LaTeX for Scientists and Engineers (AY2014/2015)

Donavan Cheah and Team

January 1, 2015

Contents

Preface

A common question that arises whenever a LaTeX course is done is "Why learn LaTeX when there exists other word processing software such as Word[1] or OpenOffice?" There exists many reasons, some of which are listed below:

1. Typing articles, reports and books simply yet professionally (such as this).
2. Preparing musical scores, charts and posters.
3. Creating slides.

This sets the stage for learning how a typesetter works. Most of us would have dealt with (and complained about) a word processor when working on significant documents. LaTeX exists as a significantly more robust solution at producing work that is professionally typeset. One big advantage of LaTeX is the convenience it affords to the end-user: deal with the content and LaTeX settles the formatting and layout – i.e. the reason why it is a typesetter rather than just a processor. Examples of this include automated hyperlinking, counting of sections and pages and the ability to typeset well.

Why is this course named "LaTeX for Scientists and Engineers?" The reason is that most of the material here will be relevant for any scientist or engineer who will have to type equations, deal with figures and references, as well as present reports. Such is the nature of any sort of writing: the need to be professional. The scientist or engineer will deal with more complicated material typographically than many others at the university[2], explaining the

[1]Not free, unless pirated.

[2]To the author, everyone can benefit from such a course, although the first group of people who will be interested is the scientist or engineer.

name.

The book is divided into three parts. Chapters 1 and 2 deal with the basics of LaTeX. They contain introductory commands. Therefore, many exercises (especially in Chapter 1) are provided to familiarise the reader with basic LaTeX skills. Chapters 3 and 4 show how LaTeX is useful beyond the basics; Chapter 3 focuses on empowering LaTeX and making use of the engine in performing many seemingly basic but tedious tasks in an expedient way, whereas Chapter 4 focuses on LaTeX on the Internet and how the online community allows the reader to find solutions to almost every problem the reader may encounter. Chapters 5 and 6 shift the emphasis of the course from one of fundamentals to one of introducing a package that settles many problems, and showing how the building on from the package leads to applications. The approach of Chapters 5 and 6 is generalistic in the sense that they focus on a framework of solving problems and how LaTeX will approach it.

Chapters marked in asterisks (*) are optional; they can be omitted on a first reading. The book explores LaTeX from the perspective of a problem-solver, not a theorist or a computer scientist, so expect the exercises and problems to be targetted at approaching problems and using LaTeX to solve them. Therefore, while we establish a working understanding of the mark-up language, we do not aspire to be particularly rigorous with the explanation of the code. The principle behind the approach is that we make use of programming and typesetting languages to aid us in our work.

In addition to the six chapters of the book, appendices have been included to pack more information that do not fit in any particular chapter. In particular, a compiliation of useful sources as well as some common errors that frequently occur are done. Useful sources change from time to time because LaTeX is still undergoing development, as understood by the developing LuaLaTeX framework, but pdfLaTeX is a more or less completed project, which is what the course bases itself on as a foundation. As LaTeX continues to undergo active development, it is hoped that the book also encourages active learning and the desire to improve. There is no end "mastery" to LaTeX, but anyone in reality can think of the methodology behind such an approach and rely on LaTeX to solve problems. This is perhaps the harder skill the book tries to emphasise on rather than rote remembrance and memory of commands, as what a book focused on just mathematical and graphical typesetting will do. Nevertheless, there have been certain books written on LaTeX that are mainstays, as well as online resources; these have been highlighted for the reader.

The book is primarily based on the course conducted on the third iteration (Dec '14), though the notes were already beginning to be written on the second iteration (June '14). This marks, as a personal milestone, what I have managed to do from learning LaTeX from a senior in the Special Programmes in Science to contributing to it through writing a series of course notes, organised in a book form, with accompanying video. Relevant sections will start off with a link to the relevant video(s) as done for the course. It is hoped that the reader will make use of both video and text, as well as practice to make the most out of this introductory material.

Writing a book for the first time has been a learning journey in itself due to the style of writing required for such material. It is felt that, after being able to compile this, my job at establishing some foothold of LaTeX in an academic setting (in this case, the Special Programme in Science) is done. May the next better person succeed me in this endeavour.

This book would not be possible without the help of a few significant people. I would personally like to extend my gratitude, in particular, to:

1. Siu Zhuo Bin (on the way to obtaining his PhD at time of writing) for supporting the general direction of the course and helping to guide students since the first iteration.

2. Wong Chi Yan for helping to write parts of the book, helping to conduct the bibliography section for the third iteration of the course, and vetting the book since the second iteration.

3. Leong Jin Feng for producing video material for both the second and third iteration of the course as well as vetting some parts of the final copy of the book.

4. Joseph Chan Man Yau and Shawn Tang for writing some parts of the book.

5. Aaron Chong and Yeo Zhen Yuan for their general assistance in helping for the course.

However, the course would not have been possible without the presence of enthusiastic students. Some students have come back from previous iterations to help guide the next batch of students, which is indeed delightful.

A class picture of the third iteration is shown in Fig. 1.

Figure 1: Group photo for the third iteration of the course conducted on 8 and 10 December. This is a batch containing primarily the batch who entered NUS in AY 14/15, and certainly brightened up my life as an undergraduate while teaching this introductory class.

It is timely that I sign off here and wish the reader best of luck in exploring what LaTeX is capable of.

As a parting gift besides the book, I present, also, the end of course video, viewable at `https://www.youtube.com/watch?v=-jjNXYoKlo8`.

Best regards
Donavan Cheah

Part I

Basics in LaTeX: Getting Started

Chapter 0

Essentials Before Starting LaTeX

0.1 Installation

Quick instructions for one who wishes to install LaTeX onto their PC/-Mac/Linux:

1. Go to the site: `http://latex-project.org/ftp.html`.
2. Pick your OS and install the relevant files.
3. **Important Notice:** While the Mac package provide TeXShop and TeXworks, these are not as friendly to the end-user as TeXMaker[1]. For that, visit: `http://www.xm1math.net/texmaker/`. We will be using this TeX editor for the course, so please install this unless you are already familiar with LaTeX. [2]
4. We will also be using a bibliography manager in this course. For the Macs, BibDesk is included in the MacTeX package and is the most convenient bibliography manager for the course. For the rest, we run JabRef. For that, visit: `http://jabref.sourceforge.net/`.

Please perform the installation procedures before attempting to read Chapter 1

[1]In my opinion.

[2]As of the time of writing, we used TeXMaker ver. 4.1.1.

0.2 What's in the Various TEX Packages?

For Macs: MacTEX provides: TEXLive 2014[3] (the distribution, i.e. the back-end – important!), BibDesk (bibliography manager – important), LATEXiT (a utility to write short LATEX typesetting matter with export capabilities), TEXShop (editor), TEXworks (editor), the spell checker Excalibur, and some documentation. Click on `http://www.tug.org/mactex/What_Is_Installed.pdf` to find out more.

For Windows: MikTEXor proTEXt. Refer to `http://www.tug.org/protext/` for more.

For Linux: Similar to Macs.

0.3 Other Materials

Important: We will be making use of a cloud system. It will be good to sign up for *both* a WriteLATEX and ShareLATEX account; a significant part of the course will be conducted on these cloud platforms. Two short links to help you create an account[4]

- ShareLATEX: `https://www.sharelatex.com?r=d9360f0d&rm=d&rs=b`
- WriteLATEX: `https://www.writelatex.com/signup?ref=6f0d2f4a7ec9`

[3]This is updated yearly, so the time of reading may not correspond to the year printed.
[4]More space is always welcome for the instructor, so these links are referral links for cloud space.

Chapter 1

Basic Matters

The first software that normally is introduced to the reader when preparing documents is Microsoft Word, where what you type is immediately what you get (a What You See Is What You Get interface, WYSIWYG for short), LaTeX isn't. This may come as surprising to the end-user who may not have dealt with such an environment before. However, while the interface is different, we will see in the material why this interface is superior for many purposes compared to a WYSIWYG interface.

The first question to ask is: how does LaTeX work? This is our starting point before the formal introduction of the syntax. The importance of fundamentals cannot be understated as they will be repeated over and over again in the course. The reader is strongly advised to always think: how can work be done on LaTeX and how can it be done better?

To help with the learning process, the book contains many exercises and examples of both the fundamental concepts as well as some exploratory exercises that will enrich the reader in grasping the capabilities of LaTeX as well as understanding the key ideas. Exercises range from simple search engine exercises to more complex interpretations of how a solution is presented. Three values will be emphasized throughout the material: thinking about how LaTeX solves a particular problem, searching for solutions, and modifying them to suit the individual needs. We eschew from a "from scratch" approach because that often is time-wasting, especially since many problems the reader will come across are already solved problems.

Some conventions to be used from here onwards: all commands are written in `verbatim` (otherwise we can't write them, as you will see why.). This includes all symbols required. We will indicate language specific to LaTeX using **boldface**, whereas exercises will be marked as **Exercise 1**. These conventions will hopefully help the reader in following the course, and searching the boldface material online to have a better idea on what it does, since the boldface words are indeed standard vocabulary in the world of LaTeX. There will be a few practices for the reader at the end of each chapter to explore other facets of LaTeX which we hope are useful. We hope a few conventional typesets will help the reader to follow through the book better.

1.1 How does LaTeX Work?

(Section 1.1 corresponds to `https://www.youtube.com/watch?v=UkK-i11WCvQ` in the course.)

LaTeX works on the basis that you leave all issues of formatting, and document arrangement to LaTeX to optimise. This differs from classic WYSIWYG interfaces such as Word. This carries with it certain advantages, including but not limited to the following:

1. .tex files on their own are just plain text. That means file size saving, higher immunity to file corruption due to bad writing of files and higher portability.

2. You don't make bad errors as a human when judging how text and graphics should appear.

3. For minimal effort, you produce work of high typographical standard.

Of course, this book is meant to be a practical introduction to LaTeX so we will not dwell too much on theoretical understanding on how it works.

1.2 Syntax

Without proper syntax, nothing works. The table below summarises the symbols that will be used by LaTeX as critical syntax to tell LaTeX to perform an action. This is shown in Table 1.1.

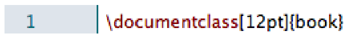
Symbol	Meaning
\	Always used at the start to invoke a command.
{ and }	Encloses the arguments to be used for the command.
[and]	Encloses options that work alongside the command.
%	Commenting – leaving out lines that will not be processed.
&	Spacings in tables and align environments.
$	Switches between a quick math environment and text environment.
#	Used in various definitions, defining number of arguments, e.t.c.

Table 1.1: Meaning of Symbols used on LaTeX

To type these symbols as they are, we use a backslash \ in front of the symbol (except to type a backslash, of course).

Exercise 1. *Typing a backslash: Search on Google if it's possible to type a backslash without the use of a* `verbatim` *command.*

To start writing a LaTeX document, we first need to ask what we want LaTeX to do, which is called from the **document class**. Examples include the **article**, **report** and **book** class. This is written using the **book** document class. Selecting the document class is written on the very first line of the text as shown in Fig. 1.1[1].



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

Figure 1.1: First line. The braces are compulsory. These are called **arguments**. They denote the document class. The square brackets are optional, and naturally are called **options**.

[1]Document classes are governed by a .cls file, which will be briefly mentioned in the course later.

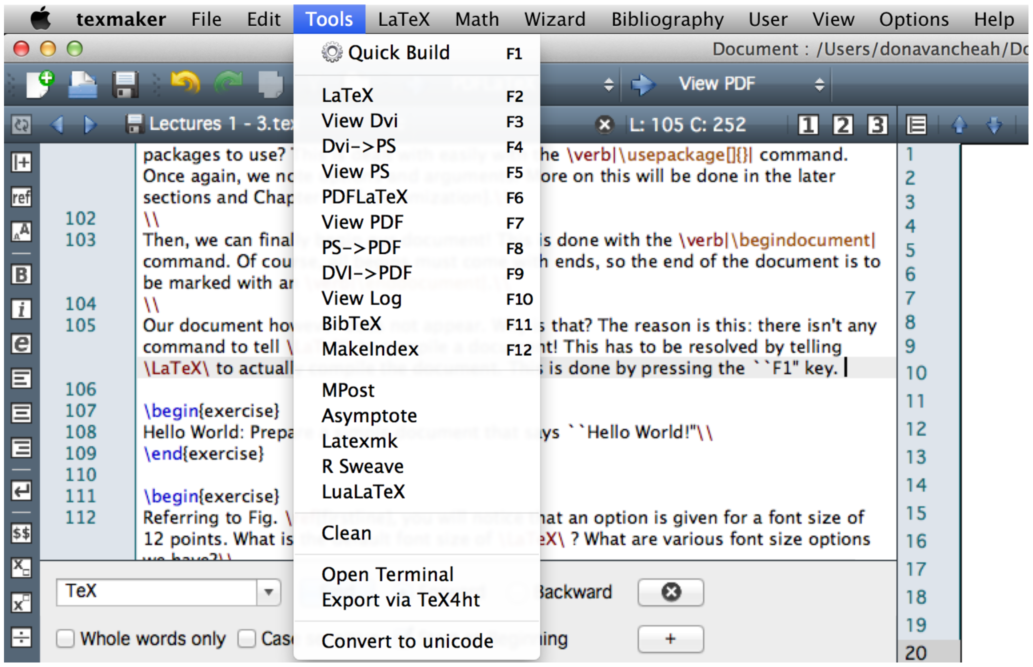
As per programming languages, there are often customised definitions in the form of **packages**. Invoking them requires the `\usepackage[]{}` command. More on this will be done in the later sections and Chapter 3.

Finally, we can invoke `\begin{document}` to begin the document. Beginning must end, calling for `\end{document}`. This applies to many commands in LaTeX; LaTeX needs to know when a command starts and ends.

Unlike a WYSIWYG system, LaTeX needs to be told to compile the document, which is done by pressing the "F1" key for TeXmaker, or finding such an equivalent "quick build" or "compile" on your editor. The options for compilations is shown in Fig. 1.2. It might be worth noting that compilations in TeXmaker are done in the pdfLaTeX format, which invokes the pdfTeX engine.

Exercise 2. *Optional exercise: Development for pdfTeX is mostly complete, which is why we study it at the beginner level. There are a few approaches to TeX: LuaTeX, XeTeXand ConTeXT. Google what each of the following is, and try to arrange them on a hierarchy to understand how TeX has evolved from its roots till today.*

You may want to attempt this exercise only after finishing the course.



```
packages to use?  he \verb|\usepackage[]{}| command.
Once again, we no  ore on this will be done in the later
sections and Chap
\\
Then, we can final  s done with the \verb|\begindocument|
command. Of cou  h ends, so the end of the document is to
be marked with ar
\\
Our document hov  s that? The reason is this: there isn't any
command to tell \  nt! This has to be resolved by telling
\LaTeX\ to actuall  s is done by pressing the ``F1" key.

\begin{exercise}
Hello World: Prepa  ys ``Hello World!"\\
\end{exercise}

\begin{exercise}
Referring to Fig. \  hat an option is given for a font size of
12 points. What is  eX\ ? What are various font size options
```

Figure 1.2: The shortcuts for compilation. You can find this under Tools. Of course, the long way to F1 is "Tools → Quick Build" for this case. An advanced user can attempt to figure out between a DVI output, PDF output and PS output in a typical LaTeX book.

This compiles the document proper and generates a bunch of files! To do that, however, LaTeX needs to be instructed where to compile the files too. Therefore, save the file first (in a folder). Then press F1. LaTeX generates a number of files, not all of which will be explained in a foundation LaTeX course, but a few files are worth mentioning at this point. These are:

1. The .tex file: where you work everything on. This is where your input is stored and should not throw this away!

2. The .log file: a log file of everything that happened with the .tex file during compilation (we will not do .log file analysis in this course.)

3. The .pdf file: the .pdf output. (Yes, it's automated!)

4. The .aux file: contains the essential information for LaTeX to plug links.

Exercise 3. *Hello World: Prepare a simple document that says "Hello World!" Open the .pdf file in a .pdf reader of your choice to verify that it is indeed correct.*

Exercise 4. *Referring to Fig. 1.1, you will notice that an option is given for a font size of 12 points. What is the default font size of LaTeX and what are various font size options we have?*

Solution: One can explicitly force a font size by introducing a modifier. Google typing some sizes such as `\Huge text` *and* `\footnotesize text`*. (Note the lack of braces.) You can read* http://texblog.org/2012/08/29/changing-the-font-size-in-latex/ *for more information on this.*

Exercise 5.

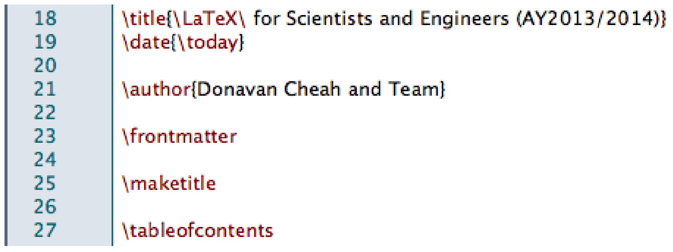
LaTeX makes use of special symbols such as the `{}` *(braces) and* `[]` *(square brackets) to enclose arguments. Braces represent compulsory arguments whereas square brackets represent optional arguments.*

1. *In many commands, there exists options for both. Discuss what happens if arguments were not inserted for braces and square brackets.*
2. *The braces used in isolation also has another meaning. What is it and why is it important to note the second meaning?*

Note: This exercise is a fundamental that will be repeated over and over again. Do this first!

1.2.1 Title Pages, Content Pages

Title pages and content pages are easily generated. These can be done simply as shown in Fig. 1.3.



```
\title{\LaTeX\ for Scientists and Engineers (AY2013/2014)}
\date{\today}

\author{Donavan Cheah and Team}

\frontmatter

\maketitle

\tableofcontents
```

Figure 1.3: LaTeX commands are very intuitive. Think you can figure out what each of the following does? (Except for the **front matter**, which is a command for the **book** document class, the rest should be obvious.)

1.3 Organising a Document

(Section 1.3 corresponds to `https://www.youtube.com/watch?v=lm0Bo186HqM` in the course.)

A long document typically has **chapters**, **sections**, **subsections**, **subsubsections**, **paragraphs**, **subparagraphs** that have to be arranged. The commands are invoked as the words suggest.

Not all document classes support everything! For example, the **article** document class does not support chapters, while the **letter** document class, being a letter, clearly won't divide into subsections! Nonetheless, document parts come in auto-numbered, auto-arranged and auto-optimised. That means you can add, remove or edit any part of the document without fear of numbering problem; LaTeX automatically updates these for you.

Exercise 6. *Fiddle around with **section**, **subsection**, **paragraph** and **subparagraph**. How different do they look? Can we move sections around without affecting the order? Try it. Also, see how many "subs" you can add. What's the maximum? (Any more and you'll be better off re-organising your document.)*

*Note: For **reports** and **books**, there exists a higher classification by the use of* `\chapter{}` *. This is not present in the **article** document class.*

Exercise 7. *Type* `\section*{Test Section}` *(note the asterisk). Then,* *generate a table of contents using* `\tableofcontents` *and repeat the exercise. Interpret your observations.*

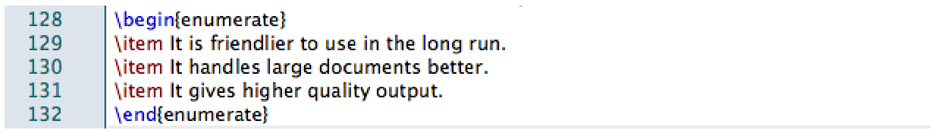
1.3.1 Lists

(Subsection 1.3.1 corresponds to `https://www.youtube.com/watch?v=d_VWrnOpcLM` in the course.)

One of the most common things in a document is to **enumerate**, **itemize**[2] or **describe** a list. Example: LaTeX is better than Word because

1. It is friendlier to use in the long run.
2. It handles large documents better.
3. It gives higher quality output.

(3 of the many reasons.) The mark-up language is:



```
\begin{enumerate}
\item It is friendlier to use in the long run.
\item It handles large documents better.
\item It gives higher quality output.
\end{enumerate}
```

Figure 1.4: To generate a numbered list, use the **enumeration** command, with each item clearly indicated by indicating the **item** for distinct numbering. Never feel irritated by Autocorrect ever again.

Exercise 8. *Make a list. Change* ***enumerate*** *to* ***itemize****. Also check how numbering and bullets can be customised. For instance, if we want enumeration to be done with Roman numerals (e.g. (i), (ii), (iii)), how should it be done?*

Exercise 9. *Write a profile of yourself, including your name, age, gender, course of study and your hobbies using* ***description****.*

Exercise 10. *Nest a list within a list. Is it possible? How many times can you do such nesting?*

[2] LaTeX spells things American. That includes commands.

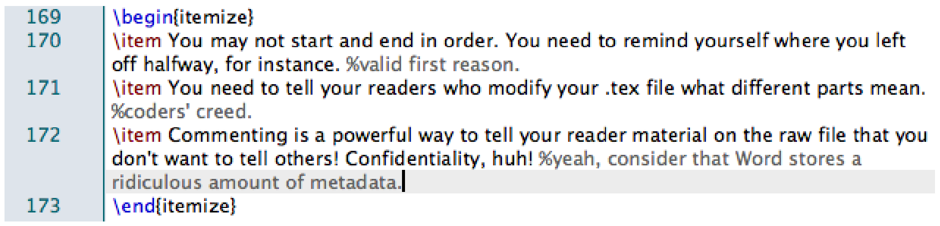
Also, by referring to Exercise 8, modify the commands to change the defaults of the nested lists (both enumerated and itemised ones).

1.3.2 Commenting

It is essential to learn how to comment on your work. Why is this so? Three good reasons :

- You may work in an order that is not chronological. Placing comments helps you to remember where you left material out.
- You need to tell your readers who modify your .tex file what different parts mean.
- Commenting is a powerful way to tell your reader material on the raw file that you don't want to tell others! Confidentiality, huh!
- Commenting is an easy way to strike out lines you do not want to typeset due to various reasons. E.g. tracing errors. This is called "commenting out" code.

Comments do not show up in main text, as shown in the example in Fig. 1.5, the itemised list typed above.



```
\begin{itemize}
\item You may not start and end in order. You need to remind yourself where you left
off halfway, for instance. %valid first reason.
\item You need to tell your readers who modify your .tex file what different parts mean.
%coders' creed.
\item Commenting is a powerful way to tell your reader material on the raw file that you
don't want to tell others! Confidentiality, huh! %yeah, consider that Word stores a
ridiculous amount of metadata.
\end{itemize}
```

Figure 1.5: Suppressing output with the use of a percent sign. All input thereafter in the line is greyed.

1.3.3 Fonts, Alignments, Footnotes

Fonts can be modified, but not all fonts exist[3]. For instance, `\textbf{}` bolds text, whereas `\textit{}` italicises text.

Exercise 11. *What other text modifications can you do? Try* `\textsc{}`, `\textsf{}` *as examples.*

Note: When introduced to the math environment, such modifiers are replaced by their math components, i.e. `\mathbf{}` *for bolded font in math environment.*

Alignment is an easy task. Typically LaTeX settles matters by justifying (and even so no problems of the Word justify kind appear). To change the alignment of something, simply invoke `\begin{center}` and `\end{center}`[4].

Sometimes more information needs to be spelt out to the reader. LaTeX does it brilliantly with footnotes. Just type `\footnote{}` and enclose your footnote within the braces.

1.4 Mathematical Typesetting with amsmath

(Section 1.4 corresponds to `https://www.youtube.com/watch?v=25SkIDr9SnM` in the course.)

Typesetting mathematics with LaTeX is one of the most widely known advantages. First, we will introduce students to the **equation environment**[5]. This can be done by using `\begin{equation}`. Remember that all begins must come with ends, so `\end{equation}` follows thereafter.

An elementary example: the sum of kinetic energy and elastic potential

[3] An example is Times New Roman.
[4] Right alignment requires you to type “flushright” instead of “center”.
[5] Strictly speaking this environment does not need the amsmath package. A basic set of mathematical typesetting is already present without the use of packages.

energy of a mass-spring system is its total energy as shown in Eqn. 1.1:

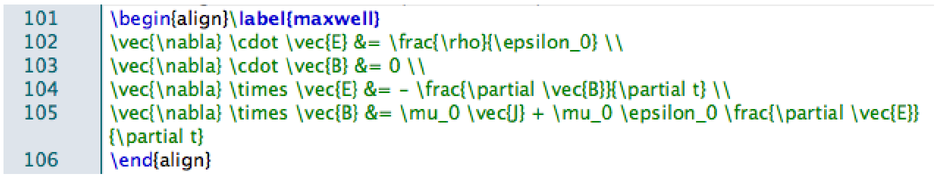
$$E_{\text{total}} = \frac{1}{2}mv^2 + \frac{1}{2}kx^2 \tag{1.1}$$

Mathematical typesetting encompass typesetting a variety of special symbols and/or accented characters (e.g. vectors), as shown in the following examples.

To write multiple equations, the **align** environment can be used instead of the **equation** environment. Below is an example of the two points earlier:

$$\begin{align}
\vec{\nabla} \cdot \vec{E} &= \frac{\rho}{\epsilon_0} \tag{1.2}\\
\vec{\nabla} \cdot \vec{B} &= 0 \tag{1.3}\\
\vec{\nabla} \times \vec{E} &= -\frac{\partial \vec{B}}{\partial t} \tag{1.4}\\
\vec{\nabla} \times \vec{B} &= \mu_0 \vec{J} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \tag{1.5}
\end{align}$$

Fig. 1.6 shows how it is done.



```
\begin{align}\label{maxwell}
\vec{\nabla} \cdot \vec{E} &= \frac{\rho}{\epsilon_0} \\
\vec{\nabla} \cdot \vec{B} &= 0 \\
\vec{\nabla} \times \vec{E} &= - \frac{\partial \vec{B}}{\partial t} \\
\vec{\nabla} \times \vec{B} &= \mu_0 \vec{J} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}
{\partial t}
\end{align}
```

Figure 1.6: Maxwell's 4 equations of electricity and magnetism in differential form from Eqn. 1.2.

Writing in boldface can be done by using `\mathbf{}`.

Exercise 12. *More on Fractions: Type fractions within fractions. Comment on what looks better.*

Also, try typing in-line fractions. Comment on the appearance.

Note: There is quite a bit of debate over how fractions should be typed in-line. This ranges from just stuffing it inline, such as $\frac{a}{b}$*, or* a/b*. Try this method of* *going about this (useful for cookbook recipes, for e.g.): Define the following in the preamble:*

```
\newcommand*\rfrac[2]{{}^{#1}\!/_{#2}}
```

and use your new command `$\rfrac{a}{b}$` *to see the results (try it yourself.). The equivalent can be achieved using the* ***xfrac*** *package and invoking* `\sfrac{#1}{#2}`.

Exercise 13. *Boldface Typing: Write the 4 Maxwell's equations in boldface. You may notice something amiss. Correct that.*

Solution: Generally non-Roman letters must be bolded using the `\bm` *command that can be invoked by loading the* ***bm*** *package. More practice is available in Exercise 14.*

Exercise 14. 1. *Typeset the following, i.e. the curl of a vector field in spherical coordinates (as stated explicitly):*

$$\begin{aligned}\nabla \times \mathbf{v} = \frac{1}{r\sin\theta}\left[\frac{\partial\left(\sin\theta v_\phi\right)}{\partial\theta} - \frac{\partial v_\theta}{\partial\phi}\right]\hat{\mathbf{r}} \\ + \frac{1}{r}\left[\frac{1}{\sin\theta}\frac{\partial v_r}{\partial\phi} - \frac{\partial\left(r v_\phi\right)}{\partial r}\right]\hat{\boldsymbol{\theta}} \\ + \frac{1}{r}\left[\frac{\partial\left(r v_\theta\right)}{\partial r} - \frac{\partial v_r}{\partial\theta}\right]\hat{\boldsymbol{\phi}}\end{aligned} \tag{1.6}$$

Note: There are many solutions to this problem. The .tex file used to generate this does it differently from the earlier section!. Therefore, do not be too alarmed!

2. *A similar way to generate this is to use an* ***equation*** *environment and a* ***aligned*** *subenvironment. What is the difference?*

Note: The numbering method for this align environment is done by a `\notag` *method.*

We then work into our arsenal of math typesetting more derivatives, such as the time derivative, indicated with `\dot{}` and `\ddot{}` (try `\dddot{}` and higher terms, if you wish). An example: we want to introduce the damped simple harmonic oscillator, which is given by $\ddot{x} + \gamma\dot{x} + \omega^2 x = \vec{F}(t)$. Notice the equation is embedded "in-line", which can be done with a single pair of `$` signs.

Exercise 15. *Can we end* `$` *with an* `\end{equation}`*?*

Solution: No, you cannot, even though they show up as changing from text to math environment and back. The reasons: in-line math is NOT a math float. Moreover, an "end" was invoked without a "beginning". That simply spells trouble.

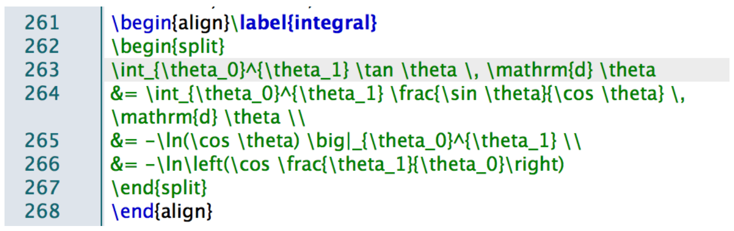
Exercise 16. *There exists other methods of invoking math environments. Attempt:* `\(` *and* `\)`*, as well as* `\[` *and* `\]`*. Then, Google "math environments" and interpret your observations.*

Suppression of equation labelling is similar to what was done in Section 1.3.

The **split** environment is one method of controlling equation numbering; sometimes some lines in an **align** environment should be numbered whereas others should not. The example is shown in Eqn. 1.7.

$$\begin{split}\int_{\theta_0}^{\theta_1} \tan\theta \,\mathrm{d}\theta &= \int_{\theta_0}^{\theta_1} \frac{\sin\theta}{\cos\theta}\,\mathrm{d}\theta \\ &= -\ln(\cos\theta)\Big|_{\theta_0}^{\theta_1} \\ &= -\ln\left(\cos\frac{\theta_1}{\theta_0}\right)\end{split} \tag{1.7}$$

Fig. 1.7 is what we type in the editor for Eqn. 1.7.



```
\begin{align}\label{integral}
\begin{split}
\int_{\theta_0}^{\theta_1} \tan \theta \, \mathrm{d} \theta
&= \int_{\theta_0}^{\theta_1} \frac{\sin \theta}{\cos \theta} \,
\mathrm{d} \theta \\
&= -\ln(\cos \theta) \big|_{\theta_0}^{\theta_1} \\
&= -\ln\left(\cos \frac{\theta_1}{\theta_0}\right)
\end{split}
\end{align}
```

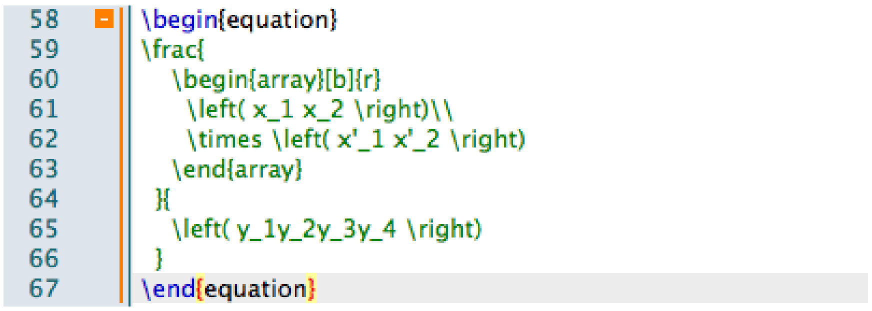
Figure 1.7: Note the importance of using braces not only to enclose commands, but enclose where a subscript or superscript command will act upon. The **split** is responsible for the equation number appearing once only, as compared to the Maxwell's Equation case.

Digressing to discuss good typesetting is timely at this point. Good typesetting means that there exists little ambiguity in what you convey in your writing. In this case, a simple integration could go very wrong if we had not paid attention to the spacing, or the fact that dθ is indeed a differential that demands integration. Note the syntax and how it corresponds to subtle details such as a missing space (resolved using the `\,` command), or an italicised d. Good typesetting is in the details. That also explains why logarithms and trigonometry come with them a backslash symbol before that to emphasise that it's a function, not a variable.

Exercise 17. *a) Delimiters: You may have noticed the* `\left` *and* `\right` *commands. Google how LaTeX interprets them and why it is necessary to tell LaTeX* `\left` *and* `\right`*.*

b) Try typesetting "`\left.`*" and "*`\right.`*". Why type that even if no output exists? Hint: This is useful for Exercise 20.*

Additional Notes: These are examples of delimiters. Generally, these are instructions for LaTeX on how certain alignment should be done beyond the basic. An example is available by typing the following in Fig. 1.11:



```
\begin{equation}
\frac{
   \begin{array}[b]{r}
     \left( x_1 x_2 \right)\\
     \times \left( x'_1 x'_2 \right)
   \end{array}
  }{
   \left( y_1y_2y_3y_4 \right)
  }
\end{equation}
```

Figure 1.8: Remember this from primary school?

This leads to:

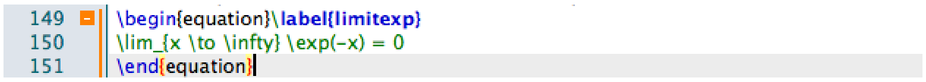
$$\frac{\begin{array}[b]{r}\left(x_1 x_2 \right)\\ \times \left(x'_1 x'_2 \right)\end{array}}{\left(y_1y_2y_3y_4 \right)} \tag{1.8}$$

Good try, but may be worth further aligning.

An example showing the typesetting of the limit and exponential function is:

$$\lim_{x \to +\infty} \exp(-x) = 0 \tag{1.9}$$

The mark-up is shown in Fig. 1.9.



```
\begin{equation}\label{limitexp}
\lim_{x \to \infty} \exp(-x) = 0
\end{equation}
```

Figure 1.9: Taking limits in Eqn. 1.9. You will notice that in TEXMaker, highlighting an end line tells you where the corresponding begin is. Try this on an equation environment and press the "minus" sign. A nifty feature.

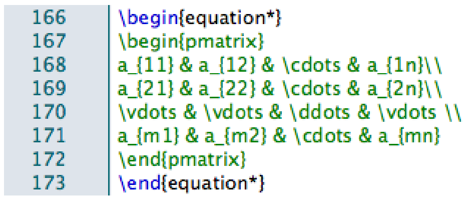
Exercise 18. 1. *Other integration symbols: Find the commands for the following: closed surface/loop integral symbol. Also, try* `\iint` *and* `\iiint`*. Try with and without limits. As a note, the* `\sum` *and* `\prod` *command works in a similar way as the integration one.*

2. *Investigate the use of the* `\displaystyle` *command.*

Typing matrices is important. By extension, this includes vectors and tensors. In LATEX , these are done as arrays, but customised matrix environments exist. If you remember the ampersand (&) symbol, you'll see it appear here for the first time. We shall type a $m \times n$ matrix.

$$\begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n}\\
a_{21} & a_{22} & \cdots & a_{2n}\\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{pmatrix}$$

Fig. 1.10 shows the mark-up language for it.



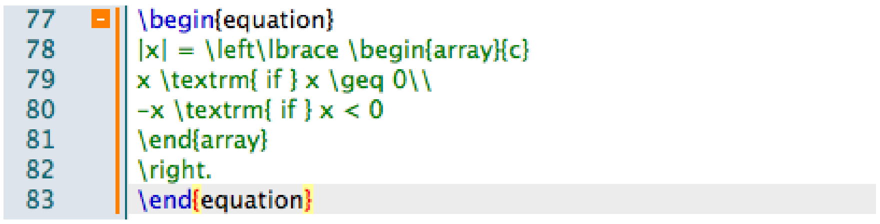
```
\begin{equation*}
\begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1n}\\
a_{21} & a_{22} & \cdots & a_{2n}\\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{pmatrix}
\end{equation*}
```

Figure 1.10: Typing dots and matrices. Also, evidence of numerical count suppression with the use of the asterisk. The ampersand is used to align the columns of the matrix.

Exercise 19. *Explore the* ***bmatrix****,* ***Bmatrix****,* ***vmatrix*** *and* ***Vmatrix****.*

Exercise 20. *Besides the* ***matrix*** *command, find other ways of generating a matrix (e.g. typing a determinant). Investigate, especially, the commands* `\binom`, `\choose`, `\begin{array}`-`\end{array}`*. Use the array command to write a conditional such as:* $|x| = x$ *if* $x \geq 0$ *and* $-x$ *if* $x < 0$*.*

Solution: An array is just a collection of elements in some sort of a table environment with every element aligned down the centre. This lends itself usefulness in describing many things, ranging from matrices to conditionals. Some things one can do with arrays are shown as examples, one of which is already shown in Eqn. 1.11.



```
\begin{equation}
|x| = \left\lbrace \begin{array}{c}
x \textrm{ if } x \geq 0\\
-x \textrm{ if } x < 0
\end{array}
\right.
\end{equation}
```

Figure 1.11: Answer to Exercise 14: Writing a Simple Conditional

That naturally leads to:

$$|x| = \left\lbrace \begin{array}{c} x \textit{ if } x \geq 0 \\ -x \textit{ if } x < 0 \end{array} \right. \tag{1.10}$$

Of course, some valuable insight should be gathered from the following:

1. *Many multi-row outputs can be written nicely in an array environment.*
2. *Usage of* `\textrm{}` *allows for local text mode, which is very valuable in writing certain descriptons, since text mode allows for manual spaces. It is also a solution to write it using* `\mathrm{}`*, but extra spaces will be required through the use of* `\,`*. I figure that is in no way any more elegant.*
3. *The* `\left.` *and* `\right.` *stand for "dummy" pairings so that you can invoke symbols defined with the* ***left*** *and* ***right*** *conventions in isolation.*

Try writing a BINGO card. An example:

BINGO Card

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}
a_{51}	a_{52}	a_{53}	a_{54}	a_{55}

An example of a BINGO Card. Not in a figure, table or picture environment!

Note the usefulness of arrays, once again.

Typically, in quantum mechanics (QM), we use the Dirac bra-ket notation. The angled brackets in bra-ket notation are not inequality signs! That is a hallmark of poor typesetting. Here we introduce a few last bits of what we want to talk about when it comes to equations.

$$\begin{align}\left\langle \Psi \right\rangle &\rightarrow \textsf{Time-averaging.} \tag{1.11}\\ \left\langle \Psi | \chi \right\rangle &\rightarrow \textsf{The inner product.} \tag{1.12}\\ \left| \Psi \right\rangle \left\langle \chi \right| &: \textsf{Not equal to the inner product!} \tag{1.13}\end{align}$$

This is shown in Fig. 1.12 to illustrate the differences in mark-up language.

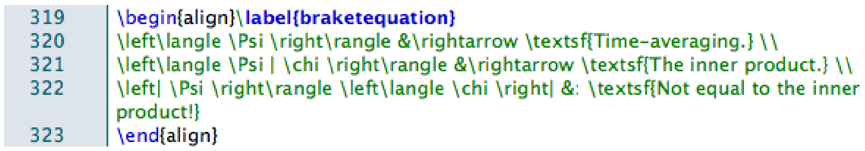

```
\begin{align}\label{braketequation}
\left\langle \Psi \right\rangle &\rightarrow \textsf{Time-averaging.} \\
\left\langle \Psi | \chi \right\rangle &\rightarrow \textsf{The inner product.} \\
\left| \Psi \right\rangle \left\langle \chi \right| &: \textsf{Not equal to the inner
product!}
\end{align}
```

Figure 1.12: Note this example; we will return to it in a later chapter since the repetition of bras and kets can be annoying. Refer to Appendix A to find out more. Also see how text can be embedded into the math environment with spaces.

Before we move into a new topic, we note that we commented in the math environment. By making use of a `\textsf{}` interface (we call that font sans serif while the default Computer modern font is serif), we emphasised our point. Sometimes comments can go awry because of inappropriate spacing. Exploration of this will be left as an exercise for the reader. We finally get to answering the question "How does LaTeX do spacing in math environment?" The answer is left for exploration in the exercise below.

Exercise 21. *Spacing: a) Try* `\,`, `\quad` *and* `\qquad`. *How much spacing is provided with each?*

b) If you are fussy, this may not be good enough since you do not have the luxury of customising spacing size. Read up how `\hskip` *and* `\vskip` *work.*

c) In (a) we dealt with horizontal spacing. Vertical spacing can be dealt *with using the commands* `\smallskip`, `\medskip` *and* `\bigskip`. *How far do they skip and how are they used?*

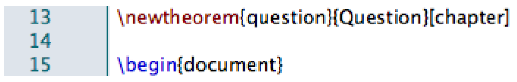
d) Why is spacing often defined both above and below in environment boxes? (e.g. equations have space above and below, as an example.)

A list of special symbols can be found in http://en.wikibooks.org/wiki/LaTeX/Special_Characters or a more comprehensive list at http://www.tex.ac.uk/tex-archive/info/symbols/comprehensive/symbols-a4.pdf (a whopping 164 pages!) for typesetting convenience. This can be Googled, of course.

1.4.1 Theorem and Proof

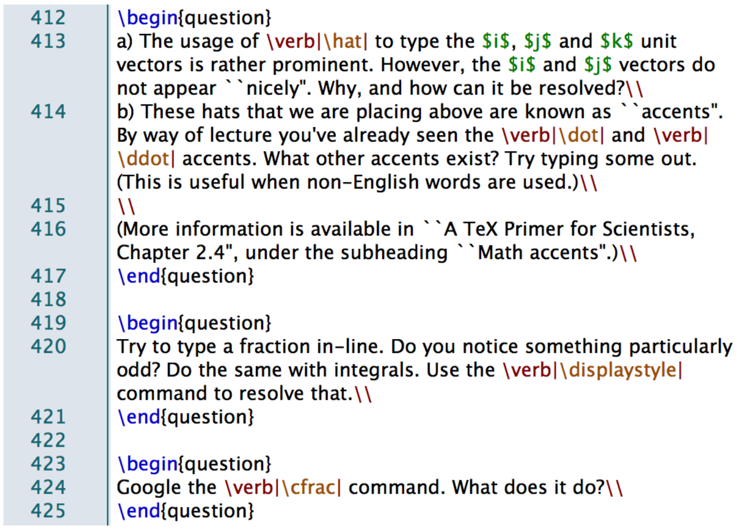
(Subsection 1.4.1 corresponds to https://www.youtube.com/watch?v=0DapVJtprrY in the course.)

LaTeX's versatility in mathematical typesetting lends naturally to the **theorem** and **proof** environments. These require the use of the **amsthm** package. Thereafter, the type of theorem can be defined and referenced, as shown in Fig. 1.13 and Fig. 1.14.



```
\newtheorem{question}{Question}[chapter]

\begin{document}
```

Figure 1.13: Theorem definition. Check with Fig. 1.14: the first set of braces define the environment name, the second set of braces is the heading name and the option represents where the counter will be reset. In this case, the question counter resets every chapter.



```
\begin{question}
a) The usage of \verb|\hat| to type the $i$, $j$ and $k$ unit
vectors is rather prominent. However, the $i$ and $j$ vectors do
not appear ``nicely". Why, and how can it be resolved?\\
b) These hats that we are placing above are known as ``accents".
By way of lecture you've already seen the \verb|\dot| and \verb|
\ddot| accents. What other accents exist? Try typing some out.
(This is useful when non-English words are used.)\\
\\
(More information is available in ``A TeX Primer for Scientists,
Chapter 2.4", under the subheading ``Math accents".)\\
\end{question}

\begin{question}
Try to type a fraction in-line. Do you notice something particularly
odd? Do the same with integrals. Use the \verb|\displaystyle|
command to resolve that.\\
\end{question}

\begin{question}
Google the \verb|\cfrac| command. What does it do?\\
\end{question}
```

Figure 1.14: An example of the theorem class defined in Fig. 1.13.

The theorem environment allows us to introduce some sort of "definition" in our own document. More on this will be done in Chapter 3.

The proof environment is simpler. All it requires is `\begin{proof}` and `\end{proof}`. The q.e.d. "white box" is typeset as well.

Proof. LaTeX is more versatile than Word in most tasks. □

1.5 Chemistry Typesetting with mhchem

(Section 1.5 corresponds to https://www.youtube.com/watch?v=3-anKI2wMiU of the course.)

The **mhchem** package enables the chemistry student to typeset chemical equations in LaTeX[6]. We will adopt a different approach and just jump into applications after introducing basic material.

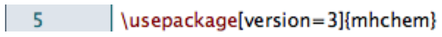
[6]Typing chemistry in Word, as far as the author knows, is quite a pain as well.

1.5.1 Chemical Equations

Here is a simple example:

$$2\,\mathrm{H_2O_2(l)} \longrightarrow 2\,\mathrm{H_2O(l)} + \mathrm{O_2(g)} \text{ and } 2\,\mathrm{H_2O_{2(l)}} \longrightarrow 2\,\mathrm{H_2O_{(l)}} + \mathrm{O_{2(g)}} \tag{1.14}$$

Chemical equations typeset differ from their math counterparts due to the lack of italics. This is achieved with the `\ce{}` command. However, this would require invoking the **mhchem** package. This is seen in Fig. 1.15. Notice the use of the option. The mhchem package also has much simpler commands for arrows as compared to the math counterparts. An example: `->` instead of `\rightarrow` in mhchem[7].



```
\usepackage[version=3]{mhchem}
```

Figure 1.15: Invoking the latest mhchem version. This option implies the 3rd version. Other options may be used on different packages; it is wise to consult the package documentation on such matters. These are usually available on the Comprehensive TEX Archive Network (CTAN) page.

You might have noticed we called for optional argument on the version number. Reason: the package works without options but is not the latest. Nevertheless, let us make a few points about the package using examples.

1. The mhchem package works in both text mode and math mode. What does this mean? Subtle but noticeable: text font and math font on LATEX is slightly different. Simply drop the `$` to type in the text mode. E.g. typesetting $\mathrm{SO_4^{2-}} + \mathrm{Ba^{2+}} \longrightarrow \mathrm{BaSO_4}\downarrow$ works as well in text mode as math mode. (If you want to see when mhchem operates on text mode and math mode, work with a sans serif font; it'll be obvious there. Refer to Chapter 3 for more information on this. However, all mathematical operations must still be in math environment.

2. The mhchem package has different arrows that are specific to chemistry typesetting[8].

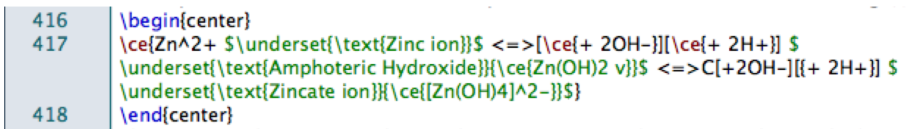
[7] mhchem is a very well-established package in chemistry circles, just as how amsmath is in mathematical circles. The exact capabilities of mhchem will not be done in the course, but a brief introduction should suffice for many purposes.

[8] Due to TikZ.

These points are illustrated in the examples that follow. We consider the following example which introduces a few ideas at one go:

$$\underset{\text{}}{\mathrm{Zn^{2+}}} \xrightleftharpoons[+2\,\mathrm{H^+}]{+2\,\mathrm{OH^-}} \underset{\text{Amphoteric Hydroxide}}{\mathrm{Zn(OH)_2}\downarrow} \xrightleftharpoons[+2\,\mathrm{H^+}]{+2\,\mathrm{OH^-}} \underset{\text{Zincate ion}}{[\mathrm{Zn(OH)_4}]^{2-}}$$

This is a typical expression (adapted from the mhchem package and corrected for English) showing how a zinc ion results in an amphoteric hydroxide upon the addition of a base, and further addition results in it being soluble once again. The reverse happens as well. This was typeset as shown in Fig. 1.16.



```
\begin{center}
\ce{Zn^2+ $\underset{\text{Zinc ion}}$ <=>[\ce{+ 2OH-}][\ce{+ 2H+}] $
\underset{\text{Amphoteric Hydroxide}}{\ce{Zn(OH)2 v}}$ <=>C[+2OH-][{+ 2H+}] $
\underset{\text{Zincate ion}}{\ce{[Zn(OH)4]^2-}}$}
\end{center}
```

Figure 1.16: A typeset equation with more details. Notice three new things: the text beneath the chemical, the generation of the down arrow in mhchem, and how the addition of the ions are typed.

In chemistry, the default LaTeX arrows tend to have arrowheads that are too large. Below shows an example of how the arrows are typeset using standard LaTeX commands and the mhchem commands. However, the command below requires the **TikZ** package! We will learn the power of **TikZ** in Chapter 6.

$$A \xleftrightarrow{\text{description}} B \text{ and } A \xleftrightarrow{\text{description}} B$$

1.6 Units

(Section 1.6 corresponds to `https://www.youtube.com/watch?v=bfPoBrvj-sk` in the course.)

Typesetting physical quantities in LaTeX properly is simple. This requires the **siunitx** package, which conveniently solves the problem[9]. We begin with

[9]Yes, you could cheat and constantly switch between text and math modes. Do let the instructor know if there is a simpler units package.

two simple examples:

`$4 \si{\, \centi \meter}$` typesets 4 cm
`$9.81 \si{\, \meter \per \second \squared}$` typesets 9.81 m/s^2

Note: you read them as they are. 4 centi (prefix) meter (SI unit). The `\,` gives the necessary spacing, which works for more complicated units as well! Some exercises are provided in Exercise 22 to get you more familiarised with the package.

Exercise 22. *a) Typeset the other base SI units and derived SI units.*

b) Some units are not that straightforward to typeset. Typeset the angstrom and degree symbol (Note: You may want to search harder for the degree symbol.).

Solution: The Google exercise should have showed 2 ways of doing the degree symbol, of which one is not correct. 3 ways of doing it right are presented in *http://en.wikibooks.org/wiki/LaTeX/Special_Characters*.

1.7 Hyperlinking with Hyperref

(Section 1.7 corresponds to `https://www.youtube.com/watch?v=A64urqHhfjw` in the course.)

LaTeX allows for powerful hyperlinking especially with the use of the **hyperref** package. Important commands: `\label{}` and `\ref{}`. Intuitively, this means, labelling subject matter and referring to them. Any object such as a figure, table or equation can be labelled and therefore referred to. More on this will be discussed in Section 1.8.

Hyperlinks appear on citations as well using the `\cite{}` command. These are automated as well! More on this will be discussed in Chapter 2.

The exploration of all the options available with the **hyperref** package is left as a work in progress for the reader to appreciate the power of this package, which comes essential to any typesetter.

1.8 Environments

(Section 1.8 corresponds to `https://www.youtube.com/watch?v=XgYcnswi6cI`, `https://www.youtube.com/watch?v=kzZDZZNpne0` and `https://www.youtube.com/watch?v=uQ3BpYciZjQ` in the course.)

Objects like tables, figures and more can be placed in **environments**. Since LaTeX compiles by block, environments help to define blocks. There exists a few environment blocks, ranging from **alignment** blocks to **object** blocks. In fact, environments were introduced when dealing with equations (object blocks) and centering (alignment blocks).

1.8.1 Some Basic Environment Concepts

Environments define a specific rule between the areas which it is defined. For that, environments start with `\begin{}` and end with `\end{}`. Examples of such rules include changing a text environment to a math environment and are encased in **floats**. All floats, by nature, can be applied a caption and label to for reference purposes, which help greatly in report-writing. Examples include equations, tables, figures and graphics. More on these will be discussed in the upcoming subsection.

1.8.2 Figures and Tables Environments

Within an environment, one can insert objects. To invoke the **figure** environment, simply include `\begin{figure}` and `\end{figure}`. This is analogous for **tables**.

Figures and Float

To include a figure in a specific place in your work, the figure environment (for the figure count), the `graphicx` package (for graphics options) and the

`\includegraphics[]{}` command must be invoked at the corresponding location in the syntax. The figure to be included should be saved in the same folder that the .tex file is saved in[10]. Below is an example of how figures can be placed in a paper.

In your preamble, insert: `\usepackage{graphicx}`

In your main body:

```
\begin{figure}
\includegraphics{picture.jpg}
\end{figure}
```

However, the command above leaves LaTeX to optimise the location of the figure, which may not suit you. Using the **float** package and modifying its location through replacing `\begin{figure}` with `\begin{figure}[h!]` calls for the figure to be clamped at that point in the syntax through `[h!]`. Explore `[t]` and `[b]` for other figure locations. If it fails, load the **placeins** package and insert a `\FloatBarrier` after `\end{figure}` to force a barrier for the figure.[11].

The size of the pictures can be controlled by inserting either a `[scale=x]` (scaling by a scale factor with respect to the image) or `[textwidth=y]` (scaling with respect to the text width) option in between `\includegraphics` and the braces enclosing the file. `x` and `y` stand for numbers, generally between 0 and 1.

`\caption{}` command introduces an enumerated caption. Below is an example of syntax of a figure with caption and label.

```
\begin{figure}[h!]
\includegraphics[textwidth=0.5]{picture.jpg}
\caption{Just for laughs}
\end{figure}
```

[10]You can manually specify the directory for the file, but we will save this for you to explore.

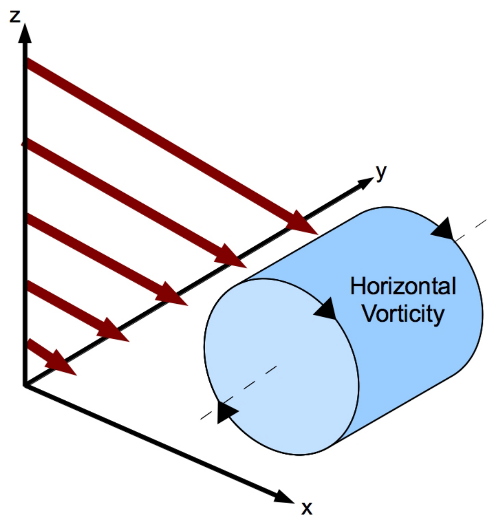
[11]It is a complicated affair to explain how floats are processed. This has to deal with the issue of overfilling or underfilling a document, which is not within the scope of the course!

```
\FloatBarrier
```

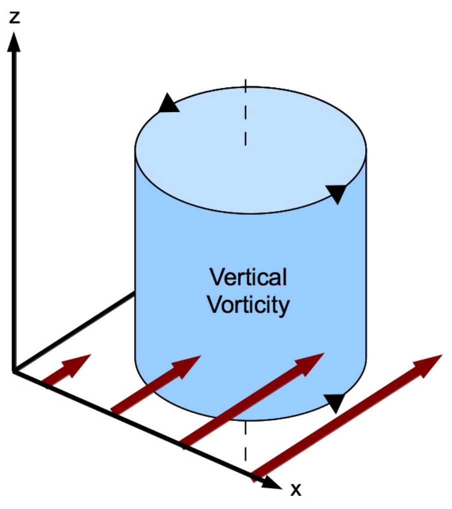
Putting Several Figures Side-by-side

Placing figures side-by-side achieves space saving, neatness and usefulness of comparison. To do so, load the **graphicx**, **floats** and **subcaption** packages. Afterwards, we create subfigure environments within figure environments for each picture, as shown in Fig. 1.17 below.

Generation of Vorticity via Wind Shear



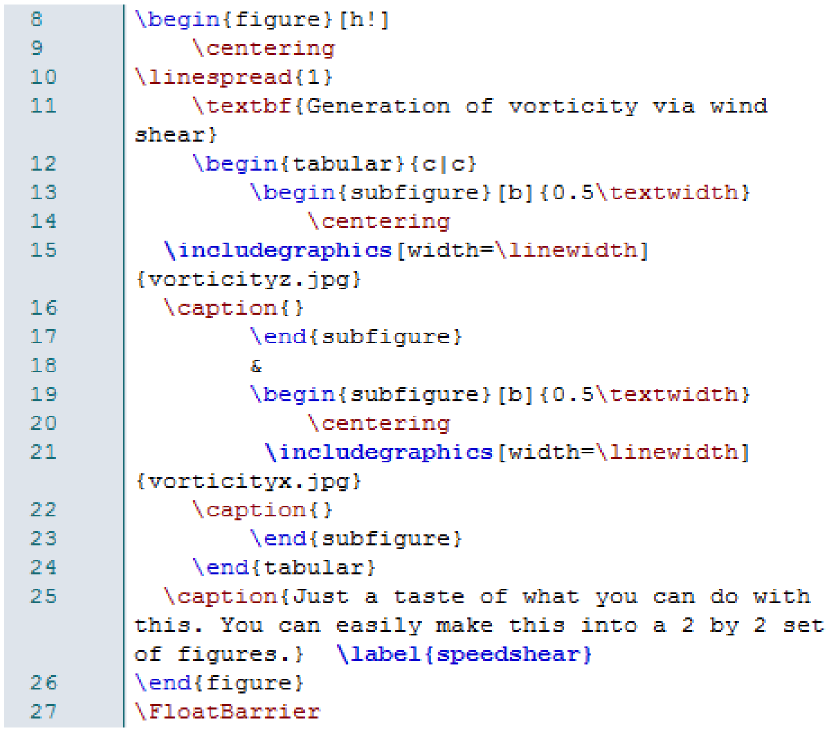
(a)



(b)

Figure 1.17: An array of figures.

An idea of the syntax for Fig. 1.17 is shown in Fig. 1.18.



```
\begin{figure}[h!]
    \centering
\linespread{1}
    \textbf{Generation of vorticity via wind shear}
    \begin{tabular}{c|c}
        \begin{subfigure}[b]{0.5\textwidth}
            \centering
  \includegraphics[width=\linewidth]{vorticityz.jpg}
  \caption{}
        \end{subfigure}
        &
        \begin{subfigure}[b]{0.5\textwidth}
            \centering
         \includegraphics[width=\linewidth]{vorticityx.jpg}
    \caption{}
        \end{subfigure}
    \end{tabular}
  \caption{Just a taste of what you can do with this. You can easily make this into a 2 by 2 set of figures.}  \label{speedshear}
\end{figure}
\FloatBarrier
```

Figure 1.18: The syntax for Fig. 1.17. Notice the centering is different. Google the difference between centering and the center environment.

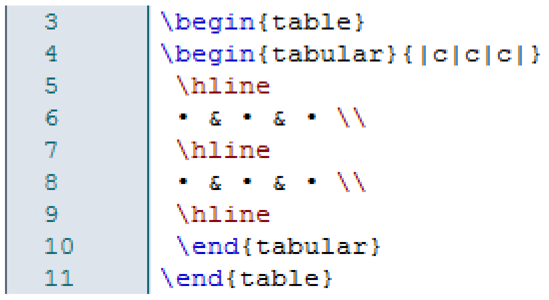
The `\FloatBarrier` command at the end anchors the diagram at the location that corresponds to the position of the syntax in the `.tex` file. As you may have expected, there was a **figure** environment invoked. Line spacing for the words is controlled with `\linespread{1}`.

The purpose of the **tabular** environment is to split the figure into 2 smaller halves (more on it in Section 1.8.2). The `{c|c}` does the splitting as it denotes 2 columns and 1 separation in between. The `c` horizontally centers elements within the column. The **subfigure** environments are then added to insert the figures. Notice that the **subfigure** environment had to have its limits defined?

By altering the syntax above a little, it is possible to create a 2 by 2 array of pictures to appear in the same area. This will be left as an exercise for the reader.

Tables

To include a table in a specific place in your work, the `\begin{table}` command must be invoked (for the environment) together with `\end{table}`[12]. The main skeleton of including a table is shown in Fig. 1.19.



```
\begin{table}
\begin{tabular}{|c|c|c|}
 \hline
 • & • & • \\
 \hline
 • & • & • \\
 \hline
 \end{tabular}
\end{table}
```

Figure 1.19: The basics of including a table.

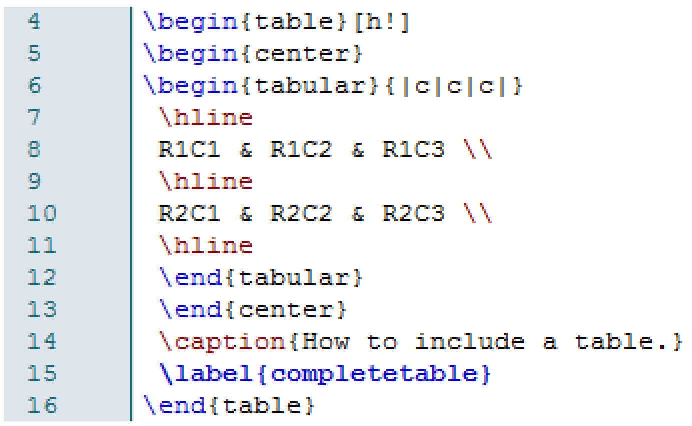
As mentioned earlier, the command in Fig. 1.19 alone may not be sufficient in making the table appear the way you want it to. To solve this problem, replace the `\begin{table}` with `\begin{table}[h!]`. For the table to be placed in the center of the document (*height-wise*), replace `[h!]` with `[c]`. If all else fails, inserting `\FloatBarrier` after the `\end{table}` command will also work for tables. Do note that the **placeins** package needs to be invoked to use `\FloatBarrier`.

Exercise 23. *Must a **table** float necessarily contain a table? Explain.*

Exercise 24. *Do other floats exist? Give examples. Why do we run different types of floats?*

An example of a completed inclusion of a table into a report is shown in Fig. 1.20.

[12]Strictly speaking we do not have to be that pedantic. However, there is good reason to float tables just as there is with figures.



```
\begin{table}[h!]
\begin{center}
\begin{tabular}{|c|c|c|}
 \hline
 R1C1 & R1C2 & R1C3 \\
 \hline
 R2C1 & R2C2 & R2C3 \\
 \hline
 \end{tabular}
 \end{center}
 \caption{How to include a table.}
 \label{completetable}
\end{table}
```

Figure 1.20: The syntax for including a table.

The result of the syntax above can be seen in Table 1.2.

R1C1	R1C2	R1C3
R2C1	R2C2	R2C3

Table 1.2: How to include a table.

The `\begin{center}` and `\end{center}` commands place the table in the center of the document (*width-wise*).

The | in |`c`|`c`|`c`| indicates a vertical line while `\hline` indicates a horizontal line. Removing a | in |`c`|`c`|`c`| removes a vertical line and removing a `\hline` removes a horizontal line.

Exercise 25. *Similar to figures, the size of the tables can be controlled. Try replacing* `c` *with* `p{z units}`*, where z is a number and "units" can be "*cm*" or "*in*".*

Solution: Replacing `c` *with* `p{z units}` *allows us to adjust the column widths.* `\begin{tabular}`|`c`|`c`|`c`| *gives the default column widths, which may not be what we desire. Hence, modification of*

`\begin{tabular}`|`c`|`c`|`c`| *to* `\begin{tabular}`|`p{4cm}`|`p{4cm}`|`p{4cm}`|

adjust column widths to be 4 cm.

Exercise 26. *Refer to Exercise 25. If text in a table exceeds length specified, what happens?*

*Note: This also introduces the common problems involving **underfull**, **overfull** and **badbox** problems that are all space-related warnings. These generally hint that LaTeX has problem finding optimal locations or placements fot text, which results in certain alignment issues.*

Exercise 27. *Modify the* `c` *in the second option in the* `tabular` *command with something like* `l` *or* `r`*. What happens?*

Exercise 28. *What does* `\multicolumn{}{}{}` *do?*

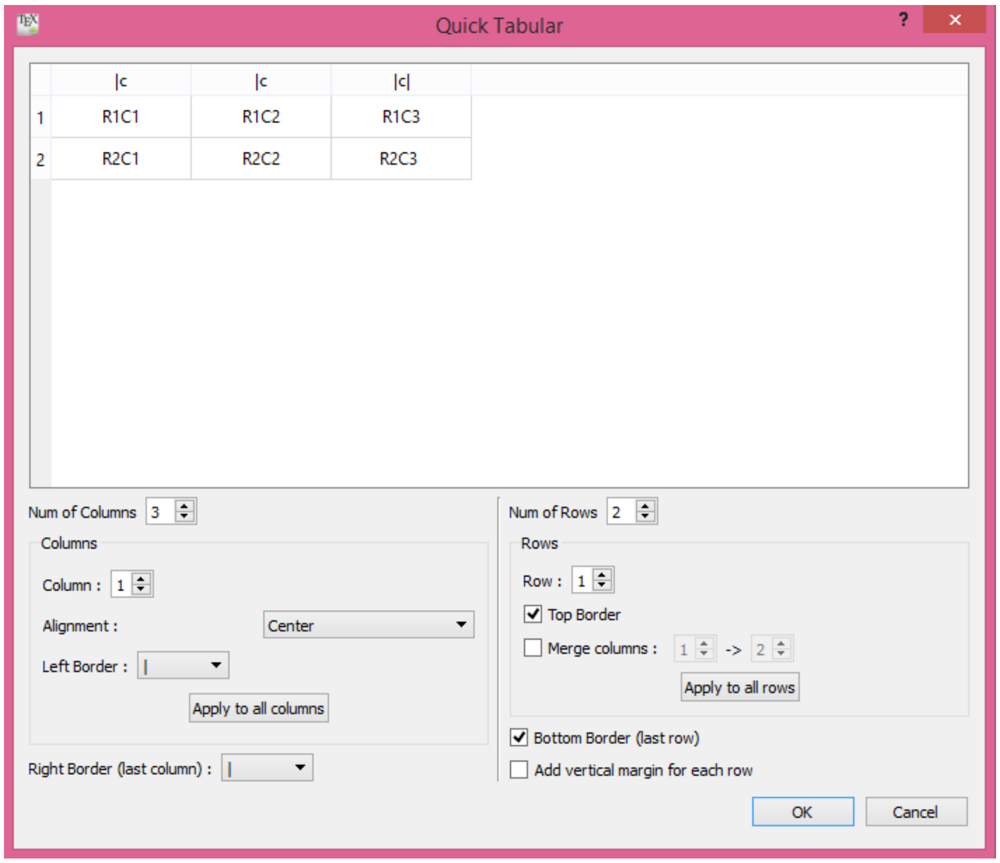
Solution: `\multicolumn{}{}{}` *in LaTeX merges cells. The first argument is for the number of cells merged. The second argument is for the positioning of the contents of the cell. The third argument is for the contents of the cell.*

Exercise 29. *As we know, some reports require tables to be formatted with no vertical lines and minimal horizontal lines (e.g. publishing in Raffles Bulletin of Zoology). Try to generate the following table shown in Table 1.3. Hint: make use of the skills learnt from the exercises above. The first column has a width of 1* cm *while the other 3 columns have widths of 3.5* cm.

	Main Heading		
	Sub-Heading 1	Sub-Heading 2	Sub-Heading 3
1			
2			
3			
4			

Table 1.3: Table to be generated.

Creating a table manually may be cumbersome. Go to "Wizard" and click on "Quick Tabular" for a GUI interface. The Quick Tabular Wizard allows GUI-level input prior to generation of mark-up language for a table, as seen in Fig. 1.21. The resultant syntax is shown in Fig. 1.22.


Quick Tabular

	|c	|c	|c|
1	R1C1	R1C2	R1C3
2	R2C1	R2C2	R2C3

Num of Columns 3

Columns

Column : 1

Alignment : Center

Left Border : |

Apply to all columns

Right Border (last column) : |

Num of Rows 2

Rows

Row : 1

[✓] Top Border

[ ] Merge columns : 1 -> 2

Apply to all rows

[✓] Bottom Border (last row)

[ ] Add vertical margin for each row

OK Cancel

Figure 1.21: The Quick Tabular Wizard.

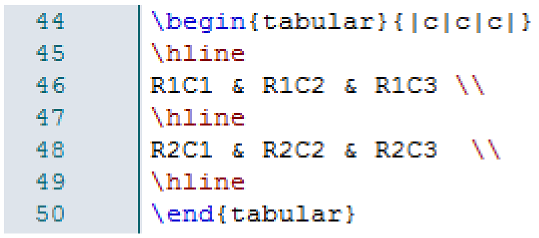

```latex
\begin{tabular}{|c|c|c|}
\hline
R1C1 & R1C2 & R1C3 \\
\hline
R2C1 & R2C2 & R2C3  \\
\hline
\end{tabular}
```

Figure 1.22: The resultant syntax after using the quick tabular wizard as shown in Fig. 1.21.

Counts: Pagination

You might notice LaTeX can keep count correctly and automatically. In fact, many documents make use of multiple counts, such as page and environment counts, which explains the use of floats. They allow LaTeX to keep track of the counts automatically (through floats). As stated earlier, an asterisk suppresses count. Generally, LaTeX does a good job at managing these counts. Another important count is the page count, as shown below:

```
\thepage \, of \pageref{LastPage}
```

Intuitively, `\thepage` references to the current page, `\pageref{X}` references to Page `X`, and `\,` just manually introduces a space. Note that the **lastpage** package needs to be introduced for a page count.

1.9 Chapter Exercises

Question 1.1. *LaTeX has superior typographical capability. Here is a question to explore that. Often, Word gets mixed up (badly) when it tries to autocorrect hyphens, dashes, and quotes. In LaTeX there are ways to ensure you type the correct thing no matter what. Find the proper way to type a em-dash, en-dash, hyphen, and encase a set of text in proper quotation marks. (That is, compare between “-”, “–” and “—”.)*

Solution: Typing dashes simply involve typing “`-`”, “`--`” and “`---`” for the various lengths. Interestingly, LaTeX will recognise `--` and take it as a negative sign in the text environment. Quotation marks use the ‘ for the open and ’ for the close quotation mark. There are no smart quotes, so a character key has to be used.

Question 1.2. *How should in-paragraph enumeration be done?*

Question 1.3. *The book uses headers and footers from the* ***book*** *document class. However, one way of customising headers and footers involves using the* ***fancyhdr*** *package. Google its use.*

Question 1.4. *a) The usage of* `\hat` *to type the i, j and k unit vectors is rather prominent. However, the i and j vectors do not appear "nicely". Why, and how can it be resolved?*

b) These hats that we are placing above are known as "accents". By way of lecture you've already seen the `\dot` *and* `\ddot` *accents. What other accents exist? Try typing some out. (Useful for non-English words.)*

(More information is available in "A TEX Primer for Scientists, Chapter 2.4", under the subheading "Math accents".)

Solution: The problem with merely typing `$\hat(i)$`, `$\hat(j)$` *is that the accent is placed above the letter, which retains the dot. In this case we do not want the dots, requiring*`$\imath$` *and* `$\jmath$` *to produce $\imath$ and $\jmath$. Then add the accents, i.e.* `$\hat{imath}$` *to produce $\hat{imath}$.*

Accents are typed pretty much the same way as the hats are applied, e.g. $\bar{X}$ refers to expectation value of X and is typed `$\bar{X}$`. *Click here for a more detailed account on treating accents. (Note that the link in this case is suppressed using the syntax* `\href{<url>}{<description}`.

Question 1.5. *Google the* `\cfrac` *command. What does it do?*

Question 1.6. *With composite expressions, it becomes difficult to tell which brackets pair with which. E.g. $\frac{\mathrm{d}}{\mathrm{d}x}(g(x,y))$. Resolve the problem.*

Solution: A way to handle this problem is by sizing the brackets. For example, `( \big( \Big( \bigg( \Bigg(` *will produce $(\big(\Big(\bigg(\Bigg($. This provides brackets in 5 sizes, which hopefully should satisfy the problem at hand.*

Question 1.7. *a) Investigate how $\overbrace{\mathcal{A}+\mathcal{B}+\cdots+\mathcal{Z}}^{\text{26 letters}}$ are typed. Do the same for investigating $\overbrace{\mathfrak{A},\mathfrak{B}\cdots,\mathfrak{Z}}^{\text{also 26 letters}}$ Hint: calligraphic letters and overbrace. This is useful in explaining implicit working (the overbrace).*

b) Find the "inexact differential" symbol. If you find these complicated to type, read Chapter 3 to learn to define a new command.

c) Type the following:

1. *The limit infimum with a limit (e.g.* $\liminf_{x\to L}$*).*
2. *Mapping arrows. (e.g.* $f : x \mapsto x + 1$*).*
3. *Set notation. (e.g.* $\forall x \in \mathbb{R}$*,* $\exists y \in \mathbb{R}$ *such that* $x + y = 4$ *(obviously trivial).)*

Note: It may take a Google search to find some of these.

Question 1.8. *a) In statistics, the overline is used to type averages. Try both* `\overline` *and* `\underline`*. What do you get? A sample looks like this:* $\sigma^2(X) = \overline{X^2} - \overline{X}^2$*.*

b) In text, one can type `\underline`*, but it does not look as nice. What's the corresponding* <u>*underlining*</u> *command?*

Question 1.9. *a) You may have tried to type verbatim or a footnote in a caption and LaTeX probably spewed out error(s). Why?*

b) The solution lies in protecting the command. Google what this means and try this out.

c) What does it mean therefore by "protecting" a command? Verify that the same solution is required for footnotes.

(Compare between protected frame and fragile frame.)

Chapter 2

Bibliography

(Chapter 2 corresponds to `https://www.youtube.com/watch?v=QHcPsdPI2Cc` and `https://www.youtube.com/watch?v=ZBcxS4kdxpk` in the course.)

An academic work with no citations is utter nonsense[1]. Often when typing any sort of work, there exists a need to manage citations.

Bibliography styles vary from APA[2] to any sort of in-house journal or society format. Often, it is desirable to keep the details of the bibliography entry using some sort of citation manager before deciding on a format of citation. Fortunately, LaTeX handles that effectively with the `\bibliography{}` and `\bibliographystyle{}` commands.

2.1 Introduction to Bibliography

Bibliography management is performed in a few methods: manual entry as a .txt file and compiling it manually as a .bib file, writing the bibliography manually and using a bibliography manager to compile, as well as just writing the bibliography manually. All methods will be discussed for the reader

[1]Unless it is revolutionary.

[2]Irritating format, as justified in Chapter 2.1.4.

to decide what might be best[3]. The following subsections will detail the compilation of the .bib files.

No matter what method one chooses to do the .bib file, it is essential to incorporate the .bib file by the use of the F11 command. F11 reads the bibliography .bib file. Recall in Chapter 1, compiling is done with F1. The sequence for compilation is to compile once (F1), run BibTEX (F11), and compile twice (F1 twice). Exercise 30 goes through the essential idea. For a more detailed account, a Google search on BibTEX is sufficient.

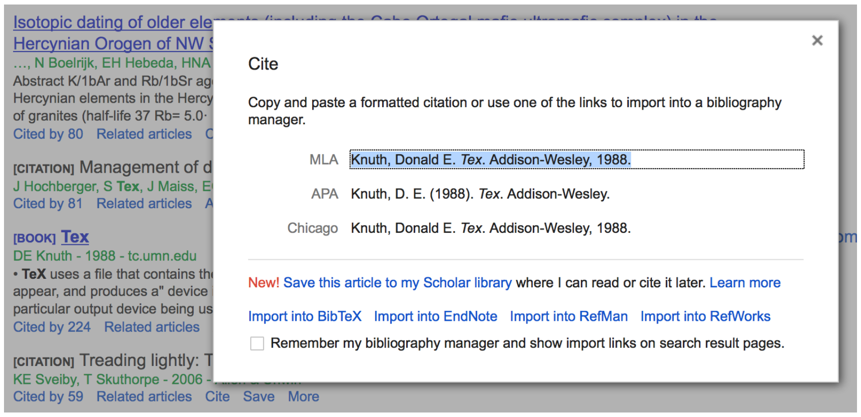
Exercise 30.
1. *Why is there a need to perform the compilation as such for a bibliography?*
2. *(Optional part-question): Each time you perform this sequence, you update merely "changes" to the document. How does LATEX then remember the order and the previous set of rules governing the compilation of the .pdf output? Hint: Refer to Chapter 1.*

Note: This is an important exercise!

2.1.1 Compiling the .bib File Manually

With the .bib entries from a BibTEX import feature, such as from Google Scholar, you will be able to compile the .bib file. This can be found as shown in Fig. 2.1

[3]In this course we will be using the BibTEX method of citation. An advanced reader can read up more on other bibliographic capabilities of LATEX to find out other things like BibLATEX (That exists as a template in ShareLATEX as an example.)



Cite

Copy and paste a formatted citation or use one of the links to import into a bibliography manager.

MLA	Knuth, Donald E. *Tex.* Addison-Wesley, 1988.
APA	Knuth, D. E. (1988). *Tex.* Addison-Wesley.
Chicago	Knuth, Donald E. *Tex.* Addison-Wesley, 1988.

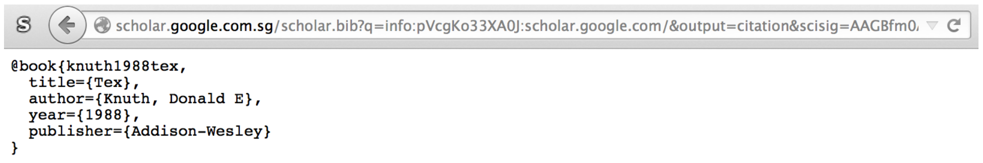
New! Save this article to my Scholar library where I can read or cite it later. Learn more

Import into BibTeX Import into EndNote Import into RefMan Import into RefWorks

☐ Remember my bibliography manager and show import links on search result pages.

Figure 2.1: Citation options on Google Scholar. Quite limited.

Import this into BibTEX and we arrive at Fig. 2.2



```
scholar.google.com.sg/scholar.bib?q=info:pVcgKo33XA0J:scholar.google.com/&output=citation&scisig=AAGBfm0
```

```
@book{knuth1988tex,
  title={Tex},
  author={Knuth, Donald E},
  year={1988},
  publisher={Addison-Wesley}
}
```

Figure 2.2: The .bib entry

All you need to now do is to assemble it in a typical text editor (e.g. where you write your LATEX files, or even Notepad[4]) and save it as a .bib file (Important!). This can be achieved by manually changing the file extension while saving the file. Be sure it reads .bib, NOT .txt or anything else! Make sure that the formatting is correct – raw data can often be inaccurate!

Citing is simply done by using `\cite{}`, where the citekey is inserted into the braces. In the case of the book used in Fig. 2.2, the citekey is `knuth1988tex`. Therefore, we key in `\cite{knuth1988tex}` when citing his book in-text. However, two questions remain unanswered:

1. Which bibliography file (.bib) will LATEX read?
2. What style do we like?

[4]If you use a Mac, be aware that your Notepad equivalent, TextEdit, writes in .rtf. Change that to a Unicode UTF+8 method of processing, i.e. a plaintext file. Otherwise BibTEX will run into problem reading it!

We therefore invoke two more commands, namely `\bibliography{}` and `\bibliographystyle{}`. `\bibliography{}` merely calls the file to be read, whereas `\bibliographystyle{}` demands for the style to be cited in (a plain citation, or journal-specific, e.g. APS/ieeetr).

Be warned that LaTeX does not recognise all kinds of works. A list is available at `http://www.dickimaw-books.com/latex/thesis/html/bibformat.html`. Examples include article, book, manual, thesis. Websites are noticeably not supported because it is not *peer-reviewed* typically; cite it as a misc source instead.

Exercise 31. *Check the command* `\citep{}`*. What variants of the* `\cite` *command are there?*

2.1.2 Usage of Bibliography Managers*

Bibliography managers are used extensively in citation because manual entry and management tends to be too tedious. Examples include **BibDesk**, **JabRef**, **Zotero** and **EndNote**. The MacTeX download contains BibDesk, making it a natural choice for Mac users. However, it is not available for Windows! Note that all these use the BibTeX method of citation. Refer to Appendix A for more information.

BibDesk (Mac Users Only)*

BibDesk is a repository for bibliographical references. Its main advantage is its direct importing of bibliographic references from academic websites and databases. To import, insert the URL of the page where your reference paper is referenced into the “Web” tab and click on the “Import” button when it appears. Note that BibDesk marks off the important fields with a boldface, and that if a new bibliography entry does not contain a citekey, you will not be able to proceed.

It is strongly recommended that the BibDesk file that you are creating for your paper should be saved in the same immediate folder of your paper’s TeX file. This will usually simplify things unless you wish to have just one central biblography file.

The most important trick is knowing what to do with any capitalisation issues that may crop up in your references list. Sometimes, BibDesk is unable to process your capitalised letters as capitalised and LaTeX considers them to be uncapitalised. The simplest solution is to insert braces `{}` around the letter/word you wish to capitalise. `{{}}` is also important: these preserve capitalisation (and lack thereof)[5].

Note that some important bibliography information may be unintentionally left out. Hence, check to make sure that all of these information are present in your BibDesk file, adding the missing information.

Some useful links:

1. `http://bibdesk.sourceforge.net/manual/`: the official manual. Good if you understand it. Contains lots of information about the organisation of BibTeX as the starting two chapters.
2. `http://sourceforge.net/apps/mediawiki/bibdesk/index.php?`: BibDesk wiki.

JabRef*

JabRef is available as a software or Java applet. This is advantageous for frequent travellers or those who use offline computers frequently. A new entry starts as per Fig. 2.3.

[5]You can do this to the .bib file directly as well.

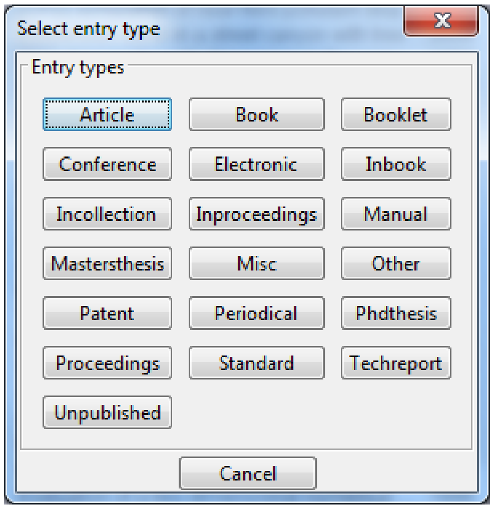
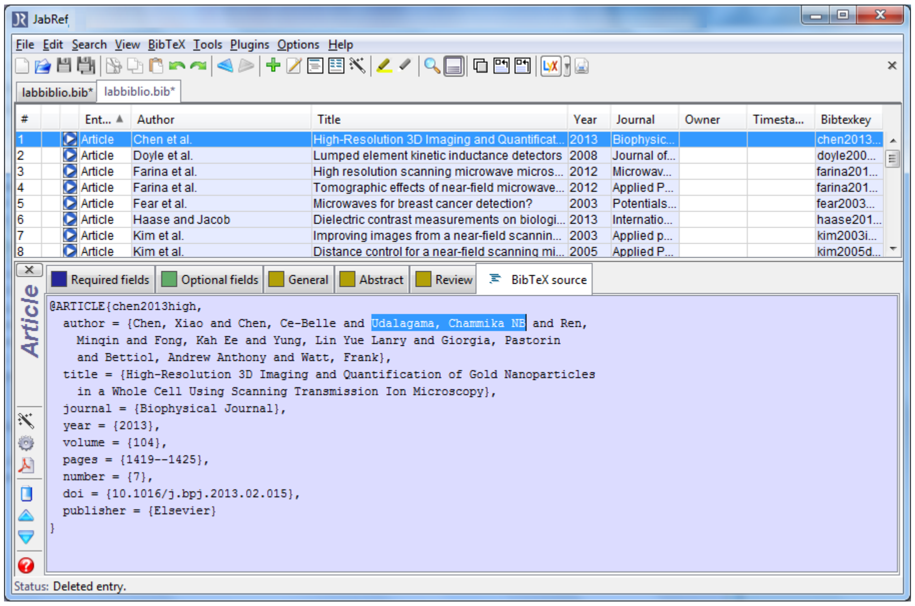


Figure 2.3: JabRef entry window: select your bibliography entry type.

JabRef works with BibTEX (Fig. 2.4) and manual entry (Fig. 2.5), just like BibDesk. The tasks that follow are all analogous to BibDesk, albeit slower. If problems are encountered, JabRef prompts for missing fields, as shown in Fig. 2.6.

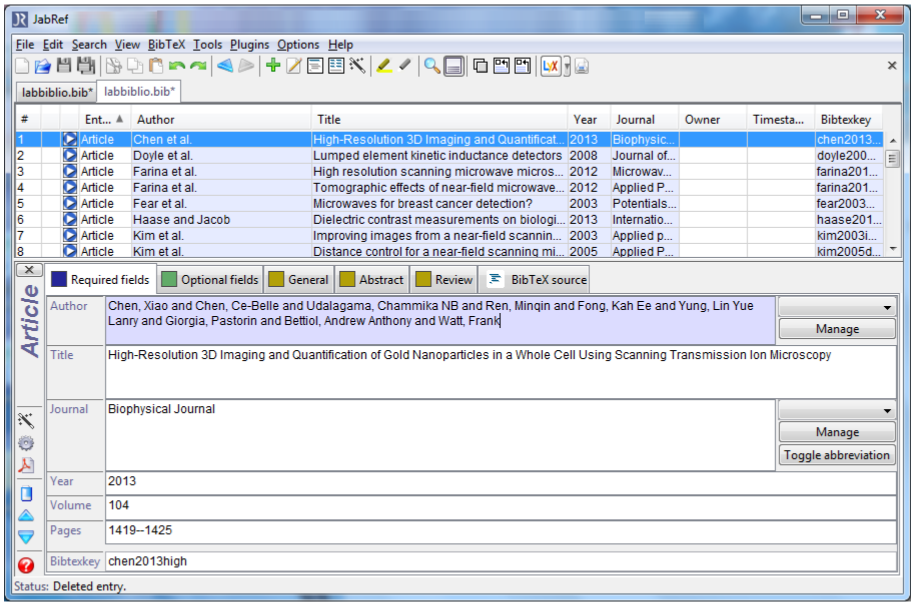

JabRef
File Edit Search View BibTeX Tools Plugins Options Help
labbiblio.bib* labbiblio.bib*
Required fields
Optional fields
General
Abstract
Review
BibTeX source
Article

#	Ent...	Author	Title	Year	Journal	Owner	Timesta...	Bibtexkey
1	Article	Chen et al.	High-Resolution 3D Imaging and Quantificat...	2013	Biophysic...			chen2013...
2	Article	Doyle et al.	Lumped element kinetic inductance detectors	2008	Journal of...			doyle200...
3	Article	Farina et al.	High resolution scanning microwave micros...	2012	Microwav...			farina201...
4	Article	Farina et al.	Tomographic effects of near-field microwave...	2012	Applied P...			farina201...
5	Article	Fear et al.	Microwaves for breast cancer detection?	2003	Potentials...			fear2003...
6	Article	Haase and Jacob	Dielectric contrast measurements on biologi...	2013	Internatio...			haase201...
7	Article	Kim et al.	Improving images from a near-field scannin...	2003	Applied p...			kim2003i...
8	Article	Kim et al.	Distance control for a near-field scanning mi...	2005	Applied P...			kim2005d...

```
@ARTICLE{chen2013high,
  author = {Chen, Xiao and Chen, Ce-Belle and Udalagama, Chammika NB and Ren,
    Minqin and Fong, Kah Ee and Yung, Lin Yue Lanry and Giorgia, Pastorin
    and Bettiol, Andrew Anthony and Watt, Frank},
  title = {High-Resolution 3D Imaging and Quantification of Gold Nanoparticles
    in a Whole Cell Using Scanning Transmission Ion Microscopy},
  journal = {Biophysical Journal},
  year = {2013},
  volume = {104},
  pages = {1419--1425},
  number = {7},
  doi = {10.1016/j.bpj.2013.02.015},
  publisher = {Elsevier}
}
```

Status: Deleted entry.

Figure 2.4: Source code entry. Similar to BibDesk.

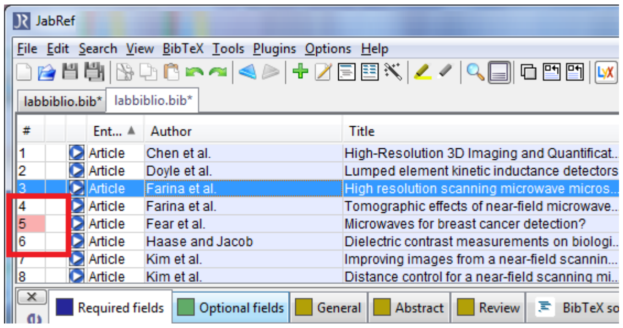


#	Ent...	Author	Title	Year	Journal	Owner	Timesta...	Bibtexkey
1	Article	Chen et al.	High-Resolution 3D Imaging and Quantificat...	2013	Biophysic...			chen2013...
2	Article	Doyle et al.	Lumped element kinetic inductance detectors	2008	Journal of...			doyle200...
3	Article	Farina et al.	High resolution scanning microwave micros...	2012	Microwav...			farina201...
4	Article	Farina et al.	Tomographic effects of near-field microwave...	2012	Applied P...			farina201...
5	Article	Fear et al.	Microwaves for breast cancer detection?	2003	Potentials...			fear2003...
6	Article	Haase and Jacob	Dielectric contrast measurements on biologi...	2013	Internatio...			haase201...
7	Article	Kim et al.	Improving images from a near-field scannin...	2003	Applied p...			kim2003i...
8	Article	Kim et al.	Distance control for a near-field scanning mi...	2005	Applied P...			kim2005d...

Author	Chen, Xiao and Chen, Ce-Belle and Udalagama, Chammika NB and Ren, Minqin and Fong, Kah Ee and Yung, Lin Yue Lanry and Giorgia, Pastorin and Bettiol, Andrew Anthony and Watt, Frank
Title	High-Resolution 3D Imaging and Quantification of Gold Nanoparticles in a Whole Cell Using Scanning Transmission Ion Microscopy
Journal	Biophysical Journal
Year	2013
Volume	104
Pages	1419--1425
Bibtexkey	chen2013high

Status: Deleted entry.

Figure 2.5: Manual entry: what you usually do if you cannot find the BibTEX entry.



#	Ent...	Author	Title
1	Article	Chen et al.	High-Resolution 3D Imaging and Quantificat..
2	Article	Doyle et al.	Lumped element kinetic inductance detectors
3	Article	Farina et al.	High resolution scanning microwave micros..
4	Article	Farina et al.	Tomographic effects of near-field microwave..
5	Article	Fear et al.	Microwaves for breast cancer detection?
6	Article	Haase and Jacob	Dielectric contrast measurements on biologi..
7	Article	Kim et al.	Improving images from a near-field scannin...
8	Article	Kim et al.	Distance control for a near-field scanning mi..

Figure 2.6: A red marking means that there is a problem.

An introductory video to JabRef is given at the following link: http://www.youtube.com.sg/watch?v=f5HadpGMjNw. Of course, if you wonder where JabRef can be downloaded, view http://jabref.sourceforge.net/

2.1.3 Manually Typing out the Bibliography

On some occasions, one might just need to type out a few references for a one-time document (for example, a lab report for Experimental Physics I or II) whereby it is probably not worth the hassle of using a bibliography manager. On other occasions, for example when you need to submit .tex files of manuscripts to journals for publication, you are expected to include the bibliography within the .tex file itself and not in a separate .bib file. This section covers how to type out the bibliography manually.

The main portions involved in doing this are illustrated in the code snippet below:

```
\begin{thebibliography}{99}
       \bibitem{RMP82_3045} Hasan, M.Z. and Kane, C.L., Colloqium :
            Topological insulators, \textit{Rev. Mod. Phys} \textbf
           {82}, 3045 (2010).
       \bibitem{PRL100_096407} Fu, L and Kane, C.L.,
           Superconducting proximity effect and majorana fermions at
            the surface of a topological insulator , \textit{Phys.
           Rev. Lett} \textbf{100}, 096407 (2008). % one of the
           papers that started it all
```

```
        \bibitem{PRL103_107002} Yokoyama, T., Tanaka, Y., and
           Nagaosa, N., Manipulation of the Majorana fermion,
           Andreev reflection, and Josephson current on topological
           insulators, \textit{Phys. Rev. Lett} \textbf{103}, 107002
            (2009).
    % ... (More references)
\end{thebibliography}
```

We start off with a `\begin{thebibliography}{99}` to begin the bibliography environment. This is typically placed near the end of the document **before** the `\end{document}` . The `{99}` in `\begin{thebibliography}{99}` tells LaTeX to set aside space in the numbers of the bibliographic entries for up to 2 digits (i.e. up to entry 99). You probably don't want to type a bibliography with more than 100 entries by hand anyway, but in principle there's nothing stopping you from typing `{999}` or `{9999}` if need be. Each entry in the bibliography is then begun by a `\bibitem{itemlabel}` where the *itemlabel* serves as a label for you to refer to the specific entry in the bibliography in the main text when you need to cite entries in the bibliography. For instance, in order to cite Hasan and Kane's colloqium in the text you would write `\cite{RMP82_3045}` in the text, like the following example:

“ `Topological insulators are very cool \cite{RMP82_3045}.` ”

It is probably not evident in the snippet above, but the bibliography in your final output document appears exactly as you type each of the `\bibitem` entries. You can include formatting directives like `\textbf` and `\textit`, etc. to effect boldfaces and italics in the `\bibitem` entries as required in the citation format you are ladened with.

The use of `thebibliography` environment thus provides a quick way of typing out your bibliography in case you are required to use some strange ordering of author first and last names, journal volume and page numbers with funny combinations of boldfaces and italics that do not fall into one of the standard LaTeX citation formats. The disadvantage is that you are also totally responsible for ensuring that you have typed the entries in the required format correctly[6].

[6]The author cannot be held responsible for the deduction of scores, especially among mentors who are pedantic with issues such as italicised full stops.

As with any other LaTeX environment you need to end it, justifying the `\end{thebibliography}` command in the end of the code snippet above.

2.1.4 APA Citation

A word of caution on APA citation. First, LaTeX does *not* recognise APA citation format as a default setting. If you wish to use a BibTeX method [7], there exists a few methods of going about this.

apa6 Class File

Suppose you need to format the entire document in an APA style. We recommend going for the **apa6** class. This is invoked simply by changing the document class to that of an apa6 one. This is a standard procedure.

apacite Package

Sometimes you need flexibility and perhaps, you only need to ensure that the references are in an APA format. We recommend, then, the use of the **apacite** package. Invoke the package, and also write, as an argument, apacite as the bibliography style. Package documentation is available at `http://texdoc.net/texmf-dist/doc/bibtex/apacite/apacite.pdf`.

2.2 Chapter Exercises

Question 2.1. *1. Compile a simple bibliography using all the methods discussed above.*

2. Find out the command citation formats that LaTeX supports.

If one wants to use MLA, how would it work?

[7] There also exists a BibLaTeX method, but we will not touch on this in this course.

Question 2.2. *Distinguish between "essential" and "optional" fields for a bibliography. Would LaTeX be able to make a distinction and hence prompt you if you miss any essential details? Why?*

1. *However, LaTeX can prompt typographical errors in the bibliography. A common one is the "missing definition". What happens?*
2. *The use of a non-standard material for bibliography entries can cause problems in LaTeX. An example: attempt to cite a webpage using BibDesk or JabRef. Why would LaTeX fail under such cases? Find a remedy.*

Note: For an example of non-standard material, attempt citing a website by using webpage in BibDesk. Done as it is, it should not work properly.

Part II

Using LaTeX to Solve Problems

Chapter 3

Empowering LaTeX: Customisation and Modifications

Packages are customisations inputted in LaTeX that define new commands beyond the standard repertoire. This improves typesetting efficiency, allowing for the typesetter to devote more time to what matters most: the content in whatever you are typing. Two ways will be presented.

3.1 Definitions

(Section 3.1 corresponds to `https://www.youtube.com/watch?v=krDa4Er5nXI` and `https://www.youtube.com/watch?v=WkOMNh7jYTg` in the course.)

(A more detailed explanation exists in "A TeX Primer for Scientists, Chapters 4, 5, 12 and 14.")

`\def` defines a new command. This means that we assign a new command to a set of instructions. These are known as macros. For instance, the slightly cumbersome way of introducing ϵ_0 can be defined with a shorter command as

```
\def\ep{\epsilon_0}
```

which calls ϵ_0 each time `\ep` is typed. Remember to type this before the

`\begin{document}`, since these define how the document will be typeset. This is an example of a substitution. A common example of many substitutions compiled together is the **braket** package that substitutes for manually typing in bras and kets. Just be wary of causing infinite loops[1].

The general syntax is therefore summarised as:

```
\def<new command>{<material to output>}.
```

Two other extremely useful commands:

`\newcommand{}{}` and `\renewcommand{}{}`

. Intuitively speaking, these introduce (or reintroduce) rules for LaTeX to follow when compiling the document. To demonstrate an example of what these do, try the following exercises.

Exercise 32. *Attempt the* `\newcommand{}{}` *and* `\renewcommand{}{}` *in the following exercises and see what you obtain:*

a) In particle physics, the electron, proton and neutron are to be typeset in roman, but the need to indicate signs means the need to introduce new commands. Use `\newcommand{\elec}{\mathrm{e^-}}` *to define a new* `\elec` *command which simply generates* `\mathrm{e^-}`*. Interpret your output and repeat it for the proton* p^+ *and neutron* n^0*. Notice the similarity with a mhchem output using* `\ce`*, while you are at it. Remember to place this in the math environment!*

b) Tell LaTeX to switch the font for an entire file to the sans serif equivalent using `\renewcommand{\familydefault}{\sfdefault}`*.*

c) Google for the solution: changing the way only headings are typeset to sans serif. Would you expect a `\newcommand{}{}` *or* `\renewcommand{}{}`*?*

Note: This exercise is probably the most important exercise as you embark on your LaTeX journey and do more serious work it, so some special mention will be made on quite a few matters on the `\newcommand` *and* `\renewcommand` *syntax and power of it. Some important information before we carry on: These*

[1]Infinite loop errors.

commands as defined in the exercise are in such a way that they admit, by default zero arguments. In that sense, when we define the command, it is standalone.

Exercise 33. *Is there any way you can tell if a command is user-defined or defined as a default?*

Solution: If a command is a default, at least, on TEXMaker, you will see that it prompts you for the command. A user-defined command (through a package or new/renew command) will not behave as such.

Exercise 34. *More mhchem Practice: This exercise is adapted from the mhchem manual, page 9 (as per June 2014).*

More user-defined command practices: We learnt in mhchem, typing `\ce{}` *invokes a chemical equation environment. It can get troublesome at times, so two new definitions often simplify matters: These are:*

1.
   ```
   \newcommand\reaction[1]
   {\begin{equation}\ce{#1}\end{equation}}
   ```

2.
   ```
   \newcommand\reactionnonumber[1]
   {\begin{equation*}\ce{#1}\end{equation*}}
   ```

With these new command definitions, type in a chemical reaction using `\reaction{}`*. What matters get simplified*[2]*?*

Solution: An example is shown using the first command definition, i.e. the `\reaction{}` *one.*

For instance, we could obtain

$$2\,C_2H_2 + 3\,O_2 \longrightarrow 2\,CO_2 + 2\,H_2O \qquad (3.1)$$

by typing `\reaction{2C2H2 + 3O2 -> 2CO2 + 2H2O}`*. As mentioned in class, the* `\newcommand` *feature allows for intuitive naming of commands. We thus want to interpret how this is done.*

[2]You may note that the syntax gets sloppy for both the newcommand and renewcommand. These are acceptable, but we will prefer to keep to proper syntax of enclosing the arguments in braces in most cases. Just for good practice.

We recall the question invoking:
`\newcommand\reaction[1]`
`{\begin{equation}\ce{#1}\end{equation}}`
We define a new command called reaction and demand 1 argument using `[1]`[3]*. Then we define what the command means with the set of braces. The command simply demands to begin equation and wrap the argument marked in* `{#1}` *as the first argument for the* `\reaction` *command. See that commands with multiple arguments, too, can be invoked by the same method, by modifying the number of arguments it requires, and then placing the arguments in the slots specified by* `{#n}`*, where* `n` *is some integer starting from 1. This is in effect, a substitution.*

3.1.1 More on New Commands and Renew Commands

The invoking of these two commands is incredibly *powerful.* Often, it results in a series of definitions unique to a document which can be copied and pasted to different .tex files. A basic introduction of this is already given in Exercise 34.

Simple Substitution

The simplest way of defining a new command is a new definition *without* an argument. For instance:

```
\newcommand{\a}{\alpha}
```

simply defines a new command `\a` as a call for the α symbol in the math environment.

Dealing with Arguments and Options

Most substitutions, however, demand arguments as well. An example is:

[3]You may already notice the lack of braces for reaction in this case. This is normal, and one can verify that quite simply.

```
\newcommand{\bb}[1]{\mathbb{#1}}
```

This defines a new command `\bb` admitting one argument (therefore in options), which has the definition `\mathbb{#1}`, where `#1` refers to the first argument.

Exercise 35. *We now attempt to define some commands in the braket package. In the braket package, an example of a defined command is the* `\Ket` *command, which introduces a ket of this kind:*

$$\left| \frac{1}{2}, -\frac{1}{2} \right\rangle \tag{3.2}$$

Write definitions to generate this command using both `\def` *and* `\newcommand`.

(Hint: The use of symbols that scale with input might be useful. Recall that they are called delimiters.)

Multiple arguments can be defined as well by simply changing the number in options and marking the arguments as `#1`, `#2` and so on. The verification of this is left as an exercise to the reader.

Exercise 36. *Defining a customised argument with options can also be easily done. An example is:*

```
\newcommand{\plusbinomial}[3][2]{(#2 + #3)^#1}
```

which, by attempting some constants such as:

```
\plusbinomial{x}{y}
```

one obtains $(x + y)^2$. *Interpret the syntax.*

Can multiple options be defined?

Exercise 37.
1. *When writing new commands, one uses* `#1`, `#2` *to define various parameters. What part of* ***newcommand*** *or* ***renewcommand*** *will differentiate between options and arguments?*
2. ***renewcommand*** *is used frequently in changing global defaults. One example that was done is to convert a whole document from a sans-*

serif font to a serif font. What does the ***renewcommand*** *command explicitly do, and does order matter on conflicting copies of* ***renewcommand****?*

Note: This idea is extremely important!

Exercise 38. *As practice for defining new commands, we will try one with options. Go to* [*http://tex.stackexchange.com/questions/175828/typeset-dalembertian-operator-in-classicthesis*](http://tex.stackexchange.com/questions/175828/typeset-dalembertian-operator-in-classicthesis)*.*

1. *Define a new command that reproduces the De Alembertian operator. (A somewhat trivial exercise.)*

2. *Define a new command that reproduces an n-th total and n-th partial derivative. The result is shown as a solution, but has no command you could borrow.*

Hint: An argument each is required for the numerator and denominator. However, if $n = 1$, you do not need a power, but if $n > 1$, you add a power, naturally implying an option with a blank default argument (why?).

Solution:

1. *The De Alembertian operator can be defined by:*

```
\newcommand{\dealem}{\Box^2}
```

2. *Since an option is required for $n = 1$, and it is the only argument required, simply write:*

```
\newcommand{\diff}[1][]
{\frac{\mathrm{d}^{#1}y}{\mathrm{d}x^{#1}}}
```

It may be helpful to note that this is a convenient technique to make commands more intuitive to the end-user, though a reminder should be placed in the preamble.

3.1.2 Writing Packages*

Packages are, in simple terms, a bunch of new commands that are assembled in a .sty (style) file. These range from the simple *braket* package which can be downloaded from the Internet to understand the new definitions it introduces. Unfortunately packages can conflict if they share the same definitions. This implies a need to reconfigure the .sty files.

Exercise 39. *Sometimes, packages clash. What are some symptoms of this and how would you know if packages clash with each other?*

Hint: Consider two cases: one with no explicit LaTeX error in the error box, and one with.

Note: It may be worth reading `http://tex.stackexchange.com/questions/655/what-is-the-difference-between-def-and-newcommand` *to understand how packages can clash. Also, you can artificially induce a package clash by defining a command a package has already defined.*

3.2 Compilations*

LaTeX has many compilation tools at one's disposal with the help of a few packages. These include compilations of indices and lists of tables and figures.

3.2.1 Listing Objects

It is often useful to prescribe in some documents, a list of tables and figures with appropriate references. This is *extremely* cumbersome to do in typical word processors because there is simply no way to tag objects. Conveniently, the **labels** assigned to floats used allow for the lists to be generated easily!

These can be done easily by invoking the `\listoffigures` and `\listoftables` commands explicitly. These easily generate a list of figures and tables that you used thanks to the figure and table **floats** introduced.

Exercise 40. *Can a list of equations or exercises for the book be compiled?*

3.2.2 Indices

The **imakeidx** package allows for an alphabetical index to be created at the end of the document. As the name suggests, it is a collection of terms used in a document. This is useful for books and is simple to implement.

An index can be created by invoking:

```
\makeindex[columns=3, title=Alphabetical Index, intoc]
```

which means three columns for the index, titling of the index as "Alphabetical Index" and "intoc" forces generation of the index in the table of contents.

Marking terms for the index is done by adding `\index{<term>}` after the term to be indexed. The term will not be typeset; the command merely labels the point to index the word.

Nesting is also possible and may be useful in describing variants of something. This is done by keying `\index{<category>!<term itself>}`. For instance, `\index{virus!flu}` will label an index entry that places flu as a particular type of virus in the index. If virus was mentioned earlier in the text, the index package points it to the first-mentioned point, and the sub-entry reflects the first-mentioned point of the sub-entry itself.

Like the table of contents, a print command must be placed at the end for the generation of the index. This is done by `\printindex`.

Exercise 41. *(Out of the scope of the course, but worth trying.): In contract documents, many terms are often redefined, calling for the use of a glossary of terms used. Google how this is done.*

3.2.3 Managing and Compiling Multiple Pieces of Work

Often group work calls for a separation of tasks and therefore different people working on different sections of a report. This implies, often, the need to adopt a modular approach, where different people work on different files, before putting them together.

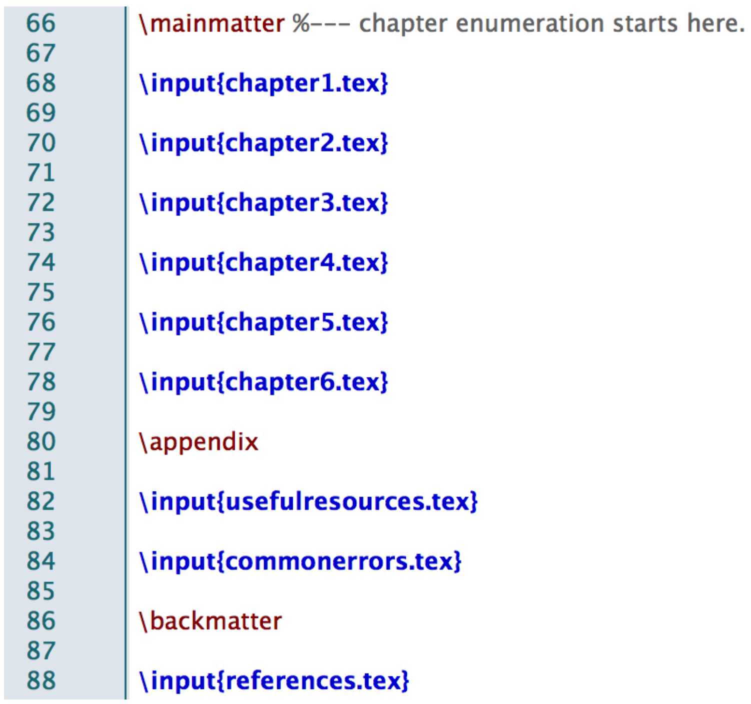
With word processors, this often causes difficulty due to the lack of standardisation of formats and conventions. LaTeX beautifully settles this for you through a few methods.

Simple Input of Files

(Subsubsections 3.2.3 and 3.2.3 correspond to `https://www.youtube.com/watch?v=6L8H23KD9V8` in the course.)

As a first example, consider a report with multiple chapters. Chapters can be split by the use of an `\input{}` command.

An example is shown in this "launcher" in Fig. 3.1.



```
\mainmatter %--- chapter enumeration starts here.

\input{chapter1.tex}

\input{chapter2.tex}

\input{chapter3.tex}

\input{chapter4.tex}

\input{chapter5.tex}

\input{chapter6.tex}

\appendix

\input{usefulresources.tex}

\input{commonerrors.tex}

\backmatter

\input{references.tex}
```

Figure 3.1: Writing a series of **inputs**.

The `\input{}` command places the contents of the file, verbatim, into the .tex file that calls for the input. In the case of Fig. 3.1, the constituents were written in separate .tex files (text files) and inputted (as text) into the main document. In other words, one can:

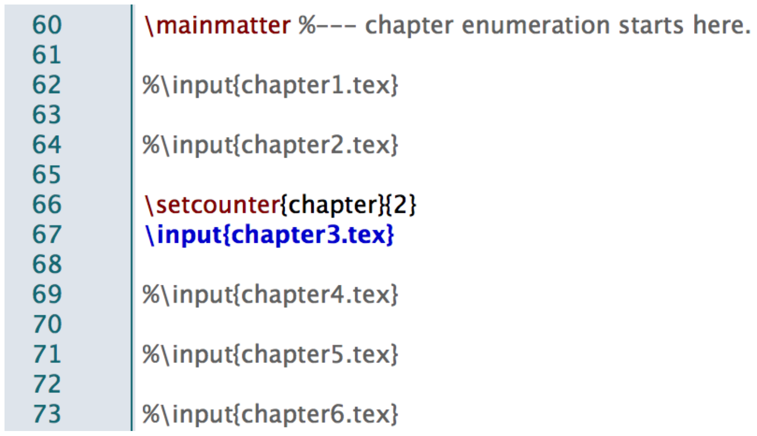
1. Write the main "launcher" that includes the preamble.
2. Write down a list of `\input{}` for the various chapter names.
3. Write only the body of the chapters (not the premable!) of the various chapters in their separate .tex files.

The significance behind such a modular approach is this:

- Different people work on entirely different files, which removes the risk of "messing other sections up".
- Each constituent .tex file is shorter, making the finding of errors significantly easier.

- One can comment various sections of the file out to suppress its generation. (Some work will need to be done to correct the numbering if it runs afoul of what you intend, but that is simple to do.)

An example of this is shown in Fig. 3.2.



```
\mainmatter %--- chapter enumeration starts here.

%\input{chapter1.tex}

%\input{chapter2.tex}

\setcounter{chapter}{2}
\input{chapter3.tex}

%\input{chapter4.tex}

%\input{chapter5.tex}

%\input{chapter6.tex}
```

Figure 3.2: Suppressing all other inputs, and resetting the "0th" count to 2 means the first count will be 3.

Exercise 42. *Write a simple problem set .tex file and solution .tex file which refers to the problem set. Compile both a problem set file and a complete file containing both problems and solutions.*

Include Versus Input

There is a similar command called **include**. Unlike **input**, each individual .tex file called upon by **include** generates its own .aux file that contains its list of references and details about what LaTeX should do to generate them. The differences are quite stark, and as such this will be left as a guided exercise in Exercise 43.

Exercise 43. *Unlike **input**, **include** allows for individual generation of .aux files and other assorted material for the chapters. That also means that the way of preparing such files is slightly different.*

1. *What are the differences between* ***input*** *and* ***include*** *in terms of how they read imported .tex files?*

2. *How should they be used?*

Note: An even more detailed method of collaboration can be viewed at `https://www.sharelatex.com/learn/Multi-file_LaTeX_projects`*.*

Exercise 44. *The* ***input*** *and* ***include*** *commands are not the only ones that allow for the insertion of material from another file. Check how* ***import*** *works and compare it with* ***input*** *and* ***include****.*

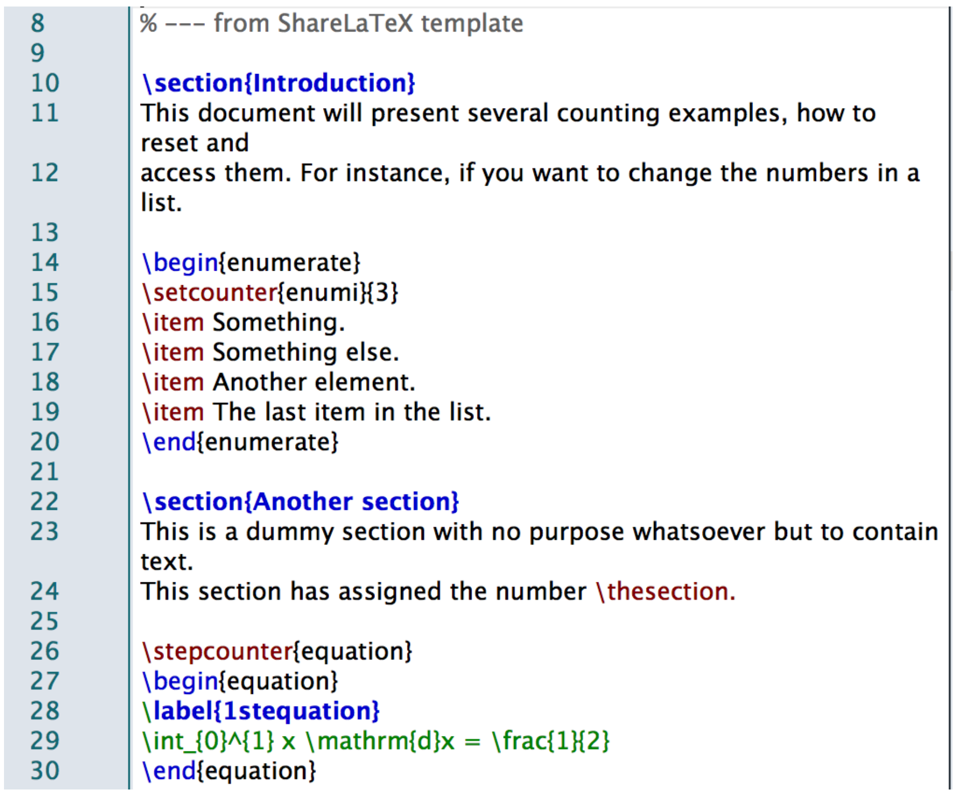
Getting started: `https://www.sharelatex.com/learn/Management_in_a_large_project`*.*

Setting, Resetting Counters

(Subsubsection 3.2.3 corresponds to `https://www.youtube.com/watch?v=nHCjWzh9048` in the course.)

In the introductory chapters, counters were introduced in the form of counting of environment boxes, pages and enumeration. You have also seen in Fig. 3.2 on how a counter can be set to a different default.

Two examples of counter manipulation are shown in Fig. 3.3.



```
% --- from ShareLaTeX template

\section{Introduction}
This document will present several counting examples, how to
reset and
access them. For instance, if you want to change the numbers in a
list.

\begin{enumerate}
\setcounter{enumi}{3}
\item Something.
\item Something else.
\item Another element.
\item The last item in the list.
\end{enumerate}

\section{Another section}
This is a dummy section with no purpose whatsoever but to contain
text.
This section has assigned the number \thesection.

\stepcounter{equation}
\begin{equation}
\label{1stequation}
\int_{0}^{1} x \mathrm{d}x = \frac{1}{2}
\end{equation}
```

Figure 3.3: From ShareLaTeX: Examples of counter manipulation in an enumerated list and an equation list.

In the case of Fig. 3.3, the enumerated counter was forced to start from 4, whereas the equation counter was increased by 1. Note that counters exist for various objects, such as enumerated lists (called by **enumi** to **enumiv** for extent of nesting of enumerated list), equations (called by **equation**) and chapter (called by **chapter**) .

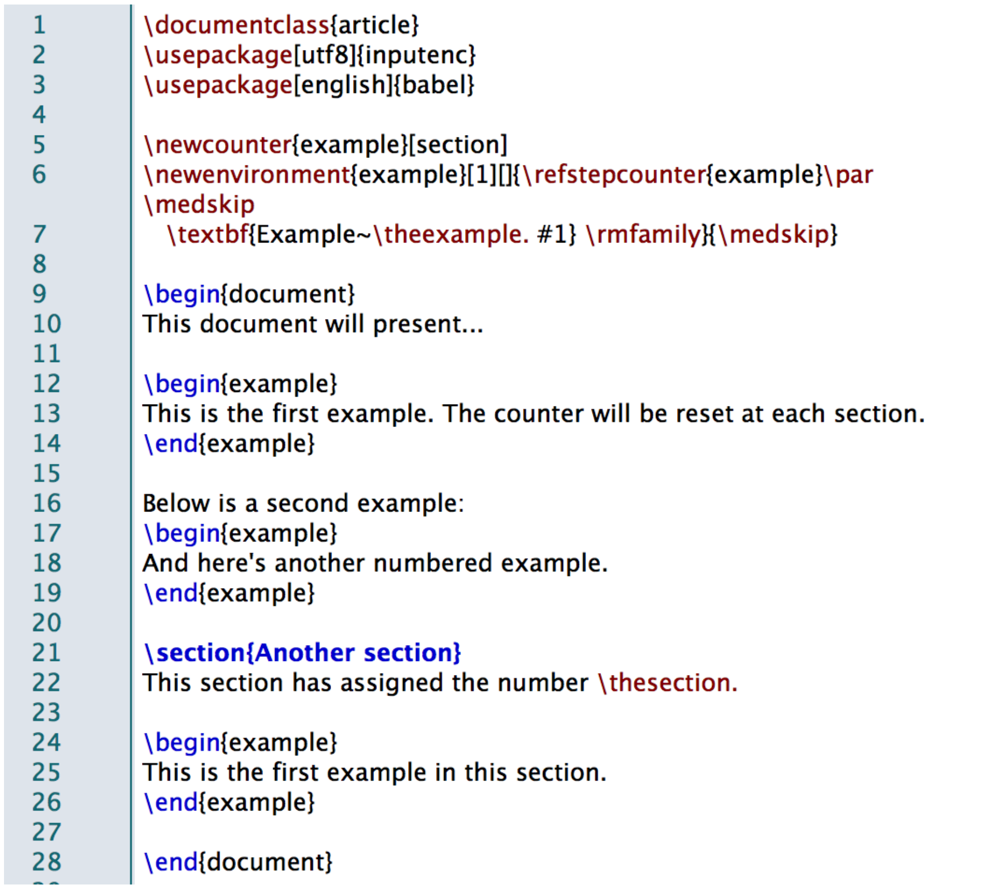
Exercise 45. *Investigate counters for each of the following and determine how they should be set, reset and stepped up/down.*

- *Section*
- *Page*
- *Figures (and by extension, tables)*

- *Exercise numbers of this book and question numbers. (Hint: You will have find the **amsthm** package first.)*

All of these modifications need to be compiled twice. What is LaTeX doing in each compilation?

New counters can be set as well. Fig. 3.4 gives an example.



```latex
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[english]{babel}

\newcounter{example}[section]
\newenvironment{example}[1][]{\refstepcounter{example}\par
\medskip
  \textbf{Example~\theexample. #1} \rmfamily}{\medskip}

\begin{document}
This document will present...

\begin{example}
This is the first example. The counter will be reset at each section.
\end{example}

Below is a second example:
\begin{example}
And here's another numbered example.
\end{example}

\section{Another section}
This section has assigned the number \thesection.

\begin{example}
This is the first example in this section.
\end{example}

\end{document}
```

Figure 3.4: From ShareLaTeX: Defining both a new counter as well as a new example environment.

The new command `\newcounter{example}[section]` defines a new "example" command that resets at the start of each section. Of course, leaving the option blank means the example counter will not reset unless being manually forced to.

The definition in the new environment contains `\refstepcounter{example}` which refers to the step counter in the example. This jumps the counter by one. `\theexample` should be obvious enough to mean: print the example number.

Exercise 46. 1. *Typeset Fig. 3.4. In particular, pay attention to the syntax of* `\newenvironment{}[][]{}{}`.

2. *A variant of* ***newcommand*** *was also introduced subtly in this example. What other new objects can be defined?*

Note: It might be of interest that, just as there are new objects that can be defined, old objects such as environments can, too, be redefined analogously to commands.

3.3 Conclusion

LaTeX can do things that are rather unimaginable to the untrained person. Once exposed to LaTeX however, many typesetting mysteries become clear. Examples include drawing, typing your own C.V. or even publishing a book! You should explore what LaTeX can do after the course; you will be pleasantly surprised that it is not only an academic typesetter. Some of these will be discussed in the exercises that follow.

3.4 Chapter Exercises

Note: Only Questions 3.1, 3.2 and 3.3 are essential.

Question 3.1. *Examine the* ***sectsty*** *package to investigate how it can change the default font of particular parts of the document only. This is in contrast to the example in Exercise 32 where the* ***renewcommand*** *changes the font of the entire document.*

Question 3.2. *Attempt the following modifications to your LaTeX document: you intend to call out the expression $[l_1, l_2] = i\hbar l_3$ repeatedly (because you could be typing a textbook). By the typing of a single command, that in-line expression is called out.*

Question 3.3. *Additional Information on Macro Definitions (Adapted from http://www.sharelatex.com/learn/Defining_your_own_commands)*

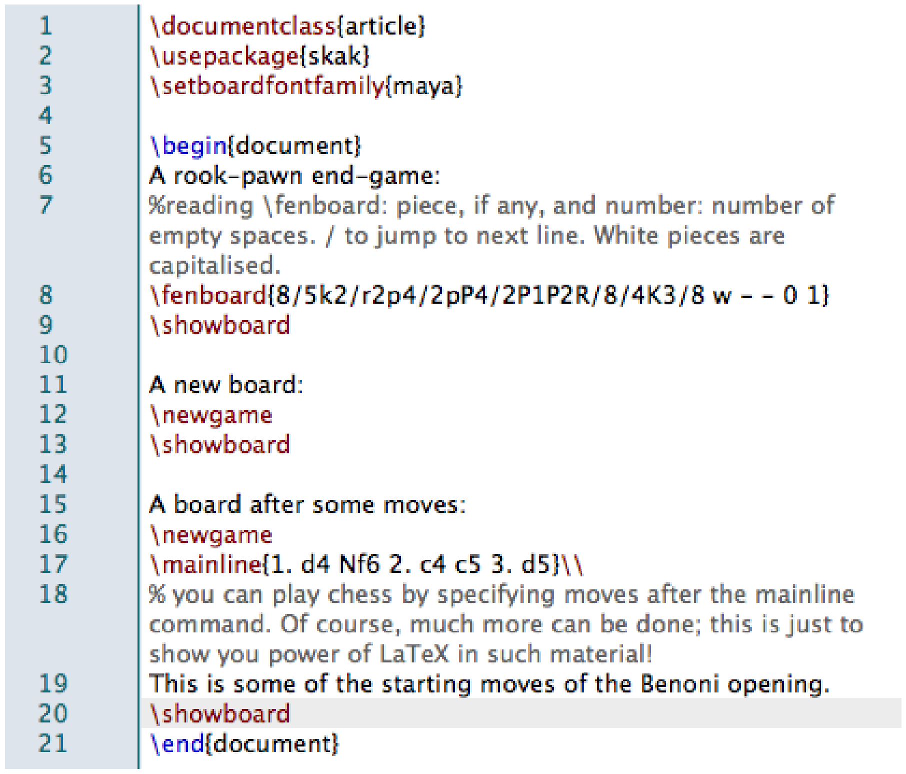
The `\newcommand{}{}` *argument can take optional arguments in between. Try the following:* `\newcommand{\taylorexpansion}[3][1]{(#1 + #2)^#3}`*. What does this mean? Attempt to type:* `$\taylorexpansion{x}{-1}$`*. Figure out what happens.*

Note: The URL provided is a good source on understanding the section with many more examples.

Question 3.4. **Exploration on LaTeX Capabilities*

(Read Chapters 10.1 and 10.7 of "The LaTeX Graphics Companion, 2nd Edition" for a more complete account of typing music scores. Try only when you have time; this is more of an fun exercise.)

Chess: There exists a ***skak*** *package which basically typesets chess games by intermixing game detail with commentary. For starters, let us define a board for a typical rook-pawn endgame. We type the following as in Fig. 3.5:*



```
\documentclass{article}
\usepackage{skak}
\setboardfontfamily{maya}

\begin{document}
A rook-pawn end-game:
%reading \fenboard: piece, if any, and number: number of
empty spaces. / to jump to next line. White pieces are
capitalised.
\fenboard{8/5k2/r2p4/2pP4/2P1P2R/8/4K3/8 w - - 0 1}
\showboard

A new board:
\newgame
\showboard

A board after some moves:
\newgame
\mainline{1. d4 Nf6 2. c4 c5 3. d5}\\
% you can play chess by specifying moves after the mainline
command. Of course, much more can be done; this is just to
show you power of LaTeX in such material!
This is some of the starting moves of the Benoni opening.
\showboard
\end{document}
```

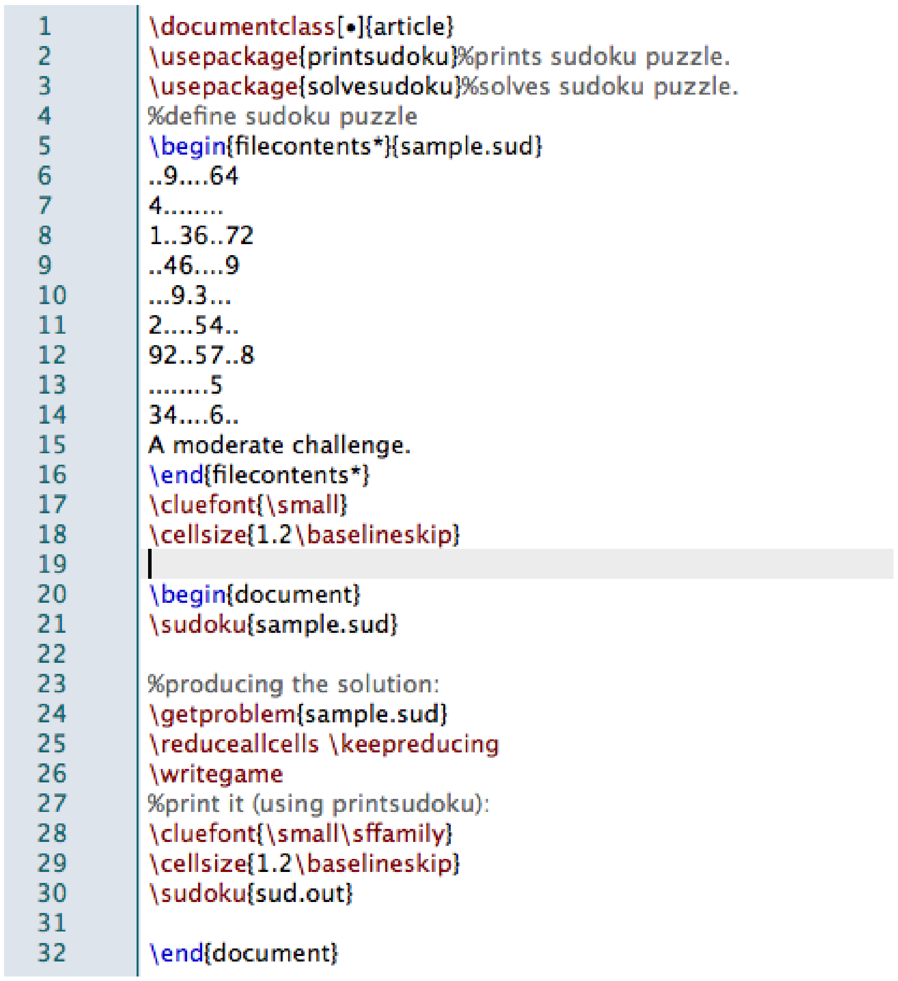
Figure 3.5: Some short code for a simple chess typeset.

1. *Typeset Fig. 3.5. Interpret the results you get. Attempt to play a simple game of chess using the third board.*

2. *The* ***texmate*** *package replaced the* ***skak*** *package eventually. Figure out what it can do.*

   Note: LATEX can import chess game files (.PGN) and typeset them. If you play chess, you might want to explore this functionality.

3. *c) LATEX is not only a typesetter: it also can solve puzzles. A very simple example is shown in Fig. 3.6.*



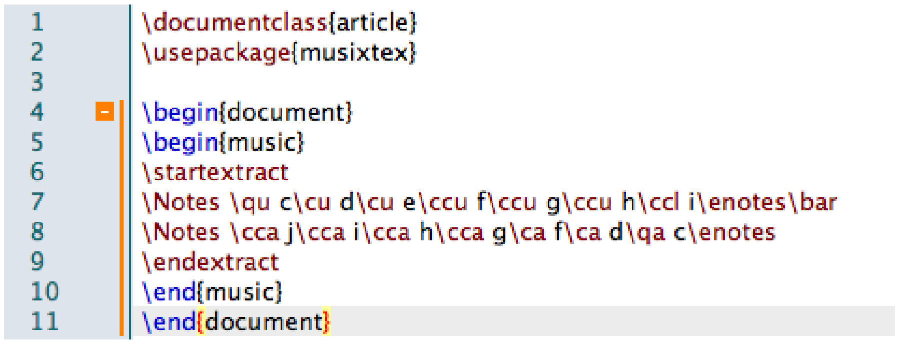
```
\documentclass[•]{article}
\usepackage{printsudoku}%prints sudoku puzzle.
\usepackage{solvesudoku}%solves sudoku puzzle.
%define sudoku puzzle
\begin{filecontents*}{sample.sud}
..9....64
4........
1..36..72
..46....9
...9.3...
2....54..
92..57..8
........5
34....6..
A moderate challenge.
\end{filecontents*}
\cluefont{\small}
\cellsize{1.2\baselineskip}

\begin{document}
\sudoku{sample.sud}

%producing the solution:
\getproblem{sample.sud}
\reduceallcells \keepreducing
\writegame
%print it (using printsudoku):
\cluefont{\small\sffamily}
\cellsize{1.2\baselineskip}
\sudoku{sud.out}

\end{document}
```

Figure 3.6: Simple Sudoku code.

Typeset the following your editor and interpret the results. Go to your favourite Sudoku puzzle book and attempt to get LaTeX to solve it.

Note: There exists software (or your friend(s)) to help solve Sudoku puzzles. However, there exists a wealth of literature to get LaTeX to typeset many things that are otherwise inconvenient to go around on a processor. Why are these skills still relevant?

Question 3.5. *Some trivia: LaTeX can typeset music too. Explore the* ***musixtex*** *package! A simple example is in Fig. 3.7. Typeset it and comment on your findings. You could read up more and produce your own music scores.*



```
\documentclass{article}
\usepackage{musixtex}

\begin{document}
\begin{music}
\startextract
\Notes \qu c\cu d\cu e\ccu f\ccu g\ccu h\ccl i\enotes\bar
\Notes \cca j\cca i\cca h\cca g\ca f\ca d\qa c\enotes
\endextract
\end{music}
\end{document}
```

Figure 3.7: A simple typeset music score.

Question 3.6. *LaTeX can type almost anything. There exists plenty of packages as well as integration between LaTeX and other software (e.g. Mathematica). Attempt to figure out how they are compatible how they could help. (As an exercise, have a look at programs you already have and look for things along the lines of export .tex, or commands to that effect.)*

For example, Geogebra exports TikZ code to LaTeX. TikZ code will be explored later in the course.

Question 3.7. *Even within seemingly standard material, there are many ways of introducing variations. An example is in the table of contents. Typing* `\tableofcontents` *automatically generates a list of chapters, sections and subsections in a document. Google on how:*

1. *One can introduce new items into the table of contents.*

2. *One can rename the title of the table of contents.*

Hint: ShareLaTeX has an article on manipulating the table of contents.

Question 3.8. *Of interest in the later part of your life (if you stay in academia or switch to teaching) is teaching your own module and setting examination problems. A template to do this is through the exam* ***class****. A link is available at* `https://www.sharelatex.com/learn/Typing_exams_in_LaTeX`*. Explore what it is capable of doing.*

Question 3.9. *(Optional problem.): Google what it means by the* ***sloppy*** *command in LaTeX. This is of some use in breaking long URLs, for example. Why is this of use?*

Chapter 4

Cloud Solutions

The cloud has become a primary collaborative and storage platform among many. LaTeX is no exception to being involved on the cloud. There exists many solutions online to help people with LaTeX collaborate real-time and obtain ideas to solve problems that may be encountered in the course of using LaTeX. We will briefly introduce the reader to interfaces on Overleaf[1] and ShareLaTeX as a resource repository and collaboration platform. In addition, the reader should also be comfortable with the idea of searching and modifying extensive pieces of mark-up language from others as a starting point to create new content.

4.1 Introduction to TeX Communities

(Section 4.1 corresponds to `https://www.youtube.com/watch?v=_mSYZizN5a0` in the course.)

Any good program will have its following. These exist as communities such as the TeX stackexchange (at `http://tex.stackexchange.com/`). There exist many LaTeX communities, both online and offline, which will be able to help with various LaTeX needs. After all, scientists work collaboratively.

[1]Formerly known as WriteLaTeX, as of the time the course was taught.

4.2 Collaboration Solutions

(Sections 4.2 and 4.4 correspond to https://www.youtube.com/watch?v=2uiaUCJ53_o in the course. Also, a preview to using ShareLaTeX while answering a few problems was done in https://www.youtube.com/watch?v=6LUNkivf0mk.)

Overleaf and ShareLaTeX are two examples of collaboration solutions. They extend beyond collaborative solutions and offer tutorials and templates for many types of documents[2].

Templates exist for a wide variety of projects, including books, thesis submissions, business cards, CV and resume templates. One might also notice **Beamer**. Beamer solutions are a special class of solutions which comprise of Beamer slides and Beamer posters. The same goes with drawing solutions in the form of **TikZ**.

As further incentives to move towards LaTeX rather than to compromise with some of your Word-loving friends[3], ShareLaTeX and Overleaf are both free[4]. Overleaf features live compilation, which can be significantly advantageous in projects, where you want to see the compilation almost immediately. ShareLaTeX has more templates as well as command suggestions. We recommend at least trying both cloud services out for free, at the least.

Exercise 47. *Use either cloud service to obtain a copy of a report template that allows for an inclusion of a university logo on the cover page, multiple authors and affiliations for individual authors. This will be of use to report-writing, such as SP2171.*

Exercise 48. *Investigate the cloud services for the following features:*

- *Chat service.*
- *History.*

[2]A notable exception is an NUS FYP template. Maybe someone can do one and upload it.

[3]The author cannot be held responsible for strained friendships.

[4]Basic account, but it's good enough.

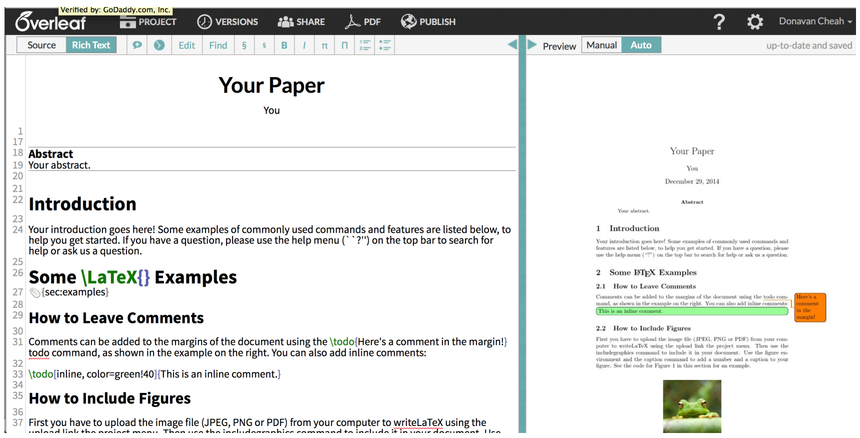
- *Types of templates.*
- *Support for supplementary files.*

What features are essential and why?

Exercise 49. *(For the **bravest** only, and for those who have prior experience in LaTeX.): Using a TeX compiler such as TeXmaker allows for compilation that generates either a PDF, DVI or PS output. Can the same be done on the cloud?*

4.3 Intermediate Interfaces

It might be initially intimidiating for a non-LaTeX user to be introduced to the typesetting style of writing. Overleaf is a good place to start: it allows for the user to interchange between source and rich text. As mentioned in Chapter 1, the WYSIWYG interface is what most are still most familiar with. On Overleaf, one can see both interfaces through the use of interchanging between source and rich text as shown in Fig. 4.1 and 4.2.



```
Your Paper
You

Abstract
Your abstract.

Introduction

Your introduction goes here! Some examples of commonly used commands and features are listed below, to
help you get started. If you have a question, please use the help menu (``?'') on the top bar to search for
help or ask us a question.

Some \LaTeX{} Examples
{sec:examples}

How to Leave Comments

Comments can be added to the margins of the document using the \todo{Here's a comment in the margin!}
todo command, as shown in the example on the right. You can also add inline comments:

\todo[inline, color=green!40]{This is an inline comment.}

How to Include Figures

First you have to upload the image file (JPEG, PNG or PDF) from your computer to writeLaTeX using the
```

Your Paper

You

December 29, 2014

Abstract

Your abstract.

1 Introduction

Your introduction goes here! Some examples of commonly used commands and features are listed below, to help you get started. If you have a question, please use the help menu ("?") on the top bar to search for help or ask us a question.

2 Some LaTeX Examples

2.1 How to Leave Comments

Comments can be added to the margins of the document using the todo command, as shown in the example on the right. You can also add inline comments:

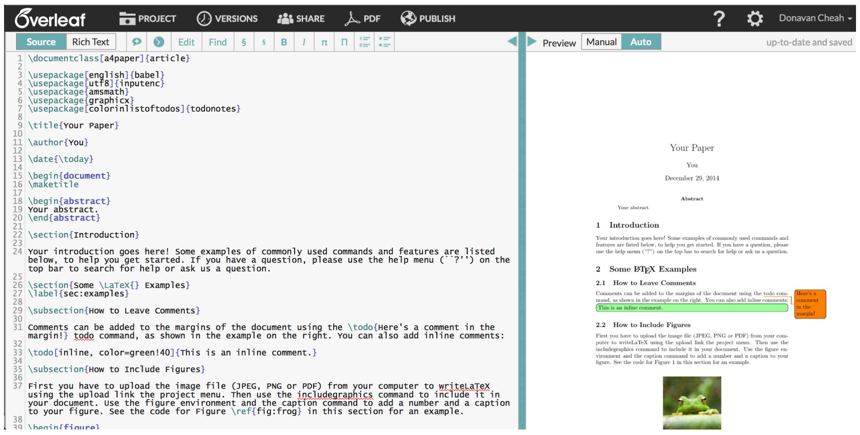
This is an inline comment.

Here's a comment in the margin!

2.2 How to Include Figures

First you have to upload the image file (JPEG, PNG or PDF) from your computer to writeLaTeX using the upload link the project menu. Then use the includegraphics command to include it in your document. Use the figure environment and the caption command to add a number and a caption to your figure. See the code for Figure 1 in this section for an example.

Figure 4.1: Showing how the WYSIWYG interface shows up in Overleaf through the selection of the the rich text option. Parts can be clicked to show the source code responsible for the generation of said rich text.



```
\documentclass[a4paper]{article}

\usepackage[english]{babel}
\usepackage[utf8]{inputenc}
\usepackage{amsmath}
\usepackage{graphicx}
\usepackage[colorinlistoftodos]{todonotes}

\title{Your Paper}

\author{You}

\date{\today}

\begin{document}
\maketitle

\begin{abstract}
Your abstract.
\end{abstract}

\section{Introduction}

Your introduction goes here! Some examples of commonly used commands and features are listed
below, to help you get started. If you have a question, please use the help menu (``?'') on the
top bar to search for help or ask us a question.

\section{Some \LaTeX{} Examples}
\label{sec:examples}

\subsection{How to Leave Comments}

Comments can be added to the margins of the document using the \todo{Here's a comment in the
margin!} todo command, as shown in the example on the right. You can also add inline comments:

\todo[inline, color=green!40]{This is an inline comment.}

\subsection{How to Include Figures}

First you have to upload the image file (JPEG, PNG or PDF) from your computer to writeLaTeX
using the upload link the project menu. Then use the includegraphics command to include it in
your document. Use the figure environment and the caption command to add a number and a caption
to your figure. See the code for Figure \ref{fig:frog} in this section for an example.

\begin{figure}
```

Your Paper

You

December 29, 2014

Abstract

Your abstract.

1 Introduction

Your introduction goes here! Some examples of commonly used commands and features are listed below, to help you get started. If you have a question, please use the help menu ("?") on the top bar to search for help or ask us a question.

2 Some LaTeX Examples

2.1 How to Leave Comments

Comments can be added to the margins of the document using the todo command, as shown in the example on the right. You can also add inline comments:

This is an inline comment.

Here's a comment in the margin!

2.2 How to Include Figures

First you have to upload the image file (JPEG, PNG or PDF) from your computer to writeLaTeX using the upload link the project menu. Then use the includegraphics command to include it in your document. Use the figure environment and the caption command to add a number and a caption to your figure. See the code for Figure 1 in this section for an example.

Figure 4.2: Showing how the WYSIWYG interface shows up in Overleaf through the selection of the the source option.

4.3.1 LyX

Another typical compromise between a WYSIWYG interface and a typesetter is a free program called LyX. This can be downloaded through http://www.lyx.org/Download. This offers a WYSIWYG interface just like Overleaf[5].

LyX offers the functionality of showing you the document output just like a real-time update as in Overleaf. Moreover, its WYSIWYG interface provides a less sharp learning curve for the non-initiate who is often intimidated at the lack of an output to constantly cross-refer. All commands that are standard in LaTeX apply to LyX as well[6].

[5]You might ask why the course did not start from introducing LyX since it would seem like a less steep learning curve. LyX is certainly a gentler learning curve, but it involves taking "two steps": one to understand syntax and one to understand structure. Instead, this course takes on a more programming-like approach, starting straightaway from the thick of action assuming that the reader already has some background on syntax. In the case of the course, it assumes the literacy in Mathematica, which was done in the freshmen year in SPS.

[6]At this point I should add a disclaimer that this assumes that you use LyX as a typesetter for a single .tex file. Typically, the modular approach of many large documents imply that editing individual .tex files would be impossible on LyX to see because documents will not generate in individual .tex files. This has to do the organisation of your work more than a LyX limitation: if you work on separate .tex files and frequently

4.4 Interpreting Templates

One key issue with making use of templates from others is interpreting them. This is the case with many highly-defined and highly-customised templates where a writer writes custom packages and classes for himself or herself. This is the purpose of exposure to the cloud: getting to know what other LaTeX-related files there are which are of use.

4.4.1 Class Files

(Mostly adopted from: `https://www.sharelatex.com/blog/2011/03/27/how-to-write-a-latex-class-file-and-design-your-own-cv.html`)

Recall the use of the **document class** from earlier. A document class has, with it, styling defaults. In this case, it takes after its styling from the **article.cls** file. The .cls extension denotes a class. Therefore, there exists non-standard classes, provided that these .cls files are first available to be called out as well.

Standard classes can be found at `http://tex.stackexchange.com/questions/782/what-are-the-available-documentclass-types-and-their-uses`.

Exercise 50. *Search for customised document clases. Why are such non-standard classes deployed?*

4.4.2 Style Files

There also exists style files with the extension .sty. These were first introduced with packages. These provide customisations to your document so that you do not need to retype definitions repeatedly. These .sty files are

introduce new, non-standard notations, it may often be best to just look at source code. Moreover the source code interface gives you a more familiar interface to work with should you switch to any programming language, which is quite important in the training of a scientist, engineer, programmer or even the casual web designer.

invoked using `\usepackage[...]{...}` used much earlier.

Finer points are highlighted in `http://tex.stackexchange.com/questions/91167/why-use-sty-files` where the comparison between .tex and .sty files is made more explicit.

The design of one's own .cls or .sty file is beyond the scope of the course, but one can start by writing customised definitions and then saving it as a package which can be inputted into TEX for future use. However, there exists a fine line between extensive customisation and collaboration; additional customisation deviates from the community standards that someone who uses TEX is expected to know. This is particularly problematic in collaborations where you may have defined short-hand that is different from the community, or in some cases, even non-existent. In these cases it is essential to inform your reader or collaborator of the deviations from the norm and provide some instructions on compilng the .tex file properly.

Exercise 51. *Go to CTAN and download the .sty file of a custom package of your choice. (A simple package is the* ***braket*** *package.) Comment on your observations.*

Exercise 52. *Sometimes, a package that may need to be used is not in the list of standard packages. Describe how you would download it and place into your TEX distribution. Also explain why this fails for the cloud.*

(For the brave.) How would you rectify this issue?

Exercise 53. *Journals have different formats and conventions they demand researchers to follow for submission. Find, on ShareLATEX, a journal template to experiment on.*

An example you could preview because of learning value: `https://www.sharelatex.com/templates/journals/aps`.

4.4.3 Lessons on the Cloud

The cloud (especially ShareLaTeX) has a rich repository of material that introduces the slightly initiated to many capabilities in LaTeX. Templates are provided for the eager learner to play with on ShareLaTeX.

4.5 Chapter Exercises

Question 4.1. *Use a cloud system to obtain a template for writing a thesis. Ideally it should contain features such as header and footer, the manipulation of graphics, and other features. Why is it advantageous to write, in some cases, customised .sty files?*

Hint: You should Google solutions, or make use of help such as the LaTeX wikibook.

Question 4.2. *Gather a few friends (if you have any) and collaborate on ShareLaTeX. Find the parallels of these features on ShareLaTeX and determine if these are pay features or free ones:*

1. *Chat*
2. *Checking of version history*
3. *Exporting of files as a .zip (why is just a .pdf insufficient?)*

It may also be worthwhile to compare the differences between ShareLaTeX and WriteLaTeX in terms of functionality.

Question 4.3. *In Microsoft Word, there exists .dot files. What are their equivalents in LaTeX?*

Question 4.4. *A common problem in collaborative tools is the issue of conflicting copies. That is, different people work on the same document and hence result in potential issues when the document is being compiled together (typically through a save feature).*

Discuss how this problem can be entirely avoided during collaboration on

LaTeX.

Hint: Recall file splitting and inputting.

4.6 A Note to the Reader

At this point in time, the reader should be fairly acquainted with the workings of LaTeX to understand the rudiments. A typical 2-session course can end here, and leave the rest of the book as an exploratory journey for the reader. However, some of the applications of LaTeX might come across as slightly unexpected to the reader, so it is essential to explore the cloud to see what it has to offer!

The later chapters expose the reader to applications of LaTeX which would not be that obvious to the reader. It might come across as a surprise, but LaTeX has great versatility and is capable of solving a wide variety of problems. Enjoy reading!

Part III

Introduction to Some Applications

Chapter 5

Beamer*

(Chapter 5 corresponds to `https://www.youtube.com/watch?v=FrDOzsizHcs` in the course.)

(Note also a change in the screenshot colour templates; these are due to the use of a different TEX editor and are cosmetic, i.e. the functionality of the commands are unchanged.)

Beamer is a document class for slides. What advantage(s) does Beamer confer? Use Beamer if:

1. You want to deliver material in a clear manner, which does not demand animation.
2. You want to deal with significant mathematical material.
3. You just cannot seem to get object placement right.
4. You are short on time, and you need to piece everything together coherently and quickly.

However, slide preparation differs from document preparation. Beamer works on the same premise as document preparation on LATEX simply because it is a LATEX-derived system. Depending on what you need, you may want to be acquainted with the best tool for the job. Nevertheless, this secton aims to show common applications in slides and how Beamer deals with them well .

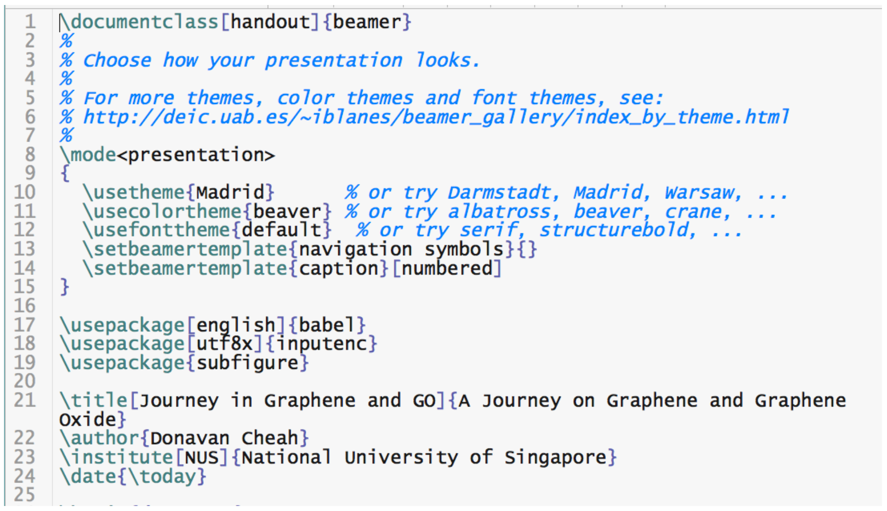
Sample templates are available on WriteLaTeX and ShareLaTeX for free use. Using them as a start to the Beamer journey is advisable; this is analogous to the use of slide templates in a presentation tool.

Note: Beamer can be used to do posters as well! However, this is slightly beyond the scope of the course. However, many templates for posters exist, and they can be complicated especially if they deviate significantly from the norm due to the need to set many default settings.

Exercise 54. *Previously, there used to exist a* ***slides*** *document class. However, this has been superceded by Beamer. Google the output of the* ***slides*** *class and the Beamer class and suggest why this might be so.*

5.1 Control on Beamer

Typically the first control you will want to have is the type of slides we want. This is easily done in a few ways as shown in Fig. 5.1



```
\documentclass[handout]{beamer}
%
% Choose how your presentation looks.
%
% For more themes, color themes and font themes, see:
% http://deic.uab.es/~iblanes/beamer_gallery/index_by_theme.html
%
\mode<presentation>
{
  \usetheme{Madrid}      % or try Darmstadt, Madrid, Warsaw, ...
  \usecolortheme{beaver} % or try albatross, beaver, crane, ...
  \usefonttheme{default}  % or try serif, structurebold, ...
  \setbeamertemplate{navigation symbols}{}
  \setbeamertemplate{caption}[numbered]
}

\usepackage[english]{babel}
\usepackage[utf8x]{inputenc}
\usepackage{subfigure}

\title[Journey in Graphene and GO]{A Journey on Graphene and Graphene Oxide}
\author{Donavan Cheah}
\institute[NUS]{National University of Singapore}
\date{\today}
```

Figure 5.1: A typical preamble for a Beamer set of slides taken from a WriteLaTeX template.

The first modifiable property is the document class [1]. Note the use of the **beamer** document class. For starters, since slides build up, there is an option to print either handouts (as noticed here) or print builds. (A print build is a print-out showing clearly every step of the presentation progression. This will become clearer later.)

Next, we move to the theme. This set of slides uses the Madrid theme. Interestingly, themes are named after major cities. Besides just slide themes, there are also colour and font themes. These are done the same way as slide themes. You could try different themes and find a thematic scheme that suits you.

Packages are standard practice, as per document preparation.

Exercise 55. *Describe explicitly the use of* `\usetheme{}`, `\usecolortheme{}` *and* `\usefonttheme{}`.

Exercise 56. *With reference to Exercise 55, explain what colour options do for the* ***beamer*** *document class. That is, typing* `\documentclass[red]{beamer}` *as an example.*

Exercise 57. *a) LaTeX generates navigation symbols. What do they do?*

b) Add `\beamertemplatenavigationsymbolsempty` *to the preamble. What happens to the slides?*

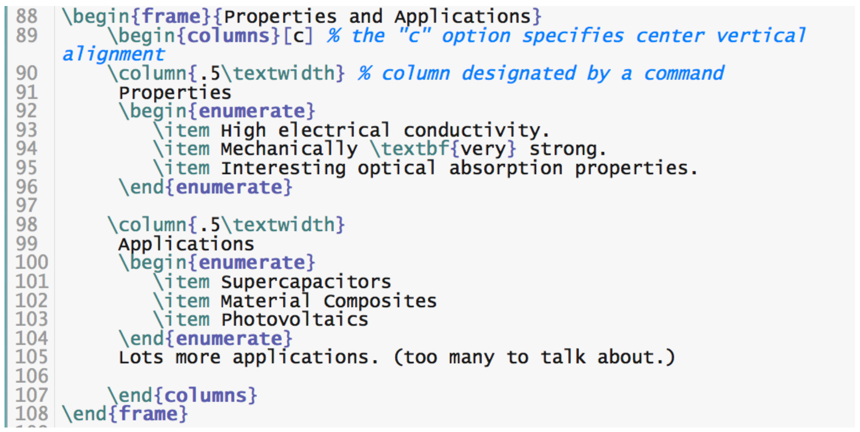
The second control you will want to have is the build itself. That is, how do you want the slides to progress? This involves the use of the `\pause` command. Explicitly, present everything up to `\pause`, until you click on your .pdf viewer to jump to the next stage in builds[2].

[1] If you refer to older books, they approach slides through an entirely different method and make use of different commands. Today's community standards generally support Beamer, so keep with Beamer solutions.

[2] Warning! Beamer works well with the free Adobe PDF, but doesn't work well on Preview, a Mac PDF viewer. Buyer beware!

5.2 Frames

LaTeX calls its slides as **frames**[3]. To show how it is done, a generic example is shown as:
`\begin{frame}{frametitle}`
`contents`
`\end{frame}`.
Replace the contents with `\titlepage` for the title page, as an example. The frame title shows up according to what it was given.
Table of contents work similarly, but Beamer offers an additional option of generating a "progress chart". While a full table of contents can be generated normally, a table of contents highlighting only the present section can be done by adding the option `\tableofcontents[current]`. Sometimes we wish the split the slide into two columns. That is easily done by the following syntax as shown in Fig. 5.2.



```
\begin{frame}{Properties and Applications}
    \begin{columns}[c] % the "c" option specifies center vertical
alignment
    \column{.5\textwidth} % column designated by a command
     Properties
     \begin{enumerate}
        \item High electrical conductivity.
        \item Mechanically \textbf{very} strong.
        \item Interesting optical absorption properties.
     \end{enumerate}

    \column{.5\textwidth}
     Applications
     \begin{enumerate}
        \item Supercapacitors
        \item Material Composites
        \item Photovoltaics
     \end{enumerate}
     Lots more applications. (too many to talk about.)

    \end{columns}
\end{frame}
```

Figure 5.2: Writing two columns in Beamer. Note the syntax.

Note that this is dependent on the earlier set of customisations we set Beamer to do in Fig. 5.1. All other commands behave as standard ones. Using this format, there also exists a fairly cute way of labelling figures using this syntax in Fig. 5.3.

[3]You might have heard of this term in Flash as well.

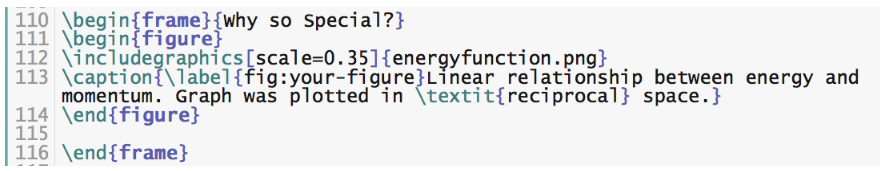

```
\begin{frame}{Why so Special?}
\begin{figure}
\includegraphics[scale=0.35]{energyfunction.png}
\caption{\label{fig:your-figure}Linear relationship between energy and
momentum. Graph was plotted in \textit{reciprocal} space.}
\end{figure}

\end{frame}
```

Figure 5.3: Inserting graphics. Note that figures are automatically centered in Beamer. The caption and label are also treated slightly differently here.

Exercise 58. *Typeset Fig. 5.2. What is its equivalent in a presentation software, if any?*

We now come to one of the important parts in Beamer: attracting audience attention. Typically slides carry at best a key point. To emphasise the key point, or to note something in general, one can enclose the text in a block using the **block** environment. This is invoked using `\begin{block}{}` and `\end{block}`. The second set of braces works analogously to the set for frames; that set gives the block a title. Everything else is automatic.

Exercise 59. *Investigate the use of the* `alertblock` *and* `exampleblock` *besides* `block` *in the same fashion. Now you have three different block styles to play with.*

Also verify that blocks appear differently under different themes. Despite this, why is it still consistent with the syntax?

Exercise 60. *How should a blank slide be included in Beamer? (Reason: perhaps you want to put a big picture.)*

Beamer also possesses other interesting capabilities ranging from the ability to create pseudo-animation to embedding media within the slide[4]. Nevertheless, because of the different structure Beamer employs, some applications are much *simpler* (e.g. trying to write a simple program for animation), whereas other applications are more difficult (e.g. trying to float text in pictures and animation text boxes to move.).

[4]A good place to start is by checking TikZ-based and PGF-based animations.

Exercise 61. *a) Produce a slide that introduces bullet points in sequential order.*

Solution: The option `[<+->]` *after the* ***itemize*** *command.*

b) You can build a frame that generates many slides. This is done through overlays. In the ***itemized*** *list, one can specify when certain bullet points appear with the following:* `<x>`, `<-x>` *and* `<x->`. *These are to be placed immediately after the item. What should appear?*

Exercise 62. *How big/small are the font sizes in Beamer? How big is* `\footnotesize`*?*

Is that consistent with the article document class?

5.3 Conclusion

As a basic introduction, the chapter stops here to present the most elementary ideas involved in using Beamer as a slide preparation tool. There are many other cool gimmicks that Beamer can do that are highlighted either in the questions or through searching for them.

It should be pointed out to the reader that command over Chapter 3 and 4 will be essential in grasping more ideas through self-study. The reason behind this is due to the complexity in making slide templates resulting in slide style files. These can often be very detailed and complicated, leaving the reader to often guess a feature by modifying various parameters to observe the changes that happen to the output. It may prove to be intimidating to the introductory learner, but it is hoped that some fundamentals presented in this chapter will prove to be essential in uncovering more aspects of LaTeX in Beamer and beyond.

5.4 Chapter Exercises

Question 5.1. *Investigate the .sty files behind various Beamer templates. As an exercise, modify the .sty files and observe what changes and why. For instance, how is colour defined in such .sty files? Beamer templates are readily available on ShareLaTeX and WriteLaTeX for free. Attempt to use one of these and notice the differences in how the slide lays out, the features present/absent, as well as the .sty files used in defining the slides.*

Question 5.2. *You may have noticed Beamer has only a few standard configurations. How will you propose manual adjustment, e.g. of colour or of the font style to be used? (Hint: refer to the .sty files.)*

The .sty files contain many defaults that can be adjusted. That is, if the template makes use of external files to control them. Generally this is trial and error, and even the author will not claim to be able to adjust these particularly efficiently.

Some Notes *The book does not go in depth into Beamer due to the nature of an introductory course. Nevertheless, here are some cool resources:*

1. `http://www.uncg.edu/cmp/reu/presentations/Charles%20Batts%20-%20Beamer%20Tutorial.pdf` *provides a nice Beamer tutorial – in Beamer of course!*

2. `http://www2.informatik.uni-freiburg.de/~frank/ENG/latex-course/latex-course-3/latex-course-3_en.html` *provides templates so that you can build up Beamer knowledge systematically.*

There are also plenty of Beamer-related sources one can search for. Both ShareLaTeX and WriteLaTeX have Beamer templates that can be played with. Adjusting the defaults can often lead to illuminating discoveries.

Question 5.3. *Google the following equivalents in presentation software:*

1. *Writing slide notes.*

2. *Hiding text.*

3. *Animation (this can be very tricky, and is left as a fun exercise).*

Disclaimer: We will not attempt to work out substitutes for features not too useful in most work such as WordArt.

Chapter 6

Drawing*

(This chapter will cover selected topics in the book: The LaTeX Graphics Companion, 2nd Edition and various material drawn from open source material such as ShareLaTeX tutorials and http://www.texample.net. This is not supposed to be extensive.)

Drawing is an extensive feature in LaTeX. A basic course will not do justice to its extensive capabilities in figure and diagram typesetting, but some ideas will be explored.

You may have recalled from the mhchem package the use of **TikZ** (pronounced Tik-Zee) to draw a double-headed arrow. **Tikz** is the starting point of many drawing applications according to a LaTeX practitioner's point of view.

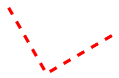
6.1 What Can LaTeX Draw and Manipulate?

Recall that graphics can be added using the `\includegraphics[]{}` command. However, LaTeX has its own drawing engine. To most LaTeX users, **TikZ** offers enormous capability to allow for LaTeX to draw graphics.

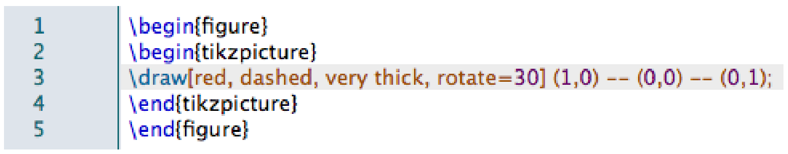
6.2 TikZ

(Section 6.2 corresponds to `https://www.youtube.com/watch?v=vccMP_LSYcs` and `https://www.youtube.com/watch?v=lVsk-AXGE6c` in the course.)

Tikz drawing makes use of Cartesian (x,y) or polar $(\theta:r)$[1] coordinates. Note the use of the respective punctuation for the coordinate types (comma for Cartesian, colon for polar). These are enclosed in a **tikzpicture** environment, as shown in Fig. 6.1.



(a) First picture drawn using the TikZ package.



```
\begin{figure}
\begin{tikzpicture}
\draw[red, dashed, very thick, rotate=30] (1,0) -- (0,0) -- (0,1);
\end{tikzpicture}
\end{figure}
```

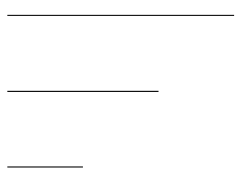
(b) Syntax for the material drawn in Fig. 6.1a.

Figure 6.1: Introduction to drawing lines.

Exercise 63. *Replicate the figure in Fig. 6.1a using polar coordinates and dropping the* ***rotate=30*** *part (why can I do that?). Modify the colour to be blue and non-dashed. Check if you can draw thin and very thin line segments.*

If we want to close the figure, we simply add `-- cycle` behind the last vertex. Multiple ways of drawing exist, as the examples below will show.

[1]The syntax for polar coordinates differs from the typical $(r:\theta)$ one!


(a) Three horizontal line segments.

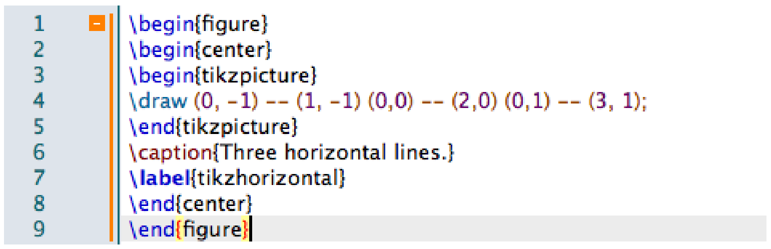

```
\begin{figure}
\begin{center}
\begin{tikzpicture}
\draw (0, -1) -- (1, -1) (0,0) -- (2,0) (0,1) -- (3, 1);
\end{tikzpicture}
\caption{Three horizontal lines.}
\label{tikzhorizontal}
\end{center}
\end{figure}
```
(b) Syntax for the material drawn in Fig. 6.2a.

Figure 6.2: More line segments.

Note that coordinates separated without a `--` between them means movement without drawing.

Sometimes a projection of a line from (x_1, y_1) to (x_2, y_2) in both horizontal and vertical axes is desired. An example is in Fig. 6.4. Notice how it can be done in two different ways, just by a slight change of syntax.

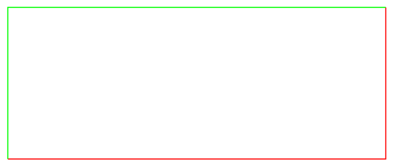
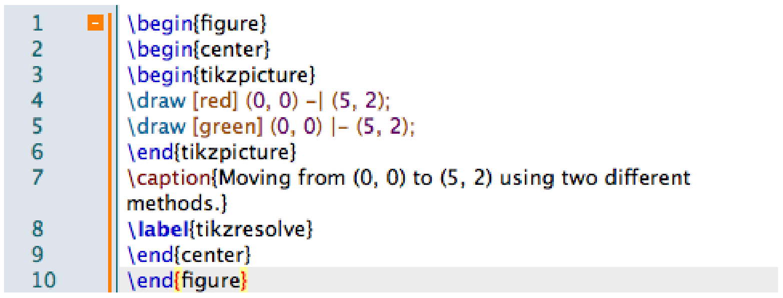


Figure 6.3: Moving from (0, 0) to (5, 2) using two different methods.



```
\begin{figure}
\begin{center}
\begin{tikzpicture}
\draw [red] (0, 0) -| (5, 2);
\draw [green] (0, 0) |- (5, 2);
\end{tikzpicture}
\caption{Moving from (0, 0) to (5, 2) using two different
methods.}
\label{tikzresolve}
\end{center}
\end{figure}
```

(a) Syntax for the material drawn in Fig. 6.3.

Figure 6.4: Even more lines.

The exercises below explore more about lines.

Exercise 64. *a) Google what a Bezier curve is.*

b) LaTeX can draw Bezier curves. Type the following into the ***tikzpicture*** *environment to understand what it does.*
`\draw (0,0) .. controls (1,1) .. (4,0)`
`\draw (5,0) .. controls (6,0) and (6,1) .. (5,2);`
Interpret your results.

Exercise 65. *Attempt the following syntax in the* ***tikzpicture*** *environment.*
a) `\draw (0,0) to (3,2);`
b) `\draw (0,0) to[out=90,in=180] (3,2);`
c) `\draw (0,0) to[bend right] (3,2);`
Interpret your results. (Note that `to` *and* `--` *are equivalent in output here.)*

The next natural progression from lines is shapes. Of which, regular shapes such as rectangles and ellipses are of interest. The following example begins with two simple rectangles.

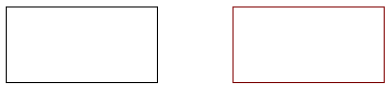


Figure 6.5: Two simple rectangles

The general syntax for a rectangle is: `\draw (x1,y1) rectangle (x2,y2);`, where you specify the coordinates for where you want the two corners of the rectangle to be at.

Exercise 66. *a) Reproduce Fig. 6.5 yourself. Hint: the red is generated by the option* `[draw=red!50!black]`*.*

b) Besides just the drawing, the shading can be customised too. The syntax `[draw=red!50!black]` *means: to mix 50 % of red and the rest be made up with black. If you want both fill and drawing, you can use the fuller command below:*

```
\filldraw[fill=?, draw=?] (x1,y1) rectangle (x2,y2);
```

This provides you the opportunity to create your own custom rectangles by specifying the fill and draw parameters.

c) Interpret the meaning of this line:

```
\shade[top color=blue, bottom color=red] (0,0) rectangle (2,1);.
```

Circles and ellipses can be drawn using similar concepts and ideas.

Exercise 67. *How many arguments will you expect for circles and ellipses? Will their options be identical?*

The exercise above should give you a sensing on what to expect for shapes you want to draw. There are two methods of doing this. The evaluation of the mark-up language and the interpretation is left as an exercise below for the reader.

Exercise 68. *The syntax for a circle or ellipse is*

```
\draw (x,y) circle [radius=r];
```

where the ordered pair (x, y) refers to the centre of the circle and r the radius. If an ellipse is demanded, replace `[radius=r]` *with*

```
[x radius=x_r, y radius=y_r].
```

More options can be specified, e.g. rotation (to be specified alongside radii options). Explore the commands and make use of knowledge in drawing of rectangles to create your own shapes.

Google Exercise: how can an arc be drawn?

Functions can be plotted on LaTeX as well! This is due to the fact that LaTeX has a mathematical engine. LaTeX has programming capability which should have been seen when it was employed to solve a Sudoku puzzle. Refer to Question 6.5 for more information. What is next after lines? Nodes, of course.

6.3 Nodes

(Section 6.3 corresponds to `https://www.youtube.com/watch?v=tLllmWjIBRQ` in the course.)

Nodes are used in defining points. These are the building blocks in charts and graphs, setting the foundation for many other applications.

A note is a point. A description can be given to a node (either a label, identity or both). They are not included on the path: nodes are only defined after the path has been drawn. In simple LaTeX syntax, a node is defined as:

```
node[<options>](<name>){<text>}
```

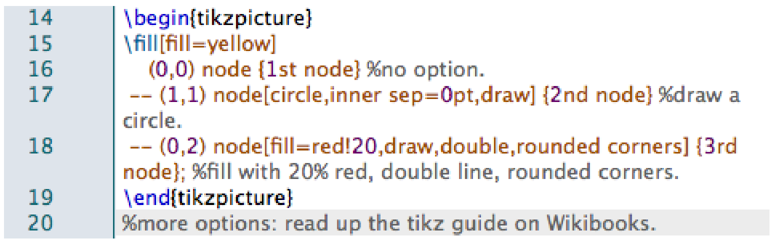
Sometimes you want no text to appear. Two options exist, simply leave no text, or invoke:

`coordinate[<options>](<name>)`

Practise comes in the form of a first exercise of creating nodes.

Exercise 69. *Invoke a **draw** command, connect the points* $(0,0)$*,* $(1,1)$ *and* $(0,2)$ *with dotted lines, and label them 1st, 2nd and 3rd node respectively. (Solution in Fig. 6.6, minus the options.)*

Next, some options are introduced, as shown in Fig. 6.6.



```
\begin{tikzpicture}
\fill[fill=yellow]
   (0,0) node {1st node} %no option.
 -- (1,1) node[circle,inner sep=0pt,draw] {2nd node} %draw a circle.
 -- (0,2) node[fill=red!20,draw,double,rounded corners] {3rd node}; %fill with 20% red, double line, rounded corners.
\end{tikzpicture}
%more options: read up the tikz guide on Wikibooks.
```

Figure 6.6: Customisation at work.

Exercise 70. *Typeset Fig. 6.6 to see what you get.*

Sometimes, the points which need definitions are on lines. In that case, the position of the node on the line can be specified with a `[pos=?]` syntax, which leads to the number line application.

Exercise 71. *Draw a number line. After the number line syntax, just before ending the **tikzpicture**, invoke 3 node commands: 1 for the start (*`[pos=0]`*), 1 for the mid-point (*`[pos=0.5]`*) and 1 for near-end (*`[pos=0.75]`*). Label these points and typeset the result. Interpret your results.*

There exists abbreviations for some positions, which are in the wikibook for TikZ.

To have the text sloped tangentially to the number line, we type `sloped` as an option. However, the numbers are typeset on the line! This can be changed be changing the anchor position. One set of options: `right`, `left`, `above` and `below`, all of which are self-explanatory. (The other set: `anchor=?`

where the anchor is to be defined with respect to a geographical direction.)

Exercise 72. *Go back to Exercise 71. Make some corrections to shift the fractions.*

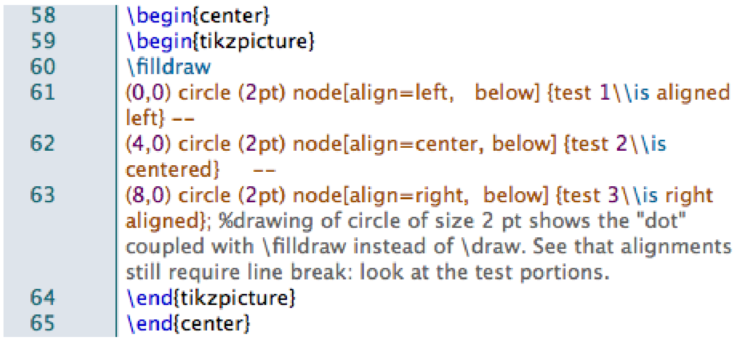
Since nodes are often the only path operation on paths, there are special commands for creating paths containing only a node, the first with text ouput, the second without:

```
\node[<options>](<name>) at (<coordinate>){<text>};
\coordinate[<options>](<name>) at (<coordinate>);
```

An example: `\path (0,0) node(x) {};`[2] will define a point $(0,0)$ and name that point as a node labelled `x`. We can label another node, and then invoke `\draw (x) -- (y);`. This is one way to draw a line with nodes.

Exercise 73. *Draw a line using the* `\coordinate` *method.*

Sometimes, nodes need multiple lines for labelling (for neatness). A neat example is shown in Fig. 6.7 as an end to this section.



```
\begin{center}
\begin{tikzpicture}
\filldraw
(0,0) circle (2pt) node[align=left,   below] {test 1\\is aligned left} --
(4,0) circle (2pt) node[align=center, below] {test 2\\is centered}     --
(8,0) circle (2pt) node[align=right,  below] {test 3\\is right aligned}; %drawing of circle of size 2 pt shows the "dot" coupled with \filldraw instead of \draw. See that alignments still require line break: look at the test portions.
\end{tikzpicture}
\end{center}
```

Figure 6.7: Code for multi-line text aligned left, centre and right.

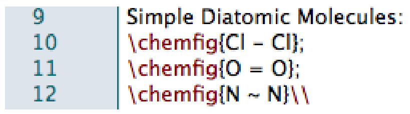
Exercise 74. *Compile Fig. 6.7. Observe what happens.*

[2]Be careful with the semicolons.

6.4 Chemistry Application: Structural Formulae

(The 2014/2015 course did not cover this section.)

Structural formulae is often the source of frustration for a typesetter[3]. This makes use of the **chemfig** package. The initial discussion starts with single, double and triple bonds shown in Fig. 6.8.



```
Simple Diatomic Molecules:
\chemfig{Cl - Cl};
\chemfig{O = O};
\chemfig{N ~ N}\\
```

Figure 6.8: Bond syntax for single, double and triple bonds. The semicolons are just for separation, and do not carry special meaning.

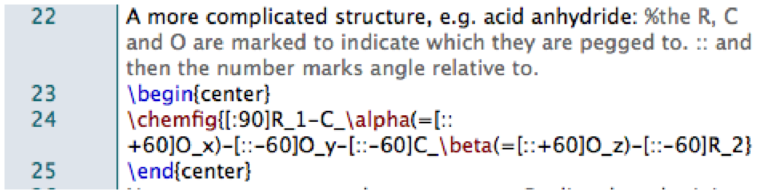
Linear molecules can be typeset quite trivially, but non-linear molecules require instructions to create bond angles. This is done by defining angular displacement from the $+x$-axis in degrees using the syntax `\chemfig{H-[:30]X}` (as an example). The `:30` refers to the angle measured from the $+x$ axis. As an example, typing `\chemfig{H-[:35]O-[:-35]H}` yields H_2O.

Exercise 75. *a) Accurately draw the chemical structure of* H_2O_2.

b) You may fuss that the bonds are too close together. Add `\setbondoffset{1mm}` *before your syntax. Does the situation improve?*

Naturally, an organic chemist wants to draw chains of molecules. The relative angle is better than defining the angle with respect to the horizontal, and is invoked by `[::x]` instead. x specifies the angle of rotation about the previously defined bond. Moreover, groups may stick out of the main chain. Example: an acid anhydride which has 2 R-groups (defining the main chain) and the doubly-bonded O (defining a branch chain), as typeset in Fig. 6.9.

[3]Hopefully it becomes a joy after the course.



```
A more complicated structure, e.g. acid anhydride: %the R, C and O are marked to indicate which they are pegged to. :: and then the number marks angle relative to.
\begin{center}
\chemfig{[:90]R_1-C_\alpha(=[::+60]O_x)-[::-60]O_y-[::-60]C_\beta(=[::+60]O_z)-[::-60]R_2}
\end{center}
```

Figure 6.9: One method of mark-up and result for the acid anhydride.

The result, quite naturally is this:

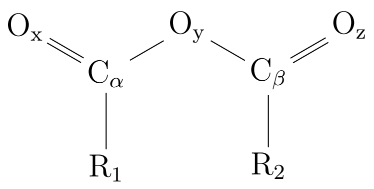


Figure 6.10: Acid anhydride structure from what you have in Fig. 6.9.

Note: Drawing the acid anhydride skeleton should start with the longest chain. In the case of the example, it starts from the $R_1 - C_\alpha - O_y - C_\beta - R_2$. The `[:90]` in front means: to rotate the figure 90 degrees anti-clockwise with respect to the $+x$ direction, fixing the starting orientation, and starting at R_1, linking to C_α with a single bond of 0 degrees with respect to starting orientation. A branch chain to O_x is introduced, and is enclosed using parentheses. The main chain continues with O_y and the necessary steps are repeated, with the use of only relative angles.

The subsequent exercises that follow will involve typesetting chemical formulae and the introduction of some nifty features.

Exercise 76. *Typeset the same acid anhydride without invoking the overall rotation command. Which is easier?*

Exercise 77. *Some molecules are cyclic. Draw a cyclic molecule.*

Hint: You'll need to invoke the "hook", which can be found in the ***chemfig*** *package manual.*

Exercise 78. *Refer to Page 13 of the manual. Non-planar structures can be drawn by drawing bonds into and out of the plane. Draw an example of a chiral molecule, including its mirror image.*

Exercise 79. *a) Remove the "C"s from the acid anhydride code as in Fig. 6.9. Interpret your observations.*

b) Exercise 75 had an offset command. Add that into your molecule here. Does anything happen?

The method of drawing cyclic molecules in Exercise 77 is tedious, which demands a better method, as shown in Exercise 80.

Exercise 80. *a) Typeset* `\chemfig{A*5(-B=C-D-E=)}`.

b) From (a), perform the following modifications: Change the number of members of the ring, break the ring and add more members than what can be contained[4].

How should benzene rings be typeset? This starts with the general syntax; the full syntax for drawing within the structure is given by:

```
<atom>**[<angle 1>,<angle 2>,<tikz>]<n>(<code>)
```

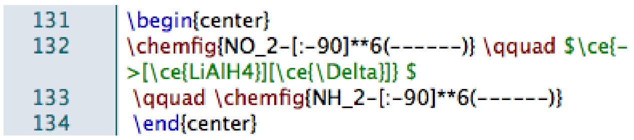
(for you to experiment). Instead, 2 specific examples as exercises for the reader allow for deeper investigation.

Exercise 81. *Typeset the following and comment on your observations. (It will be easier to comment by varying the parameters.):*
i) `\chemfig{**6(------)}`
ii) `\chemfig{**[30,330]5(-----)}`
iii) `\chemfig{[0,270,dash pattern=on 2pt off 2pt]4(----)}`

Exercise 82. *Rotate the ring for the other common configuration. Observe that the syntax builds on.*

[4]Doesn't succeed.

Besides just benzene rings, substituents are important, as shown in Fig. 6.11:



```
\begin{center}
\chemfig{NO_2-[:-90]**6(------)} \qquad $\ce{->[\ce{LiAlH4}][\ce{\Delta}]} $
 \qquad \chemfig{NH_2-[:-90]**6(------)}
 \end{center}
```

Figure 6.11: Reduction of nitrobenzene.

This should yield a slightly slipshod, but acceptable reduction as in Fig. 6.12. (Interjection: How and why is it slipshod?)

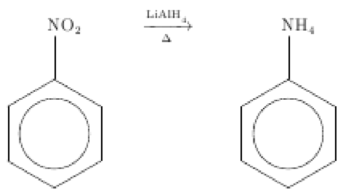
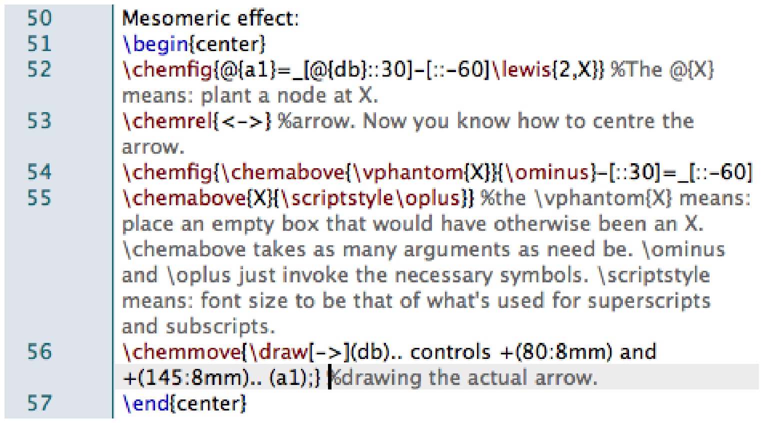


Figure 6.12: The results of typesetting Fig. 6.11.

(Note: Multiple rings and nested rings is beyond the scope of this course. A reader is advised to read Sections 10.4 and 10.5 of the **chemfig** package manual for drawing them.)

To deal with dynamics, arrows are typically used to show reaction schemes. Their starts and ends must be defined. The syntax `@{<argument>}` will be used. This allows for the placement of a **TikZ** node. It may sound complicated due to the brevity of **TikZ** coverage, glossing over the inner workings, so a step-by-step example (as per the one given in the **chemfig** documentation) is in order.

Consider the following in Fig. 6.13:



```
Mesomeric effect:
\begin{center}
\chemfig{@{a1}=_[@{db}::30]-[::-60]\lewis{2,X}} %The @{X} means: plant a node at X.
\chemrel{<->} %arrow. Now you know how to centre the arrow.
\chemfig{\chemabove{\vphantom{X}}{\ominus}-[::30]=_[::-60]
\chemabove{X}{\scriptstyle\oplus}} %the \vphantom{X} means: place an empty box that would have otherwise been an X. \chemabove takes as many arguments as need be. \ominus and \oplus just invoke the necessary symbols. \scriptstyle means: font size to be that of what's used for superscripts and subscripts.
\chemmove{\draw[->](db).. controls +(80:8mm) and +(145:8mm).. (a1);} %drawing the actual arrow.
\end{center}
```

Figure 6.13: Syntax for a simple example of a mesomeric effect.

First, draw the first compound, but add a labelling point (start point (with `@{db}`) and end point (with `@{a1}`)) as an arrow will be drawn later. We recall the angular displacement from horizontal, invoking the `\lewis{}` command. A `\chemrel{<->}` shows the bidirectional relation between this and the upcoming figure. The second `\chemfig` makes use of a "phantom" box to introduce equivalent empty space. In this case, we use `\vphantom{X}`. `\chemabove{}` just refers to the point above that area in the picture.

The the `\chemmove` [5] command starts the drawing of the arrow. Here comes the reason for the earlier definition: we start the unidirectional arrow from the double bond. Polar coordinates are given [6]. The user should be aware now that the word **controls** specifies two control points, while coordinates specify how the arrow is drawn, as shown in Fig. 6.14.

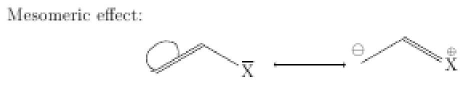


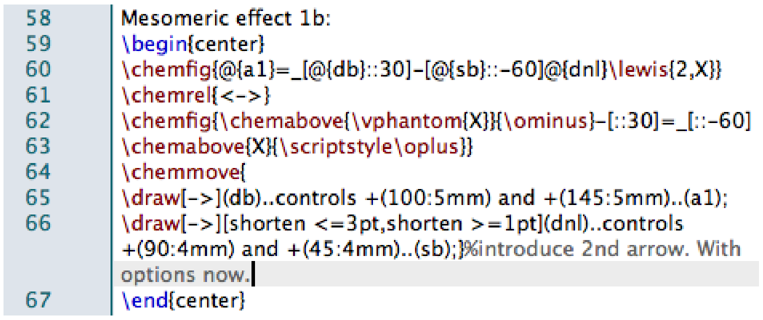
Figure 6.14: Typesetting Fig. 6.13 yields this.

There is a lot of hard work done for little result. However, the LaTeX machin-

[5]This command calls for a tikzpicture environment and in addition, remembers the original picture, and overlays on the existing picture.

[6]Reminder: Coordinates in LaTeX are defined with the angle first, and then the modulus separated with a colon!

ery in generating pictures is worth studying. Building on from this example, a second arrow in Fig. 6.14 is drawn, as shown in Fig. 6.15.



```
Mesomeric effect 1b:
\begin{center}
\chemfig{@{a1}=_[@{db}::30]-[@{sb}::-60]@{dnl}\lewis{2,X}}
\chemrel{<->}
\chemfig{\chemabove{\vphantom{X}}{\ominus}-[::30]=_[::-60]
\chemabove{X}{\scriptstyle\oplus}}
\chemmove{
\draw[->](db)..controls +(100:5mm) and +(145:5mm)..(a1);
\draw[->][shorten <=3pt,shorten >=1pt](dnl)..controls
+(90:4mm) and +(45:4mm)..(sb);}%introduce 2nd arrow. With
options now.
\end{center}
```

Figure 6.15: Syntax. Same as Fig. 6.13, but with an extra line for a new arrow and the appropriate definitions. Notice a few new things involved in the new arrow.

The interpretation of these: the second arrow now comes in-built with options because we decide to shorten the arrow tail-end and head-end.

Exercise 83. *a) Draw the figure as per Fig. 6.15.*

b) What does shorten do? Manipulate the numbers.

c) Remove the option and interpret your findings?

Customisation of arrows is a **TikZ** exercise, with all the tools from **TikZ** open for use. An attempt is shown below:

`[-stealth,thin,dash pattern= on 2pt off 2pt,red]`.

This calls for a thin, dashed, red arrow. The dash is specified by 2 pt on and 2 pt off as the pattern; the stealth corresponds to the changing of the arrowhead. In addition, adding a π label can be done with:

`node[sloped,above] {$\pi$} (a1)`.

Exercise 84. *Here is a guided walkthrough on how arrows operate on **TikZ**.*

a) Draw:

```
\chemfig{@{x1}\lewis{1:,X}}
```

```
\hspace{2cm}
\chemfig{@{x2}\lewis{2:,X}}
\chemmove{\draw[->,shorten >=4pt]
(x1).. controls +(90:1cm) and +(90:1cm).. (x2);}
```

NOTE: There MUST be no line breaks when invoking `\chemmove`*, i.e. all code is to be confined to a single line! You should obtain an arrow drawn from the left to the right X.*

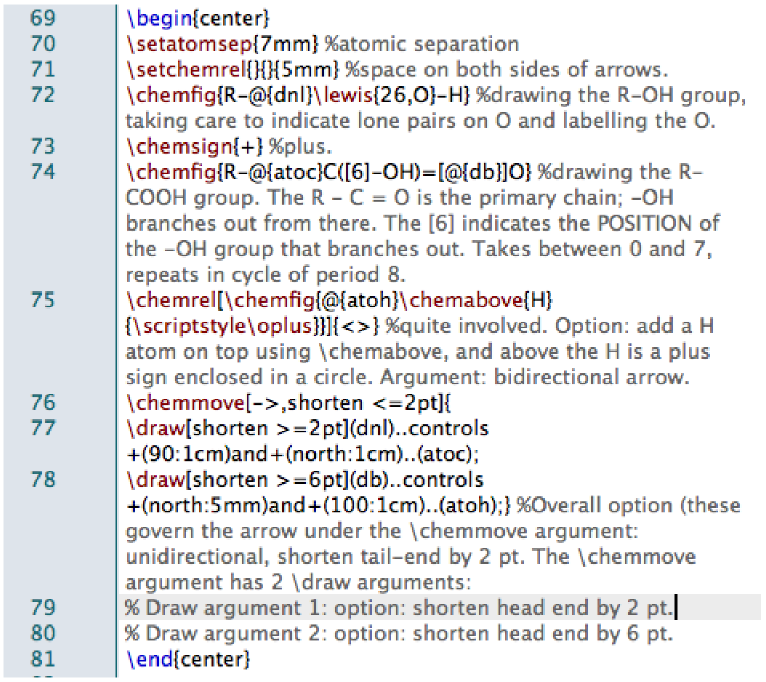
b) However that is not very accurate. The start and end direction of the arrows are incorrect. Replace the `\chemmove` *line with:*

```
\chemmove{\draw[->,shorten <=4pt,shorten >=4pt]
(x1.57).. controls +(60:1cm) and +(120:1cm).. (x2);}
```

How is this an improvement?

Note: View Question 6.7 for a more detailed walkthrough on arrows.

Before ending the section, an introduction to drawing reaction mechanisms for a single step will be introduced, as in Fig. 6.16.



```
\begin{center}
\setatomsep{7mm} %atomic separation
\setchemrel{}{}{5mm} %space on both sides of arrows.
\chemfig{R-@{dnl}\lewis{26,O}-H} %drawing the R-OH group,
taking care to indicate lone pairs on O and labelling the O.
\chemsign{+} %plus.
\chemfig{R-@{atoc}C([6]-OH)=[@{db}]O} %drawing the R-
COOH group. The R - C = O is the primary chain; -OH
branches out from there. The [6] indicates the POSITION of
the -OH group that branches out. Takes between 0 and 7,
repeats in cycle of period 8.
\chemrel[\chemfig{@{atoh}\chemabove{H}
{\scriptstyle\oplus}}]{<>} %quite involved. Option: add a H
atom on top using \chemabove, and above the H is a plus
sign enclosed in a circle. Argument: bidirectional arrow.
\chemmove[->,shorten <=2pt]{
\draw[shorten >=2pt](dnl)..controls
+(90:1cm)and+(north:1cm)..(atoc);
\draw[shorten >=6pt](db)..controls
+(north:5mm)and+(100:1cm)..(atoh);} %Overall option (these
govern the arrow under the \chemmove argument:
unidirectional, shorten tail-end by 2 pt. The \chemmove
argument has 2 \draw arguments:
% Draw argument 1: option: shorten head end by 2 pt.
% Draw argument 2: option: shorten head end by 6 pt.
\end{center}
```

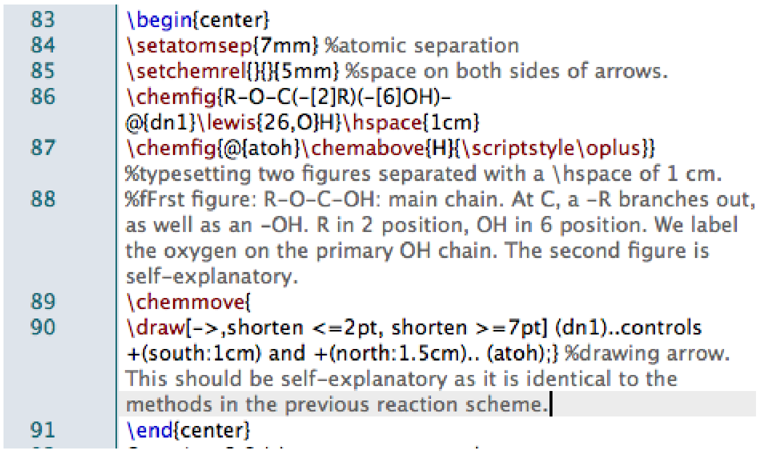
Figure 6.16: Code for esterification: Step 1

As Fig. 6.16 shows, this is not a simple task. The figure atomic separation and space between the arrows used between chemical species need to be defined. Therefore, two commands to set defaults are invoked, and then two figures are drawn, separated with a + sign. Notice the syntax difference and why `\chemsign{}` was invoked. Pay heed to the comments.

Exercise 85. *Typeset the syntax in Fig. 6.16. Interpret your results.*

The use of the `\chemabove{<code>}{<material>}` command does not change the dimensions of the bounding box of `<code>`. For this reason, there is a possibility of running into some difficulty in pointing to the symbol representing the charge carried. Unfortunately, much of this is trial and error [7]. The issue will be addressed in Exercise 86, where this is resolved.

[7]ChemDraw exists for this reason, but this shows you how to do simple chemical structures should you be stuck on a desert island.



```
\begin{center}
\setatomsep{7mm} %atomic separation
\setchemrel{}{}{5mm} %space on both sides of arrows.
\chemfig{R-O-C(-[2]R)(-[6]OH)-
@{dn1}\lewis{26,O}H}\hspace{1cm}
\chemfig{@{atoh}\chemabove{H}{\scriptstyle\oplus}}
%typesetting two figures separated with a \hspace of 1 cm.
%fFrst figure: R-O-C-OH: main chain. At C, a -R branches out,
as well as an -OH. R in 2 position, OH in 6 position. We label
the oxygen on the primary OH chain. The second figure is
self-explanatory.
\chemmove{
\draw[->,shorten <=2pt, shorten >=7pt] (dn1)..controls
+(south:1cm) and +(north:1.5cm).. (atoh);} %drawing arrow.
This should be self-explanatory as it is identical to the
methods in the previous reaction scheme.
\end{center}
```

Figure 6.17: Code for esterifcation: step 2

Exercise 86. *a) Typeset Fig. 6.17.*

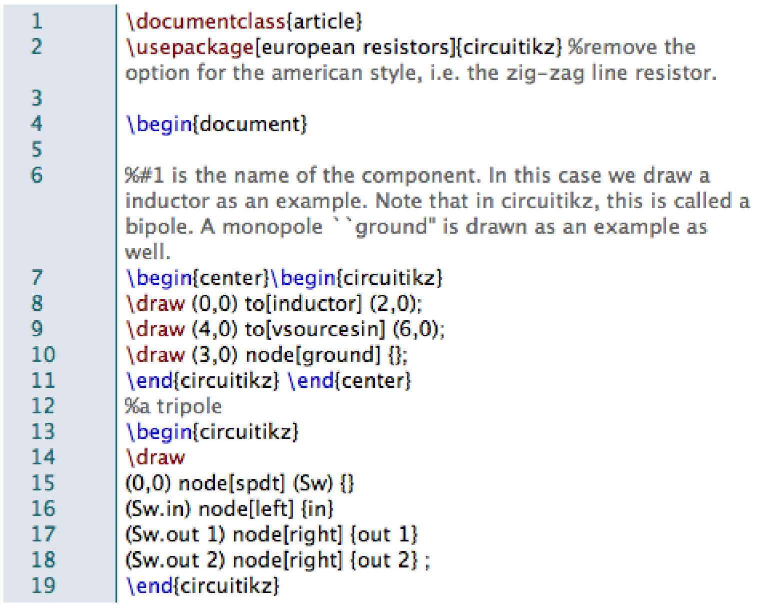
b) Complete the reaction.

The reader can afford to explore the **chemfig** package to find his or her specific needs. A supplementary question (Question 6.8) ends off the chemical structure discussion.

6.5 Engineering Application: Circuits

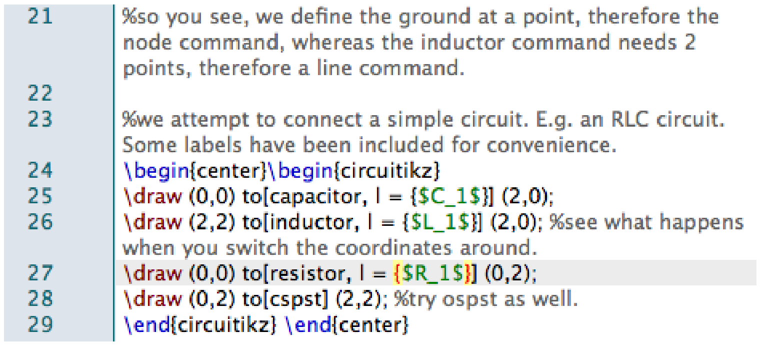
(Section 6.5 corresponds to `https://www.youtube.com/watch?v=cBZc0sHeqk4` of the course.)

Circuits are the cornerstone of any electrical engineer and anyone who has to typeset device configurations. This requires the **circuitikz** package, which borrows its syntax heavily from **TikZ**. The following examples in Fig. 6.18 and 6.19 will show the similarities.



```
\documentclass{article}
\usepackage[european resistors]{circuitikz} %remove the option for the american style, i.e. the zig-zag line resistor.

\begin{document}

%#1 is the name of the component. In this case we draw a inductor as an example. Note that in circuitikz, this is called a bipole. A monopole ``ground" is drawn as an example as well.
\begin{center}\begin{circuitikz}
\draw (0,0) to[inductor] (2,0);
\draw (4,0) to[vsourcesin] (6,0);
\draw (3,0) node[ground] {};
\end{circuitikz} \end{center}
%a tripole
\begin{circuitikz}
\draw
(0,0) node[spdt] (Sw) {}
(Sw.in) node[left] {in}
(Sw.out 1) node[right] {out 1}
(Sw.out 2) node[right] {out 2} ;
\end{circuitikz}
```

Figure 6.18: Introduction to Circuits: see how components that have two terminals are defined with 2 coordinates, similar to the line command in TikZ. The same applies to components with one terminal.



```
%so you see, we define the ground at a point, therefore the node command, whereas the inductor command needs 2 points, therefore a line command.

%we attempt to connect a simple circuit. E.g. an RLC circuit. Some labels have been included for convenience.
\begin{center}\begin{circuitikz}
\draw (0,0) to[capacitor, l = {$C_1$}] (2,0);
\draw (2,2) to[inductor, l = {$L_1$}] (2,0); %see what happens when you switch the coordinates around.
\draw (0,0) to[resistor, l = {$R_1$}] (0,2);
\draw (0,2) to[cspst] (2,2); %try ospst as well.
\end{circuitikz} \end{center}
```

Figure 6.19: A simple circuit with labelling.

6.6 Chapter Exercises

Note that unlike previous chapters, most problems here are guided and are quite independent of one another.

Question 6.1. *Some Basics on Graphic Types*

The way LaTeX draws is through vector graphics, which makes use of various primitives (e.g. node, line) written in a mathematical environment to produce graphics. Why is the use of vector graphics advantageous over typical bitmaps (which are called raster graphics). When are they not advantageous?

Note: The course has introduced most of the primitives used in vector graphics.

Question 6.2. *A Revision on Primitives*

Accurately draw the following figure.

- *A circle with centre* $(3,4)$ *and a radius* r *such that it intersects the origin. Label this graph* $(x-3)^2+(y-4)^2=r^2$*, where* r *is what you have to determine.*
- x *and* y *axes, with labels and appropriate arrowheads, with an appropriate domain to contain the whole circle.*
- *A dotted line from the origin to any point on the perimeter of the circle. Label this* $r=5$*.*

The axes should be thicker than the circle and dotted line (for this problem).

Comment on whether it is easier to precisely align the details compared to Paint, Photoshop or Powerpoint.

Question 6.3. *More on Including Graphics*

(This question is modified from the example in Chapter 2.1.3, The LaTeX Graphics Companion, 2nd Edition.)

The `\includegraphics[]{}` *command admits many options. Examples of these are:*

1. *bb: bounding box of the graphics image. You must then specify 4 dimensions, separated by spaces. These will tell what exactly is to be input as the graphic.*
2. *angle: specifies rotation angle*
3. *origin: specifies origin of rotation (the invariant point)*
4. *width/height: required width (width/height is scaled to the value)*
5. *keepaspectratio: a Boolean variable: "true" or "false".*

How do the various options work? Add them (example: `[height=20mm,width=40mm]`*) to some picture you have.*

Question 6.4. *Boxes*

(This question is modified from the example in Chapter 2.2.1, The LaTeX Graphics Companion, 2nd Edition.)

Scale boxes modify material in a local region. Some examples:

This text is at original scaling.

This text is scaled by a factor of 2.

This text is scaled by a factor of 0.5.

The syntax for this is `\scalebox{number}[option]{material}`*.*

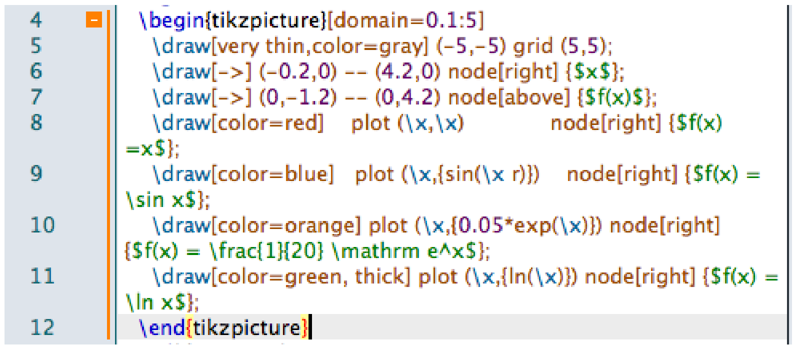
Set the number to -1 and option to 1. What happens? (This should perform identically to `\reflectbox{material}`*. Then, by plugging different numbers, interpret your findings.*

Question 6.5. *Drawing Functions on LaTeX*

(This question is modified from the LaTeX wikibook at *http://en.wikibooks.org/wiki/LaTeX/PGF/TikZ*.*)*

LaTeX can sketch simple functions[8] *using TikZ methods.*

An example is shown in Fig. 6.20.



```
\begin{tikzpicture}[domain=0.1:5]
  \draw[very thin,color=gray] (-5,-5) grid (5,5);
  \draw[->] (-0.2,0) -- (4.2,0) node[right] {$x$};
  \draw[->] (0,-1.2) -- (0,4.2) node[above] {$f(x)$};
  \draw[color=red]   plot (\x,\x)            node[right] {$f(x) =x$};
  \draw[color=blue]   plot (\x,{sin(\x r)})    node[right] {$f(x) = \sin x$};
  \draw[color=orange] plot (\x,{0.05*exp(\x)}) node[right] {$f(x) = \frac{1}{20} \mathrm e^x$};
  \draw[color=green, thick] plot (\x,{ln(\x)}) node[right] {$f(x) = \ln x$};
\end{tikzpicture}
```

Figure 6.20: Some functions that can be plotted on TikZ with their associated syntax.

a) Typeset the syntax in Fig. 6.20 and interpret the results. Try different functions as well.

b) In Fig. 6.20, the grid runs from $(-5,-5)$ to $(5,5)$, yet the domain is confined for all functions to start from 0.1. Why is it so? Modify the syntax to individualise the domains for all functions [9].

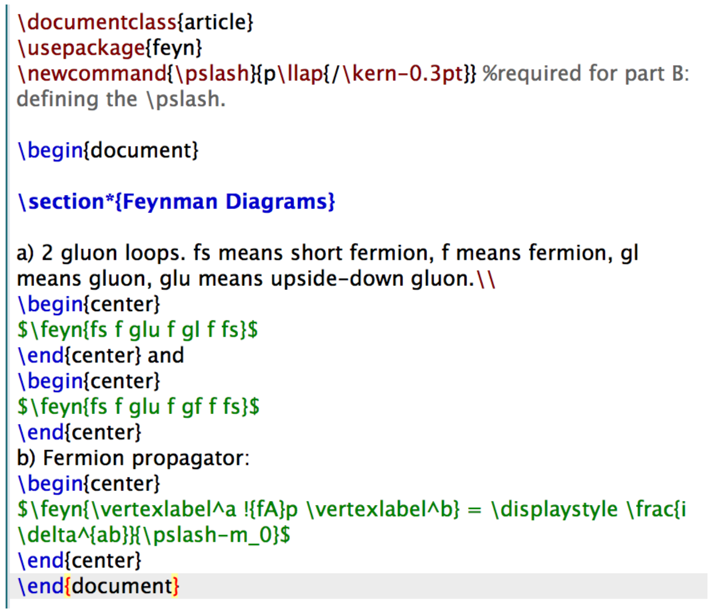
Question 6.6. *Feynman Diagrams*

(Read Chapter 8.4 of "The LaTeX Graphics Companion, 2nd Edition" for a more detailed account.)

Feynman diagrams can be drawn on LaTeX . Simplifying matters, we invoke the package ***feyn****. This question will deal with two simple examples as shown in Fig. 6.21.*

[8]If you want more complicated functions or a serious graphing solution, consider Origin or Mathematica.

[9]For $f(x) = \ln x$, only $x > 0$ is defined! The mathematical engine will choke on such an error if spotted.



```
\documentclass{article}
\usepackage{feyn}
\newcommand{\pslash}{p\llap{/\kern-0.3pt}} %required for part B:
defining the \pslash.

\begin{document}

\section*{Feynman Diagrams}

a) 2 gluon loops. fs means short fermion, f means fermion, gl
means gluon, glu means upside-down gluon.\\
\begin{center}
$\feyn{fs f glu f gl f fs}$
\end{center} and
\begin{center}
$\feyn{fs f glu f gf f fs}$
\end{center}
b) Fermion propagator:
\begin{center}
$\feyn{\vertexlabel^a !{fA}p \vertexlabel^b} = \displaystyle \frac{i
\delta^{ab}}{\pslash-m_0}$
\end{center}
\end{document}
```

Figure 6.21: Feynman diagram syntax.

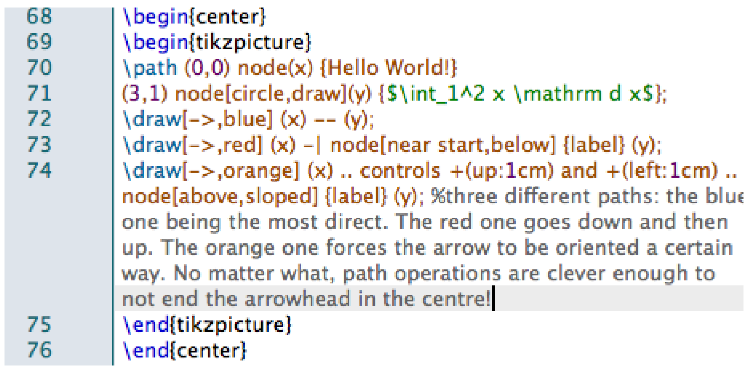
a) Typeset Fig. 6.21. Interpret the syntax.

*b) Search the **feyn** package and try your own Feynman diagrams.*

Note: There are multiple ways of drawing Feynman diagrams. This is one of the methods using a user-written package.

Question 6.7. *In the examples, only point notes were dealt with. However, some nodes demand a finite shape. The same operations can be done, such as drawing of an arrow to it. (This is of relevance to the **chemfig** package!)*

a) Typeset the code in Fig. 6.22 and interpret your results.



```
\begin{center}
\begin{tikzpicture}
\path (0,0) node(x) {Hello World!}
(3,1) node[circle,draw](y) {$\int_1^2 x \mathrm d x$};
\draw[->,blue] (x) -- (y);
\draw[->,red] (x) -| node[near start,below] {label} (y);
\draw[->,orange] (x) .. controls +(up:1cm) and +(left:1cm) ..
node[above,sloped] {label} (y); %three different paths: the blue
one being the most direct. The red one goes down and then
up. The orange one forces the arrow to be oriented a certain
way. No matter what, path operations are clever enough to
not end the arrowhead in the centre!
\end{tikzpicture}
\end{center}
```

Figure 6.22: Code for showing that path operations are “clever”.

b) Why might this be useful for drawing chemical diagrams?

Question 6.8. *This question brings you through using the* ***chemfig*** *package to label molecules.*

The syntax for naming molecules is:

```
\chemname[<dim>]{\chemfig{<code of the molecule>}}{<name>}
```

a) Typeset the following:

```
\chemname{\chemfig{R-C(-[:-30]OH)=[:30]O}}{Carboxylic acid}
                         \chemsign{+}
             \chemname{\chemfig{ROH}}{Alcohol}
                         \chemrel{->}
      \chemname{\chemfig{R-C(-[:-30]OR)=[:30]O}}{Ester}
                         \chemsign{+}
              \chemname{\chemfig{H_2O}}{Water}
```

b) Swap the order. What happens?

Note: The interested student will find it useful to refer to Section 12 of the ***chemfig*** *documentation.*

Appendix A

Useful Resources

LaTeX for Physicists: http://www.dfcd.net/articles/latex/latex.html: interesting resource that simplifies much of the typing physicists normally do.

Updating TeX : Once a year, you will need to update TeX with the latest updates. Please read http://tex.stackexchange.com/questions/55437/how-do-i-update-my-tex-distribution for a clear idea on how to go about this. Unfortunately in course time, we will not demonstrate it, since it is once a year, after all.

The TeX Stack Exchange, available at http://tex.stackexchange.com/: Very useful Q & A site.

LaTeX Wikibook at http://en.wikibooks.org/wiki/LaTeX: A nice reference material. Yes, it's a wiki, but so far the material in there looks reliable enough for day-to-day use.

LaTeX Project Page at http://latex-project.org/ftp.html: Self-explanatory. May be somewhere you'll return to when you get more advanced as well.

TeX Showcase at http://www.tug.org/texshowcase/: Very cool stuff!!

CTAN: Comprehensive TeX Archive Network at http://www.ctan.org/: A useful repository of LaTeX material.

The TEX Users Group (TUG) at http://tug.org/interest.html: Very useful resources. The group may be a paid group, but this page has many free resources which are invaluable to the TEX learner at different levels.

TEXample at http://www.texample.com. A **TikZ** support group with a number of interesting applications in **TikZ** worth exploring. Animation is one of its attractions.

Donald E. Knuth's Homepage at http://www-cs-faculty.stanford.edu/~uno/: Not the most useful resource, but a must-list since it is the homepage of the inventor of TEX and probably the guy you really need to thank (not the instructor) for this wonderful instrument.

Appendix B

Common Errors

(Chapter B corresponds to https://www.youtube.com/watch?v=A1dX3FzjDTU of the course.)

(Taken from "A TEX Primer for Scientists, Stanley A. Sawyer and Steven G. Krantz")

Sometimes LATEX comes back to you with a series of errors and warnings. This section goes through some of the typical ones you may encounter. However, error-hunting can be difficult.

B.1 Misspelling

An example of a spelling mistake in an argument is the following:

```
! Undefined control sequence.
l.11 \cemterline
                {by Otis P. Sito}
?
```

This means that LATEX doesn't recognise a command you asked it to, for e.g. the misspelled `\cemterline` which should have been `\centerline`.

B.2 Missing Syntax

Another common error: braces do not tally. Sometimes TEX tells you straightaway with an error like: `Too many }`. Often, it is not so simple. Sometimes like this could appear instead:

```
Runaway argument?
{by Otis P. Sito \medskip The transport potential of...
! Paragraph ended before \centerline was complete.
```

Obviously there is a missing close brace after `Sito`, but it may not be obvious at first reading of such errors. It may pay, therefore, to read through the errors closely; brace matters tend to be the subject of complicated error messages due to how LATEX interprets them.

Another common error could happen in the transit from an environment to another, (e.g. text to math). The simplest of them is forgetting the `$` for a math argument, in which case LATEX assumes you have a missing `$`, and then halt. The solution is simple: place the offending argument in `$` signs before and after the text.

Writing `\begin{}` without `\end{}` or vice-versa leads to errors along the lines of: `\begin{xxx} ended with \end{yyy}`.

Often this point leads to another point of good typesetting. Spaces can easily be used at the front of the line to distinguish between levels of `\begin{}` and `\end{}` you use, such as nested environments. Also, some TEX editors, for e.g. TEX maker does a good job at auto-typing the `\end{}` that corresponds to the `\begin{}` you introduce for default commands.

Sometimes you can't find an error. Try the following:

1. Take out chunks of code that you know stand alone. Basically, this is troubleshooting the document part by part.

2. Remove all files LATEX creates in the folder, and then recompile. This means files like the auxiliary file (`.aux`), the log file (`.log`). DO NOT remove files such as class files or package supplementary files, or your

own working file itself. That means, keep the `.tex`, `.cls` and `.sty` files. For the latter two, if you have them in the first place. This often arises when using someone else's custom template. (Reason: errors may have been resolved, but history lingers.)

Typesetting in LaTeX is different from Word. Did you remember to add `\` before special characters like `&` and `%`?

Another common error is forgetting to include certain packages but using commands from those packages. The error generated would be

"`! Undefined control sequence`"

, analogous to that of Section B.1.

B.3 Missing Objects

Remember that your figures, bibliography and other supplementary files have to be in the same folder as your LaTeX files under standard directories! Otherwise LaTeX simply complains of missing objects by saying that it cannot find various files.

Sometimes, if you have to debug the integrity of code and run into missing objects, yet you know these are not the source of error (e.g. you are to investigate issues pertaining to a syntax issue, but a friend sends a pile of graphic inputs), you may just comment out such error-prone lines.

References

This compilation would not have been possible if not for the good books on LaTeX written out there. In particular, I would like to give credit to a few books and good notes written by some authors, without which this set of notes would never have been written. Many have already been acknowledged in Appendix A, leaving the rest to be acknowledged here.

http://download.nus.edu.sg/mirror/CTAN/info/lshort/english/lshort.pdf: This is standard material.

Guide to LaTeX by Kopka and Daly: A good book that lays the formalisms of LaTeX that also includes understanding how it works beyond "just doing".

The LaTeX Graphics Companion by Goossens, Mittelbach, Rahtz, Roegel and Voβ: An exploratory text into what LaTeX can do. However, at least from the version I read (ver. 2), much of the syntax there has been superceded by later material.

A TeX Primer for Scientists by Sawyer and Krantz: Simple yet understandable.

Index