Getting Started with LaTeX

David McCormick

December 3, 2008

LaTeX is the only sensible way to typeset any kind of mathematics. This is because LaTeX separates editing from formatting, letting you change what the text actually says (editing) but taking care of the exact fonts, structure, etc. (formatting) for you. Word-processors like Microsoft Word are a very good compromise between the two, but just like a campervan is neither a very good car nor a very good house, Microsoft Word isn't very good at editing nor is it very good at formatting. Where word-processors fall down as completely inadequate, LaTeX shines through as by far the best way of creating good-looking mathematical (and non-mathematical) documents. Compare the equation on the left, created with Word and Equation Editor, with the one on the right, done in LaTeX:

$$\sum_{k=1}^{\infty} \ell \left(\left\langle a_k^{(n)}, b_k^{(n)} \right\rangle \right) \leq m^*(E_n) + \varepsilon_n. \qquad \sum_{k=1}^{\infty} \ell \left(\left\langle a_k^{(n)}, b_k^{(n)} \right\rangle \right) \leq m^*(E_n) + \varepsilon_n.$$

We hope you agree that the equation on the right looks much better.

Using LaTeX is, in some sense, a lot like programming code – you type plain text with special commands to tell LaTeX what you want (such as an equation, a table or something), then compile it to get a result. However, newcomers to LaTeX tend to be frightened of it, mostly because it works in a completely different way to how word-processors work. This handout should guide you through installing LaTeX, getting started using LaTeX, and where to get more information on how you do all sorts of crazy things.

1 Installing LaTeX

First, let us guide you through how to install LaTeX and the associated tools needed to get the most out of LaTeX. LaTeX is merely the name of the language, and its exact implementation is up to other people; thus there are many different distributions. It is a sad fact of statistics that most of you will be using Microsoft Windows; the rest of the installation part deals only with Windows (principally for reasons of space), for which I apologise. Fortunately, once you have installed LaTeX, in theory it should work the same regardless of what operating system you are using. Alternatively, if you don't want to install it, LaTeX is installed on the computers in the Linux room in the maths building (A0.01), so if all else fails you can work there.

1.1 Installing LaTeX on Windows

First, ensure that you have enough free space on your hard drive – the installer and LaTeX are both reasonably space-intensive, so you should expect to need about 3GB during installation (though you can probably get rid of about 1.5GB once you've installed by deleting or moving the installation files).

Step 1: Download The ProTeXt distribution, available from `http://www.tug.org/protext/`, is the easiest fully-featured distribution for beginners to install. The download is, unfortunately, 540MB in size. (However, compare this to how big Microsoft Word would be on a CD and you'll realise it's not bad, especially in these days of broadband.) This contains all the bits and pieces you will need to use LaTeX. You might as well save it to the desktop, it doesn't matter at this stage.

Step 2: Extraction Having downloaded the package, double-click on it. It will prompt you to enter a folder (other than the default one) to which to extract the installation files. Choose wherever you like (a folder on the desktop is again a reasonable choice), and click "OK" to extract the files.

Step 3: Open the Setup Wizard ProTeXt has been designed to walk you through installing the various bits and pieces you need. Navigate to the folder you extracted the files to in step 2, and double-click on the "setup.exe" file. This will offer you a choice of languages, and once you have chosen your preferred language it will bring up a PDF file. At this point you may be thinking, "why is the setup wizard in a PDF file?" Well, the designers of ProTeXt wanted to ensure that

you knew what you were doing, so they included instructions along with the installation procedure. I'm sure it could be done with a really complicated wizard-like thing, but this is much easier for the developers, and I think for the user as well.

Step 4: Read the PDF installer   Read through the entire PDF file first, BEFORE you try installing anything. The installation procedure is mildly complicated, and we don't want you to make any mistakes.

Step 5: Install MiKTEX   So, now has come the time to actually install things! §1.5 of the installer guides you through installing MiKTEX, which is the actual engine behind LATEX which turns your instructions into pretty mathematics. The ProTeXt people recommend only installing the basic packages; while that's fine for most people, if you have lots of hard disk space and you want to save yourself hassle later I would personally recommend doing the complete install. Be careful also to select the "A4" option, or you will be stuck with awful American paper sizes. The installation could take anywhere from 20 minutes to an hour, so go away and make a cup of tea.

Step 6: Install TEXnicCenter   Once MiKTEX has installed (and perhaps after you've restarted if necessary), you can now move on to the other things you're going to need. There are many different "editors" for LATEX documents out there – essentially, what you type in order to produce LATEX is plain text with some special commands in it, and there are many different products out there to allow you to type LATEX without having to remember the code for a symbol, rather allowing you to just click a button to get that symbol. The one that comes with ProTeXt is called TEXnicCenter, and if you're satisfied with that, go and install it from §1.6 now. If you don't like it, then the installer has some recommendations for other editors, but there are many out there on the internet to be found; it's very much a matter of personal taste.

Step 7: Install GhostScript and GSView   Finally, in order to be able to deal with PostScript files, you need to install GhostScript and GSView, as outlined in §1.7 and §1.8. PostScript is a file format which is a predecessor of PDF, and is usually used as an intermediate between LATEX files and PDF files to insert graphics. Don't worry too much about it now; unless you're really short of space you should install GhostScript and GSView now.

1.2   Installing LATEX on Linux

For those of you using Linux, and who are happy compiling and installing things from tarballs, you can go to the website `http://www.tug.org/texlive/` and download the TEX Live system there. If not, I suggest searching for LATEX in your package manager and installing the relevant packages. In either case, since you're using Linux I trust you have enough common sense to figure it out for yourselves. Of course, Linux has a wide range of text editors, and the likes of Kate or KWrite for KDE, or GEdit for Gnome, are perfectly well-suited to writing LATEX. If, however, you want something with symbols, then I can highly recommend Kile for KDE as a very good editor with LATEX support.

1.3   Installing LATEX on Mac

Unfortunately, never having used a Mac I am rather in the dark on them. However, for those of you using a Mac, I am told that the MacTEX distribution, available from `http://www.tug.org/mactex/`, is the best distribution available for Mac. As for editors, TEXShop and TextMate are two editors which have been recommended.

2   Using LATEX

So, you should now have everything you need to use LATEX. There's lots of information on the internet as to how you can use LATEX; a selection of some of the best follows. But first, a word of warning. LATEX was based on a system called TEX, which (if you're feeling vindictive) you can still use. But it makes life much harder for you. However, the documentation for the two often gets conflated. Be sure that whatever you're using refers to LATEX and not TEX.

A few essential bits of documentation:

- `http://www.tug.org/begin.html` has links to a number of useful reference documents. Most importantly, *The Not So Short Introduction to LATEX 2ε* is the ultimate beginner's reference guide to using LATEX. Download it now. Once you have worked through it all, you should be able to face anything that LATEX can throw at you. (It is also known by its filename, which is "lshort".)

- The *Comprehensive LATEX Symbols List* does exactly what it says. 4947 symbols on 141 pages, all neatly indexed to make it easy to find the exact symbol you want. (Imagine trying to do that in Word!) Download it now from `http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf`.

2.1 Structuring a LaTeX Document

So, how do you actually type a LaTeX document? Well, when you type, you should stop thinking "I want a 16-point bold italic bit of text on this line" and start thinking "I want a heading". LaTeX makes it very easy to typeset a document, provided you know what kind of structure you want the document to have.

First, open the editor (if you're using Windows, this is TEXnicCenter). There are certain conventions as to how you lay out the document; in particular, you can tell LaTeX about global features of the document before the start of the text in the section known as the "preamble". See for yourself:

```
\documentclass[a4paper,10pt]{article}

\usepackage{amsmath,amssymb}

\title{My First Document}
\author{Joe Bloggs}
\date{December 2008}

\begin{document}

\maketitle

\section{Heading 1}

If we want an equation inline, we use dollar signs to get, say, $x^{2} + y^{2} = z^{2}$.
If we want to ``display'' the equation, we use \verb!\[! and \verb!\]!:
\[
\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}.
\]

\end{document}
```

This tells LaTeX to use A4 paper and 10 point fonts in the class "article". (This can also be "book" or "report", but we'll not worry about that here.) The command `\usepackage` tells LaTeX to load additional functions (in this case, extra symbols which you should always load) which it can't handle natively. Then we tell it what the title, author and date we want are, and `\maketitle` is the command that actually puts the title on the page.

Mathematics is set between `$ ... $` for "in-line" maths, and between `\[ ... \]` for "displayed" maths. BE-WARE: Some things might tell you to use `$$ ... $$` for displayed maths; this is WRONG, this is a plain TEX command and will not space correctly when used in LaTeX. Anyway, this is how that code looks:

My First Document

Joe Bloggs

December 2008

1 Heading 1

If we want an equation inline, we use dollar signs to get, say, $x^2 + y^2 = z^2$. If we want to "display" the equation, we use \[and \]:

$$\zeta(2) = \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}.$$

A few points which arise from this:

- Many things can be set in environments; this is what `\begin{}` and `\end{}` do. For instance, if we want a numbered equation, we use `\begin{equation}, \end{equation}`.

- Enclosing things in braces, `{...}` makes LaTeX treat it as one object. For example, if we want to write $\int_a^b f$, we should write `\int_{a}^{b} f`; however, since each argument is one character, we can omit the braces in this case and just write `\int_a^b f`. But beware: this leads to problems when we inevitably write n^{12} as `n^12` and not `n^{12}` without thinking and get $n^1 2$.

- Commands such as `\sqrt[n]{x}` take two kind of arguments: `{...}` are mandatory arguments, while `[...]` are optional arguments which can be omitted. Thus `\sqrt{x}` simply results in $\sqrt{x}$.

There is much more to LaTeX than can be fitted into this brief summary. The documentation mentioned above is a good start; but if you're stuck, just Google it and there's very likely to be something there to help you out.

Over the page is a list of as many key commands and constructions as I could fit into one page. I hope that this will function as a ready reference for when you're starting out.

3 Commonly Used Commands and Constructions

3.1 Document structure

Headings Heading commands in descending order:
- `\part{`*title*`}`
- `\chapter{`*title*`}` (not in `article`)
- `\section{`*title*`}`
- `\subsection{`*title*`}`
- `\paragraph{`*title*`}`
- `\paragraph{`*title*`}`
- `\subparagraph{`*title*`}`

Follow with `*`, i.e. `\section*{`*title*`}`, to suppress numbering.

Lists and text environments Always close `\begin{foo}` with `\end{foo}`.
- `\begin{enumerate}` for a numbered list
- `\begin{itemize}` for a bulleted list
- `\begin{description}` for a description list

Within the environment, use `\item` *text* to add an item, and `\item[`*x*`]` to use *x* instead of the usual symbol (required for descriptions).
- `\begin{flushleft}` for a left-aligned block
- `\begin{center}` for a centred block
- `\begin{flushright}` for a right-aligned block

Alternatively, if you are already within an environment (such as a `\begin{figure}`), you can use `\raggedright`, `\centering`, or `raggedleft` respectively.
- `\begin{quote}` for an indented quotation
- `\begin{quotation}` for a quote with indented paragraphs
- `\begin{verbatim}` for reproducing code;
  - `\begin{verbatim*}` shows spaces
  - `\verb!text!` puts the text between ! and ! in verbatim

References To set a marker for an internal cross-reference, use `\label{`*marker*`}`. To refer back to it, use `\ref{`*marker*`}` to give the section/equation/figure/etc. number of the marker, or `\pageref{`*marker*`}` for just the page number. To insert a footnote, use `\footnote{`*text*`}` where you want the marker.

3.2 Figures and Tables

Tables To insert a table, use `\begin{tabular}{`*cols*`}`, where the columns are specified using:
- `l`, `c`, or `r` for left-aligned, centered or right-aligned respectively;
- `p{`*width*`}` for a paragraph box of width *width*;
- `@{`*decl*`}` to insert *decl* instead of the inter-column space;
- `|` for a vertical line between columns.

e.g. for a centered column separated by a line from a 10cm paragraph box, use `\begin{tabular}{c|p{10cm}}`. For a mathematical table inside a displayed math environment, use `\begin{array}{`*cols*`}`, with the same syntax.

Use `&` to separate columns, and `\\` to start a new row of the table; if you want a horizontal line between rows use `\hline`; if you want a cell spanning multiple columns, use `\multicolumn{`*n*`}{`*cols*`}{`*text*`}`, where `{`*cols*`}` is a column specification as above.

Floating bodies To get numbered "floating" figures and tables, use `\begin{figure}[`*place*`]` and `\begin{table}[`*place*`]` respectively, where *place* is a list of valid placements. These are `t` for top, `b` for bottom, `h` for here and `p` for on a separate page, with or without `!` to force LaTeX to place it even if ugly; e.g. if you want a figure either at the bottom of the page or on a separate page, use `\begin{figure}[bp]`. If you want the figure/table to have a number, you must give it a `\caption{`*text*`}`, and if you want to attach a label to reference the figure, you must do so immediately after the `\caption`.

3.3 Miscellaneous Text Commands

- Use `\emph{`*text*`}` to emphasise *text*.
- Use `\today` to print today's date, e.g. December 3, 2008.
- Use `~` for a non-breaking space, e.g. in `D.~M.~Q.~Mond`.
- Use `\hspace{`*l*`}` and `\vspace{`*l*`}` for horizontal and vertical spaces of length *l* respectively.
- Use `-`, `--`, and `---` for a hyphen (-) in a word, an en-dash (–) between numbers, and an em-dash (—) as punctuation respectively.
- Use `\&`, `\$`, `\%`, `\#` and `\_` if you want those symbols.

3.4 Mathematics

- To get inline mathematics, use `$ ... $`.
- To get displayed maths with no equation numbering, use `\[ ... \]`.
- To get a numbered equation, use `\begin{equation}`.
- To get a series of aligned equations, use `\begin{align}` (or `\begin{align*}` for no numbering), using `&` to indicate the alignment point and `\\` to break the line.

Mathematical Fonts
- For roman text, such as x, use `\mathrm{...}`.
- For bold text, such as **x**, use `\mathbf{...}`.
- For blackboard-bold text, such as $\mathbb{R}$, use `\mathbb{...}`.
- For typewriter text, such as `x`, use `\mathtt{...}`.

Constructions
- For superscripts and subscripts, use `x^{2}` and `x_{2}` respectively.
- For fractions such as $\frac{a}{b}$, use `\frac{a}{b}`.
- For sums and products such as $\sum_{k=1}^{n}$ and $\prod_{k=1}^{n}$, use `\sum_{k=1}^{n}` and `\prod_{k=1}^{n}`.
- For integrals such as $\int_a^b f$ use `\int_{a}^{b} f`.
- For square roots such as $\sqrt[n]{x}$, use `\sqrt[n]{x}`.
- For log, cos, etc., prefix the abbreviation with a `\`: `\log`, `\cos`.

Brackets

()	`( )`	[]	`[ ]`	||	`| |`
{}	`\{ \}`	⟨⟩	`\langle \rangle`	‖‖	`| |`

Greek letters

α	`\alpha`	ι	`\iota`	ρ	`\rho`
β	`\beta`	κ	`\kappa`	σ	`\sigma`
γ	`\gamma`	λ	`\lambda`	τ	`\tau`
δ	`\delta`	μ	`\mu`	υ	`\upsilon`
ϵ	`\epsilon`	ν	`\nu`	ϕ	`\phi`
ζ	`\zeta`	ξ	`\xi`	χ	`\chi`
η	`\eta`	o	`o`	ψ	`\psi`
θ	`\theta`	π	`\pi`	ω	`\omega`

Note omicron is `o`, not `\omicron`. Alternatives: ε `\varepsilon`, ϑ `\vartheta`, φ `\varphi`.

Γ	`\Gamma`	Ξ	`\Xi`	Φ	`\Phi`
Δ	`\Delta`	Π	`\Pi`	Ψ	`\Psi`
Θ	`\Theta`	Σ	`\Sigma`	Ω	`\Omega`
Λ	`\Lambda`	Υ	`\Upsilon`		

Other capitals are their English equivalents, e.g. `A` for capital alpha.

Symbols Miscellaneous:

$\aleph$	`\aleph`	∂	`\partial`	$\forall$	`\forall`
$\beth$	`\beth`	x'	`x'`	$\exists$	`\exists`
$\gimel$	`\gimel`	∞	`\infty`	$\emptyset$	`\emptyset`
$\daleth$	`\daleth`			$\varnothing$	`\varnothing`

Operators:

$\pm$	`\pm`	$\cap$	`\cap`	$\circ$	`\circ`
$\mp$	`\mp`	$\bigcap_1^n$	`\bigcap_{1}^{n}`	$\cdot$	`\cdot`
$\times$	`\times`	$\cup$	`\cup`	$\bullet$	`\bullet`
$\div$	`\div`	$\bigcup_1^n$	`\bigcup_{1}^{n}`		

Relations:

$\neq$	`\neq`	$\in$	`\in`	$\subset$	`\subset`
$\leq$	`\leq`	$\ni$	`\ni`	$\supset$	`\supset`
$\geq$	`\geq`	$\to$	`\to`	$\subseteq$	`\subseteq`
$\ll$	`\ll`	$\gets$	`\gets`	$\subseteqq$	`\subseteqq`
$\gg$	`\gg`	$\mapsto$	`\mapsto`	$\subsetneq$	`\subsetneq`
$\sim$	`\sim`	$\implies$	`\implies`	$\subsetneqq$	`\subsetneqq`
$\cong$	`\cong`	$\iff$	`\iff`	$\equiv$	`\equiv`