GEO 242

LaTeX

Making documents in LaTeX

Outline

LaTeX: the history and basics

Simple document editing

Equations

Figures and tables

Bibliographies

Style sheets

LaTeX

LaTeX (officially "lah-tek") is a document markup language used to typeset documents.

It was developed by Leslie Lamport around 30 years ago (LaTeX = Lamport TeX).

It differs from standard word processors in that it uses plain text with tags for formatting (similar to HTML) rather than a WYSIWYG interface

It is something of a standard for scientific texts, especially for the formatting of equations

It is freely distributed, as is the TeX language that underpins it

LaTeX and TeX

LaTeX is an interface for the TeX typesetting language, which was developed by Don Knuth in the 1970s

Knuth was preparing the second edition of his book ‘The Art of Computer Programming’, and was so unhappy with the proofs that he wrote a whole new typesetting language to do it, while on sabbatical!

LaTeX is written as a series of TeX macros that simplify document layout and other tasks – the idea is to make it easier for people to use…

LaTeX elements

LaTeX is made up of several components:

- a document editor (your choice)
- a document interpreter that produces output for viewing (latex)
- add-on packages for graphics and math (graphicsx, amsmath)
- an add-on package that produces formatted bibliographies (bibtex)
- a document viewer (xdvi)
- a document printer (dvips)

Recent versions can generate PDF files (pdflatex)

Advantages and disadvantages

Pros:

- uses plain text; files are small and manageable, especially for large writing projects
- clean rendering of documents
- easy to apply different document style
- very powerful equation rendering

Cons:

- syntax is a little arcane and unwieldy
- error messages can be opaque
- not as easy to use as a WYSIWYG word processor

A barebones LaTeX document

A basic document will include some information on what kind of document it is, a beginning and an ending, and in between, some words.

```
\documentclass[12pt]{article}

\begin{document}

Hello!

\end{document}
```

Note that all of the LaTeX commands start with backslashes…

Compiling a LaTeX document

By default, the LaTeX interpreter expects your document to be saved in a file with a ‘.tex’ filetype, e.g. blah.tex. Then, to compile it, it is simply

```
latex blah
```

and to view it,

```
xdvi blah
```

If you keep xdvi open (background it), it will update the document as you recompile it…

Document classes

In the curly braces you can specify what kind of document it is supposed to be. Your main options are:

`article` – good for most short-format documents, the go-to for most general applications

`report` – for longer format documents, can have multiple chapters, etc

`book` – even longer formats

The document class you choose determines some of the default layout settings…

Document class options

In the square brackets you can specify font sizes, paper sizes and more as a comma separated list…

`10pt, 11pt, 12pt` – 10pt is default

`a4paper, letterpaper` – letter is default

`titlepage, notitlepage` – book and report have a title page by default, article doesn’t

`onecolumn, twocolumn` – one column by default

`oneside, twoside` – for binding purposes, can do asymmetric margins

Title and author

LaTeX has commands for you to define the title and author(s) of the document

```
\title{A load of nonsense}
```

```
\author{Gareth J. Funning}
```

This will not appear unless you tell it to…

```
\maketitle
```

Section headings

If you want to add some structure to your document, there is a hierarchy of headings available…

```
\section{This is a section heading}

Blah...

\subsection{This is smaller}

Here are some more words.

\subsubsection{And this is tiny}

And even more words.
```

Chapter headings

If you have a book or report, you can divide it into chapters…

```
\chapter{Introduction}

In this chapter I will have sections.

\section{I wasn’t kidding}

Here are more of those words.
```

Documents from multiple files

In a report or book, you can streamline documents by putting different sections or chapters in different ‘.tex’ documents. To include these files in your document, you use an include command, e.g.

```
\include{chapter1}
```

This will take the contents of ‘chapter1.tex’ and, starting on a new page, incorporate it in your document. [By default, LaTeX expects your files to have ‘.tex’ filetypes…]

If this file actually contains a new chapter, you still need to start it with a `\chapter{}` command

Special formatting

To render single quotes, use the backward and forward quotes:

```
 `text in quotes'
```

To render double quotes:

```
 ``text in double quotes''
```

For en-dashes

```
 -- if you want dashes for emphasis --
```

For an em-dash

```
 This is a long dash---or so they say.
```

Special formatting

For a new line:

`\newline` or `\\`

For a new page (or new column, in a two column document):

```
\newpage
```

To enforce a space between two words on the same line (useful to keep numbers with their units):

```
it was 3~km away
```

Comments are marked with % symbols

```
% Nobody is going to read this
```

Other formatting tips

If you want to include footnotes:

```
\footnote{This will place a marker at this point
and print these words at the foot of the page.}
```

For italics:

```
\textit{This is in italics} or {\it ITALICS}
```

For boldface:

```
\textbf{This is bold} or {\bf BOLD}
```

Simple math in LaTeX

If you want to include math expressions within your text, you can put the expression within dollar symbols.

```
Here is some math: $x^2 + y^2 = r^2$.
```

where the carat (^) symbol enacts superscripts. You can do subscripts using underscores, e.g.

```
Seismic moment: $M_0 = \mu A \bar{u}$
```

More involved math

For more complex equations and math representations, you will probably want to make use of the 'amsmath' package in your document, which you would declare at the beginning, e.g.

```
\documentclass{article}

\usepackage{amsmath}

\begin{document}

 ...
```

This brings some additional benefits, like a more powerful equation environment

More involved math

With amsmath enabled, if your equation demands more prominence, you can give it its own line:

```
\begin{equation}

(x+y)(x-y)=x^2-y^2

\end{equation}

\\

and then more words...
```

The equation will also be numbered

Figures

To insert PostScript or PDF figures, is fairly straightforward… Add the following line to the beginning of your .tex file

```
\usepackage{graphicx}
```

This add-on package allows LaTeX to properly include graphics

Figures

Then, the commands for a figure are as follows:

```
\begin{figure}[tbh]
 \begin{center}
  \includegraphics[width=12cm]{map.ps}
 \end{center}
 \caption{This is the caption}
\end{figure}
```

You can also specify `[width=\textwidth`] if you want to fill your page with your figure

Tables

Tables have a similar syntax:

```
\begin{table}[tbh]
 \caption{Captions go above!}
 \begin{center}
  \begin{tabular}{|l| c c |}
   <table stuff...>
  \end{tabular}
 \end{center}
\end{table}
```

tabular

When you define your table with tabular, you specify the number of columns, the justification, and whether there are vertical lines:

```
\begin{tabular}{|l| c c |}
```

will have three columns: the first left justified, surrounded by vertical lines, the other two center justified. A third vertical line will appear to the right of these.

Simply add more column descriptors to get more columns.

tabular

To fill out your table, make sure you put something in each column. The column separator is an '&':

```
\begin{tabular}{|l| c c |}

\hline

Name & Location & Cost \\

\hline

Gareth & Riverside & 20p \\

Wayne & The Moon & \$200 \\

\hline

\end{tabular}
```

Positioning your floats

Figures and tables are both considered 'floats', and will appear in the rendered document in the first good spot after they are declared.

You can specify where you would prefer them to appear using the square brackets:

`[tbh]` specifies an order of preference of 'top' (of the page), 'bottom' and 'here'.

Depending on the paragraph structure and the number of floats that have not yet appeared, the float could end up several pages deeper into the document…

Labels and references

You may want to refer to your figures, tables and equations by number. To avoid having to keep track of them, you can use labels.

Within your figure, table or equation commands, add:

```
\label{blah}
```

then, when you want to reference that in the text, use:

```
\ref{blah}
```

and it will give the number associated with the label.

Labels and references

If the second equation in your document is

```
\begin{equation}
\label{eq:taylor}
S=\sum_{n=1}^{\infty}\frac{x^n}{n!},
\end{equation}
```

Then, if you want to refer to it,

```
...equation~\ref{eq:taylor} shows...
```

will render as 'equation 2 shows'.

Bibliographies

The `bibtex` package allows you to maintain bibliographies and use in-line citations.

A bibliography (.bib) file contains entries for articles and books, e.g.

```
@article{Barka99,
   author = "Barka, A.",
   title = "The 17 {A}ugust 1999 {I}zmit earthquake",
   journal = "Science",
   year = "1999",
   volume = "285",
   number = "5435",
   pages = "1,858-1,859",
   url = "https://doi.org/10.1126/science.285.5435.1858",
}
```

Bibliographies

Another example (a book, rather than a journal article):

```
@book{AkiRich80,
  author = "Aki, K. and Richards, P. G.",
  title = "Quantitative {S}eismology: {T}heory and
  {M}ethods",
  publisher = "Freeman",
  year = "1980",
  address = "New York",
  pages = "932 pp."
}
```

Bibliographies

To use a bibliography file, you need to add `'natbib'` to your `\usepackage` statement:

```
\usepackage{graphicx,natbib}
```

Then, at the end of your text, right before your \end{document} command, you call the bibliography as follows:

```
\bibliographystyle{agufull08}

\bibliography{refs}
```

This assumes you have a style file 'agufull08.bst' and a bibliography file 'refs.bib'

Citations

With the natbib package, you can make two types of citation, `\citep` (in parentheses) and `\citet` (in text)

```
\citet{Funn-etal2007b}
```

would render '*Funning et al.* (2007)' in the text

```
\citep{Funn-etal2007b}
```

would render '(*Funning et al.*, 2007)' in the text

```
\citep{Funn-etal2007a,Funn-etal2007b}
```

would render two references in the text, separated by a semicolon

Citations

```
\citep[e.g.][Figure 1]{Funn-etal2007a}
```

would render '(e.g. *Funning et al.*, 2007; Figure 1)'

To get the bibliography to print, you should run `latex`, then `bibtex`, then `latex` once or twice more!

Style files

One of the advantages of LaTeX is that it is easy to change the rendering of a document into a proscribed format using supplied style files.

Many journals supply LaTeX template files and style files so that your submitted manuscripts can be formatted as they want them. (You can usually also render them into the journal article proof format if you want!)

UCR supplies LaTeX templates for theses and dissertations that use the correct margin widths to pass inspection…

Applying styles

Journals may provide a template `.tex` file that you can edit directly. Otherwise, they may provide a `'.cls'` (class) file that allows you to specify a custom document class, and/or a `'.sty'` (style) file that can be included with `\usepackage`

Today:

Add one of your own GMT plots as a figure, plus a table and at least two equations, to the posted example document, label them, and refer to all of them in the text.

Optionally, add some paper citations and a bibliography to your document

Send me the completed LaTeX file, along with a rendered pdf, once you are done.