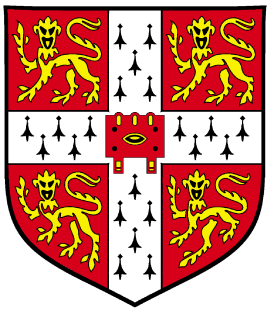
Towards the Ability to Make Super Duper LaTeX Documents



John William Roger Morgan

Department Of Chemistry

University of Cambridge

This dissertation is submitted for the degree of *Doctor of Philosophy*

November 14, 2019

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. It is not substantially the same as any that I have submitted for a degree or diploma or other qualification at any other University and no part has already been, or is concurrently being, submitted for any degree, diploma, or other qualification. It does not exceed 60,000 words, including tables, footnotes, bibliography and appendices.

Signature: _____

Date: _____

Acknowledgments

Course Design

This course and its materials were originally designed by Dr Chris Forman.

Family

Mum. Dad. Pet Dog.

Scientific Collaborations

Dr A. N. Other at uni of blah for samples and instrument time etc. My Amazing Lab Chums.

General Assistance

Cleaner, My amazing lab chums.

Financial Support

Bank, Tax payer. My amazing lab chums.

Moral Support

My Amazing Lab chums.

Academic Inspiration

Newton, My super. My Amazing Lab Chums.

Everyone else

My Amazing Lab Chums.

Summary

This document is a handbook for introducing you to LaTeX. It contains a bit of background about the history and philosophy of LaTeX. However, it concentrates on providing a guided practical introduction to the basic techniques such as

- Lists
- Mathematical and chemical equations
- Tables
- Figures
- Chapters and Sections
- Linking to other files in large documents
- Referencing

The handbook is also presented in the style of a thesis. This means that source files provided on this introductory course are a convenient template for a thesis.

Abbreviations

ADP	Adenosine Diphosphate
(nc/c)AFM	(non-contact/conducting) Atomic Force Microscopy
ATP	Adenosine Triphosphate

Contents

List of Tables

List of Figures

Chapter 1

Prelude

1.1 What is LaTeX and how does it work?

LaTeX is a typesetting program. It generates a professionally looking and easily printable document (e.g. a `pdf` file) from user provided input files. The main input is a `tex` file, which contains the source code — raw text interspersed with commands. The commands basically tell LaTeX what to do with the text. For example, you can make text **bold**, *italic*, or underlined by first typing \textbf{}, \emph{} or \underlined{}, respectively, and then placing the text inside the curly brackets.

The typesetting rules applied by LaTeX are targeted mainly at aesthetics, and these rules have been honed by professional typesetters since the time of Caxton!

1.2 Why bother with it?

LaTeX is not the "magic bullet" of typesetting — it has pros and cons — but there are good reasons why LaTeX is widely used in scientific publishing and academia.

1.2.1 Advantages

- It's free of charge.
- We can use any text editor to view and modify the source code.
- More time is spent on creating content and less time on cosmetic tweaking.
- Basic typesetting is done automatically. LaTeX uses consistent rules throughout a document, making it look professional.
- Precise changes can be introduced globally with very little effort.

- Mathematical equations, like $E = mc^2$ or $\imath\hbar\frac{\partial}{\partial t}\Phi(x,t) = \hat{H}\Phi(x,t)$ can be produced almost as fast as typing (if you know the commands!).

1.2.2 Disadvantages

- Steep learning curve.
- You don't see the output as you go.
- Sometimes LaTeX does not work quite the way you want it to, and learning how to influence it can be a challenge.
- Errors in the source code can be surprisingly difficult to locate.

1.2.3 LaTeX versus MS Word/Office

- Large technical documents are often much easier to handle in LaTeX.
- In LaTeX, technical features such as tables, equations and figures are integrated much more smoothly.
- It is generally quicker to write and debug the LaTeX source than it is to typeset an entire thesis manually and consistently in Word/Office.

1.3 LaTeX distributions and editors

LaTeX comes in multiple flavours/distributions, with the most common ones being:

- TeXLive for Windows and Unix/Linux.
- MacTeX for Mac OS.

You also need to choose an editor (sometimes referred to as "the front end") for writing the source code. I will be using gedit, a basic text editor, during this workshop. Several more advanced front ends are available on MCS Linux. All of them provide an editing environment, but use calls to the system's TeXLive distribution.

Chapter 2

My First Page

2.1 The Very Beginning

The very simplest LaTeX document might look like this:

```
\documentclass[a4paper,12pt]{article}
\begin{document}
Hello World.
\end{document}
```

Hello World.

2.1.1 Break Down

What's all the gobbledegook around my simple message??? In LaTeX we intersperse text and commands. Commands are preceded by a backslash (i.e. "\"). For example, the first line in a LaTeX document ***must*** be:

```
\documentclass[options]{class}
```

where the word "class" may be substituted for one of many things, such as: article, proc, minimal, report, book, letter, memoir, slides, beamer.

Similarly, and completely generally in LaTeX speak, the square brackets denote the existence of optional parameters. Each individual command can take its own parameters and for the "\documentclass" command there are options for controlling font size, font family, landscape, oneside, twosided, page size and so on. These options will persist throughout the entire document. For example, the document class command for this document (which is likely to be like the one you would use for a thesis) would be:

```
\documentclass[12pt, oneside, a4paper]{book}
```

Other options include:

- 10pt, 11pt, 12pt (default is 10pt).
- letterpaper, legalpaper, a4paper, executivepaper, a5paper, b5paper

and so on.

2.2 Can I start typing please?

After we have set up our document we can start actual work on it. So we must tell LaTeX that what follows is to be interpreted as a document. The command "\begin" can take many different parameters, and it tells LaTeX to enter what we call 'an environment'. Thus the commands

```
\begin{document}
```

```
\end{document}
```

tell LaTeX to enter and leave the 'document' environment, and thus constitute the outer limits of our document file. Other environments include the 'equation' environment, the 'itemize' environment, the 'figure' environment, and so on. These will be encountered in due course.

Every part of a `tex` file is therefore inside an environment of specific type, and the content within each environment consists of commands or text. That's basically it. So let's get on with it shall we...

2.3 The ground rules

> Hold on, hold on, hold on my son.
>
> First the lessons.
>
> Then the fun!

Dr Seuss.

2.3.1 Spaces

Whitespace characters, such as blank or tab, are treated uniformly as space by LaTeX. Several consecutive whitespace characters are treated as one single space. Whitespace at the start of a line is generally ignored, and a single line break is treated as whitespace. An empty line between two lines of text defines the end of a paragraph. Several empty lines are treated the same as one empty line. The text below is an example.

```
It does not matter whether you
enter one or several             spaces
after a word. The final result will be typeset beautifully
according    to     precise rules.

An empty line starts a new
paragraph. Note the indentation on the first line of the paragraph,
except at the beginning of the section.
```

It does not matter whether you enter one or several spaces after a word. The final result will be typeset beautifully according to specific rules.

An empty line starts a new paragraph. Note the indentation on the first line of the paragraph, except at the beginning of the section.

2.3.2 Special Characters

The symbols # $ % ^ & _ { } ~ \ are reserved characters that either have a special meaning under LaTeX or are unavailable in all the fonts. If you enter them directly in your text, they will normally not print, but rather make LaTeX do strange things.

To override the special meanings of these symbols and allow them to appear within your text you may use the following sequences:

```
\# \$ \% \textasciicircum{} \& \_ \{ \} \~{} \textbackslash
```

Other symbols and many more can be printed with special commands in mathematical formulae or as accents.

The backslash character '\' cannot be entered by adding another backslash in front of it (\\), because this sequence means "linebreak".

The command \~{} produces a tilde which is placed over the next letter. For example \~{n} gives ñ. To produce just the character ~, use \~{} which places a ~ over an empty box.

Similarly, the command \^ produces a hat over the next character, for example \^{o} produces ô.

2.4 Preamble

Between the \documentclass and \begin{document} commands we have what's called the "preamble". Here we can load special features that we will use throughout our document to expand on the basic capabilities of LaTeX. These are contained in modules called "packages", which we can tell our LaTeX compiler to load by using the command \include{packageName}.

There are many repositories of such packages on the web. Your compiler generally knows where to look and most standard packages are included with any install. For example, to include a special set of typesetting instructions to handle chemical equations we can grow our simple document with a single command like this:

```
\documentclass[a4paper,12pt]{article}
\usepackage[version=3]{mhchem}
\begin{document}
Hello World!

We all need \ce{H2O}.

I'm less fussed about \ce{^{235}_{92}U+}.

\end{document}
```

Hello World!
We all need H_2O.
I'm less fussed about $^{235}_{92}U^{+}$.

Here we have loaded a package called: "mhchem" which took the option "[version=3]". This is a package for drawing chemical equations easily and it has its own instruction manual which you can follow easily. It is included in the bundle of files for this course.

Other things we can do in the preamble within LaTeX are to redefine existing commands or create our own personal commands. These can be stored in a file called the 'style file' which we can load at the beginning of our document, in place of our documentclass. More about this later on...

2.4.1 Comments

It is often useful to comment your LaTeX documents. You can leave yourself amusing, sarcastic messages that won't get printed out in the final document. Use this to make your own notes as the course progresses.

To get a comment use the % command, which tells LaTeX to ignore the rest of the line, the line break and all the white space at the beginning of the next line, for example, we may add to our continually evolving document...

```
\documentclass[a4paper,12pt]{article}
\usepackage[version=3]{mhchem}
\begin{document}
Hello World!

%All humans need water and I would like
%to include this concept in my arguments.
We all need \ce{H2O}.

%Uranium 235 is toxic, which is why I don't want to consume it...
I'm less fussed about \ce{^{235}_{92}U+}.

\end{document}
```

Hello World!
We all need H_2O.
I'm less fussed about $^{235}_{92}U^{+}$.

2.5 Compiling

Once the document is finished you can try compiling it, but expect that there will often be something wrong with your file. Any errors will be highlighted in the console window or error output box of your front end. Some front ends dump the running commentary which LaTeX produces into a text file for easy reading afterwards.

In a compile attempt (successful or otherwise) LaTeX may produce the following files:

projectname.aux
projectname.bbl
projectname.lof
projectname.lot
projectname.txt
projectname.toc
projectname.dvi

These are interim files (toc= table of contents, bbl = bibliography, etc.) generated by the LaTeX compiler, and some of the content is still a mystery to me. Some versions of LaTeX only produce DVI files, and you then need to convert these to a PDF or download a DVI viewer. You can also convert DVI files to PS files and then view them. The good thing about front ends is that you can set them up to produce a specific output in a convenient manner.

OK. So we have now produced a silly document that we can understand. Now let's get on with learning how to make our own documents.

Chapter 3

Lists

Lists are great. The command \begin{} can be used to enter a list environment. For example:

```
\begin{itemize}
\item cat
\item dog
\item horse
\end{itemize}
```

Produces:

- cat
- dog
- horse

We can also replace the bullet points with numbers using the enumerate keyword.

```
\begin{enumerate}
\item cat
\item dog
\item horse
\end{enumerate}
```

1. cat
2. dog
3. horse

and we can use the description keyword which does this:

```
\begin{description}
\item[Cat] a lovely furry creature with a cute nose and whiskers.
\item[Dog] Another furry creature that smells rather well;
           its olfactory power stems from its nasal dampness.
\item [Horse] A large stinky creature with sideways facing eyes.
\end{description}
```

Cat a lovely furry creature with a cute nose and whiskers.

Dog Another furry creature that smells rather well; its olfactory power stems from its nasal dampness.

Horse A large stinky creature with sideways facing eyes.

Chapter 4

Maths Equations

4.1 Producing Beautiful Looking Mathematics

One of the best features of LaTeX is "maths mode". For example, the Schrödinger equation can be produced as follows.

```
\begin{math}
\imath\hbar\frac{\partial}{\partial t}\Phi\left( x,t \right) =
\hat{H}\Phi \left( x,t \right)
\end{math}
```

$$\imath\hbar\frac{\partial}{\partial t}\Phi\left(x,t \right) = \hat{H}\Phi \left(x,t \right)$$

There are a number of ways to switch on the maths mode. The first, as above, is to enter the mathematical environment with the \begin{math} command. You can also enter the maths mode in-line using the $ symbol. For example typing in $y=ax^2+bx+c$ yields $y = ax^2 + bx + c$. A third option is to enter the 'equation' environment, which enables you to number equations so you can then refer to them later in the text.

```
\begin{equation}
y(\left(t\right)= \sin \left(\frac{{\alpha}t}{2\pi} + \phi_0\right)
\end{equation}
```

$$y\left(t\right)= \sin \left(\frac{{\alpha}t}{2\pi} + \phi_0\right) \tag{4.1}$$

4.2 Basic Maths Mode

Once in maths mode, there is a text based code for writing down your equations. Here are the most basic symbols to get you going.

Final Result	LaTeX Code
$a + b$	a+b
$a - b$	a-b
ab	ab
$a * b$	a*b
$a \times b$	a \times b
$a \cdot b$	a \cdot b
$\frac{a}{b}$	\frac{a}{b}
a^b	a^b
a_b	a_b
$\sin a$	\sin a (same for cos, tan)
$sina$	sin a
$\sqrt{a}$	\sqrt{a}
(a)	\left(a \right)
$[a]$	\left[a \right]
α	\alpha
π	\pi

A full treatise on maths mode is not practical here. There are lots of online tutorials and summaries of symbols. It just takes a bit of practice and you can build up equations really easily. It's straight forward to learn new stuff once you've done it a few times.

4.3 Equation Arrays

Sometimes you need to arrange several equations vertically, referencing individual lines separately and aligning the equations on the = sign. This can be achieved with equation arrays as follows:

$$\begin{aligned} A\left(x\right) &= \frac{x^2+2x+1}{1+x} \\ &= \frac{\left(x+1\right)\left(x+1\right)}{1+x} \\ &= x+1 \end{aligned} \tag{4.2}$$

$$B(x,t) = \frac{e^{(\imath\omega_0 t+kx)}}{4\pi\epsilon_0} \tag{4.3}$$

```
\begin{eqnarray}
A\left( x\right) & = & \frac{x^2+2x+1}{1+x} \\
& = & \frac{\left(x+1\right)\left(x+1\right)}{1+x} \nonumber\\
& = & x+1 \nonumber\\
B(x,t) & = & \frac{e^{\left(\imath\omega_0 t + kx\right)}}{4\pi\epsilon_0}
\end{eqnarray}
```

- Note the & symbols. This tells LaTeX where to align the equations. There must be the same number of & symbols in each line.
- Note the \\ at the end of each line except the last one. This symbol tells LaTeX to add another row in the array. If you put it on the last line you get an empty row at the bottom of the array.
- Note the \nonumber command which suppresses line numbering for that line.
- Note that equation number carries on from equation 4.1 in the previous section.

4.4 Maths Packages

Maths mode comes as standard in LaTeX. However, you can download packages that buff up your maths symbol set. For example, the usual symbol for the set of real numbers ($\mathbb{R}$) is typeset using \mathbb{R} in maths mode, and it requires

```
\usepackage{amssymb}
```

Chapter 5

Chemical Equations

The mhchem package means you can do basic stuff very easily using \ce{}. For example:

```
\ce{CO2 + C -> 2CO}
\ce{CO2 + C <- 2CO}
\ce{CO2 + C <=> 2CO}
\ce{A-B=C#D\sbond E\dbond F\tbond G}
```

$$\mathrm{CO_2 + C \longrightarrow 2\,CO}$$
$$\mathrm{CO_2 + C \longleftarrow 2\,CO}$$
$$\mathrm{CO_2 + C \rightleftharpoons 2\,CO}$$
$$\mathrm{A{-}B{=}C{\equiv}D{-}E{=}F{\equiv}G}$$

You can also use math mode within chemical equations.

$$x\,\mathrm{Na(NH_4)HPO_4} \xrightarrow{\Delta} (\mathrm{NaPO_3})_x + x\,\mathrm{NH_3}\uparrow + x\,\mathrm{H_2O}$$

```
\ce{$x\,$ Na(NH4)HPO4 ->[\Delta] (NaPO3)_{$x$} + $x\,$ NH3 ^ + $x\,$ H2O}
```

And you can number chemical reactions as well by using the math mode equation environment.

$$\mathrm{CO_2 + C \rightleftharpoons 2\,CO} \tag{5.1}$$

```
\begin{equation}
\ce{CO2 + C <=> 2CO}
\end{equation}
```

However, this will number mathematical and chemical equations using the same number system. There is a discussion in the mhchem pdf which shows you how to number chemical and mathematical equations independently. It's a bit complex and involves you creating your own type of environment... bit beyond the intro level of this course...

Chapter 6

Tables

Tables are big business in LaTeX. I use these packages to help me.

```
\usepackage{multirow}
\usepackage{booktabs}
\usepackage{dcolumn}
```

Here's an example table

Fibre Type	N	M	L_t (nm)	R_t (nm)	d_t (nm)	ΔZ_t (nm)	
SS Twisted	5	4	313 (122)	5.7 (1.6)	5.4 (1.0)	150.5 (63.1)	
			L_s (nm)	R_s (nm)	W_s (nm)	ΔZ_s (nm)	ΔZ_e (nm)
SS Spiral	32	7	123 (31.6)	11.4 (3.3)	11.1 (2.6)	32.9 (22.2)	26·9
SSB Spiral	64	18	106 (22.9)	10.8 (2.7)	9.8 (1.8)	28.2 (8.1)	22·7
BSS Spiral	37	19	110 (44.9)	13.4 (3.0)	12.9 (1.8)	27.1 (9.1)	21·5
SSSB Spiral	26	6	104 (23.9)	13.3 (2.6)	16.4 (3.4)	29.9 (11.1)	22·5

Table 6.1: A funky table from Chirs Forman's thesis.

Here's what the code looks like (just for illustration!):

```
\begin{table}[!hb]
 \centering
 \begin{tabular}{@{}rccc@{~}r@{.}l*3{r@{.}l@{~}r@{.}l}D{.}{\cdot}{2,1}}
\toprule
\multicolumn{1}{c}{Fibre Type}
& N
& M
&\multicolumn{3}{c}{$L_t$ (nm)}
&\multicolumn{4}{c}{$R_t$ (nm)}
&\multicolumn{4}{c}{$d_t$ (nm)}
&\multicolumn{4}{c}{${\Delta}Z_t$ (nm)}
&\multicolumn{1}{c}{~}\\
\cmidrule(lr){1-1}
\cmidrule(lr){2-2}
\cmidrule(lr){3-3}
\cmidrule(lr){4-6}
\cmidrule(lr){7-10}
\cmidrule(lr){11-14}
\cmidrule(lr){15-18}
  SS Twisted & 5 & 4 & 313 &\multicolumn{2}{c}{(122)} & 5&7 &(1&6)  & 5&4 &(1&0)  &
  \multicolumn{19}{c}{~}\\
  %\cmidrule(lr){3-18}
  &
  &
  &\multicolumn{3}{c}{$L_s$ (nm)}
  &\multicolumn{4}{c}{$R_s$ (nm)}
  &\multicolumn{4}{c}{$W_s$ (nm)}
  &\multicolumn{4}{c}{${\Delta}Z_s$ (nm)}
  &\multicolumn{1}{c}{${\Delta}Z_{e}$ (nm)}\\
\cmidrule(lr){4-6}
\cmidrule(lr){7-10}
\cmidrule(lr){11-14}
\cmidrule(lr){15-18}
\cmidrule(lr){19-19}
  SS Spiral& 32 & 7 &123&(31&6)&11&4&(3&3)&11&1&(2&6)&32&9&(22&2)&26.9\\
  SSB Spiral& 64&18 &106&(22&9)&10&8&(2&7)&9&8&(1&8)&28&2&(8&1)&22.7\\
```

```
%\midrule
  BSS Spiral&37 &19 &110&(44&9)&13&4&(3&0)&12&9&(1&8)&27&1&(9&1)&21.5\\
%\midrule
  SSSB Spiral&26& 6 &104&(23&9)&13&3&(2&6)&16&4&(3&4)&29&9&(11&1)&22.5\\
\bottomrule
\end{tabular}
\caption[Basic Fibre Dimensions by TEM]{A funky table from my thesis.}
\label{tab:BasicXSBFibreDimensionsTEM}
\end{table}
```

6.1 Tables Made Easy

Here is a simple table followed by the code that produced it.

anchovy	banana	carrot
dog	apple	fennel
goat	strawberry	potato

```
\begin{tabular}{lcr}
anchovy & banana & carrot \\
dog & apple & fennel \\
goat & strawberry & potato
\end{tabular}
```

The tabular environment is a special case of the "array" environment for distributing content uniformly across a region of the page. This ability has already been exploited to align mathematical equations. For now though look at the first line.

```
\begin{tabular}{lcr}
```

This command tells LaTeX to enter the tabular environment. The letters l, c and r in the curly braces tell LaTeX to create a table with three columns in which the first column is left justified, the second column is centered and the third column is right justified. Lets add a fourth column and this time center justify all the columns.

anchovy	banana	carrot	Johnny
dog	apple	fennel	Pete
goat	strawberry	potato	

```
\begin{tabular}{cccc}
anchovy & banana & carrot & Johnny\\
dog & apple & fennel & Pete\\
goat & strawberry & potato &
\end{tabular}
```

Each row in the table is a list of items separated by the & symbol. The end of each row is denoted by \\. The last row in the table doesn't have a \\. You do not have to have data between the ampersands but you must have the right number of ampersands to match the number of columns that LaTeX is expecting.

6.1.1 Adding Borders To Tables

Tables should never have vertical lines. No professionally typeset table contains vertical lines. Do not put vertical lines in your tables. That said, it is easy to do.

anchovy	banana	carrot	Johnny
dog	apple	fennel	Pete
goat	strawberry	potato	

```
\begin{tabular}{|c|c|c|c|}
anchovy & banana & carrot & Johnny\\
dog & apple & fennel & Pete\\
goat & strawberry & potato &
\end{tabular}
```

Tables should have neatly headed columns with the heading for each field separated from the data by horizontal lines. The \toprule, \cmidrule and \bottomrule commands from the booktabs package are useful for controlling horizontal lines.

Ingredient 1	Ingredient 2	Ingredient 3	Source
anchovy	banana	carrot	Johnny
dog	apple	fennel	Pete
goat	strawberry	potato	

```
\begin{tabular}{cccc}
\toprule
Ingredient 1 & Ingredient 2 & Ingredient 3 & Source \\
\cmidrule(){1-4}
anchovy & banana & carrot & Johnny\\
dog & apple & fennel & Pete\\
goat & strawberry & potato & \\
\bottomrule
\end{tabular}
```

Note that when using the \bottomrule command you must add the \\ symbol to the last line of data. The last line of the table is now buried within the \bottomrule command.

6.1.2 The \cmidrule Command

This useful and versatile command takes a bunch of options to control subtleties, like only putting lines across some of the columns, or not quite making them cross the full width of the column. The (lr) option trims the left and right ends of the lines off. For example:

Recipe Version	Ingredient 1	Ingredient 2	Ingredient 3	Source
10.1	anchovy	banana	carrot	Johnny
1.34	dog	apple	fennel	Pete
709.23	goat	strawberry	potato	

```
\begin{tabular}{ccccc}
\toprule
Recipe Version & Ingredient 1 & Ingredient 2 & Ingredient 3 & Source \\
\cmidrule(lr){1-1}
\cmidrule(l){2-2}
\cmidrule(){3-3}
\cmidrule(r){4-4}
\cmidrule(lr){5-5}
10.1 & anchovy & banana & carrot & Johnny\\
1.34 & dog & apple & fennel & Pete\\
709.23 & goat & strawberry & potato & \\
\bottomrule
\end{tabular}
```

6.1.3 Aligning Decimal Points

Note that the decimal points don't line up in the new column "Recipe Version" in the previous section. There is a way round this with the DColumn package.

RecipeVersion	Ingredient 1	Ingredient 2	Ingredient 3	Source
10·1	anchovy	banana	carrot	Johnny
1·34	dog	apple	fennel	Pete
709·23	goat	strawberry	potato	

```
\begin{tabular}{D{.}{\cdot}{4,4}cccc}
\toprule
Recipe Version & Ingredient 1 & Ingredient 2 & Ingredient 3 & Source \\
\cmidrule(lr){1-2}
\cmidrule(lr){3-3}
\cmidrule(lr){4-4}
\cmidrule(lr){5-5}
\cmidrule(lr){6-6}
10.1 & anchovy & banana & carrot & Johnny\\
1.34 & dog & apple & fennel & Pete\\
709.23 & goat & strawberry & potato & \\
\bottomrule
\end{tabular}
```

The DColumn package defines a new type of column, which can be invoked by placing a capital D in the \tabular command defining the table. If defined using a D, then the column is placed in maths mode. D takes four parameters: D{a}{b}{c,d} where a is the symbol which is to be aligned, b is the symbol with which to replace the aligning character, and c,d must be integers which indicate LaTeX should have up to c white space characters before the aligning character and d afterwards, thereby defining the position of the number within the column. The command \cdot prints a special type of maths mode symbol, which is a dot that is vertically shifted and larger than a normal period, e.g.: ·

6.1.4 Final Table Trick

DColumn forces the column to be in maths mode which is why the heading "Recipe Version" was typeset in italics in the previous section. Indeed LaTeX tried to interpret the heading of the table as a number to be aligned. We can over ride this behaviour by using the \multicolumn command to locally impose a different type of justification and temporarily disable the maths mode as follows:

Recipe Version	Ingredient 1	Ingredient 2	Ingredient 3	Source
10·1	anchovy	banana	carrot	Johnny
1·34	dog	apple	fennel	Pete
709·23	goat	strawberry	potato	

```
\begin{tabular}{D{.}{\cdot}{4,4}cccc}
\toprule
\multicolumn{1}{c}{Recipe Version} & Ingredient 1 & Ingredient 2 & Ingredient 3 & S
\cmidrule(lr){1-1}
\cmidrule(lr){2-2}
\cmidrule(lr){3-3}
\cmidrule(lr){4-4}
\cmidrule(lr){5-5}
10.1 & anchovy & banana & carrot & Johnny\\
1.34 & dog & apple & fennel & Pete\\
709.23 & goat & strawberry & potato & \\
\bottomrule
\end{tabular}
```

This introduces you to the multicolumn command which you can use to span text over several columns. The format is \multicolumn{n}{j}{text} where n is the number of columns to span. j is the justification: either l,c or r; and text is the text to add. There is also a similar command \multirow.

6.2 Numbering Tables

To tell LaTeX to assign a number to a table and add it to the list of tables you must use the \begin{table} command to tell LaTeX to create a table environment, e.g.:

Recipe Version	Ingredient 1	Ingredient 2	Ingredient 3	Source
10·1	anchovy	banana	carrot	Johnny
1·34	dog	apple	fennel	Pete
709·23	goat	strawberry	potato	

Table 6.2: Recipes that ought to be banned.

```
\begin{table}[!h]
\centering
\begin{tabular}{D{.}{\cdot}{4,4}cccc}
\toprule
\multicolumn{1}{c}{Recipe Version}& Ingredient 1 & Ingredient 2 & Ingredient 3 & So
\cmidrule(lr){1-1}
\cmidrule(lr){2-2}
\cmidrule(lr){3-3}
\cmidrule(lr){4-4}
\cmidrule(lr){5-5}
10.1 & anchovy & banana & carrot & Johnny\\
1.34 & dog & apple & fennel & Pete\\
709.23 & goat & strawberry & potato & \\
\bottomrule
\end{tabular}
\caption[Table of Banned Recipes]{Recipes that ought to be banned.}
\label{tab:Recipes}
\end{table}
```

In the table environment the \caption[text1]{text2} command adds a caption, where text1 appears in the list of tables at the beginning of the document and text2 is the local caption. The label command creates a label with which to reference the table e.g. Table 6.2 is a table of recipes that have been made up to illustrate how to use tables in LaTeX. We also use the \centering command to center the table and caption within the table environment. We could also use the \begin{center} and \end{center} commands. Finally, the [!h] option in the \begin{table}[!h] command explicitly tells LaTeX where to put the table, overriding a default choice.

Chapter 7

Adding Figures To Your Document

7.1 My First Figure

Adding figures is easy in LaTeX. You just create a figure environment which is much the same as the table environment. For example, put

```
\usepackage{graphicx}
```

in the preamble, then put

```
\begin{figure}[!ht]
\centering
\includegraphics[width=137.4mm]{Images/haemStructure.png}
\caption[Haem Structure]{A fancy image.}
\label{fig:haemStructure}
\end{figure}
```

in the main document, and see next page for the result. The command \includegraphics tells LaTeX to look in the directory "Images" and incorporate the file called "haemStructure.png" into the final document, while setting the image width to 137.4mm. This command can be used to resize a graphical file using the height or width parameter, whose value can be specified in terms of specific units (e.g. mm, cm, pt, ex, em, and so on) or expressed as a fraction of another length parameter value (e.g. width=0.5\textwidth). LaTeX recognises a very wide variety of standard units and graphics formats. The command is part of the graphicx package, and so in the preamble you must include the command \usepackage{graphicx}.

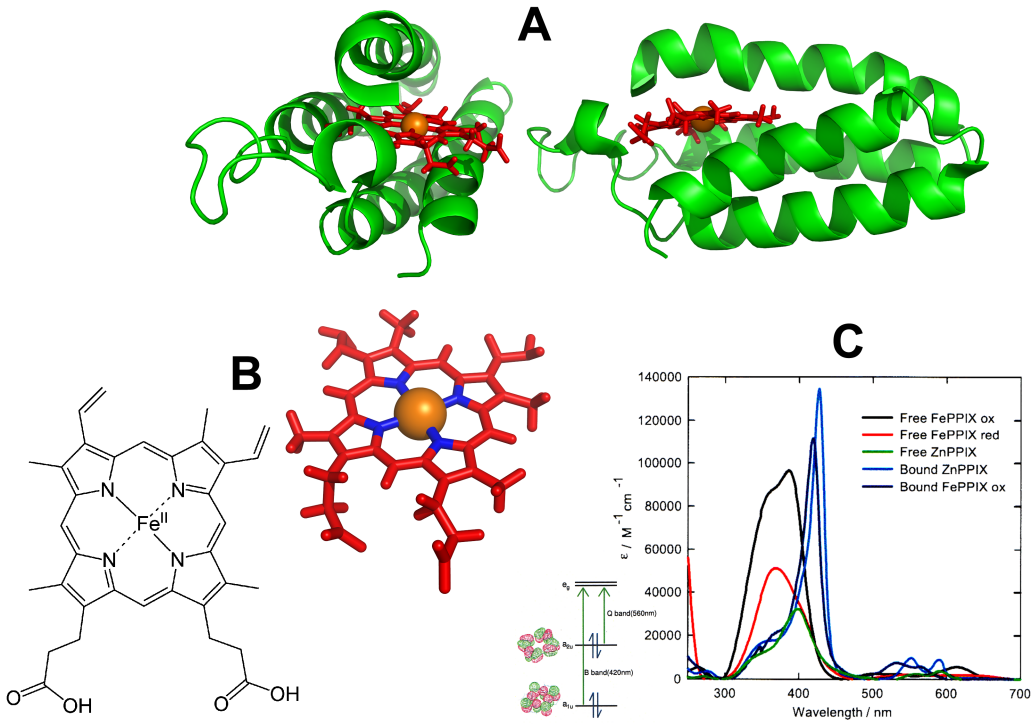


Figure 7.1: A fancy image from Chris Forman's Thesis.

Many packages have been written to provide new commands. For example the 'subfigure' package enables you to place separate images or files in the same figure side by side while giving them their own label. However, in this document we only concentrate on the very basics.

7.2 Floating Environments

Both tables and figures are examples of a "floating environments", which means LaTeX decides where to put them.

The mysterious [!ht] letters give guidance to LaTeX about roughly where the figure or table is allowed to be, but in general they can move a long way from where you position them within your text file. The letters in the square bracket can be t, h or b which stand for top, here or bottom. If you don't specify any letters LaTeX defaults to [t].

LaTeX will always preserve the order in which figures appear. If it cannot find a type setting solution, then it may move the float to its own page, and combine it with other figures. If it still can't fit that in for some reason, then it moves the float to the end of the document. This has the effect of pushing all the remaining floating environments to the end of the document also.

The exclamation mark instructs LaTeX to "try harder" at putting the float where you told it to. Often you must play around to ensure the float positioning is acceptable, but usually this can be achieved by stretching or shrinking the image slightly using the size parameter, re-ordering text, judicial applications of the \pagebreak command, or by shouting loudly and slapping the computer monitor about. Note that it is better to resize the image using the original program that was used to create the image. Also, using vector graphics (e.g. eps files) can be useful, because they are designed to scale better.

Chapter 8

The Chapter on Making Chapters

LaTeX forces you to structure your document. There is a series of simple commands for achieving this:

\chapter{chapterName}
\section{sectionName}
\subsection{subsectionName}
\subsubsection{subsubsectionName}

Easy enough. At the start of each chapter or section, just issue one of these commands to name the section and LaTeX will present the chapter heading in the right style of font, which will be the same style of all the other heading names at that level of hierarchy throughout the document. In addition, LaTeX assigns a number to that section. For example, this chapter was created using the following command.

```
\chapter{The Chapter on Making Chapters}
```

LaTeX assigns the correct chapter number to each chapter in turn and then puts it in the table of contents, as you can see by looking at the table of contents! Easy.

8.1 The Section About Sections

I think you're getting the hang of this. This section was created using the command \section{The Section About Sections}. It appears in the table of contents as section 8.1.

8.1.1 The Sub-Section About Sub-Sections

Now you're really getting the hang of this. This subsection was created using the command \subsection{The Sub-Section About Sub-Sections}.

The Sub-Sub-Section About Sub-Sub-Sections

Now you're really getting the hang of this. But.. caught you out! This sub-subsection doesn't have a number! Ha Ha! It was created using the command \subsubsection{The Sub-Sub-Section About Sub-Sub-Sections} in the same vein as all the other subsections/chapters etc. Here we have stopped the depth of the section numbering at the second level with the command

```
\setcounter{secnumdepth}{2}
```

in the preamble. Easy.

The Section About Sections That Don't Appear in the Table of Contents

Ok. Now we're throwing a spanner in the works. This section was created using the following command.

```
\section*{The Section About Sections That Don't Appear in the Table of Contents}
```

Notice how LaTeX has not assigned a number to the section and it doesn't appear in the table of contents. The effect of the * is to suppress the inclusion of a chapter, section or subsection in the automatic numbering. Easy. You can do this at any level. This is useful for things like prefaces, tables of contents or acknowledgements, for which you may or may not wish to have an entry in the main contents table. Up to you. It's your thesis. Don't just copy me.

8.2 A Small Point About Numbering

In this section notice how the numbering starts from where it left off before we suppressed the numbering on the previous section. Easy.

8.3 The Section About Internal Referencing

There are two related commands: \label{labelName} and \ref{labelName}. The \label{} command allows you to create a label in a particular environment.[1] The label name can be anything you like, as long as it doesn't contain white space or special characters. The label won't appear in the final document, it just makes it easy to refer back to any particular environment elsewhere in the document. The \ref{} command enables you to insert a reference anywhere in the document to any label in the document. For example, this is section 8.3; and no, I did not just manually type "8.3". Rather, I used the two commands: first \label{sec:InternalReferencing} at the very start of the section, and then \ref{sec:InternalReferencing} to insert the reference 8.3.

The astute among you will realise that LaTeX has to read the document several times. Once to find the labels and then again to populate the references with the correct numbers. So you have to compile a LaTeX document twice to get the referencing right. If there is a missing label or you refer to something that doesn't exist, then LaTeX inserts a convenient ? at that point. So hunting for queries is a useful way of finding broken references. LaTeX issues warnings when it finds broken references.

As you create a document you will find yourself putting labels in all over the place, so choose a sensible naming convention to help you remember the label names.

Each type of environment (equations, figures, tables, sections, etc.) has its own independent numbering system. So when you choose your label name it's a good idea to have an identifier for that type of environment. I have my own convention for label names, which I use to help me remember references, e.g. sec:SectionAboutCats, eqn:EquationAboutCats, fig:FigureAboutCats, and so on. This means you can differentiate between referring to the section or the figure more easily, even though they are about the same thing.

So internal referencing is a doddle. Easy!

[1] That's right. Chapters, sections, sub-sections, etc. are all environments!

Chapter 9

External Referencing

Referencing is usually performed using a separate program called BibTEX. This program understands the file format .bib.

To create a reference within your document first you must create a .bib file by exporting your list of references from which ever referencing software you used, such as endnote, mendeley or whatever you use.

You can refer to an entry in the .bib file using the \cite{ref:Name} command. the identifier 'ref:Name' is the unique identifier which is the first line of each item included in your bibliography file.

For example, open the file LaTeXCourseBib.bib in the bibliography directory. The first entry has the identifier 'Horcas2007'. This can be invoked as follows:

```
For example, this interesting fact\cite{Horcas2007}, is a cracking example.
```

For example, this interesting fact[1], is a cracking example.

During compilation LATEX and BibTeX co-operate. During the first compilation LATEXgenerates a list of references that it needs. During the second compilation BibTEX populates the details from the .bib file into a shorter, ordered .bib file. The third compilation inserts markers at the right place in the main file and the fourth compilation generates the final list of references. Thus the compilation sequence is LATEX, BibTEX, LATEX, LATEX.

The style of the referencing format can be changed using the command \bibliographystyle{}. e.g. the style used in this document is cjfthesisv1.bst which is invoked by

```
\bibliographystyle{Bibliography/cjfthesisv1}
```

The location of the list of references in the document is specified by issuing the \bibliography{} command which also specifies the master bibliography file.

```
\bibliography{Bibliography/LaTeXCourseBib}
```

Chapter 10

Organising Your Files

One of the drawbacks of MS Word is that the figures and tables are included within the document which becomes very very clunky quite quickly and once it reaches around 100Mb it becomes unwieldy and hard to export. One of that advantages of LaTeX is that the files are all text files and so are very light on resources and are easy to email around the place.

However, sometimes it can be difficult to navigate through a single text file, so if you had a huge document then it becomes difficult to find your way around. One way of circumnavigating this problem is to break the file up into smaller files. As long as LaTeX knows where to look you can have a file for each logical sub-division of your document.

Therefore it is worth spending a bit of time at the beginning of a large project deciding how you wish to break up your LaTeX files and organise them neatly and tidily. You can then tell LaTeX where to look for the files using the following command:

```
\include{fileName}
```

10.1 Linking a File

We are now going to copy part of the tutorial file into a separate files and link it the main file. Because this could go wrong it is sensible to back up tutorial.tex as something else. So save a copy called 'tutorialOriginal.tex' for safe keeping. Then follow the following instructions.

1. Create a new file called MyFirstPage.tex
2. Select and Cut out the first chapter of Tutorial.Tex
3. Paste the first chapter into MyFirstPage.tex
4. Save both tutorial.tex and MyFirstPage.tex
5. Then add \include{MyFirstPage} in tutorial.tex where the text used to be before you cut it out. We don't need to add the '.tex' because LaTeX can only include '.tex' files.
6. compile tutorial.tex and view the PDF

The final PDF should look exactly the same as it was! except we now have a separate file for one of the chapters.

10.2 Creating A Class File

We can also save our preamble in a separate file called a class file. This makes your main document much easier to understand. To make it work we need to use the \LoadClass{} command as follows:

1. Create another new file and save it as TutorialStyle.cls
2. Copy the preamble of tutorial.tex into the new file. Starting with the \documentclass command and copy right up to, but not including, the \begin{document} command. Delete the preamble in the original document.
3. Save TutorialStyle.cls and tutorial.tex
4. Modify the first line of TutorialStyle.cls by replacing the word 'document' with the word 'Load'.
5. Add a new first line of tutorial.tex to use your new style class \documentclass{TutorialStyle}
6. Save both files.

10.2.1 Sanity Check

The first line of TutorialStyle.cls should now be:

```
\LoadClass[12pt, oneside, a4paper]{book}
```

The first line of tutorial.tex should be:

```
\documentClass{TutorialStyle}
```

We have just created what a new document class. You can use this class file as a generic preamble for other documents as well. Compile tutorial.tex. Nothing should have changed in the final PDF.

10.3 The Structure of this Document

Lets now look at how this document is organised. It is identical to Chris Forman's thesis because it started out life as that. The structure was usurped and the headings and content were replaced to make a kind of reference manual for this introductory course. This document therefore is three things:

1. An example thesis structure to work from.
2. The LaTeX files which were used to create it can be used as a template for a thesis.
3. It is also a convenient introductory LaTeX manual!

Groovy huh?

10.4 The Master File

There is a master file where LaTeX begins the compilation procedure. This contains the \documentclass command needed to set the ball rolling. Within the file there are a series of \include{} commands, one for each Chapter, which tell LaTeX where to find the files containing each chapter.

The subfiles cannot be compiled on their own because they do not contain the \documentclass, \begin{document} or \end{document} commands. Let us look in more detail at the master file and we will learn some new LaTeX commands.

```
\documentclass{Style/LatexCourseStyle}

\begin{document}
\setcounter{secnumdepth}{2}
\setcounter{tocdepth}{1}

\frontmatter
\onehalfspacing
\include{Frontmatter/Titlepage}
\include{Frontmatter/Declaration}
\include{Frontmatter/Acknowledgements}
\include{Frontmatter/Summary}
\include{Frontmatter/Abbreviations}
\tableofcontents
\listoftables
\listoffigures

\mainmatter
\renewcommand{\sectionmark}[1]{\markright{\thesection \ #1}{}}
\include{Chapters/Prelude}
\include{Chapters/MyFirstPage}
\include{Chapters/Lists}
\include{Chapters/mathsEquations}
\include{Chapters/ChemicalEquations}
\include{Chapters/Tables}
\include{Chapters/Figures}
\include{Chapters/Sectioning}
\include{Chapters/externalReferencing}
\include{Chapters/ThesisTemplateMainFile}
%\include{Chapters/ChapterThatIsNotCompiled}

\appendix
\include{Appendices/UsefulWebsites}

\singlespacing

\backmatter
\addcontentsline{toc}{chapter}{References}
\bibliographystyle{Bibliography/cjfthesisv1}
\renewcommand{\bibname}{References}
\bibliography{Bibliography/LatexCourseBib}

\end{document}
```

10.5 Analysing the Master File

Let us go through this file and observe the new commands. Most of it is self-explanatory.

```
\documentclass{Style/LaTeXCourseStyle}
```

This command tells LaTeX that the document class is the one defined in the the style file "LaTeXCourseStyle", which is in the sub directory "style". Easy.

```
\setcounter{secnumdepth}{2}
\setcounter{tocdepth}{1}
```

The first of these commands tells LaTeX that the section depth number is 2, i.e. throughout the document only bother numbering sections, subsections and sub-subsections down to the second level (such as section 1.2.3). The second command tells LaTeX to only build the table of contents from section numbers down to the first level i.e. 1.1. Easy enough.

The command \frontmatter is exclusive to the book class, on which this style is based, and tells LaTeX that within this section, number the pages using Roman numerals, and don't include these numbers in the table of contents, just like in a book. The \mainmatter and \backmatter commands similarly delineate the other sections of the document.

Child's play. How long would it take you to figure out how to do that in MS Word? No idea. I've never tried.

Note that these rather cool commands are not available in every single class. So for a paper you would use the "article" class where such compartmentalisation of your document is not appropriate. In fact you can often download LaTeX templates from the journal you wish to submit your paper, which is why I concentrate on the thesis template in this course.

```
\onehalfspacing
```

Is obvious. "From this point forward use line spacing of one and a half lines please. Thank you." How many hours have you spent clicking on paragraph indentations for each separate paragraph in MS Word? Nightmare. Want to change the standard in the whole document? Not a chance in MS Word. In LaTeX, you can do it with a single word, carefully placed in the master file.

```
\include{Frontmatter/Titlepage}
\include{Frontmatter/Declaration}
\include{Frontmatter/Acknowledgements}
\include{Frontmatter/Summary}
\include{Frontmatter/Abbreviations}
```

The \include{} command tells LaTeX to include the specified .tex file at this point. Thus it is a simple matter to break up a large document into sub components. I have written each part of the front matter in a separate file and then grouped all the front matter into one directory called: Frontmatter.

See how the directory structure of my files reflects the structure of my book?

```
\tableofcontents
\listoftables
\listoffigures
```

Again, this is easy. These commands mean: "Please scan through the whole document and build for me a table of contents, a list of tables, a list of figures and then write down the entries alongside their section numbers (as I have them ordered in this version of the document), as well as the page numbers on which they appear and create a nice little summary table and add them to the front matter of my document. Thanks."

How does LaTeX do this? Remember the idea of environments? Well each table, figure, equation, etc. is a defined environment. LaTeX just counts them up and assigns numbers starting wherever you want and builds a table. You can tell LaTeX to ignore individual sections if you want to.

You can also control the look and feel of your table of contents with subtlety and grace. Do you want to fill the white space with lines of dots to aid readability, even though the chapter titles are all different lengths? Page numbers? How deep should the TOC be? Etc etc etc. How to do this is beyond the scope of this course, but now you know about it, go look it up. Easy as π.

Ok.

Main matter we've done.

```
\renewcommand{\sectionmark}[1]{\markright{\thesection \ #1}{}}
```

This command tells latex to 'renewcommand', i.e. change the meaning of the following command from this point in the document forward, over-riding previous definitions.

```
\sectionmark[1]{\markright{\thesection \ #1}{}
```

This is LaTeX speak for "please write the section heading at the top right hand side of each page. Thanks." This works in conjunction with a command in the class style file. You have to tell LaTeX to switch on headings as well as where to put them.

```
\include{Chapters/Prelude}
\include{Chapters/MyFirstPage}
\include{Chapters/Lists}
\include{Chapters/mathsEquations}
\include{Chapters/ChemicalEquations}
\include{Chapters/Tables}
\include{Chapters/Figures}
\include{Chapters/Sectioning}
\include{Chapters/externalReferencing}
\include{Chapters/ThesisTemplateMainFile}
%\include{Chapters/ChapterThatIsNotCompiled}
```

This is the include command again. You will notice that one of the include lines is commented out. This is a handy trick to only build certain parts of the thesis, which can be useful if individual chapters are being reviewed.

```
\appendix
%\include{Appendices/UsefulWebsites}

\singlespacing

\backmatter
\addcontentsline{toc}{chapter}{References}
\bibliographystyle{Bibliography/cjfthesisv1}
\renewcommand{\bibname}{References}
\bibliography{Bibliography/LatexCourseBib}
```

The rest is easy. \appendix means we are in the appendix now and start labelling sections with A.1.2 instead of 1.2.3.

\addcontentsline means forcibly insert a contents line about the references into the contents table.

\{bibliographystyle{Bibliography/cjfthesisv1}} means use the biblography style defined in the file cjfthesisv1. Defining a bibliography style is an advanced topic. Usually you can download one from a journal and not worry about the details. This is where you specify how the references will appear (names, journals, full stops, commas etc).

\renewcommand{\bibname}{References} means: rename the bibliography chapter as 'References'.

\biblography means add the following bibliography. You can reference entries in this bibliography file very easily.

The astute among you will realise that LaTeX will have to read the document several times before it can be compiled with all the references. Hey ho. No problem. More advanced front ends will deal with the multiple compilation steps for you. Otherwise you might like to learn about writing a Makefile. You can go and have a cuppa while LaTeX does your references for you in the style of your favorite journal... nice.

Appendix A

Useful Websites

- `https://www.ctan.org/` The Comprehensive TEX Archive Network is the standard repository for packages.
- `https://en.wikibooks.org/wiki/LaTeX` An online manual with accessible introductions on a wide variety of common topics.
- `https://tex.stackexchange.com/` Every question about LATEX you might want to ask, with useful answers.
- `http://detexify.kirelabs.org/classify.html` Quickly find how to get that symbol you don't know the name of.

References

[1] I. Horcas, R. Fernández, J. M. Gómez-Rodríguez, J. Colchero, J. Gómez-Herrero and a. M. Baro, "WSXM: a software for scanning probe microscopy and a tool for nanotechnology.", *The Review of scientific instruments*, 78(1); 013705, 2007. URL `http://www.ncbi.nlm.nih.gov/pubmed/17503926`