

Please write your report using L^AT_EX

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

November 15, 2005

Abstract

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

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

This document works with L^AT_EX2e, which is the standard version of L^AT_EX since 2003.

1 Introduction

L^AT_EX is *the* typesetting program to use in scientific publishing. It is good for writing anything from a one page letter to a six-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 `please2e.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. Get it from

http://www.inference.phy.cam.ac.uk/mackay/latex_demo/Welcome.html

1.1 Prose

Where your document consists of prose, you just type prose in your editor, and L^AT_EX 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

1.2 More exciting things

Where L^AT_EX 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

¹ Take a look at the file `please2e.tex` while you are reading this, so you can see what commands generated what. Oh, by the way: never use footnotes – if something is worth saying, say it in the main text!

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 `please2e.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**:

$$x \simeq y^2 \cos(y) \tag{4}$$

$$\gg y^2 \tag{5}$$

Please number all your equations. Maybe *you* don't want to refer to equation 4, but perhaps someone else will want to refer to it!

\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>
```

[Some versions of `gnuplot` also require the command `set output` at this point.] 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.

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.

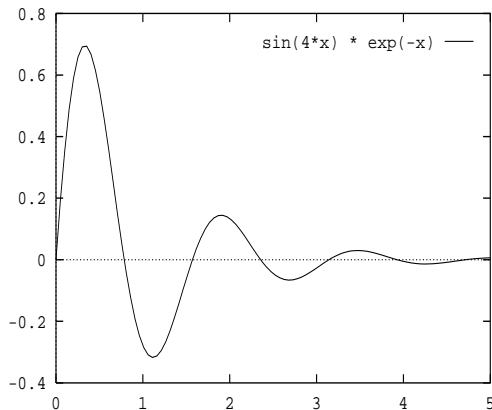


Figure 1: A figure

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}
```

4 Tables

The default style of tables in latex produces tables that look like this:

INVESTMENT COSTS	
Stage 1 (FY 1993/94)	£5000
Stage 2 (FY 1994/95)	£2000

but in my opinion (and the opinion of the authors of the `booktabs` package) these tables look unprofessional. I use the `booktabs` package (see the top five lines of `please2e.tex` for the `usepackage` command) and make tables that look like this:

INVESTMENT COSTS	
Stage 1 (FY 1993/94)	£5000
Stage 2 (FY 1994/95)	£2000

Notice the use of the commands `toprule`, `midrule`, and `bottomrule`, in place of `hline`. Vertical lines are not necessary.

5 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.

Note added Tue 15/11/05. The bibliography in this document uses the ‘plain’ citation style, which produces a simple number [1]. I strongly recommend using the ‘cite as noun’ convention, however, generating citations such as ‘according to MacKay (1992)’ or ‘Bayesian methods do complexity control (MacKay 1992)’. There are several packages that offer a `\citeasnoun` command. I used the `lsalike` package for some time. I think that the `natbib` package may be the best one to recommend.

6 How to run L^AT_EX, 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
<code>latex please2e</code>	makes <code>please2e.dvi</code> from <code>please2e.tex</code> .
<code>bibtex please2e</code>	finds the .bib entries noted in a file called <code>please2e.aux</code> that <code>latex</code> generated, and makes a file called <code>please2e.bbl</code> , which <code>latex</code> can read next time.
<code>latex please2e</code>	run <code>latex</code> 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.
<code>xdvi please2e</code>	displays your typeset <code>please2e.dvi</code> file on your X terminal, if you are using one. This is a very nice on-screen previewer; its only disadvantage is that <code>psfig</code> figures are not displayed.
<code>dvips please2e -o please2e.ps</code>	makes <code>please2e.ps</code> from <code>please2e.dvi</code> . At this stage, all the figures get included. <code>dvips</code> might also run <code>metafont</code> to create special fonts your document uses.
<code>ghostview please2e.ps</code>	displays the file <code>please2e.ps</code> as it would print out on the printer.
<code>lpr -Plpr please2e.ps</code>	prints your document on the named printer (check the printer name with your system manager).

²On some systems, `dvips please2e` will do instead of `dvitps please2e.dvi -o please2e.ps`. Some systems have a default of printing to a printer immediately.

On some old systems, the `dvips` command must be replaced by `dvitps please2e.dvi > please2e.ps`.

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

Alternatively, you can use the makefile. That way, you don't need to type six commands if you want to get from `please2e.tex` to `ghostview please2e.ps`. Instead you just type `make please2e.gv`, and all the appropriate commands are executed.

WHAT YOU WANT	COMMAND
To make <code>please2e.dvi</code>	<code>make please2e.dvi</code>
To see the document under <code>xdvi</code> (no figures)	<code>make please2e.x</code>
To see the document under <code>ghostview</code> (with figures)	<code>make please2e.gv</code>
To make <code>please2e.ps</code> , so that you can print out hard copy	<code>make please2e.ps</code>

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.