C. U. E. D Document Preparation Facilities and Local Guide

Ian Green, 5 September 1991, December 1992.
Updated to LaTeX2e by Tim Love, Sept 1994

September 4, 1994

Abstract

The process of document preparation is described. A brief description of the use of the LaTeX document preparation system is given to illustrate the ease with which well-typeset documents can be created. Other topics include the incorporation of graphics from the drawing program XFIG, the use of BibTeX for bibliography management, using XDVI to preview documents prior to printing, and printing documents on a laser printer. For experienced users, an appendix provides a *CUED Local Guide* that covers topics unique to the CUED LaTeX system. In summary this document covers:

- LaTeX— A powerful document preparation system.
- BibTeX — Managing bibliographies and citations within LaTeX.
- XDVI — Previewing LaTeX documents before printing them.
- XFIG — Incorporating XFIG drawings into LaTeX documents.
- *Local Guide* — Getting the most out of LaTeX at CUED.

Contents

List of Figures

List of Tables

1 Introduction

Writing documents—reports, papers and the like, can be a real hassle if you don't make use of computing facilities. Producing and placing figures/diagrams, constructing tables and lists, inserting and numbering footnotes, writing equations, sectioning, cross-referencing, referencing etc., all require effort on behalf of the author. Authors can spend more time producing a document than they spend writing it.

This document is intended to familiarize the reader with some of the computing facilities available here at the CUED to assist with the tiresome aspects of document preparation. It is in two parts: the main part is an beginners guide to using the CUED facilities; an Appendix includes detailed and advanced material that you may need in the future.

As a result this document contains much information; we think it's worth spending some time going through it. For example, if you've to write a first year report, then time spent now will save a lot of stress and frustration later next term! Note however that this is by no means the definitive reference for the programs etc. described here, the reader is advised to refer to other sources if need be. We include in marginal notes pointers to more specific material.

This is a marginal note.

If you find any errors or omissions in this document, please let Tim Love know (`tpl@eng`).

2 LaTeX

2.1 Introduction

You'll soon become familiar with the LaTeX logo. To put you in the picture, I'll say a little bit about where LaTeX comes from. In the beginning, a chap called Donald Knuth designed and wrote a typesetting program called TeX, which is pronounced like the first syllable of "technique". TeX is a very low-level language and even simple aspects of producing documents can take a long time to figure out and write down. Luckily however, TeX is expandable; you can parameterize frequently used sequences of TeX code into *macros*; these macros can be used over and over again in a variety of ways. Macros make writing documents easier. You can write your own, or use someone else's—one particular set of macros is referred to as "Plain TeX"; it provides a more user-friendly way of writing documents, but is still rather primitive.

Later, another chap by the name of Leslie Lamport, wrote another macro package, even more user-friendly than Knuth's Plain TeX. He called it LaTeX. It provides a natural way of writing documents, without the need to worry about the detail required when using Plain TeX.

Over the years many extensions have been developed for LaTeX. In 1994 LaTeX 2_ε was installed at CUED; it brings all such extensions back under a single format and adds many new features.

So, in summary, TeX is an executable program providing primitive typesetting commands; these primitive commands may be grouped into more powerful macros, which can then be used as commands; Plain TeX and LaTeX are examples of existing sets of macros. This document looks at the more user-friendly of the two, LaTeX. We won't talk any more about Plain TeX.

The definitive user guide and reference for LaTeX 2_ε is the LaTeX Book (second edition), written by Leslie Lamport. The LaTeX Companion by Goossens *et al* is also very useful. In marginal notes where only section numbers are mentioned, they refer to this book. Copies of the LaTeX Book may be found in the machine room.

2.1.1 The *Local Guide*

Occasionally the LaTeX Book refers to things that are not part of LaTeX but complementary to it. Such things vary from place to place and between implementations etc., depending upon things like editors, computer hardware, types of laser printers and so on. Because of this, Lamport asks you to look for this information in the *Local Guide*. The *CUED Local Guide* documents details all of the things that are specific to the Engineering Department HP Teaching System here in Cambridge. As a casual user, you'll rarely need to refer to the *CUED Local Guide*.

2.1.2 Using LaTeX

LaTeX is takes an input text file and processes it to produce a new file that can be sent to a laser printer to create a professional-looking document. LaTeX '*source files*' as they are known, describe your document in every detail to LaTeX: they include normal paragraphs, words and characters, as well as special LaTeX *commands*, which let LaTeX know *how* the final document will appear.

2.1.3 Conventions

Below we present a series of examples of LaTeX-source, starting with simple ideas and developing them quite quickly into all you'll need to know to write your own documents. These examples have have the following format:[1]

We shall give example uses of LaTeX commands by placing on the left-half of the page an example of LaTeX output, while on the right-half of the page we give the LaTeX-source which produced it.	`We shall give example uses of \LaTeX\ commands by placing on the left-half of the page an example of \LaTeX\ output, while on the right-half of the page we give the \LaTeX-source which produced it.`

We use a "`typewriter typeface`" to indicate things that might appear in a LaTeX source file, and a "roman typeface" to indicate LaTeX output (as you might guess, the document you're now reading was produced with LaTeX).

[1] If things appear in examples that you don't understand, don't worry, they won't be important.

2.2 Sentences and paragraphs

Just type what you want, for example:

The ends of words and sentences are marked by spaces. It doesn't matter how many you type; 1 is as good as 10.

One or more blank lines denote the end of a paragraph.

```
The ends of words and sentences are
marked by spaces. It doesn't matter how
many you type; 1 is as good as
10.


One or more blank lines denote the end of
a paragraph.
```

LaTeX ignores the format of your input, so you can arrange it how you like, paying attention only to the logical structure of words, sentences and paragraphs.

If you need to enter comments into your source file that you don't want to appear in the document, use the `%` character; any text from a `%` to the end of that line is ignored by LaTeX.

2.3 Quotation marks

This example illustrates the various quotation marks and their use:

LaTeX distinguishes between four types of quotation marks, unlike many other word processing systems. They are like “this” and ‘this’.

```
\LaTeX\ distinguishes between four types
of quotation marks, unlike many other
word processing systems. They are like
``this'' and `this'.
```

An apostrophe is simply a single right quotation mark ‘’’.

2.4 Dashes

Another difference between LaTeX and lesser programs is its distinction between the various types of dash. There are three types, and you ought to know when to use each:

An intra-word dash or hyphen, as in X-ray.
A medium dash for number ranges, like 1–2.
A punctuation dash—like this.

```
An intra-word dash or hyphen, as in
X-ray.
A medium dash for number ranges, like
1--2.
A punctuation dash---like this.
```

2.5 Ties and hyphenation

Ordinarily LaTeX will split words on a line so that the resulting paragraph is correctly formatted. In some cases it's inadvisable to do this and you may need to prevent adjacent words from being broken across a line. A ‘tie’ is a means of doing this—use the ‘~’ character (a tilde) to tie things together. You'll encounter ties in the examples that follow. Page 18.

2.6 Text-generating commands

Page 16. LaTeX allows you insert text using special commands. For example, the LaTeX and BibTeX logos are produced by the commands `\LaTeX` and `\BIBTeX`. LaTeX may require you to terminate a command using '`\␣`' (where the '␣' character indicates a space) otherwise the spaces after the the command will be lost:

Document preparation is easy if you use LaTeX and BibTeX together.

```
Document preparation is easy if you use
\LaTeX\ and \BIBTeX\ together.
```

2.7 Special symbols

The following symbols have a special meaning to LaTeX, so in order to put them into your document, you need some special commands:

The symbols $, &, %, #, _, { and } are easily produced.

```
The symbols \$, \&, \%, \#, \_, \{ and \}
are easily produced.
```

2.8 Changing type style

You may need to add emphasis to certain words; you can do this is by using the `\emph` command.

This is, as *you* can see, *emphasized text.*

```
This is, as \emph{you} can see,
\emph{emphasized text}.
```

The format is simply `\emph{` followed by the text you want to emphasize, followed by a closing `}` character. The command `\textbf` selects a bold typeface, `\textsl` a slanted typeface, `\textsf` a typeface without serifs and `\textsc` gives small capitals.

Similar commands can be used to change the type style to **bold**, *slanted*, sans serif and SMALL CAPS.

```
Similar commands can be used to change
the type style to \textbf{bold},
\textsl{slanted}, \textsf{sans serif} and
\textsc{Small Caps}.
```

2.9 Scope of declarations

The `\emph` command (see above) is a good example of a LaTeX command that produces neither text nor space, instead, it affects the way other text is typeset. These commands are called *declarations*. Much of the final look of your document is determined by the declarations you use—for example, the type style, the margin width, headers and footers etc. The braces `{` and `}` delimit the scope of declarations: when LaTeX reads the `}` it reverts to the declaration that was in effect *before* the matching `{`.

Page 17.

Note that besides delimiting the scope of declarations, braces are also used to denote *arguments* to commands. The `\LaTeX` and `\BIBTeX` commands described above are examples of commands that don't have arguments. Optional arguments are enclosed within brackets, `[` and `]`; an optional argument may be omitted if it's not needed: typically some *default* value for that argument will be used if it's omitted. For example, the `\framebox` command can be used to draw boxes around pieces of text. `\framebox` has an argument that is the text around which the box is to be drawn; optional arguments can be used to specify the size of this box, and where within the box the text is placed. The default for these optional arguments is a box just the right size for the given text, with the text centered within the box. Take a look at an example:

Page 97.

The word $\boxed{\text{boxed}\qquad\quad}$ is in a one-inch box, left justified. Here the box is $\boxed{\text{just}}$ right.

```
The word \framebox[1in][l]{boxed} is in a
one-inch box, left justified. Here the
box is \framebox{just} right.
```

2.10 Footnotes

Footnotes make your documents look really fancy; in LaTeX they are very easy to use but you have to control your use of them. Too many tend to distract the reader and the page can appear cluttered.

To create a footnote, use the `\footnote` command:[2]

When using footnotes, put them immediately after the word you want to place the footnote against[a] unless that word is followed a punctuation mark.[b]

[a] Like this.
[b] Like this.

```
When using footnotes, put them
immediately after the word you want to
place the footnote against\footnote{Like
this.} unless that word is followed a
punctuation mark.\footnote{Like this.}
```

2.11 Sectioning

One of the compelling reasons for using LaTeX is the support it provides for high-level document structures like chapters and sections. (This is where LaTeX comes into its own over things like Plain TeX.) LaTeX understands how to format chapters, sections and other *sectional units* as they're called. It can number them too. Page 23.

What you're reading now is a section, section 2.11 to be precise. There are many commands related to sectional units: their argument is the name of the section. They are numbered automatically and you can cross-reference this number to use elsewhere (see section 2.15). The sectioning commands include `\chapter`, `\section`, `\subsection`, `\subsubsection` and `\paragraph`.

See section 2.15 for a sample use of the `\subsection` command. (The sectioning commands are all essentially the same, but for the style in which they typeset their argument.

If you prefer not to have the section unit numbered, some of the sectioning commands have a '*-form' (*e.g.* `section*`), which don't typeset the section number.

[2] Footnotes are usually numbered in order of appearance in the document.

2.12 Environments

You've seen environments already in section 2.8 where we looked at the `{\em...}` construct. An environment *name* has the general form:

`\begin{`*name*`}...\end{`*name*`}`

Page 112. As an example, here are some *list-making* environments:

- Each list item is marked with a *label*. The labels in this list are called *bullets*.
- You can nest list environments.
  1. Like this for example.
  2. The items can be numerals or letters.
  3. Here the labels are numbers: they are generated automatically (of course), and can be cross-referenced.

  LaTeX can handle at least four levels of nesting of lists; each list ought to have two or more items.
- Blank lines before items are ignored.

```
\begin{itemize}
 \item Each list item is marked with a
\emph{label}. The labels in this
list are called \emph{bullets}.
 \item You can nest list environments.
 \begin{enumerate}
  \item Like this for example.
  \item The items can be numerals or
letters.
  \item Here the labels are numbers: they
are generated
automatically (of course), and can be
cross-referenced.
  \end{enumerate}
  \LaTeX\ can handle at least four levels
of nesting of lists; each list
ought to have two or more items.
 \item Blank lines before items are
ignored.
\end{itemize}
```

2.13 Mathematics

Page 41. See Tim Love's *LaTeX Maths and Graphics* handout.

2.14 Tables and Figures

Tables

Page 63. LaTeX provides special environments for tables and figures. Here's a simple example of a `tabular` environment:

Bicycle Colour	*Number*
Blue	32
Red	35
Black	76

```
\begin{tabular}{|l|c|}\hline
 \emph{Bicycle Colour} & \emph{Number}\\
             \hline\hline
 Blue                & 32          \\
 Red                 & 35          \\
 Black               & 76          \\
              \hline
\end{tabular}
```

The '`|`' characters in the argument indicate that a vertical line is to be drawn between columns, and the '`l`' and '`c`' mean that things in those columns (*i.e.* the first and second) must be *left-justified* or *centered* ('`r`' is, wait for it, *right-justified*). There are many other entries besides these. The '`\\`' command denotes the end of a row, and `\hline` means draw a horizontal line. Two `\hline`'s together gives two horizontal lines. The '`&`' command separates entries on the same row (see how '`Blue`' and '`32`' are separated).

Figures

Figures are made with the `figure` environment. What you put in a figure is up to you—LaTeX will treat it as a large box. You must read the relevant section of the LaTeX Book to understand exactly how you can use them. It's worth pointing out that the `figure` environment *does not* center its contents on the page so you may need to use the `center` environment. The figure environment automatically generates its own figure number that you can cross-reference with. Examples of the use of the `figure` environment will be given in the section on XFIG (see figure 2 on page 9).

2.15 Cross-referencing

One reason for numbering things like figures and equations is to refer the reader to them, as in "See figure 3 for more details". You don't want the '3' to appear in the input file because adding another figure might make this one become figure 4. Instead you can assign a *key* of your choice to this figure and refer to it by that key, letting LaTeX translate the reference into the figure number. The key is assigned with the `\label` command, and is referred to using the `\ref` command. A `\label` command appearing in ordinary text is assigned the current sectional unit number; inside a numbered environment (like `enumerate`) assigns that number to the key. For example: Page 71.

Equation 12 in section 2.3 below is Euler's famous result.

2.3 Early results

Euler's equation

$$e^{i\pi} + 1 = 0 \tag{12}$$

combines the five most important numbers in mathematics in a single equation.

```
Equation~\ref{eq:euler} in
section~\ref{sec-results} below is
Euler's famous result.
\subsection {Early results}
\label{sec-results} Euler's equation
\begin{equation} e^{i\pi}+1 = 0
\label{eq:euler} \end{equation} combines
the five most important numbers in
mathematics in a single equation.
```

The `\pageref` command is like the `\ref` command except it produces the page number of the place in the text where the corresponding `\label` command appears. This makes it easy to say things like see "section 2.3.5 on page 45 for more information".

2.16 The document and preamble

We're now ready to take a look at producing a whole document. The text of every document starts with a `\begin{document}` command and finishes with Page 21.

an `\end{document}` command (LaTeX will ignore anything coming after this command). The part of the document before the `\begin{document}` command is called a *preamble*.

The preamble can only contain declarations, not normal text. The preamble starts with a `\documentclass` command whose argument chooses one of the predefined classes. The file '`introduction.tex`' uses the `article` class. (As does this document.) In the `article` class, you can declare things like the author(s) names, title and date. Use the `\maketitle` command *after* the `\begin{document}` command to actually make the title. Here's an example document:

```
\documentclass[12pt]{article}
\usepackage{a4}
\title{A sample document}
\date{\today}
\author{A. N. Other}
\begin{document}
\maketitle
Here is some text.

This is another paragraph.
\end{document}
```

Figure 1: A simple example document.

Note that the argument to the `\documentclass` command is the style for the entire document; the optional argument, `[12pt]` indicates that a slightly larger, twelve-point typeface is to be used instead of the default ten-point. Also the article is to be formatted for A4 size paper. This is done by using a *package* called `a4`. See the *CUED Local Guide* for a list of packages and package options.

The `\title` command defines a title for the article, the date and author are defined in a similar way. *Remember*: Text defined using these *title-making* commands appears only when you use the `\maketitle` command.

3 Incorporating figures from XFIG

Page 101.

LaTeX provides a simple environment for creating pictures, called (not surprisingly) the `picture` environment. It provides circles, lines, boxes, text placement and a few other things.

A coordinate system is used to specify where things go, as if you were drawing on graph paper. For complex drawings you are often forced into using graph paper initially and then converting it (manually) into LaTeX source. This is very inconvenient.

A better approach is to use the XFIG program to produce drawings in an interactive manner, using a mouse and graphic display. Once the drawing is complete it can be converted into an equivalent LaTeX source file and subsequently included within your document. We do not describe here how to use XFIG. Once you have a file `drawing.fig` as output from XFIG, convert it into LaTeX using the `export` menu. The resulting `drawing.latex` file can be included in your document in the following way:

```
\include{transfig}
...
\begin{figure}[htbp]
 \centering                  % center it
 \input{drawing.latex}         % input drawing
 \caption{This is my drawing.}
 \label{pic:drawing}         % for x-refs
\end{figure}
```

Figure 2: Example use of `figure` environment.

The `[htbp]` options show your preferences regarding whether the figure should be placed *h*ere, at the *t*op of the page, the *b*ottom, or on the next free *p*page Note the use within the `figure` environment of the `\caption` command which places a numbered caption beneath the figure; the label `pic:drawing` is assigned this number.

Beware when using XFIG that your drawing isn't wider than an A4 page—you can't enlarge or reduce it.

LaTeX can't reproduce all the features that XFIG can. To get perfect hard-copy you really need to convert into Postscript using XFIG's `export` menu. See the *LaTeX Maths and Graphics* handout and for details and examples of how to add Postscript to LaTeX documents.

4 Automated checking

4.1 Spelling

It's nice to use a spell-checker once in a while to make sure that your document won't contain lots of mistakes when it's finally printed. Checking by hand is almost impossible, so we've installed a clever program to do it for you.

The program is called 'ISPELL' and you can use it from within the EMACS editor by choosing `spell` from the `Edit` menu or typing

```
M-x ispell-buffer
```

which you can use while editing (see the *CUED Local Guide for more details*). We recommend that you don't do it too often as it can be rather slow for large files. ISPELL is clever because it knows that you don't want to check the spelling of LaTeX commands—instead it skips over them, and indeed, any arguments that can safely be ignored. This way you only have to worry about text, not commands. (Of course, ISPELL gets it wrong sometimes...)

4.2 Consistency

LACHECK is general purpose consistency checker for LaTeX documents. To use it, just replace `latex` by `lacheck` on the command line.

5 BibTeX

BibTeX is a program used in conjunction with LaTeX to make it easy to use LaTeX's built-in bibliography and citation mechanisms. When you cite the work of others, the citation often appears appears like this [2], or this [4, 6, 20].[3] To number these manually is a big problem, but not as big as the other problem of typing in the references at the end of your document. Typically, you will compile a large number of references into a *bibliography database*, from which you extract those cited in your document. BibTeX works with LaTeX to *automatically* extract those references you require, *and* puts them into your document, correctly numbered, sorted and formatted. All you have to do is identify each reference within the bibliography database with a unique (and for your own sake, informative) key. You can then use the `\cite{`*key*`}` command to cite that particular reference. When you run LaTeX on your file, it collects all of these keys; you then run BibTeX to scan your bibliography database for them. BibTeX extracts and formats them according to certain styles. By running LaTeX a second time the reference list is incorporated into your document.

For a thorough description of BibTeX and how to use it, refer to "BibTeXing" by Oren Patashnik, which is available in the Department.

5.1 The bibliography database

The database contains a description of all the book, papers, conference proceedings, theses etc. that you've ever read (and probably some you haven't!) Let's take a look at an example database (remember that the BibTeX database is *not* a LaTeX source file).

[3] There are other ways of citing work; see the *CUED Local Guide* for alternatives.

```
@book(Aho86,
       Address = {Reading, Mass., USA},
        Author = {Aho, A. V. and Sethi, and Ullman, J. D.},
     Publisher = {Addison-Wesley},
         Title = {Compilers: Principles, Techniques, and Tools},
          Year = {1986})
@inproceedings(Giunchiglia89,
        Author = {Giunchiglia, Fausto and Walsh, Toby},
     Booktitle = {Proceedings IJCAI-89},
         Title = {Abstract Theorem Proving},
          Year = {1989})
@techreport(Wallen86,
        Author = {Wallen, Lincoln A.},
   Institution = {University of Edinburgh},
        Number = {DAI 310},
         Title = {Using Mathematical Knowledge},
          Year = {1986})
```

Figure 3: Extract from sample bibliography database

This database contains references with the keys `Aho86`, `Giunchiglia89` and `Wallen86`. The keys have been chosen to reflect the authors of the reference, and the year it was published. The '`@`' character denotes the start of a reference, and is followed by the *type* of the entry (there are many possible entries). The *fields* of each entry are things like `Author` and `Year`; different types of entry have different required and allowable fields. The `\bibliographystyle` command indicates the style of the formatting of the references. The standard one is `plain`. See the *CUED Local Guide* for others. Page 140.

5.2 Citing references

The `\cite` command is used to cite a reference within a LaTeX source file. In the LaTeX source file the argument(s) of the `\bibliography` command specify one or more files containing the bibliography database. These files are assumed to have the extension '`.bib`'. For example, the command

```
\bibliography{personal,group}
```

indicates that the references of this document are to be found in either of the bibliography databases `personal.bib` and `group.bib`. Let's assume that one of these databases contains the above three references. The example below illustrates how the cite command is used:

See Wallen [?] for a proof.

```
See Wallen~\cite{Wallen86} for a proof.

\bibliographystyle{plain}
\bibliography{personal,group}
```

Note that the `\bibliography` command incorporates the reference list automatically and titles it. Typically the `\bibliography` command appears at the end of the LaTeX source file.

6 Using LaTeX and BibTeX together

Page 28.

We've looked at what goes into both LaTeX source files (those ending in a '`.tex`' extension), and BibTeX database files (ending in '`.bib`'). Here we take a look at running these programs on these files.

Attention; LaTeX is clever because it allows you to make *forward* citations and cross-references (as we saw in section 2.15) to arbitrary pieces of information. Thus, in order to correctly compile your document LaTeX may need to scan it more than once; this is done *explicitly* by running LaTeX *multiple times* (LaTeX will inform you if this is necessary with an appropriate output message when it runs). Don't be confused when we indicate that LaTeX ought to be run again—it's not a mistake!

We assume the following files exist in the current directory, and that things are set up so you can execute LaTeX and BibTeX. Table 1 describes some of the files that will appear in your directory when you use LaTeX and BibTeX. (You may like to follow this process diagrammatically; see figure 4 on page 13.)

File description	*Filename*
LaTeX source file	`mysource.tex`
BibTeX database	`mydb.bib`
LaTeX log file: reports status of LaTeX execution	`mysource.log`
BibTeX log file: reports status of BibTeX execution	`mysource.blg`
LaTeX output	`mysource.dvi`
LaTeX auxiliary file	`mysource.aux`
BibTeX output	`mysource.bbl`

Table 1: Files generated when LaTeX runs.

The following describes how to arrive at the file `mysource.dvi` which can be then sent off for laser printing. (See the *CUED Local Guide* for more details.)

1. Execute LaTeX on your source file: You can do this using `xlatex` or by typing

   ```
   latex mysource
   ```

   producing some new files (`aux`, `.dvi` and `.log`). The `.log` file is simply a transcript of the LaTeX run, an abbreviated version of which will be shown on your terminal. The `mysource.aux` file includes information about cross-referencing labels and citation keys. If you have used the `\bibliography` command in `mysource.tex` you must use BibTeX to extract the relevant references from `mydb.bib`. (If you haven't, simply go to step 3.)

2. (Omit this step if you haven't used a `\bibliography` command.) Execute BibTeX:

   ```
   bibtex mysource
   ```

   causing BibTeX to read `mysource.aux` in order to find which references from `mydb.bib` you have cited. It compiles these into a bibliography file, `mysource.bbl`. Run LaTeX again, as in step 1 to include this `.bbl` file in your document.

3. If the terminal output of LaTeX (from either step 1 or step 2) indicates you must run LaTeX a second time, do so. (LaTeX will output something like:

   ```
   LaTeX Warning:  ...  Rerun to get cross-references right.
   ```

   to indicate you must do this.) If you're not sure, just run it again anyway, it does no harm.

   You now have a file `mysource.dvi` which describes your document completely. You can't directly read this file, but you can *preview* it to see how your document will look when printed, or you can simply print it. We STRONGLY suggest that you DO preview before printing to ensure that you don't waste paper and other resources by printing out a document containing many errors.

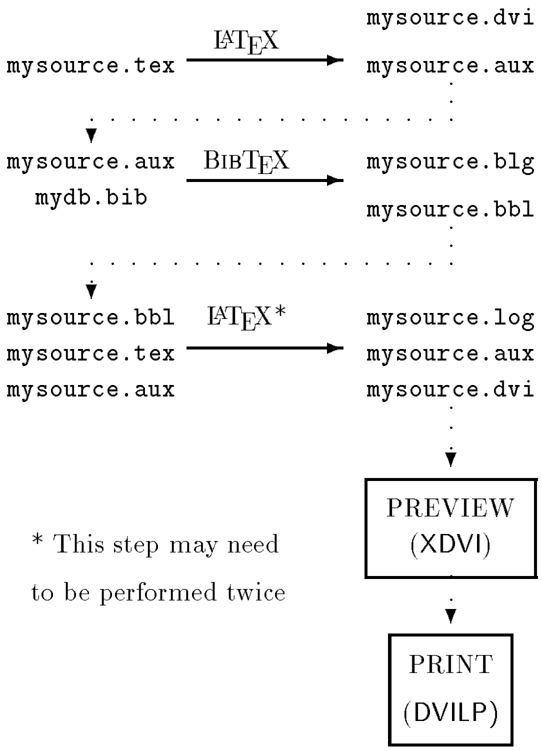

Figure 4: Diagram illustrating document preparation cycle.

7 Appendix: A *Local Guide*

7.1 Introduction

Occasionally the LaTeX Book refers to things that are not part of LaTeX but complementary to it. Such things vary from place to place and between implementations etc., depending upon things like editors, computer hardware, types of laser printers and so on. Because of this, Lamport asks you to look for information on such things in this *Local Guide*. We cover here most of the things that are specific to the Engineering department HP Teaching System here in Cambridge.

7.2 Location of files etc.

Most of the LaTeX/TeX distribution is in the `/export/latex2e` directory and you can browse there to see what appeals to you. Of particular interest is the subdirectory `/inputs`, containing all source files used by LaTeX when it runs. For example, the `article` class is defined in the file `article.clo` in this directory. There are many other files here: see sections 7.7 and 7.11 for examples.

7.3 Running LaTeX on `demo0.tex`

This section describes how to run LaTeX on the file `demo0.tex`. First copy the file into your current directory using

```
cp /export/Examples/LaTeX/demo0.tex demo0.tex
```

See section 7.5 for details of printing `sample.dvi`.

7.3.1 From the command line

Execute the following commands:

```
latex demo0
xdvi demo0
```

7.3.2 Using `xlatex`

Note that there's a program called `xlatex` which is a front end to LaTeX that also provides easy printing and previewing facilities. Do

```
xlatex demo0
```

then use the $\boxed{\text{Process}}$ button to run LaTeX, the $\boxed{\text{View}}$ button to view and the $\boxed{\text{Print}}$ button to print some or all of the document.

7.4 Stopping LaTeX

If LaTeX goes out of control and you can't seem to stop it, try some of the following:[4]

- `\stop` (this only works if LaTeX is giving you its '*' prompt;
- `C-c`, which is the standard UNIX interrupt signal;
- `C-d` which is the end of file signal. You'll find that `C-d` is useful if LaTeX can't find a particular file it thinks it needs. If this occurs, it will prompt you for a different file name, and won't stop until it's satisfied: this is a job for `C-d` to end it all.

7.5 Printing and previewing `.dvi` files

A `.dvi` file is a *device-independent* way of representing documents. All `.dvi` files conform to a standard; for each different device (printer, terminal etc.) there is a converter (called a device driver) between the standard `.dvi` format and the format required by the device.

Normally the devices in question are laser printers but they can also be simple line printers and even graphic displays on workstations. The ability to display your document on the screen of a workstation using high-resolution graphics enables you to *preview* your document before printing it.

> Previewing is a very important phase of document preparation as it reduces your consumption of paper (and other valuable resources), by showing you your mistakes before printing. Please use this facility.

Previewing under X11. To display a `.dvi` file on the screen of a workstation running the X11 windowing system, use the XDVI program:

`xdvi` *dvifile*

Type `man xdvi` for details.

Printing on HP (LaserJet II & 2000) printers. *(Remember: Always preview before printing)*. To print a `.dvi` file you must convert it first into the graphics language that is understood by the LaserJet, and then send this file. The program `lp` does this. You can also use `dvilp`. (Type `man lp` and `man dvilp`.) For example to print `mysource.dvi` on the main LaserJet (`ljmr1`) use:

```
lp -dljmr1 -odvi < mysource.dvi
```

The command `dvilp` is preferable to `lp`; it automatically selects the nearest printer for your document, and you can also give it more than one file to print. For example, to print `chapter1.dvi` and `chapter2.dvi` use

```
dvilp chapter1 chapter2
```

[4] `C-c` means control ' c'.

Using PLOTVIEW. PLOTVIEW is provides an X11 interface the above procedures for previewing and then printing documents. See `man plotview` for more information.

7.6 Editing LaTeX files

The standard editor in the department is EMACS.[5] For LaTeX users, EMACS provides a special "major-mode" called "`LaTeX`", that makes the entry of LaTeX source files less painful. To activate this mode, type:

```
M-x latex-mode
```

within EMACS. For help with this mode type:

```
C-h m
```

from within this mode. For example, `LaTeX` mode can help with the matching of braces and correctly defining environments.

7.7 Packages and package options

As well as the standard classes (`article`, `letter`, `report` and `book`), we have the following packages. They are found in `/export/latex2e/inputs`.

`7pt`, `8pt` and `9pt` Gives a normal text size of 7, 8 or 9 points; use as you would the `11pt` and `12pt` style options. In order to use this, you need to use the `xarticle` document style. (Note that there is no `10pt` option — 10 point is the default.)

`A4paper` This is the 'standard' A4 paper size option at CUED, and gives a reasonable A4 page layout, with room for marginal comments. This option will work correctly with `xarticle` document style also. Using this option, the `\WideMargins` command gives you wider margins that you might need if you make heavy use of marginal notes.

`a4` Changes the standard height of text on a page to look better on A4 size paper.

`a4wide` Same as the `a4` option, but increases width of text also.

`a4_local` Another A4 page package; this is an old version of the `a4` package option. Please use `A4paper` instead.

`a5comb` set A5 page size, with equal spaces to left and right, and twice the amount of centre space (for comb-bound documents).

`a5` set A5 page size, with equal spaces to left, middle and right.

[5] If you find EMACS difficult to use, I recommend you spend more time learning it.

`aaai` This LaTeX style is for papers to be printed in the proceedings of AAAI-88. Prepared by Peter F. Patel-Schneider, liberally using the ideas of other style hackers, including Barbara Beeton. This style is NOT guaranteed to work. It is provided in the hope that it will make the preparation of papers for AAAI-88 easier. (See `aaai-doc.tex` for details.)

`aaai-names` For use with the `aaai` option and the `aaai-named.bst` BibTeX file. Gives citations with names in them.

`agugrl` For American Geophysical Union *Geophysical Research Letters*.

`agujgr` For American Geophysical Union *Journal of Geophysical Research*.

`aip` This document style is designed for manuscripts to be submitted to one of the journals published by the American Institute of Physics.

`alleqno` Makes all displayed equations numbered by default.

`alltt` Defines the '`alltt`' environment, which is like the '`verbatim`' environment except that '`\`', '`\{`', and '`\}`' have their usual meanings. Thus, other commands and environments can appear within an '`alltt`' environment.

`amssymbols` This package option loads up the AMS symbol founts and associated symbol definitions. See the November 1985 issue of TUGBoat for a listing of these symbols. Only certain sizes are catered for. The `10pt`, `11pt` and `12pt` document package options are supported.

`apalike` For the 'apalike' bibliography style.

`array` A new and improved version of LaTeX's `array` and `tabular` environments.

`authordate1-4` For more flexible ways to cite references.

`avantgar` For using the Avant Garde postscript font.

`babel` Support for non-english languages.

`bibunits` Allows bibliographies at the end of chapters, etc.

`bigre` French.

`biihead` For underlined headings.

`bitmap` For including bitmaps.

`bnf` To help with setting Backus-Naur Form.

`bookman` For using the Bookman postscript font.

`boxedminipage` Minipage with box around it.

`breakcites` Allow citations to be broken across lines.

`calc` Does sums.

`captcont` To permit the use of captions in continuations of floats.

`catalan` Support for Catalan.

`cd` Commutative diagram macros modified from $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX.

`changebar` Adds change bars to documents.

`chapterbib` Allow multiple bibliographies in a LATEX document, including items `\cite`'d in more than one bibliography.

`chicago` For the Chicago BibTeX styles chicago.bst and chicagoa.bst.

`chomsky` For setting grammars.

`cite` Good citation format; allows line breaks within citations and compresses consecutive citations into ranges.

`citesort` Sorts citations.

`color` Provides color options.

`courier` For using the Courier postscript font.

`cprog` Use this if you want to (easily) typeset existing C, Pascal or Modula programs.

`croatian` Support for Croatian.

`cropmarks` Helps with cutting and pasting.

`cyrillic` Allow LATEX to use the AMS cyrillic founts.

`czech` Support for Czech.

`c-program` To set C programs.

`doublespace` For doublespacing. (Not recommended.)

`draftcopy` Prints a large grey "DRAFT" diagonally across each page.

`drafthead` Puts "DRAFT" with date and time at top of each page.

`drop` Macro for dropping and enlarging the first letter(s) of a paragraph.

`danish` Support for Danish.

`dutch` Support for Dutch.

`ebar` Histograms. An english version of `bar`. See the example in `/export/Examples/LaTeX`.

`eepicemu` EEPIC emulation.

`eepic` Improved drawing of circles and sloped lines (used in conjunction with drawing packages).

`endfloat` Puts all figures on pages by themselves at the end of an article. Likewise for tables.

`endnotes` Places endnotes (like footnotes) at end of chapters.

`enumerate` Extends standard LaTeX `enumerate` environment. This package option gives the `enumerate` environment an optional argument which determines the style in which the counter is printed.

`epic` Improved drawing.

`epigram` Sets epigrams.

`epsf` Include an encapsulated PostScript file in the current page.

`equations` Help with LaTeX equations; this is worth using if you typeset displayed equations.

`esperant` Support for Esperanto.

`example` Typesets twice, once properly, once verbatim.

`fancyheadings` Fancy headers and footers (as used by this document).

`finnish` Support for Finnish.

`float` To define new classes of floats.

`fnpara` Makes multiple footnotes run together.

`footnpag` Numbers footnotes subsequently per page.

`format` Floating point numbers in fixed format.

`francais` Support for French.

`frontiers` This is a document style option, for camera-ready copy for the Frontiers '88 symposium (and other IEEE conferences), to be used with the `article` document style and the `proc` document style option.

`fullpage` Yet another A4 option.

`galician` Support for Gallician.

`geophysics` For Geophysics, a publication of the Society of Exploration Geophysicists.

`germanb` For writing german texts with TeX or LaTeX.

`graphics` Provides extensive graphics facilities.

`hangcaption` Defines a variant of the caption command to produce captions with hanging indentation.

`headerfooter` To do headers and footers.

`helvetic` For using the Helvetica postscript font.

`here` Gives environments such as `figure` and `table` an `[H]` option which means 'PUT IT HERE'. As opposed to the standard `[h]` option which means 'You may put it here if you wish'.

`histogr` Histograms within LaTeX `\picture` environment.

`icassp` Style option for camera-ready copies the ICASSP '89 conference (and other IEEE conferences), to be used with the `article` document style and the `proc` document style option.

`ijcai89` For IJCAI conference submissions.

`indentfirst` Always indent a paragraph's first line.

`ipa` Phonetic fonts.

`ist21` This style option produces a title appropriate for BSI IST/21 documents, and general memos.

`isucaption` Clever captions for figures.

`italian` Support for Italian.

`jbs` Article document style for the *Journal of Business Strategies*.

`jeep` Useful style file that slightly modifies the standard document classes of LaTeX.

`jims` A class for Judge Institute Management Studies working papers.

`jmb` Journal of Molecular Biology style.

`lambda` λ-calculus stuff.

`layout` Shows the various page dimensions

`letterspace` Puts space between letters.

`lcustom` This package contains a whole bunch of useful little command definitions. The `\inputverbatim{`*file*`}` command inserts the file in verbatim mode. However, it is usually better to use the `\lgrind` package described later in this section for this purpose. A number of further commands provide a clean way of introducing diagrams and tables from subsidiary files.

`lgrind` For formatting source code. Type `man lgrind` for details.

`llncs` A Springer-Verlag style

`longtable` Alternative tables on separate page.

`lscape` Makes the production of landscape pages easier.

`ltugboat` Typesetting articles for TUGBoat.

`magyar` Support for Hungarian.

`manual` Style for manuals (similar to a book).

`memo` For memos.

`merge` Combines letter bodies with lots of addresses.

`mitpress` MIT Press format.

`mitthesis` MIT Thesis format. item [moreverb] More verbatim options (*e.g.* line numering).

multibox Makes boxes in LaTeX's picture environment.

multicol For multiple page columns.

multirow For spanning multiple rows of a table.

multind Style option for producing multiple indexes.

named Named bibliography style.

nar Nucleic Acid Research style.

natsci Correctly formats citations in the NATSCI bibliography style.

nature For writing articles to Nature.

newcentu For using the New Century postscript font.

nofm For 'n of m' style pagination.

nomathitalic Make math mode use normal italic and not math italic.

nopagenumbers A tiny package file to suppress page numbers on LaTeX pages.

norsk Support for Norwegian.

numline Puts line numbers on output.

openbib Open style bibliographies.

optima For using the Optima postscript font.

overcite Cites are superscripts rather than bracketted numbers.

palatino For using the Palatino postscript font.

pifont Makes use of the Zapf Ding-bats font easier.

polish Support for Polish.

portland For switching between portrait and landscape mode of printing.

portuges Support for Portuguese.

program For typesetting programs.

proof Typesetting proof trees.

remark For remark environments.

resume For formatting resumes.

res Ditto.

romanian Support for Romanian.

romanneg Negative roman numbers

rotating Provides a few more rotation options than the graphics package provides.

sc21 This style option produces a cover page appropriate for documents conforming to the ISO/TC97/SC21 standard.

schedule Defines macros to draw a schedule.

seceqn Package option for use with article style to cause equations to be numbered within sections.

seminar For producing OHP slides with notes.

shapepar For producing paragraphs in certain shapes (diamonds etc).

sfwmac Defines a set of macros which are useful when writing program documentation. The \pgm{*program*} command expects a program name as an argument. The name in printed (in italics) and an index entry is made for it. Similarly, the \man{*program*}(*section*) command may be used to produce references to manual page entries. The \arg{*argument*}, \switch{*switch*} and \file{*file*} commands may be used when referring to program arguments and switches and to filenames. These produce no index entry.

showlabels Change \label, \ref, and \pageref to show labels in line. (For debugging.)

siam For SIAM journals. (See other files for documentation.)

slem Makes \em use \sl rather than \it.

slovak Support for Slovakian.

slovene Support for Slovenian.

spacecites Space out citations in the report style.

spanish Support for Spanish.

sshdbk10 Sans serif headings for book style.

subeqnarray Renumbering of sub-arrays in math-mode.

subeqn Different numbering sub-arrays.

subfigure Provides figures within figures.

supertabular Much like longtable.

suthesis Stanford University Thesis style.

svlncs For Springer-Verlag's Lecture Notes in Computer Science.

swedish Support for Swedish.

tables Useful for tables.

texnames Defines things like $\mathcal{AMS}$-TEX and BIBTEX.

tgrind It is useful to be able to include program fragments as figures in a text. The tgrind package option can be used to accomplish this fairly painlessly (but see also lgrind) by defining the \tagrind command.

The \tagrind command has the following proforma:

\tagrind*[*loc*]{*file*}{*caption* }{*label* }

As usual,

- in two-column format, the ordinary form produces a single-column listing and the ∗-forms produces a double-column listing;
- *loc* is a sequence of one to four of the letters h (*here*), t (*top*), b (*bottom*) and p (*page*) specifying where the listing may be placed.
- *file* specifies the file from which the ground listing is to be taken.
- *caption* produces a numbered caption.
- *label* assigns a label to the listing, so that it may be referred to by means to the \ref{*label*} command.

The listing file should be generated using tgrind and removing the first and last lines of the resulting file. For example, if the C source file file.c were to be used as a listing,

```
tgrind -f -lc file.c | sed -e 1d -e \$d >file.tex
```

would generate a file file.tex suitable for use with tagrind. For example,

```
\tagrind{file}{This is an example}{eg}
```

NB: lgrind is worth a look too.

threeparttable Creates tables with titles, the table itself, and notes.

times For using the Times postscript font.

trademarks Defines a number of trademarks and trade names. For example, UNIX[6] (note the automatically defined footnote) was produced by a \Unix/ command. Note that there is no need to append \␣ or {} to prevent the swallowing up of succeeding spaces.

trees Makes tree-like pictures.

turkish Support for Turkish.

ucthesis University of California Thesis style.

ukdate Gives an English-style date.

ulem Replaces italics with underlining in emphasized text

undhead Underline page headings

univer For using the Universe postscript font.

utopia For using the Utopia postscript font.

verbatim Produces extra verbatim options

version Version control macros.

vdm The vdm package option aids typesetting of VDM specifications: it provides macros for typesetting formulae, data types, functions, operations and proofs in whatever the currently approved VDM manner happens to be.

version Version control macros. These let you define environments whose contents will be optionally added to or deleted from the text when you run LaTeX.

wrapfig Produces figures which text can flow around. See the example in /export/Examples/LaTeX.

xarticle Allows use of the 7pt, 8pt and 9pt style options. If you use xarticle, you may run into problems with other options.

[6] UNIX is a trademark of AT&T Bell Laboratories.

If you want to use these files it's a good idea to read through them first, as most have some explanatory comments in them. Many of them have documentation files of the form '*style-name*`.doc`' or '`*`*style-name*`-doc.txt`'. Check for these in the documentation directory, `/export/latex2e/doc`, and also the inputs directory.

Remember, class is selected using the `\documentclass` command. Packages can be added using the `\addpackage` command. You should be able to use the packages with any class. Note that the majority of these packages have not been tried (not to mention tested) by the author, so exercise caution...you have been warned!

7.8 Other languages

The `babel` package provides support for other languages, because (as yet) we don't have a version of LaTeX that will correctly hyphenate anything other than (American) English. The quality of the hyphenation will vary according to the language.

7.9 Departmental letterheads

The standard LaTeX `letter` style is rather boring and doesn't have a nice CUED letterhead, so one has been installed. The class is called `deptletter`, and is used instead of the standard `letter` class:

```
\documentclass[...]{deptletter}
```

This class places the University crest in the top-left of the page, and your name and the Department's address in the top-right. You can use the `handouts` command (see section 8) to view this letter style on-line. This is by no means an officially endorsed letterhead, and I do not claim it to be so. The format is customizable with the following commands:

Command	*Specifies*
`\address {`*address*`}`	The Department address.
`\telephone {`*number*`}`	Your full telephone number.
`\fax {`*number*`}`	Department fax number.
`\email {`*net address*`}`	Your email address.
`\name {`*name*`}`	Your name.

Table 2: Commands to configure the `deptletter` document style.

Use these commands in your own letters to change them. For example I might use:

```
\telephone {\small\sl Tel.:\/ \rm \up[+44\up] 223 33\thinspace 2639}
\fax       {\small\sl Fax:\/ \rm \up[+44\up] 223 33\thinspace 2662}
\email     {\small\sl email:\/ \rm \texttt{img}@\texttt{eng.cam.ac.uk}}
\name      {Ian Green}
```

The `\address` command has as default

```
\address {\sc Department of Engineering\\
          \sc Trumpington Street\\
          \sc Cambridge, CB2 1PZ\\
          \sc England}
```

so you probably won't need to change this.

7.10 Using lablst.tex and idx.tex

7.10.1 lablst.tex

`lablst.tex` is a utility program written in LaTeX which lists of all the arguments of `\label` commands in a document. To use it, simply type

```
latex lablst
```

and you'll be prompted for the name of the file you want to search for labels and the style of that file (*i.e.* the argument to the `\documentstyle` command. Beforehand you must run LaTeX on the file you wish to scan. Another point is that the `lablst` program doesn't expect you to type the extension of your filename (the `.tex` bit), so don't. For example, finding the labels in the file `sem.tex`:

```
latex sem
latex lablst
...
\filename=sem
...
\docsty=article
```

This produces a file `lablst.dvi` that you can preview in order view the list of labels used.

7.10.2 idx.tex

The process of obtaining a listing of all indexed entries (*i.e.*, arguments to `\index` commands) in a LaTeX file is similar to the above for labels, using `idx` instead of `lablst`. You enter the name of the file (without the extension) and it produces the file `idx.dvi` which you can preview to obtain a list of all index entries.

7.11 Bibliography styles

Here we describe some of the bibliography style files (`.bst`) we have. Note that many of them are associated with a corresponding `.sty` file, and they may need to be used in conjunction with one another. There are some however that can be used 'stand-alone'. We do not mention here the standard LaTeX `.bst` files.

`aaai-named.bst` Designed for use with `aaai.sty` for AAAI publications. Can be used without. (See this file for details.)

`acm.bst` ACM Transactions bibliography style. Like `abbrv.bst`, but names come out "Last, initials", and in `\sc`. Some dates are parenthesized.

`apalike.bst` This style should be used with the '`apalike`' LaTeX style. `\cite`'s come out like "(Jones, 1986)" in the text but there are no labels in the bibliography, and something like "(1986)" comes out immediately after the author. Author (and editor) names appear as last name, comma, initials. A '`year`' field is required for every entry, and so is either an author (or in some cases, an editor) field or a key field.

`ieeetr.bst` IEEE Transactions bibliography style. Otherwise, much like the standard, `plain.bst`, from which it is adapted.

`local_plain.bst` Variation on `plain.bst` style.

`local_sc_plain.bst` Variation on `plain.bst` style.

`local_sc_plain2.bst` Variation on `plain.bst` style.

`named.bst` Variation on `aaai-named.bst`.

`physics.bst` This is a good starting point if you want to design your own `.bst` file. It requires you to know what you're doing.

`siam.bst` For use with `siam.sty`. Numeric labels, alphabetic order, Mathematical Reviews abbreviations, names in `\sc`, titles in italics, book titles mixed upper-lower and article titles lowercase, commas separate all fields except before 'notes'.

To use bibliography style *this-style* put the command

`\bibliographystyle{`*this-style*`}`

in your document (not in the preamble).

A comment If you want to write your own `.bst` file, then go ahead. I might warn you that it can take some time, so you ought to make sure it's time well spent. Also, try to modify an existing one rather than start from scratch.

7.12 Compiling an Index

Compiling an index to your document is hard work, but the MAKEINDEX program is available to help. I refer potential index-builders to the (on-line) documentation for this program, "*MakeIndex*: An Index Processor For LaTeX" by Leslie Lamport, which is available via the `handouts` comamnd.

7.13 EMACS BIBTEX support

When EMACS is asked to edit a `*.bib` file it goes into `bibtex` mode. This, amongst other things, provides templates for entries with the appropriate optional and obligatory fields.

Table 3 on page 28 lists the predefined abbreviations of journals etc. in the `plain` bibliography style. (To add your own, you can ask Tim Love to add them to the file, or copy the file, rename it, edit it[7] and use that one instead.)

Abbreviation	*Expansion*
jan	January
feb	February
⋮	⋮
dec	December
acmcs	ACM Computing Surveys
acta	Acta Informatica
cacm	Communications of the ACM
ibmjrd	IBM Journal of Research and Development
ibmsj	IBM Systems Journal
ieeese	IEEE Transactions on Software Engineering
ieeetc	IEEE Transactions on Computers
ieeetcad	IEEE Transactions on Computer-Aided Design of Integrated Circuits
ipl	Information Processing Letters
jacm	Journal of the ACM
jcss	Journal of Computer and System Sciences
scp	Science of Computer Programming
sicomp	SIAM Journal on Computing
tocs	ACM Transactions on Computer Systems
tods	ACM Transactions on Database Systems
tog	ACM Transactions on Graphics
toms	ACM Transactions on Mathematical Software
toois	ACM Transactions on Office Information Systems
toplas	ACM Transactions on Programming Languages and Systems
tcs	Theoretical Computer Science

Table 3: List of abbreviations defined in plain.bst.

8 Additional sources of information

For a general introduction to LaTeX, including the new features of LaTeX 2_ε, you should read *The LaTeX Book* by Leslie Lamport.

A more detailed description of the new features of LaTeX, including an overview of more than 150 packages, is to be found in *The LaTeX Companion* by Michel Goossens, Frank Mittelbach and Alexander Samarin.

Some examples are in `/export/Examples/LaTeX`. The material shown in table 4 on page 29 may also be of use. It's available by the machine room and

[7]They are called 'MACROS' in the `.bst` file, the syntax is obvious.

on-line using the `handouts` command on the Teaching System. If possible, you should use the on-line version of these manuals: they will be more up to date, there are far more of them, and you save paper.

Name	*Title*	*Author(s)*
`bibtex`	BibTeXing	Oren Patashnik
`dvips`	DVIPS: A TeX Driver	Tomas Rokicki
`exletter`	(an example letter)	Prof. Grantgetter, III
`harvard`	The Harvard Family of Bibliography Styles	Peter Williams
`latex_basic`	Word Processing Using LaTeX	Richard Prager & Tim Love
`latex_advanced`	Advanced LaTeX	Tim Love
`latex_maths+pix`	LaTeX Maths and Graphics	Tim Love
`latex_gnuplot`	LaTeX and the GNUPLOT Plotting Program	David Kotz
`local_LaTeX`	C. U. E. D. Document Preparation Facilities and Local Guide	Ian Green & Tim Love
`makeindex`	*MakeIndex*: An Index Processor For LaTeX	Leslie Lamport
`sanskrit`	Using Devanagari fonts	Frans J. Velthuis
`seminar`	The seminar style	van Zandt
`longtable`	The longtable package	David Carlisle
`clsguide`	LaTeX 2_ε for class and package writers	The LaTeX3 Project
`fntguide`	LaTeX 2_ε font selection	The LaTeX3 Project
`usrguide`	LaTeX 2_ε for authors	The LaTeX3 Project

Table 4: Hardcopy documents available