An Introduction to LaTeX

Alison Ramage

16th March 2017

1 Introduction

1.1 General Comments

LaTeX is a typesetting package widely used by mathematicians. It is suitable mainly because it makes it easy to produce most mathematical formulae, cross-referencing for equations and references is automatic and it is easy to distribute the output (e.g. many journals now prefer LaTeX manuscripts submitted via the internet).

LaTeX is based on the more general language, TeX, which contains a lot of useful ways of formatting mathematics. These notes closely follow parts of

Learning LaTeX (second edition)
D.F. Griffiths and D. J. Higham, SIAM 2016

which is highly recommended as an introductory text. Other suggested references are

LaTeX : A Document Preparation System, Leslie Lamport, Addison-Wesley 1994 (2nd ed.),
The LaTeX Companion, F. Mittelbach, M. Goossens et al., Addison-Wesley 2004 (2nd ed.),

both of which contain information about LaTeX 2_ε(which is the version currently used in most Maths departments).

1.2 Running LaTeX

As LaTeX is a typesetting package rather than a word-processor, it is not an interactive on-screen package. Creating a LaTeX document is more like writing a computer program: first you must create a file containing LaTeX commands, then 'compile' it to process the commands. The precise method of doing this depends on the type of machine that you are using.

LaTeX can be run from the command line, that is, by typing commands in a shell window (in Linux) or a Command Prompt window (in Microsoft Windows). Alternatively, it is now fairly common to use a customised LaTeX User 'front-end', that is, a window-based integrated development environment for easy working with LaTeX (or TeX). There are many such environments available: examples of free versions are

- `TeXworks` (`https://tug.org/texworks/`, for Windows and other operating systems);
- `kile` (`http://kile.sourceforge.net/`, for Mac OS X and Linux operating systems);
- `overleaf` (`https://www.overleaf.com/`, online).

These notes are based on `kile`, but `TeXworks` works in a similar way. Both have many options which can be customised to your preferences once you have some experience. Note that, for this sort of tool, you may need to configure the viewing options for PS or PDF files to fit the local architecture.

In general, the steps involved in producing a document in LaTeX are:

1. Use an editor to create a file (with extension **.tex**) containing LaTeX commands, e.g. **doc.tex**.
2. Compile this, using the `latex` command or clicking on the **LaTeX** button, to produce a **.dvi** (*device independent*) file. Note: if you are using internal references, you may get the message

   ```
   LaTeX Warning: Label(s) may have changed. Rerun to get
   cross-references right.
   ```

   You must then recompile the file by running LaTeX again.
3. Preview the **.dvi** file on screen, using the `yap` command or the **ViewDVI** button, to check the document layout (the document will appear in a new window.)
4. Return to step 1: edit the file, re-LaTeX and preview the changes until you are satisfied that the document is correct.
5. Convert the **.dvi** file to a format suitable for printing (i.e., a PostScript file or a PDF file), using the `dvips` command/**DVItoPS** button, or the **DVItoPDF** button.
6. View the resulting file using an approriate viewer (or by clicking the **ViewPS** or **ViewPDF** button), and print if required.

NOTES:

1. It is possible to combine steps 2 to 5 by using the `pdflatex` command (or **PDFLaTeX** button), which LaTeXs your file and produces a PDF file directly which you can then view. This can save some time, but does not work with some packages, for example, the `beamer` package described in §8.
2. There are several different ways to produce a final PDF file:
   - using steps 2-5 above with **LaTeX** then **DVItoPDF**;
   - using **PDFLaTeX** as in note 1;
   - using **LaTeX** and **DVItoPS** followed by **PStoPDF**.

   You may have to experiment to see which of theses gives the best end result.
3. The importance of the previewing stage cannot be over-emphasised: printing is EXPENSIVE and should only be done when the document is finalised!

1.3 LaTeX errors and warnings

If your LaTeX input file contains an error, you will get an error message (usually with a line number reference) terminated with a question mark. Most messages are self-explanatory and are probably the result of typing errors, failure to match brackets or use of a control sequence (i.e. a LaTeX command) in the wrong mode. To return to the $ prompt, type `q` or `x`. To interrupt the execution of LaTeX completely, use Ctrl\c or Ctrl\x.

LaTeX also frequently generates warnings various potential problems. Mostly, these are not crucial but they can be informative. One common message is a warning about 'overfull and underfull hboxes'. The *overfull hbox* warning means that LaTeX has not been able to find a sensible place to place a line break, so the line is slightly too long for the margins. Similarly, an *underfull hbox* warning is produced when a line is too short and LaTeX has had to insert more blank space than it thinks is appropriate. You can usually just ignore these but if you are, for example, preparing your thesis or a document for journal submission, you may want to reword the offending paragraph to solve the problem. If you use the draft option with the `\documentclass` command (see below) then a black rectangle will appear in the typeset document wherever an overfull hbox occurs, which makes it easier to spot the problem lines.

1.4 Getting started

The best way of learning LaTeX commands is by looking at examples. The sample file `wpdoc.tex` and the file used to generate this document, `LaTeXnotes.tex`, are both available from the webpage

`https://alisonramage.github.io/latex_course/`

You can copy these to your own filespace and practise running LaTeX on them. (NOTE: you will also have to copy the file `fig1.ps` from the same webpage to produce the figures in this document). The slides from the accompanying talk (produced in LaTeX using the `beamer` package) are also available as LaTeX and PDF files.

2 Layout of a typical document

2.1 LaTeX commands

A LaTeX source file is a text file which contains the text which you wish to format and some LaTeX formatting commands. All LaTeX commands begin with a backslash \ and are case sensitive. A command may have a mandatory argument (enclosed in curly brackets) or an optional argument (enclosed in square brackets).

If the text of your document contains any of the following characters

$$\% * \$ \& \_ \{ \} \sim \hat{} \backslash$$

these will be interpreted as LaTeX control characters. To avoid this, use

```
\% $\ast$ \$ \& \_ \{ \} \~{} \^{} $\backslash$
```

as appropriate. Note in particular that the percentage sign `%` acts as a 'comment' symbol in LaTeX : in your LaTeX source file, anything on the line after a `%` sign will be ignored.

2.2 The preamble

Every LaTeX document begins with a standard header defining things like the type of document, the font size, the line spacing etc. This is known as the *preamble*.

The first line of every LaTeX document is a `\documentclass` statement which defines the basic format of the document. It has a mandatory argument denoting the style: the most common are `article, book, report` or `letter`. These differ mainly in the way the document is broken into sections and how the title page is formatted.

- `article`

  This is the most commonly-used style: documents of this type can be divided into sections, subsections and subsubsections, and the title is put on the first page along with an abstract and the beginning of the text (unless you specify otherwise).

- `report`

  This allows the use of chapters in addition to the above sections, although subsubsections are not numbered or indexed. The title is formed on a separate page.

- `book`

  This allows volumes as well as chapters etc. New chapters always begin on a right-hand page.

An example of the command in its simplest form is

```
\documentclass{article}
```

However, the `\documentclass` command also has several optional arguments which specify typesize, paper size etc., e.g. the command

```
\documentclass[12pt,a4]{article}
```

will use 12pt text rather than the default 10pt typesize, and use A4 paper for the default text region.

In theory, this is all that is needed in the preamble. However, it is common to insert extra commands to define the page size, margins etc. Examples are

```
\setlength{\parindent}{0.0cm}
\setlength{\textheight}{18.0cm}
\setlength{\topmargin}{5.0cm}
```

which change the amount by which new paragraphs are indented, the height of the text region and the size of the top margin respectively. You may also include any style files or extra packages you may need with the `\usepackage{`*packagename*`}` command.

The units for specifying distances in LaTeX can be specified in mm, cm, in (inches), pt (points), ex or em. A point is a unit used by typesetters where 72.27pt=1 inch. An ex is the height of a lower case x in the current font and an em is the height of an upper case M.

2.3 The document body

The text to be formatted is immediately preceded by a `\begin{document}` statement, which must be matched by an `\end{document}` statement (which will be the last line in the document). This

```
\begin{...}   ...  \end{...}
```

structure is known as an *environment*, and is an important idea in LaTeX : we will see many more examples later.

To generate a title, it is easiest to use the `\maketitle` command: see wpdoc.tex for an example.

The document is structured using paragraphs or sectioning commands. A new paragraph is started by leaving one or more lines blank in the input document. The default is for the first line of each paragraph to be indented, and for no extra vertical space to be inserted between paragraphs. This can be changed using the `\setlength` command in the preamble with `\parindent` or `\parskip`. The `\vskip` command can also be used to leave extra space between paragraphs: e.g.

```
\vskip 25mm
```

would leave an extra 25mm between this paragraph and the next. This can be useful if you need to leave space for a photograph, for example.

There are several sectioning commands in LaTeX including

- `\chapter` (available in book and report styles only)
- `\section`
- `\subsection`
- `\subsubsection`

These are used by inserting the command with section header in the appropriate place in the text, e.g. the heading for the first section in this document was produced by the command

```
\section{Introduction}
```

Note that LaTeX takes care of all of the section numbering automatically. It will also create a table of contents using all the chapter, section, subsection and subsubsection headings if required.

3 Some useful features

3.1 Cross-referencing

One major advantage of LaTeX is its ability to automatically number things like equations and references. Most LaTeX constructs (sections, equations, tables, figures etc.) can be given a label using the `\label{`*labelname*`}` command which can then be cited in the text using `\ref{`*labelname*`}`. Similarly, references can be cited using the `\cite{`*bibcode*`}` command where *bibcode* refers to the item's label in the bibliography. See the sample documents for examples of both of these constructions.

For a larger bibliography, such as that for a PhD thesis, it is a good idea to keep your reference list in a separate file and use the `BibTeX` package to keep track of citations. Details of how this is done can be found at `http://www.bibtex.org/`.

3.2 Changing fonts

A number of fonts are available in LaTeX . The following LaTeX segment

```
This text uses the default font, however, \textit{we have now switched to
italics. For some applications} \textsc{small capitals are appropriate, for
others} \textsl{slanted letters might be useful. To emphasise certain words}
\textbf{bold text can be used. To simulate output from a computer program}
\texttt{the typewriter font is often used and} \textsf{the Sans Serif font has
numerous applications. Now we are} back in the default roman font.

Font changing rules work when the relevant text and command are
enclosed in curly brackets - so we can put \textit{the rest of this sentence into
italics by using curly brackets}. Once the brackets are closed, we revert
to the original font.
```

produces the following output:

This text uses the default font, however, *we have now switched to italics. For some applications* SMALL CAPITALS ARE APPROPRIATE, FOR OTHERS *slanted letters might be useful. To emphasise certain words* **bold text can be used. To simulate output from a computer program** `the typewriter font is often used` and the Sans Serif font has numerous applications. Now we are back in the default roman font.

Font changing rules work when the relevant text and command are enclosed in curly brackets - so we can put *the rest of this sentence into italics by using curly brackets.* Once the brackets are closed, we revert to the original font.

It is also easy to change the size of the text. For example

```
\tiny This text is typeset in ``tiny'' font. \scriptsize is the next largest.
\footnotesize There are several other sizes available \normalsize which allow
you to \large produce a wide range \Large of effects, ranging \LARGE from
tiny labels to \huge large letters suitable for titles and \Huge notices.\\
```

produces

3.3 Hyphenation

LaTeX is normally fairly good about hyphenating words but will occasionally find a sentence which it cannot right-justify correctly. It will then display a warning message concerning an `overfull`

\hbox with the size in points of the error. If the line is only slightly too long (say less than 20pt) it is unlikely to be noticed. However, you can specify a "possible hyphen" in a word by using \-. So if your document contained the text `long\-word`, LaTeX would know that it was permissible to break the line at this point. Alternatively, the paragraph can be encased in a `\sloppypar` environment which causes LaTeX to relax the rules it uses for formatting paragraphs.

LaTeX has various rules which tell it where the majority of English words can be broken, and also contains a list of exceptions to these rules. If you use specialised words which LaTeX does not know about, you can add them temporarily using the `\hyphenation` command (see manuals).

You may also have a space in your text where you definitely do not want LaTeX to insert a new line. In this situation use the tilde ~ instead of a space so that the two words will be treated as a single unit, e.g. `Dr~Ramage`.

3.4 Changing the line spacing

Many documents (in particular theses) are required to be double spaced. This can be achieved by using the `doublespace` option in the documentclass command, i.e.

```
\documentclass[doublespace]{article}
```

Specific parts of the document may be singlespaced using the `singlespace` environment.

Most people find that gaps in a doublespaced document are too large. A more aesthetically pleasing effect may be obtained by putting a command such as

```
\renewcommand{\baselinestretch}{1.6}
```

in the preamble, which increases the line spacing to 1.6 times its normal value.

To force extra space between specific lines, use `\vspace*{`*x*`cm}`. The equivalent `\hspace*{`*x*`cm}` will add extra horizontal space.

3.5 Footnotes

Footnotes[1] are produced using the LaTeX command `\footnote{`*text*`}` at the appropriate point in the text[2].

3.6 Processing part of a document

It is sometimes convenient to split a LaTeX source file into several parts, e.g. one file for each chapter of a thesis. This can be done via the `\include{...}` command. Parts can then be processed separately using `includeonly{...}`, as seen in this example:

```
\documentclass{book}
\includeonly{chap1,appen1}
\begin{document}
\include{chap1}               % source file chap1.tex for Chapter 1
\include{chap2}               % source file chap2.tex for Chapter 2
\include{chap3}               % source file chap3.tex for Chapter 3
\include{appen1}              % source file appen1.tex for Appendix 1
\include{appen2}              % source file appen2.tex for Appendix 2
\end{document}
```

[1] This is an example of a footnote.

[2] Footnotes are automatically numbered.

4 Useful environments

4.1 Centred text

Output from LaTeX is normally left and right justified, i.e. the paragraphs are aligned at both sides. It is sometimes useful to have several lines of text centred on the page. This is done using the **center** environment

```
\begin{center}...\end{center}
```

Note the American spelling! For example,

```
\begin{center}
This text has been centred. A new line\\
can be forced by adding a double\\
backslash.
\end{center}
```

produces

This text has been centred. A new line
can be forced by adding a double
backslash.

4.2 Making lists

There are three basic types of lists available in LaTeX - itemised lists, numbered lists and descriptions. Some examples:

```
\begin{itemize}
\item This is the basic itemised list.
\item Each entry begins with the \verb+\item+ command.
\item Note again the American spelling.
\end{itemize}
```

produces

- This is the basic itemised list.
- Each entry begins with the `\item` command.
- Note again the American spelling.

```
\begin{enumerate}
\item Enumerated lists have numeric markers.
\item The numbering is done automatically.
\end{enumerate}
```

produces

1. Enumerated lists have numeric markers.
2. The numbering is done automatically.

```
\begin{description}
\item[The description] environment gives the option of starting each entry
with some bold text.
\item [It is useful] for describing, say, the uses of different commands.
\end{description}
```

produces

The description environment gives the option of starting each entry with some bold text.

It is useful for describing, say, the uses of different commands.

Use of these environments can be quite sophisticated:

- Each of these may be nested.
- For example
  - this is an itemised list within another list.
  - As you can see the markers have changed automatically.
- and another example ...
  1. You can also mix lists.
  2. Here is an enumerated lists within an itemised list.
- Lists can be nested up to four deep, as long as each `\begin` is matched by an `\end`.

4.3 Tables and boxes

4.3.1 The tabular environment

LaTeX uses the tabular environment to produces tables, which takes the form

```
\begin{tabular}{column_format}
entry & entry & ... & entry & entry\\
entry & entry & ... & entry & entry\\
\end{tabular}
```

where the column format parameters are

c	column of centred text
l	column of left-justified text
r	column of right-justified text
p{width}	column of specified width
|	vertical line between columns

Divisions between columns are denoted by an ampersand and a double backslash denotes the end of a line. For example, the top of the football league table might look like

TEAM	PLAYED	WON	DRAWN	LOST	GOALS	POINTS
Aberdeen	2	2	0	0	+10	6
Celtic	2	0	1	1	-5	1
Rangers	2	0	1	1	-5	1

which was produced by the LaTeX commands

```
\begin{center}
\begin{tabular}{lcccccc}
TEAM & PLAYED & WON & DRAWN & LOST & GOALS & POINTS\\
Aberdeen & 2 & 2 & 0 & 0 & +10 & 6\\
Celtic & 2 & 0 & 1 & 1& -5 & 1 \\
Rangers & 2 & 0 & 1 & 1& -5 & 1 \\
\end{tabular}
\end{center}
```

Note that the team names are left-justified while the numeric entries are centred.

4.3.2 Adding boxes

You may prefer a boxed table. This can be produced by using the vertical bar symbol (|) in the column-format field and `\hline` for horizontal bars. Changing the above input to

```
\begin{center}
\begin{tabular}{||l||c|c|c|c|c|c|}
\hline
TEAM & PLAYED & WON & DRAWN & LOST & GOALS & POINTS\\\hline\hline
Aberdeen & 2 & 2 & 0 & 0 & +10 & 6\\\hline
Celtic & 2 & 0 & 1 & 1& -5 & 1 \\\hline
Rangers & 2 & 0 & 1 & 1& -5 & 1 \\\hline
\end{tabular}
\end{center}
```

produces the boxed table

TEAM	PLAYED	WON	DRAWN	LOST	GOALS	POINTS
Aberdeen	2	2	0	0	+10	6
Celtic	2	0	1	1	-5	1
Rangers	2	0	1	1	-5	1

4.3.3 Fixed width columns

It is also possible to produces a column of a given width. The following example produces a two-column table with column widths 1.6 and 4.4 inches, respectively.

```
\begin{tabular}{p{1.6in}p{4.4in}}
\TeX & A typesetting package developed by Donald E.\ Knuth for producing
technical documents. Versions are available for many different computer
systems.\\
\LaTeX & A set of macros for \TeX which make the production of mathematics
based documents much simpler for the average user.
\end{tabular}
```

TEX	A typesetting package developed by Donald E. Knuth for producing technical documents. Versions are available for many different computer systems.
LATEX	A set of macros for TEXwhich make the production of mathematics based documents much simpler for the average user.

4.3.4 One header for several columns

The `\multicolumn` option can be used to spread captions over more than one column. For example:

```
\begin{center}
\begin{tabular}{|c|c|c|c|c|c|}  \hline
      \multicolumn{6}{|c|}{DTEST.MACRO} \\ \hline
      & Frontal &  \multicolumn{2}{c|}{PCG}
      &  \multicolumn{2}{c|}{EBE} \\ \cline{2-6}
      & CPU & CPU & Its & CPU & Its\\ \hline
      \multicolumn{6}{|c|}{EXP=1} \\ \hline
      125 elements& & & & & \\
      2262 unknowns&54.09&82.68&325 &115.20 &195 \\\hline
      1000 elements& & & & & \\
      4961 unknowns&-&1304.87&674&1304.87&1008 \\\hline
      \multicolumn{6}{|c|}{EXP=5} \\ \hline
```

```
      125 elements& & & & & \\
      2262 unknowns&56.21& 99.28 & 396&153.54&262 \\\hline
      1000 elements& & & & & \\
      4961 unknowns&-&1591.35 &822 &1783.12&1241\\\hline
\end{tabular}
\end{center}
```

produces

DTEST.MACRO					
	Frontal	PCG		EBE	
	CPU	CPU	Its	CPU	Its
EXP=1					
125 elements 2262 unknowns	54.09	82.68	325	115.20	195
1000 elements 4961 unknowns	-	1304.87	674	1304.87	1008
EXP=5					
125 elements 2262 unknowns	56.21	99.28	396	153.54	262
1000 elements 4961 unknowns	-	1591.35	822	1783.12	1241

4.3.5 The table environment

Any of the above tables can be given a caption and label for possible cross-referencing by enclosing it in the `table` environment. The `\begin{table}` command is followed by an optional argument specifying the table's position: h for here, t for top of the page or b for bottom of the page. (LaTeX may over-ride this however!). An example:

```
\begin{table}[h]
\begin{center}
\begin{tabular}{lccccc}
TEAM & PLAYED & WON & DRAWN & LOST & GOALS & POINTS\\
Aberdeen & 2 & 2 & 0 & 0 & +10 & 6\\
Celtic & 2 & 0 & 1 & 1& -5 & 1 \\
Rangers & 2 & 0 & 1 & 1& -5 & 1 \\
\end{tabular}
\caption{Scottish Premier League Table {\label{prem}}}
\end{center}
\end{table}
```

produces

TEAM	PLAYED	WON	DRAWN	LOST	GOALS	POINTS
Aberdeen	2	2	0	0	+10	6
Celtic	2	0	1	1	-5	1
Rangers	2	0	1	1	-5	1

Table 1: Scottish Premier League Table

4.4 Including figures

LaTeX provides several different options for handling images from various types of file and make them look exactly what you need. In this document, we describe one simple approach using the `graphics` package. Alternatives are (among many others) the `epsf` or `psfig` packages which will not be discussed here.

Plots etc. in PostScript (.ps) or Encapsulated PostScript (.eps) format can easily be included in a document via the `graphics` package and the `\includegraphics` command. The graphics package must be loaded by including the command

```
\usepackage{graphicx}
```

in the preamble. A figure (scaled to the correct size) can then be produced at the correct point in the text using a LaTeX segment similar to the following (which produces Figure 1):

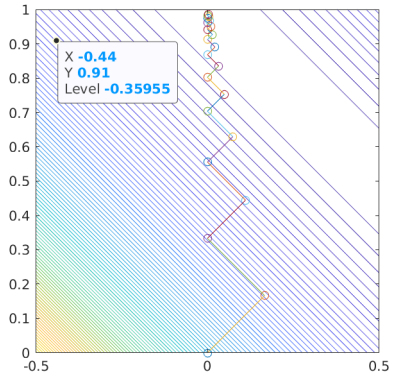


Figure 1: An example of including a picture.

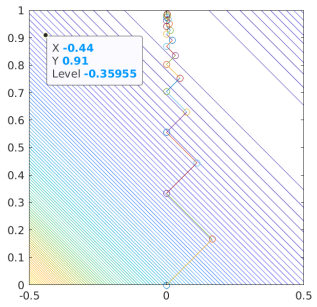
```
\begin{figure}[!hbtp]
   \begin{center}
     \scalebox{0.5}{\includegraphics{fig1.png}}
   \end{center}
   \caption{An example of including a picture.\label{fig1}}
\end{figure}
```

The `figure` environment allows the inclusion of captioned pictures and diagrams and is used in a very similar way to the `table` environment discussed above. The optional argument again specifies the figure's preferred position in order of preference: h for here, t for top of the page or b for bottom of the page (the inclusion of ! with these options means that this really is your preferred position, even if LaTeX would automatically override this). The numerical argument of `\scalebox` can be changed to alter the size of the picture.

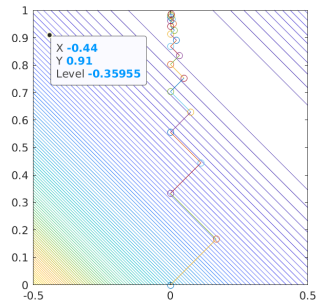
More complicated spacing of graphs can be managed by using the minipage environment. For example, Figure 2 is created by the LaTeX segment

```
\begin{figure}[hb]
\begin{center}
\mbox{
   \begin{minipage}{2.5in}
   \scalebox{0.3}{\includegraphics{fig1.png}}\\
   {\small (a) Picture 1.}\end{minipage}\qquad
   \begin{minipage}{2.5in}
   \scalebox{0.3}{\includegraphics{fig1.png}}\\
```

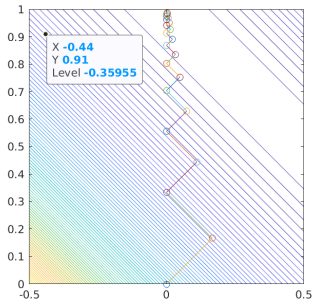
```
   {\small (b) Picture 2.}\end{minipage}
}\\
\end{center}
\begin{center}
\mbox{
   \begin{minipage}{2.5in}
   \scalebox{0.3}{\includegraphics{fig1.png}}\\
   {\small (c) Picture 3.}\end{minipage}\qquad
   \begin{minipage}{2.5in}
   \scalebox{0.3}{\includegraphics{fig1.png}}\\
   {\small (d) Picture 4.}\end{minipage}
}\\
\end{center}
\caption{An example  displaying more than one picture.\label{fig2}}
\end{figure}
```



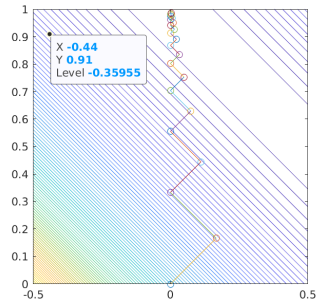
(a) Picture 1.



(b) Picture 2.



(c) Picture 3.



(d) Picture 4.

Figure 2: An example displaying more than one picture.

4.5 Verbatim

The `verbatim` environment is useful for printing things like computer code or raw LaTeX as it prints out the text exactly as it was input in typewriter font. This has been used a lot in preparing this document.

5 Mathematical Formulae

Only a few examples of the mathematical capabilities of LaTeX can be given here: for detailed information, consult the books listed in section 1 and the sample file wpdoc.tex or search online.

5.1 Math mode

Mathematics in LaTeX must be typeset in math mode. There are two basic forms: *inlinemode* and *displaymode*. Inline mode is indicated by enclosing the LaTeX command between two $ signs. For example,

```
This line contains $E=mc^2$, $A=2\pi r$ or $x=\frac{1}{2}y$.
```

produces

This line contains $E = mc^2$, $A = 2\pi r$ or $x = \frac{1}{2}y$.

There are many ways of using displaymode. The most usual is the *displaymath* environment, which centres the formula on the page in its own space, e.g.

```
\begin{displaymath}
v_{i}=\sin{\frac{i\pi}{N+1}}.\\
\end{displaymath}
```

gives

$$v_i = \sin \frac{i\pi}{N+1}.$$

The same effect is obtained by enclosing the expression in double dollar signs `$$...$$`.

If you wish to number an equation for reference purposes, use instead the `equation` environment:

```
\begin{equation}
\label{psieq}
\psi_{i}=\frac{1}{6i}-\ln{\frac{1}{2}}.\\
\end{equation}
```

which gives

$$\psi_i = \frac{1}{6i} - \ln \frac{1}{2}. \tag{1}$$

The `\label` statement means that you may refer to the equation in the text using `\ref{psieq}`.

To format several lines of equations, use the `eqnarray` environment. As in the `tabular` environment, equations are lined up using ampersands. For example,

$$\begin{aligned} \epsilon\nabla^2\psi &= -\rho && (2)\\ q\frac{\partial p}{\partial t} &= -\nabla.J_p - qR \\ q\frac{\partial n}{\partial t} &= -\nabla.J_n - qR && (3) \end{aligned}$$

was produced by

```
\begin{eqnarray}
\epsilon\nabla^2\psi & = & -\rho \\
q\frac{\partial p}{\partial t} & = & -\nabla.J_p-qR \nonumber \\
q\frac{\partial n}{\partial t} & = & -\nabla.J_n-qR
\end{eqnarray}
```

Note the use of `\nonumber` in the second line. If you wish none of the equations to have numbers, use the `eqnarray*` environment instead.

5.2 Building formulae

The strategy for building mathematical formulae is quite straightforward: simply start at the left and work along "translating" each term and remembering to match all brackets. Some common expressions:

subscripts - `$expr1_{expr2}$` produces $expr1_{expr2}$
superscripts - `$expr1^{expr2}$` produces $expr1^{expr2}$
fractions - `$\frac{expr1}{expr2}$` produces $\frac{expr1}{expr2}$
square root - `$\sqrt{expr1}$` produces $\sqrt{expr1}$
sum - `$\sum_{k=1}^n a_k$` produces $\sum_{k=1}^{n} a_k$
integral - `$\int_{x=0}^\infty e^{-x^2}dx$` produces $\int_{x=0}^{\infty} e^{-x^2}dx$
limit - `$\lim_{n\to\infty}(1-\frac{x}{n})^n$` produces $\lim_{n\to\infty}(1-\frac{x}{n})^n$
derivative - `$\frac{\partial f}{\partial x}$` produces $\frac{\partial f}{\partial x}$

Horizontal space in equations can be added using `\quad` and `\qquad`: smaller spaces come from e.g. `\,` (thin space), `\:` (medium space), `\;` (thick space) or `\!` (negative thin space).

Hats, tildes and other 'maths accents' may also be easily obtained. For example,

```
\begin{displaymath}
\hat{x},\,\tilde{v},\:\dot{u},\;\ddot{u},\quad\underline{f+g},\qquad\overline{a+b}.
\end{displaymath}
```

produces

$$\hat{x},\ \tilde{v},\ \dot{u},\ \ddot{u},\quad \underline{f+g},\qquad \overline{a+b}.$$

You can typeset your formulae in different fonts using the commands

`\mathbf`, `\mathit`, `\mathsf`, `\mathcal`

to obtain bold, italic, sans serif and calligraphic fonts, respectively (used in the same way as for normal text). To produce bold versions of Greek letters, one option is to use the `\bm` command from the `bm` package. Note that spaces are ignored in math mode: to put some normal text into a maths expression, use the `\mbox` command. For example, use

```
\begin{displaymath}
x^2-1>0\mbox{ for all }x>1.
\end{displaymath}
```

to produce

$$x^2-1>0 \text{ for all } x>1.$$

5.3 Brackets

In addition to square and round brackets, we may use curly brackets in equations provided they are proceeded by a backslash `\{...\}`. To get brackets adjusted in size to fit a particular formula, use

```
\left\{ \left[ \left( \left| ... \right| \right) \right] \right\}
```

noting that each `\left` MUST be accompanied by a `\right`, even if the bracket type is not the same (use `\left.` or `\right.` if you don't want any bracket printed).

5.4 Arrays

The `array` environment is used for formatting arrays or matrices, and it must be used in mathmode. It is similar to the `tabular` environment: the position of the entries within each column is specified by c, l or r, entries are separated by ampersands and lines are ended by a double backslash. For example, the LaTeX segment

```
\begin{displaymath}
A=\left[\begin{array}{ccc}
1 & 1 & 1\\
x & y & z\\
x^2 & y^2 & z^2
\end{array}\right], \quad
x=\left[\begin{array}{r}
1 \\ -2 \\ 4
\end{array}\right]
\end{displaymath}
```

produces

$$A = \left[\begin{array}{ccc} 1 & 1 & 1 \\ x & y & z \\ x^2 & y^2 & z^2 \end{array}\right], \quad x = \left[\begin{array}{r} 1 \\ -2 \\ 4 \end{array}\right]$$

Ellipsis of various kinds may also be useful here. For example,

```
\begin{displaymath}
T=\left[\begin{array}{ccccc}
a & b & 0 & \cdots & 0\\
c & a & b & &\vdots\\
0 &\ddots&\ddots&\ddots& 0\\
\vdots & &c & a & b \\
0&\cdots &0&c & a
\end{array}\right]
\end{displaymath}
```

produces

$$T = \left[\begin{array}{ccccc} a & b & 0 & \cdots & 0 \\ c & a & b & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & c & a & b \\ 0 & \cdots & 0 & c & a \end{array}\right]$$

6 Abbreviations

Typing common LaTeX commands over and over can be tedious. New LaTeX commands which are shorter can be defined using `\newcommand`. For example, if the abbreviations

```
\newcommand{\beq}{\begin{equation}}
\newcommand{\eeq}{\end{equation}}
```

are defined in the preamble, then LaTeX will expand every occurrence of `\beq` in the text to `\begin{equation}`. See the text of this document for more examples.

7 Packages and style files

The basic LaTeX commands can be greatly enhanced by using packages and style files. For example, many publishers provide style files (which can be downloaded from their website) to be used by contributing authors to set up articles in the journal's preferred style. The American Mathematical Society packages `amsmath`, `amsfonts` and `amssymb` can be particularly useful for producing special mathematics characters like $\mathbb{R}$, $\mathbb{N}$, $\mathbb{Z}$ etc. Several common packages are already distributed with `MiKTeX` and `kile`. Many other packages and style files are accessible via the UKTeXarchive `http://www.tex.ac.uk`. These can also be customised to produce personalised style files.

You can include a package or style file in your LaTeX document using the `\usepackage` command in the preamble. For example,

```
\usepackage{exam}
```

will include the style file **exam.sty**.

8 Slides and presentations

There is more than one way to prepare slides in LaTeX. One popular style to use is

```
\documentclass{beamer},
```

which is very similar to the `article` style except that the `frame` environment

```
\begin{frame} ... \end{frame}
```

is used to produces a series of separate slides. The slide style, colour etc. can be specified using the `\theme` and `\colortheme` commands. There are many sources of further information about `beamer` available online. In particular, The LaTeX file used to produce the slides which accompany these notes, `LaTeXslides.tex`, is available on the course webpage (see §1.4). Note that slides produced with `beamer` cannot usually be viewed in DVI format: you should first convert to PS or PDF format to view your draft slides.

9 Posters

One method of producing a large poster to display your research results is by using the `a0poster` document class

```
\documentclass[a0]{a0poster}
```

(although other classes can be used). This will produce an A0 paper sized poster in landscape format. Other than offering additional font sizes, such as

```
\veryHuge, \VeryHuge, and \VERYHuge,
```

this is very like the `article` document class.