CSE 376: Technical Writing and Presentation

LaTeX Practicals

Prepared from LaTeX for Beginners, University of Edinburgh.
Copyright ©IS 2014

December 3, 2022

Introduction

What is LaTeX?

LaTeX (pronounced *lay-tek*) is a document preparation system for producing professional-looking documents, it is **not** a word processor. It is particularly suited to producing long, structured documents, and is very good at typesetting equations. It is available as free software for most operating systems.

LaTeX is based on TeX, a typesetting system designed by Donald Knuth in 1978 for high quality digital typesetting. TeX is a low-level language that computers can work with, but most people would find difficult to use; so LaTeX has been developed to make it easier.

If you are used to producing documents with Microsoft Word, you will find that LaTeX is a very different style of working. Microsoft Word is 'What You See Is What You Get' (WYSIWYG), this means that you see how the final document will look as you are typing. When working in this way you will probably make changes to the document's appearance (such as line spacing, headings, page breaks) as you type. With LaTeX you do not see how the final document will look while you are typing it — this allows you to concentrate on the content rather than appearance.

A LaTeX document is a plain text file with a .tex file extension. It can be typed in a simple text editor such as Notepad, but most people find it easier to use a dedicated LaTeX editor. As you type you mark the document structure (title, chapters, subheadings, lists etc.) with tags. When the document is finished you compile it — this means converting it into another format. Several different output formats are available, but probably the most useful is Portable Document Format (PDF), which appears as it will be printed and can be transferred easily between computers.

Before You Start

The following conventions are used throughout the Practical sheets:

- Actions for you to carry out are bulleted with an arrow ➲.

- Text you type is written in this font.

- Menu commands and button names are shown in **bold**.

Practical 1: Document Structure

1.1 Essentials

➲ Start TeXworks.

A new document will automatically open.

➲ Go to the **Format** menu and select **Line Numbers**.

Line numbers are not essential, but will make it easier to compare your code with the screenshots and find errors.

➲ Go to the **Format** menu and select **Syntax Coloring**, then **LaTeX**.

Syntax colouring will highlight commands in blue and can make it easier to spot mistakes.

➲ Type the following:

```
\documentclass[a4paper,12pt]{article}

\begin{document}

A sentence of text.

\end{document}
```

The `\documentclass` command must appear at the start of every LaTeX document. The text in the curly brackets specifies the document class. The **article** document class is suitable for shorter documents such as journal articles and short reports. Other document classes include **report** (for longer documents with chapters, e.g. PhD theses), **proc** (conference proceedings), **book** and **slides**. The text in the square brackets specifies options — in this case it sets the paper size to A4 and the main font size to 12pt.

The `\begin{document}` and `\end{document}` commands enclose the text and commands that make up your document. Anything typed before `\begin{document}` is known as the preamble, and will affect the whole document. Anything typed after `\end{document}` is ignored.

The empty lines aren't necessary[1], but they will make it easier to navigate between the different parts of the document as it gets longer.
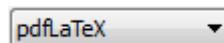
➲ Click on the **Save** button.

➲ Create a new folder called **LaTeX Practicals** in **Documents**.

➲ Name your document **Lab1** and save it as a **TeX document** in this folder.

It is a good idea to keep each of your LaTeX documents in a separate folder as the compiling process creates multiple files.

➲ Make sure the typeset menu is set to **pdfLaTeX**.

➲ Click on the **Typeset** button.

There will be a pause while your document is being converted to a PDF file. When the compiling is complete TeXworks' PDF viewer will open and display your document. The PDF file is automatically saved in the same folder as the .tex file.

---

[1] See section 2.5 for information about how LaTeX deals with empty space in the .tex file.

1.2 Troubleshooting

If there is an error in your document and TeXworks cannot create the PDF the **Typeset** button will change to red with a white X (**Abort typesetting** button) and the **Console output** at the bottom of the screen will stay open. If this happens:

➲ Click on the **Abort typesetting** button.

➲ Read the **Console output** - the last line will probably include a line number and the command that caused the error.

➲ Go to the line number in your document and fix the error.

➲ Click on the **Typeset** button again.

1.3 Creating a Title

The `\maketitle` command creates a title. You need to specify the title of the document. If the date is not specified today's date is used. Author is optional.

➲ Type the following directly after the `\begin{document}` command:
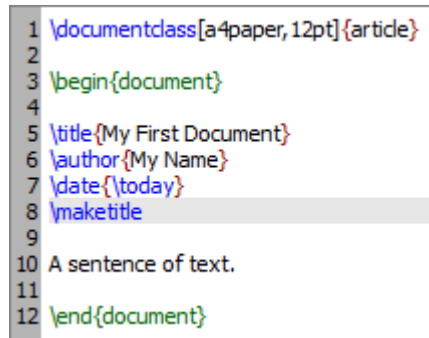
```
\title{My First Document}
\author{My Name}
\date{\today}
\maketitle
```

Your document should now look like figure 1.

➲ Click on the **Typeset** button and check the PDF.

Points to note:

- `\today` is a command that inserts today's date. You can also type in a different date, for example `\date{December 2018}`.

- **Article** documents start the text immediately below the title on the same page. **Reports** put the title on a separate page (like this Practical Sheet).

Figure 1: TeXworks screenshot showing the maketitle command.

1.4 Sections

You should divide your document into chapters (if needed), sections and sub-sections. The following sectioning commands are available for the **article** class:

- \section{...}

- \subsection{...}

- \subsubsection{...}

- \paragraph{...}

- \subparagraph{...}

The title of the section replaces the dots between the curly brackets. With the **report** and **book** classes we also have \chapter{...}.

➲ Replace "A sentence of text." with the following:

```
\section{Introduction}
This is the introduction.

\section{Methods}

\subsection{Stage 1}
The first part of the methods.
```
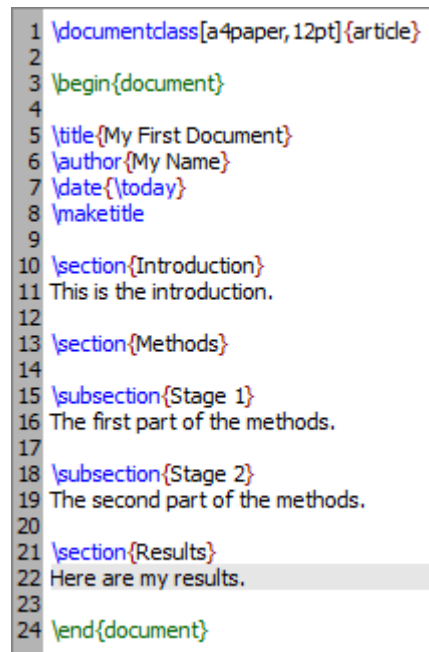
```
\subsection{Stage 2}
The second part of the methods.

\section{Results}
Here are my results.
```

Your document should now look like figure 2.

Figure 2: TeXworks screenshot of document with sections.

➲ Click on the **Typeset** button and check the PDF.

1.5 Labelling

You can label any of the sectioning commands so they can be referred to in other parts of the document. Label the section with `\label{labelname}`. Then type `\ref{labelname}` or `\pageref{labelname}`, when you want to refer to the section or page number of the label.

➲ Type `\label{sec1}` on a new line directly below `\subsection{Stage 1}`.

➲ Type `Referring to section \ref{sec1} on page \pageref{sec1}` in the **Results** section.

Your document should now look like figure 3.

```
1  \documentclass[a4paper, 12pt]{article}
2
3  \begin{document}
4
5  \title{My First Document}
6  \author{My Name}
7  \date{\today}
8  \maketitle
9
10 \section{Introduction}
11 This is the introduction.
12
13 \section{Methods}
14
15 \subsection{Stage 1}
16 \label{sec1}
17 The first part of the methods.
18
19 \subsection{Stage 2}
20 The second part of the methods.
21
22 \section{Results}
23 Here are my results. Referring to section \ref{sec1} on page \pageref{sec1}
24
25 \end{document}
```

Figure 3: TeXworks screenshot of document with labels.

➲ Click on the **Typeset** button and check the PDF. You may need to typeset the document twice before the references appear in the PDF.

1.6 Table of Contents
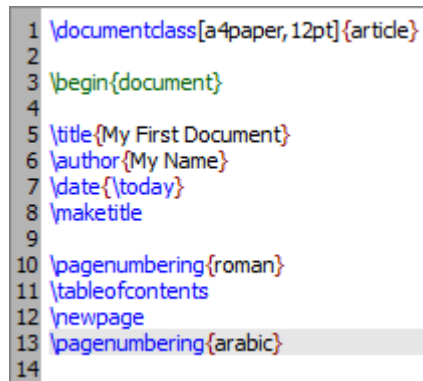
If you use sectioning commands it is very easy to generate a table of contents. Type `\tableofcontents` where you want the table of contents to appear in your document — often directly after the title page.

You may also want to change the page numbering so that roman numerals (i, ii, iii) are used for pages before the main document starts. This will also ensure that the main document starts on page 1. Page numbering can be switched between arabic and roman using `\pagenumbering{...}`.

➲ Type the following on a new line below `\maketitle`:

```
\pagenumbering{roman}
\tableofcontents
\newpage
\pagenumbering{arabic}
```

The `\newpage` command inserts a page break so that we can see the effect of the page numbering commands. The first 14 lines of code should now look like figure 4.

Figure 4: TeXworks screenshot of document showing table of contents command.

➲ Click on the **Typeset** button and check the PDF.

Checkpoint 1

When all the tasks are done, show your final tex file and pdf file to your instructor.

Practical 2: Typesetting Text

➲ Change your previous practical .tex file's **article** document class into **report**. Arrange each practical title as chapter title, such as:

```
\chapter*{Practical 1: Document Structure}
\chapter*{Practical 2: Typesetting Text}
```

Note that, here `\chapter` creates numbered chapter title and `\chapter*` creates unnumbered chapter title. The unnumbered chapter will create two issues. Firstly, the section and subsection number of the chapter will start with prefix 0; such as 0.1, 0.2 etc. Secondly, chapter title will not be visible in the table of contents. To solve the first issue, increment the Chapter number first.

➲ Type `\setcounter{chapter}{1}` on the line before `\begin{document}`.

To solve the second issue, use the `\addcontentsline` command.

➲ Type `\addcontentsline{toc}{chapter}{Chapter title}` just before the `\chapter*{Chapter title}`.

2.1 Font Effects

There are LaTeX commands for a variety of font effects:

```
\textit{words in italics}        words in italics
\textsl{words slanted}           words slanted
\textsc{words in smallcaps}      WORDS IN SMALLCAPS
\textbf{words in bold}           words in bold
\texttt{words in teletype}       words in teletype
\textsf{sans serif words}        sans serif words
\textrm{roman words}             roman words
\underline{underlined words}     underlined words
```

➲ Add some more text to your document and experiment with different text effects.

2.2 Coloured Text

To put coloured text in your document you need to use a **package**. There are many packages that can be used with LaTeX to enhance its functionality. Packages are included in the **preamble** (i.e. before the `\begin{document}` command). Packages are activated using the `\usepackage[options]{package}` command, where `package` is the name of the package and `options` is an optional list of keywords that trigger special features in the package.

The basic colour names that `\usepackage{color}` knows about are black, red, green, blue, cyan, magenta, yellow and white:

Red, green, blue, cyan, magenta, yellow and white .

The following code to produces coloured text:

```
{\color{colour_name}text}
```

Where `colour_name` is the name of the colour you want, and `text` is the text you want to be coloured.

➲ Type `\usepackage{color}` on the line before `\begin{document}`.

➲ Type `{\color{red}fire}` in your document.

➲ Click on the **Typeset** button and check the PDF.

The word 'fire' should appear in red.

It is possible to add options that allow `\usepackage{color}` to understand more colour names, and even to define your own colours. It is also possible to change the background colour of text (as for white and yellow in the example above), but this is beyond the scope of this workbook. If you want more information about see the Colors chapter in the LaTeX Wikibook[2].

2.3 Font Sizes

There are LaTeX commands for a range of font sizes:

`{\tiny tiny words}`	tiny words
`{\scriptsize scriptsize words}`	scriptsize words
`{\footnotesize footnotesize words}`	footnotesize words
`{\small small words}`	small words
`{\normalsize normalsize words}`	normalsize words
`{\large large words}`	large words
`{\Large Large words}`	Large words
`{\LARGE LARGE words}`	LARGE words
`{\huge huge words}`	huge words

➲ Experiment with different font sizes in your document.

2.4 Lists

LaTeX supports two types of lists: **enumerate** produces numbered lists, while **itemize** is for bulleted lists. Each list item is defined by `\item`. Lists can be nested to produce sub-lists.

➲ Type the following to produce a numbered list with a bulleted sub-list:

```
\begin{enumerate}
\item First thing
\item Second thing
```

---

[2]`http://en.wikibooks.org/wiki/LaTeX/Colors`

```
\begin{itemize}
\item A sub-thing
\item Another sub-thing
\end{itemize}
\item Third thing
\end{enumerate}
```

➲ Click on the **Typeset** button and check the PDF.

The list should look like this:

1. First thing
2. Second thing
   - A sub-thing
   - Another sub-thing
3. Third thing

It is easy to change the bullet symbol using square brackets after the `\item`, for example, `\item[-]` will give a dash as the bullet. You can even use words as bullets, for example, `\item[One]`.

The following code:

```
\begin{itemize}
\item[-] First thing
\item[+] Second thing
\begin{itemize}
\item[Fish] A sub-thing
\item[Plants] Another sub-thing
\end{itemize}
\item[Q] Third thing
\end{itemize}
```

Produces:

- First thing
+ Second thing
  Fish A sub-thing
Plants Another sub-thing
Q Third thing

2.5 Comments & Spacing

Comments are created using %. When LaTeX encounters a % character while processing a .tex file, it ignores the rest of the line (until the [**Return**] key has been pressed to start a new line — not to be confused with line wrapping in your editor). This can be used to write notes in the input file which will not show up in the printed version.

The following code:

```
Believe that life is worth living% Note comic irony
in the very first sentence
, and your belief will help create the fact.
```

Produces:

Believe that life is worth living, and your belief will help create the fact.

Multiple consecutive spaces in LaTeX are treated as a single space. Several empty lines are treated as one empty line. The main function of an empty line in LaTeX is to start a new paragraph. In general, LaTeX ignores blank lines and other empty space in the .tex file. Two backslashes (\\) can be used to start a new line.

⟳ Experiment with putting comments and blank lines in to your document.

If you want to add blank space into your document use the `\vspace{...}` command. This will add blank vertical space of a height specified in typographical points (pt). For example, `\vspace{12pt}` will add space equivalent to the height of a 12pt font.

2.6 Special Characters

The following symbols are reserved characters which have a special meaning in LaTeX:

```
#   $   %   ^   &   _   {   }   ~   \
```

All of these apart from the backslash \ can be inserted as characters in your document by adding a prefix backslash:

`\#  \$  \%  \^{} \&  \_  \{  \}  \~{}`

Note that you need to type a pair of curly brackets `{}` after the hat `^` and tilde `~`, otherwise these will appear as accents over the following character. For example, "`\^ e`" produces "ê".

The above code will produce:

$ %   ^   &   _   {   }   ˜

The backslash character \ can not be entered by adding a prefix backslash, `\\`, as this is used for line breaking. Use the `\textbackslash` command instead.

➲ Type code to produce the following sentence in your document:

Item #1A\642 costs $8 & is sold at a ˜10% profit.

Checkpoint 2

When all the tasks are done, show your final tex file and pdf file to your instructor.

Practical 3: Tables

The `tabular` command is used to typeset tables. By default, LaTeX tables are drawn without horizontal and vertical lines — you need to specify if you want lines drawn. LaTeX determines the width of the columns automatically.

This code starts a table:

```
\begin{tabular}{...}
```

Where the dots between the curly brackets are replaced by code defining the columns:

- `l` for a column of **left**-aligned text (letter el, *not* number one).

- `r` for a column of **right**-aligned text.

- `c` for a column of **centre**-aligned text.

- `|` for a vertical line.

For example, `{lll}` (i.e. left left left) will produce 3 columns of left-aligned text with no vertical lines , while `{|l|l|r|}` (i.e. |left|left|right|) will produce 3 columns — the first 2 are left-aligned, the third is right-aligned, and there are vertical lines around each column.

The table data follows the `\begin` command:

- `&` is placed between columns.

- `\\` is placed at the end of a row (to start a new one).

- `\hline` inserts a horizontal line.

- `\cline{1-2}` inserts a partial horizontal line between column 1 and column 2.

The command \end{tabular} finishes the table.

Examples of tabular code and the resulting tables:

```
\begin{tabular}{|l|l|}
Apples & Green \\
Strawberries & Red \\
Oranges & Orange \\
\end{tabular}
```

Apples	Green
Strawberries	Red
Oranges	Orange

```
\begin{tabular}{rc}
Apples & Green \\
\hline
Strawberries & Red \\
\cline{1-1}
Oranges & Orange \\
\end{tabular}
```

Apples	Green
Strawberries	Red
Oranges	Orange

```
\begin{tabular}{|r|l|}
\hline
8 & here's \\
\cline{2-2}
86 & stuff \\
\hline \hline
2008 & now \\
\hline
\end{tabular}
```

8	here's
86	stuff
2008	now

List of Tables

If you use captions for the tables, it is very easy to generate a list of tables. Type \listoftables where you want the list of table to appear in your document - often directly after the Table of Contents.

- \caption{....} writes the caption of the table.

- \label{....} You can label the table so that it can be referred to in other parts of the document.

Multi-column and multi-row cells

Rows and columns can be combined in a bigger cell. The example below is an example of the `\multicolumn` command to combine columns.

```
\begin{tabular}{|c|c|c|c|}
\hline
\multicolumn{4}{|c|}{Country List} \\
\hline
Country Name & ALPHA 2 Code &ALPHA 3 Code&Numeric Code\\
\hline
Afghanistan & AF & AFG & 004\\
Albania & AL & ALB & 008\\
Algeria & DZ & DZA &  012\\
Angola & AO & AGO & 024 \\
\hline
\end{tabular}
```

Country List			
Country Name	ALPHA 2 Code	ALPHA 3 Code	Numeric Code
Afghanistan	AF	AFG	004
Albania	AL	ALB	008
Algeria	DZ	DZA	012
Angola	AO	AGO	024

Checkpoint 3

↻ Write code to produce the following tables:

Item	Quantity	Price ($)
Nails	500	0.34
Wooden boards	100	4.00
Bricks	240	11.50

	Year		
City	2006	2007	2008
London	45789	46551	51298
Berlin	34549	32543	29870
Paris	49835	51009	51970

Practical 4: Figures and Equations

4.1 Figures

In this practical. you will learn how to insert an image in to your LaTeX document, which requires the `graphicx` package. Images should be PDF, PNG, JPEG or GIF files. The following code will insert an image called myimage:

```
\begin{figure}[h]
\centering
\includegraphics[width=1\textwidth]{myimage}
\caption{Here is my image}
\label{image-myimage}
\end{figure}

Figure \ref{image-myimage} shows my image.
```

`[h]` is the placement specifier. `h` means put the figure approximately here (if it will fit). Other options are `t` (at the top of the page), `b` (at the bottom of the page) and `p` (on a separate page for figures). You can also add `!`, which overrides the rule LaTeX uses for choosing where to put the figure, and makes it more likely it will put it where you want (even if it doesn't look so good).

`\centering` centres the image on the page, if not used images are left-aligned by default. It's a good idea to use this as the figure captions are centred.

`includegraphics{...}` is the command that actually puts the image in your document. The image file should be saved in the same folder as the .tex file.

`[width=1\textwidth]` is an optional command that specifies the width of the picture - in this case the same width as the text. The width could also

be given in centimeters (cm). You could also use `[scale=0.5]` which scales the image by the desired factor, in this case reducing by half.

`\caption{...}` defines a caption for the figure. If this is used LaTeX will add "Figure" and a number before the caption. If you use captions, you can use `\listoffigures` to create a table of figures in a similar way to the table of contents (section 1.6, page 8).

`\label{...}` creates a label to allow you to refer to the table or figure in your text (section 1.5, page 7).

Checkpoint 4.1

➲ Add `\usepackage{graphicx}` in the preamble of your document (before the `\begin{document}` command).

➲ Find an image and save a copy to your **LaTeX course** folder.

➲ Type the following text at the point where you want your image inserted:

```
\begin{figure}[h!]
\centering
\includegraphics[width=1\textwidth]{ImageFilename}
\caption{My test image}
\end{figure}
```

Replace `ImageFilename` with the name of your image file, excluding the file extension. If there are any spaces in the file name enclose it in quotation marks, for example `"screen 20"`.

➲ Type `\listoffigures` where you want the list of figures to appear in your document - usually directly after the Table of Contents.

➲ Click on the **Typeset** button and check the PDF.

4.2 Sub Figures

Subfigures are nothing but a series of figures under the same figure environment. The use of subfigures is interesting when you have a set of related

images since you can put all of them one next to the other and under the same caption, so that they produce a more organized and space-optimized document.

Checkpoint 4.2

➲ Write an example with two images along side each other with separate captions and labels.

➲ You need to use `subcaption` package to create subfigures.

➲ To start with, create a new figure, centre it and then create two subfigure environments. Each of these takes a mandatory argument, which is the width to be assigned to each one of them, in this case `0.4\textwidth`, that is, 40% of the width of the text in the document.

➲ Each of these environments is treated inside as a usual figure environment: load the corresponding image with `\includegraphics` (setting its width to the width of the text inside the environment with `width=\textwidth`), and then give it a `\caption` and a `\label`.

➲ Finally, give the whole figure environment its own `\caption` and `\label`.

Here is the example code:

```
\begin{figure}[h]
\begin{subfigure}{0.4\textwidth}
\includegraphics[width=\textwidth]{imageFile1}
\caption{Caption1}
\label{fig:subimage1}
\end{subfigure}
\begin{subfigure}{0.4\textwidth}
\includegraphics[width=\textwidth]{imageFile2}
\caption{Caption 2}
\label{fig:subimage2}
\end{subfigure}

\caption{Caption for this figure with two images}
\label{fig:image2}
\end{figure}
```

4.3 Equations

One of the main reasons for writing documents in LaTeX is because it is really good at typesetting equations. Equations are written in 'math mode'.

4.4 Inserting Equations

You can enter math mode with an opening and closing dollar sign `$`. This can be used to write mathematical symbols within a sentence — for example, typing `$1+2=3$` produces $1 + 2 = 3$.

If you want a "displayed" equation on its own line use `$$...$$`.

For example, `$$1+2=3$$` produces:

$$1 + 2 = 3$$

For a numbered displayed equation, use `\begin{equation}...\end{equation}`.

For example, `\begin{equation}1+2=3\end{equation}` produces:

$$1 + 2 = 3 \tag{1}$$

The number 6 refers to the chapter number, this will only appear if you are using a document class with chapters, such as `report`.

Use `\begin{eqnarray}...\end{eqnarray}` to write equation arrays for a series of equations/inequalities. For example —

```
\begin{eqnarray}
  a & = & b + c \\
    & = & y - z
\end{eqnarray}
```

Produces:

$$a \quad = \quad b + c \tag{2}$$
$$= \quad y - z \tag{3}$$

For unnumbered equations add the star symbol `*` after the `equation` or `eqnarray` command (i.e. use `{equation*}` or `{eqnarray*}`).

4.5 Mathematical Symbols

Although some basic mathematical symbols $(+\; -\; =\; !\; /\; (\;)\; [\;]\; :)$ can be accessed directly from the keyboard, most must be inserted using a command.

This section is a very brief introduction to using LaTeX to produce mathematical symbols — the Mathematics chapter in the LaTeX Wikibook is an excellent tutorial on mathematical symbol commands, which you should refer to if you want to learn more. If you want to find the command for a specific symbol try Detexify[3], which can recognise hand drawn symbols.

4.5.1 Powers & Indices

Powers are inserted using the hat `^` symbol, e.g. `$n^2$` produces n^2.

Indices are inserted using an underscore `_`. For example, `$2_a$` produces 2_a.

If the power or index includes more than one character, group them using curly brackets `{...}`, e.g. `$b_{a-2}$` produces b_{a-2}.

4.5.2 Fractions

Fractions are inserted using `\frac{numerator}{denominator}`.

`$$\frac{a}{3}$$` produces: $\frac{a}{3}$

Fractions can be nested —

`$$\frac{y}{\frac{3}{x}+b}$$` produces:

$$\frac{y}{\frac{3}{x}+b}$$

4.5.3 Roots

Square root symbols are inserted using `\sqrt{...}` where ... is replaced by the square root content. If a magnitude is required it can be added using optional square brackets `[...]`.

`$$\sqrt{y^2}$$` produces:

$$\sqrt{y^2}$$

---

[3]`http://detexify.kirelabs.org`

`$$\sqrt[x]{y^2}$$` produces:

$$\sqrt[x]{y^2}$$

4.5.4 Sums, Limits & Integrals

The command `\sum` inserts a sum symbol; `\int` inserts an integral. For both functions, the upper limit is specified by a hat ˆ and the lower by an underscore _.

`$$\sum_{x=1}^5 y^z$$` produces:

$$\sum_{x=1}^5 y^z$$

`$$\lim_{x \to \infty} f(x)$$` produces:

$$\lim_{x \to \infty} f(x)$$

`$$\int_a^b f(x)$$` produces:

$$\int_a^b f(x)$$

4.5.5 Matrices

Furthermore it's possible to display matrices in LaTeX. There is a special matrix environment for this purpose, please keep in mind that the matrices only work within math environments as described below.

Add `\usepackage{amsmath}` in the preamble of your document (before the `\begin{document}` command).

```
\begin{equation*}
    \left[
    \begin{matrix}
        1 & 0\\
        0 & 1\\
    \end{matrix}
    \right]
\end{equation*}
```

Produces:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

4.5.6 Greek letters

Greek letters can be typed in math mode using the name of the letter preceded by a backslash \. Many Greek capital letters are used in the Latin alphabet — for those that are different capitalise the first letter of the name to produce a capital Greek letter.

For example —

`$\alpha$` $= \alpha$
`$\beta$` $= \beta$
`$\delta, \Delta$` $= \delta, \Delta$
`$\theta, \Theta$` $= \theta, \Theta$
`$\mu$` $= \mu$
`$\pi, \Pi$` $= \pi, \Pi$
`$\sigma, \Sigma$` $= \sigma, \Sigma$
`$\phi, \Phi$` $= \phi, \Phi$
`$\psi, \Psi$` $= \psi, \Psi$
`$\omega, \Omega$` $= \omega, \Omega$

Checkpoint 4.3

➲ Write code to produce the following equations:

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

$$\pi = \frac{c}{d} \tag{2}$$

$$\frac{d}{dx}e^x = e^x \tag{3}$$

$$\frac{d}{dx}\int_0^\infty f(s)ds = f(x) \tag{4}$$

$$f(x) = \sum_i = 0^\infty \frac{f^{(i)}(0)}{i!} x^i \tag{5}$$

$$x = \sqrt{\frac{x_i}{z}} y \tag{6}$$

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix} \tag{7}$$

Practical 6: References

6.1 Introduction

LaTeX includes features that allow you to easily cite references and create bibliographies in your document. This practical will explain how to do this using a separate BibTeX file to store the details of your references.

6.2 The BibTeX file

Your BibTeX file contains all the references you want to cite in your document. It has the file extension .bib. The .bib file is a plain text - it can be edited using Notepad or your LaTeX editor (e.g. TeXworks). You should enter each of your references in the BibTeX file in the following format:

```
@article{Birdetal2001,
    Author = {Bird, R. B. and Smith, E. A. and Bird, D. W.},
    Title = {The hunting handicap: costly signaling in human
    foraging strategies},
    Journal = {Behavioral Ecology and Sociobiology},
    Volume = {50},
    Pages = {9-19},
    Year = {2001} }
```

Each reference starts with the **reference type** (@article in the example above). Other reference types include @book, @incollection for a chapter in an edited book and @inproceedings for papers presented at conferences[4].

---

[4]See the Bibliography Management chapter in the LaTeX Wikibook `http://en.wikibooks.org/wiki/LaTeX/Bibliography_Management` for a full list of the reference types that BibTeX knows about, and their required and optional fields.

The reference type declaration is followed by a curly bracket, then the **citation key**. Each references citation key must be unique - you can use anything you want, but a system based on the first authors name and year (as in the example above) is probably easiest to keep track of.

The remaining lines contain the reference information in the format

`Field name = {field contents},`.

You need to include LaTeX commands in your BibTeX file for any special text formatting - e.g. italics (`\emph{Rattus norvegicus}`), quotation marks (`''...''`), ampersand (`\&`).

Surround any letters in a journal article title that need to be capitalised with curly brackets `{...}`. BibTeX automatically uncapitalises any capital letters within the journal article title. For example, "Dispersal in the contemporary United States" will be printed as "Dispersal in the contemporary united states", but "Dispersal in the contemporary {U}nited {S}tates" will be printed as "Dispersal in the contemporary United States".

You can type the BibTeX file yourself, or you can use reference management software such as JabRef or EndNote to create it[5].

6.3 Inserting the bibliography

Type the following where you want the bibliography to appear in your document (usually at the end):

```
\bibliographystyle{plain}
\bibliography{references}
```

Where references is the name of your .bib file.

6.4 Citing references

Type `\cite{citationkey}` where you want to cite a reference in your .tex document. If you dont want an in text citation, but still want the reference to appear in the bibliography, use `\nocite{citationkey}`.

---

[5]Instructions for using EndNote with LaTeXare available on the HowTo wiki `https://www.wiki.ed.ac.uk/x/sZpKBg`

To include a page number in your in-text citation put it in square brackets before the citation key: `\cite[p. 215]{citationkey}`.

To cite multiple references include all the citation keys within the curly brackets separated by commas: `\cite{citation01,citation02,citation03}`.

6.5 Styles

6.5.1 Numerical citations

LaTeX comes with several styles with numerical in-text citations, these include:

Plain The citation is a number in square brackets (e.g. [1]). The bibliography is ordered alphabetically by first author surname. All of the authors names are written in full.

Abbrv The same as plain except the authors first names are abbreviated to an initial.

Unsrt The same as plain except the references in the bibliography appear in the order that the citations appear in the document.

Alpha The same as plain except the citation is an alphanumeric abbreviation based on the author(s) surname(s) and year of publication, surrounded by square brackets (e.g. [Kop10]).

6.5.2 Author-date citations

Use the **natbib** package if you want to include author-date citations. Natbib uses the command `\citep{...}` for a citation in brackets (e.g. [Koppe, 2010]) and `\citet{...}` for a citation where only the year is in brackets (e.g. Koppe [2010]). There are lots of other ways that you can modify citations when using the natbib package - see the packages reference sheet for full details[6].

Natbib comes with three bibliography styles: **plainnat**, **abbrvnat** and **unsrtnat**. These format the bibliography in the same way as the **plain**, **abbrv** and **unsrt** styles, respectively.

---

[6]Reference sheet for natbib usage `http://mirror.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

6.5.2 Other bibliography styles

If you want to use a different style (e.g. one provided by the journal you are submitting an article to) you should save the style file (.bst file) in the same folder as your .tex and .bib files. Include the name of the .bst file in the `\bibliographystyle{...}` command.

Checkpoint 6

➲ Create a new file in TeXworks (File menu > New).

➲ Find out the 10 article references from your review assignments in the correct format with the help of Google Scholar (see example at start of practical) and type in the file.

➲ Click the **Save** button, the **Save File** window will open.

➲ Give the file name as **references** and save it as a **BibTeX database** in the same folder as your .tex file.

➲ Switch to your .tex document and insert `\cite` commands to cite the 15 articles in your .tex file and don't forget to add `\bibliographystyle` and `\bibliography` commands in the relevant places. Use any style (plain/abbrv/unsrt/alpha) you want.

➲ Typeset your .tex file.

➲ Switch to your .bib file, choose **BibTeX** from the typeset menu and click the **Typeset** button.

➲ Switch to your .tex file and typeset it twice. The in-text citations and reference list should be inserted.

Further Reading

LaTeX Project
http://www.latex-project.org/
Official website - has links to documentation, information about installing LaTeX on your own computer, and information about where to look for help.

The Not So Short Introduction to LaTeX2e
http://ctan.tug.org/tex-archive/info/lshort/english/lshort.pdf
A good tutorial for beginners.

LaTeX Wikibook
http://en.wikibooks.org/wiki/LaTeX/
Comprehensive and clearly written, although still a work in progress. A downloadable PDF is also available.

Comparison of TeX Editors on Wikipedia
http://en.wikipedia.org/wiki/Comparison_of_TeX_editors
Information to help you to choose which LaTeX editor to install on your own computer.

TeX Live
http://www.tug.org/texlive/
"An easy way to get up and running with the TeX document production system". Available for Unix and Windows (links to MacTeX for MacOSX users). Includes the TeXworks editor.

Workbook Source Files
http://edin.ac/17EQPM1
Download the .tex file and other files needed to compile this workbook.