

Chapter 1

L^AT_EX

This Notes are based on Peter Shearers notes, from Scripps Institution of Oceanography.

Someday you will want to publish the results of your research. Many scientists typically use either Word or Latex for this purpose. Most of you probably already know how to use Word. It is very easy to use (WYSIWYG) and is the "standard" word processor of choice in many institutions. It is not, however, ideal for writing scientific articles. It has particular difficulty handling equations and specialized typesetting requirements. A far better choice is L^AT_EX. Guess what these notes are written in!

Advantages of LaTeX:

1. It is in the public domain so you are not supporting the Evil Empire.
2. You can run it on Macs, PCs or UNIX machines.
3. By using macros, it is a far more powerful and versatile system than all the pull down menu stuff you find in most commercial word processors.
4. AGU uses L^AT_EX for its electronic submission of articles and abstracts. They provide macros to match the style of their journals so that preparing camera-ready copy is trivial.
5. You can produce beautiful looking equations much more easily than with programs like Word.

L^AT_EX takes a little while to learn at first, but you will find it well worth your effort. I will describe running L^AT_EX from the UNIX command line although I am actually most familiar with using it on my Mac using a fancy implementation (TexShop), which you can download from multiple places. In Windows, you can use MikTeX

1.1 A simple example

Use your favorite text editor to create the file `samp1.tex` (an example from the L^AT_EXbook, Kopka and Daly, A Guide to Latex2e):

```
\documentclass{article}
\begin{document}
Today (\today) the rate of exchange between the British pound
and the American dollar is \pounds 1 = \$1.63, an increase of
1\% over yesterday.
\end{document}
```

which will produce:

Today (February 13, 2017) the rate of exchange between the British pound and the American dollar is £1 = \$1.63, an increase of 1% over yesterday.

The document must first define a document class. Options are `book`, `report`, `article`, or `letter`. The guts of the document are then enclosed between the `\begin{document}` `\end{document}` commands.

This example highlights some of the special characters and macros in Latex. The following characters are normally interpreted as Latex commands:

```
$ & % # _ { }
```

To print these out in your text, you must precede them with a backslash, i.e.,

```
\$ \& \% \# \_ \{ \}
```

In this example, `"\today"` invokes a macro to print today's date and `"\pound"` will print the British pound symbol. From this example you can see that Latex is not WYSIWYG! To typeset this document, enter:

```
latex samp1
```

Assuming you have no errors in the input file, this will generate the file `samp1.dvi`. To preview this file, enter:

```
xdvi samp1
```

To convert the file to a Postscript file, enter:

```
dvips samp1 -o samp1.ps
```

Notice that Latex automatically indented the first line of the paragraph. To avoid this, use the command for the target paragraph:

```
\documentclass{article}
\begin{document}
\noindent
Today (\today) the rate of exchange between the British pound
and the American dollar is \pounds 1 = \$1.63, an increase of
1\% over yesterday.
\end{document}
```

To globally change the paragraph indenting, you can change the default for this directly:

```
\documentclass{article}
\begin{document}
\setlength{\parindent}{0.5in}
Today (\today) the rate of exchange between the British pound
and the American dollar is \pounds 1 = \$1.63, an increase of
1\% over yesterday.
\end{document}
```

This will now indent all paragraphs by 0.5 inches. To remove paragraph indenting, just set this parameter to zero. Latex ignores the carriage returns at the ends of each line in the input file. It also ignores extra blanks; it only considers the first blank. New paragraphs are defined by adding a blank line between blocks of text.

1.2 Example with equations

```
\documentclass{article}
\begin{document}
\setlength{\parindent}{0.0in}
The function  $X(p)$  is more nicely behaved than  $T(X)$  since it does not
cross itself (there is a single value of  $X$  for each value of  $p$ ), but
the inverse function  $p(x)$  is multi valued. An even nicer function is
the combination
\begin{equation}
\tau(p) = T(p) - pX(p),
\end{equation}
where  $\tau$  is called the it delay time. It can be calculated very
simply:
\begin{equation}
\tau(p) = 2 \int_0^{z_p} \left[ \frac{u^2}{(u^2-p^2)^{1/2}} -
\frac{p^2}{(u^2-p^2)^{1/2}} \right] \, dz
\end{equation}
or
\begin{equation}
\tau(p) \approx 2 \int_0^{z_p} (u^2-p^2)^{1/2} \, dz \quad \backslash \backslash
&\approx 2 \int_0^{z_p} \eta(z) \, dz
\end{equation}
where  $\eta$  is the vertical slowness.
\end{document}
```

which typesets as:

The function $X(p)$ is more nicely behaved than $T(X)$ since it does not cross itself (there is a single value of X for each value of p), but the inverse function $p(x)$ is multi valued. An even nicer function is the combination

$$\tau(p) = T(p) - pX(p), \quad (1.1)$$

where τ is called the *delay time*. It can be calculated very simply:

$$\tau(p) = 2 \int_0^{z_p} \left[\frac{u^2}{(u^2 - p^2)^{1/2}} - \frac{p^2}{(u^2 - p^2)^{1/2}} \right] dz \quad (1.2)$$

or

$$\tau(p) = 2 \int_0^{z_p} (u^2 - p^2)^{1/2} dz \quad (1.3)$$

$$= 2 \int_0^{z_p} \eta(z) dz \quad (1.4)$$

where η is the vertical slowness.

Note that equations within the text are enclosed with \$ signs. `\begin{equation}` and `\end{equation}` are used to put the equation on a separate line. Variables within the equations are automatically put into italics. Greek letters are defined as `\tau`, `\eta`, etc. Equations are automatically numbered (this can be changed if desired). Subscripts are defined with the underscore (`_`), superscripts with the caret. Fractions are written as, for example, `x \over y`. `\int` is for the integral symbol; note how the limits are written. Curly brackets are used to separate things – they do not appear in the typeset version.

The `\begin{eqnarray}` section is used to align the = signs in the two lines of equations. Note that `&=&` is used to define what it is that is being aligned. Every line except the last line has a carriage return (`\`). Although TeX generally does an excellent job of spacing equations, sometimes some fine-tuning will help. In this case `\,` is used to place a tiny amount of space before the `dz`.

1.3 Including Graphics

Postscript, EPS, and PDF files can easily be embedded as figures in a Latex document by including Latex extension packages such as `graphics` or `graphicx`. Here is an example that uses `graphicx`:

```
\documentclass{article}
\usepackage{graphicx}
\begin{document}
\setlength{\baselineskip}{20pt}
We are now going to show how to embed a Postscript file into
```

a LaTeX document using the `includegraphics` command.

```
\begin{figure}[h]
\begin{center}
\includegraphics[scale=0.7]{figures/Fig2.pdf}
\end{center}
\caption{Here is the caption for this plot. This will be automatically
positioned below the plot.}
\end{figure}
And then here is some more text to show where the next block of text
will appear. Blah, blah, blah...
\end{document}
```

which looks like this:

We are now going to show how to embed a Postscript file into a LaTeX document using the `includegraphics` command.

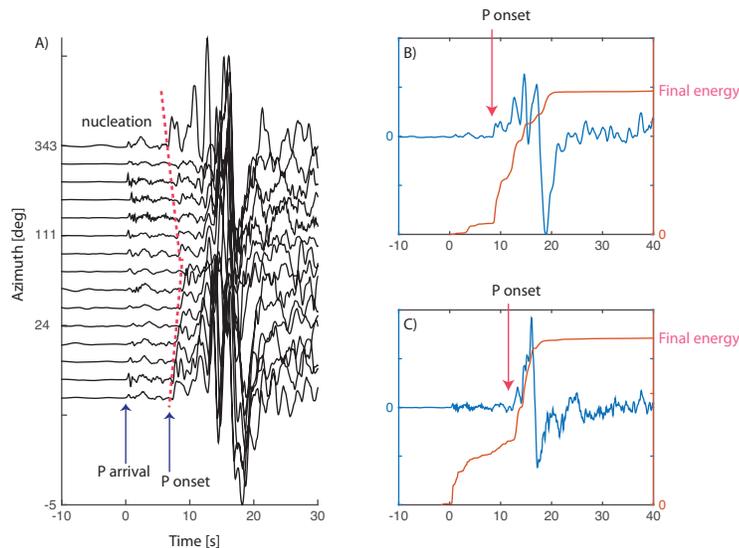


Figure 1.1: Here is the caption for this plot. This will be automatically positioned below the plot.

And then here is some more text to show where the next block of text will appear. Blah, blah, blah...

Note that the `graphicx` package is loaded with the `\usepackage{graphicx}` command at the start of the file. The `\begin{figure}` macro has various options for where the figure will be positioned, including at the present location in the text [h], or at the top [t] or bottom of the page [b]. These options can be combined, i.e., [tb] will position the figure at either then top or bottom of the page. In this example, the figure is scaled to 50% of its original size. Depending

upon how the figure is positioned in the PDF file, you also may need to apply a "bounding box" to remove the surrounding white space using the `bb=` option, e.g.,

```
\includegraphics[bb=2in 4in 7in 9in, scale=0.7]{Fig2.pdf}
```

which specifies exactly what part of the page will be windowed and displayed. This is necessary for PDF figures that appear in only part of an entire page and it can be tedious to find the right bounding box. In my experience, an easier option is to use the Mac Preview program to open Postscript or EPS files (from Adobe Illustrator) or other graphics programs) and then save them within Preview as PDF files. In this case, they are tightly windowed and the `bb` option is not needed.

Sometimes you may want to include other types of figure files (JPG, PNG, etc.), and the PDF Latex version I use on my Mac, and others of course, will allow you to do that. I use the following basic setting

```
\documentclass{book}

...

\ifx\pdfoutput\undefined
\usepackage[dvips]{graphicx}
\else
\usepackage[pdftex]{graphicx}
\pdfcompresslevel=9
\usepackage{epstopdf}
\fi
...
\begin{document}
\DeclareGraphicsExtensions{.pdf,.gif,.jpg, .eps}

...

\begin{figure}[htbp]
\centering
\includegraphics[width=13cm]{fig_any_type}
\caption{caption eeeeeee}
\label{fig:****}
\end{figure}
...
```

note that the `\label` can be used to put a name to your figure and reference it anywhere in the text, without remembering which figure number it is.

1.4 Want to know more?

There is a huge amount of material about Latex on the web. Personally, I believe the best introduction to become a basic user of L^AT_EX, is "*The Not So Short Introduction to LaTeX*" by Tobi Oetiker.