A Brief LaTeX Tutorial

Modern Physics Laboratory

October 6, 2002

Abstract

LaTeX is a typesetting language – it is not a word processor, and as such, is not a program that one tends to use for most day-to-day document preparation needs. Indeed, for most people, using LaTeX to write a letter or an essay is kind of like using a sledgehammer to kill a fly. However, when it comes to producing high quality scientific documents, it can't be beat. Probably the best evidence for this is the fact that virtually every physics and engineering journal requests that papers be prepared using some version of LaTeX.

1 Running LaTeX

We'll be using LaTeX on the PC, but pretty much everything you learn about it is transferable to Macs or UNIX/linux machines.

Unlike a word processor, LaTeX requires that you go through a number of steps in order to produce your document. Fortunately, current PC versions of LaTeX keep this stuff fairly invisible. Basically, you have to create a `.tex` file, which in turn is used to create a `.dvi` file, which is then used to create a `postscript` file, which then can be printed. Here, in slightly more detail, are the steps:

1. Double click on the LaTeX icon (or the WinEdt icon) on the desktop. This runs an editor called WinEdt, which is available as shareware for PC's (if you have a Mac, I have some suggestions for you depending on your OS), should you want a version on your home machine (this is something I would highly recommend - you need both WinEdt and MikTeX, both of which are free). WinEdt is specifically designed for editing and compiling LaTeX.
2. Load and edit your LaTeX file.
3. Compile your LaTeX file to a DVI file. This is achieved by selecting the LaTeX button on the toolbar. When you press this button, a DOS window will open, and show you the commands that are executed to compile the file. If all is well, this DOS window will disappear in a couple of seconds (or give you the option of closing it). If, on the other hand, you have an error in your file (an inevitable occurrence), the DOS window will remain open, and will show you the details of the error (line number, error type). Enter an X to exit the DOS window.
4. Attempt to fix any bugs in your LaTeX file (i.e., repeat steps 2 and 3 until the file compiles).
5. Look at the (almost) typeset result by hitting the DVI Preview button (below and to the right of the LaTeX button). Again, fix any bugs and recompile until you are happy with the result.

6. Convert the file to postscript by hitting the `dvi` → `ps` button.

7. Look at the final postscript version using the ghostview button. Once you're looking at the final version, you can print just as you would in any other application. (You can also print the `dvi` file, which saves a step, but doesn't do graphics as well.)

You can see why one would not use this system for a letter to the editor.

2 A step-by-step guide

In order to help you learn to use LaTeX, we're going to walk through the steps above very explicitly.

1. **Run WinEdt:** Just double click on the LaTeX icon or the WinEdt icon.

2. **Open an existing file:** Using the `file` menu, load the file `LabFormat.TEX`.

   You should get a window that has the following in it (hopefully, I haven't changed it too much...):

```
%%%%%%%%%%%%%%%%%%%%%%%%  LabFormat.tex %%%%%%%%%%%%%%%%%%%%%%%%
%
%
% This is a file that can be used to create documents in compliance with the
% submission requirements for Modern Physics Lab formal reports.  It contains
% pretty much all the commands necessary, and is reasonably documented.

\documentclass [10pt]{article}

% The \documentclass command tells LaTeX how to compile the document. The [10pt]
% specifies the base font size for the document. Inside the square brackets,
% you could also write [twocolumn], if you want the document to print in two-
% column format. The {article} specifier after that tells LaTeX the overall
% format.  There are a number of others, including {book}, {report}, etc.
% {article} is the standard option.

\usepackage{epsfig}

% This usepackage tells LaTeX that you intend to use encapsulated postscript
% figures.

\title{The Modern Lab Formal Report}
\author{Eric C. Martell\thanks{currently at Dept.\ of Physics and
Astronomy, Vassar College, Poughkeepsie, NY.  \hbox{E-mail: {\tt
ermartell@vassar.edu}.}}, Mark H. Somerville\thanks{currently at
The Olin College of Engineering, Needham, MA. \hbox{E-mail:~{\tt
mark.somerville@olin.edu}.}} }

\markboth{The Vassar Journal of Modern Lab Physics, Vol. IV, No.
1, January 2002}{The Vassar Journal of Modern Lab Physics, Vol.
IV, No. 1, January 2002}

% The \markboth command simply determines how the pages will be marked.
```

```
% There are two arguments, since for double-sided pages, you often want a different
% marking on facing pages.
\maketitle
\begin{document}

% Note that we are defining both the title and the author -- LaTeX will decide where
% they ought to go in the final typeset, depending on the style we've selected.
% The command \maketitle tells LaTeX that the title section is complete.
% Finally, the command \begin{document} tells LaTeX that the body of the document
% is about to begin.
```

Notice that although this is somewhat readable, it looks nothing like the final document – indeed, it looks more like some sort of computer program. Here are some facts that will make is a bit easier to decipher:

- Note that LaTeX ignores anything after an % symbol on a particular line. Thus, you can use % to write comments in your LaTeX file.
- Now, if you look a bit further into the document, you will see some text that is not so nasty looking:

```
\begin{abstract}
The abstract should summarize what exactly the {\it paper} does.  For example,
it might say something like this:

% Note that LaTeX ignores things like a single carriage return or multiple
% spaces in formatting the document. For example, a b will print the same as
% a    b, and
% a
% b
% will print the same as ab.  You need a complete blank line (two carriage
% returns) for LaTeX to begin a new paragraph. Note that three or more carriage
% returns will be treated as just the two. If you want a blank line in between
% paragraphs, use the command \vskip, like \vskip0.1in or \vskip12pt.

We present new measurements of the dropping time of apples from trees.  Our
measurements suggest that both the time for the apple to fall and the velocity
at which it hits the ground depend strongly on the height of the tree.  Using
Newton's recently proposed gravity picture, we are able to model our results
accurately.  We find the gravitational acceleration to be
$5 \times 10^3$~cubits/hour$^2$, with an uncertainty of
$\pm 0.3$~cubits/hour$^2$.

% To insert mathematics into a paragraph, bracket the mathematics code
% with dollar signs.  See the commands \times and \pm?  Note what they
% do in the text.  There is a reference 'sheet' you will have listing
% many math symbols.  Also, note the '~' symbol.  This tells LaTeX not
% to break up the things it connects, so that (in this case) the units
% won't get separated from the numerical values.
\end{abstract}

\section{Introduction}
The introduction places your work in context by summarizing the
current state of knowledge, and stating explicitly what the paper
```

```
does and how it does it.  To a certain extent, you end up
repeating things from the abstract.  For example: (blah blah blah)
```

Notice that the text in this document is not particularly neatly organized – lines are of different length, etc. LaTeX does not care about this – you can enter as many or as few words in a line as you wish, because LaTeX is going to move them all around anyhow.

Also notice the use of

```
\begin{abstract} ... \end{abstract}
```

and

```
\section{Introduction}
```

What do you think these commands do (not a tough question...)?

We will deal with some of the other stuff in this file later. First, let's finish with the step-by-step guide...

3. **Compiling the file:** Now that we've had a look at what's in this file, let's see what the typeset version looks like. Hit the LaTeX button. A DOS window will open, which will have something like the following in it...

```
This is TeX, Version 3.14159 (C version 6.1) (format=latex 98.8.27)  20 SEP 1999
 08:19
**LabFormat (LabFormat.tex LaTeX2e
<1996/12/01> patch level 1 Babel <v3.6h> and hyphenation patterns
for american, german, loaded.
(/usr/local/teTeX/texmf/tex/latex/tools/rawfonts.sty Package:
rawfonts 1994/05/08 Low-level LaTeX 2.09 font compatibility

(/usr/local/teTeX/texmf/tex/latex/tools/somedefs.sty
Package: somedefs 1994/06/01 Toolkit for optional definitions
)

...
```

This information is not terribly useful right now, although it will be when you have errors in your file.

4. **Viewing the `.dvi` file:** Now hit the DVI Preview button. This should open a viewer, which will allow you to look at the typeset file. Play with the viewer for a while to get a sense of how it works.

5. **Finalizing the file:** Although it *looks* like the DVI file is a final version, it is not quite done – first you have to convert the file to postscript. Press the `dvi` → `ps` button, followed by the Ghostview button. This will open another previewer, which allows you to print the high-resolution final version.

3 Formulas in LaTeX

One of the best things about LaTeX is it's ability to create beautiful formulas, like

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{2\pi nt}{T}\right) + b_n \sin\left(\frac{2\pi nt}{T}\right) \tag{1}$$

Now while you could do this using the equation editor in MS Word, it would be relatively time consuming, and in the end, would produce rather ugly output. LaTeX allows you to enter formulas as if you were reading

them – for example, the formula above might be *read* as "f(t) = the fraction a sub 0 over 2 plus the sum from n=1 to ∞..." Similarly, to enter this formula in LaTeX, you do something like this:

```
\begin{equation}
f(t) = \frac{a_0}{2} +
\sum_{n=1}^{\infty} a_n \cos\left({ 2\pi n t\over T}\right)
+b_n \sin\left({2\pi nt\over T}\right)
\end{equation}
```

Note the use of `over`, `sum`, and `infty` – each of these plays a role pretty much identical to the "formula-as read." You can also start and end an equation with double dollar signs, as in

```
$$
f(t) = \frac{a_0}{2} +
\sum_{n=1}^{\infty} a_n \cos\left({ 2\pi n t\over T}\right)
+b_n \sin\left({2\pi nt\over T}\right)
$$
```

The detailed rules for producing mathematical formulas in LaTeX are, of course, rather involved. Fortunately, the WinEdt program has quite a bit of this stuff automatically included in the TeX symbols GUI (on the toolbar). Furthermore, once you have the hang of it, LaTeXis actually much faster than the equation editor, and produces lovely output to boot...

4 Figures in LaTeX

4.1 Handling figures

Another lovely thing about LaTeX is its ability to handle figures well. If you've every tried to get MS Word to put a figure in the right place, you will appreciate how easily LaTeX deals with this issue. The command to insert a figure is, not surprisingly,

```
\begin{figure}

...

\end{figure}
```

When LaTeX encounters such a command, it finds the best *legal* place to put the figure: figures cannot appear on a page preceding the text in which they are referenced, but should appear as soon as possible after that, so long as the layout remains reasonable.

The figures in this file are all encapsulated postscript figures – hence the command

```
\centerline{\epsfysize= 2 in \epsfbox{apple.eps}}
```

which tells LaTeX that the figure should contain the `eps` file `apple.eps`, and that the figure should be 2 inches high.

After the information about the picture, the figure environment also allows you to enter a caption:

```
\caption{ Experimental setup. An observer (represented by a
stickman) follows apple movements.  When an apple falls, he or she
times the fall.}
```

Note that LaTeX will automatically generate figure numbers, etc. – and will fix them when you insert new figures...

4.2 Generating figures

Mathematica and LaTeX work well together, as Mathematica will easily generate `eps` files that look good in LaTeX. All you need to do is plot your graphics using the command

`Display[``filename.eps'',PlotCommand,``EPS'']`.

You can also choose to save your Mathematica file as TeXcode, which will then output the files as `eps` figures.

If you need to generate pictures, the program SmartDraw Professional is very easy to use. This is installed on all the Modern Lab machines. In the SciViz lab, Adobe Illustrator is probably the best bet. **Regardless of how you generate the figure, it must be saved in encapsulated postscript format!**

5 References

The best book I have found for using LaTeX is The LaTeX Companion, by Leslie Lamport. Typically when you start using LaTeX, you spend lots of time trying things out and getting errors – don't be afraid to come bug me if your code isn't working (Learning LaTeXis largely a process of trial and error, so don't be afraid to try different ways of achieving the same goal. Also, a good way to learn how to do something is find someone who has done it, and copy their template.).