Advanced LaTeX

Tim Love

September 11, 1997

This document follows on from the *Word processing using LaTeX*[1] document. It describes the features of LaTeX 2_ε that people at CUED are most likely to use. Further information is available from the LaTeX help page[2] and in the books available for loan from the operators in the `DPO`. The *Local LaTeX*[3] document is a companion to this one. It describes some features especially suited to writing theses (bibliography and index compilation, for example) and includes a list of available packages.

Comments and bug reports to Tim Love (tpl@eng.cam.ac.uk).

Contents

[1] http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex_basic/latex_basic.html
[2] http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/LaTeX_intro.html
[3] http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/local_LaTeX/local_LaTeX.html

5.5 An Example

`minipage` creates a miniature page complete with its own footnotes etc. It can be created of any set width. Footnotes within a minipage use a different counter to other footnotes[13]. This example ensures that all the numbers follow on in sequence. The footnotes in minipages are marked by letters rather than numbers, so here the type is changed to arabic.

```
\begin{minipage}{\textwidth}
% Set the minipage footnote counter
\setcounter{mpfootnote}{\value{footnote}}
% Redefine the command that produces the footnote number
\renewcommand{\thempfootnote}{\arabic{mpfootnote}}
\begin{tabular}{|ll|}\hline
one & two\footnote{A minipage footnote}\\\hline
\end{tabular}
\setcounter{footnote}{\value{mpfootnote}}
\end{minipage}
```

one	two[14]

[14]A minipage footnote

You might find this trick useful if you want footnotes in tables. They don't come out otherwise.[15]

Here's an example of using a package. The `multicol` file lets one change the number of columns easily. To switch into 3-column text use

```
\begin{multicols}{3}{
Put the text here. Maths, tables, pictures etc are all ok, but not
figures. But you have to remember to load in the \texttt{multicol}
package at the top of your document.
}
\end{multicols}
```

Put the text here. Maths, tables, pictures etc are all ok, but not figures. But you have to remember to load in the `multicol` package at the top of your document.

LaTeX uses the value of the environmental variable `TEXINPUTS` to decide where to look for `sty` files. If you create a directory called (say) `inputs`, copy a system `*.sty` file into it and do

```
export TEXINPUTS=inputs:/opt/latex/inputs:.
```

then your copy will be read in preference to the system one and you can customise easily.

6 More Information

- See the LaTeX help page[16]. Much of the material available from there isn't available in printed form. The guides to producing Posters and booklets Re-

[13]like this one
[15]Not for me anyway
[16]http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/LaTeX_intro.html

The related `\itshape` command doesn't take an argument. It affects all the subsequent text in the environment it's used in.

LaTeX tries to enforce the idea that the visual appearance of the document (use of fonts, indentation, etc) should derive from the logical structure of the document (*i.e.* rather than manually putting the section titles into bold, you should let the `\section` command do it). Resisting this philosophy can lead to extra, usually unnecessary work.

LaTeX is expandable. Many macros can be loaded in to provide added featues. You can also create your own commands and environments. Commands can take arguments that modify their action

- Some commands have a `*-form`, a variant on the standard command that you get by adding a `*` to the command name.
- Mandatory arguments are enclosed in `{}` braces
- Optional arguments are enclosed in `[ ]` brackets.

1.2 Classes and packages

At the top of your file you will have a line something like

```
\documentclass[12pt]{article}
```

which determines the font size and document class – the type of document you're writing. Each class has an associated `*.cls` file in the directory `/opt/latex/inputs/` (on our system) which is read in at start-up time. Other options accepted by `article` include `10pt`, `11pt` and `twocolumn`.

Then you'll probably need to load in extra macros with the `\usepackage` command. Each package has an associated `*.sty` file in `/opt/latex/inputs/`. The packages inherit the options from the `\documentclass` line and can be given others of their own. *E.g.*

```
\usepackage[dvips]{graphics}
\usepackage[dvips]{color}
```

tells LaTeX that you want to use the extra `graphics` and `color` macros, and that you are going to use `dvips` to convert the resulting file to postscript. You can shorten this to

```
\usepackage[dvips]{graphics,color}
```

1.3 Errors

Errors can be reported either from the LaTeX phase (in which case it is reported as such) or the lower level TeX phase. Don't be too put off by the half-digested text displayed. A line number is reported which usually helps to detect the error. Typing `` `h' `` sometimes produces helpful diagnostics. The most common errors reported are

- a command misspelt
- a mismatched brace

4.10 Quote, abstract

These widen the margins and change the font. The `abstract` environment also adds a title.

4.11 Letter

See `letter.tex` in `/export/Examples/LaTeX`

4.12 Curriculum Vitae

See `cv.tex` in `/export/Examples/LaTeX`

5 Customising

5.1 Macros

At the top of the source of this file is

```
\def\xdt{$\cal X\!\!$\texttt{.desktop}}
```

which defines `\xdt` to be $\mathcal{X}$`.desktop`. Using such constructions can make your document much tidier, and saves on typing.

5.2 Modifications

Many of the features of a LaTeX document are easily customised, but you'll often have to look at the class (`.cls`) files to find out what to do. For example, suppose you wanted to have **References** in a book rather than **Bibliography**. If you look in `book.cls` you'll see

```
\newcommand\bibname{Bibliography}
```

so adding

```
\renewcommand{\bibname}{References}
```

to your file should achieve what you want.

Counters (*e.g.* `figure`) have related commands (*e.g.* `thefigure`) to control their appearance, so they're easy to customise

```
\renewcommand\thefigure{\roman{figure}}
```

produces figure numbers in lower case roman numerals. Longer commands can be adapted too. Remember however, that if the command involves a `@` you have to enclose your changes in `\makeatletter` ... `\makeatother`. Here's an example that changes the appearance of section headings -

```
\makeatletter
\renewcommand{\section}{\@startsection{section}{1}{0mm}
```


1.5 How to use LaTeX at CUED

From the command line – After running `latex` at least twice, you should be able to preview your `*.dvi` document using `xdvi` and print it using `plotview`, but if postscript is implicated at all (if you load in graphics, use postscript fonts, scale, use color, or rotate) then you should convert your `*.dvi` document to postscript. A typical sequence of commands to process `doc.tex` would be

```
latex doc
latex doc
dvips doc.dvi -o doc.ps
ghostview doc.ps
lp -dljmr1 -opostscript doc.ps
```

Using `xlatex` – `xlatex` has buttons to process, preview and print your document (or selected pages of it), and convert it to postscript. Just type `xlatex` *filename*.

Writing a LaTeX document is rather like writing a program. This makes using LaTeX more difficult in some respects than using a word processor, but there are advantages too. For instance creating a table of contents is trivial. Beginners often use unnecessary '`\\`' sequences and write '`{\large \textbf{2.1 Method}}\\`' when '`\subsection{Method}`' would be much better. Users who think they know more about typesetting than LaTeX (those who, for example, like underlining) will waste a lot of time too.

Avoid repeating constructions. Instead, write your own macros and commands, and familiarise yourself with the packages described in the Local LaTeX[4] guide.

2 Document structure

2.1 Counters and Length parameters

- **Counters** :- LaTeX maintains many counter variables (*e.g.* `page`, `part`, `equation`, `footnote`, textttchapter, textttparagraph, textttsection, textttsubsection, textttsubsubsection, textttenumi, etc). You can set these counters yourself. Some examples:-

  ```
  \setcounter{page}{0}
  \addtocounter{chapter}{2}
  ```

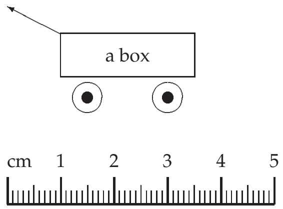
- **Length Parameters** :- LaTeX accepts the following units of length: `in, cm, mm, pt` (there are 72.27 pts to an inch), `em` (width of an `M`), `ex` (height of an `x`). These units can be used to set the values of length variables using `\setlength`. For example,

  ```
  \setlength{\parindent}{0in}
  ```

  sets to zero the amount by which the first line of a paragraph is indented. Other useful length parameters are:-

[4]http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/local_LaTeX/local_LaTeX.html

```
\multiput(5,0)(10,0){5}{\line(0,1){3,5}}
\thicklines
\put(0,0){\line(1,0){50}}
\multiput(0,0)(10,0){6}{\line(0,1){5}}
\end{picture}
```



4.6 Maths

Maths is dealt with in the *LaTeX Maths and Graphics*[11] document. Here are some examples

```
\begin{math}
 \lim_{n \rightarrow \infty}x = 0 \\
x^{2y} \\
x_{2y} \\
x^{2y}_{1} \\
\frac{x+y}{1 + \frac{1}{n+1}} \\
\end{math}
```

produces $\lim_{n\rightarrow\infty} x = 0$
x^{2y}
x_{2y}
x^{2y}_{1}
$\frac{x+y}{1+\frac{1}{n+1}}$

4.7 Figures

To leave a space for graphics, use the `figure` environment. The arguments to figure specify where the space will be made, preferentially

`h`	here
`t`	top of page
`b`	bottom of page
`p`	on a page with no text

```
\begin{figure}[htbp]
   \vspace{1cm}
   \caption{1 cm of space}
\end{figure}
```

[11] http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex_maths+pix/latex_maths+pix.html


```
\addtolength{\evensidemargin}{-1cm}
\addtolength{\oddsidemargin}{-1cm}
\addtolength{\textwidth}{2cm}
```

Further support for control of page layout is provided by the geometry[5] package. To see the current values of these dimensions, use the `layout` package, which defines a `\layout` command.

For A3 output, add `\usepackage{a3}` to your latex source file, run latex on it, convert the resulting file to Postscript using `dvips -t a3` ... then print using `lp -oa3 -dljmr2` ... .

- **Title Pages** :- The title page of this document was created by the following LaTeX commands.

```
\title{Advanced \LaTeX}
\author{Tim Love}
\date{\today}
\maketitle
```

- **Contents** :- Use `\tableofcontents` to create a contents list at that point in the document. LaTeX will pick out the sections, subsections etc for you. You'll have to run LaTeX at least twice though.

- **Page numbers and Headings** :- These are determined by the argument given in `pagestyle{}`.

  `plain` :- the page numbers are put at the bottom of the page. The top of the page is empty. This is the default mode.

  `empty` :- this suppresses page numbering altogether, except on the title page it you're using `\maketitle`. The workaround in this case is to do

  ```
  \maketitle
  \thispagestyle{empty}
  ```

  with no gap between the two lines.

  `headings` :- this puts the page numbers at the top and adds a header whose contents depend on the document style.

  `fancyhdr`[6] is a popular package that adds useful page headers when `\pagestyle{fancy}` is used. This handout uses it.

  Long section titles can cause trouble in headers. The section commands let you specify an extra, shorter title for use in the header and contents page. Section 2.3 was specified as follows

  ```
  \subsection[Pagebreaks, footnotes, etc]
     {Pagebreaks, space, footnotes, references, boxes, etc}
  ```

- **Sectioning** :- To start a chapter called **Life** in a `book`, just use `\chapter{Life}`. Similar commands to start a `part`, `section` or `subsection` also exist in most document classes (articles don't have chapters or parts though).

  If you use the `*-form` of the command then the sections will not be numbered, neither will it appear in the table of contents.

[5] http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/geometry.dvi
[6] http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/fancyhdr.dvi

4.3 Tabular

Tabular output is supported. When you create the environment you specify how many columns to have and how the contents are to be aligned (use `l`, `c` or `r` to represent each column with either left, center or right alignment) and where you want vertical lines (use |). The contents of the columns are separated by a '&' and rows by `\\`. Here's a simple example

```
\begin{tabular}{l|c|r}
left & centre & right\\
more left & more centre & more right\\
\end{tabular}
```

left	centre	right
more left	more centre	more right

To draw a full horizontal line, use `\hline` otherwise draw a line across selected columns using `\cline`. The `\multicolumn` command allows items to span columns. It takes as its first argument the number of columns to span. The following, more complicated example shows how to use these facilities.

```
\begin{tabular}{||l|lr||} \hline
\textbf{Veg}  & \multicolumn{2}{|c||}{\textbf{Detail}}\\\hline
carrots    & per pound & \pounds 0.75 \\ \cline{2-3}
           & each      &         20p  \\ \hline
mushrooms  & dozen     &         86p  \\ \cline{1-1} \cline{3-3}
toadstools & pick your own & free     \\ \hline
\end{tabular}
```

Veg	**Detail**	
carrots	per pound	£0.75
	each	20p
mushrooms	dozen	86p
toadstools	pick your own	free

Tables won't continue on the next page if they're too long. The `longtable` or `supertabular` commands are needed to do this. See the Supertabular[8] document for details and examples.

If the text in a column is too wide for the page, LaTeX won't automatically text-wrap. Using `p{5cm}` instead of `c`, `l` or `r` in the `tabular` line will wrap-around the text in a 5 cm wide column.

There are various packages to assist with table creation. The `array` package adds some helpful features, including the ability to add formatting commands that control a whole column at a time, like so

```
\begin{tabular}{>{\ttfamily}l>{\scshape}c>{\Large}r}
      Text  & More Text & Large Text\\
      Left  & Centred   & Right
\end{tabular}
```

`Text`	MORE TEXT	Large Text
`Left`	CENTRED	Right

[8] http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/supertab.dvi

- **Cross References** :- At the end of this source file is the line

  `\label{THE_END}`

  `\pageref{THE_END}` will refer to the final page by number, and `\ref{THE_END}` will refer to it by section number. The last page (page 23) is in section 6. You'll have to run LaTeX at least twice to pick up forward references like these.

- **Boxed Text** :- For short pieces of text, use `\fbox`

  `Help. I'm \fbox{trapped}`

  Help. I'm $\boxed{\text{trapped}}$

- **Comments** :- Anything to the right of a % character is ignored by LaTeX .

- **Rules** :- The `\rule` takes 2 arguments; a width and a height, so `\rule{\textwidth}{1pt}` produces

---

3 Color and Fonts

3.1 Colored text

LaTeX 2_ε introduced some commands to control foreground and background colors. To use them you need

`\usepackage[dvips]{color}`

after the `\documentclass` line but before `\begin{document}`.

- `\textcolor{`*colorname*`}{`*text*`}` writes text in a color which can be specified by name (`black, white, red, green, blue` or a color name you've defined), RGB components, or grayscale.
- `\colorbox{`*colorname*`}{`*text*`}` writes text in a box with a colored background.
- `\fcolorbox{`*colorname*`}{`*text*`}` writes text in a colored frame.
- `\pagecolor{`*colorname*`}` sets the color of the page's background.
- `\definecolor{`*colorname*`}{`*color specification*`}` lets you define new color names.

```
\definecolor{gold}{rgb}{0.85,.66,0}
This is in \textcolor{red}{red} and this box is \colorbox{gold}{gold}.
Text color can be set using RGB values
(\textcolor[rgb]{0,1,0}{like so}), or \textcolor[gray]{0.2}{shades}
\textcolor[gray]{0.5}{of} \textcolor[gray]{0.8}{grey}.
```

produces

This is in red and this box is gold. Text color can be set using RGB values (like so), or shades of grey.

Each of these commands sets one of the font attributes; `\fontsize` also sets `\baselineskip`. The actual font in use is not altered by these commands, but the current attributes are used to determine which font to use after the next `\selectfont` command.

`\selectfont` selects a text font, based on the current values of the font attributes. There *must* be a `\selectfont` command immediately after any settings of the font parameters by (some of) the five `\font<parameter>` commands, before any following text. For example, it is legal to say:

```
\fontfamily{ptm}\fontseries{b}\selectfont Some text.
```

to select bold Times Roman, but it is *not* legal to say:

```
\fontfamily{ptm} Some \fontseries{b}\selectfont text.
```

`\usefont`{*encoding*}{*family*}{*series*}{*shape*}
is short hand for the equivalent `\font`... commands followed by `\selectfont`.

4 Environments

Some examples of how to use environments are given here.

4.1 Alignments

In these environments `\\` starts a new line.

```
\begin{flushleft}
Some people like to stay firmly\\ on the left whereas others
\end{flushleft}
\begin{flushright}
feel much more at home\\ on the right.\\
\end{flushright}
\begin{center}
but most of us prefer to stay dead in the center.
\end{center}
```

Some people like to stay firmly
on the left whereas others

feel much more at home
on the right.

but most of us prefer to stay dead in the center.

4.2 Listing Items

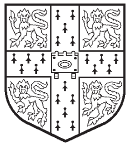
The items can be marked in one of three way:

```
\begin{itemize}
\item just by a bullet, using \texttt{itemize}
```


♣	`\clubsuit`	♢	`\diamondsuit`	♡	`\heartsuit`
♠	`\spadesuit`	$\sum$	`\sum`	$\prod$	`\prod`
$\coprod$	`\coprod`	$\int$	`\int`	$\oint$	`\oint`

One locally produced font with only one character is called `crest`.

```
\font\crest=crest
``This is the {\crest A} CUED crest''
```

produces “This is the  CUED crest”

3.3 Font Sizes

These are the available sizes.

tiny scriptsize footnotesize small normalsize
large Large LARGE huge Huge

If, for example, you want to use the smallest size, do

```
{\tiny ... }
```

If `Huge` isn't big enough for you, you can scale a *postscript* font up using the commands designed for graphics. `\resizebox{!}{5cm}{BIG}` produces

BIG

3.4 Font Types

Independent of size, these font types are at your disposal :- `\textrm` (roman), `textit` *(italic)*, `\textsc` (SMALL CAPS), `\emph` *(emphasis, but note that if you use* emph *within emphasized text, you will get* roman text*)*, `\textsl` *(slanting)*, `\texttt` `(teletype)`, `\textbf` **(boldface)**, `\textsf` (sans serif). As long as there's no conflict, these commands can be combined so that, for instance, **this is bold sans serif** can be produced by `\textsf{\textbf{this is bold sans serif}}`.