Chapter 13
TEX and LATEX

TEX can print virtually any mathematical thought that comes into your head, and print it beautifully.

— HERBERT S. WILF, *TEX: A Non-review* (1986)

I can't compose at a typewriter.
I can't even compose a letter to my relatives on a typewriter,
even though I'm a good typist.

— DONALD E. KNUTH, in *Mathematical People: Profiles and Interviews* (1985)

1992: Extensive testing shows that 98.3% of the time
no matter which of the `[h]`, `[t]`, `[b]`, *or* `[p]` *options is used,*
LATEX will put your `table` *at the end of the document.*

— DAVID F. GRIFFITHS and DESMOND J. HIGHAM, *Great Moments in LATEX History*[20] (1997)

If you don't find it in the index,
look very carefully through the entire catalogue.

— SEARS, ROEBUCK and CO., *Consumer's Guide* (1897)

[20] In [118].

Wordprocessing systems have brought a new dimension to writing. Courses are offered on writing with a computer, and books have been written on the subject [205], [302]. Producing mathematical equations is difficult or impossible with wordprocessors not designed for the purpose, but various specially designed systems are available. Because it is ubiquitous in the mathematical and computer science communities I focus in this chapter on the TEX typesetting system (the "EX" is usually pronounced like the *eck* in *deck*) and the LATEX macro package. The `troff` system (with the `eqn` preprocessor for typesetting mathematics [151]) predates TEX and was widely used on Unix systems, but has now largely been supplanted by TEX. My purpose is not to describe or teach TEX and LATEX or the associated programs BIBTEX and MakeIndex (see the references in §13.5 for suggested texts) but to explain their advantages and to give hints on their use.

13.1. What are TEX and LATEX?

TEX was developed by Donald Knuth at Stanford University, beginning in the late 1970s [161]. Knuth originally designed TEX to help him typeset books in his *Art of Computer Programming* series. It is a program that accepts a text file interspersed with formatting commands and translates it into a `dvi` (device independent) file that can readily be translated into the languages understood by output devices such as screens and laser printers. The typical usage is to prepare an ASCII file using your favourite editor, run it through TEX and preview the output on the screen. You correct errors by editing the source file and then re-running TEX. When you wish to obtain hard copy, you invoke a driver to render the TEX output on the printer.

LATEX [172] by Leslie Lamport is a macro package that "sits on top" of TEX. It was originally released in 1985 and rapidly became the package of choice for TEX-using authors in computer science, mathematics, and related areas. I recommend using LATEX rather than TEX, since it provides so many useful features that would have to be programmed if you used TEX. The original version of LATEX, LATEX 2.09, was extended in various ways by many people as its popularity grew, and so suffered from incompatible dialects. This problem was solved by LATEX 2_ε, which in 1994 replaced LATEX 2.09 as the official version of LATEX. It is supported and maintained (with a new release every six months) by volunteers in the LATEX3 Project, a long term project to produce a completely re-designed and re-implemented version of LATEX. In the remainder of this chapter, "LATEX" refers to LATEX 2_ε.

Another macro package for TEX called AMS-TEX [251], by Michael Spivak, was designed specifically for typesetting mathematics. It introduced, among other features, special fonts, flexible structures for multiline displayed equations, and facilities for producing commutative diagrams. It was developed for use by the publications of the American Mathematical Society (AMS). AMS-LaTeX from the AMS [8] provides a package amstex for LaTeX that incorporates the features of AMS-TeX.

There are several advantages to preparing your papers in some form of TeX.

1. TeX is available on many computer systems, from PCs to workstations to mainframes, and output can be obtained on devices ranging from screens to dot matrix and laser printers to professional typesetting machines.

2. Many journals now accept papers in TeX format; this saves the usual typesetting stage and its inevitable introduction of errors.

3. In LaTeX and AMS-TeX the format of the document is governed by style options, which are specified at the beginning of the source. Only the style options need to be changed to alter the format of the document. Available formats range from book to technical report to SIAM or AMS journal paper.

4. TeX source is plain ASCII text so it can readily be sent by electronic mail or ftp'ed. This is convenient when collaborating with a distant co-author or submitting a final paper in TeX format to a journal.

5. TeX's `dvi` files, or their PostScript translations, are relatively portable means for distributing papers electronically. For example, you can make your papers available from a Web page or by anonymous ftp (see §14.1). (Making the `tex` file available is less satisfactory because it is often inconvenient to collect all the source into one file, particularly if LaTeX is used with nonstandard style files.)

6. Cross-references to equations, theorems, tables, etc. and citations of bibliographic references are fully supported using symbolic labels in LaTeX. Renumbering of the cross-references is handled automatically when a new equation or bibliographic entry is added.

7. It is easy to prepare transparencies with TeX if the paper on which the talk is based is already in TeX form, as the conversion largely consists of deleting chunks of text and equations. See §10.3.

13.2. Tips for Using LaTeX

In this section I offer some suggestions on the use of LaTeX, including some finer points not often discussed in textbooks (many of the suggestions apply to TeX and AMS-TeX too). Appendix B summarizes the TeX and LaTeX symbols and spacing commands.

Dashes

There are three types of dashes in typesetting. A hyphen is produced in TeX by typing `-`, as in state-of-the-art computer. Number ranges use the en-dash, typed in TeX as `--` rather than a single dash. This gives 1975–1978 and pp. 12–35 instead of 1975-1978 and pp. 12-35. The en-dash is also used when the first part of a compound word does not modify the meaning of the second part; it can be thought of as standing for *and* or *to*. Examples: Gauss–Chebyshev quadrature, Moore–Penrose inverse, left–right evaluation. An exception is Runge-Kutta method, for which most authors use a hyphen. A dash partitioning a sentence—like this—should be typed as `---` with no space on either side; this is the em-dash. A mathematical minus sign is produced by a single `-` within mathematics mode. Compare -3, typed as `-3`, with -3, typed as `$-3$`.

Delimiters

Be wary of the `\left` and `\right` commands for automatically sizing delimiters, as they often produce symbols that are too large (sometimes depending on taste). One of the `\big`, `\Big` or `\bigg` commands (in order of increasing size) usually gives the appropriate size. Here are three examples.

$$c(A) = \min\left\{\epsilon : \sum_{i=0}^{\infty} |G^i M^{-1}| \le \epsilon|(I-G)^D M^{-1}|\right\}, \quad (\backslash\texttt{left}),$$

$$c(A) = \min\biggl\{\epsilon : \sum_{i=0}^{\infty} |G^i M^{-1}| \le \epsilon|(I-G)^D M^{-1}|\biggr\}, \quad (\backslash\texttt{bigg}),$$

$$S_n = \sqrt{\frac{2}{n+1}}\left(\sin\left(\frac{ij\pi}{n+1}\right)\right)_{i,j=1}^{n}, \quad (\backslash\texttt{left}),$$

$$S_n = \sqrt{\frac{2}{n+1}}\biggl(\sin\Bigl(\frac{ij\pi}{n+1}\Bigr)\biggr)_{i,j=1}^{n}, \quad (\backslash\texttt{Big}\text{ inner, }\backslash\texttt{bigg}\text{ outer}),$$

$$G\left(x^{(k)}\right) = -g\left(x^{(k)}\right), \quad (\backslash\texttt{left}) \qquad G\bigl(x^{(k)}\bigr) = -g\bigl(x^{(k)}\bigr) \quad (\backslash\texttt{big}).$$

Figures in LaTeX

Figures can be drawn in LaTeX using the `picture` environment, which has a repertoire of circles, ovals, and straight lines at a limited number (48) of slopes. (Figure 13.1, below, was drawn this way.) Because of the limitations of TeX for drawing pictures it is often necessary to produce figures with a program designed specifically for the purpose. This is most commonly done using figures represented in PostScript.[21]

The standard way to include PostScript figures in LaTeX is via the `\includegraphics` command in the `graphics` or `graphicx` packages (the latter offers more options and a key–value interface). These packages replace the earlier `epsf` and `psfig` style files that were commonly used with LaTeX 2.09. The usage is illustrated by

```
% Assumes the use of the dvips dvi to PostScript driver.
\usepackage[dvips]{graphicx}
...
\begin{center}
% Include eps file, applying scale factor 0.6.
\includegraphics[scale=0.6]{figure.eps}
\end{center}
```

For much more on generating and manipulating PostScript files with LaTeX, see the references in §13.5.

File Names and Internet Addresses

File names, email addresses and URLs of objects on the Web can cause formatting problems when typed within a paragraph because TeX may not be able to find a suitable line break within them. A solution is to use the LaTeX `path` package, available from the CTAN archives (see §13.5). It provides a command `\path` that is similar to the command `\verb` but which allows hyphen-less line breaks at punctuation characters. For example, I typed the anonymous ftp locator `ftp://ftp.ma.man.ac.uk/pub/narep/narep305.ps.gz` as

```
\path"ftp://ftp.ma.man.ac.uk/pub/narep/narep305.ps.gz"
```

[21] PostScript is a page description language of Adobe Systems, Inc. A PostScript file is an ASCII file of PostScript language commands that specifies how the output is to appear on the page. By convention, the first line of a PostScript file begins with the two characters `%!`. Ghostscript is an interpreter for PostScript that is useful for printing PostScript files on non-PostScript printers and previewing them on the screen. It was written by Aladdin Enterprises, Palo Alto, and runs on various Unix machines and under DOS and Windows.

Labels

It is worth using a consistent and numbering-independent system for the choice of keys in `\label` commands. For example, rather than using the keys for equations `eqn1`, `eqn2`, etc., try `g.def` (definition of function g), `CS-ineq` (Cauchy–Schwartz inequality) and `bound1` (first of several bounds). These more meaningful keys are easier to remember and make the source code easier to navigate. For figures, tables, theorems, lemmas and so on I prefer to use keys of the form `x.y`, where `x` is an abbreviation for the type of object: `Fig.residuals`, `Lem.limit`, `Thm.Fermat`, etc.

The `showlabels` package, available from the CTAN archives, can be useful at the drafting stage: it causes the key to be shown in the margin at each instance of a `\label` command.

Macros

It is a good policy to use macros to typeset important notation, whether it be words or symbols. As well as reducing the amount of typing and ensuring consistency, this helps to emphasize logical structure in the source, which will be appreciated if you have to change the notation, for only the macro definitions will need to be altered. The latter point is discussed by Lamport [171], who argues against WYSIWYG (what you see is what you get) systems and in favour of markup systems such as LATEX. As an example, among the macros used in typesetting this book are[22]

```
\def\mw{mathematical writing}
\def\norm#1{\|#1\|}
\def\Htk{\widetilde{H_k}}
```

The first macro is used to save typing: I type `\mw` instead of "mathematical writing". I find the second macro makes the source code easier to read and modify: I type `\norm{x}` to produce $\|x\|$. It makes it easy to change from a general norm to a specific one if the need arises (e.g., to convert to the 2-norm the definition is changed to `\def\norm#1{\|#1\|_2}`). The third macro is used in the source for §3.5; again, it saves typing and simplifies any future change of notation.

Miscellaneous Mathematics

The colon character is treated as a relational operator in mathematics mode, as in $x := y$ and $\{ z : |z| \le 1 \}$. If you want a colon to be

[22] The LATEX user is recommended to define macros with the `\newcommand` command rather than `\def`. With `\newcommand` an attempt to define an existing command produces an informative error message, while if `\def` is used, an error can result that is hard to track down. I tend to use `\def` anyway!

Table 13.1. Confusable pairs of symbols.

Relation or binary operation symbol	Example	Ordinary symbol	Example
`\mid`	$\{\,x \mid x > 0\,\}$	`\vert` or `|`	$\lvert z\rvert$
`\setminus`	$G \setminus H$	`\backslash`	$p\backslash n$
`\parallel`	$\vec{u} \parallel \vec{v}$	`\Vert` or `\\|`	$\lVert A\rVert$
`\perp`	$\vec{u} \perp \vec{v}$	`\bot`	$x_{\bot}$
`\in`	$x \in \mathbb{R}$	`\epsilon`	$\epsilon > 0$

treated as a punctuation mark use `\colon`. Thus, $A(1{:}\, r, 1{:}\, r)$ is typed as `$A(1\colon r,1\colon r)$`. Table 13.1 lists some pairs of symbols that might be confused.

To express that one quantity is much bigger or much less than another, use the `\gg` or `\ll` symbol.

Bad: $x >> y$ (typed as `$x >> y$`).

Good: $x \gg y$ (typed as `$x \gg y$`).

Similarly, to obtain a norm symbol $\|$ type `\|` rather than `||`.

Recall that mathematical functions should be typeset in roman (see page 32). This rule is frequently violated by TeX users, giving aesthetically disturbing results such as $sign(det(A)) = -1$ instead of $\mathrm{sign}(\det(A)) = -1$. If you are using a function for which there is not a predefined macro, define one of the form

```
\def\diag{\mathop{\mathrm{diag}}}
```

Then type formulas such as $D = \mathrm{diag}(a_{ii})$ as `$D = \diag(a_{ii})$`. More generally, any term with more than one letter should normally be set in roman, such as tol, representing a tolerance. An exception is in program or algorithm listings, where, for example, italics might be used for variables.

I prefer to use the accents `\widehat` and `widetilde` ($\widehat{x}$, $\widetilde{x}$) instead of `\hat` and `\tilde` ($\hat{x}$, $\tilde{x}$) because the wide versions are easier to read and are less likely to be mistaken for a stray blob on a photocopied page.

Exercise restraint in the use of TeX's mathematical formatting within the text—elaborate formulas can look out of place and tall ones can affect the line spacing. A common mistake is to use the LaTeX command `\frac` instead of a slash: $x = (y+z)/2$ looks better than $x = \frac{y+z}{2}$, and $\frac{a}{b}$ is better written as ab^{-1} or a/b.

An ellipsis (the sequence of three dots "...") is used to mark omission from text or in a mathematical formula such as "$i = 1, 2, \ldots, n$." In both cases if you type the ellipsis as three full stops in TEX the spacing is too tight: $i = 1, 2, ..., n$. Instead, the control sequence `\dots` should be used (or `\cdots` for centred dots in a formula—see page 32). Vertical and diagonal ellipses for use in matrices are obtained with

`\vdots` ($\vdots$) and `\ddots` ($\ddots$).

Compare the two expressions

$$S = A_{22} - A_{21}A_{11}^{-1}A_{21}^T,$$
$$S = A_{22} - A_{21}{}^{}A_{11}^{-1}A_{21}^T.$$

Notice that in the first the subscripts in the third term are not on the same horizontal level. This has been corrected in the second expression by typing `A_{21}^{}`, where the empty superscript has the effect of lowering the subscript to the desired level. Whether you think this sort of refinement is worth the trouble depends on your aesthetic judgement.

Which of the following two displays do you prefer?

$$\Pi^T(A + \Delta A)\Pi = (R + \Delta R)^T(R + \Delta R),$$
$$\mathit{\Pi}^T(A + \mathit{\Delta} A)\mathit{\Pi} = (R + \mathit{\Delta} R)^T(R + \mathit{\Delta} R).$$

The second differs from the first only in that the Greek letters have been made italic by the use of the `\mathit` command. I prefer to put uppercase Greek letters in the italic font because they then mesh better with the italic roman letters, particularly in expressions such as $\mathit{\Delta} A$.

Finally, as an example of how TEX's spacing can sometimes be fine-tuned, compare

$\mathit{\Pi}^T A\mathit{\Pi}$, typed as `$\mathit{\Pi}^T A\mathit{\Pi}$`,
$\mathit{\Pi}^T\mskip-5mu A\mathit{\Pi}$, typed as `$\mathit{\Pi}^T\mskip-5mu A\mathit{\Pi}$`.

The small negative space `\mskip-5mu`, where the value -5 was obtained by experimentation, removes the unattractive extra space between the Π^T and the A.

Quotes, Dates, Lists and Paragraphs

Be careful to distinguish between the opening quote (‘ or `\lq`) and the closing quote (’ or `\rq`). Errors such as "unmatched quotes" are irritating to the trained eye. Do not use the double quote `"`, which always produces the closing quotes ”.

In LATEX set the date explicitly using the `\date` command (assuming you use `\maketitle`)—otherwise the date that is printed will be the date on which the file was LATEXed, which is less informative.

Table 13.2. Order of invocation of LaTeX, BibTeX and MakeIndex.

LaTeX only	LaTeX and BibTeX	LaTeX, BibTeX and MakeIndex
`latex`	`latex`	`latex`
`latex`	`bibtex`	`bibtex`
	`latex`	`latex`
	`latex`	`latex`
		`makeindex`
		`latex`

Be careful not to overuse lists. LaTeX makes it very easy to produce bulleted or numbered lists, with the `itemize` and `enumerate` environments. An inappropriate use of lists is to present a series of ideas without linking them smoothly. A list may be the remnant of an outline that has not been fully developed. If a list is longer than a full page, or has more than one paragraph in an entry, you should consider converting it to running text.

It is easy to start new paragraphs unintentionally in TeX by leaving a blank line after a list, quotation or mathematical display. Watch out for this error, as it is easy to overlook when you are revising and proofreading and it can be disconcerting for the reader.

Running LaTeX, BibTeX and MakeIndex

To ensure that cross-references, references to the bibliography, and index entries are correctly resolved, you need to run LaTeX more than once. Table 13.2 shows the necessary sequences of invocations. However, in certain circumstances more invocations will be required. For example, if a BibTeX entry is used that itself contains a `\cite` command (most likely in a `note` field) then BibTeX needs to be run twice:

```
latex, bibtex, latex, bibtex, latex, latex
```

Always check the `log` file for warning messages before printing a `dvi` file.

Source Code

When typing your paper make liberal use of spaces and indentation to improve the readability of the source. Start new sentences and main phrases on new lines. This approach simplifies editing: the source is easier to navigate, and cut and paste operations are easier because they tend to act on whole sentences or phrases. It may also be desirable to limit lines to

72 characters, for reasons explained in §8.7. Here is a short extract from a paper of mine (in LaTeX) that illustrates these points:

```
We assume that $x_{k+1}$ is computed by
forming $Nx_k+b$ and then solving a linear system with $M$.
The computed vectors $\xhat_k$ therefore satisfy
\begin{equation}
        (M + \dM_{k+1})\xhat_{k+1} = N\xhat_k + b + f_k,
  \label{be1}
\end{equation}
```

Spacing in Formulas

Correct spacing in displayed formulas can usually be accomplished with the `\quad` and `\qquad` commands. A `\quad` is a printer's quad of horizontal space, which is approximately the width of a capital M, and a `\qquad` is twice the width of a `\quad`. A `\qquad` is appropriate when a formula is followed by a qualifying expression. Less often, a thick space, typed as `\;` (5/18 of a quad), is useful. Here are some examples of spacing.

$$\begin{aligned} \|A^T\|_p &= \|A\|_q, \qquad \frac{1}{p}+\frac{1}{q}=1,\\ fl(x \mathbin{\mathrm{op}} y) &= (x \mathbin{\mathrm{op}} y)(1+\delta), \qquad |\delta|\le u, \quad \mathrm{op} = *,/, \end{aligned}$$

$$\alpha_i = \alpha_j \quad (i<j) \quad \Rightarrow \quad \alpha_i=\alpha_{i+1}=\cdots=\alpha_j.$$

$$\eta(y) = \min\{\,\epsilon : (A+E)y = b+f,\ \ \|E\|\le\epsilon\|A\|,\ \ \|f\|\le\epsilon\|b\|\,\}.$$

These displays were typed in LaTeX as follows (I have wrapped some of the lines so that they fit comfortably in the given page width).

```
\def\op{\mathbin{\mathrm{op}}}  % Binary operator
\def\norm#1{\|#1\|}
\begin{eqnarray*}
    \norm{A^T}_p &=& \norm{A}_q, \qquad
                     \frac{1}{p} + \frac{1}{q} = 1, \\
     fl(x \op y) &=& (x \op y)(1+\delta),
                     \qquad |\delta|\le u, \quad \op = *,/,
\end{eqnarray*}
$$
    \alpha_i = \alpha_j \quad (i<j) \quad \Rightarrow
    \quad \alpha_i = \alpha_{i+1} = \cdots = \alpha_j.
$$
$$
```

```
 \eta(y) = \min \{\, \epsilon: (A+E)y = b+f, \;\;
                     \norm{E} \le \epsilon \norm{A},
                     \;\; \norm{f} \le \epsilon \norm{b} \,\}.
$$
```

Note the thin spaces (obtained with `\,` and of width 1/6 of a quad) placed inside the parentheses in the definition of η, which are recommended for sets defined by a generic element followed by a specific condition [161, p. 174]. Note also that the TeX construct `$$..$$` is more properly typed in LaTeX as `\[..\]` or as a `displaymath` environment.

A `\quad` is also useful for separating words from symbols:

$$
\begin{array}{c}
\mbox{minimize} \quad F(x) \quad \mbox{subject to} \quad c(x) = 0, \\
y'(t) = F(t, y(t), y(t-\tau)), \quad y(t) = \Phi(t) \mbox{ for } t \in [-\tau, 0], \\
g(n) \ge \displaystyle\frac{n+1}{2} \quad \mbox{for all } n.
\end{array}
$$

These displays were typed in LaTeX as follows.

```
\begin{eqnarray*}
  & \mbox{minimize} \quad F(x) \quad \mbox{subject to}
                                \quad c(x) = 0 & \\
  &  y'(t) = F(t, y(t), y(t-\tau)),
                             \quad \mbox{$y(t) = \Phi(t)$
                                for $t\in[-\tau,0]$} & \\
  & g(n) \ge \displaystyle\frac{n+1}{2}
                             \quad \mbox{for all $n$}. &
\end{eqnarray*}
```

When breaking a displayed formula before a plus or minus sign (see §3.7), you need to type `{}+` or `{}-` so that TeX knows the sign is acting as a binary operator. Without the `{}`, the lesser spacing assigned to a unary operator is produced. Compare these two `eqnarray`s, where the second line of the second is typed as `&& {}+ y + z`:

$$
\begin{array}{rcl} A & = & v + w + x \\ & & +y + z \end{array}
\qquad\qquad
\begin{array}{rcl} A & = & v + w + x \\ & & {}+ y + z \end{array}
$$

By default, LaTeX puts rather too much space around the centre column of an `eqnarray` environment. A simple cure, used in this book, is to use the `eqnarray` package of David M. Jones, located at `ftp://theory.lcs.mit.edu/pub/tex/eqnarray.sty`

Ties and Spaces

Avoid unfortunate line-breaks by binding units together with a tie (`~`). For example, as a matter of course type `Theorem~3` not `Theorem 3`, `Mr.~Sandman` not `Mr. Sandman`, and `1,~2 and~3` not `1, 2 and 3`. Other bad breaks, such as between the last two words in a paragraph, can be corrected as and when they appear.

Leave a space before the `\cite` command.

Bad:	`Knuth\cite{knut86}`, giving Knuth[161].
Good:	`Knuth \cite{knut86}`, giving Knuth [161].
Good:	`Knuth~\cite{knut86}`, giving Knuth [161].

The “Knuth[161]” form is a common mistake. The last form avoids a line break just before the citation.

TEX assumes that a full stop ends a sentence unless it follows a capital letter. Therefore you must put a control space after a full stop if the full stop does not mark the end of the sentence. Thus p. 12 is typed as `p.\ 12`, and cf. Smith (1988) as `cf.\ Smith (1988)`. A less obvious example: MATLAB (The MathWorks, Inc.) is typed as

```
\textsc{Matlab} (The MathWorks, Inc.)\
```

since otherwise TEX will put extra space after the right parenthesis. In the references in a `thebibliography` environment there is no need to use control spaces because this environment redefines the full stop so that it does not give end of sentence spacing.

On the other hand, if a sentence-ending full stop follows an uppercase letter you must specify that the full stop ends the sentence. In LATEX this can be done by inserting the `\@` command before the full stop, as in the sentence from page 11

```
There is also an appendix on how to prepare a CV\@.
```

The `\@` command is needed more often than you might realise—in this book there are over 10 occurrences of the command.

13.3. BIBTEX

BIBTEX, written by Oren Patashnik, is a valuable aid to preparing reference lists with LATEX. To use BIBTEX you need first to find or create a `bib` file that comprises a database of papers containing those you wish to cite. BIBTEX reads a LATEX `aux` (auxiliary) file and constructs a sorted reference list in a `bbl` file, making use of the `bib` file. This reference list is read and processed by LaTeX. A diagram showing how LaTeX interacts with BibTeX and MakeIndex is given in Figure 13.1 (MakeIndex is described in the next section). In this section I go into some detail on the use of BibTeX because it tends not to be emphasized by the books on LaTeX and yet is a tool that can benefit every serious LaTeX user.

A `bib` file is an ASCII file maintained by the user (in the same way as a `tex` file). As an example, references [158] and [161] are expressed in the BibTeX file used for this book as[23]

```
@article{knut79,
  author = "Donald E. Knuth",
  title = "Mathematical Typography",
  journal = "Bull. Amer. Math. Soc. (New Series)",
  volume = 1,
  number = 2,
  pages = "337-372",
  year = 1979
      }

@book{knut86,
  author = "Donald E. Knuth",
  title = "The {\TeX book}",
  publisher = "Addison-Wesley",
  address = "Reading, Massachusetts",
  year = 1986,
  pages = "ix+483",
  isbn = "0-201-13448-9"
      }
```

The first part of the last sentence was typed as

```
As an example, references \cite{knut79} and \cite{knut86} are
expressed in \BibTeX\ format as
```

Once you have built up a few entries, creating new ones is quick, because you can copy and modifying existing ones.

There are three advantages to using BibTeX.

1. By specifying the appropriate bibliography style (`bst` file) from the many available it is trivial to alter the way in which the references are formatted. Possibilities include BibTeX's standard `plain`, `abbrv` and `alpha` formats, illustrated, respectively, by

[23] My `bib` file uses abbreviations (described below) for the journal, publisher and address fields, but in this example I give the fields explicitly, for simplicity.

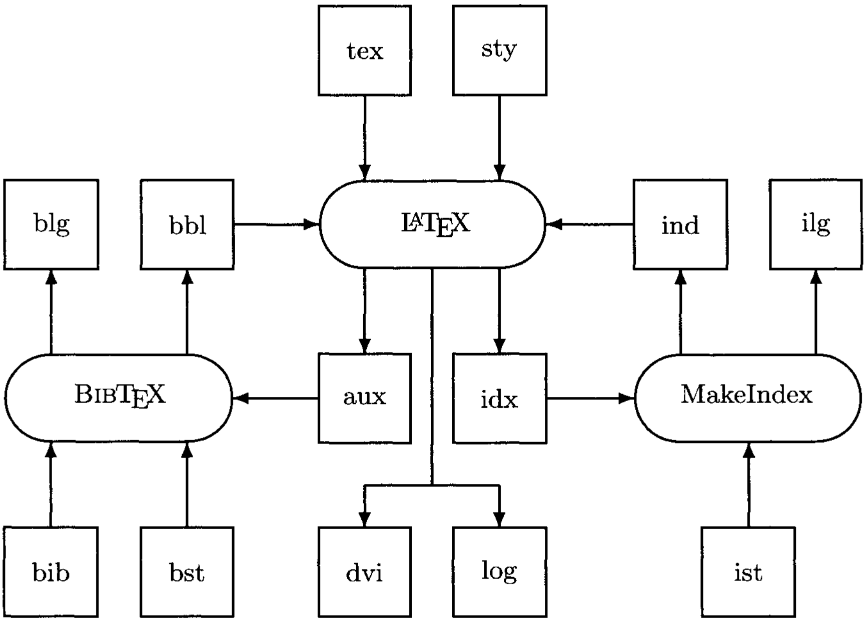

Figure 13.1. Interaction between LaTeX, BibTeX and MakeIndex. The `log`, `blg` and `ilg` files contain error messages and statistics summarizing the run. The `aux` file contains cross-referencing information. The `sty`, `bst` and `ist` files determine the styles in which the document, reference list and index are produced.

[1] Donald E. Knuth. Mathematical typography. *Bulletin Amer. Math. Soc. (New Series)*, 1(2):337–372, 1979.

[1] D. E. Knuth. Mathematical typography. *Bulletin Amer. Math. Soc. (New Series)*, 1(2):337–372, 1979.

[Knu79] Donald E. Knuth. Mathematical typography. *Bulletin Amer. Math. Soc. (New Series)*, 1(2):337–372, 1979.

With the `alpha` format, citations in the text use the alphanumeric label constructed by BibTeX ([Knu79] in this example). The `unsrt` format is the same as `plain` except it lists the entries in order of first citation instead of alphabetical order (as is required, for example, by the journal *Computers and Mathematics with Applications*). LaTeX packages and corresponding BibTeX style files are also available that produce citations and bibliographies conforming to the Harvard system (see §6.11).

In this book I am using my own modification of the `is-plain` style by Nelson Beebe, which itself is a modification of the `plain` style to add support for ISSN and ISBN fields and for formatting a pages field for books.

2. To keep the reference lists of working papers up to date (as technical reports become journal papers, for example) it is necessary only to update the master `bib` file and rerun LaTeX and BibTeX on each paper. If BibTeX were not used, the reference list in each paper would have to be updated manually. Using BibTeX also saves on storage, for the `bbl` files can be deleted once typesetting is complete and reconstructed when required by running BibTeX.

3. Since BibTeX is widely used and available on a wide range of machines it is possible for people to exchange and share databases. Two large collections of `bib` files deserve particular note.

   (a) BibNet is maintained by Stefano Foresti, Nelson Beebe and Eric Grosse. It has the URL `ftp://ftp.math.utah.edu/pub/bibnet`, and is also available from netlib (see §14.1), specifically from

   `http://netlib.bell-labs.com/netlib/bibnetfaq.html`

   The `bib` files in BibNet include ones containing all the publications by particular authors (e.g., Gene Golub), and all (or many of) the publications that have appeared in particular journals (e.g., *SIAM Review*).

   (b) The Collection of Computer Science Bibliographies (which includes bibliographies on mathematics), is maintained by Alf-Christian Achilles at

   `http://liinwww.ira.uka.de/bibliography/index.html`

   This large collection of bibliographies (which includes all those in BibNet) has an excellent Web interface and powerful search facilities.

It is, of course, advisable to check the accuracy of any entries that you have not created yourself, before using them.

Aside from these BibTeX-specific reasons, there are other reasons for keeping a personalized computer database of references. If you record every paper you read then a computer search allows you to check whether or not you have read a given paper. This is a very handy capability to have, as once you have read more than a couple of hundred papers it becomes difficult to remember their titles and authors. Also, you can put comments in the database to summarize your thoughts about papers. Some BibTeX users include an `annotate` field in their `bib` files. Although standard BibTeX style files ignore this field, other style files are available that reproduce it, and they can be used to prepare annotated bibliographies. In my BibTeX book entries I include an `isbn` field.

Here are some tips on using BibTeX.

1. To make it easier to navigate a `bib` file with your editor, keep the entries in alphabetical order by author, and use positioning lines of the form

   ```
   -md- -me-
   ```

   to indicate the start of a group of authors—this example marks the start of the authors whose last names begin with "Me", there being no last names beginning "Md".

2. Although it is tempting to save time by abbreviating `bib` file entries—typing initials instead of full author names (when they are given in the original reference), and omitting journal part numbers or institution addresses for technical reports—my experience is that it is false economy, because these details are often required eventually: for example, by a copy editor who queries an incomplete reference on page proofs. It is worth spending the extra time to create fully comprehensive entries. Note that if names are typed with no space between the initials (e.g., `author = D.E. Knuth`), BibTeX produces only the first initial. You should therefore always leave a space between initials (`D. E. Knuth`), as is standard practice in typesetting.

3. Various conventions are in use for choosing keys for `bib` file entries (they appear on the first line of the entry and are specified in the `\cite` command). My method is to use the first four letters of the author's last name followed by the last two digits of the year. If there are two authors I use the first two letters of each author's last name and if three or more, the first letter of the first three or four authors' last names. Multiple papers in the same year by the same authors are distinguished by extra letters 'a', 'b' and so on. This method has proved effective for `bib` files with a few thousand entries. Note that a key does not have to be of the form `author:gnus`, as some people

have presumed on reading Lamport's whimsical examples involving gnus [172]!

4. BibTeX allows the use of abbreviations. For example, instead of typing

```
publisher = "Society for Industrial and
             Applied Mathematics",
address = "Philadelphia",
```

you can type

```
publisher = pub-SIAM,
address = pub-SIAM:adr,
```

where the following string definitions appear prior to these lines in the `bib` file (typically at the start of the file):

```
@STRING{pub-SIAM   = "Society for Industrial and
                      Applied Mathematics"}
@STRING{pub-SIAM:adr = "Philadelphia"}
```

The use of abbreviations saves typing and ensures consistency between fields that should be identical in different entries.

A bib file `mrabbrev.bib` containing string definitions for all the standard journal abbreviations used in *Mathematical Reviews* is part of AMS-LaTeX and is also available from BibNet.

5. If you wish to include the URL of a file available on the Web, put it in a URL field. For example,

```
URL = "ftp://ftp.ma.man.ac.uk/pub/narep/narep306.ps.gz"
```

Although URL fields are not supported by the standard `bst` files, they are used in some BibNet databases and facilitate the creation of hypertext links from a `bib` file to the actual papers.

6. When writing a book or thesis you often wish to print out a draft chapter together with a bibliography comprising just the references cited in that chapter. This can be achieved using the LaTeX package `chapterbib` by Donald Arseneau, available from the CTAN archives.

Various tools are available to help maintain BIBTEX databases (see [109, §13.4] for details). These include tools to sort databases, search them, syntax check and pretty print them (the program `bibclean` [17] even checks ISBN and ISSN fields to see whether the checksum is correct), and extract from them the entries cited in a set of `aux` files (so as to create a `bib` file containing only the entries used in a particular document). Many of the utilities are available from BibNet. Some of these tasks are easy to do oneself using AWK, which is an interpreted programming language available on most Unix systems [4]. For example, the command line

```
awk 'BEGIN{ RS="" } /Riemann/' my.bib
```

searches the file `my.bib` in the current directory and prints all the entries that contain the word Riemann (assuming that records are separated by a blank line). Note that this AWK call prints complete `bib` entries, not just the lines on which the word occurs, as a `grep` search would.

13.4. Indexing and MakeIndex

An index to a book (or thesis, or report) has three main purposes.

1. To provide easy access to all the significant information.
2. To reveal relationships.
3. To reveal omissions.

A good index is therefore much more than a table of contents. But it is much less than a list of every important word, since it records useful information, not just key words. While a printed book is necessarily expressed in a linear order, the index is not constrained by ordering and can therefore reveal links between different parts of the book and bring together topics described in the text with varying terminology. A good index saves time for the reader as a result of what it *does not* contain: if a topic is not present in the index the reader can be sure that it is not covered in a significant way in the text.

An index should contain surprises—pointers to passages that the reader might overlook when scanning the book and its table of contents. It should anticipate the various ways in which a reader might search for a topic, by including it under multiple entries, where appropriate. For example, *block LU factorization* might be listed under *block*, *factorization*, *LU*, and, in a book not about matrices, under *matrix*. Since decomposition is a commonly used synonym for factorization, an entry "decomposition, *see* factorization" would also be appropriate in this example.

One source of index entries is section and subsection headings, since these provide the framework of the text. Entries should be nouns or nouns preceded by adjectives. Any conventions used in the index must be explained in a note at the beginning. For example, you might use "t" after a page number to denote reference to a table, and "f" for a figure.

If many names are to be indexed, it is worth creating separate name and subject indexes, as in this book. One reason for indexing names is to enable the reader to find where a particular paper in the bibliography is referenced, assuming, of course, that the author's name is mentioned at the point of reference.

A common mistake is to produce an index entry with too many page locators. If there are more than about five page locators, subentries should be introduced to help the reader pinpoint the information required. For example, the index entry

norm, 119–121, 123, 135, 159, 180

is much better broken down into, for example,

norm
  absolute, 119, 121
  dual, 120
  elliptic, 180
  Hölder inequality, 123
  spectral radius, relation with 135
  unitarily invariant, 159

In the following example the subentries serve little purpose because they all have the same page number:

LU factorization
  definition, 515
  existence, 515
  uniqueness, 515

This example should be collapsed into the single entry "LU factorization, 515", which is just as useful for a reader searching for information about the LU factorization.

Choose as main headings the word that the reader is most likely to look under. Thus

equations, displaying, 54

is better than

displaying equations, 54

This example is formatted as it should be if there are no other subentries of "equations". In the examples below, the subentry is assumed to be one of several and so appears on a separate line.

In subentries, use connectives to clarify the meaning of the entries. The entry

slides
    number, 138

could refer to a discussion on how to number the slides or on how many to produce. Adding the word "of" avoids the ambiguity:

slides
    number of, 138

It can be useful to add the word "of" even in unambiguous cases to make the entry read smoothly from subentry to heading:

words
    order of, 60

In traditional typesetting, indexing was a task to be done once a book was at the page-proof stage, and was often performed under severe time pressure. Nowadays, authors typesetting their own books by computer can index earlier in the production process, making use of indexing software.

MakeIndex is a C program, written by Pehong Chen [56], [109, Chap. 12] with advice from Leslie Lamport, that makes an index for a LATEX document. The user has to place `\index` commands in the LATEX source that define the name and location of the items to be indexed. If a `\makeindex` command is placed in the preamble (before `\begin{document}`) then LATEX writes the index entries, together with the page numbers on which they occur, to an `idx` file. This is read by the MakeIndex program, which processes and sorts the information, producing an `ind` file that generates the index when included in the LATEX document (see Figure 13.1). MakeIndex provides various options in the `\index` command to support standard indexing requirements, such as subentries, page ranges and cross-references to other entries. Here is how the beginning of one sentence from page 187 was typed:

```
\item  It is easy to prepare
transparencies\index{slides!preparing in \TeX}
with \TeX\ if the paper
```

The exclamation mark in the `\index` command denotes the beginning of a subentry. Multiple indexes (such as name and subject indexes) can be produced with the aid of the `index` package by David M. Jones, available from `ftp://theory.lcs.mit.edu/pub/tex/index/`

Here are some tips on indexing in LATEX.

1. Insert the index entry immediately following the word to be indexed, on the same line and with no spaces before the `\index` command (as in the example above). This ensures that the correct page reference is produced and avoids unwanted spaces appearing in the output.

2. If the scope of the item being indexed is more than one sentence, so that the scope may be broken over a page, index the item as a page range. For example, this list of tips is contained within commands

   ```
   \index{LaTeX@\LaTeX!indexing in|(}
   \index{indexing!in latex@in \LaTeX|(}
   ...
   \index{indexing!in latex@in \LaTeX|)}
   \index{LaTeX@\LaTeX!indexing in|)}
   ```

   The `|(` and `|)` strings serve to delimit the range of the index command.

3. *See* entries can be produced by commands of the form

   ```
   \index{dots|see{ellipsis}}
   ```

   To produce *see also* entries in an analogous way you can use the following definition, adapted from that for `\see` in `makeidx.sty`:

   ```
   \newcommand\seealso[2]{\emph{see also} #1}
   ```

   Place all *see* and *see also* index entries together, to make it easier to edit them and check for consistency. I suggest placing them after the last item in the book to be indexed (ideally, just before the bibliography); this ensures that *see also* appears after the page references for an entry.

4. Do not leave the task of indexing to the very last stage. For, in inserting the `\index` entries, you are likely to introduce errors (of spacing, at least) and so a further round of proofreading will be needed after the indexing stage.

AWK tools for indexing are described in [4, §5.3] and [22]; these tools do not support subentries. A simple and elegant way to construct key word in context (KWIC) indexes using AWK is also described in [4]. A KWIC index lists each word in the context of the line in which it is found; the list is sorted by word and arranged so that the key words line up. One of the main uses of KWIC indexes is to index titles of papers.

For an interesting example of an index, see Halmos's *I Want to Be a Mathematician* [127]. Priestley [230] says "If index writing has not bloomed into an art form in *I Want to Be a Mathematician*, it has at least taken a quantum leap forward. From 'academic titles, call me mister' to 'Zygmund, A., at faculty meetings' this one is actually worth reading."

13.5. Further Sources of Information

The best (and the most humorous) introduction to LaTeX is *Learning LaTeX* [118] by Griffiths and D. J. Higham. A much longer and more detailed book that is very handy for reference is *A Guide to LaTeX 2ε* [166] by Kopka and Daly. Lamport's *LaTeX: A Document Preparation System* [172] is the "official" guide to LaTeX. For those still using the obsolete LaTeX 2.09, Carlisle and Higham [53] explain the advantages to be gained by upgrading to LaTeX 2ε.

For technical details of LaTeX, BibTeX and MakeIndex, and descriptions of the many available packages, see *The LaTeX Companion* [109] by Goossens, Mittelbach and Samarin. *The LaTeX Graphics Companion* [110] by Goossens, Rahtz and Mittelbach is the most comprehensive and up-to-date reference on producing graphics with LaTeX and PostScript. If you are a really serious LaTeX or TeX user you will want to study Knuth's *The TeXbook* [161], the "bible" of TeX, or another advanced reference such as Salomon's *The Advanced TeXbook* [244].

BibTeX is described in all the LaTeX textbooks mentioned above, but most comprehensively in *The LaTeX Companion* [109].

An article by Knuth [158][24] offers many insights into mathematical typesetting and type design, and describes early versions of TeX and METAFONT (METAFONT [160] is Knuth's system for designing typefaces).

The Comprehensive TeX Archive Network (CTAN) is a network of ftp servers that hold up-to-date copies of all the public domain versions of TeX, LaTeX, and related macros and programs. The three main sites are at

`ftp.dante.de,`	`http://www.dante.de/`
`ftp.tex.ac.uk,`	`http://www.tex.ac.uk/tex-archive`
`tug2.cs.umb.edu,`	`http://tug2.cs.umb.edu/ctan/`

which are located in Germany, England and Massachusetts, USA, respectively. There are many mirror sites around the world, details of which may be obtained from the TeX Users Group (TUG) Web pages. The organization of TeX files is the same on each site and starts at `./tex-archive`. To search a CTAN site during an anonymous ftp session type the command

[24] The beginning of the abstract is quoted on page 86.

`quote site index string`, where `string` is a Unix regular expression (a filename optionally containing wildcards) on which to search.

The TUG runs courses and conferences on TeX and produces a journal called *TUGboat*. It also produces a newsletter for members called *TeX and TUG News*. Contact details for TUG are given in Appendix D.

The UK TeX Users Group, based in the UK, also organizes meetings and produces a newsletter (called *Baskerville*). It cooperates with TUG and supports the UK TeX archive (the UK node of CTAN). More information is available on the Web at `http://www.tex.ac.uk/UKTUG/` or via email to `uktug-enquiries@uk.ac.tex`

Excellent advice on preparing an index is given in *The Chicago Manual of Style* [58] and in Bonura's *The Art of Indexing* [35]. The collection *Indexers on Indexing* [130] contains articles on many different aspects of indexing that originally appeared in *The Indexer*, the journal of the Society of Indexers (UK). This society is involved in awarding indexing prizes; the 1975 Wheatley Medal was awarded for the index (by Margaret D. Anderson) to the first edition of [45]. Other good references are *Words into Type* [249] and *Copy-Editing* [45].