Please write your report using LaTeX

David J.C. MacKay
Darwin College
mackay@mrao.cam.ac.uk

January 14, 1997

Abstract

This is a sample LaTeX document. You might wish to use it as a starting point for your document. I give examples of basic LaTeX writing, including using a `.bib` file.

Unlike most books on LaTeX, I explain how to use `psfig` to include figures from `gnuplot`, and how to use a makefile to process and preview your document.

1 Introduction

LaTeX is *the* typesetting program to use in scientific publishing. It is good for writing anything from a one page letter to a two hundred page book. You write your document in a file with a name that ends in `.tex` using your favourite text editor. The file that made this document is called `please.tex`. I'd encourage you to find a copy of this file, because I'm not going to write out how to make a title, how to make section headings, etc. — it's easier for you to just look at this example file. On `mraos`, type the following:

```
mkdir ~/tex
cp /home/mraos/ftp/pub/mackay/latex_demo/* ~/tex
```

If you don't have an account on mraos, then you can get the files by anonymous ftp from `131.111.48.8`. You then have everything you need to see how the document was produced, and produce it for yourself. The instructions given work on `mraos`, and on many other `unix` systems.

1.1 Prose

Where your document consists of prose, you just type prose in your editor, and LaTeX typesets it for you.

A blank line starts a new paragraph. So prose is easy to do. If you want to change font for a few words then you use a command like `\bf` for **bold face** writing, enclosing the relevant words in braces.[1]

[1] Take a look at the file `please.tex` while you are reading this, so you can see what commands generated what.

1.2 More exciting things

Where LaTeX really gets cool is the organisation of the logical structure of the document, and the typesetting of equations. Notice I put a footnote a little while ago. That footnote was numbered 1 automatically, and I am able to *refer* to it as footnote "1" by using a *label* that I stuck in the footnote. Similarly, sections, subsections, and equations are automatically numbered, and can be labelled and automatically referred to. We will see an example of this in section 2. This is especially useful when you want to say something like:

> We substitute equation (5) into equation (1) using equation (2).

It is useful because when you rewrite the paper and include an equation before any of equations (5), (1), and (2), the references to those equations in the text are all automatically changed.

Other good features of LaTeX are:

1. It is easy to enumerate lists of things.
2. Enumerated lists are typeset beautifully.
3. Furthermore,
   (a) You can have lists within lists.
   (b) Those sublists are numbered in a sensible manner.

In the `please.tex` file I laid out the above list in a nice logical way, but this is not obligatory. LaTeX doesn't care how you lay out the `.tex` file at all in terms of white space. Only if there is an entire blank line is it possible that LaTeX cares. Blank lines start new paragraphs, as I said in section 1.1.

You can also itemize things without numbering them if you want:

- The first item.
- The second item.

2 Equations

In the text, you can have mathematical symbols enclosed by $ symbols, such as the constant α which multiplies x and y. Subscripts and superscripts are easy: $a_1x^2 + a_2x + a_3 = c_0^2$. If there is more than one character in your subscript then you put it braces: $m_\alpha = p_{i|j}$. You can also display equations:

$$\int P(p_i)p_i > \frac{F_i}{\sum_{i'} F_{i'}} \tag{1}$$

Notice how LaTeX changes the typesetting of a mathematical expression depending whether it is in a fraction or sitting in the main part of the equation:

$$\sum_{i'} F_{i'} \leq \frac{F_i}{\int P(p_i)p_i} \tag{2}$$

It is good to get parentheses the right size. Do this by using `left(` and `right)` instead of '(' and ')'. Here is a silly example.

$$\left(\int_0^\infty P(p_i)p_i\right) = \sqrt{\left[\frac{F_i}{(\sum_{i'} F_{i'})}\right]^2} \tag{3}$$

Also, when you want to make an average like $\langle f_i \rangle$ or $\langle \prod g_i \rangle$, please use `\left<` and `\right>`, not plain `<` and `>`.

Here is how to do an **equation array**:

$$\begin{aligned} x &\simeq y^2 \cos(y) && (4)\\ &\gg y^2 && (5) \end{aligned}$$

LaTeX more or less forces you to produce a beautiful looking document, but it still allows you to do silly things like include big mathematical objects in sentences: $M = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 3 \end{bmatrix}$.

2.1 Customizing LaTeX

If there is a word or symbol, or indeed a whole phrase that you use often in your document, then you can define an alias for it. For example I abbreviate `{\bf x}` to `\bx`, using the command `\newcommand{\bx}{{\bf x}}`. That makes `x` easier to type and easier to read in the `.tex` file. It is a good idea to put all your `newcommand` declarations at the beginning of your `.tex file`.

3 Figures and `psfig`

There is a rudimentary figure drawing package in LaTeX but it is probably best to use `xfig` to draw pictures, and `gnuplot` to plot graphs and data. In `xfig`, click the necessary buttons to make a Postscript file, with suffix `.ps`, and then the figure can be included using some macros called `psfig`. In `gnuplot`, use the following procedure, when you are happy with your plot (for example, $\sin(4x)\exp(-x)$):

```
gnuplot> set size 0.7,0.7 [ this is one way of making the fonts look larger ]
gnuplot> plot [0:5] sin(4*x) * exp(-x)
gnuplot> set term postscript
Terminal type set to 'postscript'
Options are 'landscape monochrome "Courier" 14'
gnuplot> set output "sin_exp.ps"
gnuplot> replot
gnuplot> set term X11
Terminal type set to 'X11'
gnuplot>
```

By following this rather tedious procedure, you have written a `.ps` file and got back into normal `X11` mode.

You can then include your figures as demonstrated in figure 1. The `psfig` command in the `.tex` file has several arguments which control the size and rotation of the `.ps` file when it appears in your document. With some old ps files you might need to add extra arguments that give bounding box information.

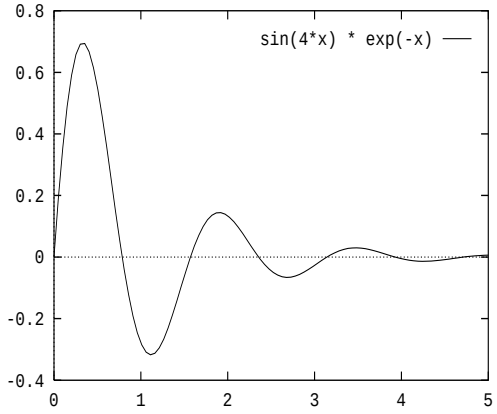

Figure 1: A figure

This figure was included using psfig. The command in the `.tex` file is:

```
\psfig{figure=sin_exp.ps,%
width=3in,height=2.53in,angle=-90}
```

For people who know a bit about TEX, what psfig creates is a box, with the dimensions specified. You can therefore, if you wish, include figures wherever you can put a box — even, for example, as a letter in the middle of a word.

4 Citations and BibTeX

LATEX handles bibliographic citations for you nicely. The command `\cite{MacKay92a}`, produces a reference to a document by MacKay thus: [1]. This document is listed in a bibliography wherever you choose in your document; conventionally, at the end (see the last page of this paper). The neatest way to create the bibliography is to have all your documents that you might cite in a `.bib` file (see `bibs.bib`). This lists the documents in a standardized format. You cite documents using the label (such as `MacKay92a`, above) that appears with them in the `.bib` file. LaTeX and another program called BibTeX collaborate in finding the required `.bib` entries and typesetting the appropriate bibliography for you. As you read new references, add them to your `.bib` file following the format rules.

Alternatively, you can type all your references out long-hand if you want. The good thing about BibTeX is that it can write the references in a number of consistent styles, controlled by a single command.

5 How to run LATEX, preview your document, and print

Having got to a stage where you would like to see your document typeset, there are two ways of proceeding. One is to type the following commands; the other is to have a makefile which knows how to use these commands. I will describe both methods. Note that although the commands

listed below are fairly standard, some of them may vary from system to system.[2]

COMMAND	WHAT IT DOES
`latex please`	makes `please.dvi` from please.tex.
`bibtex please`	finds the `.bib` entries noted in a file called `please.aux` that `latex` generated, and makes a file called please.bbl, which `latex` can read next time.
`latex please`	run `latex` a second and maybe third time if you want to make sure the cross references and citations are right. This is not obligatory, if you don't care about the cross references.
`xdvi please`	displays your typeset `please.dvi` file on your X terminal, if you are using one. This is a very nice on-screen previewer; its only disadvantage is that `psfig` figures are not displayed.
`dvips please -o please.ps`	makes `please.ps` from please.dvi. At this stage, all the figures get included. `dvips` might also run `metafont` to create special fonts your document uses.
`ghostview please.ps`	displays the file `please.ps` as it would print out on the printer.
`lpr -Plpr please.ps`	prints your document on the named printer (check the printer name with your system manager).

Alternatively, you can use the makefile. That way, you don't need to type six commands if you want to get from `please.tex` to `ghostview please.ps`. Instead you just type `make please.gv`, and all the appropriate commands are executed.

WHAT YOU WANT	COMMAND
To make `please.dvi`	`make please.dvi`
To see the document under `xdvi` (no figures)	`make please.x`
To see the document under `ghostview` (with figures)	`make please.gv`
To make `please.ps`, so that you can print out hard copy	`make please.ps`

A And finally...

If you need to know more, such as how to do particular mathematical symbols, how to make tables, etc., the best book to refer to when using LaTeX is Leslie Lamport's blue book, "LaTeX: A Document Preparation System".

References

[1] D.J.C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

Acknowledgements

I thank Tim Jervis and Martin Oldfield for teaching me about makefiles.

[2] On some systems, `dvips please` will do instead of `dvitps please.dvi -o please.ps` . Some systems have a default of printing to a printer immediately.

On some old systems, the `dvips` command must be replaced by `dvitps please.dvi > please.ps` .

Some old versions of psfig handle rotation of the figure differently (the user needs to supply bounding box information).