# Introduction to LaTeX

25th June 2015

## 1 WHAT IS LaTeX?

LaTeX is a typesetting language that is particularly useful for writing mathematical expressions. Most astronomical journals require articles to be submitted in LaTeX. Other word processors such as Microsoft Word follow the principle that What You See Is What You Get. In LaTeX, you don't see the final version of the document as you edit it and it requires knowledge of specialist syntax. This can mean that it has a steep learning curve, but this document should help you get started.

## 2 WHAT DO I NEED TO USE LaTeX?

To use LaTeX you need two things:

- A text editor or a TeX editor

  This is what you will you use to write your actual document. Examples of text editors include gedit, emacs and sublime. TeX editors have been designed specifically for use with LaTeX, and often come with helpful features that make writing documents in LaTeX easier. The JBCA computers all have the TeX editor TeXworks installed. This tutorial will assume you are using TeXworks.

- A TeX distribution

  A TeX distribution is required to compile your LaTeX file (that you wrote in your text

editor) to produce a PDF. There are many different TEX distributions, but the most common ones are TEXlive and MiKTEX. TEXworks is cross-platform, whereas MiKTEXis mainly targeted at Windows users.

**Tip** If you are using a TEX editor to write your documents it's important that you install a TEX distribution **before** installing the TEX editor.

## 3 GETTING STARTED

To get started on a LATEX document, first open TEXworks. TEXworks requires you to save your file before you can compile it. To save it, click on 'File → Save as' and save your document in a sensible place.

The first thing you write in your document is an instruction telling LATEX what kind of document to make. The `\documentclass[options]{class}` is used for this - the `class` determines the type of document to be created. Common document classes include:

- `article`

  For writing short reports, documentation and scientific articles.

- `report`

  For longer reports that contains chapters.

- `beamer`

  For producing presentations.

There are many more available, some of which are detailed here.

The `options` allow you to specify the default fontsize and papersize, amongst other things. For example, to create a document using the `letter` class using letter sized paper and size 10pt font, the document would begin:

```
\documentclass[10pt,letterpaper]{letter}
```

The only other commands required for a LATEX document are the `\begin{document}` and `\end{document}` commands. These commands enclose any text that you want to appear in your file. For example, the following would produce the letter mentioned above:

```
\documentclass[10pt,letterpaper]{letter}
\begin{document}
```

```
Dear Jack and Monique,

Why is JACS an acronym of an acronym?

Best wishes,

JACS students

\end{document}
```

Try copying this text into your file open in TEXworks. To compile the document click on the green arrow button. In the dropdown list next to the green arrow button, `pdflatex` should be selected. This command creates a PDF from a LATEX document, which TEXworks will open in a new window. If your TEXfile is called `<filename>.tex`, then the PDF created will be in the same directory and called `<filename>.pdf`

Exercises:

1. Create and compile a document using the report class that has size 12 font and is on A4 sized paper. Add some sample text to your document.

2. As you add sample text to your document, experiment with starting text on a new line. What happens when you start text on a new line? What happens when you leave a gap of one line between text?

In the last exercise you should notice that LATEX creates a new paragraph whenever at least one line is left between text.

## 4 TYPESETTING EQUATIONS

One of the most powerful uses of LATEX is in writing equations. To write an equation on a new line, use the `{equation}` environment. Try adding the following to your TEXdocument and see what the result is after you compile:

```
\begin{equation}
y=ax+c
\end{equation}
```

Notice that a number automatically appears on the right-hand side of the page next to the equation. If you use the `equation` environment, LATEX will automatically number your equations in the order that they appear. If you want to suppress numbering for a particular equation, use the following instead:

```
\begin{equation*}
y=ax+c
\end{equation*}
```

The `\begin{equation*}` and `\end{equation*}` commands can also be replaced by the simpler `\[` and `\]` commands.

The `{equation}` and `{equation*}` commands create equations on a new line. To write an inline equation, use the `$` symbol on either side of the equation. Try adding the following to your TEX document and note the output.

```
I have some data that I would like to fit with an line of the form $y=ax+b$.
```

The `{equation}` environments and `$` commands create a mathematical environment (sometimes called mathmode). In this mode, LATEX has commands for most mathematical operators and symbols. To produce greek letters, the syntax is usually `\<lettername>`. For example `\sigma` produces $\sigma$ and `\Sigma` produces $\Sigma$.

The following are some helpful mathematical operators:

```
x^2
\int x^2 dx
\frac{1}{4}
\frac{dy}{dx}
\frac{\partial f}{\partial x}
x_{n+1}=x_n + f_n x_n
y=a \times b
```

Try creating a mathematical enviroment and putting the above commands in. What do they produce?

A common LATEX mistake involves the − operator. Try the following in your document and note the output.

```
I'm going to attempt to work out $a$-$b$.

I'm going to attempt to work out $a-b$.
```

Many of the LATEX operators and symbols have common sense names. If you are unsure of what command you need, the best way to find the command name is to either search for it using a search engine, or to use Detexify. Detexify is an online application that asks you to draw your desired operator or symbol, which it then identifies the LATEX command for. There is also a Detexify app.

### 4.1 WHITESPACE IN MATHMODE

LaTeX will usually decide for itself how much whitespace to add around your equation. The following are some useful commands for adding or removing whitespace in equations:

```
{}              around an operator removes the whitespace around
                that operator
\;              inserts a thick space
\>              inserts a medium space
\,              inserts a thin space
\!              removes a thin space
\hspace{<len>}  inserts a space of <len> in text or mathmode. A
                negative value of <len> will remove whitespace
~               inserts an unbreakable space
\               (backslash space) inserts a space equivalent
                to a word space
```

## 5 PACKAGES

LaTeX doesn't handle everything natively. Sometimes you made need external packages to use particular symbols or environments. To use a package, you must import it in the preamble of your document. The preamble is the area between the `\documentclass` command and the `\begin{document}` command. For example, to use the `amsmath` package, the line

```
\usepackage{amsmath}
```

must be included in the preamble. The `amsmath` package introduces improvements for math environments.

Another useful package is the `geometry` package, which can be easily used to adjust the margins of packages. Try adding the following line to your preamble to see what happens:

```
\usepackage[a4paper]{geometry}
```

# 6 COMMENTS

Like most programming languages, LaTeX allows for the use of comments. Comments in LaTeX begin with % and are ignored by the compiler. They can be particularly useful when editing your work as it allows you to keep a copy of your text, even if you don't want it to all show up in your final document.

# 7 FIGURES

LaTeX doesn't handle figures natively, and instead requires use of the `graphicx` package. `pdflatex` (that we are using here) supports the following image formats:

- JPG
- PNG
- PDF

It is generally advised to use PNG, since LaTeX can have problems with JPG files.

The best way to include figures is to place them inside a float environment. Floats are LaTeX objects which cannot be split across multiple pages. Tables and figures are both floats. To create a figure, try the following using a PNG image:

```
\begin{figure}[t]
\includegraphics{<location of PNG image>}
\end{figure}
```

Note that the file extension of the image does not need to be included. For example, if your figure location is

```
/home/henson/myfigure
```

Then LaTeX will automatically search the `/home/henson/` directory for any compatible image files named `myfigure`. LaTeX also supports the use of relative paths.

What do you think the inclusion of `[t]` does? Try replacing it with `[b]` to see what happens.

`[t]` determines the position of the figure on the page. The syntax is `\begin{figure}[place modifier]`, where the possible place modifiers are:

```
h       %Place the figure in approximately the same place it occurs in the text
```

```
t       %Place the figure at the top of the page
b       %Place the figure at the bottom of the page
!       %Ignore LaTeX's attempts to put the figure in a 'good' place
```

There is one additional place modifier, `H`, which places the figure exactly where it is in the LaTeX code. This requires the use of the verb|float| package. It is similar to using `h!`.

The `\includegraphics` command can also be used to setting options for the figure. The syntax is

```
\includegraphics[options]{<location of image>}
```

For example, to make a figure of width 10cm,

```
\includegraphics[width=10cm]{<location of image>}
```

It's often useful to set the size of figures relative to the size of the page. Try the following with your PNG image and see what it looks like:

```
\begin{figure}[t]
\includegraphics[width=0.5\linewidth]{<location of PNG image>}
\end{figure}
```

Now try changing the size of the margins using the geometry package as before. What happens?

So far, the figures we've created have been left aligned. To center them, place the line

```
\centering
```

before `\includegraphics`.

There's still one key thing missing from the figure we've created: a caption. Captions are created in LaTeX using the `\caption{<Your caption>}` command. This will automatically number your figure.

LaTeX also automatically handles numbering your references to figures. When creating figures, it's good practice to assign them a label. This label is a unique identifier for that figure. It does not appear on the created PDF. Labels can also be used for equations and tables. To create a label for a figure, place

```
\label{fig:<shorthand name for figure>}
```

after the `\caption`. Labels cannot be used before captions. Labels can be used to refer to figures in text. For example, given a figure with the label `fig:spin-histogram`, then the following is used to refer to that figure in the text:

```
Figure~\ref{spin-histogram}
```

Exercise:

1. Include a PNG figure in your document. Ensure that it is centred horizontally, at the top of the page and is as wide at the text. Make sure it is appropriate caption and write a short piece of text that references the figure.

   In the text, try removing the ~ from after the word 'Figure'. What happens?

## 8 Quotation Marks

A commonly made LaTeX mistake relates to quotation marks. In LaTeX, writing a quotation mark using the keyboard quotation key will give unshaped quotation marks. To generate the quotation mark for the left-hand side, use the directed quote character twice, which results in ". To generate the quotation mark for the right-hand side, use the undirected quote character (or apostrophe) twice, which results in ".

## 9 BibTeX Guide

One of the main problems that people have when creating a LaTeX file is creating a bibliography. Luckily, the package `bibtex` can do this automatically in a few magical steps. To get a sparkling, brand new and automated bibliography you need to create a **.bib** file. This is a simple text file that contains all the information about the papers/books/articles etc. that you will be referencing. It should be located in the same folder as your LaTeX document or change the path in your editor so it can be found. In this example, we shall be referencing the classic Condon 1992 paper, *Radio Emission From Normal Galaxies*. To enter this into your .bib file you would use the syntax:

```
@article{condon1992radio,

title={Radio emission from normal galaxies},

author={Condon, JJ},

journal={Annual review of astronomy and astrophysics},
```

```
volume={30},

pages={575-611},

year={1992}
```

Surely there must be an easier way than writing this out? Well that is a yes!! Many of the websites that host papers have an export citation option and a subsequent bibtex option. This includes google scholar (look for the cite link and then the bibTeX link) and the NASA ADS website (click on the paper and then look for the BibTeX entry for this paper). This allows you to simply copy and paste the above into your .bib file. It is also handy to do this for any paper that you read, so you have all your references in one place.

To insert the bibliography into your LaTeX document you have to specify the bibliography style within the document itself using `bibliographystyle{option}`. The options determine what sort of reference style you have. In the example we will use the option `plainnat`. The LaTeX syntax is as follows:

```
\documentclass{article} \usepackage{bibtex}

\begin{document.}

Hello galaxies (\cite{condon1992radio})

\bibliography{bibliography.bib}

\bibliographystyle{plainnat}

\end{document.}
```

This gives an output of:

Hello galaxies (Condon, 1992)

# References

JJ Condon. Radio emission from normal galaxies. *Annual review of astronomy and astrophysics*, 30: 575-611, 1992.

Other styles and bibliography packages such as `natbib` can be found on the LaTeX wiki at

`https://en.wikibooks.org/wiki/LaTeX/Bibliography_Management`