

Installation and Basics in L^AT_EX

Carolyn Wood
PAIN UQ Physics Club

August 5 2013

Abstract

LaTeX is a typesetting language which allows you to create professional-looking documents without worrying about the formatting. It is extensively used in scientific journals. If you have started 2nd year labs you may have already been introduced to it.

The purpose of this tutorial is not to make you an expert in LaTeX, but to give you the kick start which will allow you to immediately produce simple documents without having to know all the ins and outs. The best way to learn LaTeX properly is to use it, so perhaps the most important part of this tutorial is the ‘Final Advice’ section, and the References, which you can use to find out how to do just about anything LaTeX can do.

1 Installation

LaTeX is simple to install, and free!

You will need to install two things: The LaTeX documentation, and a text editor. In this tutorial we will be using pdfLaTeX, which compiles what you write into the text editor straight to a pdf file. This is the simplest way to use LaTeX when you are just starting out.

Regular LaTeX compiles first to a dvi file (which some users might need a special reader for) which then needs to be converted to a pdf file. It also only supports .eps graphics files, so a user would need a package such as Ghostscript to handle the file conversions.

For this tutorial, you will simply need two things:

1. The LaTeX documentation

If you are a Windows user, we recommend MikTeX for your documentation:

<http://miktex.org/download>

(If you have a couple of hours to spare, install the full package, otherwise the ‘Basic’ package will be fine).

If you are a Mac or Linux user, you can use TeXLive, which you can find here:

<http://www.tug.org/texlive/acquire-netinstall.html>

OR

<http://latex-project.org/ftp.html>

Windows users can also install TeXLive.

Linux users may find they already have a version of TeXLive installed. You can check for installations of TeXLive (and which version they are) in the terminal window. The command varies with the flavour of Linux, but a quick search for the equivalent of ‘rpm’ or ‘dpkg’ on the internet should provide you with information on how to do this.

Check your version is the latest by comparing it with the one offered on the TeXLive link above.

2. Text Editors

By far the most user-friendly text editor is Texmaker, so we recommend it for beginners:

<http://www.xmlmath.net/texmaker/>

Other text editors which support the tex language, such as Emacs, gedit and kate are also great, but better for people with some programming experience. You might like to check them out later.

2 Creating a Document

LaTeX commands begin with a backslash:

```
\documentclass{article}
```

The braces (curly brackets) are used to enclose the argument of the command, with other optional parameters in square brackets.

Remember all these commands are case-sensitive!

All a LaTeX document really requires is a statement declaring the type of document we want to write, and a declaration of the beginning and end of the main document:

```
\documentclass[a4paper]{article}
\begin{document}
This is the text you want in your document.
\end{document}
```

Type the above three lines into a blank text file within Texmaker, with some text which you would like in your document, and save the file as ‘test.tex.’ In Texmaker, click on Tools and select ‘pdfLaTeX.’ This compiles your document straight to a pdf file. You should see a preview of the document, with the text you added in displayed. The pdf file can also be set to be viewed in a different way - e.g. Adobe Reader. Whenever you make a change to your document within the text file, you can recompile in order to view the changes. You can use this method, or the shortcut arrow near the top of the window. Don’t be scared to play around with settings in Texmaker (teaching them is beyond the scope of this tutorial) and don’t forget to save often!

Congratulations, you have created your first LaTeX document!

Obviously we can’t stop here though. If you typed a whole paragraph into your document you might have noticed that the margins of the document are very wide. You are also missing a title, sections and section headings. So now, let’s break down the basic structure of a document.

2.1 Preamble

All LaTeX documents begin with a preamble. This is where we declare all the formatting we want applied to the entire document, as well as the packages we want to load.

The document class statement at the start of your document tells LaTeX whether you are writing a journal article, book or some other generic format. We can also add other specifications, such as font size, paper size, columns, etc. In the example above, we used the article format (good for short documents - like lab reports) and told LaTeX to format for a4paper (the default is letter paper).

Following this line, we now list the packages we want to load. These will depend on what we are intending to include in the main document. For instance, if I am going to be including equations in my document, I want to load amsmath, or a similar package which includes mathematical symbols. To do this I would add a line to the above example:

```
\documentclass[a4paper]{article}
\usepackage{amsmath}

\begin{document}
This is the text you want in your document.
\end{document}
```

Remember those wide default margins? To make better use of the space, or to fit in with assignment requirements, we can load fullpage:

```
\usepackage{fullpage}
```

Add these two lines after your documentclass declaration and recompile the document. You should notice the change in the margins. As we continue I will mention any additional packages we need to load, but these will be enough to get us started.

2.2 Top Matter

Just after the `begin` document statement is the place to add a title, author information and date. This is called the ‘top matter’. It’s just a matter of filling out the following:

```
\title{How to Install \LaTeX}  
\author{Carolyn Wood\\  
PAIN UQ Physics Club}  
\date{February 22 2013}  
\maketitle
```

The top matter must always end with the `\maketitle` line so that LaTeX knows to stop and typeset it. Notice how we can wrap text to a new line by adding double backslashes where we want the break to appear? These are useful in many places and you will see them elsewhere in this tutorial. They are only needed in places where the text won’t split on its own - to move to a new paragraph we simply press enter twice and LaTeX does the rest.

If you want a line of whitespace to appear between paragraphs in your actual document, use the `medskip` and `bigskip` commands. For more information about paragraph formatting, check out the LaTeX Wikibook: http://en.wikibooks.org/wiki/LaTeX/Paragraph_Formatting.

Add your own ‘top matter’ lines like these to your document and recompile.¹

2.3 Main Document

Now we are ready to write the main document. As we go, experiment with some of the new commands to see what they do.

2.3.1 Abstract

Since we are working with the `article` document class, we can add an abstract with the following line:

```
\begin{abstract}  
This is the abstract  
\end{abstract}
```

Abstracts are placed before the first section of the main document (though you can place this chunk of code anywhere in your main document). Add these lines (with some more text of your own) to your file and recompile to see how it looks.

2.3.2 Sections and Subsections

Now, say we would like an introduction, with a heading. To make our title ‘Introduction’ we introduce a section heading:

```
\section{Introduction}
```

If I want sections within sections, I use this next line:

```
\subsection{I am a subsection}
```

There is one more hierarchy of section headings:

```
\subsubsection{I am a sub subsection}
```

If you put all these lines into your document, you’ll notice LaTeX numbers the headings.

This is one of LaTeX’s most powerful features.

If I come back to my document and want to add a new section heading above Introduction, or some more subsections, LaTeX will automatically renumber the headings for me. Play around with this and see it in action.

What if I don’t want the numbers though? Go back to your section headings and add an asterisk just before the left-hand curly bracket. The asterisk suppresses the numbering!

¹Notice the special way of writing LaTeX? The \LaTeX symbol is a special object. If you remove the backslash from in front of LaTeX you can see the difference.

2.4 More on LaTeX Language

Before we continue, a little bit more about using the LaTeX language.

If you would like to place comments within the code, you precede the comment with a `%`. This symbol tells the LaTeX compiler to ignore the line when reading your file. Because some symbols, like `%`, are reserved for commands, if we want that symbol to appear in our text verbatim, we must precede it with a backslash. This is called ‘escaping’ the symbol. For a verbatim backslash or tilde (the `~` symbol) there are special commands:

```
\textbackslash
\textasciitilde
```

You can come back to this once we start inserting text into a document in the next section.

To write something in single quotes, we cannot just use the apostrophe key at the beginning and end. In LaTeX the left-hand quote must be typed using the key usually located to the left of the `1` key.

This is the difference between ‘this’ and ‘this.’

Once you start writing text into your document, you can experiment with how these things work.

2.5 Italics, Superscript & Subscript

To italicize a piece of text, we simply place the text to be italicized between the braces in the following command:

```
\emph{Italics!}
```

Use this in your document to see how these types of commands work.

To write a piece of text as a sub- or superscript, enclose it within these commands:

```
\textsuperscript{}
\textsubscript{}
```

To use the subscript command you will have to load a special package like this one:

```
\usepackage{fixltx2e}
```

You can also use the following shortcuts if you enter ‘math mode’ by enclosing your text within dollar signs - `$`:

```
^{text} and _{text}
```

respectively for *super^{scripts}* and *sub_{scripts}*. If you compare the two methods you’ll probably agree that the first look much better than the second.

3 Environments in LaTeX

It is important to declare to LaTeX the ‘environment’ you want to work in. This tells LaTeX how to handle the commands you give it, whether that is formatting text, or placement within a document. These declarations follow the format

```
\begin{???}
Some other commands and text.
\end{???}
```

Where the `???` is the name of the environment. Here are some common environments you will need:

3.1 Equations

LaTeX handles equations in two ways - in-line, and in separate sections with reference numbers. To write equations you will need to load the `amsmath` package. For some special mathematical symbols you will need other packages.

To include an equation within a body of text, it just needs to be surrounded by `$` signs, as already mentioned above. For instance, the code for $E = mc^2$ is

```
$E=mc^2$
```

If I want the equation displayed separately from the rest of the text, I type:

```
\begin{equation}
E=mc^2
\end{equation}
```

Which will look like:

$$E = mc^2 \tag{1}$$

Notice that LaTeX automatically numbers the equation for you. Once again, if you add more equations above this one, it will rearrange the numbers. If you want to suppress the numbering, simply add an asterisk thus:

```
\begin{equation*}
E=mc^2
\end{equation*}
```

If we want equations with more lines, properly aligned at the equals sign, we use ampersands within the *align* environment (not the *equation* environment):

```
\begin{align}
f(x) &= (x+a)(x+b) \\
&= x^2 + (a+b)x + ab
\end{align}
```

Which will give us:

$$\begin{aligned} f(x) &= (x+a)(x+b) \\ &= x^2 + (a+b)x + ab \end{aligned} \tag{2}$$

We probably don't want LaTeX to add numbers to both lines here, so we add `nonumber` to the line we don't want numbered:

```
f(x) &= (x+a)(x+b) \nonumber \\
```

Try this in your own document and see! (Make sure you don't use the double backslashes on the final line of the equation).

3.1.1 Mathematical Symbols

As a physics student, you're going to be working with many different mathematical symbols. Here is an equation which uses some of the common symbols you're likely to need, in LaTeX syntax. Type it into your document to see what it looks like:

```
\begin{equation}
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\end{equation}
```

Now I'm going to refer you to one of my favourite online resources - Scott Pakin's *Comprehensive LaTeX Symbol List*[2]: <http://www.tex.ac.uk/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>

Whenever I need a symbol I look it up in this document to find the LaTeX syntax for it. It also will tell me if I need to load a special package in order to use the symbol. If you get a chance to browse the document you'll see that there's an amazing amount of symbols supported in LaTeX, including scripts for languages and even Feynman diagrams.

Have a look at the document and use its index to help you write the following equations²

$$\Delta x \Delta p \geq \frac{\hbar}{2} \quad (4)$$

$$\oint \vec{E} \cdot d\vec{s} = -\frac{d\Phi_B}{dt} \quad (5)$$

That last one might be tricky. If you can't find a symbol easily in the document I've given you, try Google. There is a lot of documentation out there that you can use.

3.1.2 Brackets

Brackets will scale to fit around equations of any size if we use the right syntax. We already used them for the first equation in this section, but those aren't the kinds of brackets that span several lines. What about this equation?

$$\dot{a} = H_0 \left[\frac{\Omega_{r,0}}{a(t)^2} + \frac{\Omega_{m,0}}{a(t)} + \Omega_{\Lambda,0} a(t)^2 \right]^{\frac{1}{2}} \quad (6)$$

Try finding the symbols for it yourself³. The big square brackets are made using

`\left[` and `\right]`

For curly brackets we say `\left{` and `\right\}`. The `\left` and `\right` syntax can be combined with many types of symbols[2].

3.1.3 Matrices

LaTeX is very powerful for handling matrices. Copy the following lines into your document to see how they work:

```
\begin{matrix}
a & b & c \\
d & e & f \\
g & h & i
\end{matrix}
```

The first thing you should notice is that we have a special environment for working with matrices. Next, the ampersands are used to separate columns. This will be important when we look at tables. Note also that the begin and end commands must be enclosed within dollar signs, just like we would need for an in-line equation. You will also need the `amsmath` package loaded at the top of your document to use this.

What if we want nice braces around our matrix? The best way to handle that is to switch to the special environment for bracketed matrices:

```
\begin{pmatrix}
a & b & c \\
d & e & f \\
g & h & i
\end{pmatrix}
```

LaTeX has more of these environments which you can use.⁴

²You'll notice as you work in Texmaker, that it suggests possible commands for you. This is part of what makes it so user-friendly!

³The answer is in the Appendix if you get stuck.

⁴You can find some here: http://en.wikibooks.org/wiki/LaTeX/Mathematics#Matrices_and_arrays

3.2 Floats

This is where LaTeX really wins over Microsoft Word. Tables and graphics can all be types of ‘floats.’ This means that, unless specifically told, LaTeX will place these elements where ever in the document they fit best, and won’t break them up over a page (if you’ve tried to place a long table into Microsoft Word you might have encountered the frustration of a table that breaks over the page). Of course, this may mean all your nice plots end up at the end of the document, but there are ways of forcing a float to appear in a certain spot in your text. This is called a position specifier. We will look at this as we go.

3.2.1 Tables

Simple tables are quick to produce. We work in the `tabular` environment⁵

```
\begin{tabular}{c c c }
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
```

Most of the other symbols should be looking familiar to you by now. But what is `c c c` about? The `c` stands for ‘centre.’ You can also use ‘l’ and ‘r’ for left and right alignments. The number of letters corresponds to the number of columns. We can also add lines in between columns and rows:

```
\begin{tabular}{c|c|c}
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
```

OR

```
\begin{tabular}{c|c|c}
  1 & 2 & 3 \\
\hline
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
```

Take a moment to experiment with this.

What if we want to put a large amount of data into a table? It would take far too long to type it all out. LaTeX allows us to import data tables from external sources. This is beyond the scope of this little introduction, but the LaTeX Wikibook can start you off: http://en.wikibooks.org/wiki/LaTeX/Tables#Using_spreadsheets

`excel2latex` might be particularly useful for your purposes.

Now, none of the tables we’ve made so far are floats. This is fine, it just means LaTeX will place the tables right where we put them in the text, and if that means breaking them up over a page break, it will go ahead. In academic papers it’s more professional to prevent this, and so we need to put the table into a float. We do this by nesting it within the `table` environment (not to be confused with the `tabular` environment we’re already working with - notice we have both below):

```
\begin{table}[h]
  \centering
\begin{tabular}{c|c|c}
  1 & 2 & 3 \\
\hline
\end{tabular}
\end{table}
```

⁵Note: Dollar signs are not required in this case. Matrices are special as they are mathematical objects within `amsmath`.

```

4 & 5 & 6 \\
7 & 8 & 9 \\
\end{tabular}
\caption{This is a floating table!}
\end{table}

```

Before you copy this into your document, have a look at the square brackets in the first line. This is the position specifier. In this case, we have ‘h’, which stands for ‘here.’ LaTeX will then try its best to put the float at this point in the text. If we really want to make sure the float appears ‘here’ even if it’s against LaTeX’s best judgement, we add the exclamation mark⁶

LaTeX has other position specifiers, which you can find in the Wikibook: http://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions#Floats

You’ll also notice I’ve snuck in an extra ‘caption’ line. It’s pretty self-explanatory, but also very important to know! You can add it to both tables and graphics, above or below the `tabular` lines. There is also the `centering` line, which does what it says. When placing it within an environment it doesn’t need a ‘end’ command, however regular text paragraphs don’t have an environment defined, so you will need to create one with `begin{center}` and `end{center}`.

3.2.2 Graphics

First, let’s look at how we import graphics into LaTeX.

Where possible, use vector graphics. Certainly, this will be what you want to use if you write a professional journal article. This means .eps and .pdf formats are preferred to .png and .jpg. In fact, regular LaTeX will only support eps formats.

In this tutorial we are using pdfLaTeX, which handles .jpg, .png and .pdf formats (but *not* .eps). Remember, however, that if you don’t use a vectorised format, your graphics may appear pixelated if their resolution is not good enough.

To handle graphics we must first load the `graphicx` package. Add this to your preamble just as we have added previous packages.

Now we can simply use:

```
\includegraphics{picture.pdf}
```

‘picture.pdf’ refers to a picture which is stored in the same folder as your .tex file. If your picture is located elsewhere you will need to use the full path to that picture. If you don’t have a .pdf graphic to use, a .jpg format is fine, as long as the resolution is decent.

Of course, graphics come in all kinds of sizes, so we need to be able to scale them:

```
\includegraphics[scale=0.5]{picture.pdf}
```

In this case, we mean that we want the graphic to be 0.5 times the original size.

My favourite way of sizing a graphic is to specify its width in relation to the text width:

```
\includegraphics[width=\textwidth]{picture.pdf}
```

This is a quick way to add a graphic to a document, but it doesn’t allow us much sophistication. In order to add captions and other formatting options, we must embed the command within the `figure` float environment:

```

\begin{figure}[h!]
\centering
\includegraphics[width=0.8\textwidth]{picture.pdf}
\caption{Picture, yo!}
\end{figure}

```

We can also wrap text around a picture, using the `wrapfigure` environment⁷

⁶LaTeX will still ignore this if it needs to, but it is much easier to sort out *why* it has happened than in Microsoft Word.

⁷You will also need to load the `wrapfig` package at the top of the document.


```

\begin{wrapfigure}{r}{0.5\textwidth}
  \begin{center}
    \includegraphics[width=0.48\textwidth]{picture.pdf}
  \end{center}
  \caption{Picture is surrounded by text!}
\end{wrapfigure}

```

I like using subfloats, which can make your work look very professional. The Wikibook treats this subject really nicely, so have a look: http://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions#Subfloats

3.2.3 A Small Tip...

If we have an *exceptionally* large amount of figures, LaTeX will become overwhelmed. The way around this is to introduce periodic

```
\clearpage
```

lines, which tell LaTeX to stop, place the figures it already has, and start afresh with the next group.

4 Citations, References and Footnotes

Often we want to refer to a figure or equation in-text. For instance, equation 4 above is the Heisenberg Equation. Equation 5 is one of Maxwell's equations (Faraday's Law). Equation 6 is the Friedmann equation. What if I want to move equation 6 between equations 4 and 5? Now I have to find every mention of that equation in my text and re-number them, right? Nope! LaTeX allows you to place labels with equations, figures and even section headings so that you can refer to them in-text without worrying about the order changing later on.

We simply add one `label` line:

```

\begin{equation}
\dot{a}=H_0 \left[\frac{\Omega_{r,0}}{a(t)^2}+\dots\right]^{\frac{1}{2}}
\label{eq:Friedmann}
\end{equation}

```

Try this with all your equations and then, near the bottom of your document, refer back to them using this:

```
\ref{eq:Friedmann}
```

within the text⁸.

To refer to figures, use '`fig:`' instead of '`eq:`' and to refer to sections, use '`sec:`'. You will notice in Texmaker, as you begin to type the '`ref`' command, a menu of possible tags will pop up.

You will need to compile your document twice for the reference numbers to appear. This is because LaTeX looks through the whole document for the references on the first compilation, and places the right numbers on the second.

Citations from references are handled a little differently. Instead of referring to a label, they refer back to an item in a bibliography[1]. This will be properly explained in Section 6 below.

You might have noticed a few footnotes throughout this document.⁹

These are easy! We simply add the following line straight after the spot where we want the footnote's number to appear and LaTeX does the rest:

```
\footnote{This is a really nice footnote.}
```

⁸For brackets around the number, try `eqref` instead.

⁹And how professional do they look?

5 Table of Contents

A Table of Contents can be a lot of work to make, but once again, LaTeX has a simple way of compiling one.

Go back to the start of your document and add the following line straight after your abstract (you will need to compile twice for this as LaTeX need to find all the sections on the first run and make the table on the second):

```
\tableofcontents
```

Now was that easy, or was that easy? If you have unnumbered headings or other things which you would like to have appear in the table of contents, simply mark them with:

```
\addcontentsline{toc}{subsection}{??}
```

Where ?? is the title of the section as you would like it referred to in the table of contents. We also specify ‘toc’ for table of contents, because LaTeX also allows a list of figures (`listoffigures`) and a list of tables (`listoftables`). In the `addcontentsline` command above, simply use ‘lof’ and ‘loc’ instead of ‘toc.’

6 Bibliographies

6.1 Book List

A bibliography/reference list is essential for an assignment. Once you’ve come to the end of your main text, add in this line:

```
\begin{thebibliography}{9}
```

You will also want a corresponding

```
\end{thebibliography}
```

Make sure these lines are included just before the `end document` command.

Compile your document and see what this does.

You’ll notice from the syntax that the bibliography is another type of environment. The 9 in curly brackets limits your list to 9 references. Use a two digit number and you can have up to 99 references.¹⁰

Now, between the begin and end statements, we place the information for the references. Here is an example:

```
\bibitem{Knight}
  Randall D. Knight,
  \emph{Physics for Scientists and Engineers}.
  Addison-Wesley, Massachusetts,
  3rd Edition,
  2012.
```

Place the above lines into your document.

6.2 In-Text Citations

The text contained in the `bibitem` tag is the identifier we use when we want to make an in-text citation.

So, for instance, I have mentioned the symbols list document [2] and so it needs to appear in my bibliography. To cite it in-text I add the following at the point of the citation:

```
\cite{symbols}
```

where ‘symbols’ is a tag I have given the reference in the bibliography thus:

¹⁰To read about what this is doing, check out the bibliography section in the Wikibook.

```
\bibitem{symbols}
```

Use the `bibitem` tag in your bibliography entry for Knight 2012 to cite in text. Just like with the other references for equations, etc, you will need to compile twice for LaTeX to match the `bibitem` to the citation.

To add more resources, we simply add new `bibitem` entries until we're done.

6.3 Bibtex

The above is just a simple way of including resources, but it doesn't necessarily give you the format you would like, and you must enter every reference individually.

Bibtex is a more sophisticated way of handling this. It allows you to keep your references in a separate file which LaTeX then calls when typesetting your bibliography. It is a little beyond the scope of this tutorial, but I urge you to check it out and see how useful it is:

http://en.wikibooks.org/wiki/LaTeX/Bibliography_Management#BibTeX

7 Final Advice, Online Resources & Further Topics

Beyond checking out Bibtex, or how to import data tables into LaTeX, you might also like to look into RevTeX-4, which is a set of packages often used to format documents for scientific journals, adding more of a professional appearance.

LaTeX can also be used to make PowerPoint slides. Beamer is one of the packages you can use: <http://en.wikibooks.org/wiki/LaTeX/Presentations>

The two main references this document uses are by far the most useful things I can recommend to keep you going.

LaTeX is well-documented on the internet, so a simple Google search for a query will likely turn up many cases of other people having the same problem, with solutions. Often this will lead you to sites like <http://tex.stackexchange.com/> where a whole community of users help each other with queries. My best advice is - if you know what you want to do in LaTeX but not how to do it, assume that it is possible (LaTeX is very comprehensive) and that many people before you have wondered the same thing, asked about it on the internet, and found a solution. Experiment with syntax (Texmaker might suggest the very command you're looking for) and don't be afraid to ask others for help!

Go forth and LaTeX all the things!

References

- [1] *L^AT_EX Wikibook* <http://en.wikibooks.org/wiki/LaTeX>
- [2] Scott Pakin, *The Comprehensive LaTeX Symbols List*. <http://www.tex.ac.uk/tex-archive/info/symbols/comprehensive/symbols-a4.pdf> 2009.

A Appendix

But wait, I promised you an appendix! To add an appendix, we simply place these lines:

```
\appendix  
\section{Appendix}
```

You can add as many sections as you like, *and* they will appear in the table of contents. The appendices can also be placed before or after the bibliography (but everything must appear before the end document command).

A.1 Answers

Here is the answer to Equation 6:

```
\begin{equation}
\dot{a}=H_0 \left[\frac{\Omega_{r,0}}{a(t)^2}+\frac{\Omega_{m,0}}{a(t)}+\Omega_{\Lambda,0}a(t)^2\right]^{\frac{1}{2}}
\label{eq:Friedmann}
\end{equation}
```

A.2 Extra Goodies

A.2.1 Note to Physicists

\LaTeX has packages to help you write in Dirac (bra-ket) notation, and insert Feynmann diagrams!

<http://www.dfcd.net/articles/latex/latex.html>¹¹ http://www.sharelatex.com/learn/Feynman_diagrams

A.2.2 Note to Linguists and Speakers of Other Languages

There are \LaTeX packages for syntax trees, dependency trees and glosses.

<http://en.wikibooks.org/wiki/LaTeX/Linguistics> <http://www.essex.ac.uk/linguistics/external/clmt/latex4ling/>

Many special characters are available, as you will see if you browse Scott Pakin's Symbols List, as well as whole environments for languages including Chinese, Japanese and Cyrillic¹².

<http://en.wikibooks.org/wiki/LaTeX/Internationalization>

A.3 Acknowledgements

Many thanks to the various LaTeX-knowledgeable people who suggested improvements and found mistakes when this tutorial was first made at the start of 2013, and also to the beta-testers who came to it completely new for the second revision.

Without them you would have a much more inferior product to learn from!

¹¹This page is also a quick reference to some useful maths symbols.

¹²Also Tengwar... <http://tolklang.quettar.org/fonts/>