LaTeX Is Your Friend OR ENEMY?????????.

Nathan Lundblad [1]
January 28, 2007

ABSTRACT

We present a paper on useful LaTeX stuff. Make sure to look at the source code for this document, as that is where the real story is. For more fun, look at Leslie Lamport's book in the 705 Campbell bookshelf.

Contents

1. The Big Picture

LaTeX (pronounced *lay*-teck or *lah*-teck) is a designer package based on a typesetting program called TeX which was originated by Donald Knuth[2] of Stanford many many years ago. LaTeX first

---

[1]Originally written Aug 31 1999. New LaTeX 2_εupdates and additional commentary by Erik Shirokoff (2004) and Carl Heiles.

[2]The greatest computer scientist in the world.

appeared in 1985 and is extremely popular, particularly in the scientific community where it has become an almost universal standard. Using LaTeX will result in stunningly beautiful documents and will, in the long run—because of mathematics and labels/referencing—be easier to deal with than using Micro$oft Word^{tm} and its cousins. Although creating reports and articles in a different fashion from what you may be used to can be a little intimidating at first, a few basic facts and a couple of good sample documents[3] will take you a long way.

2. How To Use LaTeX: The Most General Possible Summary

Remember on PC word processors how there is an option called *reveal codes* or some such? Well, in LaTeX you essentially write those codes yourself, and then compile them to get your printable output. You'll type up these codes in your favorite text editor and name the file something appropriate with a `.tex` suffix.

Then you must compile that file at your shell prompt by typing `latex whatever` (you can include the suffix `.tex` if you want). LaTeX will spit out some random files (provided you haven't made any errors), including `whatever.dvi`, which is your printable output. LaTeX will also print some messages on your screen. *Be sure to look at these messages!!!!!!!* If your compilation failed, they will attempt to tell you what error you may have committed. Once you figure it out, you edit the `tex` file and try running `latex whatever` again. The most common error is to forget the $ sign on each side of an equation, or to have unmatched curly brackets. The error message gives the line number; the easiest way to find the offending text is to go to that line number in your editor. *NOTE* that many times the error occurs *before* the line number given by the LaTeX output.

To view the `whatever.dvi` file on your terminal screen, type `dvi whatever &` at your shell prompt (you can, but don't need to, include the `.dvi` suffix). The nicely-formatted output appears on the `dvi` output window. You can edit the `tex` file and left click on your `dvi` output window; the updated text appears.

Look thingsg over carefully and make any changes *before printing it on paper*—support environmentalism! Finally, to print the output when you're all done, it's a three-step process[4] :

1. `dvips whatever` (creates the PostScript file `whatever.ps`)
2. Before printing, you should make one final check by looking at the PostScript file on your screen: `gv whatever.ps` .

---

[3] Available all over the place. You'll get a sample lab report done in LaTeX for example.

[4] On some systems, you can print the dvi file directly using `dvips whatever.dvi | lp` . This command creates a temporary PostScript file from `whatever.dvi`. The `| lp` "pipes" this file to the printer; omitting this part would print the PostScript file onto your screen—something you don't want because it is uninterpretable.

3. `lp whatever.ps` (prints the file).

3. Some Basic Syntax

Every LaTeX document must be enclosed by a `\begin{document}` tag and an `\end{document}` tag. Nothing goes after the latter[5], but some very important stuff goes before the former, such as documentclass declarations and suchlike, which you'll learn about in Section 5. As you may have noticed, LaTeX reserves more than a few characters for its own nefarious purposes. Generally, to produce them in your final document you must invoke the backslash, like so: "`\$12`", which results in a final output like so: $12. The same method applies to other special characters: { # } %[6].

There are three kinds of hyphens in LaTeX: -,–, and —. The first is used for intra-word dashes, the second for number ranges (41–42), and the third for the standard intra-sentence dash—it's my personal favorite. In other situations, just use whatever looks the best.

Grouping letters and words is accomplished with the { and } characters. Most commands only work on one group at a time, so surround the parts of your text you want to modify with curly brackets. For example, you can have *italicized type*, **boldface type**, and `typewriter-type type`.

Footnotes are incredibly easy to produce, and are automatically numbered.[7]

The observant student in the back of the room may cleverly ask "So...how do you create a backslash, if `\\` represents a skipped line?" [See comment on the title.]. Well, you have to use the `\verb` (verbatim) environment, which is handily revealed in the source code. The argument of the `\verb` environment is delimited by two identical characters; above, we used ampersands. You can use a pair of any normal characters as the delimiter. The `\verb` environment has an unfortunate peculiarity: you have to put all of its argument on a single typed line in the `tex` file. If you want to do a lot of verbatim stuff—really useful when you want to provide a list of IDL programming commands, for example—use `\begin{verbatim}` and `\end{verbatim}`; these don't require delimiters. For example, to list some well-documented IDL code:

```
function wopen
;+
;NAME:
;WOPEN -- return list of all open windows
;
```

[5]Except for comments which you don't want to be interpreted.

[6]The percent sign % is used for commenting your code, which is very important in, say, C programming but not too important in LaTeX.

[7]Like So. Voilà!

```
; PURPOSE:
;       Quick way to find all open windows
;
; CALLING SEQUENCE:
;       result= wopen()
;
; INPUTS:
;       NONE
;
; RETURNS: VECTOR OF OPEN WINDOWS
;
; RESTRICTIONS:
;       The current device must be X Windows.
;
; MODIFICATION HISTORY:
;       Written CARL, who finally got fed up
;-
; ARE YOU USING X WINDOWS DEVICE...
if (!d.name ne 'X') then begin
  message, 'DEVICE not set to X Windows.', /INFO
  return, -1
endif

; FIND THE OPEN WINDOWS...
device, window_state=openwindows
openwindows = where(openwindows,Nopen)

return, openwindows
end
```

`\begin{verbatim}` has the perhaps unfortunate peculiarity that it skips and starts a new line.

4. Labels/Referencing

When you're preparing a LaTeX document, it's *smart, labor-saving, sophisticated, and good practice*—but not necessary—to use the "`\label`" command. The use of labels ensures that you can refer to sections, equations, figures, and tables by a name—i.e., by *reference*—and not a number. So what's the difference? When you're inserting, cutting, and pasting, you *will* lose count of what section you're in or what equation is what, which will make referring to such objects in the text. Because I labeled the beginning of this section, I can always refer to it using the label, regardless of whether I go back and make changes in section order. For example: in the `tex` file it says, "`The current section is Section \reflabelsec`", while in the LATEX *output* it says "The current section is Section 4". There are several examples of how labels work in this primer; some are pointed out with comments.

5. Style files, packages, and user defined commands

You may have noticed the following line at the beginning of the source code:

```
\documentclass[preprint]{aastex}
```

This line sets a template for the document as a whole; it tells LATEX that you want to write an article-type document using the American Astronomical Society's preprint class [8] package. Specifically, this command instructs LATEX to read the file called `aastex.sty`, which is known as a "style file"; if you want to use AASTEX on your own computer, you need to have this file available on your path. The AASTEX class sets the font and layout for the entire document and it automatically loads some useful packages, which are collections of new commands that allow you to customize your document and do nifty things with images and layouts.

6. Mathematics

The great beauty of LATEX lies in how the math comes out. It does numbered equations exceptionally well, enables math within standard text, and has a shocking number of special characters available. Inserting standard equations into a LATEX document is done with the `\equation` environment, and works like so:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = 0 \tag{1}$$

Laplace would have loved LATEX. You can also do Greek letters easily:

$$\gamma = \frac{1}{\sqrt{1-\beta^2}} \tag{2}$$

If you want to put mathematics into text, you can use math mode, which is commonly delimited by dollar signs; thus, `$\alpha=\beta=\int_0^2 x^{-2.4} dx$` will look like $\alpha = \beta = \int_0^2 x^{-2.4} dx$. For an example of how labels work with equations, look at the code for Equation 2.

---

[8]The American Mathematical Society and the American Physical Society also have their own formats. We like AASTEX, and recommend you stick with it.

If you want to show a matrix math equation, you use the `eqnarray` environment:

$$\left[\begin{array}{cccc} [ss] & [st] & [su] & [sv] \\ [ts] & [tt] & [tu] & [tv] \\ [us] & [ut] & [uu] & [uv] \\ [vs] & [vt] & [vu] & [vv] \end{array}\right] \cdot \left[\begin{array}{c} A \\ B \\ C \\ D \end{array}\right] = \left[\begin{array}{c} [sy] \\ [ty] \\ [uy] \\ [vy] \end{array}\right] \tag{3}$$

If you want an equation, such as $\alpha = \beta \times \Lambda \cdot 4$, to be in bold—including those Greek letters—surround the whole equation by `{\boldmath ... }`; the result is $\boldsymbol{\alpha = \beta \times \Lambda \cdot 4}$.

7. Figures

If you want to bring in plots from IDL or, for that matter, an arbitrary graphic, you must first make sure that the file in question is an Encapsulated PostScript File or a PostScript file[9]. Once you have the file in the same directory as your `.tex` file, you can insert it into the document like so:

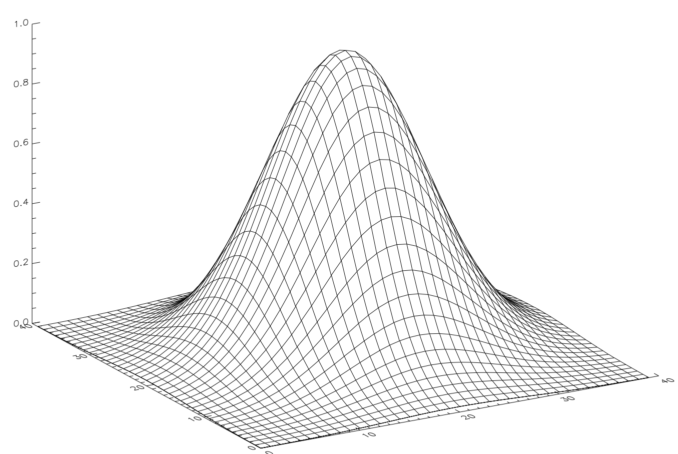

Fig. 1.— A Gaussian.

In addition to width, you can define height, angle, and scale. If you specify only width or height, the other dimension scales automatically. If you specify both, you can stretch the image. Angle rotates the image by some number of degree in the positive direction. Scale multiplies the

[9]If you have a non-PostScript figure, you can make a PostScript copy with the Linux/Unix command `convert`.

picture's original size by the number you specify. When specifying width or height, you must include units. Some options are: \textwidth, in, cm, pt, em, ex. See the Not So Short Guide for more info.

If you want to display several pictures together or have size scaling or stretching or rotation, as in Figure 2, you can do this.

One of the most difficult tasks for the novice (and, even, the experienced!) typesetter is image placement. LaTeX places floating bodies where it thinks they best fit, which isn't always the most logical place in a document. You have one way to control placement: the placement commands, which work for tables and figures. They are: `[h!], [t!], [p!], [b!]`, meaning: "put here", "put at top of page", "make a new page", "put at bottom of page". We used `[h!]` for Figure 1 and `[p!]` for Figure 2. Sometimes they are frustratingly inattentive to your desires; this occurs because LaTeX is smarter than you think it is—there's not enough space to put the figure exactly where you want it. .Judicious use of sizing (for images) and using smaller fonts (for tables[10]), or relocating, are your only options.

8. Tables

Tables are useful for displaying a large number of results. There are two environments provided for tables; `{table}`, which is a LaTeX resident environment, and `{deluxetable}`, which is an AASTeX custom environment. Table 1 is the {`table`}, a simpler version for which the placement commands work; Table 2 is the `deluxetable`, a more elaborate version for which the placement commands do not work—it always puts the table at the very end, so it's not very nice for lab reports.

Let's begin with the ordinary table, which is more flexible because you the placement commands work; here we use `[b!]`, specifying its location to be the bottom of the page...

[10]To temporarily use a smaller font: `[small text...]`or even `[tiny text...]`

Table 1: Sample table

Temperature	Voltage Drop
310K	0.6761V±0.0004V
300K	0.7064V±0.0005V
77K	1.5318V±0.001V

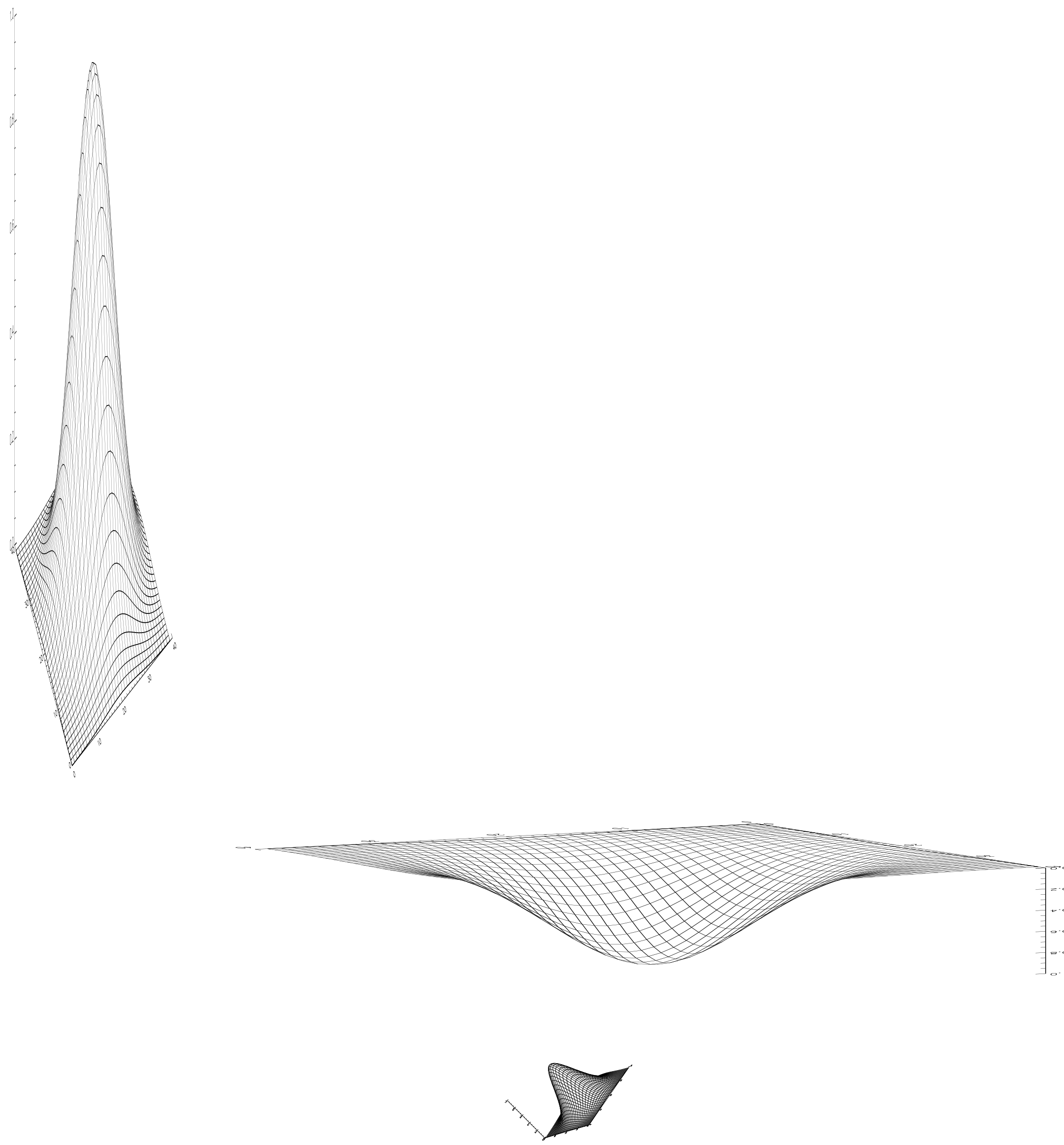

Fig. 2.— This is a very silly figure!

And now, we end with the deluxetable; it's always at the end, on its very own page. Because we're ending with it, this is one of the few instances where it's properly placed—but because it's on its own page, it's placement definitely not elegant!

Table 2. Sample table

Source	ℓ	b	τ_{max}	v_{LSR}	FWHM	ref, note
0624-058 (3C161)	215.4	–8.0	0.67	12.0	4.5	1,a
3C161	215.4	–8.0	0.88	7.6	2.5	1,a
3C161(OH)	215.4	–8.0	0.013	8.6	1.2	3
PKS0605-08	215.7	–13.5	0.80[b]	7.3	8.9	2
0530+04 (4C04.18)	200.0	–15.3	0.8:	4.3:	6.7:	2
3C135	200.5	–21.0	$\lesssim 0.11$	···	···	2
PKS0533-12	215.4	–22.2	0.36	3.9	8.0	2

References. — (1) Mebold *et al.* (1981), Mebold *et al.* (1982); (2) Crovisier, Kazès, and Aubrey (1978); (3) Dickey, Crovisier, and Kazès (1981).

[a]Mebold *et al.* (1982) list 3 components in addition to the 4 listed here.

[b]We have not listed a second, weaker Gaussian component because of poor signal/noise.

Note. — This comment applies to the whole table and you can put it either in front or behind the other comments.