 542***, *543–546*
- \startextract (MusiXTEX), **594**, *596*
- \startpiece (MusiXTEX), **594**, *599*
- state names, symbols for, 513
- staves (musical)
  - accidentals, 622, *624*, 628
  - arpeggio, *629*
  - basic duration, 622
  - beams, 631, *632, 633*
  - beams for xtuplets, *627, 628*
  - chords, *628*, 629
  - clef changes, *639*
  - defining, 652
  - definition, 617
  - dotted notes, 622
  - doubly dotted notes, 622
  - down fermata ornaments, 630
  - duration of notes, 622
  - dynamical marks, *638*
  - grace notes, 629, *630*
  - grace notes, in xtuplets, 627
  - height, 620
  - horizontal displacement, *624*
  - note parameters, *624, 625*
  - notes, 622, *623, 624*
  - number of, 619
  - octaves, *623*
  - on staves, 622–624

- staves (musical) *(cont.)*
  - ornaments, *630*, 631
  - parameters, 623, *624, 625*
  - pitch, 622
  - pointed rhythms, *624*
  - rests, 625, *626*
  - slurs, *634–638*
  - staccato ornaments, 630
  - stems, 623, 624
  - tenuto ornaments, 630
  - ties, 634, 635, *637*
  - xtuplets, 626, *627, 628*
- \stemDown (LilyPond), 663
- \stemNeutral (LilyPond), 663
- \stemNeutraltiny (LilyPond), *663*
- stems (musical), 623, 624
- \stemUp (LilyPond), *663*
- step key (beamer), *795*
- step syntax (xcolor), 734, 736
- stereochemical compounds, 530–532
- stereochemistry effects, 538
- \steroid (carom), 524, *526*
- steroid derivatives, *525, 526*
- \steroidchain (carom), 524
- stillcovered key (beamer), 768
- \STINV (circ), *578*
- \stopchemical (m-ch-en), ***541, 542***, *543–546*
- \storegame (skak), 679
- \structure (beamer), **788**, *789*
- structure syntax (beamer), *789*
- structured drawing, 20
- structures, chemical
  - atoms, aligning with bonds, *546*
  - basic commands for, *541, 542*
  - bonds
    - aligning atoms or molecules, *546*
    - chemical, *542*
    - description, 543
    - identifiers, 544
  - combinations, *544, 545*
  - combining, *534*
  - complex, *534, 535*
  - libraries of, 543
  - molecules, aligning with bonds, *546*
  - moving, *544, 545*
  - positioning, *544, 545*
  - reaction equations, *545*
  - rotating, *544, 545*
  - substructures, *543*
- Style: syntax (M-Tx), *651*, **652**
- \styleA (skak), *679*
- styleA option (skak), 679
- \styleB (skak), *679*
- styleB option (skak), 679
- \styleC (skak), *679*

styleC option (skak), 679
styles
    arrows (pict2e), *44*
    chess moves, *679*
    fills, 564, 565
    lines
        Feynman diagrams, **564, 565**, 566
        thickness, 566
    slide text, *789*
    slides, 754
    units typeset, 515
    vertices, 564, 565
SUB syntax (m-ch-en), **544**, 545
\subsection (beamer), 779
subsectionstyle key (beamer), 783
\substfont (xymtexps), *540*
\substfontsize (xymtexps), *540*
\substitutecolormodel (xcolor), 730
substitution derivation, *539*
\subtitle (beamer), *761*
subtractive color space, 715
sud.out file (solvesudoku), *711*
Sudoku, *709–711*, 712
\sudoku
    (createsudoku), *711*
    (printsudoku), *710*
    (solvesudoku), *711*
sudoku env. (sudoku), 710
sudoku package, 709, 710
sudoku-block env. (sudoku), ***709***, 710
\sudokuformat (sudoku), *709*, **710**
\sudokusize rigid length (sudoku), *709*, **710**
\sudokusolve
    (createsudoku), *711*
    (solvesudoku), 711
SVG language, 12, 13
SVG (Scalable Vector Graphics), 12, 13
svgnames option (xcolor), 721
svgnames* option (xcolor), 721
\symbishop (chessfss), 671
\symbol, *691*
symbols
    chemical diagrams, 512, 517, *518*
    electronics diagrams
        connections, 579
        electronic box, *578*
        gate, *578*
        integrated circuits, *579*
        state names, 513
        table of, *577*
        trigger, *578*
        wave names, 513
    mathematical functions, 512
    musical
        (...), slur symbol, *607, 608*
symbols *(cont.)*
        - (hyphen), tie symbol, *607, 608*
        = (equal sign), natural symbol, *605*
        [] (square brackets), chord symbols, *608*
        ^(caret), sharp symbol, *605*
        ^^(carets), double flat symbol, *605*
        _ (underscore), flat symbol, *605*
        __ (underscores), double flat symbol, *605*
        accidentals, *605*
        bar symbols, *603*, 639
        definition, 617
        notes, *592*, 593, *594*
        order of, *608*
        repeat, *603*
    Newtonian mechanics, *580*
    numbers, 512
    optics diagrams, *580*
    scientific texts, 512
    units, 512
    wave names, 513
symbols argument, 573, *575*
\symking (chessfss), 671
\symknight (chessfss), *671*
\sympawn (chessfss), *671*
\symqueen (chessfss), 671
\symrook (chessfss), 671
Systems syntax (M-Tx), *655*
systems (musical)
    definition, 617
    indentation, 620
    number of, 620
Systems: syntax (M-Tx), 652

T

T key (beamer), 781
t key (beamer), **759**, 781
t option (beamer), 753
T: syntax (abc), *601*, **602, 603**, *606*, ***608***
tabbing env., 688, 701
table env. (beamer), 780
table option (xcolor), 721, 737
table of contents, slides, 782
\tableofcontents (beamer), 752, *782*, **783**
tables, color
    cells, *741*
    columns, *738*, *747*
    entire table, *743*
    gaps between lines, 742
    gradients, *747, 748*
    headings, *748*
    highlighting elements, *745*, *749, 750*
    light text on dark background, *744*
    lines (rules)
        adding, *748*
        inside the table, *749*

- tables, color *(cont.)*
    - partial, 751
    - selected, *750*
    - whole table, *741*
    - width, *751*
  - rows
    - alternate, *739, 740*
    - selected, *746*
  - slides, 780
  - text, *745*, *748*
  - titles, *748*
- `tabular` env., 8, *39*, 702, 737, 741
  - (texmate), 680
- `tabular*` env. (colortbl), 737
- `\takecube` (bg), *698*
- TB syntax (m-ch-en), 544
- `\tb` (MusiXTeX), *599*
- `\tbl` (MusiXTeX), *596*, **597**
- `\tbu` (MusiXTeX), *596*, **597**
- `Tc` syntax (PMX), *641*
- `tcidvi` option
  - (graphics/graphicx), 24
  - (xcolor), 721
- templates, slides, 754
- tempo (musical), 602
- `\temporal` (beamer), *768*
- `tenor` syntax (LilyPond), 661
- tenuto ornaments (musical), 630
- `\tera` (SIunits), 515
- `\tesla` (SIunits), 514
- `\tetrahedral` (aliphat), ***532***, *535*, *540*
- tetrahedral compounds, *532*, 533
- tetrahedron carbon configurations, *533*
- tetraline derivatives, *525*
- `\tetralineh` (carom), 524, ***525***
- `\tetralinev` (carom), 524, ***525***
- `\tetralinevb` (carom), *525*
- `\tetralinevt` (carom), *525*
- `\tetramethylene` (methylen), *538*
- `\tetramethylenei` (methylen), *538*
- `\tetrastereo` (aliphat), *533*
- `\TeVovercsq` (hepunits), *516*
- `.tex` file extension (PMX), 621
- tex program, 618, 637
- TeX file archives, 810, *see also* CTAN
- TeX files, obtaining
  - web access, 810, 811, *812, 813*, 814
- TeX, interfaces
  - generating graphics, 8, 9
  - graphic hooks
    - `\special` commands, 9
    - built-in commands, 8
    - fonts, 8
  - graphics integration
    - `\special` commands, 6, 7
- TeX, interfaces *(cont.)*
    - fonts, 7, 8
    - half-tones, 7, 8
  - manipulating graphics, 8
  - overview, 6
- TeX-based drawing languages, 13–17
- texdoc program, 815, 816
- texdoctk program, 815–817
- `texmate` env. (texmate), 680
- texmate package, 668, 669, 673, 679, **680–687**
- `texshade` env. (texshade), ***548***, *549, 550*
- texshade package, **547–550**, 552
- `\Text` (axodraw), *559–561*
- text
  - blocks, *641*
  - colored, inside a box, 725
  - in documents, 725
  - resizing, *38, 39*
  - scaling, *37*
  - slides
    - alternative, *769*
    - boxed, *775, 776*
    - colored, *775, 776*
    - framing, *775, 776*
    - holding static, *770*
    - preformatted, *790, 791*
    - styles, *789*
    - verbatim, *790, 791*
  - tables
    - color, *745*, *748*
    - light on dark background, *744*
- `\textbf` (beamer), 788, *789*
- `\textbishop` (chessfss), **671**, *672*
- `\textcolor` (xcolor), 720, ***722***, *723, 724*
- `\textit` (beamer), 788, *789*
- `\textking` (chessfss), 671
- `\textknight` (chessfss), **671**, *672*
- `\textmove` (bg), *698*
- `textopo` env. (textopo), **551**, *552, 553*
- textopo package, 547, **551–555**
- `\textpawn` (chessfss), 671
- `\textpiece` (cchess), **688**, *689*
- `\textqueen` (chessfss), 671
- `\textrm` (beamer), 788, *789*
- `\textrook` (chessfss), 671
- `\textsf` (beamer), 788, *789*
- `\textsl` (beamer), 788, *789*
- `textstyle` option (SIunits), 515
- `texttopo` env. (textopo), *551*
- Textures program, 11, 17, 24
- `textures` option
  - (graphics/graphicx), 24
  - (xcolor), 721
- `\textwidth` rigid length (beamer), 777
- `.tfm` file extension, 666

\tgqu (MusiXTEX), 592
thebibliography env. (beamer), *782*
themes, slides, *754–757*
then (pic), *19*
theorem env. (beamer), **753**, *769*
\thicklines
(curve2e), *49*
(pict2e), 45
thickqspace option (Slunits), 515
thickspace option (Slunits), 515
\thinlines
(curve2e), *48–50*
(pict2e), 45
thinqspace option (Slunits), 515
thinspace option (Slunits), 515
\Threat (texmate), ***681***, *682*
\threat (texmate), ***681***, *682*
THREE syntax (m-ch-en), *542*
three-color harmonics, 718
three-color theory, 714
three-member carbon cycles, *528*
\threefuseh (fusering), 537
\threefusehi (fusering), 537
\threefusev (fusering), 537
\threefusevi (fusering), 537
\threehetero (hetarom), ***523***, *528*
\threeheteroh (hetarom), *529*
\threeheterohi (hetarom), *529*
\threeheterov (hetarom), *529*
\threeheterovi (hetarom), *529*
tHsb syntax (xcolor), 728, 729
\THz (hepunits), *516*
Ti syntax (PMX), *641*
ties (musical), *607*, *637*
PMX, 634, 635
tightpage option (pst-pdf), 800
tikz package, 5
\til (timing), 573
tilde (~), grace notes, *607*
\timadjust (timing), 576
\time (LilyPond), ***663***, *664, 665*
\times (LilyPond), *664*
\timescalefactor (timing), 576
timing env. (timing), **573**, *574*
timing package, 572–576
timing diagrams
annotation, 573
arrows, 575
customizing, 576
fonts, specifying, 573
labels, 573
overview, 572–576
separation between lines, 576
signal lines, 573
symbols argument, 573, *575*

timing diagrams *(cont.)*
timing values, 573
vertical line adjustment, 576
vertical lines, 576
timing values, 573
\timingcounter (timing), 573
\tin (timing), **573**, *574*
tinting, *731*
\TinveV (hepunits), *516*
\tiny (LilyPond), *663*
tiny option (skak), 675
\tinyboard
(skak), **675**, *677*
(texmate), *686*
\title (beamer), *754*, *757*, *761*
title blocks (musical), *641*
title pages, slides, *761*
Title: syntax (M-Tx), 652
\titlepage (beamer), *761*
titles
chess, 683
slides, 759
tables, *748*
\tnote (timing), **573**, *574*
to (pic), *19*
\toD (texmate), *685*
\toD* (texmate), **685**, *686*
\togglenumbers (bg), 697
top key (beamer), *795*
\topdiagramnames (texmate), 686
\totalheight (graphics/graphicx), *38*
totalheight key (graphicx), **29**, *32*
totalwidth key (beamer), 781
tpic program, 583, 584
trans option (beamer), 753
\transblindshorizontal (beamer), 774
\transblindsvertical (beamer), 774
\transboxin (beamer), 774
\transboxout (beamer), 774
\transdissolve (beamer), 774, *775*
\transduration (beamer), 774
transfig program, 13
\transglitter (beamer), 774
transitions, slides, *774, 775*
transparency, slides, *768*
transparent key (beamer), 767
\transsplithorizontalin (beamer), 774
\transsplithorizontalout (beamer), 774
\transsplitverticalin (beamer), 774
\transsplitverticalout (beamer), 774
\transwipe (beamer), 774
\treble (MusiXTEX), *596*
treble syntax (LilyPond), **661**, *664*
\trebleclef (MusiXTEX), 592
trees, *see* META *and* PSTricks *index*

\triazinev (hetarom), 530
\triazinevi (hetarom), 530
tricyclic carbocycles, *525*
trigger symbols, *578*
trigonal units, 532, *533*
\Trille (MusiXTEX), 592
\trille (MusiXTEX), 592
trim key (graphicx), **28**, 29, *30*
\trimethylene (methylen), *538*
\trimethylenei (methylen), *538*
trimming space, 28, *30*
triplets (musical), *605*
LilyPond, *664*
troff program, 17
TrueTeX program, 24
truetex option
(graphics/graphicx), 24
(xcolor), 721
\tslur (MusiXTEX), *596*, **597**, *599*
Tt syntax (PMX), *641*
\ttfamily (beamer), 764
TUG home page, 810, *811*
\turn (MusiXTEX), 592
turn env. (rotating), *42*
turtle graphics, *see META index*
two-color harmonics, 718
type key (graphicx), **29**, 35
typesetting, overview, 2, 3
typographic conventions, this book, xxix, xxxi

U

\U (circ), *577*
u syntax
(PMX), **625**, 631, *633, 634*, *636*
(abc), *607*
U: syntax (M-Tx), **657**, *658*
ucs option (beamer), 753
ucs package, 753
UML diagrams, *see META and* PSTricks *index*
\uncover (beamer), **767**, ***768***, *785*
uncoverenv env. (beamer), 770
\underline, 672
underscore (_), flat symbol (musical), *605*
underscores (__), double flat symbol (musical), *605*
\unit
(SIunits), *515, 516*
(hepunits), *516*
\unitlength rigid length
(curve2e), *48*
(pict2e), 45, 46
(timing), 573
units
base, 514
combining, *516*
derived, 514
units *(cont.)*
high-energy physics, *516*
prefixes, 514
SI (International System of Units), 512–516
spacing between, *515*
symbols for, 512
typeset style, 515
units key (graphicx), **40**, 42
units package, 513
unitsdef package, 513
\upbow (MusiXTEX), 592
upper key (beamer), *776*, **778**
\uppz (MusiXTEX), 592
\Uptext (MusiXTEX), *599*
\uptrio (MusiXTEX), 592
\upz (MusiXTEX), 592
\upzst (MusiXTEX), 592
\usebeamercolor (beamer), *794*
\usebeamerfont (beamer), *777*, *794*
\usebeamertemplate (beamer), *777*
\usecolortheme (beamer), *758*
\usefonttheme (beamer), **758**, *760*
\usegoban (igo), **694**, 695
\useinnertheme (beamer), 758
usenames option (xcolor), 721
\useoutertheme (beamer), *758*, *773*
\useouthertheme (beamer), 758
\usepackage, xxxii
(beamer), 754, 758
usepdftitle option (beamer), 753
\usesymfig (chessfss), *672*
\usetextfig (chessfss), *672*
\usetheme (beamer), ***758***, *760*
\usf (MusiXTEX), 592
\usfz (MusiXTEX), 592
\usk (SIunits), *515, 516*
\ust (MusiXTEX), 592
utf8 option
(beamer), 753
(inputenc), 753
\Utrigonal (aliphat), *533*
\utrigonal (aliphat), *533*
\Uvar (circ), *577*

V

V syntax (PMX), *640*
\V (circ), *577*
v syntax (abc), *607*
V: syntax (abc), *610*
\var (texmate), ***682***, *683*
\var* (texmate), 682
\variation (skak), **677**, *678, 679*
variations env. (texmate), ***682***, *683*
variations* env. (texmate), 682
\VariationsEnvironment (texmate), *683*

- \VECTOR (curve2e), 47, 50
- \Vector (curve2e), 47, *48*
- \vector, *43*
  - (curve2e), 47, *48*
  - (pict2e), *43*, *44*, *46*
- \VectorARC (curve2e), *50*
- \VectorArc (curve2e), *50*
- \verb
  - rotating output, *42*
  - (beamer), 790
- verbatim env., 13
  - (beamer), 790
- \Vertex (axodraw), 559, *560*
- vertex dots (Feynman diagrams), *560*
- vertex mode (Feynman diagrams)
  - algorithmic layout, 563–569
  - blobs, 566
  - coloring diagrams, 567
  - complex vertices, 567
  - definition, 563
  - external vertices, placing, *564*
  - fill styles, 564, 565
  - freezing a diagram, 567
  - internal vertices, 566
  - labels, 567, *568*, 569
  - line styles, 564, 565
  - line thickness, 566
  - line-drawing keywords, 566
  - polygon keywords, 567, 568
  - vertex styles, 564, 565
  - vertex-drawing keywords, 567
  - vertices, as dots, 566
  - vertices, connecting, 565
- vertex styles (Feynman diagrams), 564, 565
- vertex-drawing keywords (Feynman diagrams), 567
- \vertexlabel (feyn), *557*
- vertical shading syntax (beamer), *795*
- vertices (Feynman diagrams), 565, 566
- \vflipgoban (igo), 695
- video, slides, 774
- viewport key (graphicx), **28**, 29, *30*
- viewports, 28, *30*
- violet syntax (xcolor), 726
- \visible (beamer), **768**, *791*
- visibleenv env. (beamer), 770
- vlabellift key (chessboard), *669*
- \VLens (circ), *580*
- \vline (colortbl), 741
- vmode key (beamer), 777, *794*
- voice (musical)
  - definition, 617
  - labels, 653
  - spacing after, 653
- \volt (SIunits), 514, *515*
- voltas (musical), *640*
- VTeX program, 11, 24, 797
- vtex option
  - (graphics/graphicx), 24
  - (pict2e), 43
  - (xcolor), 721
- \vtopin (circ), ***579***, *581*
- Vx syntax (PMX), *640*

W

- W syntax (PMX), *630*
- W. syntax (PMX), 643
- w.eps file (tlgc), 26
- W: syntax (abc), *608*
- w: syntax (abc), *611*
- \wall (circ), *580*
- watermarks, *see* PSTricks *index*
- \watt (SIunits), 514, *516*
- \wattpersquaremetresteradiannp (SIunits), *516*
- wave syntax (xcolor), 728, 729
- wave names, symbols for, 513
- \wbetter (skak), *678*
- wd key (beamer), *776*, 777, *794*
- \wdecisive (texmate), *682*
- WebCGM, 13
- \weber (SIunits), 514
- \wedgehashedwedge (xymtexps), ***538***, *539*
- \welo (texmate), 683
- wget program, 814
- \wh (MusiXTEX), 592, **593**, *594*
- \white (igo), **691**, *692–695*
- white syntax (xcolor), **722**, *723*, **726**
- \whitebar (bg), 697
- \whitecube (bg), 697
- \whitename (texmate), *683*
- \whiteonmove (bg), *696*, **697**, *698*
- \whitepoint (bg), *696*
- \whitestone (igo), *695*
- \whp (MusiXTEX), 592
- \width (graphics/graphicx), 38
- width (pic), *19*
- width key
  - (beamer), **778**, *792*
  - (graphicx), **28**, 29, *31–33*
- \wire (circ), 579
- \withidea (texmate), *681*
- \wmove (skak), 679
- \wname (texmate), **685**, 686
- \Word (cwpuzzle), *707*
- words (musical), 617
- \writegame (solvesudoku), *711*
- \writepuzzle (printsudoku), 710
- \wwire (circ), 579

X

- X syntax (PMX), **632**, *633*, **643**

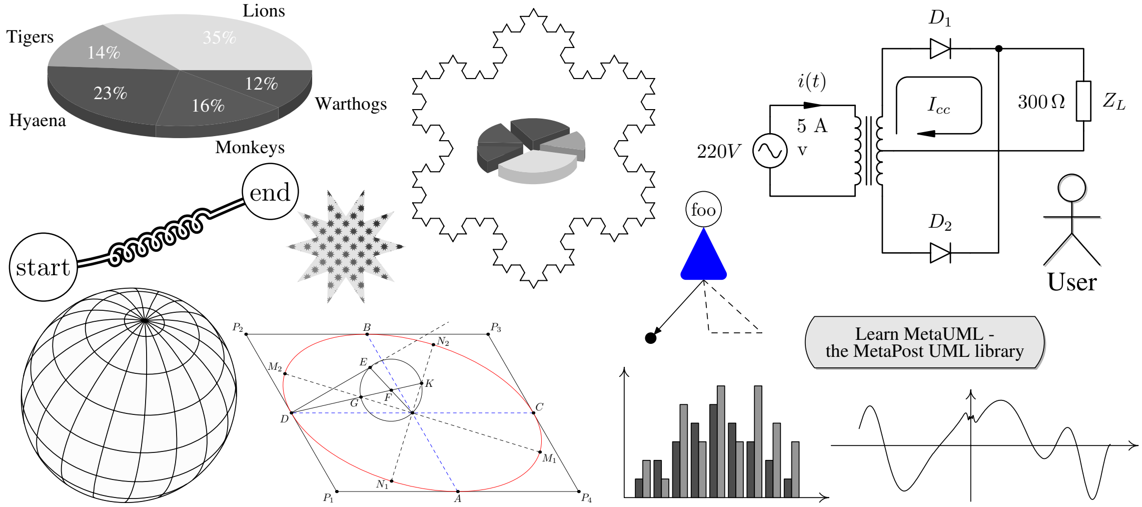
x key (graphicx), **40**, 41
x syntax (PMX), **625**, *627, 628*, ***630***
x11names option (xcolor), 721
X: syntax
  (PMX), 643
  (abc), *601*, **602, 603**, ***608***
XCircuit program, 576, 586
xcolor option (beamer), 753
xcolor package, 7, 713, **719–737**, 740, 747, 753
.xcp file extension (xcolor), 721
xdvi option (pict2e), 43
xdvi program, 24
xetex option (xcolor), 720, **721**
xetex program, 798, 803
xfig program, 1, 6, 13, 21, 586
\xglobal (xcolor), 726
xiangqi chess, 687, *688–690*
\XNOR (circ), *578*
\XOR (circ), *578*
xpdf program, 12, 804
xq package, 688
\xqu (MusiXTEX), 592
xtuplets (musical), 626, *627, 628*
XΥMTEX package, 520–540
xymtex package, 520, 537
xymtexps package, 537
xymtx-ps package, 537
Xy-pic package, xxvi, xxviii, 5, 9, 16, *see also Xy-pic index*

Y

y key (graphicx), **40**, 41
yellow syntax (xcolor), **722**, *724*, **726**
\yocto (SIunits), 515
\yotta (SIunits), 515
Young-Helmholtz Law, 714
\yqu (MusiXTEX), 592

Z

Z syntax (m-ch-en), 544
z syntax
  (PMX), *628*
  (abc), ***603***, *604*, ***607***
\z... (MusiXTEX), 594
Z0 syntax (m-ch-en), **544**, 546
\zbreve (MusiXTEX), 592
\zcharnote (MusiXTEX), *599*
\ZD (circ), *577*
\zepto (SIunits), 515
\zetta (SIunits), 515
\zh (MusiXTEX), *594*
\ZigZag (axodraw), *559, 560*
zigzag lines (Feynman diagrams), *559, 560*
zlib program, 799
\zlonga (MusiXTEX), 592
\zmaxima (MusiXTEX), 592
\znotes (MusiXTEX), 595
\zq (MusiXTEX), *596*
\zqb (MusiXTEX), *596*
\zw (MusiXTEX), *594*
\zwq (MusiXTEX), 592



METAFONT and METAPOST

Symbols

\((pst-pdf), 800
\) (pst-pdf), 800
++ syntax (META), 52
+-+ syntax (META), 52
- syntax (META), 54
.. syntax (META), 54
_T (METAOBJ), *114*
3-D extensions
  animations, 209
  cubes, *210*
  curve intersections, computing, 211
  globes, *209*
  hexagonal meshes, *210*
  labels in space, *211*
  METAPOST files, creating, 209
  overview, 207
  packages for, 208–212
  perspective projection, 208
  physics diagrams, *209*
  projected segments, *211*
  requirements, 207
3DLDF program, 211, 212
3d METAPOST package, 68, 207–209
3dgeom METAPOST package, 208

A

abs (META), *56*
Acrobat Distiller program, 797, 798
active option (pst-pdf), 800
activities, UML
  beginning, *187*
  constructing, *187*
  ending, *187*
Activity (metaUML), *187*
Actor (metaUML), *187*
actors, *187*
addto (META), ***143***, *146*, *150*, 176
\addtocounter (mfpic), *136*
Adobe Reader program, 804, 817
Adobe Illustrator program, 65, 137, 138
affine transforms
  mfpic, 136
  META language, 53
align key (METAOBJ), *101–103*
alignment (METAOBJ)
  boxes
    horizontal, *101*
    horizontal separation, *102*
    mixed objects, *102, 103*
    vertical, *101*, *103*
    within frames, *104*

- alignment (METAOBJ) *(cont.)*
  - trees, *107, 108*
- analytical curves (mfpic), 133
- angle (META), **53**, ***142***, *191*, *205*
- angle key (METAOBJ), 86
- angle dimensions (mfpic), 127
- angleA key (METAOBJ), 85, *87–92*, *94*, *177*
- angleB key (METAOBJ), 85, 87, *88–91*, 92, *94*
- animation
  - 3d package, 208
  - m3d package, 209
  - METAPOST techniques, 156, 157
- annotations
  - mfpic, 134
  - drawings, 134
  - pictures, *61–64*, 65
- \arc (mfpic), *127*, **128**
- arcangle key (METAOBJ), 86
- arcangleA key (METAOBJ), 85, *88*, *93*
- arcangleB key (METAOBJ), 85, *88*, *93*
- arclength (METAPOST), ***142***, *191*
- arcs
  - mfpic, 128
  - METAOBJ, *88*
- arctime (METAPOST), *142*
- arm key (METAOBJ), 86
- armA key (METAOBJ), 85, *89–91*, *177*
- armB key (METAOBJ), 85, *89–91*
- \arrow (mfpic), *127*, **132**, *135*
- arrows
  - mfpic
    - drawing, 132
    - length, 132
    - shape, 132
  - cmarrows, *188*
  - connections (METAOBJ), *87*
- arrows key (METAOBJ), 84, 85, *87*, *94*, *118*
- METAPOST geometry, *195*
- associations, UML, *186*
- augment (graph), **161**, ***162***, *164*, 167, *169*
- AutoCAD program, 137
- autogrid (graph), **158**, ***159***, *163*, *165–167*
- \axes (mfpic), *123, 124*, *127*, **128**, *130, 131*, 132
- axes, drawing (mfpic), 128
- \axis (mfpic), 128
- \axisheadlen rigid length (mfpic), 128, 132
- \axismarks (mfpic), 129

B

- babel package, 124
- bar package, 162
- bar charts
  - mfpic, *130*
  - graph, *162, 163*, 164, *166*
- \barchart (mfpic), ***130***, 131
- base (exteps), *156*
- basic objects, 82, 83
- battery (makecirc), *197*, 199
- bbox (METAPOST), ***62***, *163*, *165*
- bcircle (metafun), 74
- \bclosed (mfpic), *127*, **132**
- Begin (metaUML), ***187***, *188*
- beginchar (META), **68**, 72
- begineps (exteps), *156*
- beginfig (METAPOST), **72**, 73, 80, 156
- begingraph (graph), ***157, 158***, *169*
- Bézier curves (METAOBJ), *87, 88*
- Bézier paths (mfpic), 128
- bibtex program, 801, 806
- Bigbrace (cmarrows), *189*
- bigbrace (cmarrows), *189*
- Biggbrace (cmarrows), *189*
- biggbrace (cmarrows), *189*
- bitmap (.gf) output files, 69, *70*
- black (METAPOST), 60
- block drawing, 177
- blockdraw METAPOST package, 177
- blue (METAPOST), 60
- bluepart (METAPOST), *150*
- blurred effects, *152*
- \bmarks (mfpic), 129
- Bond graphs, 177
- boolean (META), 53, *56*
- border key (METAOBJ), 85
- bordercolor key (METAOBJ), 85
- bot syntax (METAPOST), 61
- bounded (METAPOST), 67, *150*
- bounding box (mfpic), 124
- BoundingBox (PostScript), 72
- Box (METAOBJ class), 95, 96, 99
- box-line diagrams, *178–180*, 181
- boxdepth key (METAOBJ), 85, 92
- boxes
  - alignment (METAOBJ)
    - centering, *103*
    - horizontal, *101*
    - horizontal separation, *102*
    - mixed objects, *102, 103*
    - vertical, *101*, *103*
    - within frames, *104*
  - empty, *82, 83*
- boxes METAPOST package, 57, 75, 76, 79–81, 177
- boxheight key (METAOBJ), 85, 92
- boxit (boxes), **76**, ***77***, *78*
- boxjoin (boxes), **76**, ***77***, *78, 79*
- boxsize key (METAOBJ), 85, 92, *93*
- bpath (METAPOST), ***77, 78***, *79*
- btex (METAPOST), ***61–63***, *95*, *157, 158*, 159, *162*, *164*
- \btwnfcn (mfpic), 133

- `buildcycle` (METAPOST), *165*

C

- `capacitor` (makecirc), ***196, 197***, *198*, ***199***, *201*
- capacitors, *196*
- captions, centering, 124, 134
- card boxes, *180*
- `\cbclosed` (mfpic), 132
- Celtic artwork, 148
- centering (mfpic)
  - captions, 124, 134
  - ellipses, 128
  - symbols, 129
- `centerto` (makecirc), ***198***, *199, 200*, *202*
- `centreof` (makecirc), ***196***, ***198***, *199*, *202*
- `\chartbar` (mfpic), 130
- `Circle` (METAOBJ class), 114
- `\circle` (mfpic), *127*, 128
- `circleit` (boxes), **76**, *77*, *78, 79*
- circles
  - connections (METAOBJ), *92*
  - diagrams, *179*
  - diameter (mfpic), 132
  - drawing (mfpic), 128
  - filled and centered, 129
  - filling (mfpic), 132
  - nine points circle of a triangle, *190*
  - wedge of (mfpic), 129
- `circmargin` (boxes), 76, *79*
- `circmargin` key (METAOBJ), *98*, *100*
- circular
  - containers (METAOBJ), *98–100*
  - gradients, *143, 144*
- `Class` (metaUML), **181**, ***182***, *183–186*
- class
  - relations (UML diagrams), *184*
  - templates (UML diagrams), *183*
- `classStereotypes` (metaUML), *183*
- `ClassTemplate` (metaUML), *183*
- clearing (mfpic)
  - closed objects, 133
  - symbols, 124
- `clearObj` (METAOBJ), 81
- `\clearsymbols` (mfpic), 124
- `clink` (metaUML), *186*
- `clip` (METAPOST), ***63***, ***143***, **145**, *148*, *150*, *206*
- `\clipmfpic` (mfpic), 124
- `clipped` (METAPOST), 67, *150*
- clipping
  - figures (mfpic), 124
  - tools, *148*
- `clipping` (exteps), *156*
- closed
  - objects
    - clearing, 133
    - filling, 133
  - polygons (mfpic), 129
- `closefrm` (METAPOST), 67
- `\closegraphsfile` (mfpic), 125
- closing objects
  - mfpic, 132
  - META language, *54*
- cmarrows METAPOST package, 188
- CMYK color, 75
- `coilarm` key (METAOBJ), 86
- `coilarmA` key (METAOBJ), 85, 94
- `coilarmB` key (METAOBJ), 85, 94
- `coilaspect` key (METAOBJ), 85, 94
- `coilheight` key (METAOBJ), 85, 94
- `coilinc` key (METAOBJ), 85, 94
- coils, connections (METAOBJ), *94*
- `coilwidth` key (METAOBJ), 85, *94*
- color
  - mfpic, 127
  - CMYK, 75
  - drawings, 127
  - graying, 75
  - labels, 120
  - METAFONT vs. METAPOST, 60
  - transparency, 75
- `color` (METAPOST), **60**, ***64***, *79*, **209**
- commands (mfpic), *127*
- comments (mfpic), 134
- Comprehensive TEX Archive Network, *see* CTAN
- `connect` env. (mfpic), **126**, 132, 133
- connections (METAOBJ)
  - arcs, *88*
  - arrow style, *87*
  - behind objects, *90*
  - Bézier curves, *87, 88*
  - circles, *92*
  - coils, *94*
  - curved boxes, *93*
  - double straight line, *87*
  - inside boxes, *92, 93*
  - labels for, *95*
  - line starting point, *87*
  - line style, *86*
  - line thickness, *86*
  - looping lines, *91, 92*
  - multi-segment lines, *89–91*
  - overview, 84–86
  - rounded corners, *93*
  - straight lines, *86, 87*
  - zigzags, *94*
- connectors, diagrams, *180*
- `Container` (METAOBJ class), 104

- containers (METAOBJ)
  - circular, *98–100*
  - description, 95
  - double-walled
    - box, *99, 100*
    - circle, *100*
    - ellipsis, *100*
  - elliptical, *98–100*
  - margins, *96, 97*
  - oval boxes, *96*
  - polygons, *97*
  - rounded corners, *96*
  - simple box, *95*
  - square box, *95*
- `contour` (META), ***143***, *150*
- control points, 53
- convert program, 806
- coordinate dimensions (mfpic), 127
- coordinate system, specifying (mfpic), 126
- `coords` env. (mfpic), 136
- Corel Draw program, 137, 138
- `cosd` (META), 53, *195*
- CTAN (Comprehensive TeX Archive Network)
  - archived files, finding and transferring, 813
  - description, 810
  - files, from the command line, *814*
  - TeX file catalogue, 811
  - web access, 810, 811, *812, 813*, 814
- `ctext` (makecirc), ***200***, *201*
- cubes, *210*
- `curl` (META), 54, *55*
- `current` (makecirc), ***197***, **199**, *201, 202*
- `currentpen` (META), *146*
- `currentpicture` (META), ***62***, **65**, ***66***, *155, 156*, 176
- `\curve` (mfpic), *127*, **128**, *136*
- curved box connections (METAOBJ), *93*
- curves
  - function drawing, *168, 169*
  - intersections, computing, 211
  - META language
    - 3-D, 57, *58*
    - controlling, *55*
    - drawing, *54*
    - `path` data, 53
  - polar coordinates, *169*
  - through points (mfpic), 128
- `cutafter` (METAPOST), ***77***, *78, 79*
- `cutbefore` (METAPOST), ***77***, *78, 79*
- `cycle` (META), ***54***, *56*, ***161***, *162*, *164*
- `\cyclic` (mfpic), 128

D

- `Dalign` key (METAOBJ), *107*, *110, 111*, *114*
- `\darkershade` (mfpic), 132
- `dashed` (METAPOST), ***79***, ***86***, *88*, *157*, 158, *162*
- dashed lines (mfpic), 133
- `dashes` (expressg), *180*
- dashes (mfpic)
  - gap between, 131, 133
  - length, 132
  - length of, 131
  - lines, 133
  - spacing, 132
- `\dashlen` rigid length (mfpic), 131–133
- `\dashlineset` (mfpic), 132
- `\dashspace` rigid length (mfpic), 131–133
- data types, META language, 53
- `DBox` (METAOBJ class), 99
- debugging figures (mfpic), 125
- `def` (META), 57
- `defaultdx` (boxes), 76
- `defaultdy` (boxes), 76
- `defaultfont` (METAPOST), ***61***, *79*, *163*, *165*, **174**
- `defaultscale` (METAPOST), **61**, ***62***, *78, 79*, *163*, *165–167*
- `DefinePattern` (piechartMP), 175, *176*
- diagrams
  - block drawing, 177
  - Bond graphs, 177
  - box-line, *178–180*, 181
  - card boxes, *180*
  - circles, *179*
  - connectors, *180*
  - diamond boxes, *180*
  - embedding in LaTeX, 120, 121, *122*
  - flow charts, *177*, *181*
  - graphs, 176
  - index boxes, *180*
  - ovals, *179*
  - relations, *180*
  - rounded boxes, *179*
  - slanted rectangles, *179*
- diamond-shaped boxes, *180*
- `diode` (makecirc), ***197***, **199**, *202*
- `dir` (META), **54**, ***55***, *77–79*
- `direction` (META), *142*, *205*
- disadvantages, 139
- `displaymath` env. (pst-pdf), 800
- `displaymath` option (pst-pdf), 800
- distance dimensions (mfpic), 127
- `\doaxis` (mfpic), 128
- documentation, *see also* online resources
  - command-line interface, *815*
  - panel interface, *816*
  - search by name, *815*
  - search by product, *816*
  - `texdoc`, *815*
  - `texdock`, *816*
- `dotlabel` (METAPOST), *61*
- `dotlabels` (METAPOST), *62*
- `\dotlineset` (mfpic), 132

- dots (shading), gap between (mfpic), 131, 133, *134*
- \dotted (mfpic), *127*, **133**
- dotted lines (mfpic), 133
- double-walled containers (METAOBJ)
  - box, *99, 100*
  - circle, *100*
  - ellipsis, *100*
- doublearrow (cmarrows), *189*
- doubleline key (METAOBJ), 85, *87, 88*, *94*
- doublesep key (METAOBJ), 85
- dpi (dots per inch), 70
- draft option (pst-pdf), 800
- \draw (mfpic), **133**, *134*
- draw (META), ***54***, *55, 56*, 76, 84, 87, 158, 189
- draw_hatched_band (hatching), *150*
- draw_Obj (METAOBJ), ***114***, *118*
- drawarrow (METAPOST), ***77***, *78, 79*, 84, 87, 189
- drawBINARY (expressg), 178
- drawBOOLEAN (expressg), 178
- drawboxed (boxes), **76**, ***77***, *78*
- drawboxes (boxes), **76**, *77*
- drawcardbox (expressg), *180*
- drawcirclebox (expressg), 179, *181*
- \drawcolor (mfpic), 127
- drawCOMPLEX (expressg), 178
- drawdashA (expressg), *180*
- drawdashcircle (expressg), *179*
- drawdashellipse (expressg), *179*
- drawdashO (expressg), *180*
- drawdashOA (expressg), *180*
- drawdblarrow (METAPOST), 77
- drawdiamondbox (expressg), *180, 181*
- drawEXPRESSION (expressg), 178
- drawGENERIC (expressg), 178
- drawGEVENT (expressg), *179*
- drawindexbox (expressg), *180*
- drawing
  - animation, *156*, 157
  - blurred effects, *152*
  - boxes
    - commands for, 76
    - committing to the page, 76
    - joining, *77*
    - labeling connections, *78*, 79
    - relationships between, 76
  - Celtic artwork, 148
  - circles, 74
  - circular gradients, *143, 144*
  - clipping, *148*
  - diamonds, *74*
  - gradients, *143, 144*
  - grids, *147*, 148–150
  - hatching, *148–150*
  - lines
    - creating grids, *147*

drawing *(cont.)*
  - hiding, *145*
  - repeating, *147*
  - morphing, *152*
  - multipaths, 145
  - parallel gradients, *143, 144*
  - paths
    - interrupting, 145, *146*
    - multipaths, 145, *146*
  - patterns, 147–150
  - PostScript commands, *155, 156*
  - rounded corners, *75*
  - simplified paths, *75*
  - squares
    - creating grids, *147*
    - repeating, *147*
  - squeezing shapes, *74*
  - text along a curve, *142*
  - tilings, 147–150
  - transparency, *150, 151*
  - turtle graphics
    - classic style, *153*
    - turtle style, *153, 154*
- drawing (mfpic)
  - affine transforms, 136
  - analytical curves, 133
  - angle dimensions, 127
  - annotations, 134
  - arcs, 128
  - arrowheads
    - drawing, 132
    - length, 132
    - shape, 132
  - axes, 128
  - bar charts, *130*
  - basic commands, 128–130
  - Bézier paths, 128
  - bounding box, 124
  - centering
    - captions, 124, 134
    - ellipses, 128
    - symbols, 129
  - circles
    - diameter, 132
    - filling, 132
    - simple, 128
  - clearing
    - closed objects, 133
    - symbols, 124
  - clipping figures, 124
  - closed polygons, 129
  - closing open objects, 132
  - color, 127
  - commands, *127*
  - comments, 134

- drawing (mfpic) *(cont.)*
  - coordinate dimensions, 127
  - coordinate system, specifying, 126
  - curves through points, 128
  - dashed lines, 133
  - dashes
    - gap between, 131, 133
    - length, 132
    - length of, 131
    - spacing, 132
  - debugging figures, 125
  - distance dimensions, 127
  - dots (shading), gap between, 131, 133, *134*
  - dotted lines, 133
  - figure modifiers, 132, 133
  - filled centered circles, 129
  - filling closed objects, 133
  - functions, 133
  - global modifiers, 132
  - grids, 129
  - hash marks, length of, 131
  - hatching, line spacing, 131, 133, *134*
  - joining objects, 126
  - labels, 124, 134
  - line segments, 129
  - looping, *136*
  - METAFONT mode, *123*
  - METAPOST mode, *124*
  - modifiers, *127*
  - numbering pictures, 126
  - object outlines, 133
  - options, 124, 125
  - pen, setting width, 132
  - pie charts, *131*
  - plotting functions and parametric curves, 133, *135*
  - pretty printing, 137
  - primitives, 126
  - processing, 123
  - rectangles, 129
  - regular polygons, 129
  - repetitive, 134
  - reversing objects, 133
  - rotating objects, 133, *135*
  - shading, dot spacing, 131, 132, *134*
  - size, specifying, 126
  - spirals, *136*
  - symbolic names, 129
  - syntax, 125–127
  - unit length, basic, 132
  - wedge of a circle, 129
- `drawINTEGER` (expressg), *178*
- `drawLEVENT` (expressg), *179*, *181*
- `drawLOGICAL` (expressg), 178
- `drawnormalCA` (expressg), *180*
- `drawnormalCD` (expressg), 180
- `drawnormalD` (expressg), *180*
- `drawnormalDCA` (expressg), *180*
- `drawnormalF` (expressg), *180*
- `drawnormalOA` (expressg), *180*
- `drawnormalOD` (expressg), *180*
- `drawNUMBER` (expressg), 178
- `drawObj` (METAOBJ), ***81***, *82, 83*, *95*, *177*
- `drawObject` (metaUML), ***182***, *183*, *186–188*
- `drawObjects` (metaUML), ***183, 184***, *185–187*
- `drawoptions` (METAPOST), *148*
- `drawovalbox` (expressg), *179*, *181*
- `drawREAL` (expressg), 178
- `drawroundedbox` (expressg), *179*
- `drawSTRING` (expressg), 178
- `drawthickO` (expressg), *180*
- `drawunboxed` (boxes), **76**, *77*, *79*
- dual bar charts, *164*
- `duplicateObj` (METAOBJ), *117*
- `.dvi` file extension (META), 63
- dvipdfm program, 797, 798, 803
- dvipdfmx program, 797–799, 803, 804, 806
- dvips program, 62, 65, 797–801, 803–806
- dvitomp program, 63
- `dx` key (METAOBJ), *96*, *100*, *104*
- `dy` key (METAOBJ), *96*, *100*, *104*

E

- electrical circuits
  - capacitors, *196*
  - centering elements, *198*
  - centering text, *200–202*
  - command syntax, 199
  - element abbreviations, 198
  - element types, 199
  - elements of, 196–199
  - inductors, *196*
  - pin connections, 200
  - resistors, *196*
  - symbols, 196, 197, *198*
  - wiring type, *198*
- `\ellipse` (mfpic), **128**, *136*
- ellipses
  - centered, 128
  - in a parallelogram, *191*
- elliptical containers (METAOBJ), *98–100*
- emp env. (emp), 121
- emp package, 120, 121, 167
- `empcmds` env. (emp), 121
- `empdef` env. (emp), 121
- `empfile` env. (emp), 121
- `empgraph` env. (emp), 122
- `\empprelude` (emp), 122
- empty boxes (METAOBJ), *82, 83*
- `EmptyBox` (METAOBJ class), 82, 83, 95
- `\empuse` (emp), 121

End (metaUML), ***187***, *188*
end (META), 72
endchar (META), **68**, 72
endeps (exteps), *156*
endfig (METAPOST), **65**, **72**, 73, 80
endfor (META), 52, *55*
endgraph (graph), ***157, 158***, *169*
EntryPoint (metaUML), 188
EPS output files, *72*, 73
epsdrawdot (exteps), *156*
epstopdf program, 804, 806
eqnarray env. (pst-pdf), 800
equation env. (pst-pdf), 800
etex (METAPOST), ***61–63***, *95*, *157, 158*, 159, *162*, *164*
exitif (META), *56*, *204*
ExitPoint (metaUML), 188
METAPOST, 137, 138
expr (META), 57
expressg METAPOST package, 177, 178, 181, 182
extendObjLeft (METAOBJ), 108
extendObjRight (METAOBJ), *108*, 109
extensiblebrace (cmarrows), *189*
exteps METAPOST package, 155

F

fanlinearc key (METAOBJ), *114*
fanlinestyle key (METAOBJ), *114*
FAQs (Frequently Asked Questions), 809, *see also* online resources
\fcncurve (mfpic), 128
featpost METAPOST package, 207, 209
feynmf package, 120
feynmp package, 120
figure modifiers (mfpic), 132, 133
file input/output, 67, *68*
fill (META), ***56***, 76, **150**, *151*, 158
\fillcolor (mfpic), 127
fillcolor key (METAOBJ), *83*, *104*, *114*
filled (METAPOST), 67
filled key (METAOBJ), *83*, *96*, *98*, *100*, *104*, *114*
fills (mfpic)
  centered circles, 129
  closed objects, 133
fills, closed objects, 133
final option (pst-pdf), 800
finite state diagram, *79*
fit key (METAOBJ), *97, 98*, *100*, *102, 103*, *177*
flipping trees (METAOBJ), *110*
floor (META), 53
flow charts, *177*, *181*
font files, *69*
fonts
  encoding, 65
  magsteps, 70, 71
  PostScript, 71
fonts *(cont.)*
  size, *70*, 71
for (META), **52**, ***55***, 59, **66**, **150**
forever (META), *56*, *204*
format (graph), *159*
fractals
  Hilbert's curve, *194*
  Koch flake, *105*
  METAOBJ, 104, *105*
  METAPOST, *194, 195*
  Sierpiński's curve, *194*
  Verhulst diagrams, *195*
frame (graph), **158**, ***159***, *160–162*, *164–166*
framecolor key (METAOBJ), *83*, *104*
framed key (METAOBJ), *82, 83*, *104*
frames
  aligning boxes (METAOBJ), *104*
  graphs, 158, *159*
  trees (METAOBJ), *112, 113*
framestyle key (METAOBJ), *177*
framewidth key (METAOBJ), *83*
Frequently Asked Questions (FAQs), *see* online resources
fullcircle (META), ***63***, ***66***, 74, *165*
fulldiamond (metafun), *74*
fullsquare (metafun), 74
\function (mfpic), *123, 124*, **133**
functions
  drawing, *168, 169*
  plotting (mfpic), 133, *135*

G

\gclear (mfpic), **133**, *134*
gdata (graph), ***160, 161***, *162, 163*, *165*, ***166***, *167*
gdotlabel (graph), 158
gdraw (graph), ***157, 158***, *160*, *162*, *164–166*, *169*
gdrawarrow (graph), 158
gdrawdblarrow (graph), 158
generator (makecirc), *197*, 199
geometriesyr16 METAPOST package, 192
geometry
  art, *195*
  ellipse in a parallelogram, *191*
  fractals, *194, 195*
  golden ratio, *192*
  hand-drawn figures, *192*
  Hilbert's curve, *194*
  nine points circle of a triangle, *190*
  plane, *190, 191*, 192
  space, 192
  Verhulst diagrams, *195*
.gf file extension (META), 69–71
\gfill (mfpic), 127, *131*, **133**, *134*
gfill (graph), ***159***, 160, *161–165*, *167*
gftopk program, 70
ghostscript program, 798

- ghostview program, 804
- glabel (graph), ***157, 158***, *162–167*, *169*
- global modifiers (mfpic), 132
- globes, *209*
- gnuplot program, 137
- golden ratio, *192*
- gpdata METAPOST package, 167
- gradients, tools, *143, 144*
- grap program, 157
- graph METAPOST package, 75, 122, **157, 158**, 159, 162, 167–169
- graphics package, 72
- graphicx package, 800
- graphs
  - bar charts, *162, 163*, 164, *166*
  - Bond, 177
  - data files
    - comment lines, 167
    - reading, *160–162*
  - dual bar charts, *164*
  - frames, 158, *159*
  - grids, 158, *159*
  - inserting in LaTeX, 167
  - labels
    - aligning, 173
    - annotations, 134
    - creating, *159*, 160
    - pie charts, 173, 174
    - positioning, 173
    - shifting, 173, *174*
  - overview, *157, 158*
  - pie charts
    - drawing, *165*, 171–173
    - height, 171
    - labels, 173, 174
    - observation angle, 171
    - offsets, 171
    - radius, 171
    - segments, ***170***, *171, 172*, 175, 176
    - setup for, 174, 175
    - text handling, 174
  - scales, 158, *159*
  - text, printing, *167*
  - ticks, 158, *159*
  - types of, 162–167
- graying, 75
- green (METAPOST), 60
- greenpart (METAPOST), *150*
- \grid (mfpic), 129
- grid
  - (exteps), *156*
  - (graph), **158**, 159
- grids
  - mfpic, 129
  - from lines, *147*
- grids *(cont.)*
  - from multiple base patterns, *147*
  - from squares, *147*
  - graphs, 158, *159*
- ground (makecirc), *197*, 199

H

- halign key (METAOBJ), *116*
- hand-drawn figures, *192*
- hash marks, length of (mfpic), 131
- \hashlen rigid length (mfpic), 129, 131
- \hatch (mfpic), 131, **133**
- hatch_match (hatching), *149*
- hatchfill (hatching), ***149***, *150*
- hatching
  - hatch macro, *148*
  - hatching package, *149, 150*
  - line spacing (mfpic), 131, 133, *134*
- hatching METAPOST package, 149
- hatchoptions (hatching), *149*
- \hatchspace rigid length (mfpic), 131, 133
- HBox (METAOBJ class), 100, 102, 106
- hbsep key (METAOBJ), *102*, *107*, *110, 111*
- \headlen rigid length (mfpic), 132
- \headshape (mfpic), 132
- help, *see* online resources
- hexagonal meshes, *210*
- hexagonaltrimesh (featpost), *210*
- HFan (METAOBJ class), 113, 114
- hideleaves key (METAOBJ), *110–114*
- hiding/showing lines, 145
- Hilbert's curve, *194*
- History (metaUML), 188
- hookleftarrow (cmarrows), *189*
- hookrightarrow (cmarrows), *189*
- horizontal
  - box alignment (METAOBJ), *101*
  - box separation (METAOBJ), *102*
  - fans, trees (METAOBJ), 113, *114*, 115
- How To Ask Questions The Smart Way, 810
- HRazor (METAOBJ class), 82, 114
- hsep key (METAOBJ), *102*, *108–113*, *118*
- hyperlinks, slides, *797–818*
- hyperref package, 798, 803–805

I

- ifthen package, 136
- image (METAPOST), **95**, *146, 148, 149*, ***163***, *165*, 176
- imesh (makecirc), **199**, *202*
- impedance (makecirc), ***197***, **199**, *202*
- METAPOST, 137, 138
- inactive option (pst-pdf), 800
- index boxes, *180*
- inductor (makecirc), ***196, 197***, *198*, ***199***, *200*

- inductors, *196*
- infont (METAPOST), *163*, *165*
- init_numbers (graph), 159
- initlatex
  - (latex), 64
  - (makecirc), *196*
- input (META), **67**, 75
- internal structures, 65, *66*, 67
- interpath (META), *152*
- interpol METAPOST package, 167
- interpolate (metafun), 152
- interpolating (METAPOST), 167
- intersectionpoint (META), *191*
- intersectiontimes (META), *148*, *205*
- introspection, 66, *67*
- item (metaUML), *186*
- itick (graph), 158, *159*

J

- joining objects (mfpic), 126
- .jpeg file extension (pst-pdf), 806
- junction (makecirc), ***197***, **199**, *200–202*

K

- kindofcube (featpost), *210, 211*
- Koch flake, *105*

L

- labangle key (METAOBJ), 95, 119
- labcard key (METAOBJ), 119
- labcolor key (METAOBJ), 119, 120
- labdir key (METAOBJ), *95*, *118*, 119
- labdist key (METAOBJ), 95
- Label (piechartMP), ***170***, ***173***, *174*
- label (METAPOST), ***61***, *64*, 78, 119, 158, 200
- labelinspace (featpost), *211*
- labeloffset (METAPOST), *61*
- labels
  - mfpic, 124, 134
  - color, 120
  - connections (METAOBJ), *95*
  - erasing beneath, 120
  - graphs
    - aligning, 173
    - creating, *159*, 160
    - positioning, 173
    - shifting, 173, *174*
  - in pictures, ***61, 62***, *63*, *64*, 65
  - in space, *211*
  - METAOBJ, *118*, 119, 120
  - METAPOST, 124
  - on graphs (mfpic), 134
  - pie charts, 173, 174
  - positioning, 119
- labels *(cont.)*
  - rotating, 120
  - shifting, 120
- laberase key (METAOBJ), 119, 120
- labpathid key (METAOBJ), *118*, 119
- labpathname key (METAOBJ), 119
- labpic key (METAOBJ), *95*
- labpoint key (METAOBJ), 119
- labpos key (METAOBJ), 95, 119
- labrotate key (METAOBJ), 119, 120
- labshift key (METAOBJ), 119, 120
- Lalign key (METAOBJ), *108*, *110–113*
- lamp (makecirc), *197*, 199
- latex METAPOST package, 64, 196
- latex program, 797, 800, 801, 803, 804, 806
- LaTeX files, obtaining
  - web access, 810, 811, *812, 813*, 814
- latex.mp METAPOST package, 64
- latexMP METAPOST package, 59, 64, 151
- lcircle (metafun), 74
- \lclosed (mfpic), 132
- lefthalfarrow (cmarrows), *189*
- length (META), **52**, ***66***, ***78***, *79*, ***142***
- lft syntax (METAPOST), 61
- libraries
  - boxes package, 75–79
  - metafun package, 74, 75
- \lightershade (mfpic), 132
- linear equations, solving, 53
- linear transformation (METAOBJ), *81*
- linearc key (METAOBJ), 85, *93, 94*
- linecolor key (METAOBJ), 85, *88–93*
- lines
  - creating grids, *147*
  - hiding, *145*
  - repeating, *147*
  - segments (mfpic), 129
  - starting point (METAOBJ), *87*
  - styles (METAOBJ), *86*
  - thickness (METAOBJ), *86*
  - UML diagrams, *185*
- \lines (mfpic), *127*, **129**, *135*
- linestyle key (METAOBJ), 85, *86*, *88*, *93*
- linetension key (METAOBJ), 86, *88*, *94*
- linetensionA key (METAOBJ), 85, 88
- linetensionB key (METAOBJ), 85, 88
- linewidth key (METAOBJ), 85, *86*, *88–94*
- link (metaUML), ***184***, *185*, *188*
- llcircle (metafun), 74
- llcorner (METAPOST), *150*
- llft syntax (METAPOST), 61
- lltriangle (metafun), 74
- \lmarks (mfpic), 129
- .log file extension (mfpic), 124

- looping
  - mfpic, *136*
  - commands, *56*
  - connection lines, *91, 92*
  - lines (METAOBJ), *91, 92*
- `loopsize` key (METAOBJ), 85, *91*
- `lrcircle` (metafun), 74
- `lrt` syntax (METAPOST), 61
- `lrtriangle` (metafun), 74

M

- m3d METAPOST package, 209
- macros, META language
  - arguments, 59
  - default behavior, 59
  - defining, 57–60
  - key=value pairs, *59*, 60
  - parameters, 57
  - string evaluation, 57
  - types of, 57
  - variable names, 57
- magsteps, 70, 71
- makecirc METAPOST package, 196, 198
- makeindex program, 123, 806
- makempx program, 63
- `makepen` (META), 53
- Manhattan paths, *184*
- `mapstoarrow` (cmarrows), *189*
- margins, containers (METAOBJ), *96, 97*
- mathptm package, 65
- matlab METAPOST package, 167
- `matpos` (METAOBJ), 118
- `Matrix` (METAOBJ class), 115
- `mcangle` (METAOBJ), 118
- `mcangles` (METAOBJ), 118
- `mcarc` (METAOBJ), 118
- `mcarcbox` (METAOBJ), 118
- `mcbox` (METAOBJ), 118
- `mccircle` (METAOBJ), 118
- `mccoil` (METAOBJ), 118
- `mccurve` (METAOBJ), 118
- `mcdiag` (METAOBJ), 118
- `mcdiagg` (METAOBJ), 118
- `mcline` (METAOBJ), 84, 118
- `mcloop` (METAOBJ), 118
- `mczigzag` (METAOBJ), 118
- `meains` (makecirc), *197*, 199
- mechanical drawings, 203
- `message` (META), 68
- META language, 51–167
  - affine transforms, 53
  - closing objects, *54*
  - control points, 53
  - curves
    - 3-D, 57, *58*

META language *(cont.)*
    - controlling, *55*
    - drawing, *54*
    - `path` data, 53
  - data types, 53
  - description, 52, 53
  - drawing commands, storing, 53
  - linear equations, solving, 53
  - looping commands, *56*
  - macros
    - arguments, 59
    - default behavior, 59
    - defining, 57–60
    - key=value pairs, *59*, 60
    - parameters, 57
    - string evaluation, 57
    - types of, 57
    - variable names, 57
  - `pair` data, 53
  - `path` data, 53
  - paths, transforming, *56*
  - `pen` data, 53
  - pens, 53, *55*
  - `picture` data, 53
  - point representation, 53
  - segments, 53
  - straight lines, drawing, *54*
  - `transform` data, 53, *56*
- METAFONT mode (mfpic), *123*
- metafun METAPOST package, 61, 73–75, 138, 143, 151, 152
- Metagraf program, 209
- METAOBJ METAPOST package, 80–120
  - basic objects, 82, 83
  - box alignment
    - centering, *103*
    - horizontal, *101*
    - horizontal separation, *102*
    - mixed objects, *102, 103*
    - vertical, *101*, *103*
    - within frames, *104*
  - concepts, 81
  - connections
    - arcs, *88*
    - arrow style, *87*
    - behind objects, *90*
    - Bézier curves, *87, 88*
    - circles, *92*
    - coils, *94*
    - curved boxes, *93*
    - double straight line, *87*
    - inside boxes, *92, 93*
    - labels for, *95*
    - line starting point, *87*
    - line style, *86*
    - line thickness, *86*

METAOBJ METAPOST package *(cont.)*
- looping lines, *91, 92*
- multi-segment lines, *89–91*
- overview, 84–86
- rounded corners, *93*
- straight lines, *86, 87*
- zigzags, *94*

containers
- circular, *98–100*
- description, 95
- double-walled box, *99, 100*
- double-walled circle, *100*
- double-walled ellipsis, *100*
- elliptical, *98–100*
- margins, *96, 97*
- oval boxes, *96*
- polygons, *97*
- rounded corners, *96*
- simple box, *95*
- square box, *95*

description, 80

empty boxes, *82, 83*

fractals, 104, *105*

labels, *118*, 119, 120

linear transformation, *81*

principles, 80

recursive objects, *104*, 105

trees
- aligning, *107, 108*
- flipping, *110*
- framing, *112, 113*
- horizontal fans, 113, *114*, 115
- left to right, *109*
- mixed directions, *110*
- mixed objects, *111*
- overlapping subtrees, *111*
- overview, 105
- right to left, *108*
- root at the bottom, *109*
- separating, *111*
- vertical fans, 113, *114*, 115

METAPOST mode (mfpic), *124*

MetaUML METAPOST package, 181

metric (.tfm) output files, 69

\mfpdefinecolor (mfpic), 128

\mfpic (mfpic), 124, 125

mfpic env. (mfpic), 124, 125, **126**, 135

mfpic package, 52, 120, **122–136**, 139

\mfpicdebugfalse (mfpic), 124

\mfpicdebugtrue (mfpic), 124

\mfpicdraft (mfpic), 125

\mfpicfinal (mfpic), 125

\mfpicnowrite (mfpic), 125

\mfpicnumber (mfpic), 126

\mfpicunit rigid length (mfpic), 126, 132

\mfpverbtex (mfpic), 124

mft program, 137

mftoeps METAFONT package, 138

mode, *69, 70*

mode (META), 69

mode_setup (META), 70

modifiers (mfpic), *127*

morphing, *152*

motor (makecirc), *197*, 199

.mp file extension (METAPOST), 63

mpattern METAPOST package, 148

mpcirc METAPOST package, 196, 203

mpos (METAOBJ), 118

mproof package, 73, 74

.mps file extension (METAPOST), 72

mpsproof package, 73, 74

mpt program, 137

mptopdf program, 73, 75

mptotex program, 63

.mpx file extension (METAPOST), 63

Mreadpath (graph), 167

multi-segment lines (METAOBJ), *89–91*

multipaths, 145

N

\name (mfpic), 129

name key (METAOBJ), 85, 119

naming output files, 70

nb (METAOBJ), ***116***, *117*

ncangle (METAOBJ), ***89***, 90

ncangles (METAOBJ), ***89***, *90*, 91

ncarc (METAOBJ), ***88***, 93

ncarcbox (METAOBJ), 85, **92**, ***93***

ncbar (METAOBJ), **88**, ***89***, *177*

ncbox (METAOBJ), 85, ***92***, *93*

nccircle (METAOBJ), 84, *92*

nccoil (METAOBJ), *94*

nccurve (METAOBJ), 85, ***87***, *88*

ncdiag (METAOBJ), *90*

ncdiagg (METAOBJ), *90*

ncline (METAOBJ), ***84***, ***86***, *87*, *95*, 119, *177*

ncloop (METAOBJ), 85, **90**, ***91***

nczigzag (METAOBJ), *94*

new_Box (METAOBJ), 81

new_Box_ (METAOBJ), 81

new_Circle (METAOBJ), *114*

new_HFan (METAOBJ), *114*

new_HFan_ (METAOBJ), *114*

new_RBox (METAOBJ), *114*

newBox (METAOBJ), **81**, ***95***, *96*, **100**, ***101***, *102–104*, **114**, *177*

newCircle (METAOBJ), ***86***, ***99***, *104*, *177*

newContainer (METAOBJ), *104*

\newcounter (mfpic), *136*

newDBox (METAOBJ), *99, 100*

newDEllipse (METAOBJ), ***81***, ***100***, *112, 113*

- newEllipse (METAOBJ), ***98***, **100**, *104*, *113*, *177*
- newEmptyBox (METAOBJ), *82*
- newHBox (METAOBJ), **100**, ***101***, *102*
- newHFan (METAOBJ), 114
- newHRazor (METAOBJ), **82**, ***83***, *102*
- newMatrix (METAOBJ), ***115, 116***, *117*
- newPolygon (METAOBJ), **96**, ***97***, *102, 103*, *177*
- newPTree (METAOBJ), 105
- newRandomBox (METAOBJ), *83*
- newRBox (METAOBJ), 96, *104*, 114
- newRecursiveBox (METAOBJ), *104*
- news groups, 810, *see also* online resources
- newTree (METAOBJ), **105**, ***107***, *108–113*
- newVBox (METAOBJ), 102, *103*
- newVFan (METAOBJ), 114
- newVonKochFlake (METAOBJ), *105*
- newVRazor (METAOBJ), 82, 83, *103*
- nine points circle of a triangle, *190*
- \nocenteredcaptions (mfpic), 124
- \noclearsymbols (mfpic), 124
- \noclipmfpic (mfpic), 124
- nodesep key (METAOBJ), 86
- nodesepA key (METAOBJ), 85, 87, *92, 93*
- nodesepB key (METAOBJ), 85, *87*, *92, 93*
- \nomplabels (mfpic), 124
- \nooverlaylabels (mfpic), 124
- nopstricks option (pst-pdf), 800
- normaldeviate (META), 53
- notightpage option (pst-pdf), 800
- \notruebbox (mfpic), 124
- ntreepos (METAOBJ), 120
- nullpen (META), 53
- nullpicture (META), ***66***, *150*
- numbering pictures (mfpic), 126
- numeric (META), 53

O

- Obj (METAOBJ), 81, *84*, *114*, *118*, 120
- object outlines (mfpic), 133
- ObjLabel (METAOBJ), *118*, 119
- observation angle, pie charts, 171
- offset key (METAOBJ), 86
- offsetA key (METAOBJ), 85, *87*, *90, 91*, 120
- offsetB key (METAOBJ), 85, 87, *91*, 120
- offsets, pie charts, 171
- oldtexarrow (cmarrows), *189*
- online access to CTAN, 810, 811, *812, 813*, 814
- online resources
  - archived files, finding and transferring, 813
  - CTAN (Comprehensive TEX Archive Network), 810
    - web access, 810, 811, *812*, 813, 814
  - documentation
    - command-line interface, *815*
    - panel interface, *816*
    - search by name, *815*
- online resources *(cont.)*
    - search by product, *816*
    - texdoc, *815*
    - texdock, *816*
  - FAQs (Frequently Asked Questions), 809
  - files, getting from the command line, *814*
  - How To Ask Questions The Smart Way, 810
  - news groups, 810
  - program files, obtaining
    - web access, 810, 811, *812*, 813, 814
  - TEX file catalogue, 811
  - TEX files, 810
  - TEX user groups, 817, 818
  - TUG home page, 810, *811*
- open objects, closing, 132
- \opengraphsfile (mfpic), 124, **125**
- optical drawings, 204, *205, 206*
- origin (META), 160, *161*
- otick (graph), **158**, ***159***, *166*
- OUT syntax (METAPOST), 158
- output files
  - bitmap (.gf), 69, *70*
  - EPS (Encapsulated PostScript), *72*, 73
  - metric (.tfm), 69
  - naming, 70
  - PDF (Portable Document Format), 72, 73
- oval box containers (METAOBJ), *96*
- ovals, *179*
- overlapping subtrees (METAOBJ), *111*
- \overlaylabels (mfpic), 124

P

- pair (META), **53**, *56*, 60, 84
- \parafcn (mfpic), **133**, 136
- parallel gradients, *143, 144*
- parallelarrows (cmarrows), *189*
- paralleloppositearrows (cmarrows), *189*
- paralleloppositelefthalfarrows (cmarrows), *189*
- paralleloppositerighthalfarrows (cmarrows), *189*
- parametric curves, plotting, 133, *135*
- path (META), **53**, ***55***, *56*
- pathCut (metaUML), 185
- pathfillcolor key (METAOBJ), 85
- pathfilled key (METAOBJ), 85
- pathHorizontal (metaUML), 185
- pathManhattanX (metaUML), 184
- pathManhattanY (metaUML), 184
- pathofstraightline (featpost), *211*
- pathpart (METAPOST), *66*, 150
- paths
  - between object centers, *186*
  - between objects, *185*
  - Bézier, 128
  - interrupting, 145, *146*
  - multipaths, 145, *146*

- paths *(cont.)*
  - transforming, *56*
  - UML diagrams
    - arbitrary, relations between, 184
    - between object centers, *186*
    - between objects, *185*
    - lines, *185*
    - Manhattan, *184*
    - rectangular, *184*
    - stair-like, 184, *185*
- `pathStepX` (metaUML), 184
- `pathStepY` (metaUML), 184
- `pathVertical` (metaUML), 185
- patterns, 147–150
- `.pdf` file extension (pst-pdf), 806
- PDF output files, 72, 73
- pdfcrop program, 804
- pdfinfo program, 804
- pdflatex program, 797, 800, 801, 803, 805, 806
- PDFs
  - creating
    - dvipdfm program, 798–800
    - dvipdfmx program, 798–800
    - from LaTeX, 803–807
    - from PostScript, 800, *801*, 802, 803
    - overview, 797
    - pst-pdf package, 800, *801*, 802, 803
- pdftex program, 797, 798
- pdftops program, 806
- `\pen` (mfpic), *127*, **132**, *134*
- `pen` (META), 53
- `pencircle` (META), **53**, ***55***, *56*, *79*, *162*
- pens
  - META language, 53, *55*
  - setting width (mfpic), 132
- `pensquare` (META), *166*
- perspective projection, 208
- physics diagrams, *209*
- `pic` (boxes), 76, *77*, *79*
- pic language, 75
- `pickup` (META), ***55***, *56*, *79*, *162*, *166*
- `picture` (META), **53**, ***62***, *63*, **65**, *66*, *95*, **146**, *206*
- `picture` env., 797
  - (emp), 121
- pictures
  - annotating, ***61, 62***, *63, 64*, 65
  - numbering, 126
  - size, specifying, 126
  - text in, *61–64*, 65
- pie charts
  - mfpic, *131*
  - drawing, *131*, *165*, 171–173
  - height, 171
  - labels, 173, 174
  - observation angle, 171
- pie charts *(cont.)*
  - offsets, 171
  - radius, 171
  - segments, **170**, *171, 172*, 175, 176
  - setup for, 174, 175
  - text handling, 174
- `PieChart` (piechartMP), ***170, 171***, *172–174*
- `\piechart` (mfpic), *131*
- `PiechartBBox` (piechartMP), *176*
- piechartMP METAPOST package, 143, 170, 176
- `\piewedge` (mfpic), *131*
- pin connections, 200
- `.pk` file extension (META), 69, 70
- plain METAPOST package, 74, 75
- plane geometry, *190, 191*, 192
- `\plot` (mfpic), 125
- `plot` (graph), *158*
- `\plotnodes` (mfpic), 125
- `\plotsymbol` (mfpic), 124, 125, **129**
- plotting functions and parametric curves (mfpic), 133, *135*
- `\plrfcn` (mfpic), 133
- `\plrregion` (mfpic), 133, **134**
- `.png` file extension (pst-pdf), 806
- `\point` (mfpic), 124, 125, **129**, 132
- `point` (META), ***78***, *79*, ***142***
- point representation, 53
- `\pointdef` (mfpic), 129
- `pointfilled` boolean (mfpic), 132
- `\pointfillfalse` (mfpic), 125
- `\pointfilltrue` (mfpic), 125
- `\pointsize` rigid length (mfpic), 129, 132
- polar coordinates, *169*
- `Polygon` (METAOBJ class), 97
- `\polygon` (mfpic), 129
- polygons
  - closed, 129
  - containers (METAOBJ), *97*
  - regular, 129
- `\polylines` (mfpic), 129
- `polymargin` key (METAOBJ), *97*, *102, 103*, *177*
- `pos` key (METAOBJ), 84, 86
- `posA` key (METAOBJ), *81*, 84–86, *87*
- `posB` key (METAOBJ), *81*, 84–86
- positioning labels
  - connections, *95*
  - overview, 119
- PostScript
  - commands, *155, 156*
  - fonts, 65
  - PDFs from, 800, *801*, 802, 803
- `postscript` env. (pst-pdf), 802
- pretty printing (mfpic), 137
- preview package, 800–802
- `\PreviewEnvironment` (pst-pdf), 801

previewing
    characters, 69
    drawings, *73*, 74
primitives (mfpic), 126
printing text, *167*
PrivatePattern (piechartMP), *176*
program files, obtaining
    web access, 810, 811, *812, 813*, 814
projected segments, *211*
prologues (METAPOST), 64, 65
ps2pdf program, 797, 801–806
ps2pdf13 program, 804, 805
psfonts.map file (dvips), 65
psmatrix env. (pst-pdf), 800
pspicture env. (pst-pdf), 800
pst-pdf package, 797, **800–803**, 805, 806
\pst@object (pst-pdf), 800
pstricks option (pst-pdf), 800
pstricks package, 797, 800

R

radius, pie charts, 171
Ralign key (METAOBJ), *109–113*
random number generators, *203*
RandomBox (METAOBJ class), 83
randomized (metafun), *74*
rbox_radius key (METAOBJ), *96*
rboxes METAPOST package, 76
rboxit (rboxes), 76
rcircle (metafun), 74
rdrawarrow (METAOBJ), 84
readfrom (METAPOST), 67, 68
rebindrelativeObj (METAOBJ), *108, 109*
rebindVisibleObj (METAOBJ), ***112***, *113*
\rect (mfpic), 129
rectangles
    slanted, *179*
    with corners (mfpic), *129*
rectangular paths, *184*
recursive objects (METAOBJ), *104*, 105
RecursiveBox (METAOBJ class), 104
red (METAPOST), 60
redpart (METAPOST), *150*
reflectedabout (META), *62*
\regpolygon (mfpic), 129
regular polygons (mfpic), 129
relations, diagrams, *180*
repeating lines, *147*
repetitive drawings (mfpic), 134
resistor (makecirc), ***196, 197***, *198*, ***199***, *200, 201*
resistors, *196*
\reverse (mfpic), 133
reversing objects (mfpic), 133
rheostat (makecirc), *197*, 199
righthalfarrow (cmarrows), *189*
\rmarks (mfpic), 129
rncangle (METAOBJ), 118
rncangles (METAOBJ), 118
rncarc (METAOBJ), 118
rncarcbox (METAOBJ), 118
rncbar (METAOBJ), 118
rncbox (METAOBJ), 118
rnccoil (METAOBJ), 118
rnccurve (METAOBJ), 118
rncdiag (METAOBJ), 118
rncdiagg (METAOBJ), 118
rncline (METAOBJ), 118
rncloop (METAOBJ), 118
rnczigzag (METAOBJ), 118
rotated (META), ***55***, *56*, *63*, *162–165*
rotatedabout (META), *62*, *194*
rotatedaround (META), *56*
rotateObj (METAOBJ), *81*
\rotatepath (mfpic), 133
rotating
    labels, 120
    objects (mfpic), 133, *135*
round (META), *161*
rounded boxes, *179*
rounded corners (METAOBJ)
    connections, *93*
    containers, *96*
rpathHorizontal (metaUML), 185
rpathManhattanX (metaUML), 184
rpathManhattanY (metaUML), 184
rpathVertical (metaUML), 185
rt syntax (METAPOST), 61
running, 68–73

S

scaled (META), ***55***, *56*, *62, 63*, *66*, *74*, *79*, *162, 163*, *165, 166*
scaleObj (METAOBJ), ***81***, *104, 105*, *107–112*, ***113***, *117*
scales, 158, *159*
scantokens (META), **57**, **68**, ***160***, *161–165*, ***166***, *167*
science and engineering drawings
    electrical circuits
        capacitors, *196*
        centering elements, *198*
        centering text, *200–202*
        command syntax, 199
        element abbreviations, 198
        element types, 199
        elements of, 196–199
        inductors, *196*
        pin connections, 200
        resistors, *196*
        symbols, 196, 197, *198*
        wiring type, *198*
    mechanical drawings, 203
    optics, 204, *205, 206*

science and engineering drawings *(cont.)*
- random number generators, *203*
- simulation, *203*

`\sclosed` (mfpic), 132
`\sector` (mfpic), 129
`Segment` (piechartMP), ***170***, *171–174*, *176*
segments
- META language, 53
- pie charts, *170–172*, 175, 176
- projected, *211*

`SegmentState` (piechartMP), 171, **172**, *173*, ***174***
`setbounds` (METAPOST), ***155***, *156*
`setcoords` (graph), 160
`\setcounter` (mfpic), *136*
`setCurveDefaultOption` (METAOBJ), 84, 86
`setObjectDefaultOption` (METAOBJ), *110*, *114*
`setrange` (graph), **160**, ***161***, *162, 163*, *166, 167*
`\setrender` (mfpic), 126
`SetupColors` (piechartMP), *173, 174*
`setupLaTeXMP` (latexMP), *64*
`SetupName` (piechartMP), 175
`SetupNumbers` (piechartMP), 174
`SetupPercent` (piechartMP), ***170***, *174*, 175
`SetupText` (piechartMP), 174, *175*
`SetupValue` (piechartMP), *175*
`\shade` (mfpic), *127*, 131, **133**
`\shadespace` rigid length (mfpic), 131–133
shading, dot spacing (mfpic), 131–133, *134*
`shifted` (META), ***56***, *62*, *66*, *142*
shifting labels, 120
`shortaxisarrow` (cmarrows), *189*
`show_empty_boxes` (METAOBJ), 82, *83*
Sierpiński's curve, *194*
`simplified` (metafun), *75*
simulation, *203*
`sind` (META), 53
slanted rectangles, *179*
slides (color), overlay specification
- hyperlinks, *797–818*

`smoothed` (metafun), *75*
`source` (makecirc), ***197***, **199**, *201, 202*
space geometry, 192
`spatialhalfcircle` (featpost), *209*
`\special`, 797
`special` (META), ***155***, 156
spirals (mfpic), *136*
`sqrt` (META), 53, *195*
square box containers (METAOBJ), *95*
squares
- creating grids, *147*
- repeating, *147*

`squeezed` (metafun), *74*
stair-like paths, 184, *185*
`State` (metaUML), *187, 188*
states, UML
- composite, *188*
- defining, *187*
- internal transitions, *188*
- special, 188

`stateTransitions` (metaUML), *188*
`step` (META), ***55***, *205*
stereotypes, UML, *183*
straight lines
- connections (METAOBJ), *86, 87*
- drawing, *54*

`string` (META), 53, *142*
`stroked` (METAPOST), *66*, 67
styles
- arrows, *188*
- lines
  - connections, *86*
  - thickness, *86*
- turtle graphics
  - classic, *153*
  - turtle, *153, 154*

`subpath` (META), *146*
`substring` (META), *142*
`suffix` (META), 57
`switch` (makecirc), *197*, 199
symbolic names (mfpic), 129
symbols
- centered, 129
- clearing, 124
- electrical circuit diagrams, 196, 197, *198*

syntax (mfpic), 125–127

T

`T_` (METAOBJ), *118*
`tailarrow` (cmarrows), *189*
`TC` (METAOBJ), *114*, *118*
`Tc` (METAOBJ), *118*
`tcangle` (METAOBJ), 118
`tcangles` (METAOBJ), 118
`\tcaption` (mfpic), 124, **134**, *135*
`tcarc` (METAOBJ), 118
`tcarcbox` (METAOBJ), 118
`tcbox` (METAOBJ), 118
`tccircle` (METAOBJ), 118
`tccurve` (METAOBJ), 118
`tcdiag` (METAOBJ), 118
`tcdiagg` (METAOBJ), 118
`tcircle` (metafun), 74
`tcline` (METAOBJ), 84, 118
`tcloop` (METAOBJ), 118
`Template` (metaUML), 184
template objects, UML, 184
`tension` (META), **54**, ***78***, *79*
`Terminate` (metaUML), 188
`TEX` (TEX), 64

- TEX METAPOST package, 64
- TeX file archives, 810, *see also* CTAN
- TeX files, obtaining
  - web access, 810, 811, *812, 813*, 814
- `texarrow` (cmarrows), *189*
- texdoc program, 815, 816
- texdoctk program, 815–817
- text
  - along a curve, *142*
  - centering, *200–202*
  - in pictures, ***61, 62***, *63, 64*, 65
  - pie charts, 174
  - printing, *167*
- `text` (META), 57, 59
- `textext` (latexMP), *64*
- `textual` (METAPOST), 67
- `Tf` (METAOBJ), 96, *114*
- `.tfm` file extension (META), 61, 70
- `thelabel` (METAPOST), ***62***, *63*, *142*, *206*
- ticks, 158, *159*
- `tightpage` option (pst-pdf), 800
- tiling, 147–150
- `time` (META), 68
- `\tlabel` (mfpic), **134**, *135*
- `\tmarks` (mfpic), 129
- `Tn` (METAOBJ), 82
- `top` syntax (METAPOST), 61
- `Toval_` (METAOBJ), 98
- `Tr_` (METAOBJ), 96
- `transform` (META), 53
- `transformer` (makecirc), ***197***, **199**, *202*
- `transistor` (makecirc), ***197***, **199**, *201*
- transparency, **75**, *150, 151*
- `Tree` (METAOBJ class), 86, 106, 113
- `treemode` key (METAOBJ), *108–113*, *118*
- trees (METAOBJ)
  - aligning, *107, 108*
  - flipping, *110*
  - framing, *112, 113*
  - horizontal fans, 113, *114*, 115
  - left to right, *109*
  - mixed directions, *110*
  - mixed objects, *111*
  - overlapping subtrees, *111*
  - overview, 105
  - right to left, *108*
  - root at the bottom, *109*
  - separating, *111*
  - vertical fans, 113, *114*, 115
- `tripplearrow` (cmarrows), *189*
- troff program, 64, 65, 75
- `tropicalglobe` (featpost), *209*
- `true` (META), *56*
- TUG home page, 810, *811*
- `\turn` (mfpic), *134*, **136**
- `\turtle` (mfpic), 129
- turtle graphics
  - classic style, *153*
  - turtle style, *153, 154*
- `twoheadarrow` (cmarrows), *189*
- `twowayarrow` (cmarrows), *189*
- `twowaydoublearrow` (cmarrows), *189*
- `twowayoldarrow` (cmarrows), *189*
- txp METAPOST package, 142

U

- `Ualign` key (METAOBJ), *109, 110*
- `ulcircle` (metafun), 74
- `ulft` syntax (METAPOST), 61
- `ultriangle` (metafun), 74
- UML diagrams
  - activities
    - beginning, *187*
    - constructing, *187*
    - ending, *187*
  - actors, *187*
  - arrows, *188*
  - associations, *186*
  - between object centers, *186*
  - between objects, *185*
  - braces, 188
  - class relations, *184*
  - class templates, typesetting, *183*
  - overview, 181
  - paths
    - arbitrary, relations between, 184
    - between object centers, *186*
    - between objects, *185*
    - lines, *185*
    - Manhattan, *184*
    - rectangular, *184*
    - stair-like, 184, *185*
  - rectangular, *184*
  - sample, *181*
  - stair-like, 184
  - states
    - composite, *188*
    - defining, *187*
    - internal transitions, *188*
    - special, 188
  - stereotypes, defining, *183*
  - template objects, creating, 184
  - use cases, *186*
- `unfill` (META), ***56***, *151*, ***163***, *165*, *206*
- `uniformdeviate` (META), **53**, ***204***, *210*
- unit length, basic (mfpic), 132
- `unitcircle` (metafun), 74
- `unitdiamond` (metafun), 74
- `\unitlength` (emp), 121
- `unitsquare` (META), ***74***, *75*, ***151***, 153

`unitvector` (META), *191*
`until` (META), *55*
`upto` (META), *56*
`urcircle` (metafun), 74
`urcorner` (METAPOST), *142*, *150*
`urt` syntax (METAPOST), 61
`urtriangle` (metafun), 74
use cases, UML, *186*
`Usecase` (metaUML), *186*
`\usecenteredcaptions` (mfpic), 124
`\usemetapost` (mfpic), 124
`\usemplabels` (mfpic), 124
`\usetruebbox` (mfpic), 124

V

`valign` key (METAOBJ), *116*
`vardef` (META), 57, *78*
`VBox` (METAOBJ class), 100, 102, 106
`vbsep` key (METAOBJ), *103*
`verbatimtex` (METAPOST), ***63***, 124, 175
Verhulst diagrams, *195*
vertical fans, trees (METAOBJ), 113, *114*, 115
`VFan` (METAOBJ class), 113, 114
`viewcentr` (featpost), 209
`visible` key (METAOBJ), 85
`VonKochFlake` (METAOBJ class), 105
`VRazor` (METAOBJ class), 83, 114
`vsep` key (METAOBJ), *110–113*
VTeX program, 797

W

wedge of a circle (mfpic), 129
wget program, 814

`whatever` (META), **160**, *162*, *166*, ***190***
`\whiledo` (mfpic), *136*
`white` (METAPOST), 60
`wire` (makecirc), ***196***, ***198***, *199–202*
`wireU` (makecirc), ***200***, *202*
wiring type, *198*
`withcolor` (METAPOST), ***62***, ***66***, ***74***, *79*, **149**, 158, *159*, *161–163*, *165*, *167*
`withdots` (METAPOST), *88*, *162*
`within` (METAPOST), ***66***, 67, *146*, ***150***
`withpen` (META), 158
`write` (METAPOST), 68

X

`\xaxis` (mfpic), 128
xetex program, 798, 803
`\xmarks` (mfpic), **129**, 130
`xpart` (META), **53**, ***56***, 198
xpdf program, 804
`xscaled` (META), *149*
`.gf` (bitmap) output files, 69, *70*
`.tfm` (metric) output files, 69

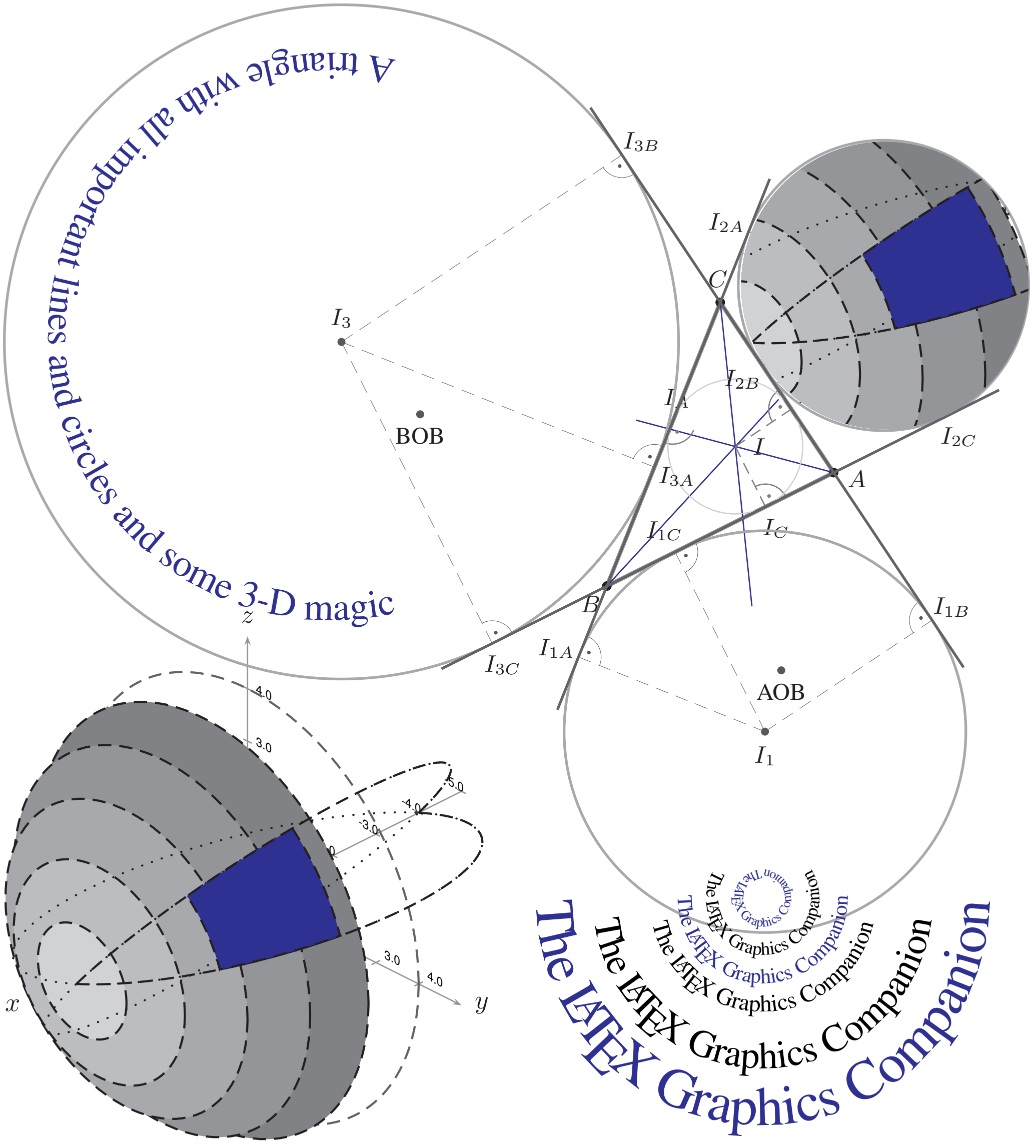
Y

`\yaxis` (mfpic), 128
`\ymarks` (mfpic), 129
`ypart` (META), **53**, ***56***, 198
`yscaled` (META), ***55***, *149*

Z

zigzag lines (METAOBJ), *94*
zlib program, 799

PSTricks


A triangle with all important lines and circles and some 3-D magic
I_{3B}
I_{2A}
C
I_3
BOB
I_{2B}
I_A
I
I_{2C}
I_{3A}
A
I_{1C}
I_C
B
I_{1B}
I_{1A}
I_{3C}
AOB
I_1
z
4.0
3.0
5.0
4.0
3.0
3.0
4.0
x
y
The LaTeX Graphics Companion
The LaTeX Graphics Companion
The LaTeX Graphics Companion
The LaTeX Graphics Companion
The LaTeX Graphics Companion
The LaTeX Graphics Companion

PSTricks

Symbols

\((pst-pdf), 800
(-) key value (pstricks), 261
\) (pst-pdf), 800
)-(key value (pstricks), 261
* key value (pstricks), 252
- key value (pstricks), 261
- key value (pstricks), 261
*O key value (pstricks), 267
*D key value (pstricks), 270, 271
*L key value (pstricks), 270, 271
*R key value (pstricks), 270, 271
*U key value (pstricks), 270, 271
+ key value (pstricks), 252
- key value (pstricks), 261
-) key value (pstricks), 263, 264
-<< key value (pstricks), 260
-> key value (pstricks), 259, 260, 262, 264
-] key value (pstricks), 260, 264
-o key value (pstricks), 264
<-> key value (pstricks), 261
<<- key value (pstricks), 260
<<->> key value (pstricks), 261
> syntax (pst-node), 356
>- key value (pstricks), 260
>-< key value (pstricks), 261
>>-<< key value (pstricks), 261
[-] key value (pstricks), 261
\jobname.tmp file (pst-tree), 376
{ } (curly braces), 304
^ syntax (pst-node), 356
_ syntax (pst-node), 356
]- key value (pstricks), 260
]-[key value (pstricks), 261
]-o key value (pstricks), 260
]-| key value (pstricks), 260
| key value (pstricks), 252
|*-|* key value (pstricks), 261
|-| key value (pstricks), 261
|<->| key value (pstricks), 261
|>-<| key value (pstricks), 261
3-D coordinates, 219
3-D parallel projections
  3-D lines, 402
  boxes, 404
  circles, 405
  coordinate axes, specifying, 401, 402
  dotted lines, 402
  ellipses, 405
  keywords for
    axes labels, moving, renaming, 413
    circular arcs, 412
    coordinate system rotation, 410
    dimension scale, changing, 411
    drawing style, 414, 415
    edge appearance, 412
    elliptical arcs, 412
    hidden lines, drawing, 415, 416
    list of, 410
    plane, specifying, 413
    plot points, 411

- 3-D parallel projections *(cont.)*
        - positioning the origin, *414*
        - spherical coordinates, *416*
        - suppressing coordinate axes, *411*
    - plotting mathematical functions and data, *407–409*
    - rectangles, *404*
    - spheres, *406*
    - square, *403*
    - triangle, *403*
- 3-D representation
    - buttons, *447*
    - framed objects, *447*
    - geometric objects, *445, 446*
    - grids, *447*
    - hidden lines or surfaces, *445*
    - keywords, 395
    - light effects, *447*
    - normal vector direction, *397–399*
    - rotating, *397*, *399*
    - shading, *394*
    - sides hiding sides, 397
    - types of objects, *393*
    - view angle, *397*
    - viewpoint, 395, *396, 397*
    - views, 219, 397
- 3-D views, 219

@

- \@ifnextchar, 328

A

- a key value (pst-tree), 380
- \AAJ (rrgtrees), *425*
- absolute key value (pstricks), 235, *239*
- absorption key (pst-spectra), *432*
- absorption spectra, *432*
- Acrobat Distiller program, 797, 798
- active option (pst-pdf), 800
- Add key value (pstricks), 252
- addfillstyle key (pstricks), 253, *257*
- \addto@pscode (pstricks), 292, **305**
- Adobe Reader program, 804, 817
- affected key (pst-pdgr), *431*
- algebraic key (pstricks-add), *423*
- alignment, tree node labels, *379*, *381, 382*
- all key value (pst-plot), 315, 318, 319
- Alpha key (pst-3dplot), *401*, *408, 409*, ***410***, *411*
- \AltClipMode (pstricks), 276
- \altcolormode (pstricks), 304
- amplitude1 key (pst-osci), *434*
- amsmath package, 361
- angle key (pst-node), 297, 299, *300*, *343*, 349, **351**, *352*
- angleA key (pst-node), *342–345*, 346, *348*, 349, ***351***, *352*, *360, 361*
- angleB key (pst-node), *338*, *342–345*, *348*, 349, ***351***, *352, 353*, *360, 361*
- angles
    - connections, *351*
    - in arguments, 218
    - specifications, 218, *302*
- Apollonius circles, 456
- arcangle key (pst-node), *341*, *347*, 349, **351**, *355*
- arcangleA key (pst-node), 349, **351**
- arcangleB key (pst-node), 349, **351**
- arced box connections, *347*
- \ArcL (vaucanson-g), *440*
- arcs
    - 3-D parallel projections
        - circular, *412*
        - elliptical, *412*
    - bent lines, *238*
    - commands for, 241, *242*
    - ellipses, *243*
    - separation, 247
- arcsep key (pstricks), ***247***, *248*
- arcsepA key (pstricks), *247*
- arcsepB key (pstricks), *247*
- \ARG (rrgtrees), *425*
- arm key (pst-node), 341, 349, **351**, *352*, *360*
- armA key (pst-node), *343*, 344, 349, 351, **352**, *360*
- armB key (pst-node), *344, 345*, 349, 351, **352**
- armB key value (pst-node), 342
- array env., 361
- \arraycolsep rigid length, 364
- arrayjob package, 322
- \arraystretch, 364
- ArrowA (PostScript), **294**, *295*
- ArrowB (PostScript), **294**, *295*
- ArrowFill key (pstricks-add), **418**, *419, 420*
- arrowinset key
    - (pstricks-add), *419*
    - (pstricks), **260**, *262*
- ArrowInside key (pstricks-add), **418**, *419, 420*
- ArrowInsideNo key (pstricks-add), 419
- ArrowInsideOffset key (pstricks-add), 419
- ArrowInsidePos key (pstricks-add), 419
- arrowlength key (pstricks), **260**, *262*
- arrows
    - creating your own, *264, 265*
    - custom style, *295*, 418, *419, 420*
    - inside lines and curves, *419*
    - keywords for, 260–264, 418
    - length, *262*
    - line termination, **259**, 260, 261, *263*
    - notch depth, *262*
    - pre-defined, 259–261
    - round bracket termination, *263*
    - rounded ends, *261*
    - scaling factor, *264*

- arrows *(cont.)*
  - size, 261
  - square bracket termination, *263*
  - strut width, *263*
  - transparent, unfilled, *419*
  - unfilled, inside, *420*
- `\arrows` (pstricks), **294**, *295*
- `arrows` key (pstricks), **235**, 237, 259, ***260***, *262–264*
- `arrowscale` key (pstricks), **260**, 263, *264*, *365*, *419*
- `arrowsize` key (pstricks), **260**, 261, *262*
- art, geometry, *456, 457*
- `Asterisk` key value (pstricks), 252
- `asterisk` key value (pstricks), 252
- `\attributeof` (pst-dbicons), *445*
- `auto` key value (pst-fill), 386
- automata, 438, *439–442*
- aux file (pst-tree), 376
- axes
  - 3-D parallel projections
    - labels, moving, *413*
    - renaming, *413*
    - specifying, *401, 402*
    - suppressing, *411*
  - plots
    - origin, *316*
    - specifying, *319*
- `axes` key value (pst-plot), 314, 315
- `axesstyle` key (pst-plot), **314**, *315, 316*, *321, 322*, *391, 392*

B

- `b` key value (pst-tree), 380
- `B+` key value (pstricks), 252
- `B-cp` key value (tlgc), *265*
- `BALLON` key (pst-labo), *433*
- `Bar` key value (pstricks), 252
- `\Bar` (pst-3d), 390
- bar charts, *450*
- bar codes, *453*
- `barstyle` key (pst-bar), *450*
- `baseColor` key (pst-fractal), *456*
- `Basterisk` key value (pstricks), 252
- `bbd` key (pst-tree), 370, 378
- `bbh` key (pst-tree), 370, 378
- `bbl` key (pst-tree), 370, 378
- `bbllx` key (pst-eps), *457*
- `bblly` key (pst-eps), *457*
- `bbr` key (pst-tree), 370, 378
- `bburx` key (pst-eps), *457*
- `bbury` key (pst-eps), *457*
- `Bdiamond` key value (pstricks), 252
- beamer document class, 440
- `\begin@AltOpenObj` (pstricks), 307
- `\begin@ClosedObj` (pstricks), 307
- `\begin@OpenObj` (pstricks), 307
- `\begin@SpecialObj` (pstricks), 307
- `beginAngle` key (pst-3dplot), *405*, 410, ***412***, *416*
- `belowtext` key (pst-pdgr), *431*
- bending lines, *238*
- `Beta` key (pst-3dplot), *401*, *408, 409*, ***410***, *411*
- Bézier curves
  - connections, *345*, *352*
  - drawing, *244, 245*, *291*
- `\bhpBox` (tlgc), *274*
- bibtex program, 801, 806
- `black` key value (pstricks), 216, 235
- blank spaces, tree nodes, *369*
- `\blue` (pstricks), *216*
- `blue` key value (pstricks), 216, *221*, *232*
- `blur` key (pst-blur), *450*
- `blurradius` key (pst-blur), *450*
- blurred shadows, *450*
- `Bo` key value (pstricks), 252
- `BoldAdd` key value (pstricks), 252
- `BoldAsterisk` key value (pstricks), 252
- `BoldBar` key value (pstricks), 252
- `BoldCircle` key value (pstricks), 252
- `BoldDiamond` key value (pstricks), 252
- `BoldHexagon` key value (pstricks), 252
- `BoldMul` key value (pstricks), 252
- `BoldOplus` key value (pstricks), 252
- `BoldOtimes` key value (pstricks), 252
- `BoldPentagon` key value (pstricks), 252
- `BoldSquare` key value (pstricks), 252
- `BoldTriangle` key value (pstricks), 252
- Boolean keys, 311, 312
- `border` key (pstricks), **235**, *239*, 281, *346, 347*
- `bordercolor` key (pstricks), **235**, *239*
- borders, *239*
- `bottom` key value (pst-plot), 315, **320**
- bounding boxes
  - creating, *220, 221*
  - shifting, *221–223*
  - tree nodes, *378*
- boxes, *see also* frames
  - % (percent sign), comment character, 277
  - 3-D parallel projections, *404*
  - clipping, 274, *275, 276*
  - commands for, 271–273
  - connection lines
    - drawing, *346, 347*
    - size, *353*
  - diamond-shaped, *273*
  - double frame, *272*
  - equilateral triangle, *273*
  - framing, *270*
  - ignoring spaces, 277
  - internal margins, *270*
  - isosceles triangle, *273*
  - keywords for, 270, 271
  - math, *278*, 279

boxes *(cont.)*
  oval-shaped, *273*
  rotating, *276*, 277
  scaling, 276, *277*
  separation, *270*
  shadows, *272*
  simple, *271*
  size, *270*, *273, 274*
  triangular frames, *271*, *273*
  verbatim, *278*, 279
`boxfill` key value (pstricks), 253, *255*, *257*
`boxfill` option (pst-fill), 383
`boxsep` key (pstricks), ***270***, *273*
`boxsize` key (pst-node), 346, 347, 349, **353**, *355*
`Bpentagon` key value (pstricks), 252
`br` key value (pstricks), *267*
`bracketlength` key (pstricks), **260**, *263*, *265*
`Bsquare` key value (pstricks), 252
`Btriangle` key value (pstricks), 252
`Bullet` key value (pstricks), 252
buttons, 3-D, *447*
`B|` key value (pstricks), 252

C

`C` key value (pst-node), 362, 363
`C` syntax (pstricks), 260, 261
`c` key value (pst-node), 362
`c` syntax (pstricks), 260, 261
`C-C` key value (pstricks), *261*
`c-c` key value (pstricks), *261*
calc package, 323
calendars, *452*
Cartesian coordinates, 224–226, **296**
`cc` syntax (pstricks), 260
`cc-cc` key value (pstricks), *261*
`ccurve` key value (pst-plot), 332, 333, *334*
cells, matrices
  empty cells, nodes for, *363*
  names, *364*
  spacing, *364*
`changeOrder` key (pstricks-add), *422*
charts, *see* graphs
`Circle` key value (pstricks), 252
`\Circle` (tlgc), *255*, *257*
`circle` key value (pst-node), 362, 363
`\circledipole` (pst-circ), *435*
`\circlenode` (pst-node), ***338***, 363
circles
  3-D parallel projections, *405*
  center, specifying, *241, 242*
  degrees in, specifying, 218
  fills, *241*
  keywords for, 247–249
  overview, 240
  sectors, *242*
`CircMultiply` key value (tlgc), *250*
`CircPlus` key value (tlgc), *250*
circular
  connection lines, *346*
  nodes, 337, *338*, *350*
civil engineering analysis, *436*
`\CLAUSE` (rrgtrees), *425*
`\clipbox` (pstricks), **274**, *275*
clipping boxes, 274, *275, 276*
`\closedshadow` (pstricks), ***289***, *290*
`\closepath` (pstricks), *284*
`closepath` (PostScript), **284**, *294*
closing paths, *284*
`cm->` key value (tlgc), *264*
`cm-cm` key value (tlgc), *264*
`cm-cp` key value (tlgc), *264*
`cmyk` key (pst-lens), *452*
`\Cnode` (pst-node), ***338***, *350–352*, 363, *365*
`\cnode` (pst-node), *273*, **337**, 338, *351*, *353–361*
`\cnodeput` (pst-node), *338*
`\code` (pstricks), *234*, 280, ***292***, *293–295*, **305**, 327
`coilaspect` key (pst-coil), *455*
`coilheight` key (pst-coil), *455*
coils, *455*
`coilwidth` key (pst-coil), *455*
color
  conflicts, resolving, 304
  fills, *255*
  gradients, *448–450*
  lines, *235*
  overview, 216
  setting, *295*
`\color`, 216
color package, 215, 216, 235, 304
`colsep` key (pst-node), ***362***, *363–365*
columns, matrices
  combining, *362*
  hooks, 362
  width, *365*
`comma` key (pstricks-add), 418
command summary, 459–466
commands, 219, 220
comment indicator, percent sign (%), 277
commenting out grids, 230, *231*
components
  basic packages, loading, 215, 216
  color, 216
  kernel, 214, 215
Comprehensive TeX Archive Network, *see* CTAN
connections, *see also* lines, *see also* nodes
  labels
    above the line, *357–359*
    below the line, *357–359*
    horizontal center, *359*
    middle of line, *353, 354*

- connections *(cont.)*
    - on specified segments, *355*
    - on the line, *357–359*
    - positioning, *357–359*
    - relative position, *356*
    - rotating, *354*, *357*
    - short forms, *356*
    - vertical center, *359*
  - package description (pst-node), 334, 335
  - pst-coil, *455*
  - to node center, 347, *348*
  - to node edge
    - angle, *351*
    - arced box, *347*
    - Bézier curves, *345*, *352*
    - box lines, *346, 347*
    - box size, *353*
    - circular lines, *346*
    - curved, *341*, *351*
    - diagonal lines, *342, 343*
    - gradient angle, *351*
    - looped lines, *345*, *352*
    - multiple per node, *360, 361*
    - parallel lines, *353*
    - railroad diagrams, *345*
    - segment arms, *352*
    - segmented line, *342*, *344*
    - segments, counting, *355*
    - segments, maximum number of, 354
    - separation from nodes, *350, 351*
    - straight line, *341*
- continuum spectra, *432*
- convert program, 806
- `\coor` (pstricks), ***293***, 294
- coordinates
  - 3-D, 219
  - 3-D parallel projections, rotating, *410*
  - angle specifications, *302*
  - axes, specifying, *401, 402*
  - calculating with PostScript, 296, 297, *298*
  - Cartesian, 296
  - default, 219, 296
  - determining, 296
  - double, 298, *299*
  - overview, 223, *224*
  - plotting functions and data, 314
  - polar, 296
  - relative translations, *299, 300*
  - saving and restoring, ***288***, 305
  - units, calculating, *421, 422*
- `\CORE` (rrgtrees), *425*
- `Corners` key (pst-ob3d), *446*
- `CornersColor` key (pst-ob3d), *446*
- `cornersize` key (pstricks), 233, **235**, 238, *239*
- `\cput`
  - (pst-node), 338
  - (pstricks), ***269***, 272
- `crosshatch` key value (pstricks), **253**, *255–257*, 258
- crosshatch fills, *255*
- `crosshatch*` key value (pstricks), **253**, *255*
- crossing lines, *239*
- CTAN (Comprehensive TeX Archive Network)
  - archived files, finding and transferring, 813
  - description, 810
  - files, from the command line, *814*
  - TeX file catalogue, 811
  - web access, 810, 811, *812, 813*, 814
- curly braces ({ }), 304
- `curvature` key
  - (pst-plot), 333
  - (pstricks), **247**, ***248***, *249*
- `curve` key value (pst-plot), 323, 332, *333*
- curved line connections, *341*, *351*, 369, *376*
- curves
  - arc separation, 247
  - Bézier, *244, 245*, *291*
  - coordinates relative to current point, *292*
  - curvature control, *247*
  - gradients, *248, 249*
  - keywords for, 247–249
  - mathematical plots, closing, *333*
  - overview, 240
  - parabolas, *245*
  - pen behavior, 240
  - points, displaying, 237
  - smooth
    - Bézier curves, *244, 245*
    - overview, 244
    - through a list of points, 245, *246*
- `\curveto` (pstricks), ***291***, 292
- `curveto` (PostScript), **291**, *295*
- `cyan` key value (pstricks), 216
- `\CylindreThreeD` (pst-vue3d), *445*

D

- `D` key value (pstricks), 270, *271*
- `d` key value (pstricks), 269
- `darkgray` key value (pstricks), 216, 235
- `dash` key (pstricks), **235**, *236*, *300*
- `dashed` key value (pstricks), *220, 221*, 235, *236*, 240, 281, *300*, *302*
- dashed lines, 235, 240
- `\dashedV` (tlgc), *280*
- dashes, 236
- `\Data` (tlgc), 328
- `dataError.dat` file (tlgc), 328, 329
- `\dataplot` (pst-plot), 323, *325*
- `\dataplotThreeD` (pst-3dplot), *409*
- `deceased` key (pst-pdgr), *431*
- `Decran` key (pst-vue3d), *445*

- `\def`, 328
- `\define@boolkey` (pst-xkey), *311*
- `\define@key` (pst-xkey), **311**, 312
- `\definecolor` (color), *235*, *258, 259*
- `\definecolorseries` (xcolor), *459*
- `\defineTColor` (pstricks-add), 257
- `\DefList` (pst-asr), *424*
- `\degrees` (pstricks), **218, 219**, 296, 297
- degrees, specifying for circles, 218
- `dia` key value (pst-node), 362, 363
- diagonal connections, *342, 343*, *377*
- diagrams
  - ER, *442–445*
  - graphs
    - rotating, *327*
    - within text, *439–442*
  - UML, *442–445*
- `Diamond` key value (pstricks), 252
- `diamond` key value (pstricks), 252
- `diamond*` key value (pstricks), 252
- diamond-shaped boxes, *273*, *339*
- diamonds, *233*
- `\dianode` (pst-node), ***339***, 363
- differential equations, plotting, *424*
- `\dim` (pstricks), **292**, *293*
- `dimen` key
  - (pst-node), *344*
  - (pstricks), **235**, *237*
- dimension keys, 312
- dimension scale, changing, *411*
- `\diode` (pst-circ), *435*
- `dirA` key (pst-jtree), *425*
- `displaymath` env. (pst-pdf), 800
- `displaymath` option (pst-pdf), 800
- `\displaystyle` (tex), 278
- `\Distillation` (pst-labo), *433*
- `dIter` key (pst-fractal), *456*, 457
- `dl` key value (pstricks), 269
- `\DoCoordinate` (tlgc), 329
- documentation, *see also* online resources
  - command-line interface, *815*
  - panel interface, *816*
  - search by name, *815*
  - search by product, *816*
  - `texdoc`, *815*
  - `texdock`, *816*
- `\dolinks` (rrgtrees), *425*
- `\DontKillGlue` (pstricks), 223, *303*
- `dot` key value (pst-node), 362, 363
- `dotangle` key (pstricks), 251, *252*
- `dotGrid` key value (tlgc), *228*, 229
- `\dotnode` (pst-node), **339**, *340*, 363
- dots
  - as nodes, *340*
  - defining, *250*, 251
- dots *(cont.)*
  - definition, 249, *250*
  - keywords for, 251
  - pre-defined styles, 251
  - rotating coordinates, *252*
  - size, *251*
- `dots` key value (pst-plot), 332, *333*
- `dotscale` key (pstricks), 236, 238, *251*, 252, *298*, *300*, *302*, *340*
- `dotsep` key (pstricks), **235**, *236*
- `dotsize` key (pstricks), 236, 238, *250–252*, 340
- `dotstyle` key (pstricks), 249, *250–252*, *298*, *340*
- `dotted` key value (pstricks), *221*, 235, *236*, 240, 281, *300*
- dotted lines, 235, 236, 240, *402*
- double coordinates, 298, *299*
- double frame boxes, *272*
- double lines, *236*
- `doublecolor` key (pstricks), **235**, *236*, *241*
- `doubleline` key (pstricks), **235**, *236*, *238*, *269*, 281
- `doublesep` key (pstricks), **235**, *236*, *241*
- `dr` key value (pstricks), 269
- `draft` option (pst-pdf), 800
- `drawCoor` key (pst-3dplot), *402–404*, *411*
- `\drawedge` (gastex), *439*
- `drawing` key (pst-3dplot), 410, ***411***
- `\drawloop` (gastex), *439*
- `drawStyle` key (pst-3dplot), 410, ***414***, *415, 416*
- duplicate macro names, 458
- dvipdfm program, 797, 798, 803
- dvipdfmx program, 797–799, 803, 804, 806
- dvips program, 305, 306, 797–801, 803–806
- `Dx` key (pst-plot), *224*, 315, ***317***, *318*, *324, 325*
- `dx` key (pst-plot), 315, ***317***, *318*, 319, *324, 325*
- `Dy` key (pst-plot), 315, ***317***, *318*
- `dy` key (pst-plot), 315, **317**, *318*, 319

E

- `ecurve` key value (pst-plot), 332, 333, *334*
- `ED` (PostScript), 365
- `\edef` (tex), 304
- `edge` key (pst-tree), 370, ***376***, *377*
- `\EdgeL` (vaucanson-g), *440*
- edges, 3-D parallel projections, *412*
- electrical circuits, pst-circ package, *435*
- `element` key (pst-spectra), *432*
- ellipses
  - 3-D parallel projections, *405*
  - arcs, *243*
  - drawing, *243*
  - keywords for, 247–249
  - overview, 240
  - sectors, *243, 244*
- `embedangle` key (pst-3d), 395, **399**
- emission spectra, *432*
- `emnode` key (pst-node), **362**, *363*
- `\empty`, 380

\end@ClosedObj (pstricks), 307
\end@OpenObj (pstricks), 307
\end@SpecialObj (pstricks), 307
endAngle key (pst-3dplot), *405*, 410, ***412***, *416*
endX key (makeplot), *430*
endY key (makeplot), *430*
\entity (pst-dbicons), *445*
.eps file extension (pst-eps), 457
epstopdf program, 804, 806
eqnarray env. (pst-pdf), 800
equation env. (pst-pdf), 800
equilateral triangle boxes, *273*
ER diagrams, *442–445*
error margins, mathematical plots, *329*
error messages, mathematical plots, *330*
Euclidean geometry, *426*
\everypsbox (pstricks), ***278***, *359*
extensions, lines, *234*

F

f key value (pst-node), 362, 363
\FanEnd (rrgtrees), *425*
fanned tree nodes, *369*
fansize key (pst-tree), *370*
FAQs (Frequently Asked Questions), 809, *see also* online resources
\fbox, 270, 272
\fboxrule rigid length, 272
\fboxsep rigid length, 270, 272
female key (pst-pdgr), *431*
\file (pstricks), 280, ***294***
\fileplot (pst-plot), **323**, *324*, ***325***
\fileplotThreeD (pst-3dplot), ***408***, 409
files, inserting, *294*
\fill (pstricks), ***285***, *286*
fill (PostScript), 285
fillangle key (pst-fill), *384*
fillcolor key (pstricks), 220, *233*, 253, *254–256*, *285*, *289*, 338, *392*
fillcycle key (pst-fill), 384, **385**
fillcyclex key (pst-fill), 384, *385*, *387*
fillcycley key (pst-fill), 384, *385*
fillloopadd key (pst-fill), *383*, 384, 386, *387*
fillloopaddx key (pst-fill), 384, 386
fillloopaddy key (pst-fill), 384, 386
fillmove key (pst-fill), 384, **385**
fillmovex key (pst-fill), 384, ***385***, *386*
fillmovey key (pst-fill), 384, ***385***, *386*
filloopadd key (pst-fill), 386
fills, *see also* tiling
  automatic vs. manual, **383**, *386*
  circles, *241*
  color, *255*
  complex patterns, *386*
  creating your own, *257*

fills *(cont.)*
  crosshatch, *255*
  debugging, *387*
  horizontal lines, *254*
  keywords for, 253, *383–387*
  line color, *257*
  line distance, *256*
  line gradient, *257*
  line width, *256*
  overview, 253
  package description (pst-fill), 383
  paths, 285
  rotating patterns, *384*
  row/column shifting, *385*
  simple patterns, *383*
  solid, *254*
  standard styles for, 253
  tile separation, *384*
  vertical lines, *254*
  whitespace, *256*
  with graphics, *387*
  with objects, *255*
  without marginal lines, *286*
fillsep key (pst-fill), *384*
fillsepx key (pst-fill), **384**, *385*
fillsepy key (pst-fill), **384**, *385*
fillsize key (pst-fill), 384, ***386***
fillstyle key
  (pst-fill), *383–387*
  (pstricks), 220, *233*, 253, *254–257*, *279*, *281*, *284, 285*, *289*, *392*, *448, 449*, *451*
final option (pst-pdf), 800
finite state diagrams, 438–442
floating point number keys, 312
Flower key value (tlgc), *250*
\fmark (gastex), *439*
\fnode (pst-node), ***340***, *350*, 363
\focalPoint (tlgc), *310*, 311
four corner node definition, 336
fp package, 458
fractals, *456, 457*
frame key value (pst-plot), 314–316
framearc key (pstricks), *233*, **235**, 238, ***239***, *258*, *271*, 272
FrameBoxThreeDColorHSB key (pst-fr3d), *447*
FrameBoxThreeDOn key (pst-fr3d), *447*
frames, *see also* boxes
  3-D objects, *447*
  boxes, *270*
  nodes, *340*, *350*
  rounded corners, 238, *239*
framesep key (pstricks), ***270***, *271*, 272
framesize key (pst-node), *340*, 349, ***350***
\FrameThreeD (pst-vue3d), *445*
\FRectangle (tlgc), 383
Frequently Asked Questions (FAQs), *see* online resources

`\FSquare` (tlgc), 383
- `full` key value (pst-plot), 315, **320**

`\func` (tlgc), 406

G

- `gangle` key (pstricks), *233*, **235**
- gastex package, 438, 439
- geographical representations, *438*
- geometric objects, 3-D, *445, 446*
- geometry
  - Apollonius circles, 456
  - fractals, 456, 457
  - Koch flake, 456
  - Mandelbrot set, *456*
  - Phyllotaxis, *457*
  - Sierpinski triangle, 456
- ghostscript program, 330, 798
- ghostview program, 804
- glue, 303
- gnuplot program, 330
- gradient angle connections, *351*
- gradients
  - color, *448–450*
  - curves, *248, 249*
- graphics package, 277
- graphicx package, 800
- graphs, *see also* diagrams, *see also* plotting
  - rotating, *327*
  - within text, *439–442*
- `gray` key value (pstricks), 216
- `green` key value (pstricks), 216, *241*

`\grestore` (pstricks), **285**, *286*, 288, 290
- `grestore` (PostScript), 276, 284, 285, **286**, 305, 306
- `gridcolor` key
  - (pst-gr3d), *447*
  - (pstricks), **226**, ***227***, *228*
- `griddots` key
  - (pst-plot), *332*
  - (pstricks), **226**, ***227***, *228*
- `gridlabelcolor` key (pstricks), *227*
- `gridlabels` key (pstricks), ***227***, *228*, *394*
- grids
  - 3-D, *447*
  - Cartesian coordinate system, 224–226
  - commands, defining new, 228
  - commenting out, 230, *231*
  - creating, *225*
  - embellishing pictures, *229, 230*
  - highlighting, 226
  - labels
    - font size, *227*
    - positioning, *225, 226*
  - lines
    - color, specifying, 226, *227*
    - dotted, 226, *227*

- grids *(cont.)*
    - width, specifying, *226*
  - overview, 224–226
  - subdivisions
    - creating, 227, *228*
    - line color, *228*
    - line width, 228
- `gridstyle` key value (pstricks), 222
- `GridThreeDNodes` key (pst-gr3d), *447*
- `GridThreeDXPos` key (pst-gr3d), *447*
- `GridThreeDYPos` key (pst-gr3d), *447*
- `gridwidth` key (pstricks), ***226***, **227**, *228*

`\gsave` (pstricks), **285**, *286*, 288, 290
- `gsave` (PostScript), 276, 284, 285, **286**, 305, 306

H

- Hénon attractor, *326, 327*
- `hatchangle` key (pstricks), 253, 254, *255–257*
- `hatchcolor` key (pstricks), 253, *255*, 256, *257*, *279*, *285*
- `hatchsep` key (pstricks), 253, *256*, *279*
- `hatchsepinc` key (pstricks), 253, *256*
- `hatchwidth` key (pstricks), 253, 255, *256*, *279*, *285*
- `hatchwidthinc` key (pstricks), 253, 255, *256*

`\hbox` (tex), 270
- header files, 302, 303
- help, *see* online resources
- `Hexagon` key value (pstricks), 252
- hexagons, *308, 309*
- hidden lines
  - 3-D, *445*
  - algorithms, *414*
  - drawing, *415, 416*
- hidden surfaces, 3-D, *445*
- `hiddenLine` key (pst-3dplot), **406**, 410, *411*, **414**
- hiding/showing tick marks, *316*
- high level macros, 309, *310*
- highlighting grids, 226
- `hlines` key value (pstricks), **253**, *254*, 255, *256, 257*, *281*
- `hlines*` key value (pstricks), **253**, *254*, 255
- `hooklength` key (pstricks-add), 418
- `hookwidth` key (pstricks-add), 418
- horizontal lines, fills, *254*
- How To Ask Questions The Smart Way, 810
- `href` key (pst-node), **348**, 349
- `HRInner` key (tlgc), *308, 309*

`\ht` (tex), *229–231*
- hyperlinks, slides, *797–818*
- hyperref package, 798, 803–805

I

- `iangle` key (gastex), *439*

`\IBox` (tlgc), *229–231*

`\ifcase`, *322*
- ifthen package, 323

- illustrations, *see* pictures
- images, *see* pictures
- \imark (gastex), *439*
- inactive option (pst-pdf), 800
- infix (algebraic) notation, *429, 430*
- infix-RPN package, 430
- \infixtoRPN (pst-infixplot), *430*
- information theory, *439–442*
- \Initial (vaucanson-g), *440*
- inner key value (pstricks), *237*
- \input (tex), 214
- integer keys, 312
- intensitycolor key (pst-circ), *435*
- intensitylabelcolor key (pst-circ), *435*
- intensitywidth key (pst-circ), *435*
- invisibleLineStyle key (pst-3dplot), 410, **415**
- isosceles triangle boxes, *273*
- isosceles triangles, *233*

J

- \jobname (pst-tree), 376
- .jpeg file extension (pst-pdf), 806
- \jtlong (pst-jtree), *425*
- \jtree (pst-jtree), *425*

K

- key key (pst-dbicons), *445*
- key/value interface
  - Boolean keys, 311, 312
  - defining commands with, 310–312
  - defining new keywords, 311
  - dimension keys, 312
  - floating point number keys, 312
  - integer keys, 312
  - low-level declaration, 310–312
  - real number keys, 312
  - string keys, 312
- key/value specification, 217
- keyval package, 217
- keywords
  - 3-D parallel projections
    - axes labels, moving, renaming, *413*
    - circular arcs, *412*
    - coordinate system rotation, *410*
    - dimension scale, changing, *411*
    - drawing style, *414, 415*
    - edge appearance, *412*
    - elliptical arcs, *412*
    - hidden lines, drawing, *415, 416*
    - list of, 410
    - plane, specifying, *413*
    - plot points, *411*
    - positioning the origin, *414*
    - spherical coordinates, *416*
- keywords *(cont.)*
    - suppressing coordinate axes, *411*
  - 3-D representation, 395
  - arrows, 260–264, 418
  - boxes, 270, 271
  - circles, 247–249
  - curves, 247–249
  - dots, 251
  - ellipses, 247–249
  - fills, 253, *383–387*
  - lines, 234
  - nodes, *370–378*
  - polygons, 234
  - pspicture environment, 221–223
  - PSTricks, summary, 459–466
  - symbols, 251
  - trees, *370–378*
- \KillGlue (pstricks), 223, *303*
- Koch flake, 456

L

- L key value (pstricks), 270, *271*
- l key value
  - (pst-node), 362
  - (pst-tree), 380
  - (pstricks), 269
- lab apparatus, *433*
- labels
  - 3-D parallel projection axes, moving, *413*
  - centering on objects, *269*
  - commands for, *267*
  - connections
    - above the line, *357–359*
    - below the line, *357–359*
    - horizontal center, *359*
    - middle of line, *353, 354*
    - on specified segments, *355*
    - on the line, *357–359*
    - positioning, *357–359*
    - relative position, *356*
    - rotating, *354*, *357*
    - short forms, *356*
    - vertical center, *359*
  - coordinate axes, 268
  - directions, short forms, 238
  - grids
    - font size (labels), *227*
    - positioning, *225, 226*
  - overwriting, *267*
  - plots
    - axis origin, *316*
    - axis, specifying, *318*
    - fonts (labels), 318
    - hiding, *316*
    - omitting, *319*

labels *(cont.)*
    origin, hiding, *319*
    placing, *315*
    point of origin, *316*
    spacing, *317*
    symbols as, *322, 323*
    text as, *322, 323*
  points in a graphic, *268*
  reference points, 266
  rotation angle, 266
  tree nodes
    aligning, *379*, *381, 382*
    creating, 379
    examples of, *380*
    positioning, *378*
    separation, *381*
labels key (pst-plot), 315, ***318***, *319–322*
labelsep key (pstricks), 240, 265, 268, 314, 315, 318, *345*, *357*
latex program, 797, 800, 801, 803, 804, 806
LaTeX files, obtaining
  web access, 810, 811, *812, 813*, 814
lB key value (pstricks), *267*
lb key value (pstricks), 229, *231*, *267*
length
  arrows, *262*
  ticks, *321*
  units
    converting to TeX, *293*
    setting and changing, 217
\lens (pst-optic), *434*
lenses, *434*
lensGlass key (pst-optic), *434*
LensMagnification key (pst-lens), *452*
lensScale key (pst-optic), *434*
levelsep key (pst-tree), 370, *372, 373*, ***374***, *375–377*, *382*
liftpen key (pstricks), **235**, 240, *282, 283*, *286–288*
light effects, 3-D, *447*
lightgray key value (pstricks), 216, *223*
LightThreeDColorPsCommand key (pst-light3d), *447*
line key value (pst-plot), 323, 332, *333*
lineAngle key (pstricks-add), 418
linear rays, *434*
linearc key (pstricks), *232*, **235**, *238–240*, *343*, *345*, *352*, *355*, *360*
lineColor key (pst-3dplot), *402*
linecolor key
  (pst-node), *346, 347*
  (pstricks), 219, *231–234*, ***235***, 236, *239*, *241*, *281*, *283*, *285*, *296*, *298*
linejoin key (pst-3dplot), 234, 410, ***412***
lines, *see also* connections, *see also* paths
  3-D parallel projections, *402*
  bending, *238*
  borders, *239*

lines *(cont.)*
  color
    fills, *257*
    grid subdivisions, *228*
    user defined, *235*
  crossing, *239*
  custom styles, *282, 283*, *285–291*
  double, *236*
  drawing, *231, 232*
  end markings, 237, *238*
  extensions, *234*
  fills, distance, *256*
  from current point, *285–291*
  gradient fills, *257*
  grids
    color, specifying, 226, *227*
    dotted, 226, *227*
    width, specifying, *226*
  hidden line algorithm, *414*
  hidden, drawing, *415, 416*
  keywords for, 234
  mathematical plots, customized, 328
  positioning, *237*
  styles
    custom, *282, 283*, *285–291*
    dashed, 235, 236, 240
    dotted, 235, 236, 240, *402*
    fills, *256*
    grid subdivisions, 228
    solid, 235
    width, 228, *256*
  width, 228, *234*, *256*
  zigzag, *455*
lines key value (pst-plot), 330
linestyle key (pstricks), 220, **235**, *236*, *276*, 285, *315, 316*, *332*
\lineto (pstricks), *291*
lineto (PostScript), **291**, *294*
linetype key (pstricks), 235, 240
linewidth key (pstricks), 220, *230*, *232*, *234*, ***235***, *236*, *239*, *241*, *248, 249*, 251, *259*, *261*, 262, *268, 269*, *281*, *285*
linguistics, *424, 425*
Lissajou figures, *332*
\listplot
  (pst-plot), **323**, 325, *326, 327*
  (pstricks-add), *421*
\listplotThreeD (pst-3dplot), *409*
\loop (pstricks-add), 422
looped connection lines, *345*, *352*
looping, *422*
\LoopL (vaucanson-g), *440*
\LoopN (vaucanson-g), *440*
\LoopS (vaucanson-g), *440*
loopsize key (pst-node), 344, *345*, 349, ***352***
loose key (pst-tree), 373
low level macros, 307–309

- lozenges, horizontal, *233*
- LR (restricted horizontal Left-Right) mode, 269
- `lrbox` env., 276

M

- macros
  - assigned to tree node edges, *377*
  - duplicate names, 458
  - high level, 309, *310*
  - low level, 307–309
  - special, 303–307
- `magenta` key value (pstricks), 216, *235*, *279*
- magnifying glass effect, *452*
- `\makeatletter`, 264, 365
- `\makeatother`, 264, 365
- `\makebox`, 337
- makeindex program, 806
- `makeplot` env. (makeplot), *430*
- makeplot package, 430
- `male` key (pst-pdgr), *431*
- `Mandel` key value (pst-fractal), 456
- Mandelbrot set, *456*
- `mapCountry` key (pst-geo), *438*
- maps, *438*
- `markZeros` key (pst-func), *427*
- math boxes, *278*, 279
- mathematical plots
  - adding values to data points, *327*
  - curves, closing, *333*
  - customized lines, 328
  - data delimiters, 324
  - data file, size limits, 325
  - error margins, *329*
  - error messages, *330*
  - external data, *324*
  - functions, *332*
  - Hénon attractor, *326, 327*
  - Lissajou figures, *332*
  - loading data records, 328
  - maximum upper/lower deviations, 328
  - package description (pst-plot), **323, 324**, 325, 326
  - plot points, 334
  - plot style, 332, *333*, 334
  - printing, 330
  - relative mean power values, *331*
  - rotating a graph, *327*
  - RPN (Reverse Polish Notation), 329
  - saving data records, 328
  - stack system, 329
  - symbols in data files, 324
  - tab characters, 324
  - third degree parabola with inverse function, *331*
  - watermarks, *326*
- mathematics
  - drawing polygons, *431*
  - Euclidean geometry, *426*
  - infix (algebraic) notation, *429, 430*
  - plotting `matlab` files, *430*
  - plotting special functions, *427*
  - Poisson distribution, *427*
  - PostScript extensions, *428*
  - RPN (Reverse Polish Notation), *430*
- `\mathrm`, 361
- `matlab` files, plotting, *430*
- matrices
  - nodes
    - cell names, *364*
    - cell spacing, *364*
    - column width, *365*
    - combining columns, *362*
    - empty cells, nodes for, *363*
    - node type, defining, *363*
    - overview, 361
    - positioning, *364*
    - row spacing, *364*
    - row/column hooks, 362
  - plotting, *422*
- `mcol` key (pst-node), **362**, *364*
- medical pedigrees, *431*
- `middle` key value (pstricks), *237*
- `minipage` env., 393
- mirrors, *434*
- `mnode` key (pst-node), **362**, *363, 364*
- `mnodesize` key (pst-node), **362**, 364, *365*
- Moiré effect, *258*
- monohedral tiling, 383
- `Month` key (pst-calendar), *452*
- `\movepath` (pstricks), *290*
- `\moveto` (pstricks), ***283***, 284, *291, 292*
- `moveto` (PostScript), **283**, *294*
- `\mrestore` (pstricks), *288*
- `\msave` (pstricks), *288*
- `Mul` key value (pstricks), 252
- `\multidipole` (pst-circ), *435*
- `\multido` (multido), *236*, *258*, *296*, 458, *459*
- multido package, 216, **458, 459**
- `\multips` (pstricks), ***269***, *298*
- `\multirput`
  - (pst-fill), 383
  - (pstricks), **267**, *268*, 269
- `mv` key (pst-dbicons), *445*
- `\myCoil` (tlgc), *269*
- `\myGrid` (tlgc), 229

N

- `\n?put` (pst-tree), 380
- `nab` key value (pst-node), 349, **355**
- `nAdjust` key (gastex), *439*

nAdjustdist key (gastex), *439*
name key (pst-node), 361, **362**, *363, 364*
nameX key (pst-3dplot), 410, ***413***
nameY key (pst-3dplot), 410, ***413***
nameZ key (pst-3dplot), 410, ***413***
naming nodes, 335
\naput (pst-node), *343*, 356, **357**, *358*
nArrow key (pstricks-add), 418
\nbput (pst-node), *345*, *355*, 356, ***357***, *358*
\nc???? (pst-node), 340
\ncangle (pst-node), ***343***, *344*, *351*, 355
\ncangles (pst-node), *344*
\ncarc (pst-node), *273*, *337*, ***341***, *350, 351*, 355
\ncarcbox (pst-node), 346, *347*, 353, *355*
\ncbar (pst-node), ***343***, *352*, 355, *360*, *377, 378*
\ncbox (pst-node), ***346***, *353*, 355
\nccarcbox (pst-node), 346
\nccircle (pst-node), 345, ***346***, 355
\nccurve (pst-node), *338*, ***345***, *351, 352*, 355, *360, 361*
nccurve key (pst-node), *338*
\ncdiag
  (pst-node), **341**, *342, 343*, 355
  (pstricks-add), 418
\ncdiagg (pst-node), **342**, *343*, 355, *377*
\ncline (pst-node), *230, 231*, *335, 336*, *338–340*, ***341***, *342*, *345*, *349–351*, *353–359*, *362–365*, 370, *374*
\ncloop (pst-node), **344**, *345*, *352*, 354, 355
\ncput (pst-node), *230, 231*, *344, 345*, *353–356*, ***357***, *358, 359*, *374*
\ncputicon (pst-uml), *442*
\ncSE (pst-uml), *442*
\ncSXE (pst-uml), *442*
ncurv key (pst-node), *345*, 349, **352**
ncurvA key (pst-node), 349, **352**
ncurvB key (pst-node), 349, **352**
nEnd key (pstricks-add), 418
nesting nodes, *335*
\newcommand, *228*
\newif, 311
\newpath (pstricks), *284*
newpath (PostScript), 284
\newpsfontdot (pstricks), **250**, 251
\newpsobject (pstricks), *228*, ***280***
\newpsstyle (pstricks), **222**, *228*, ***279***, 280
\newpssytle (pst-3dplot), *414*
news groups, 810, *see also* online resources
\newtier (pst-asr), *424*
nil tree nodes, *368*
Nmarks key (gastex), *439*
Nmr key (gastex), *439*
\node (gastex), *439*
nodealign key (pst-node), **362**, *364*
\nodeBetween (tlgc), *337*
nodes
  center, determining, *335, 336*
  center, moving, 348, *349*

nodes *(cont.)*
  circular, 337, *338*, *350*
  connections, *455*
  connector separation, *350, 351*
  defined radius, 337
  diamond shaped, *339*
  dots, *340*
  four corner definition, 336
  frames, *340*, *350*
  in a matrix
    cell names, *364*
    cell spacing, *364*
    column width, *365*
    combining columns, *362*
    empty cells, nodes for, *363*
    node type, defining, *363*
    overview, 361
    positioning, *364*
    row spacing, *364*
    row/column hooks, 362
  in running text, 337
  multiple connections, *360, 361*
  naming, 335
  nesting nodes, *335*
  oval shaped, *339*
  placing, 335
  plotting curves, *336*
  positioning, *336, 337*, 361
  radius, setting, *338*
  simple, *335*
  symbol size, *340*
  trees
    blank spaces, inserting, *369*
    bounding boxes, *378*
    command names, 367
    curved connectors, 369, *376*
    diagonal connectors, *377*
    distance between, *372–376*
    fanned, *369*
    keywords for, *370–378*
    level separation, *375, 376*
    macros, assigned to edges, *377*
    nil, *368*
    order, changing, *371*
    predecessors, 367–369
    reference points, setting, *368*
    reserving space for, *368*
    sets of branches, combining, *370*
    successors, 367–369
    tree direction, specifying, *371*
    types, *367*
  trees, labels
    alignment, *379*, *381, 382*
    creating, 379
    examples of, *380*

- nodes *(cont.)*
    - - positioning, *378*
      - separation, *381*
    - triangular, *339*
- `nodesep` key (pst-node), *251*, 297, *299, 300*, *335, 336*, *340, 341*, *343*, *346*, 348, 349, ***350***, *351*, *353–356*, *359*, 360, *362–364*, *368*, *374*
- `nodesepA` key (pst-node), 349, **350**, *360*
- `nodesepB` key (pst-node), 349, **350**, *360*, *368*, *374*, *377*
- `nodeWidth` key (pst-geo), *438*
- `none` key value
    - (pst-node), 349, **355**, 362, 363
    - (pst-plot), **314, 315**, 316, 318, 319
    - (pstricks), 220, 235, *236*, **253**, *276*, *289, 290*
- `nopstricks` option (pst-pdf), 800
- `normal` key (pst-3d), 395, **397**
- normal vector direction, 3-D, *397–399*
- `\NormalCoor` (pstricks), 219, 296
- `normaleLatitude` key (pst-vue3d), *445*
- `normaleLongitude` key (pst-vue3d), *445*
- `notightpage` option (pst-pdf), 800
- `noxcolor` option (pstricks), **215**, 216
- `npos` key (pst-node), *344, 345*, 349, **354**, *357, 358*, *442*
- `\nput` (pst-node), *344*, *357*, ***359***
- `nrot` key (pst-node), *344, 345*, 349, ***354***, *358*, *442*
- `nStart` key (pstricks-add), 418
- `nStep` key (pstricks-add), 418
- `\NUC` (rrgtrees), *425*
- `Nw` key (gastex), *439*

O

- `o` key value (pstricks), *251*, 252
- `o-o` key value (pstricks), 261
- object types, 307
- objects, as fills, *255*
- `offset` key (pst-node), 297, *299, 300*, 349, ***353***, *354, 355*, 360
- `offsetA` key (pst-node), 349, **353**, *360*
- `offsetB` key (pst-node), 349, **353**, *360*
- online access to CTAN, 810, 811, *812, 813*, 814
- online resources
    - archived files, finding and transferring, 813
    - CTAN (Comprehensive TeX Archive Network), 810
        - web access, 810, 811, *812*, 813, 814
    - documentation
        - command-line interface, *815*
        - panel interface, *816*
        - search by name, *815*
        - search by product, *816*
        - `texdoc`, *815*
        - `texdock`, *816*
    - FAQs (Frequently Asked Questions), 809
    - files, getting from the command line, *814*
    - How To Ask Questions The Smart Way, 810
    - news groups, 810
- online resources *(cont.)*
    - program files, obtaining
        - web access, 810, 811, *812*, 813, 814
    - TeX file catalogue, 811
    - TeX files, 810
    - TeX user groups, 817, 818
    - TUG home page, 810, *811*
- `onset` key (pst-asr), *424*
- `oo-oo` key value (pstricks), 261
- `\openshadow` (pstricks), ***289***, *290*
- `operation` key (pst-osci), *434*
- `Oplus` key value (pstricks), 252
- `oplus` key value (pstricks), 252
- `\OPR` (rrgtrees), *425*
- optical systems, *434*
- `origin` key
    - (pst-3dplot), 410
    - (pstricks), **223**, *224*, 281
- origin (3-D), positioning, *414*
- origin of ordinates, translating, *286*
- oscilloscope channels, *434*
- `Otimes` key value (pstricks), 252
- `otimes` key value (pstricks), 252
- `outer` key value (pstricks), 235, *237*
- `oval` key value (pst-node), 362, 363
- oval-shaped boxes, *273*, *339*
- `\ovalnode` (pst-node), ***339***, *342*, *345*, *348*, *352, 353*, 363
- `Ox` key (pst-plot), 315, ***316***
- `Oy` key (pst-plot), 315, **316**, *317*

P

- `p` key value (pst-node), 362, 363
- `\parabola` (pstricks), *224*, ***245***
- parabolas, *245*
- parallel connection lines, *353*
- `\parametricplot` (pst-plot), **330**, *332*
- `\parametricplotThreeD` (pst-3dplot), 405, **407**, *408*
- `\parbox`, 272, 389, 393
- paths, *see also* lines
    - closing, *284*
    - creating, *284*
    - deleting, *284*
    - filling, 285
    - moving, *290*
    - stroke, 284, *285*
- `\pc`*????* (pst-node), 348
- `\pcangle` (pst-node), 348
- `\pcangles` (pst-node), 348
- `\pcarc` (pst-node), 348
- `\pcarcbox` (pst-node), **348**, 353
- `\pcbar` (pst-node), 348
- `\pcbox` (pst-node), 348
- `\pccurve` (pst-node), **348**, *360*

`\pcdiag`
    (pst-node), *348*
    (pstricks-add), 418
`\pcdiagg` (pst-node), 348
`\pcline` (pst-node), *251*, **348**
`\pcloop` (pst-node), 348
`.pdf` file extension (pst-pdf), 806
PDF files, 458
pdfcrop program, 804
pdfinfo program, 804
pdflatex program, 457, 458, 797, 800, 801, 803, 805, 806
PDFs
    creating
        dvipdfm program, 798–800
        dvipdfmx program, 798–800
        from LaTeX, 803–807
        from PostScript, 800, *801*, 802, 803
        overview, 797
        pst-pdf package, 800, *801*, 802, 803
pdftex program, 797, 798
pdftops program, 806
pen behavior, 240
`Pentagon` key value (pstricks), 252
`pentagon` key value (pstricks), 252
`pentagon*` key value (pstricks), 252
percent sign (%), comment indicator, 277
`period1` key (pst-osci), *434*
perspective projection, *see* tilting
`phB` key (pst-asr), *424*
`PHI` key (pst-vue3d), *445*
Phyllotaxis, *457*
`picture` env., 223, 303, 797
pictures, embellishing with grids, *229, 230*
placement, *see* positioning
`plain` option (pstricks), 215
`plane` key (pst-3dplot), 410, ***413***, *414*
plot points, 3-D parallel projections, *411*
`plotpoints` key (pst-plot), *224*, 330, *332*, *334*, *405*, 406
`plotstyle` key (pst-plot), *224*, 323, *324–327*, *330–334*, *411*
plotting, *see also* graphs
    coordinate system, 314
    coordinate units, calculating, *421, 422*
    differential equations, *424*
    labels
        axis origin, *316*
        axis, specifying, *318*
        fonts, 318
        hiding, *316*
        omitting, *319*
        origin, hiding, *319*
        placing, *315*
        point of origin, *316*
        spacing, *317*
        symbols as, *322, 323*
        text as, *322, 323*
plotting *(cont.)*
    looping, *422*
    mathematical plots
        3-D parallel projections, *407–409*
        adding values to data points, *327*
        curves, closing, *333*
        customized lines, 328
        data delimiters, 324
        data file, size limits, 325
        error margins, *329*
        error messages, *330*
        external data, *324*
        functions, *332*
        Hénon attractor, *326, 327*
        Lissajou figures, *332*
        loading data records, 328
        maximum upper/lower deviations, 328
        package description (pst-plot), **323, 324**, 325, 326
        plot points, 334
        plot style, 332, *333*, 334
        printing, 330
        relative mean power values, *331*
        rotating a graph, *327*
        RPN (Reverse Polish Notation), 329
        saving data records, 328
        stack system, 329
        symbols in data files, 324
        tab characters, 324
        third degree parabola with inverse function, *331*
        watermarks, *326*
    `matlab` files, *430*
    matrices, *422*
    package description, 313
    special functions, *427*
    step functions, *423*
    ticks
        axes, specifying, *319*
        axis origin, *316*
        hiding, *316*
        length, *321*
        point of origin, *316*
        position, *321*
        size, *322*
        style, *320, 321*
`.png` file extension (pst-pdf), 806
`\pnode` (pst-node), *230, 231*, *299, 300*, *310*, **336**, 337, 363, *436*
points
    current, moving, *283*
    curves, displaying, 237
    displaying, 237, *238*
Poisson distribution, *427*
polar coordinates, *296*
`polarplot` key (pst-func), *427*
`polygon` key value (pst-plot), 332, *333*

- polygons, *see also pecific polygons*
  - drawing, *232*, *431*
  - keywords for, 234
- PolyNbSides key (pst-poly), *431*
- pOrigin key (pst-3dplot), *414*
- PosAngle key (pst-eucl), *426*
- positioning
  - labels
    - connections, *357–359*
    - tree nodes, *378*
  - lines, *237*
  - nodes, *336, 337*, 361
- PostScript
  - % (percent sign), comment character, 265
  - code, in PostScript output, *292*, *305*, 306, 307
  - coordinates, converting to TEX, *293, 294*
  - mathematical extensions, *428*
  - PDFs from, 800, *801*, 802, 803
  - sending information to TEX, 365, 366
  - stack state, saving, *286*
- postscript env. (pst-pdf), 802
- predecessor tree nodes, 367–369
- preview package, 458, 800–802
- \PreviewEnvironment (pst-pdf), 801
- printing plots, 330
- printValue key (pst-func), *427*
- .pro file extension (pstricks), 302
- program files, obtaining
  - web access, 810, 811, *812, 813*, 814
- \protect (pst-node), 335, 337
- ps2pdf program, 797, 801–806
- ps2pdf13 program, 804, 805
- \psaddtolength (pstricks), 218
- \psAppolonius (pst-fractal), 456
- \psarc (pstricks), **241**, *242*, *247, 248*, *281*, *302*, *344*
- \psarcn (pstricks), **241**, *242*, *247*, *281*, *344*
- \psArrowCivil (pst-stru), *436*
- \psaxes
  - (pst-plot), *224*, 266, *276*, **314**, *315–327*, *329–334*, *391, 392*, *459*
  - (pstricks-add), 418
- \psbarchart (pst-bar), *450*
- \psbarcode (pst-calendar), *453*
- \psbarscale (pst-bar), *450*
- \psbezier (pstricks), ***244***, *245*, 282, *290*, 291
- \psBinomialN (pst-func), *427*
- \psboxfill (pst-fill), *255*, *257*, ***383***, *384–387*
- \psCalDodecaeder (pst-calendar), *452*
- \psCalendar (pst-calendar), *452*
- \psccurve (pstricks), ***246***, *336*
- \pscharpath (pst-text), *450*
- \pscircle (pstricks), **234**, *238*, ***241***, *247*, *255*, *257*, *259*, *275*, *308, 309*
- \pscirclebox
  - (pst-node), 338
- \pscirclebox *(cont.)*
  - (pstricks), 269, 270, ***272***
- \psclip (pstricks), *276*
- psclip env. (pstricks), *259*, *275*, 276
- \psCoil (pst-coil), *455*
- \pscolhook (pst-node), *362*
- \pscolhook???? (pst-node), 362
- \pscurve (pstricks), ***245***, 246, *248, 249*, 282, *283, 284*
- \pscustom (pstricks), 234, 240, *276*, **280**, *281–290*, 293, *294, 295*, 305, 327, *436*
- \psdblframebox (pstricks), **271**, *272*
- \psdiabox
  - (pst-node), 339
  - (pstricks), *273*
- \psdiamond (pstricks), *233*
- \psdot (pstricks), 236, **249**, *250–252*, *296*, *298*, *300*, *302*, 339
- \psdot* (pstricks), 252
- \psdots (pstricks), **249**, *250*, 282, *296*
- \psecurve (pstricks), *246*
- \psedge (pst-tree), 369, 376
- \psellipse (pstricks), *239*, ***243***
- \psellipticarc (pstricks), *243*
- \psellipticarcn (pstricks), 243
- \psellipticwedge (pstricks), *244*
- \pserrorLine (tlgc), 329
- \psFArrow (pst-fractal), 456
- \psFern (pst-fractal), 456
- \psforeach (pstricks-add), *422*
- \psFractal (pst-fractal), *456*
- \psfractal (pst-fractal), 456
- \psframe (pstricks), **232**, *233*, *237*, 238, *239*, *267*, 270, *303*, *306*, 340, *383*, *393*
- \psframebox (pstricks), *258*, *270*, ***271***, *272*, *274*, *278, 279*, *352*, *448, 449*
- \psgraph (pstricks-add), *421*
- psgraph env. (pstricks-add), 421
- \psgrid (pstricks), ***225, 226***, *227–230*, 282, *324, 325*, *331*
- \psHexagon (tlgc), **307**, *308, 309*
- \pshlabel (pst-plot), *318*, ***322***
- \psKochflake (pst-fractal), 456
- \pslabelsep rigid length (pstricks), 240
- \psLame (pst-func), *459*
- \pslbrace (pstricks), 304
- \psline (pstricks), 218, *219*, ***231***, 232, *234–236*, ***237***, *238, 239*, *247*, *259–263*, *268*, *281–283*, *291*, *299, 300*, *302*, *365*
- \psline* (pstricks), 220
- \pslinecolor (pstricks), 220
- \pslinewidth (pstricks), 235, 261, 263
- \psmathboxfalse (pstricks), *278*
- \psmathboxtrue (pstricks), *278*
- psmatrix env.
  - (pst-node), **361**, *362–365*
  - (pst-pdf), 800
- \psMatrixPlot (pstricks-add), *422*
- \psovalbox (pstricks), 270, **272**, *273*, 339

- \psPhyllotaxis (pst-fractal), 456, *457*
- pspicture env.
  - (pst-pdf), 800
  - (pstricks), 218, ***220–223***, 225, 229, 303, 457
- pspicture environment
  - bounding boxes
    - creating, *220, 221*
    - shifting, *221–223*
  - keywords for, 221–223
  - missing values, determining, 221
  - whitespace between commands, 223
- pspicture* env. (pstricks), **220**, 275
- \psPlot (pst-infixplot), *429*
- \psplot (pst-plot), *224*, *276*, *283*, *285–289*, *306*, *323*, ***330***, *331*, *333, 334*, *428*
- \psplotDiffEqn (pstricks-add), **423**, *424*
- \psplotImp (pst-func), *427*
- \psplotThreeD (pst-3dplot), 406, *407*, *411*
- \pspolygon
  - (pst-plot), *320*, 333
  - (pstricks), ***232***, 237, 238, *248*, *271*, *310*
- \pspolygonbox (pst-poly), *431*
- \pspred (pst-tree), **369**, **376**, 379
- \psPTree (pst-fractal), 456, 457
- \psrbrace (pstricks), 304
- \psrowhook (pst-node), *362*
- \psrowhook*????* (pst-node), 362
- \psrunit (pstricks), 218
- \psscalebox (pstricks), 277
- \psscaleboxto (pstricks), 277
- \psset (pstricks), ***217***, *218*, 232, *259*, *311*, *418*
- \pssetlength (pstricks), 218
- \psshadow (pst-3d), ***388***, *389*
- \psshadowbox
  - (pst-tree), *378*
  - (pstricks), ***272***, 378
- \psSier (pst-fractal), 456
- \psspan (pst-node), **361**, *362*
- \psspectrum (pst-spectra), *432*
- \psStep (pstricks-add), *423*
- \pssucc (pst-tree), **369**, **376**, 379
- pst-3d package, 216, **388–400**
- pst-3dplot package, 217, 234, 313, 388, **400–416**
- pst-all package, **216**, 313
- pst-asr package, 217, **424**
- pst-bar package, 450
- pst-barcode package, 453
- pst-blur package, 449, 450
- pst-calendar package, 452
- pst-circ package, 309, **435**
- pst-coil package, 216, **455, 456**
- pst-dbicons package, 445
- pst-dots.pro file (pstricks), 250, 302
- pst-eps package, 216, **457**
- pst-eucl package, 426
- pst-fill package, 216, 255, 257, **383–387**
- pst-fr3d package, 388, **447**
- pst-fractal package, 456, 457
- pst-func package, 427
- pst-geo package, 437, 438
- pst-gr3d package, 388, **447**
- pst-grad package, 216, **448**
- pst-infixplot package, 429, 430
- pst-jtree package, 425
- pst-labo package, 433
- pst-lens package, 452
- pst-light3d package, 447
- pst-map2d package, 438
- pst-map2dll package, 438
- pst-map3d package, 438
- pst-map3dll package, 388, 438
- pst-math package, 224, **428**, 429
- pst-node package, 214, 216, 313, **334–366**, 379, 424
- pst-node.pro file (pstricks), 302
- pst-ob3d package, 388, **446**
- pst-optic package, 434
- pst-osci package, 434
- pst-pdf package, **457, 458**, 797, **800–803**, 805, 806
- pst-pdgr package, 431
- pst-plot package, 214, 216, 266, **313–334**, 400, 406, 424, 426
- pst-poly package, 431
- pst-slpe package, 449
- pst-spectra package, 432
- pst-stru package, 436
- pst-text package, 216, **451**
- pst-tree package, 214, 216, **366–382**, 424
- pst-uml package, **442**, 443
- pst-view3d package, 400
- pst-vue3d package, 388, 393, **445**
- pst-xkey package, 217, **310–312**
- \pst@arrowtable (pstricks), *264*
- \pst@checknum
  - (pst-xkey), 312
  - (pstricks), 312
- \pst@def (pstricks), 307
- \pst@getcoor (pstricks), 310
- \pst@getint
  - (pst-xkey), 312
  - (pstricks), 312
- \pst@getlength
  - (pst-xkey), 312
  - (pstricks), 312
- \pst@object (pstricks), 253
- \pst@Verb (pstricks), 305
- \pst@object (pst-pdf), 800
- pstcol package, 215
- PstDebug key (pst-fill), 384, ***387***
- \PstDie (pst-ob3d), *446*
- \pstextpath (pst-text), *451*
- \PstFrameBoxThreeD (pst-fr3d), *447*

- \PstGridThreeD (pst-gr3d), *447*
- \pstheader (pstricks), **302**, 303
- \psTilt (pst-3d), **389**, 390, ***391***, 392
- \pstilt (pst-3d), **389**, ***390***, *391, 392*
- \psTilt{30}{\Bar} (pst-3d), 390
- \pstilt{30}{\Bar} (pst-3d), 390
- \pstInterLL (pst-eucl), *426*
- \PstLens (pst-lens), *452*
- \PstLightThreeDGraphic (pst-light3d), *447*
- \PstLightThreeDText (pst-light3d), *447*
- \pstPlanePut (pst-3dplot), *413–415*
- \PstPolygonNode (pst-poly), *431*
- \pstProjection (pst-eucl), *426*
- psTree env. (pst-tree), *366*
- \pstree (pst-tree), ***366***, *367–382*
- \pstree,TC,Toval (pst-tree), *372*
- \pstRelationship (pst-pdgr), *431*
- \pstriangle (pstricks), *233*
- \pstribox
  - (pst-node), 339
  - (pstricks), *271*, ***273***
- pstricks option (pst-pdf), 800
- pstricks package, **213–466**, 797, 800
- PSTricks packages, *see* 3-D parallel projections, *see* 3-D representation, *see specific packages*, *see* arrows, *see* connections, *see* fills, *see* nodes, *see* plotting, *see* sciences, *see* trees
- pstricks-add package, 224, 257, 318, 323, **418–424**
- pstricks.pro file (pstricks), 302, **305**, 307, 365
- pstricks.sty file (pstricks), 215
- pstricks.tex file (pstricks), 214, 215
- \PSTricksfalse (pstricks), 303
- \PSTricksOff (pstricks), 303
- \PSTricksOn (pstricks), 303
- \pstScalePoints (pstricks-add), *421*
- \pstThreeDBox (pst-3dplot), ***404***, *415, 416*
- \pstThreeDCircle (pst-3dplot), *405*
- \pstThreeDCoor (pst-3dplot), ***401***, *402–416*
- \pstThreeDDot (pst-3dplot), ***402***, *403–405*, *411*, *416*
- \pstThreeDEllipse (pst-3dplot), 404, *405*, *412*, *416*
- \pstThreeDLine (pst-3dplot), ***402***, *403*
- \pstThreeDNode (pst-3dplot), 402
- \pstThreeDPut (pst-3dplot), **401**, *402*, 414
- \pstThreeDSphere (pst-3dplot), 405, *406*
- \pstThreeDSquare (pst-3dplot), **403**, *404*
- \pstThreeDTriangle (pst-3dplot), ***403***, *412*
- \PSTtoEPS (pst-eps), *457*
- \pstTriangle (pst-eucl), *426*
- \pstVerb (pstricks), 221, *224*, *234*, 303, **305**, *306*
- \pstverb (pstricks), 280, 303, **305**, 306
- \pstverbscale (pstricks), 221, **305**
- \psunit (pstricks), **218**, 292
- \psverbboxfalse (pstricks), 279
- \psverbboxtrue (pstricks), *279*
- \psvlabel (pst-plot), *318*, ***322***
- \pswedge (pstricks), 237, ***242***, 244
- \psxunit (pstricks), 218
- \psyunit (pstricks), **218**, 222

Q

- \qdisk (pstricks), *224*, ***241***, *268*, 282
- \qline (pstricks), ***232***, 282

R

- R key value
  - (pst-node), 362, 363
  - (pstricks), 270, *271*
- r key value
  - (pst-node), 362, 363
  - (pst-tree), 380
  - (pstricks), 269
- \radians (pstricks), 218, 219
- radius key
  - (pst-node), *338*, 349, ***350***, *351, 352*
  - (pst-tree), *366*, *369–374*, *376*, *379–382*
- railroad diagrams, *345*
- \raisebox, 221
- rand (PostScript), *298*
- RandomFaces key (pst-ob3d), *446*
- rB key value (pstricks), *267*
- rb key value
  - (pst-node), 353
  - (pstricks), *267*
- rbracketlength key (pstricks), **260**, *263*
- rC key value (pstricks), *231*
- \rcoor (pstricks), ***294***, *295*
- \rcurveto (pstricks), *292*
- rcurveto (PostScript), 292
- \readdata
  - (pst-3dplot), 409
  - (pst-plot), ***325***, **328**, *329*
- \readpsbardata (pst-bar), *450*
- real number keys, 312
- rectangles
  - 3-D parallel projections, *404*
  - horizontal, 232, *233*
- \red (pstricks), *216*
- red key value (pstricks), 216
- ref key
  - (pst-node), 349, **353**
  - (pst-tree), 368
- \reflectbox (graphics), 277
- refrigerantBoulles key (pst-labo), *433*
- \relationshipbetween (pst-dbicons), *445*
- relative key value (pstricks), 235, 239
- relative mean power values, *331*
- \resetOptions (pstricks-add), 424
- restricted horizontal Left-Right (LR) mode, 269
- \rlineto (pstricks), *291*

- rlineto (PostScript), **291**, *294*
- \Rnode (pst-node), ***336***, 348, *349*, *359–361*, 363
- \rnode
  - (pst-node), *299*, ***335***, 336, 337, *341–348*, *352, 353*, *355*, *360*, 363, *364*
  - (pst-tree), *374–377*
- rot key (pst-node), 349, **356**, *357*
- \rotate (pstricks), *287*
- rotate (PostScript), *287*
- \rotatebox (graphicx), 277, 397
- Rotatedown env. (pstricks), *277*
- \rotatedown (pstricks), *276*
- Rotateleft env. (pstricks), 277
- \rotateleft (pstricks), *276*
- Rotateright env. (pstricks), 277
- \rotateright (pstricks), *276*
- rotating
  - 3-D objects, *397*, *399*
  - boxes, *276*, 277
  - connection labels, *354*, *357*
  - coordinate system, *410*
  - dot coordinates, *252*
  - fill patterns, *384*
  - graphs, *327*
  - objects, *287*
  - symbols, *252*
  - text, *392*
- rotating package, 392
- rows, matrices, 362, *364*
- rowsep key (pst-node), ***362***, *364, 365*
- \rPERIPH (rrgtrees), *425*
- RPN (Reverse Polish Notation), 329, *430*
- \rput (pstricks), *229–231*, *261*, 266, ***267***, *268*, 269, *271*, *299*, *331*, *341, 342*, *355*, *368*
- rrgtrees package, 424, 425
- \Rrnode (pst-node), *360*
- runit key (pstricks), **218**, 296

S

- \savedata (pst-plot), 328
- saving
  - coordinates, ***288***, 305
  - data records, 328
  - PostScript stack state, *286*
- \sbox, 229
- \scale (pstricks), ***287***, *288*
- scale (PostScript), *287*
- Scalebox env. (pstricks), 277
- \scalebox (graphics), 277
- Scaleboxto env. (pstricks), *277*
- \ScalePoints (pst-plot), 326
- scaling
  - boxes, 276, *277*
  - objects, *287*
- sciences
  - absorption spectra, *432*
  - civil engineering analysis, *436*
  - continuum spectra, *432*
  - electrical circuits, *435*
  - emission spectra, *432*
  - geographical representations, *438*
  - lab apparatus, *433*
  - lenses, *434*
  - linear rays, *434*
  - maps, *438*
  - medical pedigrees, *431*
  - mirrors, *434*
  - optical systems, *434*
  - oscilloscope channels, *434*
- sectors
  - circles, *242*
  - ellipses, *243, 244*
- SegmentColor key (pst-3dplot), *406*
- segmented connections
  - arms, *352*
  - counting, *355*
  - drawing, *342*, *344*
  - maximum number of, 354
- SegmentSymbol key (pst-eucl), *426*
- setcmykcolor (PostScript), *298*
- \setcolor (pstricks), *295*
- setlinejoin (PostScript), ***234***, *294*, 412
- setlinewidth (PostScript), *294*
- sfg package, 442
- \sfgbranch (sfg), *442*
- \sfgcurve (sfg), *442*
- \sfgnode (sfg), *442*
- \sfgtermnod (sfg), *442*
- shading
  - 2-D
    - as highlighting, 239, *240*
    - boxes, *272*
    - custom styles, *289*
    - packages, *388, 389*
  - 3-D, *394*
- shadow key (pstricks), *233*, **235**, 239, *240*, *272–274*, 281, *303*
- shadowangle key (pstricks), *233*, **235**, 239, *240*, *289*, *303*
- shadowcolor key (pstricks), *233*, **235**, 239, *289*, *303*
- shadows
  - as highlighting, 239, *240*
  - boxes, *272*
  - custom styles, *289*
  - packages, *388, 389*
- shadowsize key (pstricks), **235**, 239, *289, 290*, *303*
- shift key (pstricks), *221, 222*
- shortput key (pst-node), *273*, 349, **355**, *356*, *359*
- showbbox key (pst-tree), 370
- showbox key (pst-tree), *378*
- showFP key (tlgc), *311*

showgrid key (pstricks), **222**, *223*
showing, *see* hiding/showing
showorigin key (pst-plot), 315, ***319***, *323*
showpoints key (pstricks), **235**, 237, *238*, *243*, 281, 323, *326, 327*, *330, 331*, *334*, *405*
Sierpinski triangle, 456
\skiplevel (pst-tree), 382
\skiplevels (pst-tree), *382*
skiplevels env. (pst-tree), 382
slanting, *see* tilting
slides (color), overlay specification
  hyperlinks, *797–818*
smooth curves
  Bézier curves, *244, 245*
  overview, 244
  through a list of points, 245, *246*
solid key value (pstricks), 220, 235, *236*, **253**, *255*, *279*, *283*, *285*
solid fills, *254*
SolidAsterisk key value (pstricks), 252
SolidDiamond key value (pstricks), 252
SolidHexagon key value (pstricks), 252
SolidOplus key value (pstricks), 252
SolidOtimes key value (pstricks), 252
SolidPentagon key value (pstricks), 252
SolidSquare key value (pstricks), 252
SolidTriangle key value (pstricks), 252
space
  as fill, *256*
  between commands, 223
  ignoring/preserving, 277, *303*
  inserting, 304
\space
  (pst-tree), 374
  (tex), *304*
\special, 797
  (tex), 214, 280, 292, **302**, 303, 304, 306
special.pro file, 305
\SpecialCoor (pstricks), **219**, ***296***, *298–300*, *302*, 310, 336, 337, 347, *348*, *365*
SpericalCoor key (pst-3dplot), 410
spheres, 3-D, *406*
spherical coordinates, *416*
SphericalCoor key (pst-3dplot), 411, ***416***
spotX key (pst-3dplot), 410, ***413***
spotY key (pst-3dplot), 410, ***413***
spotZ key (pst-3dplot), 410, **413**
Square key value (pstricks), 252
square key value (pstricks), *251*, 252
square* key value (pstricks), 252
squares, 3-D parallel projections, *403*
stack system, 329
startX key (makeplot), *430*
startY key (makeplot), *430*
\State (vaucanson-g), *440*
step functions, *423*
StepType key (pstricks-add), *423*
straight connection line, *341*
string keys, 312
\stroke (pstricks), **284**, *285*
stroke (PostScript), **284**, *294*
stroke, paths, 284, *285*
style key
  (pst-calendar), *452*
  (pst-jtree), *425*
  (pstricks), **229**, *258*, 279
styles
  3-D parallel projections, *414, 415*
  arrows, *295*, 418, *419, 420*
  dots, 251
  fills, 253
  lines
    custom, *282, 283*, *285–291*
    dashed, 235, 236, 240
    dotted, 235, 236, 240, *402*
    fills, *256*
    grid subdivisions, 228
    solid, 235
    width, 228, *256*
  mathematical plots, 332, *333*, 334
  shadows, *289*
  symbols, 251
  symbols, pre-defined, 251
  ticks, *320, 321*
  user-defined
    closed curves, concatenating, *281*
    defining, *279, 280*
    fills, *281*
    lines, *281*
    PostScript output, 280
subgridcolor key (pstricks), 227, *228*
subgriddiv key
  (pst-plot), *332*
  (pstricks), ***227***, *228*
subgriddots key (pstricks), 227, *228*
subgridwidth key (pstricks), 226, **227**, ***228***
successor tree nodes, 367–369
\swapaxes (pstricks), **287**, *288*
swapaxes key (pstricks), ***224***, *232*, 281
swapping axes, *288*
syB key (pst-asr), *424*
symbols
  defining, *250*, 251
  definition, 249, *250*
  in data files, 324
  keywords for, 251
  pre-defined styles, 251
  rotating, *252*
  size, *251*

T

- `tab` key value (pst-node), 349, **355**, 356
- tab characters, 324
- `\tabcolsep` rigid length, 272
- `tablr` key value (pst-node), 349, **355**, 356
- `tabular` env., 272
- `\taput` (pst-node), 356, **358**
- `tbarsize` key (pstricks), **260**, 262, *263*, *352*
- `\tbput` (pst-node), 356, **358**
- `\TC` (pst-tree), *366*, ***367***, *369–371*, *373, 374*, *376*, *378–382*
- `\Tc` (pst-tree), ***367***, *378–382*
- `\TCircle` (pst-tree), 367
- `\Tcircle` (pst-tree), **367**, *371–373*
- `\Tdia` (pst-tree), 367
- `\Tdot` (pst-tree), *367*
- `tensioncolor` key (pst-circ), *435*
- `tensionlabelcolor` key (pst-circ), *435*
- tessellation, *see* tiling
- TEX
  - % (percent sign), comment character, 265
  - getting information from PostScript, 365, 366
- TEX file archives, 810, *see also* CTAN
- TEX files, obtaining
  - web access, 810, 811, *812, 813*, 814
- texdoc program, 815, 816
- texdoctk program, 815–817
- text
  - along a path, *451*
  - rotating, *392*
  - shapes, *448–450*
  - slanting, *392*
- `\text` (amsmath), 361
- `\textcolor`, 216
- `\Tf` (pst-tree), *367*
- `\Tfan` (pst-tree), **368**, *369, 370*
- `THETA` key (pst-vue3d), *445*
- third degree parabola with inverse function, *331*
- `thislevelsep` key (pst-tree), 370, **374**, *376*, *379, 380*
- `thistreefit` key (pst-tree), 370, 372, **373**
- `thistreenodesize` key (pst-tree), 370, 373, ***374***
- `thistreesep` key (pst-tree), 370, ***372***, *379, 380*
- `\thput` (pst-node), **358**, *359*
- three dimensional, *see* 3-D
- `\ThreeDput` (pst-3d), ***393***, *394*, *397*, *399*, *446*
- ticks
  - axes, specifying, *319*
  - axis origin, *316*
  - hiding, *316*
  - length, *321*
  - point of origin, *316*
  - position, *321*
  - size, *322*
  - style, *320, 321*
- `ticks` key (pst-plot), 315, ***319***, *320*
- `ticksize` key (pst-plot), 315, ***321***, *322*
- `tickstyle` key (pst-plot), 315, ***320***, *321, 322*
- `tight` key (pst-tree), 373
- `tightpage` option (pst-pdf), 800
- tiling, 383, *see also* fills
- `tiling` option (pst-fill), 383, 386
- tilting, *390–392*
- `\tlput` (pst-node), 356, **358**
- `\Tn` (pst-tree), *367, 368*
- `tndepth` key (pst-tree), 380, **381**
- `tnheight` key (pst-tree), 380, ***381***
- `tnpos` key (pst-tree), ***380***, *381*
- `tnsep` key (pst-tree), 380, **381**
- `tnyref` key (pst-tree), 380, 381, ***382***
- `\TOP` (rrgtrees), *425*
- `top` key value (pst-plot), 315, **320**
- `\Toval` (pst-tree), *366*, ***367***, *369–380*
- `\Tp` (pst-tree), *367*
- `tpos` key
  - (pst-node), 349, **356**
  - (pst-tree), *378*
- `\TR` (pst-tree), ***367, 368***, *369*, *374*, *377*
- `\Tr` (pst-tree), ***367, 368***, *374–377*
- transforms, *see specific transforms*
- `\translate` (pstricks), ***286***, *287–290*
- `translate` (PostScript), 286
- transparency, 257, 258
- `TransparentMagenta` key value (tlgc), *279*
- `\transy` (pst-calendar), *453*
- `treefit` key (pst-tree), 370, 372
- `treeflip` key (pst-tree), 370, **371**, *372*
- `treemode` key (pst-tree), *367*, 370, ***371***, *372*, *374–377*, *379, 380*, *382*
- `treenodesize` key (pst-tree), *367*, 370, **373**, *374*
- trees
  - general syntax, *366*
  - nodes
    - blank spaces, inserting, *369*
    - bounding boxes, *378*
    - command names, 367
    - curved connectors, 369, *376*
    - diagonal connectors, *377*
    - distance between, *372–376*
    - fanned, *369*
    - keywords for, *370–378*
    - level separation, *375, 376*
    - macros, assigned to edges, *377*
    - nil, *368*
    - order, changing, *371*
    - predecessors, 367, 369
    - reference points, setting, *368*
    - reserving space for, *368*
    - sets of branches, combining, *370*
    - successors, 367–369
    - tree direction, specifying, *371*
    - types, *367*

trees *(cont.)*
nodes, labels
aligning, *379*
alignment, *381, 382*
creating, 379
examples of, *380*
positioning, *378*
separation, *381*
skipping levels, *382*
`treesep` key (pst-tree), *369*, 370, **372**, *373*, *380–382*
`\Tri` (pst-tree), *367*
`tri` key value (pst-node), 362, 363
`Triangle` key value (pstricks), 252
`triangle` key value (pstricks), *251, 252*
`triangle*` key value (pstricks), 252, *298*
triangles, 3-D parallel projections, *403*
triangular frames, *271*, *273*
triangular nodes, *339*
`trimode` key
(pst-node), *339*
(pstricks), **270**, *271*, *273*
`\trinode` (pst-node), ***339***, 363
`\trput` (pst-node), 356, **358**
`Tshadowangle` key (pst-3d), **388**, *389*
`Tshadowcolor` key (pst-3d), *388*, ***389***, *390, 391*
`Tshadowsize` key (pst-3d), *388*, ***389***
`\tspace` (pst-tree), *369*
`\Ttri` (pst-tree), 367
TUG home page, 810, *811*
`\tvput` (pst-node), 358
`tx@NodeDict` (PostScript), 365
`type` key (pst-fractal), *456*

U

`U` key value (pstricks), 270, *271*
`u` key value (pstricks), 269
`ul` key value (pstricks), 269
uml package, 443
UML diagrams, *442–445*
`\umlArgument` (uml), *443*
`\umlAttribute` (uml), *443*
`\umlClass` (pst-uml), *442*
`\umlSchema` (uml), *443*
`\umlSubClass` (uml), *443*
`unit` key (pstricks), ***218***, 262, *269*
`\uput` (pstricks), *224*, *230, 231*, ***268***, *300*, *320*, *331*, *333*
`ur` key value (pstricks), 269
`\usebox`, *229–231*
`\usepackage`, *215*

V

vaucanson-g package, 439, 440
`VCPicture` env. (vaucanson-g), *440*
`\verb`, 277, 279
`verbatim` env., 277
verbatim boxes, *278*, 279
vertical lines as fills, *254*
view angle, 3-D objects, *397*
`viewangle` key (pst-3d), 395, **397**, 399
`viewpoint` key (pst-3d), *393, 394*, **395**, 396, *397*, 398, *399*
viewpoint, 3-D objects, 395, *396, 397*
views (3-D), order of, 397
`visibleLineStyle` key (pst-3dplot), 410, **415**
`vlines` key (pstricks), *392*
`vlines` key value (pstricks), **253**, *254*, 255, *256*, *279*, *281*, *285*
`vlines*` key value (pstricks), **253**, *254*, 255
`vref` key
(pst-node), **348**, 349, *360*
(pst-tree), 381
`\vspace` (pst-tree), 366
VTeX program, 365, 797

W

watermarks, *326*
`\wd` (tex), *229–231*
wget program, 814
`white` key value (pstricks), 216, 235
whitespace, *see* space
`\wire` (pst-circ), *435*
`\WORD` (rrgtrees), *425*
`\WorldMap` (pst-geo), *438*
`\write` (tex), 304
writing objects into files, on the fly, *457*

X

`x` key value
(pst-plot), 315, 318, 319
(pstricks), 252
`xAxisLabel` key (pstricks-add), *421*
`xAxisLabelPos` key (pstricks-add), *421*
`xbbd` key (pst-tree), 370, 378
`xbbh` key (pst-tree), 370, 378
`xbbl` key (pst-tree), 370, *378*
`xbbr` key (pst-tree), 370, *378–380*
xcolor package, 215, 216, 235, 258, 304, 406
`xEnd` key (pstricks-add), 418
xetex program, 798, 803
`xgap` key (pst-asr), *424*
xkeyval package, **217**, 310
`xLines` key value (pst-3dplot), 414
`xMax` key (pst-3dplot), *401*, *410*, ***411***
`xMin` key (pst-3dplot), *401*, *410*, ***411***
`Xnodesep` key (pst-node), 297, *300*, 349, 350, **351**
`XnodesepA` key (pst-node), 349, 350
`XnodesepB` key (pst-node), 349
xpdf program, 804
`xPlotPoints` key (pst-3dplot), *407, 408*
`xPlotpoints` key (pst-3dplot), **406**, 410, ***411***, *415*

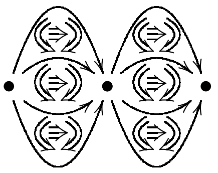
`xStart` key (pstricks-add), 418
`xStep` key (pstricks-add), 418
`xThreeDunit` key (pst-3dplot), 410, **411**
`xunit` key (pstricks), ***218***, *224*, 227, 296, *298*, *323*
`xWidth` key (pst-fractal), *456*
`xyAxes` key (pstricks-add), 418
`xyDecimals` key (pstricks-add), 418
`xyLines` key value (pst-3dplot), 414

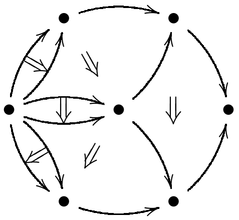
Y

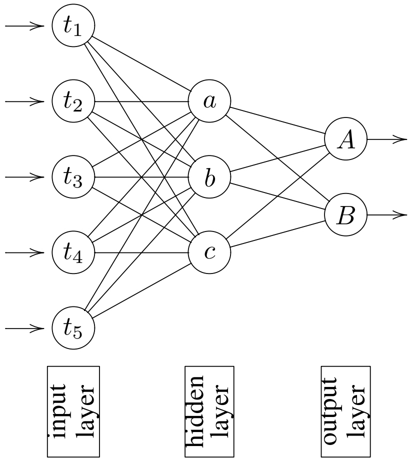
`y` key (pst-plot), 319
`y` key value (pst-plot), 315, 318, 319
`yAxisLabel` key (pstricks-add), *421*
`yAxisLabelPos` key (pstricks-add), *421*
`Year` key (pst-calendar), *452*
`yellow` key value (pstricks), 216
`yEnd` key (pstricks-add), 418
`yLines` key value (pst-3dplot), 414
`yMax` key (pst-3dplot), *401*, 410, **411**
`yMin` key (pst-3dplot), *401*, *410*, ***411***
`Ynodesep` key (pst-node), 297, *300*, 349, 350, ***351***, 361
`YnodesepA` key (pst-node), 349, *361*
`YnodesepB` key (pst-node), 349
`yPlotpoints` key (pst-3dplot), **406**, *407*, 410, ***411***, *415*
`yStart` key (pstricks-add), 418
`yThreeDunit` key (pst-3dplot), 410, **411**
`yunit` key (pstricks), ***218***, *224*, 227, 296, *298*, *306*
`ywidth` key (pst-fractal), *456*
`yxLines` key value (pst-3dplot), 414

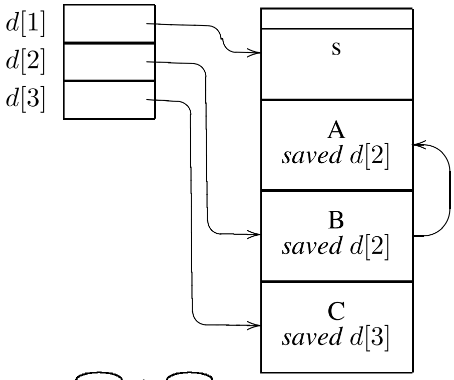
Z

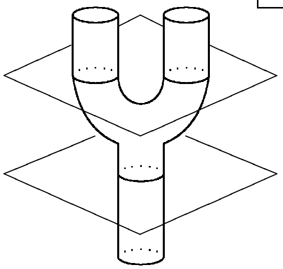
zigzag lines, *455*
zlib program, 799
`zMax` key (pst-3dplot), *401*, *410*, ***411***
`zMin` key (pst-3dplot), *401*, 410, **411**
`zThreeDunit` key (pst-3dplot), 410, **411**



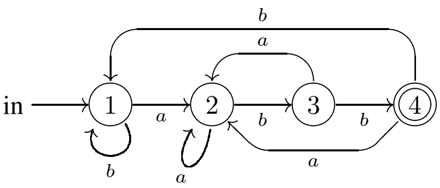


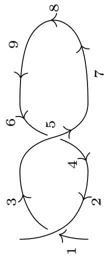






$$\begin{tikzcd}
 & \Sigma^L \arrow[rrr, "\varphi^r"] \arrow[ddd, "\varphi^m"'] & & & \Sigma^R \arrow[ddd, "\varphi^{m^*}"] \\
L \arrow[ur, "\lambda^L"] & & L_r \arrow[ll, hook', "i_1"'] \arrow[r, "r"] & R \arrow[ur, "\lambda^R"'] & \\
L_m \arrow[u, "i_2"] \arrow[dd, "m"'] & & K_{r,m} \arrow[ll, hook', "i_3"'] \arrow[u, "i_4"'] \arrow[r, "r"] \arrow[dd, "m"] & R_{m^*} \arrow[u, "i_6"'] \arrow[dd, "m^*"] & \\
 & \Sigma^G \arrow[rrr, "\varphi^{r^*}"'] & & & \Sigma^H \\
G \arrow[ur, "\lambda^G"] & & G_{r^*} \arrow[ll, hook', "i_5"'] \arrow[r, "r^*"'] & H \arrow[ur, dotted, "\lambda^H"'] &
\end{tikzcd}$$





Xy-pic

Symbols

! syntax, ***472***, *473*, **488, 489**, **494**
" syntax, 494
"..." syntax, 470
’ syntax, **480**, 482, **494**
(..,..) syntax, 470
(0.*x*x) syntax, 482
* syntax, *468–470*, ***471***, *472, 473*, *475*, 476–478, 481, 488, 503
** syntax, ***470***, *471*, 472, *475, 476*, 477, 498
+ syntax, *468*, ***471***, ***473***, *475*
++ syntax, **471**, *473*, *485*
+= syntax, *473*
, syntax, 469
− syntax, ***473***, 480, **487**
- syntax, *473*
/.../ syntax, 472
/^.../ syntax, 472
/_.../ syntax, 472
/d.../ syntax, *485*
/l.../ syntax, *486*, **487**
/r.../ syntax, **470**, *486*
/u.../ syntax, *485*
: syntax, 470, 487
; syntax, **470**, 477, 479
< syntax, **471**, **504**, 505
<..,..> syntax, 469
<<<< syntax, 482
= syntax, **470**, ***473***, 479, 494
> syntax, **471**, **504**, 505
? syntax, **471**, *475*, **476**
?! syntax, *471*
?< syntax, *471*
?>>> syntax, *471*
[F] syntax, *468*, 469, *471–473*, 474, **478**, ***479***, *485, 486*, *488*, *500*
[o] syntax, ***471***, *473*, *475*, ***479***, *485*, *488*, *499*
& syntax, 468, 475, **481**, 487
\⟨*c*⟩cross, 504
\⟨*c*⟩twist, 504
\⟨*cc*⟩compositemap, 493
\⟨*cc*⟩lowertwocell, 493
\⟨*cc*⟩twocell, 493
\⟨*cc*⟩uppertwocell, 493
^ syntax, **478**, 480, 494, 502, 506
~ syntax, 495, 507, 509
~* syntax, ***476***, *496, 497*, *499*, *507, 508*
~** syntax, *476*
~: syntax, **497**, *498, 499*, ***507***, *508*
~< syntax, ***496***, *498, 499*
~<> syntax, **496**, *497–499*
~<< syntax, 496, *497*
~= syntax, *496*, *499*, **500**, *508*
~> syntax, ***496***, *497*, *499*, *507, 508*
~>< syntax, **496**, *497*
~>> syntax, **496**, *497*
\\, 468, **473**, **481**
_ syntax, **478**, 480, 494, 506
‘ syntax, **480**, 482, 490, **494**
| syntax, 480, **504**, 505
0 syntax, 470, 478
1 syntax, 478

- 2 syntax, 478
- 2cell option, 493
- 3 syntax, 478

@

- @ syntax, **472**, 478
- @*[F] syntax, *486*, **487**
- @*[r] syntax, *481*, **482**
- @{*} syntax, ***472***, *473*
- @{+} syntax, ***472***, *475*, 476
- @{-} syntax, **470**, *471*, *500, 501*
- @{-} syntax, ***470***, *471*, *488*, *490*, *498, 499*
- @{.} syntax, **470**, *471*, *498, 499*
- @{<<} syntax, *471*
- @{==} syntax, *476*
- @{=} syntax, ***470***, *497*
- @{>} syntax, *471*
- @{o} syntax, *472*
- @{x} syntax, *472*
- @‘{...} syntax, ***479***, *508*
- @H syntax, *486*, **487**
- @M syntax, *486*
- @R syntax, *486*
- @W syntax, *486*, **487**

A

- Adobe Reader program, 817
- all option, 468, 478
- amsmath package, 483, 484
- \ar, *468*, *472*, **478**, *479–481*, *485, 486*, *488*, *494, 495*, *500–503*
- arc option, 500
- arcs, 501, *502*
- arrow option, 468, **478**, 479, **480, 481**, 487, **495**, **503**
- arrows
  - custom, 478, *479, 480*
  - in commutative diagrams, *481–484*

B

- braids, *509*

C

- C syntax, 472
- category theory, *509*
- circles, *500*, 501
- \circuit (private), **489**, *490*
- CMacTeX program, 468
- cobordism of Morse theory, *510*
- color option, 468, **474**
- commutative diagrams
  - 3 x 2 diagrams, *484*
  - 3 x 3 diagrams, *484*
  - annotations, *483*
  - cubical, *481*
  - description, 481
- commutative diagrams *(cont.)*
  - pullbacks, *484*
  - square, *482, 483*
  - triangular, *483*
- Comprehensive TEX Archive Network, *see* CTAN
- connections, *470, 471*
- \croplattice, *503*
- crossings
  - knots, *504, 505*
  - links, *504, 505*
- \crv, ***475***, *476*
- CTAN (Comprehensive TEX Archive Network)
  - archived files, finding and transferring, 813
  - description, 810
  - files, from the command line, *814*
  - TEX file catalogue, 811
  - web access, 810, 811, *812, 813*, 814
- curly braces ({ }), *477*
- curve option, 468, **474**, 475, 479, **500**, **503**
- curves, *475, 476*

D

- D syntax, 472
- diagram package, 482
- diagxy package, 482
- documentation, *see also* online resources
  - command-line interface, *815*
  - panel interface, *816*
  - search by name, *815*
  - search by product, *816*
  - texdoc, *815*
  - texdock, *816*
- drawing
  - arcs, 501, *502*
  - arrows
    - custom, 478, *479, 480*
    - in commutative diagrams, *481–484*
  - braces, *477*
  - brackets, 476, *477*, 478
  - braids, *509*
  - category theory, *509*
  - circles, *500*, 501
  - cobordism of Morse theory, *510*
  - connections, *470, 471*
  - constructing pictures, *468*
  - curves, *475, 476*
  - ellipses, 500, *501*
  - extensions, 468
  - features, 468
  - frames, 476, *477*, 478
  - globular 3-morphisms, *509*
  - graphic notions, 467
  - graphs
    - basic principle, *487*
    - hidden layers, 489

- drawing *(cont.)*
  - input layers, 489
  - linguistics trees, *491, 492*
  - logical circuit diagrams, 489, *490*
  - neural network diagrams, *488*, 489
  - output layers, 489
  - tree branching, *488*
  - kernel, 467
  - knots
    - crossings, *504, 505*
    - joins, *505–508*, 509
  - lattices, *502, 503*
  - links
    - crossings, *504, 505*
    - joins, *505–508*, 509
  - matrix-like diagrams
    - 3 x 2, *484*
    - 3 x 3, *484*
    - annotations, *483*
    - command syntax, 480
    - commutative diagrams, *481–484*
    - finite state diagrams, *485*, 486, 487
    - homology, *484*
    - pullback effect, *484*
    - square, *482, 483*
    - stack diagrams, 485, *486*, 487
  - modules, 468
  - object margins, 473
  - objects
    - bounding box, 473
    - definition, 468
    - dropping, 471, *472*, 473
    - edge, 473
    - shifting, 472
    - sizing, 473
  - options, 468
  - pentagonal sphere, *510*
  - polygons
    - 3-D, *498*
    - cubes, *499*
    - general form, 495
    - hexagons, *496, 497*
    - nesting, *499*
    - perspective drawings, *498*
  - positions
    - absolute, 469
    - definition, 467
    - initial, 469
    - specifying, *469, 470*
  - spline curves, *475, 476*
  - string diagram, *510*
  - text, in pictures, *473*
  - two-cell diagrams, *493–495*
  - web structures, 502, 503
- `\drop`, *502*, ***503***

E

- `\ellipse`, *490*, ***500***, *501, 502*
- ellipses, 500, *501*
- `\endxy`, **469**, 479
- `\entrymodifiers`, *485*

F

- FAQs (Frequently Asked Questions), 809, *see also* online resources
- `frame` option, 468, **474**, **476**, 477, 479
- frames, 476, *477*, 478
- Frequently Asked Questions (FAQs), *see* online resources
- `\frm`, *472*, **476**, *477*, **478**, *507*

G

- globular 3-morphisms, *509*
- `graph` option, 468, **487, 488**, 506
- graphs
  - basic principle, *487*
  - hidden layers, 489
  - input layers, 489
  - linguistics trees, *491, 492*
  - logical circuit diagrams, 489, *490*
  - neural network diagrams, *488*, 489
  - output layers, 489
  - tree branching, *488*

H

- `\hcap`, *506*
- help, *see* online resources
- hidden graph layers, 489
- How To Ask Questions The Smart Way, 810
- hyperlinks, slides, *809–818*

I

- ifthen package, 503
- `\ifthenelse` (ifthen), *503*
- `\iiixii` (diagxy), *484*
- `\iiixiii` (diagxy), *484*
- `import` option, 474
- input graph layers, 489

J

- joins
  - knots, *505–508*, 509
  - links, *505–508*, 509

K

- kernel, 467
- `knot` option, 478, **503**
- `\knotholesize`, *507*, ***508***

- knots
  - crossings, *504, 505*
  - joins, *505–508*, 509

L

- `L` syntax, 472
- `\labelstyle`, ***494***, *504–508*
- LaTeX files, obtaining
  - web access, 810, 811, *812, 813*, 814
- `\latticeA`, *503*
- `\latticeB`, *503*
- `\latticebody`, *502*, ***503***
- lattices, *502, 503*
- `\latticeX`, 503
- `\latticeY`, 503
- `Lc` syntax, 476
- `line` option, 468, **474**
- linguistics trees, *491, 492*
- links
  - crossings, *504, 505*
  - joins, *505–508*, 509
- logical circuit diagrams, 489, *490*

M

- `matrix` option, 468, 478, **480, 481**, **487**
- matrix-like diagrams
  - 3 x 2, *484*
  - 3 x 3, *484*
  - annotations, *483*
  - command syntax, 480
  - commutative diagrams, *481–484*
  - finite state diagrams, *485*, 486, 487
  - homology, *484*
  - pullback effect, *484*
  - square, *482, 483*
  - stack diagrams, 485, *486*, 487
- `\morphism` (diagxy), *482, 483*

N

- nesting, polygons, *499*
- neural network diagrams, *488*, 489
- `\newdir`, **470**, *481*, 482
- `\newgraphescape`, *488*, **489**, *490*
- news groups, 810, *see also* online resources

O

- object margins, 473
- `\objectmargin` rigid length, 496
- objects
  - bounding box, 473
  - definition, 468
  - dropping, 471, *472*, 473
  - edge, 473
  - shifting, 472
  - sizing, 473
- `\objectstyle`, ***494***, *497*, *499*, *504*, *507, 508*
- `\omit`, *493, 494*, ***495***
- online access to CTAN, 810, 811, *812, 813*, 814
- online resources
  - archived files, finding and transferring, 813
  - CTAN (Comprehensive TeX Archive Network), 810
    - web access, 810, 811, *812*, 813, 814
  - documentation
    - command-line interface, *815*
    - panel interface, *816*
    - search by name, *815*
    - search by product, *816*
    - `texdoc`, *815*
    - `texdock`, *816*
  - FAQs (Frequently Asked Questions), 809
  - files, getting from the command line, *814*
  - How To Ask Questions The Smart Way, 810
  - news groups, 810
  - program files, obtaining
    - web access, 810, 811, *812*, 813, 814
  - TeX file catalogue, 811
  - TeX files, 810
  - TeX user groups, 817, 818
  - TUG home page, 810, *811*
- output graph layers, 489

P

- pentagonal sphere, *510*
- pic program, 487
- `\place` (diagxy), *483*
- `poly` option, **495**, 507
- polygons
  - 3-D, *498*
  - cubes, *499*
  - general form, 495
  - hexagons, *496, 497*
  - nesting, *499*
  - perspective drawings, *498*
- `\POS`, **480**, *486*, *488*, *490*
- positioning
  - absolute, 469
  - definition, 467
  - initial, 469
  - specifying, *469, 470*
- program files, obtaining
  - web access, 810, 811, *812, 813*, 814
- `ps` option, 475
- `\pullback` (diagxy), *484*

R

- `R` syntax, 472
- `\restore`, ***486***, 487, *488*, *490*, *507*
- `rotate` option, 468, **474**, 489
- `\rrtwocell`, *494, 495*

- `\rtwocell`, *493*, 494

S

- `\save`, ***486***, 487, *488*, *490*, *507*
  - slides (color), overlay specification
    - hyperlinks, *809–818*
- `\SloppyCurves`, 476
  - spline curves, *475, 476*
- `\splinetolerance`, 476
- `\Square` (diagxy), *483*
- `\square` (diagxy), *482, 483*
  - square brackets (`[]`), 476, *477*, 478
  - string diagram, *510*

T

- TEX file archives, 810, *see also* CTAN
- TEX files, obtaining
  - web access, 810, 811, *812, 813*, 814
- texdoc program, 815, 816
- texdoctk program, 815–817
- text, in pictures, *473*
- `tile` option, 474
- `tips` option, 468, **474**, **481**
- `\Tree` (xyling), *491*
  - tree branching, *488*
  - TUG home page, 810, *811*
- `\turnradius`, 487
  - two-cell diagrams, *493–495*
- `\twoar` (diagxy), *483*
- `\txt`, *473*

U

- U syntax, 472
- `\UseAllTwocells`, ***493***, *494*
- `\UseCompositeMaps`, 493
- `\UseHalfTwocells`, 493
- `\UseTwocells`, ***493***, *495*

V

- `\vcap`, *507*
- `\vcross`, *504*
- `\vloop`, *508*, **509**
- `\vover`, *504*, ***506***, *508*
- `\Vtrianglepair` (diagxy), *483*
- `\vtwist`, *504*
- `\vunder`, *504*

W

- `web` option, 502
- web structures, 502, 503
- wget program, 814

X

- `\xoverv`, *508*
- `\xtwocell`, 493, ***494***, *495*
- `\xunderv`, ***507***, *508*, 509
- `\xy`, 469
  - `xy` env., **469**, 495
- `\xybox`, ***497***, *502, 503*
- `\xyconnect` (xytree), *492*
- `\xygraph`, ***487***, *488*, 489, *490*, *506–508*
- `\xylattice`, *502*
  - xyling package, 491
- `\xymatrix`, ***468***, **480**, *481*, 482, *485, 486*, *493–495*
- `\xynode` (xytree), 491, *492*
- `\xyoption`, 468
- `\xypolygon`, **495**, *496–499*, 507
- `\xypolyname`, *499*
- `\xypolynode`, ***497***, *499*, *507, 508*
- `\xypolynum`, *497*
- `\xytree` (xytree), *492*
  - xytree package, 491

Y

- `\yynode` (xytree), 491, *492*
- `\yytree` (xytree), *492*

People

Abraham, Paul, 709
Akhmadeeva, Leila, 431
Aplevich, Dwight, 203, 583
Apollonius, 192, 194
Appelt, Wolfgang, 668
Arnold, Doug, 491

Bächle, Dirk, 687
Barnard, Frederick R., 1
Barr, Michael, 482
Bauke, Heiko, 518
Beccari, Claudio, 47
Beitz, Eric, xxxiv, 547, 551
Berners-Lee, Tim, 12
Berry, Karl, 69
Bibby, Duane, 7
Bleser, Joachim, 15
Bolek, Piotr, 148
Bos, Victor, 691
Braams, Johannes, 15
Brown, Terry, 16
Buckley, Andy, 512, 516, 560
Burton, Terry, 453
Bustamante Argañaraz, Gustavo S., 196, 576

Carlisle, David, 7, 47, 557, 719, 737
Charpentier, Jean-Côme, 429
Cho, Jin-Hwan, 798
Cholewo, Tomasz, 203
Chupin, Maxime, III
Clark, Adrian, 8
Clark, James, 17
Coulon, Jean-Pierre, 588
Coxeter, Harold Scott MacDonald, 192

Díaz, José Luis, 64, 196
Dahlgren, Mats, 517
Deutsch, L. Peter, 11
Diamantini, Maurice, 442
Dirr, Ulrich, xxxiv, 673
Duggan, Angus, 7
Dunker, Rainer, 647, 659
Dupuis, Étienne, 691

Edwards, Tim, 586
Egler, Andreas, 589
Ekola, Tommy, 188
Els, Danie, 513
Esser, Thomas, 815, 816

Fairbairns, Robin, 809, 810
Finston, Laurence D., 211, 212
Fischer, Ulrike, xxxiv, 668, 669
Frampton, John, 424, 425
Fraser, James, III
Frischauf, Adrian, 13
Fujita, Shinsaku, 520

Gäßlein, Hubert, xxxiv, 43, 457
Gabo, Naum, 57, 58
Garcia, Federico, xxxiv, 666, 668, 680
Gardner, D. J., 424
Gastin, Paul, 15, 438
Geisler, Martin, 194
Gheorghieş, Ovidiu, 181
Giese, Martin, 449
Gieseking, Martin, 13
Gilg, Jürgen, xxiv
Girou, Denis, 214, 431, 446, 447, 452, 457
Gjelstad, Ellef, 443
Gonzato, Guido, 609
Gray, Norman, 555
Gregorio, Enrico, 612
Gurari, Eitan M., 15

Hàn, Thế Thành, 24, 798
Haas, Roswitha T., 518
Hafner, Jim, 719
Hagen, Hans, 73, 138, 520, 541
Hamilton Kelly, Brian, 702
Happel, Patrick, 513
Hefferon, Jim, 810
Heldoorn, Marcel, 513
Hilbert, David, 52, 194
Hirata, Shunsaku, 798
Hobby, John, 21, 71, 75, 80, 157
Hoenig, Alan, 52, 56
Hoffmann, Torben, 668, 673
Hwang, Andrew D., 20

Jackowski, Bogusław, 138, 149
Jalbert, François, 589
Jeffrey, Alan, 65
Jorssen, Christophe, 428, 429, 434, 435
Jørgensen, Palle, 155

Kane, Kevin C., 518
Kelley, Colin, 17
Kern, Uwe, xxxiv, 719
Kernighan, Brian, 17
Kiffe, Thomas, 468
Kinch, Richard, 24
Kneifl, Stanislav, 636
Knuth, Donald, 6–9, 51, 137, 698
Koch, Helge von, 105, 194
Kołodziejska, Hanna, 691
Krysztofiak, Claudia, xxxiv

Lamers, Jürgen, 687
Lamport, Leslie, 7, 8
Lauda, Aaron, xxxiv, 509
Laurie, Dirk, 590, 616, 647, 651, 659
Leathrum, Thomas E., 122
Leech O'Neale, Susan, xxxiv
Leilich, Jens, 572
Lesenko, Sergey, 24
Lester, Paul Martin, 1
Levine, Michael, 555
Lindenmayer, Aristid, 154
Lombardy, Sylvain, 439
Luecking, Daniel H., 73, 122
Luque, Manuel, 433, 434, 437, 445, 452

Maclaine-cross, Ian, 15, 47
Matarazzo, Giuseppe, 436, 437
Mattes, Eberhard, 24
May, Ludwig, 572
May, Wolfgang, 445
Milne, James, 481
Mitchell, Ross, 589
Mittelbach, Frank, 7, 688
Moon, Alun, 148
Moore, Ross, xxxiv, 16, 467, 488
Morawski, Jens-Uwe, 59, 60, 64, 170
Morimoto, Hiroaki, 637
Muelas, Santiago, 142, 209

Navarria, Janice, xxxiv
Neugebauer, Gerd, 702, 704
Newton, Isaac, 714
Nienhuys, Han-Wen, xxxiv, 661
Niepraschk, Rolf, 43, 457
Nieuwenhuizen, Jan, 661
Nobre Gonçalves, Luís, 209

Ohl, Thorsten, 120, 555, 561, 566
Oswald, Urs, 194
Otten, A. F., 520, 541

Phan, Anthony, II, 66, 150, 209
Pianowski, Piotr, 138
Pipping, Nils Johan, 193
Podar, Sunil, 15
Poulain, Christophe, 148, 192

Rahtz, Sebastian, 7, 42
Ramek, Michael, 518
Raymond, Eric, 810
Reichert, Axel, 513
Richer, Jacques, 688
Richter, Jörg, 696
Ristow, Alan, 450
Rodriguez, Dominique, 423, 426
Roegel, Denis, 80, 207, 208
Rokicki, Tom, 11, 24, 65
Rose, Kristoffer H., 16, 467
Rowley, Chris, 7
Rubinstein, Zalman, 668
Ruedas, Thomas, 816
Ryćko, Marek, 138

Sabo, Rudolf, 13
Sakarovitch, Jacques, 439
Sarlat, Jean-Michel, IV, 195
Schöpf, Rainer, 810
Scherer, Andreas, 167
Schmid, Hanspeter, 442
Schmittbuhl, Arnaud, 432
Schnell, Andreas, 14
Schofer, Angelika, 589
Sendoukas, Hippocrates, 24
Sierpiński, Wacław, 52, 194
Simons, Don, 590, 616, 618
Smith, Brian, 13
Sowa, Friedhelm, 7
Steinbach, Andrea, 589

Tannert, Sebastian, 576
Taupin, Daniel, v, vi, 589, 591, 592
Tidefelt, Henrik, 177
Tille, Andreas, 576
Tobin, Geoffrey, 122
Tutelaers, Piet, 668

Un, Koaungli, 491

van der Laan, Kees, 57, 58, 147, 699, 701
Van Zandt, Timothy, 214, 448, 451, 455, 458
Verhulst, Ferdinand, 195
Vermaseren, Jos, 555, 558
Veytsman, Boris, 431
Vieth, Ulrik, 67, 137, 167
Vila-Forcen, Jose-Emilio, 430
Voß, Herbert, 214, 434, 435, 437, 453
Vogel, Ralf, xxxiv, 491
Vulis, Michael, 11, 797

Walshaw, Chris, 600, 654
Wanske, Helene, 587
Weinhold, Stephan, 688
White, Jan, 742
Wichura, Michael, 13
Wicks, Mark A., 24, 798
Williams, Graham, 811
Williams, Thomas, 17
Wilson, Peter, 178, 181, 710
Wyart, Damien, xxxiv
Wythoff, Willem Abraham, 192

Yang, Yang, 167
Young, Thomas, 714